

Редакция Алексея Кривицкого

ЗАДАЧИ

1. Редактирование текста - в процессе
2. Редактирование иллюстраций
3. Заполнение глоссария в Предисловии к русскому изданию

ГЛОССАРИЙ

Оригинальный термин	Перевод	Комментарии
Framework Framework Scrum Framework LeSS	(вместо инфраструктура)	Термин "каркас" принят в agile-среде. Используется в книгах, статьях. Вместе с заимствованием "фреймворк". Перевод как "инфраструктура" не корректен, так как под иной понимается внутреннее устройство (к примеру инфрастр

		<p>уктура офиса). Смысл слова "framewor k" от "frame" понимает , что даётся некоторая рамка процесса (набор принципо в и правил), но внутренн ость имеет пустоты, который должен заполнит ь пользоват ель каркаса. Отюда и "inspect and adapt" - необходи мость адаптаци и процесса под себя.</p> <p>Существенная правка.</p>
Management practices	Практики управления (вместо нормы практик управления)	Так ближе к оригинальному смыслу и проще звучит.
Experimentation mindset	нацеленность на непрерывное экспериментирование	Звучит лучше, чем "умонаст роение"
Product groups Large product groups	Продуктовые группы (вместо групп продуктов или больших групп продуктов)	Имеются в виду "кросс-функциональные отделы" внутри предприятий, а не группы продуктов.

		Но слово "отделы" тут использовать некорректно, из-за не-иерархичности структуры LeSS-организации.
LeSS rules and LeSS guides	Правила LeSS и руководства LeSS (вместо правил и наставлений LeSS)	Scrum Guide переводится как "руководство по Scrum". А значит guide - руководство. Слово "наставление" имеет немного религиозную коннотацию (наставления буддийских философов)
Product Backlog	Беклог продукта	Я всего пишу именно через "е", а не "э". Кажется, так эстетичнее.
Fist-level manager (line manager)	руководитель первого звена (вместо руководителя низшего уровня)	
Product	продукт (вместо продукции)	Имеется в виду программный продукт или продукт как результат интеллектуальной работы. "Продукт инженерной мысли". Это не производство, это инженерия. Термин же "продукция" применим к производству (эта книга про инжиниринг, а не производство и сбыт).
Inspect and Adapt	Инспекция и адаптация (вместо обследования и адаптации)	Узнаваемая, устоявшаяся манTRA Scrum.
Defined process(es)	Детерминированный процесс(ы) (вместо "определенный процесс")	Определенный - это тоже правильный перевод, но "определенный процесс" может иметь в виду "этот, а не тот", "конкретный". Здесь же defined - как противовес empirical.
Cross-function team	кросс-функциональная Команда	Устоявшийся термин
Team lead	руководитель команды	
Scrum Master coaches team	Scrum-мастер коучит или Scrum-мастер выступает в роли коуча. (вместо "роли наставника". Наставник это mentor. И в LeSS это другая функция).	Это устоявшиеся термины: coach и mentor (коуч и ментор/наставник). Это разные функции и методы.
Potentially Shippable Product Increment	<i>потенциально готовый к поставке</i> <i>инкремент продукта</i>	

(PSPI)		
Empirical Process Control	Эмпирический контроль процесса (вместо контроль эмпирического процесса).	Это очень существенная правка - мы эмпирически (на основании собираемой информации контролируемы процессы, как вождение автомобиля). Основа Скрам.
cross-functional, cross-component, full-stack feature teams	кросс-функциональные, кросс-компонентные, полно-стековые, продукто-ориентированные команды	
Offshore outsourcing	Аутсорсинг и удалённая разработка	
Orhanizational Design	организационной дизайн (вместо организационного проектирования)	Устоявшееся названия раздела коучинга организаций.
Multi-site environment	много-офисная среда	
Multi-site teams	Команды разных офисов или много-офисные команды (вместо географически-разнесенные команды много-офисные команды географически-разнесенные много-офисные команды)	Есть еще "распыленные команды" - dispersed teams.
Shared code environment	среда с общим кодом	
done items	готовые элементы Беклога Продукта	без приписки "Беклога Продукта, неясно.
Sprint Planning One	первая часть Планирования Спринта (вместо Первое Планирование Спринта)	У sprint planning две части. Но оно одно! Поэтому перевод Первое Планирование Спринта и Второе Планирование Спринта очень неудачен.
vision and direction	видение (концепция) продукта и руководство направлением разработки	
product manager	продукт менеджер	
Multi-team Product Backlog Refinement workshop	многокомандные сессии по Проработке Беклога Продукта	
Single-team Product Backlog Refinement workshop	однокомандные сессии по Проработке Беклога Продукта	
Manager-teacher (managers-teachers)	менеджер-учитель (вместо "менеджер-наставник")	Термин из Lean

story map или user story map	карта пользовательских историй	техника работы с требованиями
meeting / workshop	совещание / сессия (рабочая сессия)	Так как workshops, такие как Design Workshop, Backlog Refinement Workshop - не являются официальными совещаниями (meetings, events) в Scrum каркасе, для них стоит использовать другое слово. "Рабочая сессия", на мой взгляд, вполне подходит.
Development Activities	активности по разработке (вместо мероприятия по разработке - это на "1ое мая были запланированы мероприятия", а команда занимается активностями)	
Scrum-of-Scrum	не переводим!	
Open Space	не переводим!	
Facilitator	фасилитатор	Не ведущий, не медиатор, не тренер, не инструктор.
Continuous Delivery	Непрерывная поставка (вместо "доставка" - доставляют пиццы, поставляют ценность)	
Design Workshop	Рабочую Сессию по Проектированию	
design	проектирование и дизайн системы	Дизайн имеет два смысла: внутреннее устройство системы и визуальный дизайн. Использую "дизайн системы", так как в примере вроде: "They are discussing design" - "они обсуждают дизайн" ("проектирование" не говорится).
Whiteboard	Доска или маркерная доска (вместо "белая доска")	"белая доска" не говорят. Использую попеременно "маркерная доска" и просто "доска". Будет понятно.
perfection goal	цель совершенствования (вместо цели совершенство - это переводчик этой книги - "совершенство", а цель lean - никогда не прекращать совершенствоваться)	
build fails	не проходит сборка (вместо "нарушается сборка")	Она не нарушается, она не проходит.

just talk	просто поговорите (правило в LeSS)	
handoff	передача (незавершенной работы, частично завершенной работы, работы на доработку)	имеется в виду передача работы от одного человека в процессе к другому (а не передача продукта в эксплуатацию)
site manager	координатор офиса	
business analysts	бизнес-аналитик (вместо "аналитик деловой активности")	переводчик даже глазом не моргнул погуглить...
product overview	общая картина продукта	
product manager	продукт-менеджер	
Sprint Review	Ревью Спринт или Спринт Ревью попеременно	
specification by example	спецификации через примеры	есть такая книга (не переведенная)
area product backlog	<i>Беклог Области Продукта</i>	
new market onboarding	подключение новых рынков (вместо "освоения...")	
spike	спайк (исследование через прототипирование) есть сноска	не переводим, термин из XP
mob-programming	mob-программирование есть сноска	не переводим

Оглавление

Вступительное слово Стивена Деннинга	13
Вступительное слово от Алексея Кривицкого	
Предисловие к русскому изданию	
Предисловие	15
Глава 1. Добиваться большего средствами LeSS	19
Введение в LeSS	21
Глава 2. LeSS	23
Структура LeSS	67
Глава 3. Внедрение	69
Глава 4. Организация по потребительской ценности	93
Глава 5. Управление	129
Глава 6. Scrum-мастера	149
Продукт в LeSS	165
Глава 7. Продукт	167
Глава 8. Владелец Продукта	183
Глава 9. Задел Продукта	211
Глава 10. Критерий Готовности	245
Спринт в LeSS	259
Глава 11. Уточнение Задела Продукта	261
Глава 12. Планирование Спринта	289
Глава 13. Координация действий и интеграция	301
Глава 14. Подведение итогов и ретроспектива	329
Больше или меньше LeSS	343
Глава 15. Что делать дальше	345
Приложение А. Рекомендованная литература	347
Приложение Б. Правила LeSS	349

Содержание

Вступительное слово Стивена Деннинга	13
Вступительное слово от Алексея Кривицкого	
Предисловие к русскому изданию	
Предисловие	15
Назначение книги	15
Кому адресована книга	15
Структура глав книги	16
Стиль изложения	16
Организационные термины	16
Благодарности	17
Глава 1. Добиваться большего средствами LeSS	19
Назначение LeSS	19
Введение в LeSS	21
Глава 2. LeSS	23
Scrum для одной команды	23
LeSS	24
Предпосылки	25
Эксперименты, наставления, правила и принципы	26
Принципы LeSS	28
Два каркаса LeSS и LeSS Huge	30
Каркас LeSS	31
Краткое изложение инфраструктуры LeSS	31
Истории в LeSS	32
История в LeSS: поток команд	33
История в LeSS: поток элементов	47
Каркас LeSS Huge	51
Область требований	51
Владельцы продуктов в отдельных областях	53
Команды в областях функциональных средств	53
Краткое изложение каркаса LeSS Huge	55
Истории в LeSS Huge	56
История в LeSS Huge: новая Область Требований	56
Многоместные команды: термины и рекомендации	63
История в LeSS Huge: многоместные команды	63
Что дальше	66

Структура LeSS	67
Глава 3. Внедрение	69
Scrum для одной команды	69
Внедрение LeSS	70
Правила LeSS	70
Наставление: три принципа внедрения	71
Наставление: начало внедрения	75
Наставление: культура следует структуре	80
Наставление: безопасность труда, а не роли	81
Наставление: организационная концепция совершенства	81
Наставление: непрерывное совершенствование	84
Наставление: наращивание внедрения	86
LeSS Huge	87
Правила LeSS Huge	87
Наставление: постепенное эволюционное внедрение	87
Наставление: поочередное создание	88
Областей Требований	88
Наставление: параллельные организации	88
Глава 4. Организация по потребительской ценности	93
Scrum для одной команды	93
Организация по потребительской ценности в LeSS	94
Правила LeSS	94
Наставление: построение организации на основе команд	95
Наставление: представление о командах для функциональных средств	97
Наставление: карты внедрения команд для функциональных средств	105
Наставление: отдавайте предпочтение специализации в предметной области клиентов	110
Наставление: организационная структура LeSS	112
Наставление: организация многоместной работы в LeSS	115
LeSS Huge	116
Правила LeSS Huge	116
Наставление: Области Требований	117
Наставление: динамика Областей Требований	119
Наставление: переход к командам для функциональных средств	120
Наставление: организационная структура LeSS Huge	123
Глава 5. Управление	129
Scrum для одной команды	129
Управление в LeSS	130
Правила LeSS	131
Наставление: представление о теориях Тейлора и Файоля	131
Наставление: теории управления	133
Наставление: руководители не обязательны	135

Наставление: организация, практикующая LeSS	136
Наставление: посещение рабочих мест	140
Наставление: руководители в качестве учителей и учащихся	142
Наставление: возможности как предметной области, так и технические	143
Наставление: количественные показатели и цели в LeSS	144
Наставление: список рекомендованной литературы по управлению	145
Глава 6. Scrum-мастера	149
Scrum для одной команды	149
Роль Scrum-мастера в LeSS	149
Правила LeSS	150
Наставление: основные сферы деятельности Scrum-мастера	150
Наставление: пять инструментальных средств Scrum-мастера	154
Наставление: содействие крупным группам в проведении совещаний	155
Наставление: содействие в обучении и приобретении многих навыков	156
Наставление: работа с сообществами	157
Наставление: руководство Scrum-мастера по выживанию	158
Наставление: список рекомендованной литературы для Scrum-мастера	161
Наставление: особое внимание Scrum-мастера	162
LeSS Huge	163
Наставление: исключение разобщенности Областей Требований	163
Продукт в LeSS	165
Глава 7. Продукт	167
Scrum для одной команды	167
Продукт в LeSS	168
Правила LeSS	169
Наставление: назначение продукта	169
Наставление: определение продукта	174
Наставление: расширение определения продукта	178
Наставление: продукт вместо проекта или программы	179
LeSS Huge	180
Глава 8. Владелец Продукта	183
Scrum для одной команды	183
Роль Владельца Продукта в LeSS	183
Правила LeSS	184
Наставление: кто должен быть Владельцем Продукта	184
Наставление: начало внедрения с временным заместителем Владельца Продукта	187
Наставление: кем являются потребители и пользователи	188
Наставление: назначение приоритетов вместо прояснения	190
Наставление: чего не должен делать Владелец Продукта	190

Наставление: помощники Владельца Продукта	191
Наставление: пять взаимоотношений	192
Наставление: сотрудничество с клиентами превыше всего	199
Наставление: выпуск хотя бы в каждом Спринте	200
Наставление: непреклонность	201
Наставление: ослабление контроля	202
Наставление: незавершенная работа не по вине Владельца Продукта	203
Наставление: совещания в LeSS	203
LeSS Huge	204
Правила LeSS Huge	205
Наставление: Владелец Продукта в LeSS Huge	205
Наставление: Владельцы Продукта в Областях	206
Наставление: помощь, оказываемая Scrum-мастером Команде Владельцев Продукта	207
Глава 9. Задел Продукта	211
Scrum для одной команды	211
Задел Продукта в LeSS	212
Правила LeSS	212
Наставление: минимизация ограничений вместо управления зависимостями	212
Наставление: надкусывание	216
Наставление: обращение с родительскими элементами	218
Наставление: обращение со специальными элементами	221
Наставление: инструментальные средства для ведения крупных заделов продуктов	224
Наставление: больше конечных и меньше непосредственных результатов	227
LeSS Huge	229
Правила LeSS Huge	229
Наставление: заделы в областях	229
Наставление: максимум три уровня	235
Наставление: новая область для очень большого требования	236
Наставление: обращение с очень большими требованиями	237
Глава 10. Критерий Готовности	245
Scrum для одной команды	245
Готовность в LeSS	246
Правила LeSS	246
Наставление: составление Критерия Готовности	246
Наставление: развитие Критерия Готовности	255
LeSS Huge	257

Спринт в LeSS	259
Глава 11. Уточнение Задела Продукта	261
Scrum для одной команды	261
Уточнение Задела Продукта в LeSS	262
Правила LeSS	263
Наставление: виды Уточнения Задела Продукта	263
Наставление: общее Уточнение Задела Продукта	265
Наставление: многокомандное Уточнение Задела Продукта	266
Наставление: многоместное Уточнение Задела продукта	268
Наставление: первоначальное Уточнение Задела Продукта	269
Наставление: разделение	275
Наставление: оценивание в крупных масштабах	284
LeSS HuGe	286
Глава 12. Планирование Спринта	289
Scrum для одной команды	289
Планирование Спринта в LeSS	289
Правила LeSS	290
Наставление: Первое Планирование Спринта	290
Наставление: многокомандное Второе Планирование Спринта	294
Наставление: ведение Задела Спринта без программных средств	295
LeSS HuGe	296
Наставление: совещание Команды Владельцев Продукта	297
Глава 13. Координирование действий и интеграция	301
Scrum для одной команды	301
Координирование действий и интеграция в LeSS	301
Правила LeSS	302
Наставление: поговорить	303
Наставление: удобная среда для координации действий	304
Наставление: поддержание связи через код	308
Наставление: интеграция непрерывно	309
Наставление: сообщества	311
Наставление: межкомандные совещания	314
Наставление: многокомандное совещание по проектированию	316
Наставление: рабочее совещание по текущей архитектуре	319
Наставление: кураторы компонентов	320
Наставление: Открытое Пространство	321
Наставление: спутники	322
Наставление: разведчики	323
Наставление: стоит ли проводить Схватку над Схватками	323
Наставление: ведущая команда	324
Наставление: сочетание разных методов	325
LeSS HuGe	326

Глава 14. Подведение итогов и ретроспектива	329
Scrum для одной команды	329
Подведение Итогов Спринта и Ретроспектизы в LeSS	329
Правила LeSS	330
Наставление: ранняя и частая адаптация продукта	331
Наставление: базар при подведении итогов	332
Наставление: Общая Ретроспектива	333
Наставление: совершенствование системы	336
LeSS Huge	340
Правила LeSS Huge	340
Наставление: подведение итогов и ретроспектизы во многих областях	340
Больше или меньше LeSS	343
Глава 15. Что делать дальше	345
Приложение А. Рекомендованная литература	347
Приложение Б. Правила LeSS	349
Правила инфраструктуры LeSS	349
Структура LeSS	349
Продукт в LeSS	350
Спринт в LeSS	350
Правила инфраструктуры LeSS Huge	352
Структура LeSS Huge	352
Продукт в LeSS Huge	352
Спринт в LeSS Huge	353
Предметный указатель	355

Вступительное Слово Стивена Деннинга

Каркас LeSS (Large-Scale Scrum — крупномасштабный Scrum¹) продолжает важные открытия в области управления, демонстрируя, как внедрить гибкую разработку и Scrum в крупном масштабе.

В XX веке иерархической бюрократии удавалось организовывать совместную работу крупных групп людей, чтобы добиться выдающихся успехов в повышении производительности труда. Но с тех пор мир изменился. Дeregулирование, глобализация, появление научноемкого труда и новых технологий (особенно Интернета) полностью изменили положение. В частности, увеличилась конкуренция, ускорились темпы перемен, программное обеспечение компьютеров обеспечило заметный рост производительности труда, но в то же время привело к значительному усложнению. По мере смещения власти на рынке от продавца к покупателю потребитель, а не производитель оказался в центре коммерческой вселенной. Такое смещение потребовало совершенно иного управления, способного мобилизовать творческий потенциал всех сотрудников в самой организации и за ее пределами на решение новых и более трудных задач, чтобы удовлетворить потребности клиентов. Перемены вышли далеко за пределы внесения корректива в существующие практики управления. Методики гибкой разработки и Scrum предлагают явные альтернативы устоявшимся, очевидным и бесспорным, на первый взгляд, представлениям об управлении.

LeSS показывает, как организовать крупномасштабную и сложную разработку. Самоуправляемые команды разработчиков не просто вызывают любопытство. Они способны вести обширные международные инициативы, обладающие большой технической сложностью. И эти практики не просто хорошо масштабируются, в отличие от бюрократии они масштабируются без маразма.

LeSS продолжает процесс основательной перестройки управления путем внедрения хорошо усвоенных уроков из более чем десятилетнего опыта масштабирования методов гибкой разработки и Scrum. Она демонстрирует, как бороться с непомерной сложностью, стремясь к простоте.

LeSS намеренно сделан неполным. Он оставляет обширное место для применения ситуационного управления. Не дает окончательные ответы. И не пытается удовлетворить наблюдавшиеся в XX веке попытки найти ясно сформулированные ответы или очевидно надежные и упорядоченные подходы, создающие удобную иллюзию предсказуемого управления. Вместо этого LeSS сосредоточен на

¹ Large-Scale Scrum можно также перевести также как "Scrum в масштабе предприятия", но это будет не совсем корректно, так как масштабы внедрения LeSS могут распространяться за рамки отдельно взятого предприятия. К примеру, если два предприятия совместно производят один с точки зрения конечного пользователя продукт, то с точки зрения LeSS эти компании составляют одну т.н. "продуктовую организацию" с географическим разделением, а согласно принципам LeSS эта организация должна быть оптимизирована как единное целое (об этом ниже). Так что использовать формулировку "масштаб предприятия" к LeSS не совсем некорректно. — Примеч. ред

минимальной сути, которая требуется при масштабировании, включая постоянное внимание к техническому совершенству и нацеленность на непрерывное экспериментирование. Она подразумевает бесконечные попытки новых экспериментов ради совершенствования. Как и Scrum, LeSS направлен на поиски золотой середины между абстрактными принципами и конкретными практиками.

Как и Scrum, LeSS является не процессом или методом создания продуктов, а каркасом, в которой процессы или методы можно адаптировать к потребностям конкретной ситуации. Он направлен на то, чтобы прояснить, каким образом практики управления и разработки могут обеспечить непрерывное совершенствование, повышающее потребительскую ценность.

Вместо конкретных ответов LeSS предоставляет отправную точку для понимания и принятия его глубинных принципов. Вместо вопроса “Как организовать гибкую разработку в масштабах сложной иерархической бюрократии?” она ставит другой и более основательный вопрос: “Как упростить организацию и стать гибкими?”

Такого равновесия LeSS стремится достичь в крупных продуктовых группах². Он привносит в Scrum более конкретную структуру, сохраняя в то же время радикальную прозрачность процесса и подчёркивая важность цикла “инспекции и адаптации”. Это позволяет группам непрерывно совершенствовать свои подходы к работе. Она отвечает на следующий основной вопрос: как распространить то, что вполне работоспособно на уровне отдельной команды, в масштабах всей организации?

Для масштабирования методик гибкой разработки и Scrum остается еще немало изучить и сделать. И эта книга одновременно сообщает о достигнутом прогрессе и открывает перспективы на будущее. В настоящее время во многие организации еще не научились тому, чтобы их многочисленные команды работали согласованно над различными частями продуктов и платформ. Как показывают опросы, большинство команд, работающих ныне по методикам гибкой разработки и Scrum, испытывают напряжение между тем, как они работают, и как управляется остальная организация. Эта книга служит практическим пошаговым руководством для снятия подобного напряжения.

Стивен Деннинг, автор книги *The Leader's Guide to Radical Management*,
27 апреля 2016 г.

² Здесь имеются в виду группы кросс-функциональных продукто-ориентированных команд, работающие совместно над созданием одного крупного продукта. Этот термин будет объяснен ниже в главах, связанных со структурой LeSS-организации. В книге мы попеременно используем “продуктовые группы” и “группы продукта” как синонимы. — Примеч. ред

Вступительное Слово Алексея Кривицкого

Перед Вами - книга, которую я лично ждал не один год.

Последние годы, будучи активным LeSS-практиком, мне довелось довольно близко познакомиться с авторами Крегом и Басом и сообществом единомышленников, которое они растят, с их философией, образом мышления и особенным чувством юмора. Будучи кандидатом в LeSS-тренеры, в 2015-2016 годах мне попались в руки черновики оригинальной книги на английском. Я был потрясен ее простотой и мощью. Мощью простоты. И мы делаем все возможное, чтобы эти качества сохранились в русском издании. Порой, это непросто (Мы не жалуемся! И об этом ниже в предисловии к русскому изданию).

Я честно говоря, никогда не мечтал, что мне выпадет честь редактировать русское издание, которое благодаря смелости и прозорливости издателей выйдет фактически с неощутимым опозданием, после выхода на рынок оригинала книги. (Это большое везенье для нашего рынка, где книги обычно выходят с задержками в десятилетия, а то и никогда. Это большой успех.) Но я не мечтал, что буду писать эти строки...

... когда в 2008 году я сижу на докладе и слушаю Баса Водде на стокгольмской конференции Scrum Gathering, где он в своей очень расслабленной манере, шутя о том, как он ненавидит финскую зиму (он голландец) и почему переезжает жить в Сингапур, очень просто и доходчиво показывает, как из-за неверно собранных команд с нехваткой навыков рождается ужас-ужасов неэффективности - проектный водопад. "Waterfall" со всеми вытекающими: большими задержками при выпусках, низким качеством, низкой мотивацией, недостатком внимания к техническому совершенству, культурой поиска виновных и перевода стрелок, избыточным менеджментом - типичные болезни традиционных компаний. Кажется, что хуже уже не может быть, но тут Бас берет маркер другого цвета и пририсовывает по одному беклогу на каждую команду. Он просто и доходчиво иллюстрирует, как из-за такого казалось бы логичного структурного решения, рождается иерархия Владельцев Продуктов, игра в оптимизации локальных интересов, отсутствие цельного видения продукта, и как следствие еще больше усугубляются все те же: большие, низкое качество, низкая мотивация, недостаток внимания к техническому совершенству, культура перевода стрелок, избыточный менеджмент...

Советы, которые я тогда услышал от Баса по устройству структуры команд и беклогов, для меня стали чем-то вроде спасительных огней. Я понял, как можно стремиться к гибкости в

масштабе, намного большем, чем одна Scrum команда. Я принял и полностью согласился с его выкладками. Они были интуитивно разумны. Я их применял годами со своими клиентами и не думал, что кто-то может быть настолько недалек, чтобы строить компании, используя абсолютно обратно противоположные принципы (и не для научного эксперимента, а в качестве попыток двигать многомиллионный бизнес)...

... пока в 2013 году я не окунулся в мир по-настоящему крупной разработки. Сотни разработчиков, десятки команд, пяток офисов в разных странах. Все команды работают по Scrum. У каждой - свой Владелец Продукта и Беклог со здоровым горизонтом на квартал вперед, цели на год, здоровый коллектив, выделенный Scrum-мастер на 1-2 команды, много внимания к техническим деталям, новейшие технологии отслеживания изменений и интеграции кода, налаженные частые циклы выпуска частей продукта и независимых микросервисов, достаточно много модных средств взаимодействия между командами (включая рекомендованный в LeSS "пойди и поговори, если нужно"), полки с новыми книгами, внутренние конференции, сообщества... В общем все, о чем только можно мечтать в продуктовой компании. Пока я не копнул глубже и выяснил, что ...

... чуть раньше, еще в мае 2012 года, ряд пользователей платформы подают на компанию жалобу - мол, профили пользователей, которые осуществляют коммерческую деятельность, не имеют информации о регистрации бизнеса, налоговых номеров и контактных данных - как это принято в Европе. По сути, для удовлетворения этого громкого требования в профилях пользователей нужно просто добавить одно текстовое поле в пару строк длиной, в которое те смогут вписать свои коммерческие данные. По сути ничего больше...

11 месяцев спустя, когда этого функционала все еще нет (у команд свои приоритеты развития продукта, и у каждого Владельца Продукта свое видение и квартальные планы), директор поддержки клиентов эскалирует эту проблему до уровня высокого руководства, решение быстро принимается и спускается по иерархии - на стол Владельцу Продукта, команда которого не по совпадению занимается профилями пользователей.

"Как пользователь платформы, занимающийся коммерческой деятельностью, у меня должна быть возможность вписать и потом редактировать информацию о своем бизнесе, как того требует законодательство" - два месяца спустя *веб-команда* пользовательских профилей выпускает эту возможность, и отдел связи с клиентами пишет гордый анонс. Стress снят. Пользователи довольны. Почти все.

Кроме тех, которые пользуются мобильными приложениями продукта! У них как этого поля не было, так и нет. Обман! В течение пары месяцев (десятков совещаний и сотня писем) принимается решение адаптации этого веб-функционала и выпуска его на обеих поддерживаемых мобильных платформах. Владелец Продукт команды "Профиль" создает в системе электронного ведения беклогов соответствующие элементы, назначает их на Владельца Продукта *мобильной* команды и утирает руки...

Казалось бы, дело сделано. Что может быть проще добавить в iOS и Android приложения по полю для отображения и ввода неформатированной текстовой информации. И правда. Дело на полдня с тестированием... Но это при условии, что есть откуда читать данные и куда передавать введенные. А про этот *маленький нюанс* Владелец Продукта команды веб-

профилей как раз и не подумал. Вернее, этот нюанс может быть даже и не его головная боль, ведь мобильная команда должна быть в состоянии добавить для себя пару необходимых REST-интерфейсов. Так то оно так, да только мобильная команда умеет писать только мобильный код. Доступа к репозиторию REST-интерфейсов на Ruby on Rails у нее никогда не было, и делать эту работу она, честно говоря, не собирается, даже если бы и умела - ее задача писать мобильные приложения. Jawohl³!

Владелец Продукта мобильных команд, не долго думая, открывает систему ведения беклогов и заводит запись о создании REST-интерфейсов, назначает ее на нашего старого знакомого - Владельца Продукта команды профилей и утирает руки. На времена проблема решена, можно заняться своими планами по повышению показателей мобильных приложений, за которые, он, к слову, в конце года ожидает получить премию...

Не стоит расписывать эту историю еще на несколько страниц. Но стоит лишь заметить, что подобные *пинг-понги* повторялись еще несколько раз: ко всеобщему удивлению разработчиков мобильной команды, интерфейс вычитки текстового поля, который для них разработали инженеры по профилю, оказался кривоват. Они просили в течение пяти недель команду профиля переделать этот интерфейс, эскалировали эту ситуацию, но Владелец Продукта команды профиля на отрез отказался - у него и без того уже съехал квартальный план. Мобильным разработчикам пришлось писать неоптимальный код, который ломал принятую в приложениях концепцию взаимодействия с пользователем. Функциональность и правда, даже на мой вкус, выглядела кривовато, когда...

... с 2013 по 2014 годах я как раз работал с этими мобильными командами в качестве Scrum-мастера и сам недоумевая, почему команды, от которых мы зазависим ведут себя так неадекватно. Пока позже в 2015 году я не провел археологические раскопки этого проекта по "добавлению текстового поля" и не открыл для себя мир, который еле поместился на переносную магнитную доску в моей части офиса:

³ Нем: "так точно!". — Примеч. ред

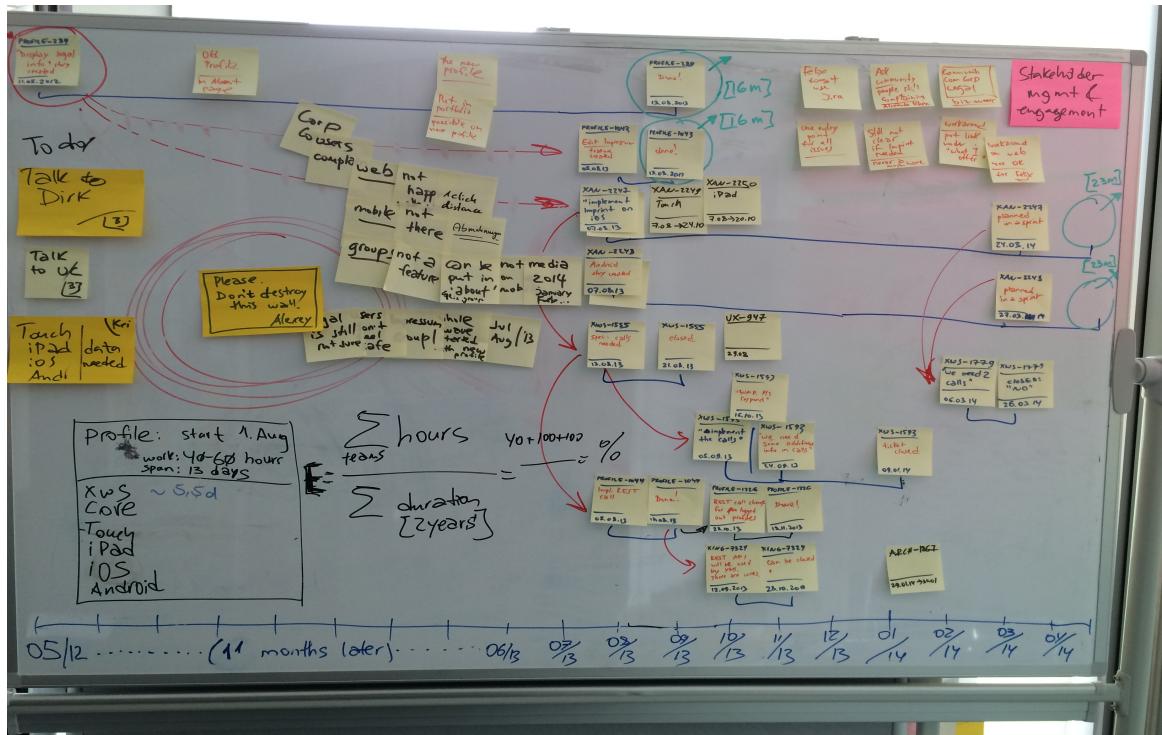


Figure 1 Раскопки проекта по добавлению текстового поля.

Мои раскопки показали, что на этот проект ушло 23 месяца.

Не могу удержаться: вот для чего реально хороши электронные системы ведения беклогов! Я смог найти все записи и всю переписку, все споры и пинг-понги, все обвинения и эскалации. Все было как на ладони. Представляете, если бы функцию редактирования текстового поля на всех пользовательских интерфейсах разработали бы одной небольшой командой из шести человек (пара мобильных и две пары веб-разработчиков) за неделю, используя простые kleящиеся карточки для записи беклога - нечего было бы даже вспомнить! А так - яркая история на три страницы и 23 месяца работы.

К слову, в течение из 23 месяцев *длительности* проекта реально же всеми вовлечеными разработчиками всех пяти команд (в итоге столько команд и столько Владельцев Продуктов было вовлечено) было потрачено шесть человеко-недель (эквивалент шести инженерам, вместе работающим одну неделю). "Куда же делись остальные *года*?!?", - спросит нетерпеливый читатель. "Ожидание, прояснение полученной информации, снова ожидание, еще ожидание, эскалация, переназначение записи в беклоге с одного Владельца Продукт на другого, два новогодних корпоратива плюс два цикла летних отпусков...", - ответим мы.

...

Когда в книге, которую вы держите в руках, и от которой вас по сути отделяют мои присказки, Бас Водде и Крег Ларман говорят "ожидания", "передача незавершенной работы между командами", "локальная оптимизация", "один беклог на продукт" и "полные кросфункциональные кросс-компонентные полно-стековые команды, ориентированные на конечный продукт и счастливых клиентов" - я надеюсь, что перед вами будет всплывать это злополучное текстовое поле. Возможно, самое длинное (по времени) в истории информационных технологий.

Следующие страницы откроют перед вами удивительный мир LeSS. Удивительный тем, как чрезмерно упрощенные структуры и процессы, явно усеченные иерархии и суженные множества ролей, - как все это может чудесным образом оживлять работу такого огромного, динамического и сложного организма как продуктовая организация.

Вся сила в системном упрощении процессов.

Да пребудет она с вами.

More with LeSS!

Алексей Кривицкий, Agile-коуч, Scrum-тренер и LeSS-практик.

10 февраля 2017 года.

Предисловие в русскому изданию

Эта книга наполнена терминами, которые читателю, не знакомому с каркасом Scrum, могут показаться новыми, непонятными и отчасти заимствованными из других языков.

С другой стороны, читателю, хорошо знакомому с тематической литературой, и в частности оригинальными англоязычными терминами, наш перевод Scrum-жаргона может показаться чужеродными, непривычным и отчасти вычурным.

Что ж, на нас лежит очень ответственная задача - найти изящный баланс между использованием чистого русского языка, как и где это возможно, при этом оставаясь близкими к источникам и отчасти кустарно сложившейся традицией заимствований из английского, столь привычной в IT-среде.

Поэтому не судите нас строго, если вы встретите русские аналоги привычных вам оборотов (вроде "Product Owner" и "Product Backlog Refinement workshop"), которые зачастую слышны в кулуарах русскоязычных IT-компаний в их оригинальном звучании. Эти термины, к примеру, мы переводим как "Владелец Продукта" и "рабочая сессия Проработки Беклога Продукта".

И также не судите нас уж слишком, если некоторые, казалось бы переводимые термины (как собственно "Product Backlog" и сам "Scrum"), мы будем использовать без перевода вовсе, так как "задел" и "схватка" конечно имеют славянские корни, но в данном контексте никакой пользы не принесут.

Ниже, приводится таблица соответствия оригинальных терминов каркасов Scrum и LeSS - тем, которые мы выбрали и используем в данном издании этой книги:

Оригинальный термин	Русский перевод
Scrum	Scrum
LeSS	LeSS
LeSS Huge	LeSS Huge
РОЛИ:	
Product Owner	Владелец Продукта
Scrum Master	Scrum-мастер
Team	Команда

АРТЕФАКТЫ:	
Product Backlog	
Area Product Backlog	
Sprint Backlog	
Potentially Shippable Product Increment	
Definition of Done	
....	
ЦЕРЕМОНИИ	
Meeting	
Workshop	
Sprint	
Sprint Planning	
Sprint Planning One	
Sprint Planning Two Multi-team Sprint Planning Two	
Daily Scrum	
Scrum-of-Scrum	
Sprint Review	
Sprint Retrospective Overall Sprint Retrospective	
Product Backlog Refinement (PBR) Team PBR Multi-team PBR Overall PBR	
Multi-Team Design Workshop	
...	
ДРУГИЕ ТЕРМИНЫ	

Test-Driven Development (TDD)	
Automated User Acceptance Testing	
Continuous Integration	
Continuous Delivery	
Build	
Failed build	
...	

Предисловие

Все великие истины начинаются с богохульства.
— Джордж Бернард Шоу

Добро пожаловать в мир LeSS, где более простые структуры заменяют организационную сложность, фокусируясь на людях и приобретаемых знаниях. Для некоторых людей LeSS кажется романтическим и безнадежно идеалистичным. Тем не менее сегодня он уже стал реальностью для многих продуктовых групп!

Зачем эта книга?

В ответ на отзывы о двух предыдущих книгах по LeSS, где многие понятия были представлены, исходя из слишком малого числа отправных точек, один из авторов, Крейг, обратился к другому автору, Басу, с предложением написать еще одну книгу. Бас отклонил это предложение, поскольку он с нетерпением ожидал появления на свет своего второго сына. Непреклонный Крейг все же убедил Баса, что написать новую книгу будет нетрудно. Крейг был не прав.

Первоначально мы, авторы книги, намеревались написать учебник для начинающих изучать LeSS по материалам предыдущих своих книг. Но в конечном итоге у нас получилась совсем другая книга, поскольку наши дальнейшие исследования отправных точек, на которых базируются практики масштабирования, привели к минимализации основ масштабирования. Результат? Правила LeSS, руководство LeSS плюс сама эта книга.

Правила и руководства по LeSS, безусловно, важны, но они являются далеко не единственными соображениями, из которых следует исходить при масштабировании. Прежде чем перейти к LeSS, мы хотели бы явно подчеркнуть два важных момента: постоянное внимание к *техническому совершенству* и *нацеленность на экспериментирование*.

Кому адресована книга

Эта книга адресована всем, кто занимается разработкой программных продуктов. Единственным предварительным условием для ее чтения является знание основ Scrum. Если у вас нет таких знаний, рекомендуется прочитать *Руководство по Scrum* (Scrum Guide; <http://scrumguides.org>) и *Учебник по Scrum* (Scrum Primer; <http://scrumprimer.org>). Каждая глава данной книги будет начинаться с краткого напоминания основных положений Scrum, связанных с рассматриваемой темой.

Структура глав книги

Каждая из основных глав книги имеет следующую структуру.

- > **Scrum для одной команды.** Подытоживает применение Scrum в одной команде, чтобы заложить основание для изучения LeSS.
- > **LeSS.** Охватывает основной каркас LeSS. Этот раздел имеет такую структуру.
 - > Введение в тему и изложение связанных с ней принципов LeSS.
 - > Правила LeSS.
 - > Руководства LeSS.
- > **LeSS Huge.** Этот раздел имеет такую же структуру, как и раздел, посвященный основному каркасу LeSS.

Стиль изложения

Мы решили придерживаться следующего стиля изложения.

- > Термины LeSS и Scrum выделены прописными буквами, например: Спринт, Беклог Продукта, Команда. **Примечание:** *Команда* — это роль в LeSS, тогда как *команда* — общее понятие, обозначающее коллектив работников.
- > На протяжении всей книги слово *вы* употребляется для обращения к читателю. При этом допускается, что читатель вовлечен во внедрение LeSS, и его роль связана с темой конкретной главы. Например, в главе, посвященной Владельцу Продукта, *вам* отведена роль Владельца Продукта.
- > Для выделения отдельных положений в тексте книги употребляется *наклонное, полужирное* начертание и врезки.
- > В данной книге намеренно дано немного библиографических ссылок. Поэтому для более основательных ссылок рекомендуется обратиться к двум предыдущим нашим книгам, где приведена обширная библиография.

Организационные термины

Большая часть терминов, определяется при употреблении в первый раз. Но нам пришлось привести разные организационные термины, употребляемые в различных компаниях, к какому-то общему знаменателю. С этой целью мы ввели в употребление термины, которые будут очевидны для одних читателей данной книги, но неочевидны для других. Эти термины вкратце поясняются ниже.

- > **Продуктовая группа.** Это все люди, вовлеченные в создание продукта. Термин *проект* нередко употребляется в организациях для обозначения всех людей, задействованных в разработке, но в данной книге этот термин *не* употребляется потому, что в ней делается акцент на разработку продуктов. Отсюда и термин *продуктовая группа*.

- > **Линейная организация.** Формальная организация, обычно изображаемая в виде схемы организационной структуры. Как правило, линейная организация включает в себя аттестацию, наем, увольнение и повышение квалификации работников. В компаниях может применяться матричная организация проектов (она не должна существовать в LeSS), а также организация управления и поддержки персонала.
- > **Линейный руководитель и руководитель первого звена.** Линейным является руководитель, перед которым отчитывается работник при линейной организации, а руководителем низшего (первого) звена — тот линейный руководитель, перед которым непосредственно отчитывается работник.
- > **Руководитель высшего уровня, или управляющий.** Это руководитель, работающий на уровне, близком к высшему руководству. В крупной организации такие руководители действуют за пределами продуктовой группы.
- > **Управление и маркетинг продукции.** Функция продуктовых организаций, для изучения рынка сбыта и выбора содержимого продуктов. Как правило, она *не* имеет линейной связи с командами.
- > **Руководитель продуктовой группы.** Работник, который руководит продуктовой группой и перед которым все члены группы отчитываются по линейной связи.
- > **Руководитель проекта или программы.** Этой роли по традиции отводится планирование выпуска продуктов. У такой роли, как правило, *отсутствует* линейная связь с командой, поскольку она носит краткосрочный, временный характер. Подобные роли не должны существовать в организации, работающей по методике LeSS.

Благодарности

Мы привлекли к рецензированию данной книги большое число специалистов. Ниже перечислены те, кто рецензировал не одну ее главу.

Янн Кохвакка (Janne Kohvakka), Ганс Ноймайер (Hans Neumaier), Рафаэль Саббах (Rafael Sabbagh), Рэн Наймен (Ran Nyman), Ахмад Фами (Ahmad Fahmy), Майк Кон (Mike Cohn), Гойко Адзич (Gojko Adzic), Ютта Экштайн (Jutta Eckstein), Роэн Баннинг (Rowan Bunning), Жан-Марк Жербер (Jeanmarc Gerber), И Лв (Yi Lv), Стив Спирмэн (Steve Spearman), Карен Гривз (Karen Greaves), Марко Зеелманн (Marco Seelmann), Сезарио Рамос (Cesario Ramos), Маркус Гётнер (Markus Gärtner), Виктор Гргич (Viktor Grgic), Крис Чен (Chris Chan), Нильс Бернерт (Nils Bernert), Вячеслав Розет (Viacheslav Rozet), Эдвард Даллёф (Edward Dahllöf), Лиза Криспин (Lisa Crispin), Майк Дуайер (Mike Dwyer), Франческо Сферлацца (Francesco Sferlazza), Натан Слиппен (Nathan Slippen), Мика Сьюман (Mika Sjöman), Тим Борн (Tim Born), Чарльз Брэдли (Charles Bradley), Тимоти Корсон (Timothy Korson), Эрин Перри (Erin Perry), Грег Хатчингз (Greg Hutchings), Джез Хамбл (Jez Humble), Алексей Кривицкий

(Alexey Krivitsky), Александр Гербер (Alexander Gerber), Питер Браун (Peter Braun), Юрген де Шмет (Jurgen De Smet), Эвелин Тиан (Evelyn Tian), Сами Лилья (Sami Lilja), Стивен Мак (Steven Mak), Александр Коттинг (Alexandre Cotting), Боб Шатц (Bob Schatz), Боб Сарни (Bob Sarni), Милинд Кулкарни (Milind Kulkarni), Джанет Грегори (Janet Gregory), Джерри Раджамани (Jerry Rajamoney), Карл Коллишан (Karl Kollischan), Шив Кумар Мн (Shiv Kumar Mn), Дэвид Нунн (David Nunn), Рене Аманн (Rene Hamann), Илан Гольдштайн (Ilan Goldstein), Хуан Габардини (Juan Gabardini), Мехмет Йитмен (Mehmet Yitmen), Кай-Уве Рупп (Kai-Uwe Rupp), Кристиан Энгблом (Christian Engblom), Джеймс Греннинг (James Grenning), Венкатеш Кришнамурти (Venkatesh Krishnamurthy), Питер Хандермарк (Peter Hundermark), Арне Аландер (Arne Ahlander), Даррен Лай (Darren Lai), Маркус Зайтц (Markus Seitz), Гейр Амсё (Geir Amsjø), Рам Шринивасан (Ram Srinivasan), Марк Брегенцер (Mark Bregenzer), Аарон Сандерс (Aaron Sanders), Майкл Балле (Michael Ballé), Стюарт Тернер (Stuart Turner), Эалден Эскакан (Ealden Escacan), Стивен Кох (Steven Koh), Кен Ягучи (Ken Yaguchi), Майкл Джеймс (Michael James), Маной Вадаккан (Manoj Vadakkan), Петер Цуркирхен (Peter Zurkirchen), Ласло Черекли (Laszlo Csereklei), Гордон Уэир (Gordon Weir), Лоран Карбонно (Laurent Carboneaux), Элад Софер (Elad Sofer).

Особая признательность выражается Берни Куа (Bernie Quah) — за художественное оформление книги, а также Терри Йину (Terry Yin) — за оказанную поддержку практически по всем вопросам. Благодарим также Криса Гудзиковски (Chris Guzikowski) из издательства Addison-Wesley — за проявленное терпение на протяжении всей долгой работы над книгой.

Больше с LeSS⁴

Самыми дешевыми, быстрыми и надежными являются отсутствующие компоненты.

— Гордон Белл

Зачем нужен LeSS?

Почему методика Scrum получила бурное развитие за последнее десятилетие? На этот вопрос мы искали ответ за кружкой пива в сингапурском кабачке.

Одни усматривают причину в упрощенной модели сертификации. Возможно, но ведь другая методика гибкой разработки под названием DSDM предоставляла сертификацию задолго до того, как методика Scrum получила широкое распространение.

Другие усматривают причину в появлении курсов, доступных для Scrum-мастеров. И действительно, большое влияние оказали первоначальные курсы для Scrum-мастеров, организованные Кеном Швабером (Ken Schwaber). Тем не менее появившиеся раньше курсы по экстремальному программированию не нашли столь широкого распространения.

Возможно, причина кроется в простоте Scrum? По сравнению с экстремальным программированием Scrum предоставляет более простую каркас. Но еще более простые методики вроде Crystal так и не достигли вершин популярности. После дополнительных дискуссий и размышлений Крейг, наконец, предложил следующее объяснение.

⁴ В оригинале: "More with LeSS". Тут и дальше игра слов: так как "less" - по-английски означает "меньше", выходит "**больше с меньшим**" или "получать больше меньшими средствами". Этот слоган очень часто используют авторы Крейга и Баса, так как он очень четко описывает саму философию LeSS, где наименьшими средствами (минимумом ролей, встреч, артефактов) мы учимся достигать больших результатов. — Примеч. ред

В методике Scrum соблюдается идеальный баланс между абстрактными принципами и конкретными практиками.

На этом дискуссия завершилась, и мы пропустили еще по одному пиву.

В упомянутых выше конкретных практиках акцент делается на *эмпирическом управлении процессом*, составляющим главный принцип Scrum. Именно эмпирическое управление процессом отличает Scrum от других подходов гибкой разработки. Это хорошо объясняется в *Руководстве по Scrum* следующим образом.

Scrum является не процессом или методом создания продуктов, а каркасом, который позволяет применять различные процессы и методы. Scrum помогает увидеть связь между относительной эффективностью управления продуктом и практиками разработки - так, что у вас появляется возможность улучшать процесс..

А проще? Это означает, что благодаря эмпирическому управлению процессом можно больше не фиксировать ни объем работ, ни сам *процесс создания продукта*. Вместо этого в течение коротких циклов мы создаем небольшие готовые к поставке части продукта, время от времени изучая, что у нас получилось и как мы это делали, за счет встроенных механизмов обеспечения прозрачности. Столь ясная инспекция процесса и продукта оказывается возможной благодаря встроенным механизмам прозрачности в Scrum.

Сами по себе *принципы* замечательны, но они не всегда помогают понять, как действовать. Зато небольшой ряд конкретных практик позволяет легко начать использовать Scrum: его чёткие роли, артефакты и события.

Такие практики позволяют хотя бы с чего-то начать, но они намеренно сделаны “неполными”, оставляя командам и группам *свободное место* для постоянного обучения и совершенствования внутри Scrum каркаса, - и это важно, так как мы осознаем, что работаем в областях повышенной сложностью, где готовые рецепты для производственных процессов оказываются слишком упрощенными.

Конкретные практики Scrum служат отправными точками для реализации его более глубоких принципов. Идеальный баланс.

Large-Scale Scrum (LeSS) позволяет добиваться такого же баланса в крупных продуктовых группах. Он привносит чуть более конкретную структуру в Scrum, где преследуется цель поддерживать прозрачность и подчеркивать важность цикла инспекции и адаптации таким образом, чтобы группы могли непрерывно совершенствовать свои навыки работы.

Как и Scrum, LeSS намеренно сделан неполный, оставляя место для ситуационного обучения. Он дает немного определенных ответов и не может удовлетворить тех, кто ищет шаблонные ответы или очевидно безопасные и дисциплинированные подходы⁵, создающие удобную иллюзию предсказуемого управления посредством детерминированных процессов. Такие подходы нарушают принцип эмпирического управления и ощущение владения процессами и практики.

Менее определенный процесс ведет к большим открытиям.

Это, по существу и означает: больше меньшими средствами. More with less.

⁵ Здесь авторы намеренно используют слова "безопасные и дисциплинированные" (по англ.: "safe and disciplined") - как отсылка к одноименным популярным подходам масштабирования гибкой разработки и Scrum, которые базируются на существенно отличающихся от LeSS принципах. — Примеч. ред

Введение В LeSS



Большая карта пользовательских историй (техника "story map") во время церемонии начального Прояснения Беклога Продукта ("initial Product Backlog Refinement" или "initial PBR") в LeSS.

2

LeSS

*Существуют два подхода к дизайну:
один — сделать его настолько простым, чтобы в нем явно
отсутствовали недостатки,
а другой — сделать его настолько сложным, чтобы
в нем не нашлось явных недостатков.
— K.A.P. Хоар*

Scrum для одной команды

Scrum — это каркас эмпирического управления процессом разработки, где кросс-функциональная самоуправляемая *Команда* разрабатывает программный продукт итеративным и инкрементным способом⁶. В течение каждого ограниченного по времени *Спринта* производится, а в идеальном случае — поставляется *потенциально готовый к поставке инкремент продукта*. Единственный *Владелец Продукта* отвечает за достижение максимальной ценности продукта, приоритезацию отдельных элементов в *Беклоге Продукта* и адаптивное определение цели каждого следующего *Спринта* на основании постоянной обратной связи и полученным знаниям. Небольшая *Команда* отвечает за достижение поставленной цели Спринта. Команда не ограничена работой узко-специализированных ролей. *Scrum-мастер* обучает каркасу Scrum и тому, как Scrum помогает поставлять ценность. Он выступает в роли коуча для Владельца Продукта, Команды и организации, помогая понять, как использовать Scrum и сам является зеркалом для них. Здесь нет места проектным менеджерам и руководителям команд⁷.

⁶ В предисловии к данной книге поясняется, почему ее главы начинаются именно с этого раздела, основная структура повторяется в каждой главе наряду с определением ряда ключевых терминов и стиля изложения.

⁷ Здесь в книге авторами используется часто встречаемый в ИТ-компаниях термин "team lead", который может быть хорошо знаком читателю. Мы его переводим как "руководитель команд". Но как бы мы эту должность не называли, она вступает в конфликт с сутью самоуправляемой команды. И именно поэтому в LeSS не применима. Почти то же и с ролью проектного менеджера, плюс в добавок тот факт, что LeSS не признает саму суть проекта. — Примеч. ред

Для управления эмпирическим процессом требуется *прозрачность*, которая обеспечивается короткими циклами разработки и обратной связи по поставляемому инкременту продукта. При этом делается акцент на непрерывном обучении и инспекции и адаптации как самого продукта, так и способа его создания.

Такое управление основывается на представлении о том, что процесс разработки является слишком сложным и динамичным для следования подробным и шаблонным рецептам, сдерживающих пытливость, вовлеченность и совершенствование.

В *Руководстве по Scrum* и *Учебнике по Scrum* внимание уделяется работе одной Команды, а не многим совместно работающим командам. И это, естественным образом наводит на мысли о *крупномасштабной Scrum*.

LeSS

LeSS—это Scrum примененный ко многим командам, совместно работающим над одним продуктом.

См. главу 3

LeSS - это Scrum⁸ и это ни новый или какой-то усовершенствованной Scrum. Это также ни в коем случае *ни Scrum на уровне команд, и что-то другое на уровнях выше*. Напротив, LeSS - про то, как применять принципы, цели, элементы и изящество Scrum в полном масштабе и при этом как можно проще. Подобно Scrum и другим подлинно гибким каркасам, LeSS по ряду веских оснований является так сказать “едва достаточной методикой” .

Масштабный Scrum не является специальным каркасом масштабирования, включающим в себя Scrum только на уровне команд. Настоящий масштабный Scrum - это Scrum в масштабе

См. главу 4

См. главу
13

⁸ LeSS предполагает крупномасштабный Scrum, но также упрощение при масштабировании, чтобы добиваться большего меньшими средствами - "less".

...применимый ко многим командам. А это **кросс-функциональные** (имеющие во См. главу⁷ навыки для работы над всем продуктом и его выпуском), **кросс-компонентные** (работающие над кодом всех компонентов, продукта), **полно-стековые** (владеющие полным стеком технологий продукта) и **продукто-ориентированные** (разрабатывающие полезную функциональность с точки зрения конечного пользователя) **команды**⁹ из 3–9 человек, сосредоточенных на обучении и способных делать все виды работ (от проектирования взаимодействия с пользователем до программирования и подготовки видеоматериалов) с целью реализации элементов Беклопа Продукта и готового к поставке продукта.

...для совместной работы. Команды работают вместе, поскольку у них есть общая цель — выпустить один готовый к поставке продукт в конце общего Сprintа, и каждая команда заинтересована в достижении этой цели, поскольку они (как продукто-ориентированные команды)

...нар одним продуктом. Что такое продукт? Завершенное полностью интегрированное клиенто-ориентированное решение, которое реально используется . Это не компонент, платформа, слой или библиотека.

⁹ Во избежание недопонимания здесь стоит привести оригинальную сочную терминологию LeSS: "cross-functional, cross-component, full-stack feature teams". Вы заметите, как подобное описание Команды в LeSS повторяется вновь и вновь. Это не дань занудству и не сила гипноза. Это сделано для того, чтобы мы вами смогли осознать важность и неподкупность этих существенных характеристик Команд, ведь такие Команды - сердце Scrum, а значит и неотъемлемая часть философии LeSS. — Примеч. ред

Предпосылки

В 2002 году, когда Крейг написал книгу *Agile & Iterative Development*, многие считали гибкую разработку применимой только в небольших группах. Но мы (Крейг и Бас) заинтересовались и стали получать все больше запросов на применение Scrum в крупной, географически распределенной по многим странам разработке. Поэтому в 2005 году мы объединили свои усилия для работы с клиентами над масштабированием Scrum. Ныне два каркаса LeSS (меньший - просто "LeSS" и больший - "LeSS Huge¹⁰") уже приняты в ряде перечисленных ниже больших групп, по всему миру.

- > Телекоммуникационное оборудование — Ericsson & Nokia Networks¹¹.
- > Инвестиционные и розничные банки — UBS.
- > Торговые системы — ION Trading.
- > Маркетинговые платформы и аналитика продвижения товаров — Vendasta.
- > Проведение видеоконференций — Cisco.
- > Онлайн-игры и ставки — bwin.party.
- > Аутсорсинг и удалённая разработка — Valtech India¹².

С точки зрения *крупной* разработки, какой пример внедрения LeSS можно привести? Это, вероятно, пять команд в одном или двух офисах. Нам приходилось внедрять LeSS в подобных масштабах: от сотни до более тысячи людей (в случае с LeSS Huge), в большом количестве распределенных географических точек, с десятками миллионов строк кода на C++ да еще и с специализированным оборудованием в придачу.

Дополнительные источники для изучения LeSS

Чтобы помочь людям в изучении LeSS, мы издали в 2008 и 2010 годах две книги по масштабированию гибкой разработки с помощью LeSS, основываясь на своем опыте работы с клиентами.

1. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. В этой книге поясняются перемены, которые требуются в мышлении, руководстве и организационном дизайне.

¹⁰ Очередная игра слов: "less" - меньший, "huge" - громадный. То есть каркас применим к *громадным* группам (8 команд и более), но мы все же пытаемся в стиле LeSS применять к ним *минимально возможный* процесс. — Примеч. ред

¹¹ Nokia Networks — это *не* фирма, производящая мобильные телефоны и приобретенная корпорацией Microsoft.

¹² См. разбор других практических примеров на веб-сайте less.works.

2. *Practices for Scaling Lean & Agile Development: Large, Multi-site & Offshore Product Development with Large-Scale Scrum.* В этой книге приводятся сотни примеров конкретных экспериментов с LeSS, основанных на опыте работы с клиентами. Это эксперименты в управлении продуктами, архитектуре, планировании, географически-распределенной разработкой, контрактами и пр.

Данная книга является третьей из серии, посвященной LeSS. Она тематически связана с двумя предыдущими книгами из данной серии и служит учебником для начинающих изучать LeSS. В ней соединено, прояснено и выделено самое важное.

Помимо этих книг и библиографии в приложении А, рекомендуются доступные по адресу <http://less.works/> учебные ресурсы, в том числе главы из книг, статьи и видеоматериалы, а также курсы и коучинг.

Эксперименты, руководства, правила и принципы

В первых двух из упомянутых выше книг по LeSS подчеркивалось следующее: *в разработке продуктов не существует таких понятий, как передовые или лучшие практики. Имеются лишь те практики, которые пригодны в определенном контексте.*

Практики ситуативны. Они отрываются от намерений и контекста и становятся ритуальными, если беспечно объявить их "передовыми" или "лучшими". А навязывание так называемых передовых практик убивает культуру изучения, выведения, за-интересованного участия и непрерывного совершенствования. Зачем улучшать то, что уже и так "передовое и лучшее"?

Поэтому в предыдущих книгах по LeSS мы поделились с читателями своими экспериментами, опробованными вместе с нашими клиентами, поощряя с тех пор и поныне именно такой экспериментаторский настрой. Но со временем мы обнаружили следующие две проблемы, когда людям даются только описания и результаты экспериментов.

- > Начинающие группы принимали неумелые решения во вред себе, внедряя LeSS не по назначению, что приводило к очевидным осложнениям. Например, группы создавали Области Требований на уровне каждой команды. Ой!
- > Начинающие группы спрашивали: "Где нам начать? Что самое важное?" Они понятное дело не владели самыми основами для ответов на эти вопросы.

Размышляя над этой реакцией, мы вернулись к модели обучения по принципу *Сю–Ха–Ри* японского боевого искусства, где *Сю* означает следовать правилам изучения основ, *Ха* — нарушать правила и обнаруживать контекст, а *Ри* — овладевать предметом и искать свой путь. При внедрении LeSS на уровне Сю даются несколько *правил* для *едва достаточного каркаса*, для того, чтобы начать эмпирически управлять процессом и сосредоточиться на всем продукте в целом¹³. Эти правила определяют

¹³ Каркас Scrum состоит из нескольких правил по тем же причинам, что и каркасы LeSS.

Таким образом, LeSS включает в себя следующее.

- > **Правила.** Это ряд правил, с которых следует начинать, чтобы заложить прочный фундамент. Они определяют главные элементы *каркасов LeSS*, которые должны быть на месте для поддержки эмпирического управления процессом и сосредоточении на продукте в целом. Примером тому служит правило *проводить Общую Ретроспективу в каждом Спринте*.
- > **Руководства.** Это умеренный ряд рекомендаций для эффективного внедрения правил и проведения подмножества экспериментов, которые стоит проводить, следуя, многолетней экспериментатской традиции LeSS. Наставления содержат *советы*. Как правило, они полезны и удобны для непрерывного совершенствования. Примером тому служит наставление *соблюдать Три Принципа Внедрения*.
- > **Эксперименты.** Многие эксперименты, которые привязаны к конкретной ситуации, а возможно, и не стоят того, чтобы их проводили. Примером тому служит эксперимент *попробовать ввести Переводчика в Команду*.
- > **Принципы.** Это, по существу, ряд принципов, кристаллизованных из опыта внедрения LeSS, которые направляют правила, руководства и эксперименты. Пример тому — принцип *внимания на продукте в целом*.

Руководства и эксперименты в LeSS необязательны. И хотя руководства полезны и рекомендуются для соблюдения, тем не менее, ими можно пренебречь, если они ограничивают возможности совершенствования или просто непригодны в текущей ситуации.

Представить инфраструктуру LeSS удобно в виде *полной картины*, как наглядно показано на приведенном ниже рисунке.

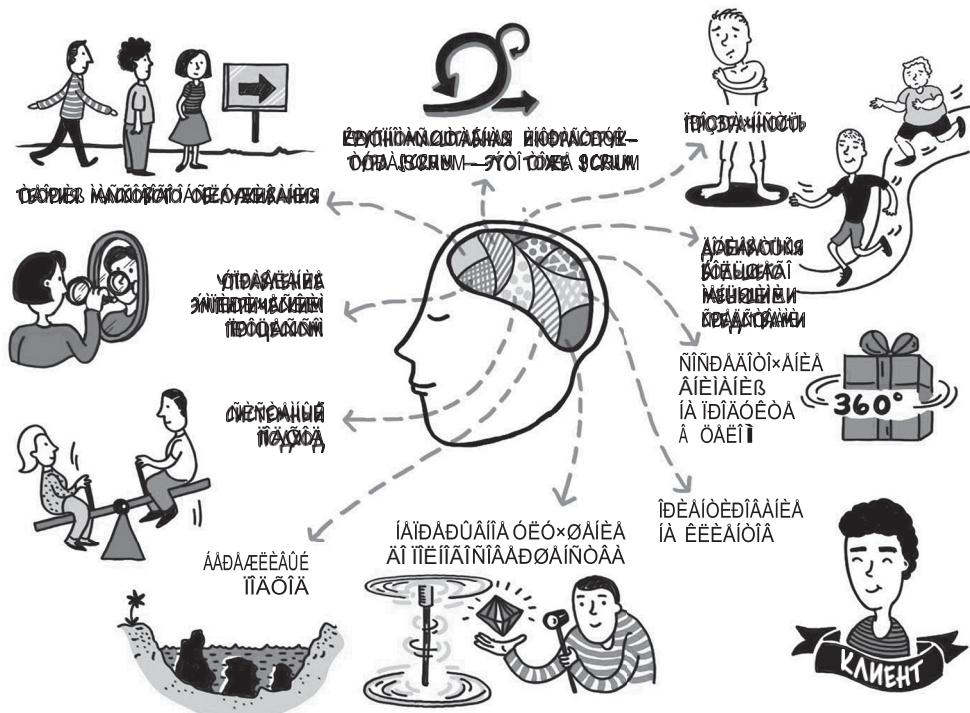


Полная картина LeSS определяет то, как мы обычно знакомим с LeSS.

1. Принципы LeSS, рассматриваемые ниже.
 2. Каркасы LeSS, определяемые по **правилам** и рассматриваемые в дальнейшем в этой главе.
 3. Руководства LeSS, рассматриваемые в последующих главах данной книги.
 4. Эксперименты в LeSS, уже описанные в первых двух книгах по LeSS.

Принципы LeSS

Правила LeSS определяют каркасы LeSS. Но эти правила минималистичны и не отвечают на вопрос, как применять LeSS в конкретном контексте. А *принципы* LeSS служат основанием для принятия подобных решений, как поясняется и показано ниже.



Крупномасштабный Scrum — это Scrum. LeSS позволяет применять принципы, правила, элементы и цели Scrum в полном масштабе - и при этом как

можно проще¹⁴.

Прозрачность (Transparency). Основывается на ощущимых законченных элементах Беклога Продукта, коротких циклах, совместной работе, общих победах и преодолении страхов на рабочем месте.

¹⁴ "...как можно проще" - вы, наверное, уже заметили эта мысль повторяется от главы к главе. Ей пронизана вся книга и сама философия LeSS. Если вы имели возможность ознакомиться с другими подходами и каркасами масштабирования гибкой разработки, вы должны были заметить существенную разницу в простоте и доступности LeSS. Задача LeSS-практика - не натянуть существующий каркас на живого *монстра* текущей организации с ее структурами, правилами и привычками. Задача обратная - следуя советам и предписаниям LeSS, упростить организацию, так чтобы она стала более понятна и прозрачна, что включит механизмы обратной связи и начнет менять сложившуюся динамику. — Примеч. ред

Добиваться большего меньшими средствами (More with less). Нам не нужны дополнительные роли, поскольку чем *больше ролей*, тем меньше ответственности у Команд. Нам не нужны дополнительные артефакты, поскольку чем *большие артефакты*, тем больше Команды отдаляются от клиентов. Нам не нужно *большие процессы*, поскольку это снижает уровень обучения и владения процессом в команде. Вместо этого мы хотим *более ответственные* Команды, с меньшим количеством определенных внутри них ролей; Команд, которые были ли *больше ориентированными на клиентах* и создающие полезные продукты, используя поменьше артефактов; *большего владения процессом* и *более значимой работы* при наличии менее определенных (детерминированных) процессов. Таким образом, мы *стремимся добиваться большего меньшими средствами*.

Внимание на всем продукте в целом (Whole-Product Focus). Это означает наличие одного Беклога Продукта, одного Владельца Продукта, одного поставляемого продукта, одного Спринта, независимо от того, задействованы ли 3 или 33 команды. Клиентам требуется ценный функционал в целостном продукте, а не технические компоненты в отдельных его частях.

Ориентированность на заказчика (Customer-Centric). Это означает сосредоточение основного внимания на изучении реальных проблем клиентов и их разрешении. Определение ценности и потерь глазами платящих клиентов¹⁵. Сокращение времени ожидания с точки зрения клиентов. Расширение и усиление цепей обратной связи с клиентами. Все участники проекта ясно понимают, что их нынешняя работа непосредственно связана с выгодами, которые получат платящие клиенты.

Непрерывное улучшение до полного совершенства (Continuous Towards Perfection). Конечная цель совершенствования состоит в следующем: создавать и доставлять продукт практически постоянно, почти без затрат и дефектов, полностью удовлетворяя клиентов, совершенствуя окружение и делая жизнь лучше. Чтобы добиться этой цели, придется пойти на бесконечные эксперименты, начиная со скромных и кончая радикальными.

¹⁵ Под потерями (waste) здесь имеются в виду избыточные накладные расходы неоптимального процесса производства (а не прямые растраты средств или недополучения прибыли). — Примеч. ред

Бережливое мышление (Lean Thinking). Это означает создание организационной системы, основу которой составляют менеджеры-учителя, применяющие бережливый подход и обучающие ему. Они руководят с целью совершенствования, способствуют философии моментального устраниению недостатков (stop-and-fix) и практикуют принцип "увидеть все своими глазами" (go see)¹⁶. К этому следует еще добавить две столпа бережливого управления: уважение к людям и настрой постоянного ломания сложившихся устоев. Все это - только во имя конечной цели совершенствования.

Системное мышление (System Thinking). Это означает видеть, понимать и оптимизировать систему в целом, но не отдельные ее части, а также применять системное моделирование для исследования динамики поведения системы¹⁷. При этом следует избегать локальной субоптимизации, когда основное внимание уделяется эффективности и продуктивности труда отдельных работников и команд.

Клиентов волнует общий ход потока преобразования идеи в деньги , а не отдельные его промежуточные стадии.

¹⁶ "Go See" (по-японски звучит как "Genchi Genbutsu" и дословно означает "пойти на место и увидеть реальную ситуацию, чтобы понять ее") - это принцип из управления Тойоты и порождённой ей философии бережливого производства (Lean). Исходя из этого принципа менеджеры-учителя, обязаны регулярно и самолично посещать среду производства для того, чтобы получать реальную информацию о ней. Такой подход существенно отличает японскую философию управления от западного тейлоризма, где в последнем менеджеры физически отделены от работников и производства и получают периодические упрощенные отчеты о статусе *своих* проектов. Крег Ларман (со-автор книги) демонстрирует принцип "Go See" каждый раз при работе с клиентами, уделяя не меньше времени коучингу разработчиков, чем их руководителей. — Примеч. ред

¹⁷ Под системой здесь понимается все, что угодно: от абстрактного понятия до наличных средств, а также вся ее динамика, наблюдаемая во времени и пространстве, главным образом, с точки зрения клиента и пользователя.

Кроме того, локальная оптимизация отдельной части практически всегда приводит к субоптимизации целого.

Эмпирическое управление процессом (Empirical Process Control). Это непрерывное обследование и адаптация продукта, процессов, видов поведения, организационного дизайна и практик для выявления путей, подходящих в конкретной ситуации. Такой подход следует предпочесть соблюдению предписанного ряда так называемых передовых норм практики, пренебрегающих контекстом, создающих предпосылки для ритуального следования, препятствующих обучению и переменам, а также подавляющих в людях ощущение заинтересованного участия и владения.

Теория массового обслуживания (Queueing Theory). Дает представление о том, как системы с очередями ведут себя в области исследований и разработки. Подобные представления можно применять для регулирования размеров очередей, пределов незавершенных работ, многозадачности, комплексных работ и изменчивости.

Два каркаса: LeSS и LeSS Huge

В крупномасштабном Scrum есть две каркасы:

- > LeSS на 2–8 Команд;
- > LeSS Huge на 8 и больше Команд.

Термином *LeSS* обозначается как крупномасштабная инфраструктура Scrum в целом, так и меньший из каркасов.

Волшебное число восемь

На самом деле в числе *восемь* нет ничего волшебного, и если группа успешно применяет меньший каркас LeSS с большим, чем восемь, числом команд, – то и прекрасно! Но нам не приходилось наблюдать нечто подобное на практике... пока еще. Данное число просто обозначает верхний предел эмпирических наблюдений. И в некоторых случаях, как например, при разнообразных сложных целях, которые ставятся перед неопытными, иноязычными командами, работающими из разных офисах, это число может быть меньше восьми.

Но в любом случае в какой-то момент единственный Владелец Продукта может, во-первых, оказаться не в состоянии охватить понимание всего продукта, во-вторых, он не сумеет гармонично сосредоточивать свое внимание на внешних и внутренних задачах, а в-третьих, Беклог Продукта может оказаться настолько крупным, что одному человеку с ним просто не справиться.

Когда группа достигает такого переломного момента, то именно тогда, возможно, и следует сменить меньший из каркасов LeSS на LeSS Huge. С другой стороны, мы рекомендуем попробовать сначала стать лучше, меньше и проще, и уже только потом – крупнее.

Что общего у обоих каркасов

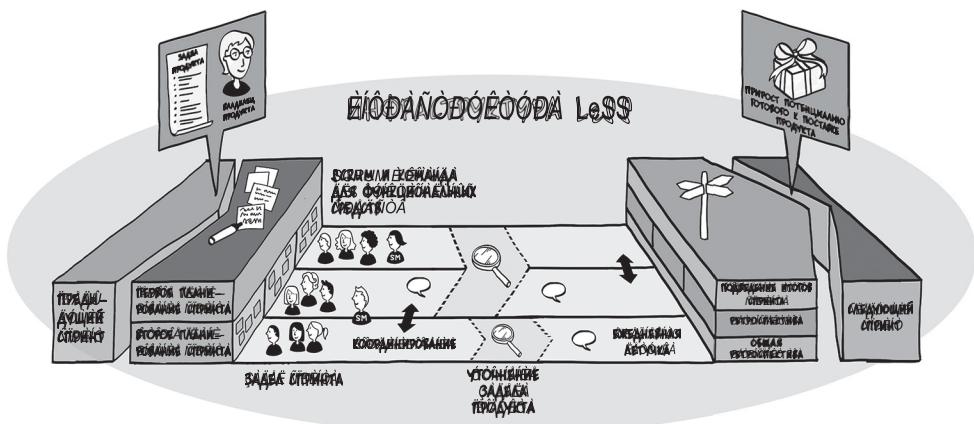
Каркасы LeSS и LeSS Huge имеются следующие общие элементы.

- > Один Владелец Продукта и один Беклог Продукта.
- > Один общий Спринт для всех команд.
- > Один готовый к поставке инкремент продукта.

Оба каркаса поясняются в двух последующих разделах. Сначала речь пойдет о меньшем LeSS каркасе, а затем о более крупном - LeSS Huge.

Каркас LeSS

Меньший каркас LeSS наглядно показан на приведенном ниже рисунке.



Краткое описание каркаса LeSS

Меньший каркас LeSS рассчитан на одного (и только одного) Владельца Продукта. Он владеет продуктом, ведет один Беклог Продукта, над которым команды работают в течение одного общего Спринта, а также оптимизирует продукт в целом. Элементы каркаса LeSS практически такие же, как и у каркаса Scrum для одной команды. Они перечислены ниже.

Роли. Это роли одного Владельца Продукта, от двух до восьми Команд и одного Scrum-мастера на одну, две или три Команды. Самое главное, что Команды являются кросс-функциональными и кросс-компонентными полно-стековыми командами, совместно работающими в среде с общим кодом и делающие все от них зависящее для создания готовых элементов (Беклога Продукта).

Артефакты. К их числу относится один потенциально готовый к поставке инкремент продукта, один Беклог Продукта и отдельные Беклоги Спринта на каждую из Команд.

События. Один общий Спринт на весь продукт, в котором участвуют все команды, а завершается он одним потенциально готовым к поставке инкрементом продукта. Дальнейшие истории и отдельные главы книги объяснят детали.

Правила и руководства. К ним относятся правила едва достаточного каркаса масштабирования для эмпирического управления процессом с вниманием на всем продукте. Руководства могут оказаться полезными.

Истории LeSS

Изучение LeSS. Один способ учиться - читать всякого рода художественные вставки, но некоторые читатели предпочтут пропустить этот раздел и сразу перейти к введению в каркас LeSS Huge, а затем к последующим главам. Те же, кому нравятся истории, продолжайте читать.

Простые истории. В таких историях не исследуются сложности крупномасштабной разработки (от установки правил до приоритезаций), которые возникают при консультировании. Эти сложности раскрываются в последующих главах. А здесь намеренно рассматриваются простые и понятные истории, знакомящие с основами Спринтов в LeSS. Если же вам нужны увлекательные диалоги и драма - тогда вам, пожалуй, больше подойдет книга по бережливому производству, на которую мы ссылаемся.

Правила и наставления. При чтении этих историй на полях можно обнаружить ссылки на связанные с ними правила и руководства LeSS. Они проясняют ситуацию и устанавливают нужные логические связи.

Две перспективы. Ниже перечислены две связанные вместе истории, по отдельности сосредоточивающие внимание на двух главных перспективах с целью упростить описание потоков:

1. Поток команд через Спринт в LeSS.
2. Поток клиенто-ориентированных элементов продукта (т.е. единиц функциональности).

История LeSS: поток команд

Эта история сосредоточивает основное внимание на потоке *команд* через Спринт, а не на потоке *элементов*. В действительности большая часть времени в Спринте отводится на решение текущих задач разработки, а не на *совещания*. Тем не менее в этой истории основное внимание уделяется совещаниям и взаимодействиям, поскольку ее цель — прояснить, каким образом многие команды должны работать вместе на протяжении событий в LeSS, и как они ежедневно координируются.

Карл¹⁸ входит в помещение, где работает его команда, под названием “Торговля” и встречает Киру¹⁹, которая говорит: “Доброе утро! Напомню, что мы - представители команд на этот Спринт, а первая часть Планирования Спринта начнется через 10 минут.” “Хорошо, — говорит Карл. — Встретимся в большом зале”.

Совет: меняйте представителей от Спринта к Спринту

Планирование Спринта, Первая Часть

¹⁸ Для того, чтобы помочь запомнить персонажи истории, в именах используется аллитерация, к примеру Сэм - Scrum-мастер, Паоло - Владелец Продукта.

¹⁹ Следуя традиции аллитерации авторов книги, оригинальную Миру пришлось переименовать в Киру (так как она участница Команды). Так же как и оригиналный Марк стал Карлом. Карлом, Марк! Но надеемся, что сути истории это ничуть не поменяло! К слову, этот же принцип часто используется для персонализации пользователей продукта: в зависимости от их роли им даются соответствующие имена. — Примеч. ред

(см. раздел “Руководство: Планирование Спринта Часть Один” в главе 12)

Наступает время для проведения общей Первой Части Планирования Спринта (Sprint Planning One)²⁰. В большом зале собираются десять представителей пяти команд данной продуктовой группы. Все они работают над флагманским продуктом для торговли облигациями и производными финансовыми инструментами (деривативами). Сэм, Scrum-мастер команд “Торговля” и “Маржа”, тоже здесь. Он собирается наблюдать за происходящим и коучить по мере необходимости.

Много Спринтов назад все члены команд посещали эту часть Планирование Спринта. Это было полезно, когда группа еще не умела прояснить и готовить элементы и еще не имела обширных знаний среди команд. В те времена Первая Часть Планирования Спринта служила для поиска ответов на главные вопросы, которые все должны были услышать. Но в дальнейшем положение намного улучшилось, и теперь группа экспериментирует со ротацией представителей, что сделало совещание проще и быстрее, где поднимается несколько менее важных вопросов. Если же новый подход не зарекомендует себя, он будет, вероятнее всего, вынесен на обсуждение на Общей Ретроспективе, и тогда может быть начат другой эксперимент для Планирования Спринта.

Правило: на уровне продукта существует лишь один Спринт, а не разные Спринты для каждой из Команд.

Правило: Планирование Спринта состоит из двух частей. Первая Часть является общей для всех команд, тогда как вторая обычно проводится отдельно для каждой команды. Первую часть Планирования для многих команд следует проводить в общем пространстве для обсуждения тесно связанных между собой элементах продукта.

²⁰ Традиционно в Scrum каркасе планирование спринта разделяется на две части. Польза от такого дизайна процесса становится наиболее очевидной, когда мы начинаем проводить многокомандные планирования. В LeSS для краткости эти части планирования спринта называются просто "Sprint Planning One" и "Sprint Planning Two". Но во избежание ложной трактовки (будто в каркасе LeSS два совещания по планированию спринта, это не так, оно одно, но из двух частей), в этой книге мы решили использовать чуть менее лаконичные, но недвусмысленные: "Планирование спрингта часть один" и "Планирование спрингта часть два". —

Примеч. ред

Правило:

Планирование Спринта Часть Один посещают Владелец Продукта и Команды или их пред-ставители. Совместно они предварительно выбирают элементы, над которыми предстоит работать каждой команде в следующем Спринте.

Входит Паоло и здоровается с присутствующими. Он является Владельцем Продукта и главным ответственным за выпуск продукта²¹. Паоло раскладывает на столе 22 карточки и говорит: “Вот крупные темы: немецкий рынок, управление заказами и некоторые регулятивные отчеты. Я разложил их в своем порядке приоритетов. Полагаю, что всем понятен такой порядок приоритетов, поскольку мы его уже достаточно подробно обсуждали при Прояснении Беклога Продукта. Пожалуйста, спрашивайте, если что-то не ясно.”.

**Совет:** Команды выбирают свои элементы.

Кира и Карл подходят к столу (вместе с представителями других команд) и выбирают две карточки для элементов, связанных с облигациями на немецком рынке. В течение двух по-

²¹ В продуктовых компаниях роли *продуктового менеджмента* или *маркетинга продуктов* (в сотрудничестве с командами) сфокусированы на определении видения (концепции) продукта и направления разработки, поощрении нововведений, анализе конкурентов, выявлении потребностей заказчиков и тенденций рынка. А во внутренних группах разработки эту роль может выполнять ведущий пользователь из операционной бизнес группы. Владелец Продукта в Scrum и LeSS, как правило, выбирается из тех, кто выполняет эти роли. В данном случае роль Владельца Продукта выполняет Паоло — главный продукт менеджер. Подробнее об этом речь пойдет в главе 8.

следних Спринтов их команда рассматривала эти элементы во всех подробностях на однокомандных рабочих сессиях по Прояснению Беклога Продукта (Product Backlog Refinement, или PBR).

См. раздел

“Руководство:

Многокомандное

Прояснение

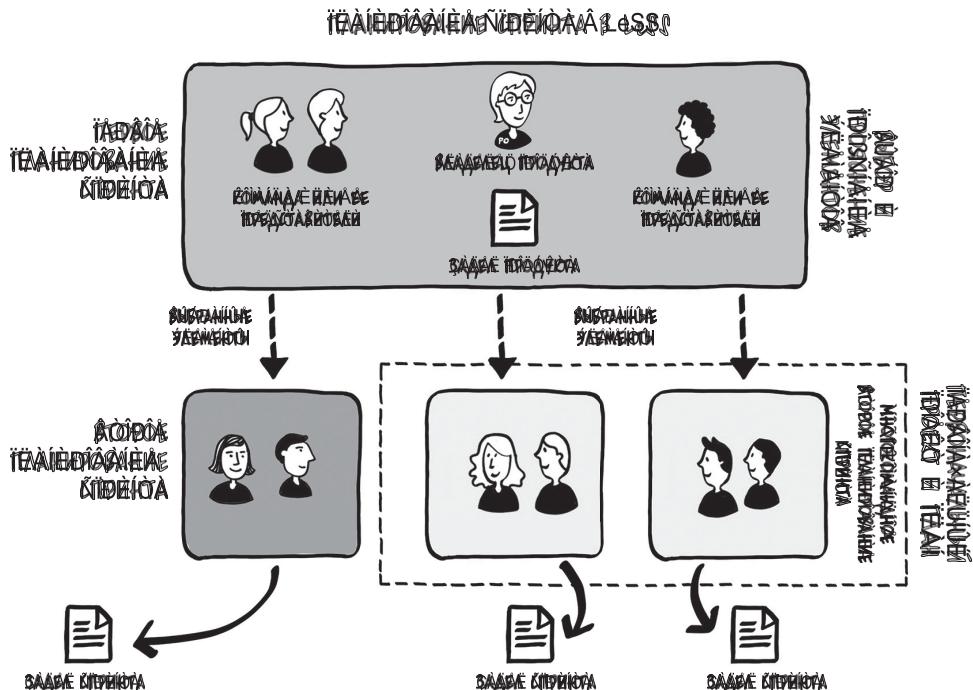
Беклога Продукта” в
главе 11.

Они выбирают еще две карточки для элементов, связанных с управлением заказами, в котором довольно хорошо разбираются обе Команды “Торговля” и “Маржа”. Дело в том, что обе эти команды работали вместе над этими элементами на многокомандных сессиях по Проработке Беклога Продукта. При этом ко-

манды стремились выбрать для себя элементы как можно позже, отложив решение до момента, когда будет проведено Планирование Спринта. Благодаря этому повышается *гибкость* группы, т.е. ее способность легко реагировать на изменения, а широкое понимание всего продукта благоприятствует самоорганизованному координированию действий в группе.

Минуту спустя, просмотрев карточки других групп, Кейт из группы “Маржа” задает их представителям следующий вопрос: “Вы не против, если мы сделаем этот отчет? Нам уже приходилось делать нечто подобное в последнем Спринте, и я уверена, что мы сможем справиться быстро. Можно ли поменять этот элемент немецкого рынка?” И они соглашаются. Через несколько минут команды завершают выбор и смену элементов, исходя из своих интересов, возможностей и пожеланий сгруппировать связанные вместе элементы, чтобы сосредоточить на них основное внимание.

Совет: не предрешайте распределение элементов по командам.



Scrum-мастер Сэм говорит: “Должен заметить, что у Команды “Маржа” имеются четыре высокоприоритетных элемента. Не может ли это стать проблемой?” В течение краткого обсуждения группа осознает, что один из высокоприоритетных элементов разрабатываемого продукта может быть упущен, если дело у Команды “Маржа” не заладится. Поэтому присутствующие решают распределить несколько высокоприоритетных элементов между командами (ограничившись знаниями каждой команды, чтобы сделать более вероятным завершение высокоприоритетных элементов).

См. раздел
“Руководство: пять инструментальных средств Scrum-мастера” главы 6.

Совет: распределяйте высокоприоритетные элементы.

Представители команд выбрали в общей сложности 18 карточек, оставив четыре низкоприоритетных элемента на столе. Паоло просматривает карточки невыбранных элементов, выбирает две из них и говорит: “Эти два элемента крайне важны для меня в этом Спринте. Я мог бы, конечно, повысить их приоритет, но не сделал этого, но теперь я не прочь передумать. Давайте попробуем каким-то образом поменять их на элементы из тех, что вы уже выбрали. И если, конечно, вашей команде удастся завершить работу раньше, возьмите, пожалуйста, невыбранные элементы”. Как только данное затруднение разрешается, Паоло говорит:

Правило: команды оценивают возможности для совместной работы и проясняют завершающие вопросы.

Совет: расходитесь, чтобы прояснить дело.

“Что ж, уделим немного времени разрешению накопившихся вопросов. Как вы знаете, я уделял больше внимания выяснению приоритетов элементов, и большинство из вас знают эти элементы лучше, чем я, но давайте все же по-смотрим, что мы можем сделать вместе, чтобы хотя бы немного прояснить, что осталось”.

Между тем, Кира, Карл и другие участники совещания напряженно размышляют над завершающими незначительными вопросами, чтобы прояснить выбранные ими элементы, и пишут некоторые из этих вопросов на перекидных листах бумаги, закрепленных на стенах. Паоло обходит их, обсуждая вопросы. Участники перемешиваются и помогают, в тех вопросах, которые могут разрешить. После 30 минут все незначительные вопросы, которые могли быть разрешены, закрываются.

Группа становится в круг для обсуждения. Но никто не поднимает никаких вопросов координации, и поэтому Сэм наконец-то берет на себя инициативу, говоря: “Должен заметить, что Команды “Торговля”, “Маржа” и “Не Деривативы” выбрали сильно связанные элементы управления заказами”. На это Кира заявляет: “Так давайте объединим усилия Команд “Торговля”, “Маржа” и “Не Деривативы” на многокомандной второй части Планирования Спринта! Ведь у вас уже были возможности работать вместе”. Все соглашаются, и на этом совещание завершается.

Планирование Спринта Часть Два - однокомандный и многокомандный варианты

(см. раздел “Руководство: многокомандное Планирование Спринта Часть Два” главы 12)

Правило: у каждой Команды имеется свой Беклог Спринта.

Правило: проводите многокомандную вторую часть Планирования Спринта в общем пространстве для тесно связанных элементов продукта.

совещании по многокомандной Проработке Беклога Продукта. Они также предвидят определенную ценность в тесном сотрудничестве.

После перерыва две из пяти команд проводят свои совещания по однокомандному по второй части Планирования Спринта. Их цель — создать Беклоги Спринтов, спроектировать и спланировать свою работу в Спринте.

В свою очередь, Команды “Торговля”, “Маржа” и “Деривативы” проводят в большом зале совместное совещание по второй части Планирования Спринта, поскольку они реализуют сильно связанные элементы, которые были предварительно прояснены на общем

Сначала они проводят предварительное обсуждение в течение 10 минут, определяя совместные виды работ (общие задачи) и выясняя вопросы проектирования. Затем они приступают к проектированию, ограниченному по времени 30-минутными

Совет: Проводите совместные рабочие сессии по проектированию и обсуждению совместной работы

рамками, соглашаясь, что ход дискуссий лучше *визуализировать*: больше рисования на доске, меньше разговоров без рисунков. В ходе дискуссии обнаруживается дополнительная общая работа, которая также отображается на маркерной магнитной доске.

Динь-дилинь! По прошествии 30 минут все еще остается немало неисследованных подробностей, тем не менее, команды двигаются дальше. Каждая из команд переходит в свой угол большого зала, где они сосредоточиваются на своей сфокусированной части Планирования Спринта, обсуждая более подробно вопросы проектирования и создавая свой Беклог Спринта на карточках. Дальнейшее координирование на совещании осуществляется продвинутой разновидностью метода “просто поговорите”, известного в LeSS, - “просто покричите”.

В ходе обсуждения у команд возникает потребность в проведении более основательной Сесии по Проектированию. И они соглашаются провести ее позже в тот же день.

См. раздел
“Руководство: ведение Беклога Спринта без программных средств” главы 12.

См. раздел
“Руководство: Просто Поговорите” главы 13.

Многокомандная Рабочая Сессия по Проектированию

(см. раздел “Руководство: многокомандная рабочая сессия по проектированию” главы 13)

После Планирования Спринта и очередного перерыва Кира и Карл из Команды “Торговля”, а также представители команд “Маржа” и “Не Деривативы” проводят в течение одного часа многокомандную Рабочую Сессию по Проектированию, чтобы подробно обсудить общий и согласованный дизайн системы²². Собравшись у большой маркерной доски, они набрасывают эскизы и обсуждают вопросы проектирования, чтобы прояснить и согласовать общий подход к проектированию и общие технические задачи. К счастью, выводы, к которым они приходят, не оказываются серьезного влияния на планы Спринта, но тем не менее они ощущают неловкость, признавая, что могли бы предвидеть потребность разрешить столь крупные вопросы дизайна раньше.

Активности по разработке, поддерживающие

²² Здесь и далее (если не сказано обратное) под “дизайном” понимается внутреннее устройство продукта (дизайн системы), а не визуальный дизайн (пользовательских интерфейсов). — Примеч. ред

Координирование и Непрерывную Поставку

После Планирования Спринта команды приступают к разработке элементов продукта, делая акцент на *коммуникации через код*. Все команды осуществляют *непрерывно интеграцию*. Благодаря непрерывной интеграции всего кода от всех команд, создается возможность для сотрудничества, так как видно кто еще вносит изменения в компонент, над которым ведется работа. И это очень удобно, поскольку

См. раздел
“Руководство:
Коммуникация через
код” главы 13.

См. раздел
“Руководство:
Непрерывная
Интеграция” главы 13.

Правило: отдавайте предпочтение децентрализованному и неформальному координированию над централизованным.

См. раздел
“Руководство:
Просто Поговорите”
главы 13.

ему стало известно, что *сейчас* требуется координирование и также понятно с *кем*, он сразу же отправляется по коридору с визитом в Команду “Маржа”, чтобы *просто поговорить* о том, как организовать совместную на пользу всем.

Для элемента, разрабатываемого в команде “Маржа” (а по существу, для всякого элемента в каждой команде), прежде чем приступить к разработке кода конкретного решения, были написаны автоматизированные приемочные тесты. Таким образом, помимо непрерывной интеграции кода, команды осуществляют интеграцию автоматизированных тестов. Подобные приемочные тесты запускаются часто членами команды, и если какой-нибудь из тестов не проходит, то это служит команде явным сигналом для координирования действий. Сам код извещает команду: “Эй! Есть проблема!”. Остается поговорить и все уладить..

Правило: цель совершенствования — улучшить Критерии Готовности таким образом, чтобы результатом каждого Сprintа (или еще чаще) стал готовый к поставке продукт.

интеграция используется в группе как способ извещения и поддержки координации.

Например, сутра на второй день Сprintа Карл, разработчик из Команды “Торговля”, локально скачивает последнюю версию и быстро проверяет недавние изменения в компоненте, над которым работает его команда. При этом он обнаруживает изменения, внесенные в код Кириллом из Команды “Маржа”. Он знает, что эта команда работает над сильно связанным элементом, и поэтому не особенно удивлен. Поскольку посредством коммуникации через код

Естественно, еще одним преимуществом практики, применяемой в группе для непрерывной интеграции, автоматизированного тестирования и оперативного устранения неполадков всякий раз, когда не проходит сборка, заключается в том, что разрабатываемый продукт более или менее постоянно готов к передаче в эксплуатацию. Для интеграции или тестирования не существует отдельных команд, поскольку это добавило бы в процесс задержек, передач частично завершенной²³ работы и сложности.

²³ Здесь авторами используется термин "handoff" - передача информации от одного субъекта процесса другим. С точки зрения бережливого производства (lean) транспортировка материалов или информации приводит к потерям (waste) - это траты времени, усилий, а в случае информации - еще и искажение и неверная трактовка. Поэтому в LeSS команды делают работу от начала до конца (end-to-end) с минимальными пассами другим командам. — Примечание редактора

Общая Ретроспектива

(см. раздел “Наставление: Общая Ретроспектива” главы 14)

Правило: Общая Ретроспектива проводится после Командных Ретроспектив для обсуждения межкомандных и системных вопросов, а также для планирования

На второй день спринга Сэм и еще один Scrum-мастер, Владелец Продукта Паоло, координатор офиса и представители из большинства команд - все собираются вместе на не более чем 90-минутную Общую Ретроспектиvu предыдущего Спринга.

Почему участники проекта не провели Общую Ретроспективу *до* начала нового Спринга? Они могли бы это сделать, но, как правило, они завершают Спринг в пятницу и начинают новый Спринг в понедельник, несмотря на предложение Сэма попытаться завершить Спринг в среду или четверг. А в последнюю пятницу они провели как Ретроспективу Спринга, так и

Ретроспективы на уровне команд, после чего им уже недостало сил на проведение Общей Ретроспективы в конце дня. Поэтому они дружно решили перенести совещание на начало следующего Спринта. Сам же Сэм считает, что такая задержка не на пользу, а Планирование Спринта следовало бы начать немного позже после данного совещания. Но он хочет, чтобы группа пришла к этому самостоятельно.

Группа сосредоточивает основное внимание на системных вопросах и улучшениях: как координировать действия, обмениваться информацией и решать вопросы во всей группе на протяжении Спринта? Раньше они пробовали проводить совещания на уровне "Scrum-of-Scrum", но нашли такие встречи не особенно эффективным. Сэм поясняет методику "Open Space", и они соглашаются опробовать ее в текущем Спринте.

Активности по Координации (см. главу 13)

Четвертый день демонстрирует разнообразные идеи по координирования в LeSS.

В LeSS каждая команда как обычно проводит свой Ежедневный Scrum (Daily Scrum). Для поддержки координации между Командами "Торговля" и "Маржа" Кира берет на себя роль *скаута-разведчика*, чтобы понаблюдать за Ежедневной Scrum встречей в Команде "Маржа", а затем возвращается и делится со своей командой тем, что ей удалось выведать. И кто-то из Команды "Маржа" делает то же самое.

По предварительной договоренности на Общей Ретроспективе группа проводит 45-минутное совещание в стиле "*Open Space*" для координирования действий и обучения с напитками и закусками перед его началом. Сэм берет на себя функции фасилитатора, чтобы научить группу проводить подобные совещания. На это совещание приглашаются все, но большинство команд решает послать лишь несколько представителей. Кира и Карл приходят на совещание от имени Команды "Торговля". Группа собирается проводить "*Open Space*" раз в неделю.

Сообщество по тестированию, которое составляют добровольцы из большинства команд, собирается на полчаса, чтобы послушать предложение Кейт опробовать новое инструментальное средство для проведения автоматизированных приемочных тестов. Они с воодушевлением принимают предложение, и Мэри от имени своей Команды "Маржа" берет на себя добровольное обязательство по проведению экспериментов в следующем Спринте, поскольку они по-настоящему заинтересованы обучиться этому.

экспериментов по улучшению. Это совещания посещают Владелец Продукта, Scrum-мастера, представители Команд и руководители, если нужно.

См. раздел
"Руководство:
Улучшайте
Систему" главы 14.

Правило: решение о способах межкомандной координации - в руках у самих команд..

См. раздел
"Руководство:
Скауты" главы 13.

См. раздел
"Руководство: Open
Space" главы 13.

См. раздел
"Руководство:
Сообщества" главы 13.

Кира является членом проектно-архитектурного *сообщества*. Для обсуждения общей архитектуры в текущем Спринте не требуется проводить Рабочую Сессию по Проектированию, но она считаем нужным провести в следующем Спринте полудневный *спайк*²⁴ для освоения новой технологии. Она делится своей идеей с сообществом с помощью инструмента общения, предлагая провести спайк вместе с коллективным mob-программированием²⁵ для повышения результаты изучения.

²⁴ Spike - одна из инженерных практик Экстремального Программирования (XP), следуя которой разработчик, встречающийся с задачей с повышенной технической неопределенностью (подключение новой библиотеки, интеграция с новым внешним интерфейсом и т.д.) реализует быстрый *прототип на выброс*, который в простейшем виде реализует необходимую задачу с точки зрения пользователя сквозь все слои архитектуры системы. Само слово "спайк" означает длинный шип (костыль), который вбиваются в дерево, с целью его *подрасколоть* перед работой более мощными инструментами (топором). Поэтому программистский спайк - это реализация функционала "вертикально" через разные слои, но при этом минимальной "толщиной". — Примеч. ред

²⁵ Mob-programming - расширение "парного программирования", когда код, разрабатываемый текущим по очередности программистом (или большой монитор) в комнате, где присутствуют другие программисты, задача которых обсуждать и решать, что и как писать дальше. В терминах парного программирования - в группе одновременно работают над одной задачей: один драйвер (владеет клавиатурой) и много навигаторов (решающих, что драйверу писать). — Примеч. ред

Совет: Организуйте архитектурное сообщество.

Совет: Оперативно устраняйте обнаруженные неполадки.

Совет: Эксперты должны учить остальных.

Правило:
Выясняйте возникающие недостатки в идеале между Командами и пользователями и прочими заинтересованными сторонами.

Совет: Ранняя обратная связь.
См. раздел "Наставление: поддержание связи через код" главы 13.
См. раздел "Наставление: интеграция непрерывно" главы 13.

ка, а следовательно, самое время *остановиться и устраниТЬ ее!* В текущем Спринте эта обязанность возложена на Команду "Торговля", аней — на Карла, так как это его второй конек. Поэтому он вызывается устраниТЬ ошибку, приглашая себе в пару еще одного члена команды, чтобы его коллега мог тоже научиться решать подобные проблемы.

В дальнейшем Кира и ряд других участников команды посещают группу поддержки и обучения клиентов, которая тесно взаимодействует с реальными пользователями. Ее команда завершила работу над своим первым элементом, и поэтому им требуется получить как можно раньше обратную связь от людей, близких к клиентами. Один из инструкторов по обучению свободен и пробует новый функционал. Команда "Торговля" уходит с рядом новых идей по улучшению..

Чуть позже в тот же день Карл и остальные члены Команды "Торговля" решают задачи нового элемента. Карл только что завершил 10-минутный цикл разработки через тестирования (TDD) и почистил стабильный код после внесения незначительных корректировок. И вот снова - каждые 10 минут он посыпает крошечные изменения кода в центральный репозиторий (в голову главной ветки - "head of trunk") для *непрерывной интеграции* как со своей командой, так и с другими. Затем он бросает взгляд на большой красно-зелеными экран на стене и замечает, что система сборки проходит все тесты во всей группе

Общая Проработка Беклога Продукта

(см. раздел "Руководство: виды Проработки Беклога Продукта" главы 11)

Правило: проводите многокомандную или общую Проработку Беклога Продукта (PBR), чтобы расширить представление о продукте и воспользоваться возможностями координирования при наличии тесно связанных элементов или потребности в более обширной входной информации и обучении.

На пятый день Карл и Кира посещают совещание по *общей Проработке Беклога Продукта* вместе с представителями всех команд и Паоло, Владельцем Продукта. Паоло начинает совещание, делясь своими текущими мыслями о направлении развития продукта и дальнейших краткосрочных шагах, и самое главное — о том, *зачем и почему*. Чтобы присутствующим стали понятнее его рассуждения, он показывает свою модель приоритизации вместе с группой, где во внимание принимаются: влияние продукта на прибыль и клиента, технический бизнес риски, стоимость задержки и прочие факторы.

Паоло просит группу высказывать соображения и отзывы на предстоящее направление развития продукта, и группа обсуждает какие элементы следует проработать первыми. И хотя Паоло известно, что окончательное назначение приоритетов придется делать именно ему, он делает все возможное, чтобы команды поняли ход его рассуждений и чтобы он сам что-то узнал и научился чему-то от них. Он хочет, привлечь команду ко *владению продуктом*.

Затем группа разбивает ряд новых крупных элементов, выполняя легковесное прояснение деталей (подробнее об этом речь пойдет далее) и оценивание сложности при помощи покера для планирования - все это, чтобы узнать больше о предстоящей работе, а не просто формально произвести оценки.

Представители трех команд, включая команды “Торговля” и “Маржа”, решают провести чуть позже многокомандный PBR для некоторых элементов, чтобы повысить общее представление о них, поскольку они сильно связаны. А представители двух других команд выбирают элементы, на которых сфокусироваться отдельно на командных Проработках Беклога Продукта.

Многокомандная и Однокомандная Проработка Беклога Продукта (см. раздел “Руководство: многокомандная Проработка Беклога Продукта” главы 11)

На шестой день все члены трех команд собираются в большом зале на *многокомандаая PBR сессия..*

И хотя основное занятие их компании — создание и продажа решения по торговле ценными бумагами, в ней все же имеется небольшая группа трейдеров, которые его серьезно используют, но все таки оперируют небольшими позициями на бирже с небольшими рисками для бизнеса. Таким образом у компании есть предпосылки для лучшего понимания тенденции на рынке и прекрасная возможность для диалога между командами разработчиков и опытными пользователями.

Совет: ротируйте представителей каждый Спринт.

См. раздел “Руководства: Приоритизация вместо Прояснения” главы 8.

См.раздел
“Руководства: пять взаимоотношений”
главы8.

Совет: Владелец Продукта вовлекает команды к владению продуктом.

См. раздел
“Руководство:
Разбитие
Элементов
Беклога” главы 11.

См. раздел
“Руководство:
оценивание
сложности в
контексте
масштабных
продуктов” главы 11.

Правило: вся приоритизация проходит через Владельца Продукта, но прояснение деталей по возможности проходит непосредственно между Командами, клиентами, пользователями и прочими заинтересованными сторонами.

Таня и Тед являются трейдерами облигаций, сообщившими Паоло о тенденции на рынке ценных бумаг, которая привела к проработке конкретных элементов на многокомандной PBRсессии. Поэтому они оба подключаются в качестве экспертов, чтобы помочь командам изучить и прояснить новые элементы.

Две другие команды, общаясь с другими трейдерами, проводят отдельные PBR сессии для полного прояснения некоторых уже частично детализированных элементов и ряда новых.. Кроме того, один из трех юристов компании, специализирующийся на финансовом регламенте и нормативноправовом соответствии, присоединяется к одной из этих команд, чтобы помочь им в прояснении..

См. раздел
“Руководство:
инструментальные
средства для ведения
крупных беклогов
продуктов” главы 9.

Совет: пользуйтесь
вики-страницами для
подробного изучения
элементов.

На последней стадии PBR совещания его участники делают фотографии всех стен и досок. Они размещают эти фотографии на своих wiki-страницах, предназначенных для регистрации всего, что известно о каждом элементе. Кроме того, они обновляют и редактируют текст и таблицы на wiki-страницах, которые были бегло напечатаны во время обсуждений.

Беседа о Беклогах и Владельцах Продуктов командного уровня

После совещания по многокомандному PBR, Карл (который только недавно начала работать в компании) встречается с Сэмом за чашкой кофе и затевает с ним разговор. Карл говорит: “Сэм, меня интересует твое мнение кое о чем. На только что завершившемся совещании по уточнению я, конечно, заметил, что мы работали непосредственно с некоторыми трейдерами облигациями, чтобы вместе прояснить непонятные моменты. Но разве это эффективно? На моей предыдущей работе у каждой команды был свой Владелец Продукта, который писал истории, составлял wireframes-эскизы и спецификации, а затем передавал их нам на реализацию. Это давало нам возможность сосредоточиться непосредственно на программировании. И каждая команда имела свой Беклог Продукта, в котором командный Владелец Продукта выставлял приоритеты. Но здесь я этого не вижу. Почему здесь все так по-другому?”

На это Сэм отвечает: “Интересные вопросы. Ты не против, если я задам тебе пару вопросов в ответ, чтобы мы вместе поисследовать эту тему?”

“Конечно, давай”.

“Давай сперва посмотрим на один Беклог Продукта в сравнении со множеством беклогов на уровне команд. Допустим, у каждой команды есть свой беклог. Насколько просто и эффективно одному действительно всеобщему Владельцу Продукта держать в голове общую картину? И сколько знаний будет у каждой команды о требованиях и дизайне элементов в беклогах других команд?”

Карл отвечает: “Ну это очевидно из моего опыта на предыдущем месте работы, я могу ответить вполне определенно: немного”.

Сэм продолжает: “А теперь допустим, что есть восемь команд с восемью беклогами. Что, если с точки зрения компаний или продукта по какой-нибудь

причине элементы в двух из восьми беклогах окажутся существенно более важными или высокоприоритетными. Возможно, эта ситуация возникнет в связи с какими-нибудь переменами на рынке. И тогда к тебе несколько вопросов: сумеют ли шесть команд, работающих над низкоприоритетными беклогами, легко переключиться к работе над высокоприоритетными элементами в двух других беклогах? И сможет ли эта группа вообще обнаружить эту проблему, тогда как она ограничена тем, что каждая команда работает над своим беклогом со своими локальными приоритетами?"

Карл отвечает: "Наши команды на моем прежнем месте работы работали только над элементами из своего беклога. Они не могли переключиться на другие. Но зачем это им вообще нужно? Это же неэффективно?"

Сэм отвечает: "Что ж, в такой ситуации и правда может казаться что команды работают "эффективно". Но заметь, большинство - только над низкоприоритетными заданиями. Это связано с тем, что вы накопили много узконаправленных знаний, когда каждая команда на долго сосредоточилась на отдельном беклоге, и также потому, что общий приоритет и общая картина продукта просто никому не видны. Позволь мне задать тебе ряд вопросов. Кажется ли это тебе больше *негибким* или *все-таки гибким* - agile-подходом? И насколько такая система оптимизируется, чтобы люди работали над высокоприоритетными заданиями с точки зрения компании в целом?"

Задумавшись, Карл отвечает: "А, я понял. Это, конечно же, *нисколько не гибко*, хотя в нашей группе считали, что они "*работают по agile*". Мы вообще, выходит, не могли реагировать на изменения, представлявшие наибольшую общую ценность. А мой прежний командный Владелец Продукта говорила, что приоритезирует так, чтобы наша команда в каждый момент времени работала над самым важным с точки зрения нашего беклога. Но теперь я понимаю, что наша команда, похоже, напряженно и эффективно работала - но, увы, только над тем, что имело низкую ценность, если посмотреть на это с уровня выше".

Сэм говорит: "Вот именно. И это одна из нескольких причин, по которым у нас здесь имеется лишь *один* Беклог Продукта и отсутствуют беклоги команд, несмотря на то, что команд у нас много. Если кратко, то такая организация процесса поддерживает фокус на всем продукте в целом, оптимизации системы и гибкости. И конечно же становится легче видеть, что происходит во всей группе".

"А кроме того, — комментирует Карл, — я заметил, что в моей предыдущей компании всем командам было намного труднее работать реально *вместе*, поскольку мы работали над совершенно разными целями в асинхронных Спринтах. А здесь, чувствуется, как у всех команд у нас всех, в едином Спринте есть общий сфокус и направление".

"Верно! — восклицает Сэм и затем продолжает. — А вот еще один вопрос: если по факту должен быть только один Беклог на весь продукт и один реальный Владелец Продукта, чтобы определять в нем приоритеты, но у каждой команды по-прежнему имеется свой, так называемый, Владелец Продукта, который по определению не может приоритезировать на уровне команды, то чем он вообще занимается тогда целый день?"

Карл отвечает: "Что ж, на моей прежней работе в обязанно-

Правило:
существует только
один Владелец
Продукта и только
один Беклог
Продукта для всего
поставляемого
продукта.

сти Владельцы
Продукта на уровне
команды входили
общение с

Правило: Владелец Продукта не должен работать сам над проработкой Беклога Продукта. Его поддерживает множество Команд, работая непосредственно с клиентами, пользователями и прочими заинтересованными сторонами.

пользователями и написание историй для своей команды, чтобы они могли эффективно сосредоточиться на программировании, в то время, как как Владелец Продукта команды занимается сбором и описанием требований”.

Правило: все приоритеты назначаются Владельцем Продукта, а прояснение происходит как можно более напрямую между Командами и клиентами, пользователями и прочими заинтересованными сторонами.

“А нынче на PBR совещании, — спрашивает Сэм, — ты же общался с трейдерами, которые там присутствовали?”

“Дай-ка вспомнить, — отвечает Карл. — Да, я действительно разговаривал с Таней по поводу ее идеи проанализировать торговлю корпоративными облигациями российских компаний. Этот вопрос показался мне не совсем понятным, и я попросил у нее пояснений. Она объяснила, что все дело в отмывании денег на офшорных счетах. Но она не знала, что мы как раз недавно работали над другой функциональностью,, которая об интеграции с новыми базами данных для доступа к регламентам ЭС и США для проверки таких рисков. Так что я предложил ей другой подход к решению данной проблемы - лучший, на мой взгляд, - с чем она и согласилась”.

“А теперь, когда я об этом думаю, — размышляет он вслух, — такое на моей прежней работе вряд ли бы произошло, поскольку я редко общался непосредственно с пользователями”.²⁶

Дальнейшая разработка

Каждую минуту, день за днем, команды разрабатывают код, непрерывно интегрируя его совместно с прогонкой полного набора автоматических тестов. Когда же сборка не проходит, они останавливаются и устраниют проблемы, постепенно приближаясь к своей цели совершенствования - постоянно иметь готовый продукт, который они могли бы поставлять заказчикам. Таким образом, когда Спринт почти завершен и команды готовятся присоединиться к Ревью Спринта, им не нужно в сумасшедшей спешке интегрировать и тестировать большие куски кода, - он интегрировался и тестировался на протяжении всего Спринта.

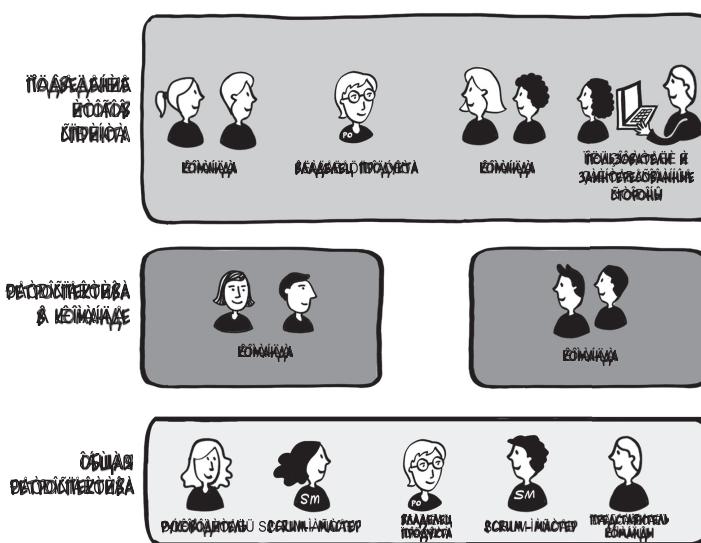
Ревью Спринта

(Ревью и Ретроспективы, см. главу 14)

²⁶ Этот диалог, часть большей истории, не просто демонстрирует позицию Scrum-мастера как коуча и наставника по отношениям к членам команды. Он также показывает, как коучу в LeSS рекомендуется подходить к работе с клиентом: его задача не научить клиента каркасу и правилам LeSS - нет. Коуч помогает клиенту осознать системную динамику своей организации и то, к каким результатам она сейчас приводит (в этом примере это тот "ага" момент, когда Карл понимает, что в его предыдущей компании наличие множества Владельцы Продуктов и Беклогов создавало иллюзию приоритезации и эффективности). Без подобных личных прозрений невозможна полноценная трансформация организации, ее ценностей и культуры. Представьте силу подобного диалога, когда на месте Карла были бы директора компаний, которые по-настоящему заинтересованы в повышении эффективности бизнеса. Это может быть кардинальным событием в истории конкретной организации. — Примеч. ред

Правило: для всех команд проводится одно общее Спринт Ревью.

Наконец-то, последний день и самое время для общего Ревью Спринта. Кто в нем участвует? Паоло (Владелец Продукта и главный продукт-менеджер), все внутренние трейдеры облигаций, ряд инструкторов по обучению и представителей служб по работе с клиентами, несколько человек из отдела продаж и четыре пользователя от внешних клиентов, которые платят меньшие ежегодные взносы и в обмен за это регулярно участвуют в подобных мероприятиях. Здесь также присутствуют все члены команд.



См. раздел
“Руководство:
Сprint Ревью в
формате
ярмарки” главы
14.

Так как рассмотрения сегодня ждет много элементов, группа начинает одновременную ярмарку, напоминающему также выставку научных достижений, где для рассмотрения разных элементов в зале разложено и расставлено множество устройств. Некоторые члены команд стоят в заранее установленных местах для сбора отзывов и обратной связи, тогда как все остальные пользуются новыми функциями продукта и обсуждают их.



Совет: обсуждайте направление предстоящих спринтов.

Через час группа собирается вместе в сессию для обсуждения вопросов и отзывов, проводимую Паоло. После чего они обсуждают будущее направление развития продукта. Паоло делится с присутствующими тенденциями на рынке и положением дел у конкурентов, а также своими мыслями о том, куда двигаться дальше с продуктом, после чего он просит у них совета.

Представители команд

Правило: в каждой команде проводится своя Ретроспектива Спринта.

было далеко не идеально, поскольку рассмотрение существенных вопросов было отложено до последнего момента. А ведь эти вопросы могли серьезно застопорить и усложнить разработку. Таким образом, они решают, что в следующем Спринте на сессии Проработки Беклога Продукта сделают все возможное, чтобы определить все существенные элементы, которые стоит обсудить с другими командами. И в этом случае провести многокомандную Рабочую Сессию Проектирования как можно раньше.

После перерыва Команда “Торговля” (и все остальные команды) самостоятельно проводят Ретроспективу Спринта. Они решают, что проведение Рабочей Сессии по Проектированию с командой “Маржа” после, а не до Планирования Спринта

Завершение

Рекомендация: бельгийское пиво марки Tripel Karmeliet.

Итак, Спринт завершен! Сэм приглашает Команду “Торговля” присоединиться к нему и Кире в соседнем бельгийском пивном паб за углом, чтобы отпраздновать день рождения Кирры в ее любимом заведении

Краткие итоги

Главные выводы из рассмотренной выше истории.

- > В данной истории акцент делается на поток людей и команд через Спринт в LeSS.
- > Элементы истории связываются с конкретными руководствами и правилами LeSS.
- > Читателю, знающему Scrum, эти мероприятия должны быть знакомы.
- > Данная история демонстрирует фокус на всем продукте в целом — даже при наличии многих команд.
- > Эти мероприятия подчеркивают обучение и координацию на уровне команд.
- > Разрабатываемые элементы продукта непрерывно интегрируются, чтобы помимо непрерывной поставки коммуникация через код еще и поддерживала децентрализованное координирование и практику "просто поговорите".
- > Команды непосредственно проясняют детали пользователями и заказчиками, чтобы уменьшить передачи и расширить понимание, эмпатию и чувство со-владения.

История в LeSS: поток элементов

Эта история сосредоточивает основное внимание на потоке элементов (функциональных возможностей продукта) через часть Спринта, главным образом, на стадии проработки и разработки.

Петра завершает встречу с чиновниками из регулятивного органа и направляется в аэропорт, домой. Она является еще одним продукт-менеджером, помогая Паоло в области регулирующих норм и аудита²⁷.



Позже Петра встречается с Паоло. Делая заметки на карточках, она подытоживает новые правила, которые должны, по ее мнению, повлиять на разрабатываемый продукт и, прежде всего, на функционал, которые по ее мнению, клиент захотят увидеть в первую очередь. Указывая на пять карточек, Паоло спрашивает: “Итак, это охватывает всю работу, насколько тебе известно?” На это Петра улыбается и отвечает: “Это регламент. Он никогда не завершается и не проясняется”.

²⁷ Помимо главного ответственного за выпуск продукции, выполняющего роль Владельца Продукта, во многих крупных группах имеются свои вспомогательные ответственные за выпуск продукта, каждый из которых специализируется на отдельном сегменте рынка или клиентской области.

Паоло просит: “Можешь разместить все это в неупорядоченном пока что виде на дне Беклога Продукта?”
“Конечно”.

См. раздел
“Руководство:
Помощники
Владельца Продукта”
главы 8.

См. раздел
“Руководство:
инструмен-
тальные средства
для ведения крупных
Беклогов
Продуктов” главы 9.
Совет: пользуйтесь
электронными табли-
цами и wiki для
крупного Беклога
Продукта.

Через неделю Паоло обращается к Петре: “Вскоре я хочу начать выпускать некоторые части большого регуляторного требования для деривативов. На следующих сессиях по Проработке Беклога я намерен попросить некоторые команды сосредоточиться на этом. Никто, кроме тебя, этого лучше не знает, поэтому будь, пожалуйста, на всех подобных совещаниях и всех командных встречах, куда команды тебя могут позвать. Кроме того, сможешь организовать wiki-страницу со ссылками на новые нормативные документы, чтобы сделать их общедоступными для всех команд?”

“Уже готово”, — отвечает Петра.

Общий PBR

См. раздел
“Руководства: виды
виды сессий
Проработки Беклога
Продукта” главы 11.

Паоло начинает краткое совещание по общей Проработке Беклога Продукта: “У нас много работы по новыми регламентам. Вскоре нам будет пора начать выпускать связанный с ним функционал из-за конца финан-

сового года. После разбиения и оценивания мы, конечно, поймем лучше, но меня не удивит, если в конечном счете реализация потребует более трех команд и кучу времени”.

См. раздел
“Руководство:
разбитие
элементов Беклога
Продукта” в главе
11.

Группа разбивает новый гигантский элемент лишь на несколько крупных частей, чтобы изучить главные элементы. Дополнительное разбиение произойдет позднее при одно- или многокомандной Проработке Беклога а Продукта. Петра на-

правляется к доске. Слева она пишет “регламенты для производных облигаций”. Затем при последующем обсуждении с группой на доске набрасываются три древовидные схемы с четырьмя ответвлениями, обозначающими разделение на четыре главных под-элемента. Но они не углубляются дальше, избегая чрезмерного анализа.

См. раздел
“Руководство:
оценивание
сложности
в крупных масштабах”
главы 11.

Затем группа создает четыре карточки для новых элементов, и все вместе они оценивают элементы в ходе покера планирования, присваивая им относительные единицы размера, исходя из уже имеющихся, хорошо известных элементов в Беклоге Продукта. Их главная цель — не произвести оценки, а выявить вопросы и спровоцировать дополнительные дискуссии, которые они проводят вместе с Петрой.

Далее Паоло спрашивает: “Итак, Петра, какой из этих четырех крупных элементов будет первым?”

Указывая на вторую карточку, она отвечает: “Экзотические внебиржевые производные облигации”.

Паоло говорит: “Нам нужно приступить к поставке чего-то из этого как можно скорее. Поэтому он перемещается к вершине Беклога Продукта. И мне бы хотелось, чтобы одна из команд в следующем Спринте. Кому интересно?”

На это вызывается Команда “Торговля”. В конце концов, члены трех других команд решают провести многокомандный PBR для рассмотрения связанных элементов.

Командный PBR: проба на вкус

На следующий день команда “Торговля” проводит вместе с Петрой совещание по Проработке Беклога Продукта. Они сосредоточивают свое внимание только на одном из четырех гигантских элементов, а именно: на новых регламентах по экзотическим внебиржевым производным облигациям. На совещании присутствует также Сэм, их Scrum-мастер. Петра говорит: “Это довольно крупный и сложный элемент, относящийся к области, которую, честно говоря, мало кто понимает. Такт что будьте готовы: нам придется потратить немало времени, чтобы разбить его на более мелкие части, по-настоящему уяснить и как следует детализировать”.

Сэм спрашивает: “Должны ли мы действительно понимать это *все*? И научит ли весь этот анализ нас чему-то большему, чем мы знаем сейчас, или же наоборот - *отложит* наше обучение?”

Он обсуждает с ними идею *Надкусывания*: отделить от анализируемого элемента один мелкий фрагмент, по-настоящему разобраться в нем и быстро реализовать. В итоге Сэм заключает: “Как известно, блок-схемы не крещатся, а документы не исполняются”.

Вместе с Петрой команда отделяет один мелкий кусочек от тонкого слоя законченного с точки зрения пользователя элемента.

С этого момента они фокусируются на этом крошечном куске, проясняя и реализуя его. И только после реализации и обратной связи, они вернутся к последующему разбиению и уточнению. Используя *спецификации через примеры*, Петра и Команда “Торговля” провели оставшуюся часть дня пережевыванием откушенного кусочка.

См. раздел
“Руководство:
Надкусывание” главы
9.

Совет: см. Спецификации через Примеры подразделе “Прояснение” из раздела “Руководство: Детализация” главы 11.

Многокомандный PBR с чередованием

Так, в результате общей проработки было решено о “надкусывании” в Команде “Торговля”. А также, что три остальные команды проведут сессию по многокомандной Проработке Беклога Продукта для рассмотрения связанных между собой элементов, а также чтобы повысить уровень понимания и гибкости - когда несколько команд смогут понимать и думать об одних и тех же элементах.

См. раздел
“Руководство: PBR с многими офисами”
главы 11.

Помимо всех членов трех команд, на это совещание пришли внутренние трейдеры облигациями Таня, Тед и Трейвис, чтобы помочь командам приступить к прояснению десятков новых элементов.

Сначала они образовали три временные смешанные группы из участников каждой команды. Смешанные группы начали с прояснения различных элементов в разных частях комнаты - каждая с доской,

большим свободным пространством стены, ноутбуком и проектором. С первой группой была Таня, со второй — Тед, а с третьей — Трейвис.

Затем они произвели *проработку с чередованием* (рис. 2.1): каждые 30 минут срабатывания таймера каждая группа переходит к следующей зоне - и так по кругу, но Таня, Тед и Трейвис остаются на своих местах. Таймер снова запущен, трейдеры проясняют текущие результаты проработки вновь пришедшей группе, после чего проработка продолжается..



Рис. 2.1. Многокомандная Проработки Беклога Продукта с чередованием

В течение дня, когда различные элементы становятся более-менее понятными - или остаются, обвешанными вопросами, к которым нужно будет вернуться позже, - для последующего рассмотрения добавляются все новые элементы. Некоторые из крупных элементов в результате этого процесса разделяются на два или три более мелких куска.

См. раздел
“Руководство:
оценивание
сложности
в крупных масштабах”
главы 11.

В течение дня группа несколько раз приостанавливает процесс прояснения и погружается в оценивание сложности, главным образом, для изучения и катализации дискуссий. С этой целью они пользуются относительными оценками сложности (также известными как "story points"). А для того чтобы придерживаться общей шкалы, они производят калибровку относительно ряда готовых и хорошо известных элементов из Беклога Продукта.

Обновление Беклога Продукта и информирование Владельца Продукта

На следующий день после совещаний по Проработке Беклога Продукта Петра и ряд членов команд:

- > Обновляют Беклога Продукта новыми элементами, полученными из первоначальных элементов путем разбиения, удаляя начальные;
- > Добавляют ссылки на новые wiki-страницы с деталями, полученными на PBR сессиях;
- > Регистрируют новые оценки сложности и помечают элементы, готовые к реализации.

См. раздел
“Руководство:
помощники
Владельца Продукта”
главы 8.

См. раздел
“Руководство:
обращение с
родительскими
элементами” главы
9.

Позже Петра и эти же члены команд встречаются с Паоло, чтобы вместе просмотреть изменения Беклога Продукта и ответить на его вопросы.

Конец

Вот главные выводы из рассмотренной истории:

- > Если имеется крупный элемент, необходимо его "надкусить", чтобы начать научиться, поставляя что-то небольшое, избежав при этом преждевременного и чрезмерного анализа.
- > Проводите многокомандные PBR сессии - это создает общее понимание между командами, что в свою очередь повышает организационную гибкость, расширяет понимание всего продукта и создает предпосылки для самоорганизуемой координации.
- > Стремитесь к фокусу внимания на всем продукте в целом, даже с большим числом команд.

Далее - следующий раздел смещает внимание на каркас LeSS Huge, который применяется в больших группах из множества команд.

Каркас LeSS Huge

Область Требований

С тысячью или даже только сотней человек на одном продукте, принцип “раз- деляй и властвуй” кажется просто неизбежным из-за сложности такого большого количества требований и человек. Традиционная крупномасштабная разработка разделяется так:

- > Однофункциональные группы (например, группы анализа, тестирования и т.д.).
- > Архитектурно-компонентные группы (например, группы уровня пользовательского интерфейса, серверная группа, группа компонентов доступа к данным и т.д.).

Такая организационная структура приводит к медленной негибкой разработке с высоким уровнем потерь (склады, незавершенная работа, передача работы, разброс информации и пр.), задерживающийся период возврата инвестиций, усложненным планированием и координацией, издержками на дополнительный менеджмент, слабой обратной связью и обучением. Такая структура организована как бы *внутрь* - вокруг узких навыков, архитектуры и управления, а не *снаружи* - к потребительской ценности.

См. выше раздел
“Волшебное число
восемь”.

Но в каркасе **LeSS Huge**, когда есть восемь и более команд, разделение проходит по главным областям потребительских интересов, называемых **Областями Требований**. Такое разделение отражает принятый в LeSS принцип *ориентированности на заказчика*.

Масштабы. Область Требований - *большая* область, обычно от четырех до восьми команд, но не с одной или двумя. Соответствующие тому причины поясняются далее, в разделе “Продукто-ориентированные Команды областей требований”.

Динамичность. Области Требований - *динамические*. Со временем важность области изменится, она разрастется или сократится с соответствующим приходом или уходом команд - скорее всего в другие или из других областей.

Пример. Так, в продукте “Ценные бумаги” (для торговли им), могут присутствовать следующие Области Требований, представляющие наибольший интерес с точки зрения потребителей:

- > Управление торговыми операциями (от ценообразования до регистрации сделки и завершения сделок).
- > Обслуживание активов (например, дробления пакета акций, начисление дивидендов).
- > Подключение новых рынков (например, в Нигерии).

Концептуально в *единый* Беклог Продукта вводится атрибут “Область Требований”, и каждый элемент ассоциируется с одной и только одной областью, как показано ниже.

Элемент	Область требований
B	Подключение новых рынков
C	Управление торговыми операциями
D	Обслуживание активов
F	Подключение новых рынков
...	...

Затем участники просто могут сфокусировать на одном *Беклоге Области Продукта* (Area Product Backlog) - по существу, представлении единого Беклога Продукта) - таком как, к примеру, "подключение новых рынков":

Элемент	Область требований
B	Подключение новых рынков
F	Подключение новых рынков

Общий Спринт. Действует ли каждая Область Требований отдельно в своем Спринте и откладывает таким образом интеграцию до определенной даты в отдаленном будущем? Нет.

В LeSS Huge интегрируйтесь постоянно
в одном общем Спринте.

На уровне продукта существует один Спринт, а не разные
Сprintы для каждой Области Требований. Он
завершается одним интегрированным цельным
продуктом, а все команды во всех Областях Требований
стремятся к непрерывно интегрироваться в пределах
всего продукта.

Владельцы областей продукта

В каркасе LeSS Huge вводится еще одна новая роль. В каждой Области Требований имеется свой *Владелец Области Продукта* (Area Product Owner), который специализируется на данной *Области*, уделяя основное внимание Беклогу Области Продукта.

У крупных продуктовых групп обычно имеется несколько поддерживающих продукт-менеджеров, специализирующихся на разных клиентских областях, и некоторые из них, вероятнее всего, выполняют роль Владельцев Областей Продукта в отдельных Областях. Иногда Владелец Продукта несет двойную службу, также выполняя обязанности Владельца Области Продукта для одной из областей, что более всего характерно для *менее* крупных групп, практикующих LeSS Huge²⁸.

²⁸ И опять забавная игра слов: "small less huge LeSS Huge groups". — Примеч. ред

Продукто-ориентированные Команды области продукта

Продукто-ориентированные Команды области (Area feature team) работают в одной Области Требований (например, в обслуживании активов) с одним Владельцем Области Продукта, который уделяет внимание элементам одного Беклога Области Продукта. С точки зрения команды *работа в области сродни работе в меньшем каркасе LeSS* - она взаимодействует со своим Владельцем Области Продукта как с обычным Владельцем Продукта, и так далее.

Члены команд со временем начинают хорошо разбираться в данной предметную область клиентов. И к счастью, элементы одной Области Требований обычно охватывают наполовину предсказуемое подмножество всего кода продукта, благодаря чему сокращая объем того, что требуется знать в пределах обширного продукта.

Важный момент касательно масштаба: в отдельной Области Требований работает много продукто-ориентированных команд.

Как правило, в Области Требований работает от четырех до восьми команд. Это означает, что Область Требований должна быть крупной.

Волшебное число четыре

Прежде всего, возникает вопрос: почему в качестве верхнего предела для Области Требований предполагается *восемь* команд? Ответ на этот вопрос был дан ранее, в разделе “Волшебное число восемь”.

А что же нижний предел из *четырех* команд? Почему не одна или две? Естественно, что число *четыре* не является волшебным, но оно стремится соблюсти баланс таким образом, чтобы группа продукта не состояла из множества крохотных Областей Требований.

В чем же недостаток большого числа мелких областей? Они уменьшают обзор приоритетов с уровня всего продукта, увеличивают локальные оптимизации, усложняют координацию, требуют расширения штата и создания команд со слишком узкой специализацией и недостаточной гибкостью для переключения и работы с высокоприоритетными с точки зрения компании элементами. Кроме того, Владелец Области Продукта мелкой области, начнет себя вести как бизнес аналитик между пользователями и одной-двумя командами.

Имеются ли какие-нибудь благоразумные *исключения* из правила иметь не менее четырех? Да:

- > Ранний переходной период, когда группа постепенно дорастает до новой области, в которой вполне ожидается в конечном итоге наличие четырех или более команд. Тогда начинайте с малого и простого - с одной команды.
- > Перераспределение нагрузки при переводе команд из области с понижаемыми нуждами в область с повышающим, в результате чего область сокращается с четырех до трех команд. И в конечном счете две сокращенные мелкие области сливаются в новую крупную область.

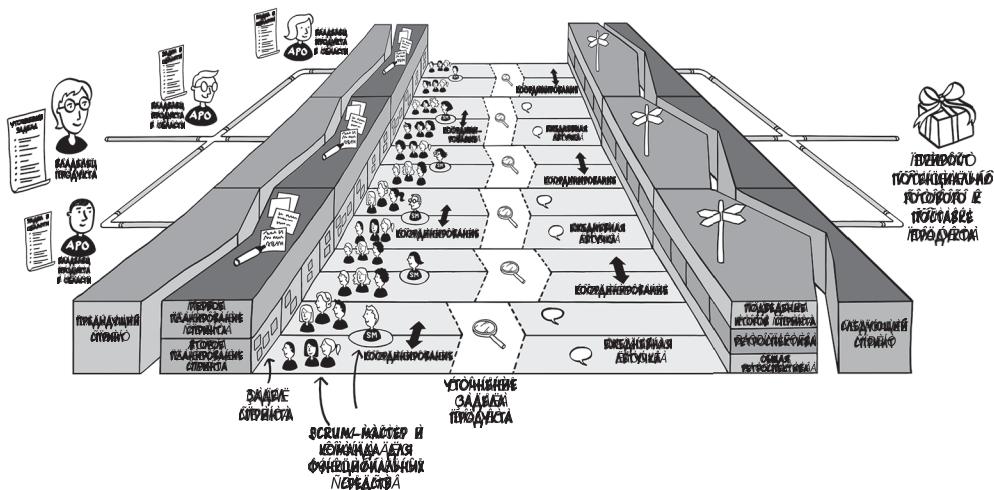
Пример Областей Требований и Команд

В целом для создания продукта “Ценные бумаги” может потребоваться:

- > Один Владелец Продукта и три Владельца Области Продукта, образующие Команду Владельца Продукта.

- > Шесть producto-ориентированных команд для области управления торговыми операциями.
 - > Четыре producto-ориентированных команды для области подключения новых рынков.
 - > Четыре producto-ориентированных команды для области обслуживания активов.

Краткое изложение каркаса LeSS Huge



Каждая Область Требований функционирует на основе (меньшего) каркаса LeSS, действующих параллельно в одном общем Спринте. Иногда мы объясняет Спринт в LeSS Huge как склад из меньших LeSS, один на одном.

С точки зрения команды внутри одной области, с точки зрения мероприятий и активностей LeSS Huge выглядит как (меньший) LeSS карках.

Так же как в и в меньшем LeSS каркасе, здесь (в LeSS Huge) есть **правила** и необязательные к использованию **руководства**; они представлены нижеприведенным историям и также раскрываются в больших деталях в дальнейших главах книги.

Роли. Те же, что и в LeSS, плюс два или более Владельца Области Продукта, и от четырех до восьми Команд в каждой Области Требований. Единственный Владелец Продукта (уделяющий основное внимание

оптимизации всего продукта в целом) и несколько Владельцев Областей Продукта в образуют **Команду Владельца Продукта**.

Артефакты. Помимо тех же самых артефактов, что и в LeSS, у каждого элемента в общем Беклоге Продукта имеется атрибут принадлежности к *Области Требований*, а следовательно, каждая область имеет представление в виде **Беклога Области Продукта**.

События. Для продукта по-прежнему имеется лишь один общий Спринт. В нем участвуют все команды, производящие в конечном итоге общий потенциально готовых к поставке инкремент продукта.

Истории LeSS Huge

Изучение LeSS Huge. Те читатели, которые предпочитают сразу же окунуться в детальные толкования могут смело перейти к материалу последующих глав, пропустив эти истории.

Простые истории. Эти истории намеренно упрощены с целью просто представить базовые принципы LeSS Huge.

Две темы. Ниже приведены две истории с различной тематикой:

1. Создание и наращивание новой Области Требований для удовлетворения новых громадных требований.
2. Работа с командами из разных офисов (multi-site teams). (Такая работа тоже ведется и в меньшем каркасе LeSS, но она особенно характерна для LeSS Huge.)

История LeSS Huge: новая Область Требований

См. раздел
“Руководства:
Владелец Продукта в
LeSS Huge” главы 8.

Прити приветствует Петру²⁹ в первый день на ее новой работе. В качестве руководителя производства среднего звена в отделении Ценных Бумаг крупной торговой компании, а также Владельца Продукта внутренней системы “Ценные бумаги” Прити отвечает также за поиск и удержание кадров для Команды Владельца Продукта, состоящей из Владельцев Областей Продукта. Она уверена, что Петра - настоящая находка, так как ее опыт - это то, что нужно для работы с некоторыми новыми огромными требованиями.

²⁹ Напоминаем, что имена персонажей используют так называемую аллитерацию, чтобы было легче вспомнить их роли. Так Прити - Владелица Продукта, Петра - Владелица Области Продукта, Сьюзен - Scrum-мастер, Кирилл - член Команды.

Во время последнего собеседования, когда Петра еще работала продукт менеджером и специализировалась на вопросах нормативных документов в компании, выпустившей систему для торговли облигациями, Прити изложила возникшую ситуацию: “Петра, после последнего обвала рынка регулирующие органы стали жестко требовать от нас соблюдение акта Додда–Франка³⁰. Пока что мы не знаем, что это конкретно означает и как это повлияет на нашу систему. У нас есть необычайные знания в данной области и отличная профессиональная сеть связей с регулирующими органами. Мне бы хотелось, чтобы ты вошла в нашу группу и помогла нам справиться с данной ситуацией”.

³⁰ Законодательный акт США, принятый 21 июля 2010 г. в целях снижения рисков американской финансовой системы. — *Примеч. пер*

Большая неожиданность

Несколько дней спустя... Прити приглашает Петру, Питера и Сьюзен в свой кабинет. Питер является Владельцем Области Продукта по подключению новых рынков, а Сьюзен — Scrum-мастером в области управления торговыми операциями.

Прити говорит: "Как вы знаете, вступает в силу довольно солидный закон Додда-Франка. Но вы еще не знаете, что этим утром регулирующие органы обратились к нам с требованием принять меры *безотлагательно*. Я-то надеялась, что мы сможем приступить к работе со следующего года. Но нам придется адаптироваться, это очень серьезно."

"Не думаю, что кто-нибудь хорошо разбирается в этом законе по всех деталях — даже регулирующие органы. И мы не знаем, как это повлияет на нашу систему и сколько работы еще потребуется, хотя потрудиться придется немало! Но теперь к нам присоединилась Петра, и она лучше всех разбирается в этом деле, хотя наша система для нее совершенно нова. Так чем мы можем помочь ей, чтобы свернуть эти горы работы?"

Сьюзен спрашивает: "Вы, коллеги, вероятно, знакомы с дислексическими зомби?"³¹ Питер и Прити согласно кивнули. Команда "Дислексических Зомби" имела, пожалуй, самый широкий опыт среди всех команд. Они были в компании годами и в свое время были настоящим препятствием для внедрения LeSS. В данной команде было два бывших члена теперь уже упраздненной архитектурной группы, и пара ребят, которые работали над системой уже более пятнадцати лет. Их сопротивление внедрению LeSS стало легендарным, поскольку они боялись потерять свое "представление о всей системе". Но к их удивлению произошло совершенно обратное! Благодаря своим глубоким знаниям им постоянно доставались самые трудные элементы и им регулярно приходилось участвовать в качестве экспертов-учителей в сессиях по ознакомлению с текущей архитектурой (current-architecture-learning workshops) для новичков компании. А Кирилл, один из бывших так называемых "PowerPoint-архитекторов", теперь ведет архитектурное сообщество³². Когда его нагрузить достаточным количеством пива, Кирилл признает, что более тесная работа с кодом и тестами *действительно* расширила его понимание системы.

Сьюзен продолжает: "Если и есть команда, способная быстро помочь Петре понять масштабы и влияние закона Додда-Франка, - это "Зомби". Несколько лет назад они занимались внедрением закона Сарбейнза-Оксли. Завтра они проводят свой PBR. Они только начали разбираться с новым функционалом. Почему бы нам не переориентировать совещание, включив в него обсуждение закона Додда-Франка,



³¹ Игра слов: в оригинале зомби с проблемами чтения ищут "BRIANS" (имя) вместо "BRAINS" (мозги). Авторы утверждают, что так и правда называлась команда в одной из компаний Лиссабона.
— Примеч. ред

³² Вы, наверное, замечаете отношение авторов к архитекторам? Они небезосновательно считают, что самое лучшее, что реальные архитекторы и эксперты могут делать - это быть "в окопах", передавая свои хорошие инженерные привычки и навыки (сообщества, парное и тоб-программирование, сессии по изучению текущей архитектуры и проч.), вместо рисования PowerPoint-презентаций о "правильной архитектуре". — Примеч. ред

Проработка с “Зомби”

На следующий день на совещании по проработке вместе с командой “Зомби” Петра объясняет ситуацию: “Вам, вероятно, приходилось слышать о законодательстве Додда–Франка. Но вся неожиданность в том, что регулирующие органы потребовали от нас принять неотложные меры, чтобы продемонстрировать к концу года существенное соответствие этому закону. В противном случае они ограничат нашу торговлю ценными бумагами”.

Члены команды “Зомби” заметно удивлены. До них, конечно, доходили слухи, но они не ожидали такой спешки!

Кирилл говорит: “Хорошо, Петра. Поясни нам вкратце, что это означает для нас. И чем этот закон отличается от Сарбейнза–Оксли?”



Петра берет маркер и начинает чертить набросок на доске. Через 45 минут она завершает свой краткий обзор, и члены команды “Зомби” выглядят несколько бескураженными.

“Они говорят “к концу года”? — восклицает Марио. — Даже если *вся* группа начнет работу завтра, она не успеет ее завершить. Это слишком огромный объем!”

Он берет ручку и делает грубый набросок их системы, обсуждая с остальными членами “Зомби” возможное влияние данного закона на систему.

Он говорит: “Петра, давай воспользуемся этим шансом, чтобы помочь тебе лучше понять систему. Задавай любые вопросы”.

Порция: “Одну секунду. Дай я запущу видеозапись, которая поможет мне вспомнить все, что здесь будет сказано”.

Карим, ветеран команды: “Нам бы лучше начать работу над этими изменениями с разработки какой-нибудь реальной части, так мы сможем узнать больше по ходу дела, иначе мы будем анализировать до бесконечности. Я уже видел эту печальную историю раньше”.

Их Scrum-мастер Сьюзен говорит: “Помнится, Том ДеМарко как-то сказал, что причина всякого неудачного проекта заключается в том, что он начинается слишком поздно”. Все смеются, а она продолжает: “Так что предлагаю *надкусить*”.

См. раздел
“Руководство:
надкусите” главы 9.

Создание новой Области Требований

На следующий день Петра, Прити и остальные члены Команды Владельца Продукта встречаются вместе. Порция вкратце излагает фронт работ, как она его теперь понимает.

Прити говорит: “Это даже больше, чем я ожидала, а нам нужно продемонстрировать регулирующим органам какой-то ощутимый прогресс в течение нескольких месяцев и существенный прогресс уже до окончания финансового года, т.е. через семь месяцев. Очевидно, им разрешено потребовать от нас большего, и у них есть все полномочия закрыть нас. Как вы знаете, в прошлом месяце генеральный директор дал ясно понять, что новые требования регулирующих органов получают наивысший приоритет над всеми остальными интересами. Как подсказывает мой опыт, нам лучше проявить добрую волю, гибкость и прозрачность, как можно раньше предоставив регулирующим органам что-нибудь существенное и оперативно реагируя на их отзывы. Вот, что мы будем делать”.

Прити продолжает: “Мне кажется, что для этой большой неожиданности нам потребуется новая область. И это, разумеется, повлияет на уже имеющиеся у нас высокоприоритетные цели, поскольку нам придется переместить некоторые команды. Давайте подготовимся к более основательному обсуждению влияния этих перемен на общие приоритеты через пару дней. А сейчас я бы хотела услышать от вас предложения относительно раскрутки новой области”.

См. раздел
“Руководство: новая
область для
громадного требова-
ния” главы 9.

После краткого обсуждения становится ясно, что все осознают важность создания новой области.

Далее Прити говорит: “Петра, я знаю, что ты у нас совсем недавно, но не могла бы ты взять на себя обязанности Владельца Продукта в данной Области?”

Петра кивает.

Прити продолжает: “Питер, как тебе кажется, могут “Зомби” приступить к работе над этим? И нам нужно, чтобы они ознакомились с Доддом–Франком, а также выяснили его влияние на нашу систему, прежде чем мы сможем ввести в данную область больше команд”.

См. раздел “Руководство:
ведущая команда” главы
13.

Питер: “Не думаю, что у нас есть какой-нибудь другой выбор”.

Прити: “Что ж, Петра, в настоящий момент у нас уже имеется несколько элементов в Беклоге Области Продукта, который ведет Питер. Один крупный элемент, который, я думаю, вы называли “остальная часть закона Додда–Франка”, и мелкий, который команда “Зомби” отделила от него. Попросите Питера показать вам, как организовать новую область в Беклоге Продукта и переместить в нее элементы”.

Прити продолжает, обращаясь к группе: “Следующий Спринт начинается через три дня. Давайте переместим команду “Зомби” в нашу часть офиса и приступим к этому монстру. Вероятно, через пару Спринтов мы будем - и должны быть - готовы расширить нашу область, переместив в нее еще одну команду. Подумайте, ребята, о следующих двух важных вещах. Во-первых, о подготовке к совещанию по поводу влияния на приоритеты через несколько дней. И, во-вторых, о тех командах, которые лучше всего подходят для новой области”.

Планирование Спринта в новой Области Требований

В каждой Области Требований более или менее параллельно проводятся совещания по Планированию Спринта. В новой для Петры области она начинает свое первое совещание по Планированию Спринта, вводя два незнакомых лица в команду “Зомби”.

Она говорит: “Джиллиан и Зак находятся в постоянном контакте с регулирующим органами и помогут нам конкретизировать стоящую перед нами задачу. Они согласились помочь нам на Планировании, в течение наших сессий Проработки Беклога и по мере возможности ежедневно в течение предстоящих Спринтов”.

Она продолжает: “Вот мой предварительный план атаки следующих двух Спринтов. Во-первых, вместе нам нужно узнать больше об этом законе Додда–Франка и разделить его на ряд основных и выполнимых кусков, чтобы рассеять туман неизвестности и лучше уяснить вопрос приоритетов.

И, во-вторых, мы реализуем небольшой откусенный ранее кусочек, начав в этом Спринте. Благодаря этому мы лучше изучим реальный объем работы и ее влияние на наш продукт. И мы продемонстрируем вполне конкретный видимый прогресс.

И, в третьих, мы подготовим площадку для большего числа команд для присоединения к нашей области. Что вы думаете о таком плане? Ваши предложения?”

См. раздел
“Руководства:
ведущая команда”
главы 13.

В ходе краткого обсуждения Кирилл обращается к своей команде: “Позвольте дополнительно ввести вас в курс дела, поскольку я представлял нашу команду на последнем совещании Команды Владельца Продукта при участии Владельцев Про-

всех в Владельцев Областях и Притти. Прежде всего, для нас это всего лишь начало. Нам предстоит возглавить работу по ранней реализации, получив общее представление об этом элементе и уяснив его общее влияние на нашу архитектуру”. Кати прерывает его вопросом: “Как штурмовая группа, работающая над новым продуктом?”

“Да, что-то вроде этого, — отвечает Кирилл. — Рассматривайте поддержку закона Додда–Франка как новый продукт, который нужно непрерывно интегрировать в остальной продукт. Но нам нужно спешить, и у нас куча работы, поэтому через несколько Спринтов к нам присоединится одна команда, и чуть позже - еще, вероятно, две. Мы также продолжим разработку, но будем *ведущей командой*, а это означает, что нам придется вводить в курс дела другие команды, чтобы ускорить процесс, держа в уме продукт в целом”.

Кати восклицает: “Начинает звучать, как будто мне постепенно превращаемся в команду архитекторов и менеджеров проекта!”

Кирилл смеется: “Нет, с этим я завязал. Мы по-прежнему остаемся обычной продукто-ориентированной командой, но, помимо разработки, мы сосредоточимся на наставничестве, чтобы как можно скорее ввести новые команды в курс дела. Следует, однако, признать, что координация и управление по-прежнему остаются обязанностями каждой отдельной команды”.

Первый Спринт в новой Области Требований

Первый Спринт имеет необычный баланс между прояснением и разработкой, хотя весьма полезным в экстремальных случаях, как этом. Они провели почти половину времени Спринта, проясняя и детализируя вместе с Порцией, Джиллианом и Заком. Дело в том, что даже для такого очень мелкого куска продукта попытки понять, что же на самом деле необходимо - в условиях столь неясной области новых нормативных документов и в отсутствие прямого доступа к их составителям и законодателям - требует много исследований, чтения, дискуссий и коммуникации с внешними источниками.

Они ожидают, что в последующих Спринтах время, необходимое для прояснений, снизится до привычных 10–15% от времени Спринта.

И поэтому команда также тратит всего половину Спринта на разработку *одного* мелкого элемента. Тем не менее обсуждения и прямое изучение от кодирования окупаются с лихвой. Медленно, но верно они начинают разделять закон Додда–Франка на части — по крайней мере, на те, которые могут быть понятны любому из них.

Реализуя мелкий элемент, который был откушен первым, они проводят большую часть времени вместе за досками, обсуждая последствия дизайна решения на систему. Команда быстро переходит от кода к стене с набросками и обратно.

Ревью Спринта в новой Области Требований

Вся продуктовая группа “Ценные бумаги” совместно работает в одном Спринте над одним окончательным готовым к поставке инкрементом продукта. Но в каждой Области Требований более или менее параллельно проводится свое совещание по обзору спринта - Ревью Спринта.

В ходе подведения итогов в той области, которой заведует Петра - она, Джиллиан и Зак исследуют один “готовый” элемент, который команде “Зомби” удалось завершить и проинтегрировать в общий продукт. Первоначально команда прогнозировала, что завершит два элемента, но Петру удивляет даже то, что им удалось завершить хотя бы один элемент, если учесть, насколько в срочном порядке на них была брошена эта работа.

Второй Спринт

Во втором Спринте им удалось достичь чуть большего прогресса в реализации элементов, хотя они снова проводили немало времени на прояснение вместе с Петрой, Джиллианом и Заком.

См. раздел
“Руководство:
надкусывание” главы
9.

См. раздел
“Руководство:
обращение с
крупными тре-
бованиями” главы 9.

См. раздел “Руководства:
рабочая сессия по
текущей архитектуре”
главы 13.

В середине Спринта они проводят совещание по многокомандной Проработке Беклога Продукта вместе со второй командой, которую планируется ввести в данную область и которой теперь поясняются особенности закона Додда–Франка. А для ознакомления второй команды с уже готовыми элементами проводится сессия по изучению текущей архитектуры.

Команда “Зомби” уже знает, насколько велик объем работы, и поэтому стремится заручиться дополнительной поддержкой.

Совещание Команды Владельца Продукта

См. раздел
“Руководство:
совещание
Команды Владельца
Продукта” главы 12.

Несколько Спринтами позже наступает снова время для совещания Команды Владельца Продукта, которое проводится раз в спринт. На этом совещании согласовываются и координируются действия Владельцев Продукта в разных Областиах, и Петра их направляет.

Каждый из Владельцев Области Продукта в делится сложившимся положением дел и ближайшими целями. Когда наступает черед ее, Петра говорит: “Ни для кого из нас не секрет, что прогресс незначителен, а неожиданности велики. Но туман постепенно рассеивается, а я и команды постепенно начинаем понимать суть работы. Джиллиан и Зак оказали нам сильно помогли”.

Пабло, Владелец Продукта в Области управления активами, комментирует некоторые тесные взаимосвязи элементов, которые он теперь усматривает между областями. Петра соглашается чуть позже встретиться с Пабло и некоторыми представителями команд.

Прити спрашивает: “Порция, что на счет предстоящего Спринта? Каковы твои цели?”

Ввод третьей команды

Двумя Спринтами позже... На координационном совещании Команды Владельца Продукта Прити говорит: “Как вы знаете, в области, которую ведет Петра, по-прежнему работают всего две команды. Я знаю, что Паоло хотелось бы сохранить свои шесть команд в области управления активами, но закон Додда–Франка крайне важен для меня в этом году. Поэтому мы собираемся переместить одну команду из области Паоло в область Петры. Паоло, выясни, пожалуйста, какая из команд добровольно согласится на это, и дай знать мне и Петре”.

Конец

Ниже перечислены главные выводы из рассмотренной выше истории LeSS Huge.

- > Владелец Продукта отвечает за поиск Владельцев Области Продукта в и развитие их способностей.
- > Владелец Продукта отвечает за решение начать, расширять или сворачивать Области Требований.
- > Области Требований велики и обычно требуют от четырех до восьми команд, но на начальной стадии они могут быть меньше, особенно если они инициируются одной командой подходом ‘Надкусывания’.

- > Ведущая команда работает отдельно над очень крупным элементом до тех пор, пока ее члены не уяснят предметную область и особенности разработки. После этого они помогают дополнительным командам поскорее подключиться, чтобы помочь выполнении обширных работ.

Много-офисные команды: термины и рекомендации

Далее следует история LeSS Huge с командами из разных офисов (multi-site teams). Но сначала необходимо дать ряд проясняющих определений, поскольку общепринятый термин *распределенные команды* запутывает, имея множество значений. Ниже перечислены термины, проясняющие ситуацию:

- > **Рассредоточенная команда (dispersed team).** Это одна команда, состоящая, например, из семи человек, рассредоточенных по разным местам, будь то помещения, здания или города.
- > **Совмещенная команда (co-located team).** Это одна команда, работающая буквально за одним столом.
- > **Много-офисные команды (multi-site teams).** Одна совмещенная команда работает одном месте, тогда как другая совмещенная команда работает из другого .

Далее приводятся некоторые наблюдения и рекомендации.

- > Рассредоточенная команда редко оказывается *настоящей* командой. Она скорее похожа на слабо связанную группу отдельных личностей. Трения в общении и при координации в такой команде сильны, и поэтому из нее редко получается *спаянный коллектив*.
- > Даже когда группа продукта состоит из 50 или 500 человек, *рассредоточенные команды не обязательны*. Каждую команду, состоящую из плюс-минус семи человек, нетрудно сделать совмещенной. Однако, разные команды могут находиться в разных местах, и поэтому в группе продукта имеются *много-офисные* команды. Наличие рассредоточенных команд - зачастую следствие неудачных организационных решений и неведении о цене, которую приходится платить за отсутствие полноценных совмещенных команд.

Правило: каждая команда, во-первых, самоуправляемая, во-вторых, межфункциональная, в-третьих, совмещенная и, в-четвертых, долговечная.

История LeSS Huge: многоместные команды

Петра является Владельцем Продукта в новой Области Требований к системе торговли ценными бумагами. Эта новая область была начата лишь одной командой ради фокуса и простоты. А через несколько Спринтов в ее область уже была введена третья команда. Две первые команды располагаются вместе с Петрой в Лондоне, а третья команда “Дом Дракулешть” — в румынском городе Клуж, где находится основное место для разработки в компании.

Почему было не добавить третью команду в Лондонском офисе? Это позволило бы избежать многих неприятных ситуаций между командами и потерю эффективности, которые неминуемые при много-офисной разработке внутри единой Области Требований - потеря настолько высоких, что порой добавление команды имеет такой же эффект, как *удаление команды*.

Но с другой стороны, в качестве положительных моментов в данном случае следует отметить, что разница между Клуж и Лондоном - всего два часовых пояса, и все румынские работники в говорят по-английски. Они все - сильные разработчики с высшим образованием в области компьютерных наук из города, где давно ценится инженерное дело и мастерство разработки. Кроме того, данный офис - специальный выделенный центр разработки компании, и поэтому здешние команды имеют глубокие знания продукт и предметной области.

Таким образом, Прити как Владелица Продукта не захотела сдвигать никакие лондонские команды из их областей.

Прити знает, что много-офисные команды вновь для Петры, и поэтому на их следующем совещании она говорит: “Попросите Scrum-мастера поговорить с Ситой, а Ситу — покоучить проведение некоторых ваших мероприятий. Она Scrum-мастер в области управления активами и вот уже несколько лет наблюдает сложившуюся много-офисную ситуацию. Она хорошо знает, насколько важно, чтобы Scrum-мастера были рядом со своими командами, и она помогала проведению многих много-офисных совещаний”.

Прити продолжает: “Кроме того, у нас выдался сверхприбыльный год, и поэтому я выделяю дополнительное финансирование вам и команде “Зомби” — по крайне мере, тем, кто может поехать - провести Спринт в Клуже и как можно скорее. Работайте с ними непосредственно — все в одном помещении. Команда может, конечно, приехать из Клужа сюда, в Лондон, но мы должны дать им ясно понять, что они нужны, и на своем месте. Постарайтесь избежать почувствовать, что Лондон важнее, чем Клуж. О, и кстати, вам придется посещать их регулярно - каждые несколько месяцев”.

Много-офисное Планирование Спринта, Часть Один

См. раздел
“Руководство: Пла-
нирование Спринта,
Часть Один” главы
12.

Несколько Спринтов спустя Петра входит в комнату, Там установлен проектор, присоединенный к ноутбуку, на которым видна комната в Клуже. Вся команда в Клуже собралась в ожидании. Сита предложила для повышения обучения и вовлечения всей команде (а не только представителям) из Клужа принимать участие в много-офисных в течение первых нескольких месяцев после присоединения к области.

Представители команд пришли на совещание со своими планшетами или лаптопами.

Петра начинает: “Приветствую всех, и давайте начинать. Мои предложения по реализации элементов на этот Спринт выделены в доступной всем электронной

таблице. Всем ли видно? Думаю, все понимают, почему здесь приведены именно эти темы и приоритеты, поскольку мы уже обсуждали их при Проработке Беклога Продукта, и таблица также отображает ваши и мои пожелания. Но, пожалуйста, задавайте вопросы, если что-то требует прояснения. Если же вопросов нет, приглашаю всех вводить названия ваших команд рядом с наименованиями элементов, которые вы желаете взять в реализацию”.

Покончив с этим, группа переходит к сессии вопросов и ответов, связанных с реализуемыми элементами. Представители команд в Лондоне прикрепляют к стене большие листы флипчарта и приступают к составлению вопросов. Члены эже команды в Клуже вводят свои вопросы на отдельных закладках общей электронной таблицы. Петра

уделяет сначала некоторое флипчартам, обсуждая вопросы и делая наброски маркером, а затем к электронной таблице, вводя ответы для команды в Клуже, одновременно общаясь с ними в режиме видеоконференции.

После получаса отдельные вопросы разрешены, после чего Петра просит всех собраться снова вместе. Она говорит: “Есть ли еще какие-нибудь спорные или не-разрешенные вопросы, которые требуется разрешить совместно, прежде чем мы разбежимся?”

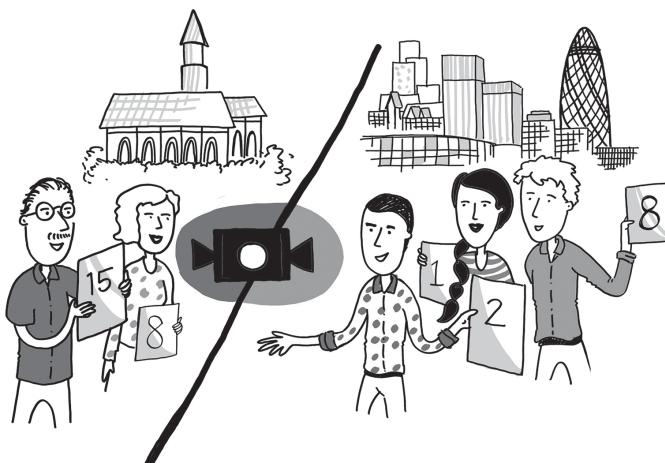
Общая многокомандная Проработка Беклога Продукта из разных офисов

Приглашенные входят в комнату совещаний в Лондоне, где настроены два проектора. Один из них служит для показа видео из Клужа, а другой отображает браузер с компьютера Петры.

Она говорит: “Итак, начнем. Я хотела бы уделить внимание разбиению некоторых элементов. С этой целью я пригласила к нам Зака, поскольку он хорошо разбирается в данной теме”. Пользуясь графическим средством для майнд-меппинга прямо в браузере, Зак начинает создавать ряд связей, обсуждая их вместе с группой.

См. раздел
“Руководство: виды
Проработки Беклога
Продукта” главы 11.

См. раздел
“Руководство: много-
офисная Проработка
Беклога Продукта ”
главы 11.



После этого, пользуясь общей электронной таблицей, они обсуждают и записывают по одному примеру по каждый из новых разделенных элементов, чтобы присутствующие в обоих офисах получили легковесные но очень конкретные детали требований. Позднее, группа приступает к оцениванию сложности новых элементов, используя специальные большие карты для покера планирования, которые хорошо видны в камеру и на видеоизображении во время голосований.

Конец

Ниже перечислены главные выводы из рассмотренной выше истории из LeSS Huge.

- > Много-офисные команды нередко создают как очевидные, так порой и не незаметные ситуации, приводящие к затратам, которые оказывают, на удивление, существенное отрицательное влияние.
- > Качества, которые снижают негативные ситуации с командами из удаленных офисов, - это одинаковый или близкий часовой пояс; внутренние выделенные центра разработки (вместо аутсорсинга); разработчики, свободно владеющие общим языком; место и культура, которые высоко ценят долгосрочное прикладное совершенство разработчиков.
- > Scrum-мастер должен находиться в одном офисе со своими командами.
- > Каждая сторона должна считаться равноправной, а не незваным гостем из третьего мира.
- > Офисы должны регулярно посещаться и перекрестно опыляться.
- > На совещаниях старайтесь применять инструментальные средства для непосредственного общения в режиме видеоконференций.
- > Использование документов в общем доступе упрощает одновременную работу с артефактами для всех сторон.

Идем дальше

Вместо того чтобы спрашивать "Как внедрить гибкую разработку в крупных масштабах в нашей сложной и непростой организации?", лучше задавать такой вопрос: "Как нам упростить организацию и *стать гибкими*, вместо того, просто *работать гибко*?" И поскольку настоящее масштабирование Scrum начинается с изменений организации, а не адаптаций самого каркаса Scrum, следующая большая часть книги уделяет внимание фокусировке на понимании и внедрении более простых LeSS организаций, ориентированных на заказчиков.

Затем следуют главы, посвященные более ориентированным на заказчика *продукту* и *Спринту* в упрощенной LeSS организации.

Структура LeSS



Руководители, обдумывающие улучшения, в качестве помощи

3

Внедрение

*Если ты не меняешь направление,
то можешь оказаться там, куда направляешься.
— Лао Цзы*

Однокомандный Scrum

Scrum прост. Внедрение Scrum — нет. Почему так?

Scrum - это не процесс. Он не решает все ваши проблемы как по волшебству и не создает "гиперпродуктивные" команды по взмаху волшебной палочки³³. Это каркас, который встраивает короткие петли обратной связи, значительно увеличивающие прозрачность. Он действует подобно зеркалу, показывающему, на сколько команда хороша в создании продукта. Он также обнажает проблемы, возникающие в команде и во всей организации. Такая видимость опирается на эмпирическое управление процессом, которое наряду с циклами испекции и адаптации направляет команду, Владельца Продукта и всю организацию к непрерывному циклу совершенствования.

Это хорошие новости. Плохие же новости в том, что видеть все это сложно. В действительности же прозрачность неудобна и даже пугает, что затрудняет внедрение Scrum.

Однокомандный каркас Scrum мало что говорит о внедрении Scrum, кроме необходимости начинать "по всем правилам". И это не потому, что приверженцы Scrum хотят навязать свои правила всему миру, но из-за признания того факта, что совершенствование начинается с соблюдения и уяснения норм и стандартов. Или если это сказать терминами бережливого производства: "Там, где нет норм, не может зародиться кайзен³⁴". Опыт работы по Scrum по

³³ Отсылка к риторике Джейффа Сазерленда (Jeff Sutherland, со-автор Scrum), который часто использует оборот "гиперпродуктивные команды". — Примеч. ред

³⁴ Kaizen - один из столпов бережливого мышления. Дословно с японского переводится как "улучшения к лучшему". Этот термин - синоним культуре постоянных улучшений на всех уровнях. — Примеч. ред

всем правилам дает понимание взаимосвязи принципов и практик - взгляд системного мышления. И это имеет решающее значение для успешного внедрения Scrum.

Опытный Scrum-мастер и команда с глубоким пониманием Scrum существенно повышают вероятность успешного внедрения.

Внедрение LeSS

LeSS внедряется в крупных организациях, в которых обычно сосредоточено много умов с глубоко укорененными представлениями о том, как *должны* работать организации, и как нет. Для успешного внедрения LeSS требуются ломка подобных представлений и *упрощение организационной структуры*, которая как правило включает много избыточных унаследованных правила, вытеснивших нормальные человеческие отношения - все то, что встречается в работе больших групп. Внедрение LeSS требует, чтобы каждый начал совершенствоваться, стремясь к общей цели.

При масштабировании следует придерживаться такого принципа внедрения: **непрерывное улучшение до полного совершенства**.

Естественно, что группа, внедряющая LeSS, приносит весь свой багаж представлений и привычек ко внедрению изменений. Но каковы они и подходят ли они для внедрения LeSS? Они чаще всего состоят в том, чтобы создать громкую концепцию³⁵ инициативы изменений и запустить много проектов по их внедрению. И потом как только начальная цель кажется "достигнутой":

1. изменения помечаются как "внедренные" и
2. организация переходит к новому порядку вещей до тех пор, пока
3. не появляется новая инициатива по изменениям, и тогда
4. предыдущие изменения отменяются.

Этот классический подход подобен последовательному подходу к разработке программного обеспечения "большими партиями", когда изменение является *исключением*, строго регулируемым *многими* контролирующими органами...

При внедрении LeSS отсутствует как инициатива изменения, так и группа и менеджеры по реализации изменений. Изменения в LeSS происходят *постоянно* посредством экспериментирования и совершенствования, и *сами изменения являются новым порядком вещей*.

Правила LeSS

Для продуктовой группы с самого начала создается полная структура LeSS. Это очень важно для внедрения LeSS.

В более крупной организации, выходящей за пределы одной группы, внедрение LeSS происходит эволюционным путем, используя принципу "Go and See" для создания организации, где экспериментирование и совершенствование являются нормой.

³⁵ Термины "концепция" и "видение" (англ. "vision") мы используем в книге пополам как синонимы. — Примеч. ред

Руководство: три принципа внедрения

Приведенные ниже принципы имеют решающее значение для организованного внедрения LeSS.

- > Лучше глубоко и узко, чем широко и мелко.
- > Сверху вниз *и* снизу вверх.
- > Добровольное участие.

Лучше глубоко и узко, чем широко и мелко

Предпочтение следует отдавать внедрению LeSS в одной группе продукта³⁶ но очень качественно, чем некачественно во многих группах.

Неудачное внедрение LeSS наносит ущерб. Недостаток глубокого понимания нарушает прозрачность и цепочки обратной связи, которые являются главными составляющими эмпирического управления процессом и непрерывного совершенствования. Нам доводилось видеть, как LeSS был ошибочно использован в качестве инструмента микро-менеджмента. Когда LeSS в стиле микро-менеджмента стал нормой поменять что-то уже *крайне* сложно. Очень трудно переобучить тому, что уже известно.

Таким образом, внедрение LeSS следует сосредоточить на одной группе продукта, оказать ей всю необходимую поддержку и убедиться, что она функционирует по-настоящему хорошо. Благодаря этому снижается риск, а при возникновении серьезных затруднений появляется возможность для *сосредоточенного обучения*. А в случае успеха создается реальный положительный пример, который очень важен для при дальнейшем внедрении в других частях организации.

Сверху вниз *и* снизу вверх

³⁶ В случае LeSS Huge, речь идет об одной Области Требований.

Нас часто спрашивают, как лучше внедрять LeSS: сверху вниз или снизу вверх. Это ложная дилемма. При выборе только одного вас ждет вероятная неудача. Делайте и так и так.

Подробнее об управлении в LeSS см. в главе 5.

Внедрение только сверху вниз

Внедрение LeSS по приказу начальства "делай, как я говорю" вызывает невольное сопротивление и приводит организацию к неудаче. Приказывать командам "перейти на самоуправление" - это противоречие. Для внедрения LeSS требуется глубокое понимание, которого можно достичь в ходе обсуждения, а не директивами. И только понимание, возможность выбора и чувство личной безопасности помогут людям взять на себя дополнительную обязанность размышлять и улучшать. Отсутствие этих предпосылок обостряется отношениями "мы-они" между руководящими и подчиненными работниками. В такой обстановке принудительное внедрение LeSS в организации приводит к поведению "преследуемых жертв" и дальнейшему ухудшению отношений - люди начнут жаловаться: "У нас нет выбора, начальство заставляет нас работать по LeSS!". И тайно, а возможно, и неосознанно они останутся в этом удобном - или как минимум привычном - для себя положении жертв.

Внедрение только снизу вверх

Такие внедрения LeSS нежизнеспособны. Поначалу у людей, стремящихся делать все правильно, возникает порыв энтузиазма. Это приводит к непредубежденности, ускорению обучения и более глубокому пониманию, что, конечно, замечательно! Но затем эти заряженные энергией люди энергично натыкаются на неприступные организационные стены. Бам! Без поддержки сверху энтузиасты, стремящиеся изменить структуры и правила, утрачивают свой пыл и теряются, встретив препятствия и непоколебимость. В конечном итоге многие уходят или испытывают горечь от разбившихся надежд. Нас это печалит не меньше..

Внедрение сверху вниз и снизу вверх

Для успешного внедрения LeSS требуется энергия людей, стремящихся делать все правильно, а также поддержка тех, кто обладает организационной властью. Руководители должны быть настроены на поддержку, а не на контроль. Они должны обеспечить надлежащую поддерживающую структуру, чтобы первоначальный энтузиазм подчиненных мог разрастись и развиться.

Нам часто приходится слышать желание поддержки со стороны руководителей. Поэтому будьте осмотрительны со своими желаниями, принимая во внимание следующее.

- > Отсутствие поддержки со стороны руководства нередко приводит к поведению преследуемых жертв. "Мы ничего не можем сделать без поддержки начальства".
- > Наличие поддержки со стороны руководства может привести к еще худшей ситуации. "Мы должны работать по методике LeSS, поскольку нам так говорит наше начальство". Такое бездумное повиновение подрывает любое внедрение LeSS.

Какого рода поддержка со стороны руководства требуется?

Поддержка со стороны руководства, обладающего организационной властью вносить структурные изменения в группу (как правило, это руководитель группы продукта). Такая поддержка должна ... поддерживать.

Подлинная поддержка начинается с самообразования. Все руководители из продуктовой группы должны уделить время изучению LeSS, в которое входит вводный курс в течение *нескольких дней* и чтение обязательной литературы. Помимо образования, руководители должны дать ясно понять и предпринять следующие действия:

1. Выразить твердое намерение внедрить LeSS.
2. Пообещать внести необходимые структурные изменения.
3. Организовать обучение и коучинг.

Какого рода поддержка со стороны руководства не требуется?

Поддержка со стороны высшего руководства, контролирующих многие продуктовые группы, помимо вашей одной группы, нередко приводит к обратному результату. Как же так? Неосведомленность в насущных вопросах - они недостаточно вовлечены в реальную разработку.

Поэтому их поддержка обычно включает в себя решения, принимаемые в целях “оптимизации” и “гармонизации” и кажущиеся здравыми с их вышестоящей точки зрения, но редко приводящие к реальным выгодам на *гембе* —месте, где, собственно, происходит настоящая ценная работа. И что происходит тогда? Расхлебывание последствий подобных вредных решений, принимаемых с благими намерениями, отнимает энергию, которую следовало бы направить на разрешение настоящих проблем.

Также вам не нужна поддержка и со стороны тех руководителей, которые еще недостаточно глубоко понимают LeSS и его влияние. Нас нередко просят свести трехдневный углубленный учебный курс к одночасовой презентации, поскольку руководители “слишком заняты” и не могут найти время, чтобы пройти трехдневный курс. И до сих пор нам не удавалось сократить свой трехдневный курс до одночасовой презентации, которую можно было бы понять. Возможно, в этом наша беда.

Добровольное участие

Как же сформировать новые команды? И кто будет участвовать в сообществе?
Как найти ответы на эти и многие другие вопросы внедрения LeSS?



Привлекайте добровольцев! Подлинное добровольное участие — это эффективный способ привлечь умы и сердца людей. И он применяется недостаточно, вероятно, потому, что руководители чувствуют, что они могут потерять контроль. Но для добровольных команд это ощущается, как *наделение властью*.

Добровольное участие начинается с образования. Допустим, вы предлагаете добровольцам поэкспериментировать со *смешанными парами* (*promiscuous pairing*). Охотников вряд ли найдется много, а те, кто отзовутся, в лучшем случае — просто запутаны. Но если вы сначала объясните, что *смешанные пары* — это техника парного программирования, предполагающая частые смены в паре для *повышения уровня обучения*, то обнаружите больше добровольцев и достигнете лучших результатов. Поэтому постарайтесь сначала предоставьте достаточно образования и обсуждения, чтобы люди поняли, на что они добровольно соглашаются.

Ниже приведены некоторые примеры добровольного участия.

Первоначальное добровольное участие в группе продукта. Какой должна быть первая группа продукта для внедрения LeSS со всеми вытекающими изменениями в организационной структуре? Найдите добровольную группу, опросив старших руководителей R&D и продукт-менеджеров.

Первоначальное добровольное участие в командах. Допустим, первоначальная группа продукта, предназначенная для внедрения LeSS, уже составлена и насчитывает около 50 человек. Вне этой группы могут оказаться люди, которые очень заинтересованы войти в нее, а в самой группе — люди, желающие выйти из нее! Следовательно, прежде чем “перевернуть всю группу”, прибегните снова к добровольному участию, пригласив всех желающих в вашей организации (пояснив, что и зачем). А кроме того, предложите людям в самой группе добровольно покинуть ее. Таким образом, первоначально приглашенные люди будут готовы учиться и брать на себя ответственность. Из них, вероятнее всего, получатся успешные первоначальные команды, поскольку это уже не простое *поголовье*, а заинтересованные всем *сердцем* люди.

Добровольное участие в формировании команд. Как же сформировать команды в LeSS? Поддержите “самоформирующиеся команды” (self-designing teams). Это делается на фасилитируемой сессии, которую посещают все члены будущих продукто-ориентированных команд. Фасилитатор начинает с объяснения назначения продукта и цели данной сессии. А затем они все вместе определяют шаблон типичной команды с учетом всех заранее определенных ограничений. (Фасилитатору уже известен удачный шаблон команды, но лучше, если группа придет к нему самостоятельно.) Ниже приведен пример такого шаблона.

- > Каждая команда является совмещённой.
- > Каждая команда является кросс-функциональной для доведения работы до полной готовности.
- > Каждая команда хорошо разбирается в нескольких компонентах.
- > Каждая команда состоит приблизительно из семи человек.

Детали “кросс-функциональности” и “кросс-компонентности” обсуждаются при определении шаблона команды. Далее в течение короткого периода времени (например, около 15 минут) участники добровольно формируют новые команды, руководствуясь составленным шаблоном. А после этого они анализируют рождающиеся команды. Если они сформированы недостаточно хорошо, группа продолжает еще несколько раундов; обычно достаточно от двух до четырех³⁷.

³⁷ Подробнее о том, как сформировать команды в крупномасштабном Scrum, а также об истории самоформирующихся команд, см. на веб-странице <https://www.scrumalliance.org/community/articles/2013/2013-april/how-to-form-teams-in-large-scale-scrum-a-story-of>.

Наставление: начало внедрения

Три упомянутых выше принципа применяются на начальной стадии внедрения в одной группе продукта. Как же увеличить вероятность успеха?

- 0.** Обучите всех.
- 1.** Определите "продукт".
- 2.** Определите, что значит "готово" (критерии готовности).
- 3.** Сформируйте правильно структурированные команды.
- 4.** Постановите, что Владелец Продукта обеспечивает команды работой.
- 5.** Держите менеджеров проектов подальше от команд.

0. Обучить всех

В самых лучших внедрения LeSS, которые нам приходилось наблюдать, все принимали участие в обучении по Scrum и LeSS в течение нескольких дней. После этого следовал коучинг по командным, организационным и техническим вопросам.

Это шаг - не для того, чтобы продать больше курсов по программе "Certified LeSS Practitioner", хотя мы будем не против. Для этой цели годится любое отличное образование - самое главное, что без образования вам вряд ли удастся привлечь большое число добровольцев, применяя принцип *добровольного участия*.



Обучить "зачем".

Помимо обучения особенностям внедрения LeSS, еще важнее помочь всем понять, зачем это вообще нужно. Ведь нередко приходится наблюдать слепое соблюдение процессов без понимания их причин.

Отличный тренер и опытный коуч сделают акцент на причинах и существенно повлияют качество внедрения LeSS. Как их выбрать? Следуйте следующим советам:

- > **Отдавайте предпочтение практическому опыту.** Убедитесь, что ваш тренер/коуч имеет практический опыт работы как в самом LeSS (в качестве члена команды), так и за ее пределами (в качестве коуча). Избегайте провайдеров обучения, которым все равно, кто обучает, а также тренеров, имеющих лишь теоретические знания. Пользы от них немного.
- > **Оценивайте личность, а не компанию.** Ищите особую личность. Коучинг — дело личное. Найдите коуча и нацельтесь на долгосрочные отношения с ним. Избегайте крупных консультационных и обучающих компаний.
- > **Требуйте глубоких технических знаний и понимания.** LeSS требует *технического совершенства*. Технология, команда и организационные решения тесно связаны вместе, и поэтому ваш коуч должен иметь широкие и глубокие представления об этом. Избегайте людей с ограниченным техническим опытом или полным его отсутствием. К их числу зачастую относятся бывшие руководители проектов в стиле Института Проектного Менеджмента (PMI).
- > **Планируйте долгосрочное вовлечение.** Внедрение LeSS требует терпения и занимает время. Найдите коucha, готового *годами* присматривать за внедрением LeSS в вашей организации. Избегайте коучей типа "проезжал мимо - решил зайти". Они, как правило, приходят, комментируют, критикуют и уходят.
- > **Отдавайте предпочтение качеству над стоимостью услуг.** Нанять недорогого, но плохого тренера/коуча, проигнорировав предыдущие факторы, на самом деле означает сэкономить копейку, а потерять рубль. Неудачные и негодные внедрения LeSS вполне возможны, найм плохого коуча здесь не поможет.
- > **Не поручайте выбор другим.** Это очень важное решение, чтобы оставить его тем людям, которые не будут непосредственно с ним связаны. Поэтому не поручайте выбор коуча другому подразделению, например, отделу управления проектами (PMO), закупок или кадров - они недостаточно вовлечены, чтобы учесть важные факторы.
- > **Не придавайте особого значения сертификации.** Большинство сертификаций людей и курсов почти бессмысленно. И хотя наличие сертификации, *наверное*, не повредит, тем не менее, она не является сколько-нибудь надежным руководством к выбору кандидаты. Приведенные выше пункты многое важнее³⁸.
- > **Оценивайте многих кандидатов.** Самые лучшие группы оценивали многих кандидатов, прежде чем принять решение и вкладывать средства в долгосрочное сотрудничество с ними.

³⁸ К этому относятся курсы подготовки "Certified LeSS Practitioner" (сертифицированный практик LeSS). Мы рекомендуем эти курсы, но для *обучения*, а не для *сертификации*.

1. Определите Продукт

Определение продукта означает определение пределов внедрения LeSS, содержащего Беклого Продукта и подходящей кандидатуры на роль Владельца Продукта. Чем шире определение продукта, тем лучше, но оно должно быть достаточно практическим для старта.

Составление определения продукта включает в себя следующее:

- > Расширение определения продукта посредством *расширяющих* вопросов вроде следующего: "Что думает заказчик является нашим продуктом?".
- > Ограничение определения продукта посредством *ограничивающих* вопросов вроде следующего: "Что практически в нашей текущей организационной структуре?".
- > Исследование усовершенствований с целью расширения определения продукта.

Подробнее о причинах, по которым чем шире определение продукта, тем лучше, а также о составлении определения продукта речь пойдет в главе 7.

2. Определите, что значит "Готово"

Чем лучше и строже понимание готовности - Критерий Готовности (Definition of Done или DoD), тем более обширные навыки требуются в командах. Так, если в Критерий Готовности входит тестирование производительности, команды должны приобрести подобный навык. Они могут быть приобретены путем обучения, но зачастую специалиста по тестированию производительности просто перемещают из специализирующейся на этом группы в команду. С другой стороны, если тестирование производительности исключено из Критерия Готовности, то отдельная группа, специализирующаяся на тестировании производительности, останется и будет действовать, как прежде *до тех пор, пока* Критерий Готовности не будет расширен. Из этого можно сделать следующий вывод.

Более совершенный и строгий Критерий Готовности означает более существенные организационные перемены (упразднение групп, ролей, должностей и т.д.), чем более скучный и ослабленный критерий.

Кроме того, менее строгий Критерий Готовности влечет за собой дополнительный риск и задержку! Все эти вопросы будут дополнительно обсуждаться в главе 10.

Влияние на масштабы организационных перемен делает Критерий Готовности крайне важным инструментом управления для внедрения LeSS. Руководителям приходится идти на компромисс между строгим Критерием Готовности, приводящим к большим организационным переменам и снижению риска и задержки, и ослабленным Критерием Готовности, приводящим к меньшим организационным переменам и повышению риска и задержкам.

Перед ними встает следующий вопрос: "С какими масштабами перемен способна в данный момент справиться их организация?"

3. Сформируйте правильно структурированные команды

Каждая команда разделяет ответственность за достижение поставленной общей цели. Чтобы поддержать эти успехи, необходимо структурировать их надлежащим образом. Ниже перечислены требования, предъявляемые к первоначальным командам.

- > **Выделенность.** Каждый работник является членом одной и только одной команды.
- > **Стабильность.** Члены команд меняются нечасто.
- > **Долговечность.** Команда не является временной проектной командой, а работает вместе *годами*.
- > **Кросс-функциональность.** В команде присутствуют все функциональные навыки для достижения оговоренного состояния готовности функционала продукта.
- > **Совмещенность.** Команда находится в одном месте, зачастую работая за одним большим рабочим столом. Благодаря непосредственному общению лицом к лицу возрастает доверие, а благодаря тому, что члены команды учатся друг у друга, повышается уровень их обучения.

См. главу 4.

Перечисленные выше требования к командам более подробно рассматриваются в главе 4.

Эта новая организационная структура подразумевает, что работники покидают свои функциональные подразделения, чтобы присоединиться к новым кросс-функциональным командам на постоянной основе. В связи с этим функциональные подразделения должны быть упразднены.

Почему бы не сохранить подотчетность людей руководителям своих функциональных подразделений? Дело в том, что это приведет к *конфликту лояльности*, нарушающему общие обязанности и сплоченность команды. Теперь вы можете подумать: "Они преувеличивают. В моей компании это может оказаться вполне приемлемо". Ничего подобного. Нам приходилось видеть немало потерпевших неудачу попыток добиться этого. Поэтому ни в коем случае не пытайтесь повторить этот печальный опыт. Вместо этого у всех членов команды должен быть один и тот же руководитель, перед которым стоит ясная задача создать благоприятную среду для успешной работы команды.

4. Постановите, что только Владелец Продукта обеспечивает команды работой

Вам, вероятно, знакомо это ощущение?... Длинный-предлениный рабочий день, куча-куча-куча работы, что же я успел *сделать*? Это вампир переключения контекста высасывает из вас жизненные силы. Непроизводительный, рассредоточенный и лишен всякой мотивации день.

Перед первоначальными командами стоит нелегкая задача: сосредоточиться на общих целях создания продукта, но в то же время преодолеть горы препятствий в своей рабочей среде. Такие препятствия (слабая автоматизация тестов, неподходящие инструментальные средства, неудобные правила и т.д.) проявляются во время работы кросс-функциональной команды в коротком цикле над доведением работы до состояния "готовности".

Эти первоходцы закладывают основание для формирования будущих команд, и поэтому потребность в их *сосредоточенности* на своей работе важна вдвое. Как они могут утратить? С благими намерениями и идеально разумные **прерывания** и запросы на дополнительную работу от непосредственных руководителей, отдела продаж, директора, кого-то из отдела кадров... Этого нельзя ни в коем случае допустить!

Во избежание этого следует постановить, что только Владелец Продукта обеспечивает команду работой. Это не только поддерживает фокус внимания, но и снижает напряженность из-за попыток каким-то образом отреагировать и удовлетворить противоречивые голоса, кричащие: "Я первый! Я первый! Нет, я первый!". Назначение приоритетов входит в обязанности Владельца Продукта, а не команды.

См. главу 8.

5. Держите проектных менеджеров подальше от команд

Роль руководителя проекта в группе продукта упраздняется в организациях, набирающих опыт работы по LeSS. Эта роль больше не нужна, и обязанности по управлению проектом разделяются между Владельцем Продукта и Командами.

В большинстве случаев внедрения LeSS роль руководителя проекта может быть упразднена *сразу*. В более редких случаях она остается *временно* необходимой - это обычно когда, либо Критерий Готовности неидеален (а следовательно, есть незавершенная работа на выходе команд), или есть серьезные потребности в координации, выходящей за пределы продукта или между продуктами. В подобных случаях организациям совсем не обязательно сразу же забывать о роли руководителей проектов.

Таким образом, проектные менеджеры иногда какое-то время еще будут нужны. Но дело в том, что они, скорее всего, будут регулярно вмешиваться и прерывать работу команд, внося противоречивые приоритеты. Поэтому руководителям проектов должно быть запрещено вмешиваться в работу команд, координировать их действия и поручать им какую-нибудь работу.

Эта рекомендация, по существу, равнозначна предыдущему постановлению о том, что работой команду должен обеспечивать только Владелец Продукта. Она распространяется и на другие руководящие роли. Мы посчитали важным указать на это явно.

И, кстати, ... просто взять и переименовать всех руководителей проектов в Scrum-мастера не поможет.

См. далее законы Лармана.

Что дальше?

См. главу 11.

В этом руководстве даны рекомендации относительно того, как начать внедрение LeSS, организовав правильную структуру. Следующий шаг состоит в том, чтобы сформировать Беклог Продукта. С этой целью можно воспользоваться первоначальным мероприятием по Проработке Беклога Продукта. Соответствующие рекомендации можно найти в главе 11.

Руководство: культура следует за структурой

Культура следует структуре — это, по существу, четвертый из "Законов Организационного Поведения Лармана". Люди в организациях приучены демонстрировать поддержку мероприятий в духе "голосование за усовершенствование месяца", по сути *ничего не меняя*. Нам приходилось наблюдать это явление неоднократно. Почему же это происходит?

Крейг имеет долгую карьеру разработчика, которая начиналась с программирования на языке APL в 1979 году и постепенно развилась до оказания помощи крупным продуктовым группам во внедрении современных практик управления. За кружкой пива он может упомянуть об уходе на пенсию. Но недавно он обеспокоился, когда обнаружил, что ни один из законов не назван в его честь. Поэтому он решил описать "Законы Организационного Поведения Лармана" в качестве напоминания о таком "патологическом" и корыстном стиле управления, которое поражает многие организации.

Законы Организационного Поведения Лармана гласят следующее:

1. Организации неявно оптимизированы, чтобы избежать изменения существующего порядка вещей (статус-кво), сложившегося среди руководителей среднего и нижнего звена, в должностях узких специалистов и во властных структурах.
2. Как следствие из первого закона: любая инициатива внедрения изменений будет сведена к переопределению или перегрузке новой терминологии, которая будет, по существу, обозначать то же самое что и существовавший статус-кво.
3. Как следствие из первого закона: любая инициатива внедрения изменений будет осмеяна как "туристская", "теоретическая", "революционная" и "требующая pragmatичной адаптации под локальные потребности", что отвлечет внимание от устранения слабостей в сложившем порядке вещей среди руководителей и специалистов.
4. Культура следует за структурой.

Предвосхищая ваши мысли, следует также сказать, что и *структура следует за культурой*, особенно в молодых компаниях - стартапах. Но это утверждение имеет скорее поэтическое, чем буквальное значение.

Что мы имеем в виду? Пока *структурные* элементы (группы, роли, иерархии, правила, а в более широком смысле — *организационная структура* или *организационный дизайн*) не изменяются, не изменяется ни поведение, ни образ мышления. Джон Седдон (John Seddon), идеальный лидер системного мышления, поясняет утверждение о том, что "культура следует за структурой", следующим образом.

Попытка изменить культуру, сложившуюся в организации, безрассудна и обречена на неудачу. Поведение людей (т.е. культуры) является продуктом системы. Если изменить систему, то изменится и поведение людей.

Нам приходилось наблюдать, как многие организации пытались внедрить LeSS, но отказывались соответственно поменять организационную структуру, роли и правила. Всем этим организациям так и не удалось извлечь полную выгоду из применения LeSS.

Часть трудностей составляет *личная безопасность*. Безусловно, люди не хотят потерять работу из-за структурных изменений. И это одна из причин, по которой при внедрении LeSS акцент делается на принципе бережливого производства - гарантии работы, но не роли.

Руководство: гарантия работы, но не гарантия роли

Человека трудно заставить что-то понять, когда его работа зависит от его непонимания.
— Алан Синклер

Кто будет стремиться к непрерывному совершенствованию, если в конечном итоге он может потерять работу? Никто. Поэтому при внедрении LeSS крайне важно установить правило, что никто не потеряет работу — по крайней мере, не из-за упразднения должности или роли вследствие структурных изменений, обусловленных внедрением LeSS. Об этом необходимо говорить ясно и неоднократно.

Работники из упраздненных функциональных подразделений вступают в ряды команд LeSS. Бывшие руководители функциональных подразделений могут поступить так же, поскольку они, как правило, имеют навыки и практический опыт работы по созданию ценности. А организация должна активно помогать всем своим работникам в поисках их *новых ролей* в новой структуре.

Подробнее об изменениях в управлении см. в главе 5.

Руководство: организационное видение совершенства

Организации представляют собой удивительно сложные системы, в которых невозможно контролировать все или знать все.

Каждый работник принимает мелкие решения, из которых возникает организационное поведение. Люди принимают решения, опираясь на свой опыт, цели, принципы и ценности. Если эти решения не согласованы, то люди мечутся с благими намерениями в разных направлениях, что заводит в организационный тупик или безвыходное положение. А если эти решения согласованы, то энергия направляется в нужное русло и дело начинает двигаться и улучшаться.

Это особенно справедливо в отношении улучшений. Нам приходилось видеть немало “улучшений” с благими намерениями, которые только приводили к дополнительной бюрократии и вызывали больше мучений. Когда же улучшение действительно оказывается улучшением? Очевидно, что это должны быть глобальные системные улучшения, а не локальная оптимизация. Но как об этом узнать? Отделить настоящие

системные улучшения от локальной оптимизации помогут поиски ответов на следующие вопросы.

- > Приблизит ли нас улучшение к нашей организационному видению совершенства?
- > Станет ли улучшение действительно улучшением на гембе — реальном месте работы?

См. главу 5.

Понятие гембы (*gemba* - место создания ценности) рассматривается в руководстве по поводу принципа "увидеть все своими глазами" (*Go See*) в главе 5. А в этом руководстве основное

внимание уделяется организационному видению совершенства. Но прежде необходимо выяснить, что же такое видение совершенства?

Классическое видение совершенства в *бережливом производстве* - это система "точно вовремя" или "точно в срок" (*just-in-time*) компании Toyota: всякий раз, когда покупатель приобретает автомобиль, точно вовремя производится ровно один автомобиль. Подобное видение совершенства привело к идеальному потоку единичных изделий (*one-piece flow*), в котором система производства настроена на производство небольших партий (*batch*), в идеальном случае — одиночных. Такой идеал вряд ли вообще достижим, но в компании Toyota он способствовал непрерывному совершенствованию системы производства десятилетиями.

Ниже приведено видение совершенства для LeSS, которым мы пользуемся:

Создавать организации, способные доставлять ценность или изменять направление в любой момент без дополнительных затрат.

Видение совершенства отличается от обычного видения (концепции). Цель обычного видения — достичь его, а цель видения совершенства — направить усовершенствования. Когда вы достигнете видения, вы отпразднуете успех, но когда достигнете видения совершенства, то огорчитесь тем, что оно только что стало бесполезным.

Успешные продуктовые группы, с которыми нам приходилось работать, имели *организационное видение совершенства* — недосягаемую цель того, чем группа стремится стать и как она мечтает функционировать. Как же оно используется на практике? Люди обсуждают и оценивают решения, исходя из того, насколько близко они приближают их к видению совершенства.

Обсуждение, конечно, важно, но слова забываются. Поэтому, чтобы быть вместе на одной волне, люди стараются записать свое видение. В качестве примера ниже приведен ранний вариант принципов, установленных клиентом, внедряющим LeSS Huge в группе продукта.

-
1. Цель совершенства — постоянно иметь выпускаемый продукт. Периоды стабилизации выпусков должны быть сокращены и в конечном счете устранины.
 2. Совмещенные, самоуправляемые и кросс-функциональные Scrum-команды являются основными строительными блоками организации. Ответственность и подотчетность - на уровне команд.
 3. Большая часть команд организованы как продукто-ориентированные команды с ориентацией на заказчиков.
 4. Управление разработкой продуктов ведется через роль Владельца Продукта. Обязательства по выпуску продукта (по срокам и объему работы) не налагаются на команды.
 5. Линейная организация является кросс-функциональной. Функционально специализированные линейные организации постепенно интегрируются в кросс-функциональную линейную организацию.
 6. Специальные координирующие роли (например, менеджер проектов) упраздняются, а за координирование отвечают команды.
 7. Главной обязанностью руководства является улучшение, в том числе обучения команд, эффективности и качества. Содержание работы всегда определяется Владельцем Продукта.
 8. Ветвление в разработке отсутствует. А вариации в продуктах не отражаются в системе контроля версий исходного кода.
 9. Все тесты автоматизированы, за исключением исследовательского тестирования (*exploratory testing*), тестирования удобства использования (*usability testing*) и тестов, требующих физического перемещения. Все работники должны приобрести навыки автоматизации тестирования.
 10. Внедрение изменений происходит постепенно и эволюционно. Эти принципы учитываются при каждом решении.

Это, конечно, всего лишь пример, но вы вольны использовать его в качестве отправной точки для обсуждения вашего видения совершенства.

Руководители вместе со всей группой должны установить такое организационное видение совершенства, чтобы оно направляло принятие решений. Как правило, это делается в ходе неформальных обсуждений и рабочих сессий, на которых вырабатывается наставляющее организационное видение и его принципы. Представить себе организационное видение можно двумя способами.

1. Представьте себя на работе - как выглядит и как функционирует идеальная совершенная организация.
2. Мысленно представьте идеальный совершенный продукт, а затем создающую его организацию.

Руководство: непрерывное улучшение

Внедрение LeSS завершается лишь тогда, когда достигается совершенство и мировое господство. Без этого всегда найдется, что улучшать.

См. главу 5.

Задача руководителей — создать среду, в которой команды непрерывно доставляют ценность и улучшаются. Желательно, чтобы большую часть улучшений осуществляли

сами команды, хотя руководители и Scrum-мастера нередко вовлекаются в улучшения организации и среды.

Советы:

> **Сфокусируйтесь!** Не делать никаких усовершенствований только потому, что все заняты выдумыванием новых улучшений, — это самая большая ошибка, препятствующая непрерывному улучшению: "А давайте проведем еще одну оценку текущего состояния", "Слушайте, оценки прежние... Интересно, почему?".

Или популярная альтернатива: "Давайте попробуем внедрить ГеркуЛеSS, потому как LeSS, кажется, у нас не работает" (без того, чтобы по-настоящему попробовать LeSS)

Какой выход из этого положения? Перестать проводить постоянные оценки и начать что-то делать! Следует всегда иметь в виду два самых насущных улучшения и сосредоточиться на них. Если улучшений

См. главу 14.

не происходит, команды быстро потеряют к ним интерес и перестанут думать о новых.

> **Пользуйтесь ретроспективами для создания улучшений.** Основным местом для выявления новых улучшений служат Ретроспективы в командах и также Общая Ретроспектива.

> **Уделите основное внимание подлинным улучшениям.** Не все улучшения являются подлинными. Некоторые из них оказываются лишь видами локальной оптимизации, не улучшающими систему в целом, а только одну ее сторону. К числу наиболее распространенных видов локальной оптимизации относится 1) функциональная и 2) на основе неоспоримых допущений. **Функциональная локальная оптимизация** означает улучшения с точки зрения одной функциональной специализации, что нередко отрицательно оказывается на выходных характеристиках системы. На пример: "В связи с тем что тестировать в каждом Спринте неудобно, тестирование следует начать, когда система будет завершена, а следовательно, тестирование можно выполнить более эффективно". **Локальная оптимизация на основе неоспоримых допущений** - это улучшения,

основанные на допущениях (предпосылках) о том, как "все работает", которые оказываются ложными. Существенные улучшения системы часто требуют оспоривания подобных убеждений, иначе локальные улучшения мало что дадут. "Программирование следует завершить до тестирования", "Было бы эффективнее, если бы у всех был лишь один навык".

Предположения относительно улучшений, которые могут оказаться видами локальной оптимизации, ценные как возможности для изучения и расширения кругозора. Когда они высказываются, их следует анализировать вместе с этим лицом или командой, от которых они исходят. Такое обсуждение расширяет перспективы и закладывает основание для дальнейших улучшений.

- > **Избегайте представителей отделов контроля качества, технологических процессов, трансформаций и усовершенствований.** В крупных организациях обычно имеются отделы контроля качества и технологических процессов, сотрудники которых носят черные пояса Шести Сигм и занимаются проектами по усовершенствованию. Или даже лучше - в некоторых организациях существует отдел трансформаций. Избегайте их! Непрерывное улучшение должно осуществляться везде, всеми и постоянно. Наличие одного отдела, отвечающего за улучшения, — это самый верный способ уничтожить как сами улучшения, так и участие в них команд. Вместо этого пользуйтесь существующими организационными структурами для поддержки внедрения и усовершенствований.
- > **Избегайте специальных команд для улучшений; используйте обычные команды.** Эта рекомендация связана с предыдущей. В организациях часто создаются *команды для улучшений*, которым ставится задача реализации элементов улучшений³⁹. Нам не раз приходилось наблюдать, как такой подход терпел неудачу. Лучший подход состоит в том, чтобы над элементами улучшений работали обычные команды. Над такими задачами можно разрабатывать вместе с обычными или сосредоточиться только на элементах улучшений в течение нескольких Спринтов. Существенное преимущество такого подхода заключается в том, что обычная команда, вероятнее всего, сама воспользуется впоследствии своим улучшением, а следовательно, она реализует его более полезным и пригодным для употребления.

³⁹ Такое организационное поведение отражает влияние приверженцев теории научной организации труда Фредерика Тейлора, называемых тейлористами. Обсуждается в главе "Менеджеры".

- > Избегайте проектов по улучшению; используйте
Беклог Продукта. Также, в организациях нередко
допускается, что все улучшения должны осу-

См. главу 9.

ществляться в отдельных "проектах". Такие проекты ведутся отдельно и выполняются командами для улучшений (см. предыдущую рекомендацию) или, что еще хуже, с привлечением людей из обычных команд. В последнем случае создается организационную суету и борьба за "ресурсы", а команды теряют сосредоточенность и чувство разделяемой ответственности. Вместо этого следует привлекать обычные команды, предлагая им элементы улучшений через Беклог Продукта. Таким образом, все работа будет видна в Беклоге Продукта, а непрерывное совершенствование станет нормальной системой.

Наиболее частой причиной краха непрерывного совершенствования служит фактическая неспособность к совершенствованию. Это вызывает разочарование в командах и недоверие к руководителям. Когда происходит нечто подобное, руководители должны остановиться, поразмыслить и задать себе вопрос: "Какого рода услуги мы предоставляем?".

Руководство: рост внедрения

Итак, первое внедрение LeSS завершено! Что же дальше? Достигли ли вы совершенства и мирового господства? Если нет, то сделайте следующее.

См. главу 10.

- > **Распространите действие LeSS на ряд других продуктов с той же самой поддержкой.** Очевидно, вы расширились, но на сколько продуктов? Возможно, на два следующих продукта, а не на один, но не более. Главным ограничением в данном случае являются люди, ресурсы и фокус, которые вы сможете привлечь, чтобы сохранить сосредоточенность и даже улучшить поддержку каждого продукта. Типичное затруднение, которое мы видим в том, что полная сосредоточенность на поддержке первого внедрения ослабевает и рассеивается при дальнейшем распространении LeSS. Не позволяйте этому произойти. Каждому следующему продукту требуется такая же самая поддержка и сосредоточенность.
- > **Усильте Критерий Готовности.** Критерий Готовности, скорее всего, неидеален. Усильте его, повысив кроссфункциональность команд, тем самым выявив новые препятствия для преодолений.
- > **Расширьте определение продукта.** Первоначальное определение продукта нередко оказывается ограниченным организационной структурой. Постарайтесь расширить его, чтобы улучшить назначение приоритетов, усилить сосредоточенность на клиентах и упростить организацию.
- > **Улучшите отдачу команд и подумайте, как распространить ноу-хау.** Результаты, достигнутые первыми командами, вряд ли окажутся превосходными.

См. главу 7.

Они выявили недостатки в своей среде и практиках разработки. Им пришлось немало учиться и улучшать, но многие ограничения все еще остались. Если устраниить их, повысится отдача. Распространите принятые с этой целью решения на другие команды и продукты.

- > **Улучшите поддержку.** Выясните, насколько эффективной была поддержка первоначальных команд. Получите с этой целью отзывы команд и воспользуйтесь ими, чтобы улучшить поддержку (обучение, коучинг, организационные изменения и т.д.), чтобы она была доступна для будущих продуктов, где внедряется LeSS.
- > **Направьте в нужное русло всю энергию, идущую снизу вверх.** Положительные результаты, достигнутые первоначальными командами при работе над первым продуктом, могут заставить команды в других группах внедрить LeSS, не дожидаясь одобрения высшего руководства. Вместо того чтобы подавлять эти инициативы, позвольте им происходить и поддержите их, используя эту энергию снизу.

LeSS Huge

При дальнейшем масштабировании возникает следующее дополнительное затруднение:

Масштабы слишком велики для одновременных структурных изменения - в очень большой продуктовой группе намного труднее вносить крупные структурные изменения. И дело не только в количестве людей и вовлеченных умов, а в следующем:

- > Имеется масса клиентов, которым обещаны новые продукты и функционал к определенным датам, что делает крупные изменения рискованными.
- > Организационные политики служат причиной того, что подобные изменения ограничивают *карьерный рост*.
- > При таких масштабах трудно обеспечить достаточное обучение и коучинг.

Таким образом, внедрение LeSS Huge осуществляется более эволюционным путем.

Правила LeSSHuge

Внедрение LeSSHuge, включая структурные изменения, осуществляется икрементальным эволюционным путем.

Каждый день следует помнить, что внедрение LeSS Huge отнимает много месяцев или даже лет и требует бесконечного терпения и чувства юмора.

Руководство: инкрементальное эволюционное внедрение

Внедрять LeSS лучше всего сразу, тогда как LeSS Huge —инкрементальным эволюционным путем. Имеются два следующих подхода ко внедрению LeSS Huge.

См. раздел
“Руководство: карты
внедрения для
продукто-
ориентированных
команд” главы 4.

- > **Постепенное инкрементное внедрение во всей продуктовой группе.** В этом случае все команды постепенно расширяют свои пределы и возможности в одном и том же темпе. Это может быть достигнуто за счет расширения Критерия Готовности на уровне продукта и применения таких инструментов, как карты внедрения для producto-ориентированных команд.
- > **Сосредоточенное углубленное внедрение в одной части продуктовой группы.** Улучшения сосредоточиваются на совершенствовании нескольких команд, а затем распространяются по очереди команда за командой. Это может быть достигнуто путем расширения Критерия Готовности в нескольких командах, предоставления им возможности работать над отдельными элементами улучшений, а также сосредоточенного коучинга.

Оба упомянутых выше подхода работоспособны. *Нетерпеливое* постепенное инкрементное внедрение обладает преимуществом *гипотетически* быстрее распространится и принести результаты на всем продукте, хотя этого зачастую не происходит, поскольку все команды должны одновременно разрешать одни и те же проблемы, что порождает новые проблемы. Сосредоточенное углубленное внедрение кажется медленным, но избегает страданий во всех командах одновременно. Его недостаток, безусловно, заключается в том, что уже существующие "боли" не устраняются, поскольку на них (пока еще) не сосредоточено основное внимание внедрения.

Принципы внедрения LeSS отдают предпочтение сосредоточенному углубленному внедрению, которое рассматривается далее в этой главе. А постепенное поэтапное внедрение рассматривается в главе 4.

Руководство: одна Область Требований за раз

Самым простым постепенным шагом для внедрения LeSS Huge является, внедрение LeSS в одной Области Требований. Благодаря этому внедрение LeSS сосредоточивается на той области, где выгоды велики, а риски малы — или по меньшей мере на последнем.

Это подразумевает одновременное создание только *одной* новой Области Требований за раз.

Но сделать это непросто потому, что новая, а, возможно и единственная, Область Требований по-прежнему является частью продукта. Следовательно, будут зависимости между Областями Требований и от обширной “старой организации” и ее частей. Самое трудное — найти золотую

середину между поддержкой этой новой Области Требований, с одной стороны порвав со “старой организацией”, а с другой стороны соблюдая соответствия организационным интерфейсам.

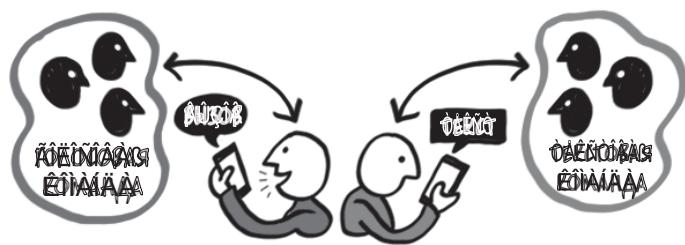
Решать вам. Один разрыв со “старой организацией” должен произойти таким образом, чтобы *отказаться от индивидуального или командного владения кодом*. В противном случае новая Область Требований не имеет никаких шансов выжить.

Наставление: параллельные организации

Предыдущее наставление служит примером более общего подхода к внесению структурных изменений, *ничего не меняя*. Такой подход заключается в построении параллельной организации. Это означает, что существующая организация сохраняется по мере построения рядом с ней новой организации, начиная с нескольких продукто-ориентированных команд или одной Области Требований. Такой подход вполне пригоден для продукто-ориентированных команд, поскольку у них, по существу, нет никаких зависимостей. И как только первые команды начинают нормально работать, вы постепенно перемещаете команды из традиционной организации в новую. А когда появляется достаточная подъемная сила, старая организация просто вливается в новую.

Ниже приведены некоторые предостережения.

- > Параллельная организация не является пилотной, а следовательно, линия организационной подчиненности должна быть отделена от традиционной организации.
- > Не допускайте разветвления кода в параллельной организации, поскольку это приведет к так называемому "аду слияния кода" (merge-hell). Обе организации действуют по отдельности, но работают над одним и тем же продуктом и с тем же самими кодом.
- > Четко поясните, что в конечном итоге все окажутся в новой организации. Это очень важно объяснить, чтобы работники в старой организации не пытались соперничать с работниками в новой организации.



Команды, организованные вокруг потребительской ценности

Организовывайте исходя из потребительской ценности

Я хочу, чтобы картинка была прозрачной, но при этом фон не должен проявляться.
— Анонимный заказчик

Scrum для одной команды

Центральной темой в Scrum является непреклонная сосредоточенность на доставке потребительской ценности. Порядок выполнения работы основывается на доставке ценности потребителям (заказчикам или клиентам), а не на удобстве разработки. Тем разработчикам, которые хотят сперва писать каркасы архитектуры, подобный фокус на проверке технических решений путем поставки ценности дается с трудом.

Золотую середину между непреклонной сосредоточенностью на потребительской ценности и стремлением к техническому совершенству обеспечивают следующие три роли в Scrum.

- > *Владелец Продукта* отвечает за возврат инвестиций. Ему приходится принимать трудные бизнес-решения. Что входит в продукт? А что нет? Когда выпускать продукт? Сколько инвестировать? У него должно быть ясное представление о продукте с точки зрения его потребителя.
- > *Команда* является кросс-функциональной, самоуправляемой и состоит из профессиональных разработчиков, разделяющих общую ответственность за доставку *готового* работоспособного и поддерживаемого функционала каждый Спринт. Они решают, как создать продукт и что для этого требуется.

- > *Scrum-мастер* отвечает за практическое и выгодное для организации применение Scrum. Он уделяет основное внимание росту нормально функционирующей производительной Команды и ответственного Владельца Продукта, а также непрерывному совершенствованию организации.

Организовывайте исходя из потребительской ценности в LeSS

При масштабировании подобную организацию описывают следующие принципы:

Ориентация на клиентов. Если продукт небольшой и разрабатывается одной командой, то организация по потребительской ценности не составляет большого труда. Чем больше команд, тем в большей степени они становятся винтиками в сложной машине разработки. Подобно Чарли Чаплину в фильме “Новые времена”, их задача — крутить гайки, не имея ни малейшего представления, каким образом потребитель будет пользоваться продуктом и кем он есть на самом деле. Как же в таком случае масштабировать, сохраняя *в то же время* основное внимание на потребителе?

Крупномасштабный Scrum — это Scrum. Однажды нам пришлось посетить команду, которая хотела внедрить Scrum. Мы объясняли им LeSS, когда они вдруг воскликнули: “Так вы хотите, чтобы мы делали то, что делали раньше, когда у нас была всего одна команда?” “О да!” - сказали мы. По мере того, как компания быстро росла, она привлекла “профессиональный менеджмент” и стала накапливать слои проектов, программ, портфелей и прочие атрибуты управления. Эта дополнительная структура разрушила саму суть компании — построение отличных продуктов. Как же сохранить масштабированный Scrum таким же простым, как сам Scrum?

Системный подход и сосредоточение на продукте в целом. Традиционные организации содержать немало видов локальной оптимизации, в том числе непреклонное стремление оптимизации индивидуального выхлопа. Как же структурировать организацию с большей сосредоточенностью на всем продукте и непреклонным стремлением поставлять потребительскую ценность?

Правила LeSS

Структурируйте организацию, используя настоящие команды в качестве базовых строительных блоков.

Каждая команда, во-первых, самоуправляемая, во-вторых, кроссфункциональная, в-третьих, совмещенная и, в-четвертых, долговечная.

Большинство команд - клиенто-ориентированы продуктовые команды.

Руководство: построение организации на основе команд

Ёширо Накамацу изобрел гибкий диск. К числу других его изобретений относится таблетка от засыпания, сигарета, активизирующая деятельность мозга, и презерватив со встроенным магнитом. Он считается мировым рекордсменом по количеству изобретений, имея более 4000 патентов, и служит ярким примером современного “сумасшедшего ученого”... Не смотря на это, большинство изобретений - и большую долю программных продуктов - создают *команды*, а не одиночки.

Продукты создаются командами, хотя традиционные (западные) организации построены на индивидуальной ответственности. В частности, подчиненный отвечает перед своим начальником за индивидуальную производительность своего труда. И это отражается в таких практики, как поручение работы *отдельным работникам*, оценка *индивидуальных показателей* производительности работников и *индивидуальное* вознаграждение. Подобные практики воспевают “сумасшедших ученых” индивидуалистов, но не нормально функционирующие команды, разделяющие *общую ответственность* за достижение их цели.

Организация, построенная на основе команд и практикующая LeSS, имеет следующую структуру:

- > **Д команды.** Каждый член команды посвящает все свое рабочее время одной и только одной Команде. Такой подход может показаться негибким, но члены команды должны работать целенаправленно, если требуется, чтобы они, во-первых, разделяли общую ответственность за достижение цели, стоящей перед командой, и, во-вторых, вступали во владение процессами, определяющими порядок работы в команде.
- > **Межфункциональные команды.** Каждая команда имеет или приобретает функциональные навыки для производства готового к поставке продукта. Традиционные функционально специализированные команды могут показаться самыми эффективными с точки зрения реализуемых функций, но большая часть усилий и трудностей в разработке продукта приходится на взаимосвязи *между функциями*, и поэтому команды должны быть межфункциональными, если требуется сосредоточить их основное внимание на продукте в целом.
- > **Совмещенные команды.** Каждая команда совместно размещается в одном помещении¹. На первый взгляд это может показаться неразумным. Не стоит ли в современном глобализованном мире воспользоваться услугами самых квалифицированных специалистов там, где они находятся постоянно? Нет, не стоит. Ведь нам нужны самые лучшие команды, члены которых разделяют общую ответственность за конечный результат и учатся друг у друга. Общая ответственность требует доверия, и, вероятнее всего, оно возникнет

¹ Но это совсем не означает, что все команды должны находиться в одном месте, хотя такое желательно. К сожалению, многоместная разработка весьма распространена в организациях, практикующих LeSS.

у людей в результате тесного сотрудничества и непосредственного общения. Совмещение способствует также ускорению ответной реакции и обучению команды, что составляет саму суть непрерывного совершенствования.

- > **Долговечные команды.** Команда остается вместе *навсегда*. И хотя это может показаться идеалистичным, тем не менее, Команды должны обладать постоянством, если требуется, чтобы они работали как одна команда. Всем, кому пришлось работать в по-настоящему долговечных командах, известно, что команды становятся лучше по мере того, как их члены узнают друг друга и вместе учатся выполнять и совершенствовать свою работу.

Столь выгодная организационная структура на основе команд становится причиной интересной динамики, которую очень важно распознать. Ведь иногда она кажется нелогичной и может вызвать *организационные опасения*. Признаки такой динамики описываются ниже.

Обучение людей вместо “трудовых ресурсов с одним навыком”. Организации часто рассматривают людей как трудовые ресурсы, относя их к той же самой категории, что и денежные средства, машины и служебные записки. Ресурсы имеют один навык. Так, машина делает именно то, что от нее требуется, а когда нужно сделать что-нибудь другое, потребуется *новая машина*. А люди рождаются вообще без навыков. Но они обладают необыкновенным метанавыком приобретать новые навыки. Именно этот метанавык наиболее важен для организаций, которые стремятся действовать гибко. Наличие целевых, долговечных команд автоматически вынуждает людей приобретать новые навыки в ходе обучения.

Команды вместо индивидуальных работников в качестве единицы выделения ресурсов. Под выделением ресурсов подразумевается процесс подбора людей для работы над продуктом, что обычно делается на индивидуальной основе. Если же следовать организационной структуре на основе команд, то больше не возникает вопрос: какие индивидуальные работники нужны? Вместо этого встает вопрос: какие команды нужны?

Работа поручается творческим командам вместо формирования команд вокруг работы. В традиционных организациях проектная группа формируется, исходя из нужных навыков и людей, требующихся для каждого вновь запрашиваемого функционального средства. А в организациях с долговечными командами не прибегают к реорганизации, а разделяют работу и поручают ее имеющейся команде, способной учиться и адаптироваться.

См. раздел “Наставление: разделение” главы 11.

Устойчивые организации вместо динамических матричных структур. Постоянное изменение организационной структуры не способствует гибкости, а вызывает недоразумение.

Напротив, подлинная организационная гибкость возникает в результате тщательно продуманного разделения работы с ориентацией на клиентов и ее поручение подходящим командам, которые приобретают недостающие навыки в ходе умелого обучения. В итоге организаций, практикующие LeSS, отказываются от матричных структур в пользу устойчивых организационных структур.

Наставление: представление о командах для функциональных средств

Большинство крупных групп продуктов организовано по технологии, следуя модели, которую можно назвать *команды для компонентов*. А группы продуктов в LeSS организованы по потребительской ценности, следуя модели, которую можно назвать *команды для функциональных средств*². Переход от технологии к потребительской ценности при организации групп имеет большое значение.

Что такое команды для функциональных средств

Команда для функциональных средств (рис. 4.1) является постоянной, долговечной командой, создающей комплексные функциональные средства, ориентированные на клиентов.³ Такая команда доставляет *готовые* функциональные средства в каждом Спринте.

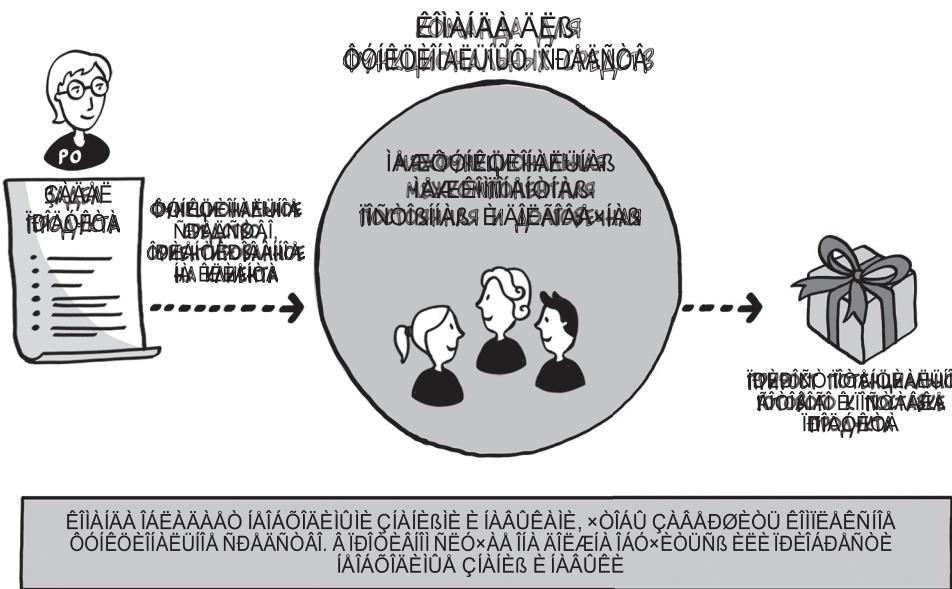


Рис. 4.1. Команда для функциональных средств

² Об этих двух моделях мы уже немало писали. Поэтому ниже следует краткое изложение наших предыдущих трудов. Подробнее об этих моделях можно узнать в упоминавшейся ранее книге *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, а также по адресу http://less.works/less/structure/feature_teams.html или <http://featureteams.org/>.

³ Но это совсем не означает, что *любая* команда может доставлять любое *функциональное средство*. Команды могут специализироваться на определенных видах функциональных средств, при условии, что они по-прежнему доставляют большую ценность.

Команды для функциональных средств обладают среди прочего следующими преимуществами.

- > **Совершенно четкие обязанности.** Цель команды для функциональных средств совершенно ясна. Функциональное средство (т.е. элемент из Задела Продукта) должно быть *готово*, прежде чем завершится Спринт. Все, что требуется сделать для достижения этой цели, относится к обязанностям команды. Благодаря этому упрощается планирование и разрешаются зависимости.
- > **Целеустремленность и сосредоточенность на потребностях клиентов.** Каждая команда для функциональных средств говорит на языке клиентов. Они создают функциональные средства для реальных людей, чтобы сделать лучше их жизнь, а не создавать технологию ради технологии. Столь повышенное внимание к клиентам и целеустремленность позволяют команде работать непосредственно с клиентами, говоря на их языке и создавая наилучший продукт вместе с ними, что весьма эффективно.
- > **Гибкость и обучение.** Вместо кошмарного планирования и непомерной зависимости от матриц достаточно найти подходящую команду, если потребуется новое функциональное средство.⁴ Возможно, у такой команды и не будет для этого необходимых навыков, но она может применить на практике свой метанавык обучения.

См. раздел “Наставление: разделение” главы 11.

См. раздел “Наставление: обращение с очень большими требованиями” главы 9.

Типичное недоразумение в отношении команд для функциональных средств заключается в том, что команда получает очень крупное функциональное средство, охватывающее всю систему, и поэтому ей приходится вносить изменения повсюду в системе. Но это не так. Напротив, очень крупное функциональное средство разбивается на более мелкие, ориентированные на клиентов, комплексные части, прежде чем оно передается на разработку команде для функциональных средств. Главное отличие заключается в том, что работа разделяется на части, ориентированные на клиентов, а не на компонентные части.

Переход к командам для функциональных средств требует ясного представления о том, как и почему они работают. Далее вкратце излагаются отличия команд для функциональных средств от команд для компонентов и анализируются их достоинства и недостатки. Ведь недостатки имеются и у команд для функциональных средств. Они не могут разрешить все затруднения, а для их внедрения требуется немало времени.

⁴ Очень важно понять, что функциональные средства не распределяются произвольно среди команд без учета их квалификации и опыта.

Модель команды для компонентов

Команды для компонентов организованы по архитектуре, как показано на рис. 4.2. Каждая команда специализируется на отдельной части системы или технологии. Это может быть клиентская или серверная часть приложения, язык Java или C++, а в более общем смысле — компонент (модуль, подсистема, каркас, библиотека и т.д.)

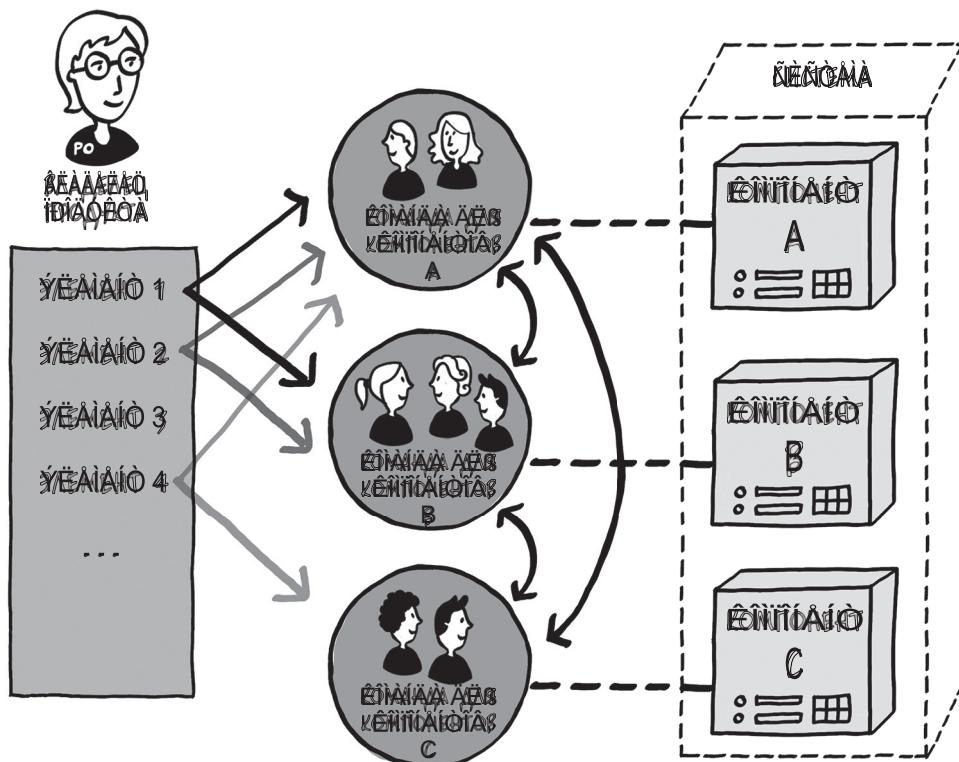


Рис. 4.2. Модель команды для компонентов

Такая модель является стандартной для большинства групп продуктов и обладает следующими преимуществами.

- > Четкое владение кодом и проектным решением.
- > Четкие границы (у каждой команды своя “песочница”).
- > Глубокая специализация.

Но эти преимущества не даются без значительных затрат, как поясняется ниже.

Четкое владение кодом и проектным решением. Владение кодом и проектным решением создает индивидуальность и четкую ответственность. Так, если обнаружена ошибка в *нашем* коде, то ясно, что ответственность за ее устранение несем именно *мы*.

С другой стороны, только одна команда может изменить код, что приводит к появлению узких мест. Кроме того, владельцы получат немного отзывов об альтернативном коде или проектном решении, поскольку никого больше не интересует *их* код.

Четкие границы. Например, в *нашей* области мы можем делать все, что угодно, тогда как другие команды не имеют права вмешиваться в *нашу* работу. С другой стороны, интеграция редко означает лишь собрать все вместе. Если интеграция завершится неудачно, то приходится выяснять, кто и за что отвечает, а это неприятно и требует времени. Ради снижения риска при создании продукта в LeSS основное внимание сосредоточено на всем продукте и на непрерывной интеграции, и благодаря этому удается избежать изоляции в “песочнице”.

Глубокая специализация. В частности, *наша* система сложна, и никто не может знать ее полностью. У *нашей* команды своя область, в которой мы специализируемся годами, и благодаря этому мы делаем свою работу лучше и эффективнее.

С другой стороны, специализация — это лишь одно, техническое измерение. Но за такое преимущество специализации (т.е. более локальную эффективность) приходится платить отсутствием специализации в других измерениях. Подробнее об этом речь пойдет в последующем наставлении отдавать предпочтение специализации в предметной области клиентов.

Модели команды для компонентов присущи следующие серьезные недостатки⁵.

- > Неуравновешенные и асинхронные зависимости.
- > Сосредоточенность на объеме выпуска продукции, а не на ее ценности.
- > Появление последовательных жизненных циклов и длительного цикла выпуска.

Анализ этих недостатков и типичных путей их устранения показывает, что добиться гибкости в работе команд для компонентов вряд ли вообще удастся. Перечисленные выше недостатки команд для компонентов вкратце описываются ниже.

Неуравновешенные и асинхронные зависимости. Клиентам требуются функциональные средства, которые обычно составляются из многих компонентов. Это приводит к появлению зависимостей между командами. Эти зависимости, во-первых, не уравновешены (например, команде “Зомби” приходится много работать, тогда как команде “Дракулы” — мало) и, во-вторых, асинхронны (например, работа команды “Маши” зависит от результатов работы команды “Оборотни”, которая занята другими более важными элементами). Это приводит к серьезным затруднениям в координировании действий и интеграции.

В качестве типичного выхода из этого положения приходится, во-первых, больше планировать, во-вторых, создавать новую роль для координирования действий и, в-третьих, формировать “проектную” команду, регулярно собирающуюся на совещаниях типа планерок. Все эти так называемые решения бесполезны. Зависимости вряд

⁵ Подробнее об этих недостатках можно узнать в упоминавшейся ранее книге *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, а также по адресу http://less.works/less/structure/feature_teams.html.

ли будут разрешены со временем, а быстрое устранение существующих в системе недостатков вызывает значительные трудности, муки, страдания и ужасные конфликты. У вас может сложиться впечатление, что мы преувеличиваем. Но если присмотреться внимательнее к тому, что действительно происходит за фасадом аккуратных отчетов о состоянии — даже в группах, которые годами пытались работать эффективно, то можно обнаружить полный беспорядок.

Сосредоточенность на объеме выпуска продукции, а не на ее ценности. Специализация в технической области может способствовать увеличению объема выпуска продукции, измеряемого в строках написанного кода, что не равнозначно потребительской ценности, особенно если оптимальное повышение эффективности оказывает влияние на назначение приоритетов для разработки функциональных средств. Что в конечном счете предпочут ваши заказчики: большой объем кода или ценные функциональные средства?

Появление последовательных жизненных циклов и длительного цикла выпуска. Кто проводит анализ исходных требований заказчика? Кто определяет работу над техническими компонентами, разрабатываемыми командой? Кто будет интегрировать и тестировать все функциональное средство, ориентированное на клиента? Аналитическая, архитектурная команда или же команда, выполняющая системное тестирование? Для этого придется вернуться к последовательному жизненному циклу со всеми его практическими проблемами и дополнительными задержками с длинными циклами выпуска.

Все эти недостатки, присущие модели команды для компонентов, хорошо известны, и никакие экстренные меры неспособны их устраниТЬ. От этих недостатков можно избавиться только в том случае, если перейти к модели команды для функциональных средств.

Модель команды для функциональных средств

Команды для функциональных средств организованы по потребительной ценности, как показано на рис. 4.3. Каждая такая команда может специализироваться на одном или нескольких видах функциональных средств в предметной области клиентов. Это может быть диагностика, торговля облигациями или администрирование.

Команды для функциональных средств обладают следующими преимуществами.

- > Четкое владение функциональными средствами.
- > Отсутствие зависимостей, вызывающих задержки.
- > Организация разработки на языке, понятном клиентам.

Как и в модели для компонентов, эти преимущества не даются даром. Ниже поясняется вкратце, в чем именно они состоят.

Четкое владение функциональными средствами. Кто отвечает за работоспособность всего функционального средства, ориентированного на клиента, в существующей системе? Во многих организациях любят играть в “отфутболивание”, постоянно перекладывая ответственность на другие команды. Такое безответственное поведение

полностью исключается благодаря тому, что команды для функциональных средств всегда берут на себя ответственность за разрабатываемые функциональные средства.

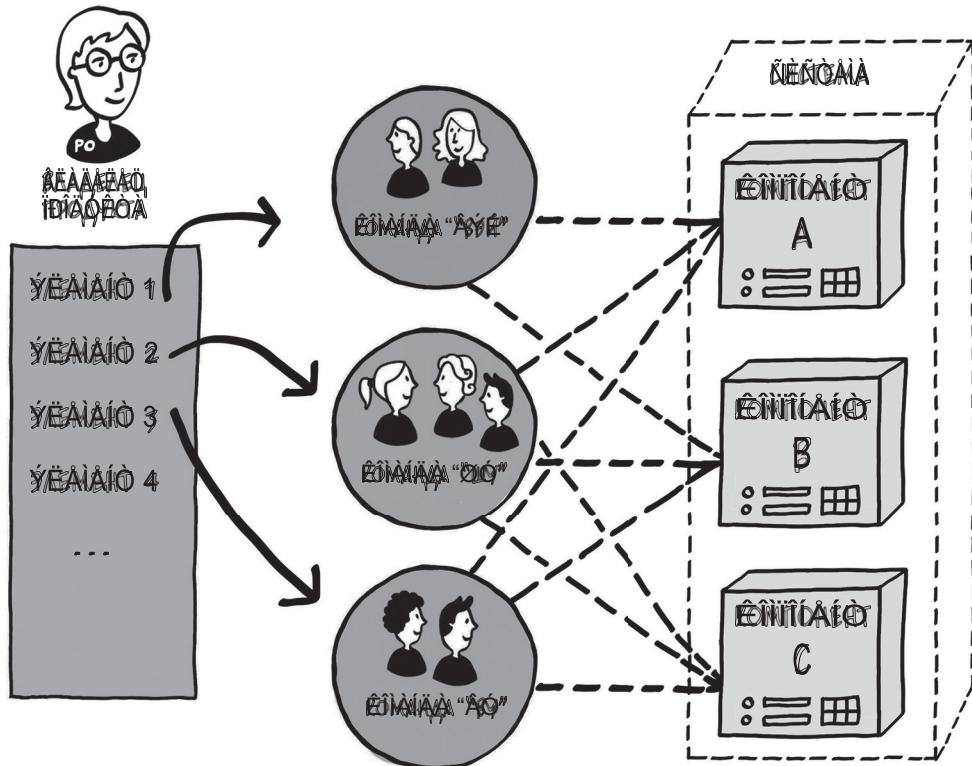


Рис. 4.3. Модель команды для функциональных средств

Подробнее о кураторах компонентов и многокомандных технических совещаниях см. в главе 13.

С другой стороны, команда для функциональных средств работает над многими компонентами. Над теми же самыми компонентами одновременно работают и другие команды, что в конечном итоге окажет влияние на проектирование и программирование этих компонентов. Такое влияние может быть

положительным, когда проектирование и программирование совершенствуются. Но многих беспокоит, что процесс разработки станет более беспорядочным. Поэтому внедрение современных норм практики разработки вроде модульного тестирования, беспощадной реорганизации кода, непрерывной интеграции, многокомандных технических совещаний и эволюционного проектирования способно предотвратить деградацию компонентов, способствуя здоровому развитию продукта. Кроме того, кураторы и сообщества компонентов обеспечивают обучение и поддержку, необходимые для внесения изменений в те компоненты, с которыми команда еще незнакома.

Отсутствие зависимостей, вызывающих задержки. Если функциональное средство требует внести изменения в компонент, то эти изменения вносит команда, разрабатывающая данное функциональное средство, не дожидаясь, пока их внесет какая-нибудь другая команда. Благодаря этому сокращается потребность в синхронизации для доставки функциональных средств клиентам, а это, в свою очередь, значительно сокращает время, проходящее от запроса функционального средства до доставки присущей ему ценности.

С другой стороны, разрабатываться могут общие компоненты или платформы. Если каждая команда для функциональных средств сосредоточена *только* на реализации *своих* функциональных средств, то в конечном итоге одни и те же функциональные средства могут быть реализованы неоднократно. И такие команды теряют возможность для сотрудничества с другими командами. В качестве выхода из этого затруднительного положения можно сделать акцент на более тесном сотрудничестве команд, связанных с *технической реализацией*. К числу полезных для этой цели методов относятся многокомандное Уточнение Задела Продукта и многокомандное второе Планирование Спринта.

Соответствующие
наставления
см. в главах 11 и 13.

Организация разработки на языке, понятном клиентам. Команды для функциональных средств говорят на одном языке с заказчиками, что дает им возможность задавать заказчикам прямые уточняющие вопросы. Благодаря этому работа становится более целенаправленной, поскольку команды хорошо знают, кто, что и зачем должен делать, а кроме того, сокращаются уровни косвенности (аналитиков, ответственных за выпуск продукции, и руководителей проектов) между заказчиками и разработчиками как исполнителями.

Однако некоторые инженеры никогда не считали общение с заказчиками необходимым навыком. Одним из них было бы желательно вообще не общаться с заказчиками, а другие вообще не были на это способны. Как показывает наш опыт, расширение навыков воздается сторицей, хотя поначалу это может показаться неудобным.

Модели команды для функциональных средств присущи следующие недостатки.

- > Требует от разработчиков изучать большую часть системы.
- > Способна вызвать беспорядок в проектировании и программировании.
- > Оказывает влияние на порядок распределения работы.

Это довольно серьезные недостатки, хотя они и не кажутся непреодолимыми. Перечисленные выше недостатки команд для функциональных средств вкратце описываются ниже.

Требует от разработчиков изучать большую часть системы. Разработчикам приходится изучать большую часть системы, хотя типичное недоразумение состоит в том, что разработчики или команды должны знать *всю* систему, что неверно. У членов команды должна быть своя основная специализация, а у команд — специализация в конкретной области. Допустим,

См. раздел “Наставление: рабочее совещание по текущей архитектуре” главы 13.

система состоит из 50 компонентов. По традиции разработчики хорошо знают один компонент, тогда как членам команды для функциональных средств нужно хорошо знать несколько компонентов и бегло быть знакомыми с десятком компонентов, но совсем *не* обязательно знать все 50 компонентов.

Способна вызвать беспорядок в проектировании и программировании. Как упоминалось ранее, избавление от владения компонентами может привести к ухудшению качества проектирования и программирования. Это объясняется следующим мнением:

Подробнее о полезных нормах практики см. в главе 13.

общая ответственность означает отсутствие всякой ответственности. Техническое совершенство и современные нормы практики разработки позволяют воспрепятствовать ухудшению качества проектирования и программирования. А иногда подобное ухудшение не происходит потому, что разработчики знают,

что их код проверяют другие, и поэтому они прилагают дополнительные усилия для поддержания своей профессиональной репутации. Такую гордость за свой код следует всячески поощрять.

Оказывает влияние на порядок распределения работы. При наличии команд для компонентов работа разделяется на технические задачи разработки компонентов. Как правило, это делается отдельным лицом или группой лиц: архитекторов, аналитиков или составителей спецификации. Такого рода разделение

См. раздел "Наставление: разделение" главы 11.

происходит в предметной области клиентов и выполняется на совещаниях по уточнению Задела Продукта. Разделить работу, ориентируясь на клиентов, нетрудно, но делается это иначе.

Если у команд для функциональных средств отсутствует ясное представление о разделении работы с ориентацией на клиентов, то выполнить такую работу невозможно.

И хотя упомянутые выше недостатки реальны, они вполне устранимы. Переход к командам для функциональных средств несложен, если LeSS внедряется в 4 командах. Но совсем другое дело, когда LeSS Huge внедряется в 100 командах, на что уходят месяцы, а то и годы. Тем не менее такой переход вполне возможен и очень выгоден.

Зависимости в командах для компонентов и функциональных средств

Обе рассмотренные выше модели команд представлены для сравнения на рис. 4.4. Из такого сравнения можно сделать ряд важных выводов.

Главная трудность в работе с командами для компонентов заключается в асинхронном характере зависимостей между командами, связанными с функциональными средствами, ориентированными на клиентов. А команды для функциональных средств разрешают подобные зависимости, и благодаря этому создаются возможности для извлечения командами взаимной выгоды при выполнении общей работы, которой не препятствуют зависимости. Если применять нормы практики, возникшие еще в 1980-е годы, когда приходилось составлять обширную документацию на бумаге, прежде чем писать код, а интегрировать — только готовые части системы, то общая работа

оказывается сущим наказанием, поскольку такая работа — всего лишь абстрактное понятие. Но если применяются современные нормы практики гибкой разработки, когда основное внимание уделяется ясности кода, беспощадной реорганизации кода и непрерывной интеграции, то общая работа становится реальной возможностью. Зависимости команд, предназначенных для разработки компонентов, от функциональных средств не разрешаются, поскольку они носят системный характер⁶. Таким образом, для внедрения LeSS требуется, чтобы большинство команд были специально предназначены для разработки функциональных средств.

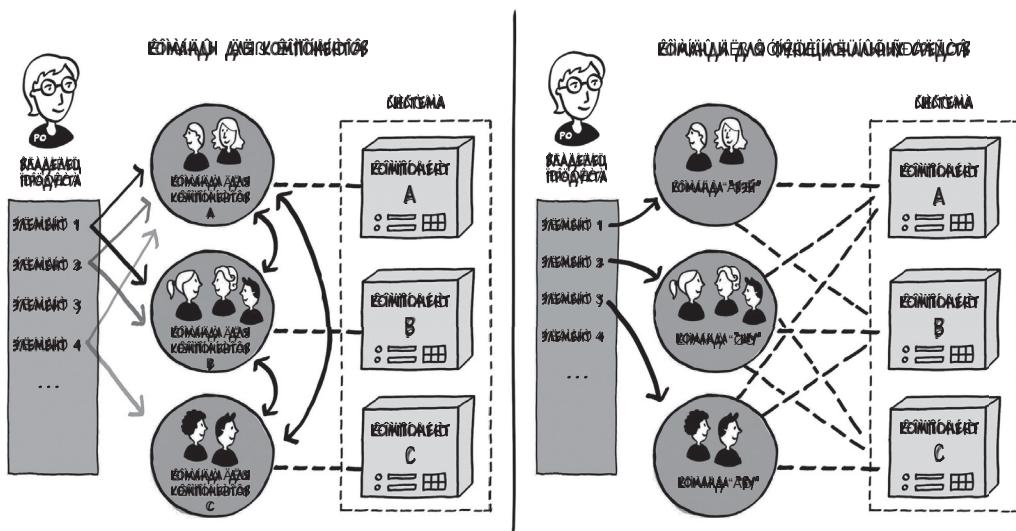


Рис. 4.4. Сравнение моделей команд для компонентов и функциональных средств

Наставление: карты внедрения команд для функциональных средств

Что такое компонент? Что такое функциональное средство? И что означает функциональная специализация? До сих пор мы рассматривали эти вопросы раздельно, но ответы на них лежат где-то посередине. Фронт работ одной группы может ограничиваться единственным классом, тогда как другая группа может работать над целой системой. Они относятся к разным видам команд для компонентов.

Существует аналогичный масштаб, связанный с функциональной специализацией. В некоторых группах продуктов имеется пять уровней тестирования, что придает весьма неоднозначный смысл намерению включить тестирование в обязанности команды. Если представить эти масштабы графически, как показано на рис. 4.5,

⁶ Нам приходилось не раз наблюдать, как организации неоднократно пытались устраниТЬ недостатки в командах для компонентов. Но дело в том, что подобные недостатки вообще не устранимы. К сожалению, многие организации познали это на своем горьком опыте.

то легче понять, чего следует ожидать от организационных изменений в связи с внедрением команд для функциональных средств.

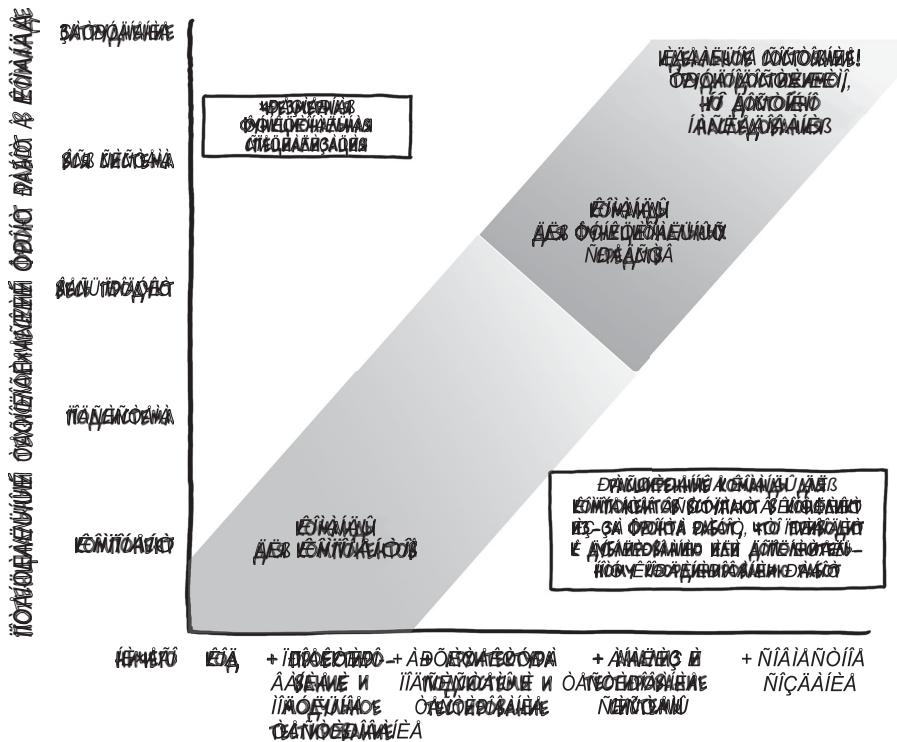


Рис. 4.5. Карта внедрения команд для функциональных средств

По оси Y на такой карте откладывается постепенно увеличивающийся объем работ для команд, выражаемый в виде архитектурного разложения на составляющие и расширения определения продукта. А по оси X откладывается степень межфункциональности команд, выражаемая в виде постепенно повышающегося Критерия Готовности.

На рис. 4.5 показаны следующие четыре области.

- > **Команды для компонентов.** Любая команда, сосредоточенная на реализации отдельных частей продукта, а не на функциональных средствах, ориентированных на конечного потребителя, или же на завершении задачи, а не на доставке прироста продукта, является командой для компонентов. Чем меньше объем работ и уже специализация, тем больше затруднений возникает у команды для компонентов.
- Более широкие определения продукта см. в главе 7, а определение Критерия Готовности — в главе 10.

- > **Команды для функциональных средств.** Любая команда, сосредоточенная на продукте в целом и вовлеченная во все виды работ от прояснения функциональных средств, ориентированных на клиентов, до тестирования, является командой для функциональных средств. Команды для функциональных средств существуют и в масштабах крупного проекта. Их обязанности могут быть ограничены только реализацией функциональных средств, определяющих их потребности. А если определение продукта оказывается довольно широким, то они могут быть вовлечены в выявление и разрешение реальных проблем клиентов, а следовательно, в совместное создание продукта во всей системе.
- > **Команда с чрезмерной функциональной специализацией.** Любая команда, выполняющая ограниченную задачу из крупного объема работ, вероятнее всего, будет иметь чрезмерную функциональную специализацию. А это, в свою очередь, приведет к значительным потенциальным проблемам из-за частых передач работ по кругу, чего следует всячески избегать.
- > **Расширенные команды для компонентов.** Любая команда, имеющая ограниченный объем работ, но ответственная за проверку работоспособности своей части крупного продукта, является расширенной командой для компонентов. В такой команде возникает внутренний конфликт из-за наличия как ограниченного фронта работ над компонентом, так и фронта работ над продуктом в целом. Такой конфликт приводит к дублированию работ, поскольку несколько команд создают одни и те же тесты, или же к необходимости прилагать дополнительные усилия для координации действий команд по тестированию продукта. Аналогичный конфликт из-за фронта работ возникает и при прояснении требований. Владельцу Продукта приходится напоминать команде, что в конце Спринта элементы должны быть *полностью* готовы. Такие команды, вероятно, лучше, чем обычные команды для компонентов, но и они сильно не дотягивают до выгод от команд для функциональных средств в отношении доставки ценности.

См. раздел “Наставление: непреклонность” главы 8.

Идеальной считается такая команда для функциональных средств, которая работает над системой в целом и создает продукт вместе с его фактическими пользователями. Это благородная, но трудно достижимая цель совершенства.

Примеры

Принимая во внимание такую цель совершенства, можно воспользоваться рассмотренной выше картой внедрения команд для функциональных средств. Обратимся для этого к двум конкретным примерам.

На рис. 4.6 показана карта внедрения команд для функциональных средств крупной системы телекоммуникационной сети в компании, внедрившей LeSS Huge. Когда они приступили к внедрению, у них существовали традиционные команды для

компонентов. Они выбрали такую стратегию внедрения, чтобы расширить функциональные рамки команд и создать *расширенные* команды для компонентов. На ближайшие несколько лет они поставили перед собой цель перейти к полноценным командам для функциональных средств на уровне всего продукта. Но у них имелся ряд общих компонентов, созданных равноправными группами продуктов, что затрудняло включение в продукты этих компонентов, поскольку для этого требовалась значительно более крупные организационные изменения. В итоге эти компоненты были исключены из их текущей цели.

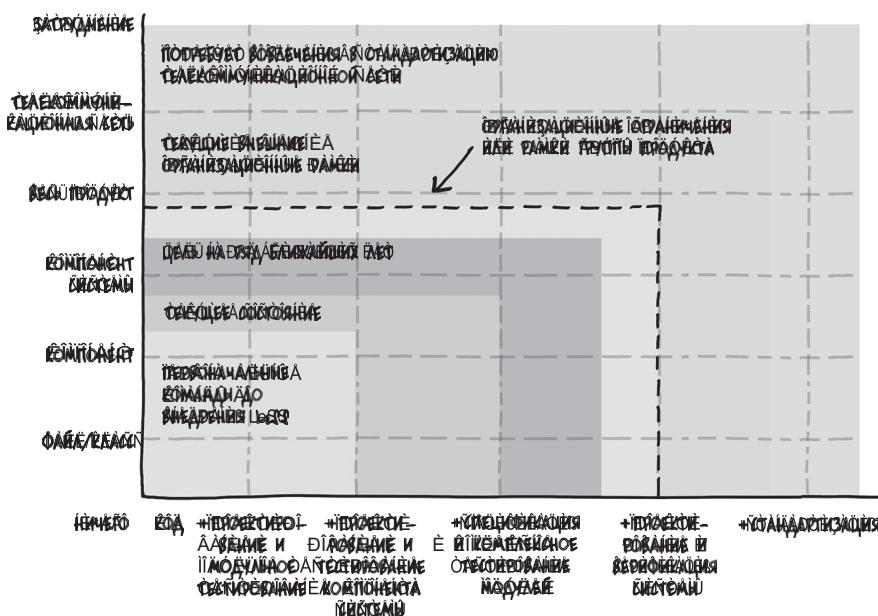


Рис. 4.6. Карта внедрения команд для функциональных средств системы телекоммуникационной сети

Расширить фронт работ до уровня системы нелегко, поскольку для этого требуется несколько кодовых баз с миллионами строк кода в каждой, непомерная степень функциональной специализации и полная реорганизация, затрагивающая тысячи людей. Поэтому координирование действий в группах продуктов и мероприятия по интеграции, скорее всего, останутся прежними в течение следующего десятилетия, постоянно вызывая осложнения.

На рис. 4.7 показана карта внедрения команд для функциональных средств продукта для финансовой торговли в компании с намного меньшими масштабами внедрения LeSS Huge. У них была та же самая отправная точка, что и у компании, разрабатывавшей телекоммуникационную сеть, но они решили внедрить все сразу. Разворачивание продукции по-прежнему выходит за рамки команд для функциональных средств, что отражает неидеальный Критерий Готовности.

Масштабы изменений, требующиеся при внедрении сразу всей инфраструктуры LeSS Huge, нередко оказываются слишком крупными, чтобы организация сумела с ними справиться. Именно поэтому внедрять сразу всю инфраструктуру LeSS Huge не рекомендуется. И данный случай служит ярким тому примером. Группа продукта внедрила команды для функциональных средств всего продукта, но за одним исключением: один очень важный компонент организационно относился к другой группе продукта. Переменные в организации, и особенно в данной группе продукта, послужили одной из причин, по которым при внедрении пришлось отступить на несколько шагов назад. Крупномасштабные организационные изменения служат благодатной почвой для воцарения скверных правил.

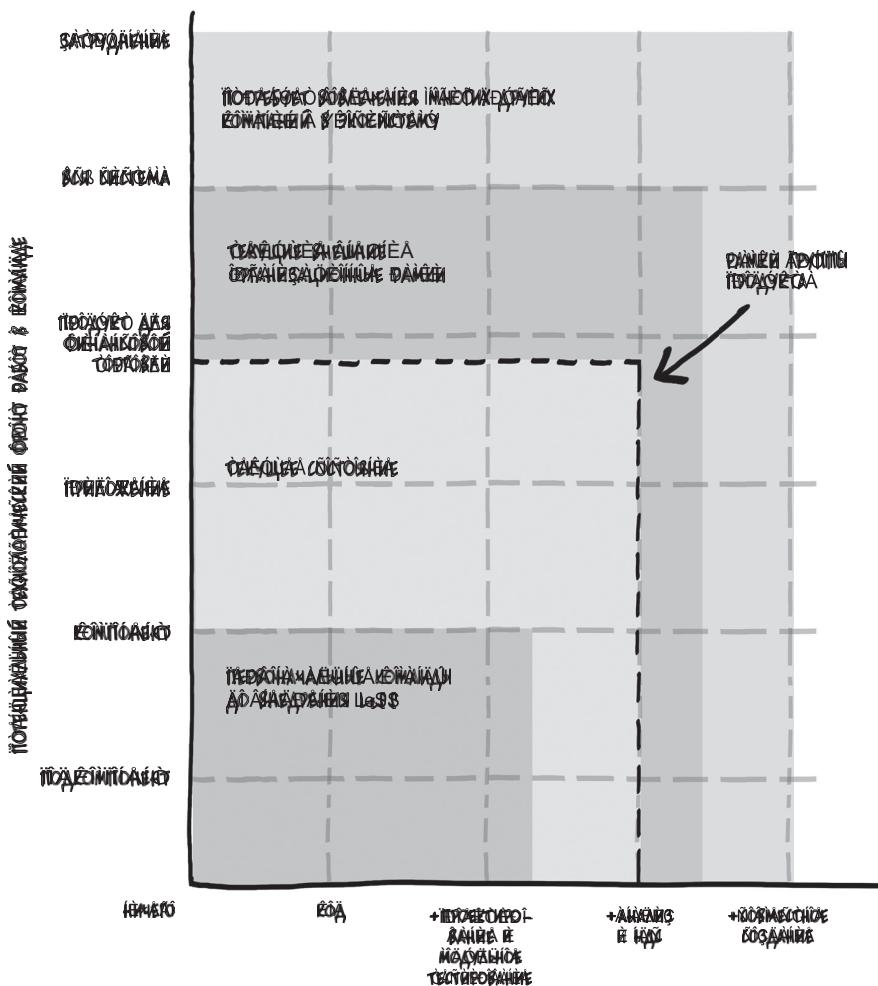


Рис. 4.7. Карта внедрения команд для функциональных средств системы финансовой торговли

Помощь в принятии решений

Карта внедрения команд для функциональных средств является важным инструментальным средством при внедрении LeSS. Она помогает в принятии следующих решений.

- > **Что означает “все и сразу”?** При более мелких масштабах внедрения инфраструктуры LeSS требуется менять *все и сразу* на команды для функциональных средств. Кого затронет такое требование изменить “все и сразу” — зависит от рамок команд для функциональных средств.
- > **Будущие цели улучшений.** Данной картой можно воспользоваться для постановки будущих целей, как это было сделано в примере с группой продукта телекоммуникационной сети. Такие будущие цели зачастую ставятся вместе с расширением Критерия Готовности. Данная карта показывает также ожидаемые перемены и связанные с ними трудности, поскольку для выхода за текущие организационные границы требуется немало потрудиться над преодолением “политических” границ.
- > **Что предпочесть: LeSS или LeSS Huge?** Рамки команд для функциональных средств определяют масштабы внедрения и могут повлиять на выбор LeSS Huge вместо LeSS. Например, инструментальное средство для анализа производительности сети является продуктом, ориентированным на клиентов, а для его разработки требуется группа, которой достаточно и более мелкой инфраструктуры LeSS. Но если этот продукт *всегда* продается как составная часть системы управления сетью, то его рамки изменяются и, вероятнее всего, потребуют внедрения LeSS Huge.

Наставление: отдавайте предпочтение специализации в предметной области клиентов

Одним из основополагающих принципов работы команд для функциональных средств является организация и специализация в предметной области клиента, а не применяемой технологии. Именно этим принципом и следует руководствоваться, принимая другие решения относительно структуры LeSS.

Общее недоразумение по поводу команд для функциональных средств заключается в том, что наличие таких команд вообще исключает специализацию. Отчасти такое недоразумение объясняется ложным разделением на две крайности: специализацию в одном компоненте или полное отсутствие всякой специализации, как подробно пояснялось ранее при описании команд для функциональных средств. А отчасти данное недоразумение объясняется убеждением в том, что специализация одномерна и распространяется на один конкретный компонент. Но на самом деле специализация многомерна. Исследуя ее измерения, можно найти лучшие способы уравновесить их.

Принято считать, что специализация почти полностью связана с навыкам разработки функциональных средств или компонентов, как демонстрирует карта внедрения команд

для функциональных средств. Но ведь существуют и другие измерения специализации. К их числу относится язык программирования, оборудование, операционная система, прикладной программный интерфейс API, рынок, тип заказчика и функционального средства. Эти измерения можно разделить на технологические (компоненты, операционные системы и т.д.) и ориентированные на клиентов (рынок, тип функционального средства и т.д.). Если рассматривать внедрение команд для функциональных средств, исходя из этих измерений, то можно построить диаграмму, приведенную на рис. 4.8.

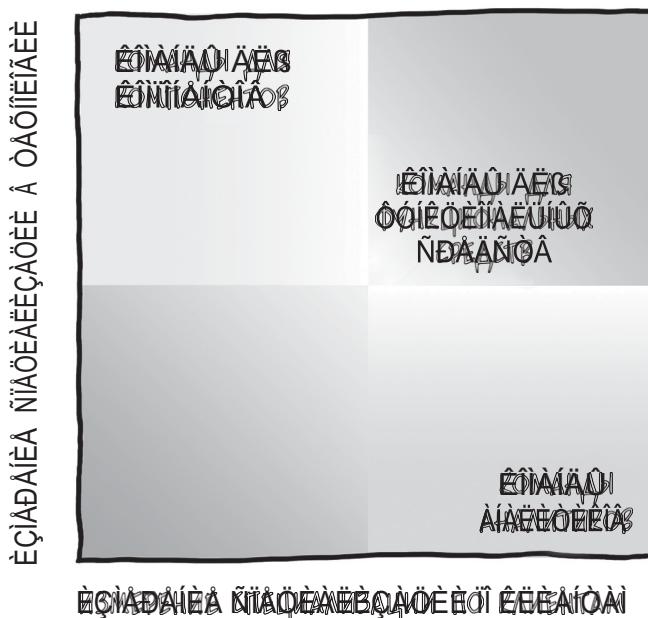


Рис. 4.8. Две категории измерений специализации

LeSS сближает пользователей и разработчиков. Точка зрения пользователя почти всегда теряется в традиционных группах крупных продуктов. Команды для функциональных средств служат удобным, но не единственным средством для организации по потребительской ценности. Принцип предпочтения специализации в предметной области клиентов приводит также к другим решениям относительно структурирования.

Например, банки создают приложения для оказания банковских услуг на мобильных устройствах. Как правило, команды организованы по платформам, например, команды iOS и Android, которые являются командами для функциональных средств и специализируются в техническом измерении, т.е. на конкретной платформе. Но они могут быть организованы по предметным областям клиентов, например, по платежам, администрированию и отчетности через мобильные устройства. В итоге команды реализуют однотипные функциональные средства на разных платформах вместо многих типов функциональных средств на одной платформе.

Какое измерение специализации лучше? Традиционные организации обычно выбирают специализацию в технологии, вероятно, потому, что технология воспринимается труднее, а следовательно, специализация в данном измерении может ускорить разработку. А в LeSS предпочтение отдается специализации в предметной области клиентов, чтобы расширить сотрудничество с реальными пользователями, устранив передачи работ по кругу и придать работе больший смысл. Обратимся к еще одному примеру.

Как-то нам пришлось работать с компанией, производящей графические адаптеры. Они структурировали свою организацию по технологии, сформировав одну команду для проектирования оборудования, вторую команду для разработки драйвера под Linux и третью — для разработки драйвера под Windows. Это были команды для компонентов, но для перехода к командам для функциональных средств потребовалась межфункциональная программно-аппаратная команда. Добиться такой цели в большинстве компаний-производителей оборудования трудно по причинам культурного характера. Программные команды дополнительны специализировались на прикладном программном интерфейсе API для драйверов. Такая организация основывалась на том предположении, что изучение прикладного программного интерфейса API для системных драйверов должно быть более важным и трудным, чем понимание оборудования — продукта, производимого компанией. А в LeSS предпочтение отдается организации вокруг заказчика, и поэтому может быть предложена альтернативная организация из команды для проектирования микросхемы, воспроизводящей двухмерную графику, и команды для проектирования микросхемы, воспроизводящей трехмерную графику.

Где находится золотая середина между специализацией в технологии и специализацией по клиентам? Внедряя LeSS, отдавайте предпочтение *специализации в предметной области клиентов*.

Наставление: организационная структура LeSS



Каким же образом все сказанное выше вписывается в организационную структуру? Безусловно, у каждой организации свои особенности, тем не менее, организации, практикующие LeSS, обычно придерживаются удивительно простой структуры. Первое отличие таких организаций от традиционных заключается в том, что их структура устойчива, поскольку работа организована по командам, а недостаток навыков побуждает к обучению и координированию действий в существующих командах.

Типичная схема организационной структуры LeSS приведена на рис. 4.9.



Рис. 4.9. Типичная организационная структура LeSS

Обратите внимание на отсутствие в такой организационной структуре следующих элементов.

- > **Функциональные организации.** Если члены команды, которые занимаются программированием, отчитываются перед руководителем отдела разработки, а члены команды, которые занимаются тестированием, подотчетны руководителю отдела контроля качества, то из такой команды ничего хорошего не выйдет. Ведь такая подотчетность приводит к *конфликту лояльности*, когда представитель отдела контроля качества проявляет одну лояльность к команде в отношении всей выполняемой работы, а другую лояльность — к руководителю отдела контроля качества в отношении своей функциональной специализации. Такой конфликт исключается в организациях, практикующих LeSS, упразднением функциональной организации и созданием вместо нее межфункциональных линейных организаций.
- > **Проектно-программная организация либо отдел руководства проектом или программой.** Столь традиционные контролирующие организации упраздняются в организационной структуре LeSS, поскольку их обязанности распределяются между командами для функциональных средств и Владельцем Продукта. Настояние на сохранении подобных организаций приведет к недоразумениям и конфликтам обязанностей.
- > **Группы поддержки, например, управления конфигурацией, непрерывной интеграции или контроля качества и технологических процессов.** Организации, практикующие LeSS, предпочитают расширять ответственность существующих команд, включая все перечисленные функции, чем создавать более сложную организацию со специализированными группами поддержки. У таких групп своя сфера деятельности, и поэтому они становятся узким местом.

Ниже вкратце рассматривается организационная структура в LeSS.

Подробнее о роли
руководства
см. в главе 5.

Руководитель группы продукта. В большинстве организаций, разрабатывающих программные продукты по методике LeSS, по-прежнему имеются руководители, в том числе и руководитель группы продукта. Они поддерживают команды по принципу посещения рабочих мест, помогая им преодолевать препятствия и совершенствоваться. (Более подробно об обязанностях руководителя речь пойдет в главе 5.) В организациях, практикующих LeSS, отсутствуют матричные структуры и функциональные руководители.

Название “руководитель группы продукта” может вызвать недоразумение, поскольку в большинстве организаций эта должность называется иначе. В данном случае имеется в виду линейный руководитель всех команд, как бы он ни назывался в вашей организации.

Команды для функциональных средств. Именно здесь и выполняются все проектные работы. Каждая команда для функциональных средств является межфункциональной, самоуправляемой и опекается Scrum-мастером. Такие команды являются постоянными единицами, действующими вместе в течение всего периода работы над продуктом, а иногда и дольше. Желательно, чтобы члены всех команд находились в прямом подчинении руководителя группы продукта. Нам приходилось наблюдать, как одному руководителю напрямую подчинялось 150 работников, а большинство мероприятий по управлению проводились командами. Но в некоторых крупных организациях, практикующих LeSS, имеется дополнительная структура линейных руководителей. Страйтесь избегать излишней организационной сложности при всякой возможности.

Владелец Продукта (Команда). Иногда еще называется *ответственным за выпуск продукции*. Владельцем Продукта может быть один человек, но в крупных организациях, практикующих LeSS, эта роль может также поддерживаться другими ответственными за выпуск продукции.

Следует иметь в виду, что в такой организационной структуре Команды и Владелец Продукта *равноправны* и не связаны иерархическими отношениями. На наш взгляд, очень важно сохранить равномерное распределение власти между отдельными ролями. Команды и Владелец Продукта должны быть связаны равноправными отношениями совместного *сотрудничества* для построения как можно лучшего продукта,

См. раздел “Наставление: кто должен быть Владельцем Продукта” главы 8.

а именно это и поддерживает равноправная структура. Более подробно этот вопрос рассматривается в главе 8.

Такая организационная структура особенно характерна для производственных компаний. Зачастую ее альтернативой, особенно для внутренней разработки, служит принадлежность Владельца Продукта другой организации — с деловой стороны. Следовательно, он *не* относится к иерархии руководителя группы продукта. Поэтому рекомендуется, чтобы Владелец Продукта поддерживал тесные отношения с Командами, хотя это нередко требует дополнительных усилий.

Отдел незавершенных работ — такой отдел в идеальном случае не существует. К сожалению, иногда команды неспособны еще создавать прирост по-настоящему готового к поставке продукта в каждом Спринте. И это отражается на их Критерии

Готовности, который не соответствует потенциально готовому к поставке продукту. Отличие между этими состояниями готовности продукта называется *Незавершенной Работой*. Кому-то все же придется выполнить эту Незавершенную Работу, и с этой целью обычно создаются отдельные группы, образующие *отдел незавершенных работ*. Подробнее об этом речь пойдет в главе 10.

Отделы незавершенных работ (например, тестирования, контроля качества или анализа деловой активности) *не должны вообще* существовать в группах, образующих мелкую инфраструктуру LeSS. Вместо этого они должны быть с *самого начала* включены в состав команд. С другой стороны, нам, к сожалению, приходится часто наблюдать отделы незавершенных производственных или эксплуатационных работ в организациях, внедряющих LeSS, где они нередко пересекают организационные границы.

Цель всякого внедрения LeSS — исключить отдел незавершенных работ. Сколько времени это потребует? Ответ на этот вопрос сильно зависит от того, насколько быстро организация усовершенствует свои возможности.

Наставление: организация многоместной работы в LeSS

Когда мы работали в одной компании, выпускающей сетевые игры, к нам присоединилась новая Владелица Продукта. Она спросила: “Где находятся мои команды?” Кто-то перечислил три города в Восточной Европе. Тогда она спросила: “Сколько лететь до первого города?” Этот вопрос вызвал всеобщий смех. Ей ответили: “Рейс и аэропорт отсутствуют. Вам придется лететь до Киева и ехать три часа поездом!” Новая Владелица Продукта была поражена. Это место было вскоре ликвидировано.

Разработку продуктов лучше всего выполнять лишь в одном месте. Впрочем, имеется немало веских (и не вполне веских) оснований для разработки во многих местах. Поэтому, выбирая подходящее место, руководствуйтесь следующими принципами.

- > **Сокращайте места.** Многоместная работа может оказаться неизбежной по внешним причинам. Но и в этом случае следует придерживаться правила совмещать рабочие места как можно больше. Ликвидируйте мелкие места или хотя бы сократите разницу во времени.
- > **Сокращайте разницу во времени.** Время служит большим препятствием, чем расстояние. Преодолевать физическое расстояние можно с помощью видеоконференций и текстовой интерактивной переписки, хотя все предпочитают личное общение у белой доски. Но единственный способ преодолеть разницу во времени — сдвинуть свой рабочий день. Большинство команд предпочитают не делать этого, и поэтому большая разница во времени гарантирует задержку в общении на один день.
- > **Совмещайте целые команды.** Члены команд разделяют ответственность за совместно выполняемую работу. А разделяемая ответственность требует высокого уровня доверия. К сожалению, расстояние порождает недоверие, поскольку людям трудно доверять тем, кого они не видят и с кем не общаются непосредственно. Кроме того, члены одной команды должны работать рядом, чтобы учиться друг у друга.

- > **Избегайте специализации отдельных мест по функциональным навыкам.** Неудачное, хотя и довольно частое распределение работы по местам осуществляется, исходя из функциональной специализации, например, разработки в одном месте, а тестирования (ради сокращения расходов) — в другом. Но такое распределение работы в LeSS не годится, поскольку оно приводит к тому, что члены каждой межфункциональной команды окажутся в разных местах.
- > **Избегайте специализации отдельных мест по компонентам.** Еще один распространенный способ разделения ответственности по местам состоит в том, чтобы взять схему архитектуры и назначить работу над отдельными частями архитектуры по местам. И этот способ не годится, когда внедряются команды для функциональных средств.

LeSS Huge

При масштабировании во внимание следует принять контекст и следующие вопросы.

- > **Ориентация на клиентов.** При крупных разработках очень легко забыть о клиенте, когда внедряемая структура отдаляет команды от заказчиков в сторону специализации только на технологии. Как воспрепятствовать этому и сохранить тесную связь многих (возможно, даже тысяч) разработчиков с клиентами?
- > **Добиваться большего средствами LeSS.** При увеличении масштабов до LeSS Huge кажется неизбежной потребность в какой-нибудь дополнительной структуре. Такую структуру обеспечивают Области Требований и роль Владельца Продукта в Области, сохраняя небольшой саму инфраструктуру.

Правила LeSSHuge

Требования заказчика, тесно связанные с его точкой зрения, сгруппированы в Областях Требований.

Каждая команда специализируется в одной Области Требований. Команды долгое время остаются в одной области. При появлении дополнительных ценностей в других областях команды могут изменить Область Требований.

Для каждой Области Требований имеется свой Владелец Продукта в данной Области.

В каждой Области Требований действует 4–8 команд. Эти пределы нельзя нарушать.

Наставление: Области Требований

Область Требований — это совокупность элементов из Задела Продукта, логически связанных вместе с точки зрения клиента (например, управление торговыми операциями или освоение новых рынков). Области Требований позволяют управлять отдельной областью как самостоятельным продуктом со своим внедрением LeSS в более мелких масштабах. Область Требований состоит из следующих элементов.

- > **Задел Продукта в Области.** Это подмножество Задела Продукта, относящееся к одной области. Это не отдельный задел, но логически связанное представление единого Задела Продукта, хотя его можно вести как отдельный задел. Более подробно об этом речь пойдет в главе 9.
- > **Владелец Продукта в Области.** Это отдельный Владелец Продукта, специализирующийся на логической области требований заказчика. Владелец Продукта в Области действует как обычный Владелец Продукта по отношению к командам. Он работает также с общим Владельцем Продукта и Владельцами Продукта в других Областях как член Команды Владельцев Продукта с целью сохранить сосредоточенность на продукте в целом. Более подробно об этом речь пойдет в главе 8.
- > **Команды для функциональных средств.** Это Команды, специализирующиеся на отдельных частях продукта, говоря в то же время на одном языке с клиентом. Каждая Команда относится только к одной Области Требований.

Области Требований являются главными структурными дополнениями LeSS при увеличении масштабов свыше 8 команд и создании инфраструктуры LeSS Huge. Они служат для разрешения следующих затруднений, возникающих при масштабировании LeSS.

- > **Задел Продукта слишком велик.** Допустим, что на каждую команду в течение Спринта приходится четыре элемента и в течение трех Спринтов 20 команд должны получить до конца проясненные готовые элементы. Это подразумевает наличие 240 элементов в мелкоструктурном разделе Задела Продукта. При таком количестве элементов в одном мелкоструктурном разделе Задел Продукта становится неуправляемым, не говоря уже о многих других менее мелкоструктурных его разделах.
- > **Владелец Продукта слишком перегружен.** Со сколькими командами может нормально работать Владелец Продукта? Если Владелец Продукта не вовлечен в подробное прояснение каждого элемента, а сосредоточен только на назначении приоритетов, работе с клиентами и сотрудничестве команд, то переломный момент наступает где-то в пределах от 5 до 10 команд (например, 8 команд). Сверх этого Владельцу Продукта трудно будет долго сосредоточивать в равной степени внимание на внутренних и внешних факторах.
- > **Совещания слишком многолюдны.** Если на совещании по Планированию Спринта присутствуют по два представителя от каждой из 20 команд, то оно становится слишком крупным. Продуктивно, сосредоточенно и эффективно проводить такие совещания очень трудно.

> **Команды теряют сосредоточенность.** Команды расстраиваются и замедляют свою работу, когда они слишком часто переключают свое внимание или охватывают слишком обширную область. Специализация в ориентированной на клиентов области способствует сосредоточенности, которая требуется для организации продуктивной команды.

На рис. 4.10 приведен пример структуры Областей Требований.

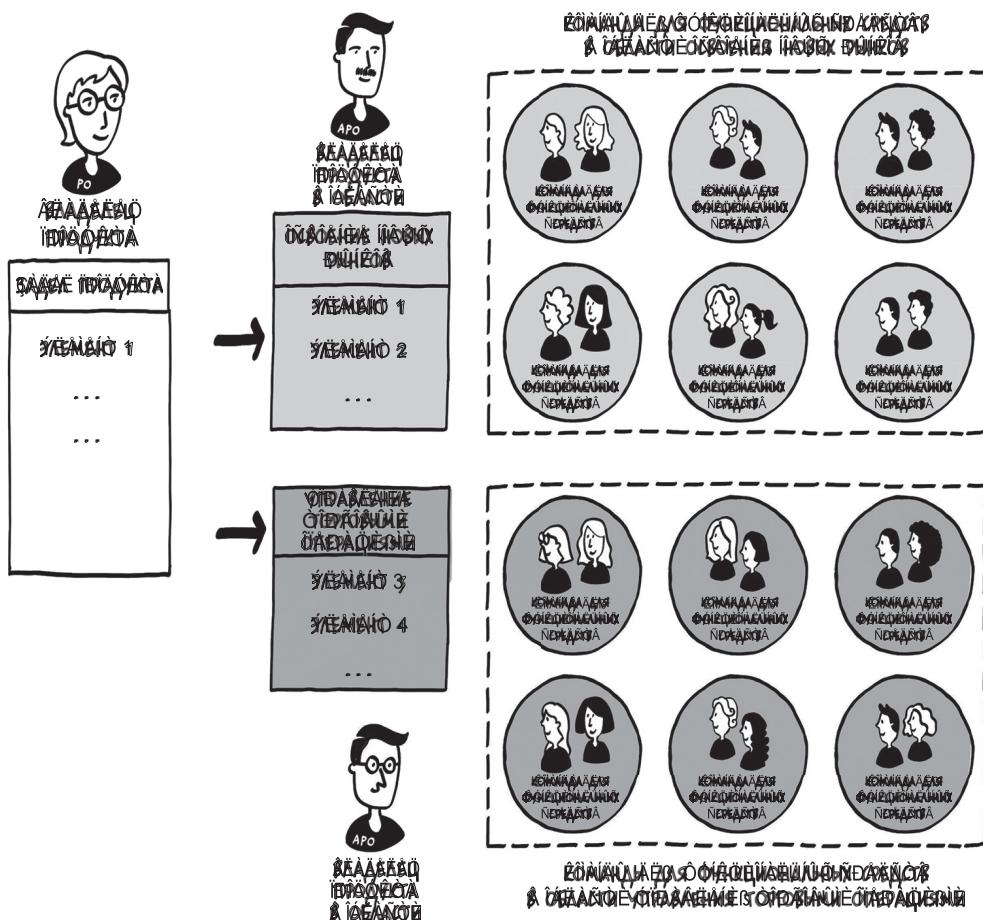


Рис. 4.10. Области Требований

Задел Продукта содержит все входящие в него элементы. Каждый из этих элементов назначается для одной и только одной Области Требований. В каждой Области Требований имеется свой Владелец Продукта, и все элементы, принадлежащие этой Области Требований, образуют Задел Продукта в данной Области, причем каждый элемент надолго относится к одной Области Требований.

Общий Владелец Продукта контролирует ценность элементов во всех областях. И когда отличие в ценности областей становится слишком большим, Владелец Продукта может переместить команду в другую область. Подобным образом Владелец Продукта уделяет особое внимание окупаемости инвестиций в весь продукт.

Наставление: динамика Областей Требований

Области Требований содержат от четырех до восьми команд. Но почему четыре команды? Сокращение Области Требований неизбежно приводит к недостатку прозрачности и локальной оптимизации. Чтобы выяснить причины этого явления, исследуем сначала эволюцию Области Требований во времени.

Зарождение. Новая Область Требований зарождается в следующих двух случаях:

- > Когда Область Требований становится слишком крупной, ее лучше всего разделить на две более мелкие, сгруппировав элементы в Заделе Продукта данной Области, чтобы обнаружить в нем естественное разделение. Это наиболее предпочтительный способ создания новых областей, сравнимый с наращиванием внедрения более мелкой инфраструктуры LeSS до масштабов LeSS Huge.
- > Когда появляется возможность создать (вероятно, крупный) новый продукт⁷, существенно отличающийся от предыдущих. В этом случае можно создать совершенно новую Область Требований, переместить в нее одну команду и постепенно нарастить ее как минимум до четырех команд.

См. раздел “История в LeSS Huge: новая Область Требований” главы 2.

Признаки того, что Область Требований становится слишком крупной, см. выше в разделе “Наставление: Области Требований”.

См. раздел “Наставление: обращение с очень большими требованиями” главы 9.

Зрелый возраст. Относительное значение Области Требований будет изменяться в течение строка ее существования. Дело в том, что клиенты неточно разделяют свои потребности на Области Требований, но вместо этого одна область получает на некоторое время более высокий приоритет, тогда как другие области — более низкий. Общий Владелец Продукта обязан распознать эту ситуацию и динамически откорректировать Области Требований, переместив команды туда, где имеется наибольшая ценность. Если же Области Требований не являются динамическими, это служит явным признаком более глубоких систематических затруднений.

Выбытие. Очень редко Область Требований исчезает сама по себе, поскольку в ней всегда происходят мелкие изменения. Но в конечном счете Области Требований сокращаются до масштабов менее четырех команд и объединяются. С этой целью следует взять две Области Требований, расширить их рамки до одинаковых масштабов,

⁷Это может быть совершенно новый продукт на рынке или непомерно крупное функциональное средство, создание которого потребует много команд и времени.

Это нередко происходит, когда объединяются места и Области Требований.

См. выше раздел “Наставление: организация многоместной работы в LeSS”.

а затем объединить Заделы Продукта в этих Областях в один задел, который продолжал бы вести один Владелец Продукта в объединенной Области. Лучше иметь полноценную объединенную область, но если это невозможно, то для начала следует соединить союзом и имена обеих объединяемых областей Требований.

Так зачем объединять мелкие области, избегая количества команд меньше четырех? Мелкие Области Требований в лучшем случае увеличивают объем работы для общего Владельца Продукта, которому приходится распределять приоритеты между Областями Требований, и благодаря этому Области Требований быстро изменяются. В худшем случае этого не происходит, а заодно теряется распределение приоритетов между Областями Требований и надзор за Заделом Продукта. Наличие мелких областей Требований обычно служит явным признаком следующих затруднений.

1. Наличие разрозненных областей Требований с Владельцами Продукта в этих областях со слишком большими полномочиями.
2. Отсутствие сосредоточенности на клиентах, приводящее к недостаточному распределению приоритетов в общем Заделе Продукта.
3. Чрезмерное участие в прояснении Владельца Продукта в Области, и вследствие этого он просто не в состоянии справиться более чем с двумя командами.

Наставление: переход к командам для функциональных средств

При внедрении (более мелкой) инфраструктуры LeSS переход к командам для функциональных средств происходит сразу во всей организации. А при внедрении инфраструктуры LeSS Huge можно выбрать одну из нескольких стратегий такого перехода. Какая же из них наиболее подходящая? Приведенные ниже действия помогут вам выбрать наилучшую стратегию для своей организации.

1. Определите свой контекст.
2. Выберите свою стратегию перехода.

Рассмотрим эти действия более подробно.

1. Определите свой контекст

На переход к командам для функциональных средств оказывают влияние следующие факторы.

- > **Величина группы продукта.** Очевидно, что группе продукта, состоящей из 10 команд, проще перейти к командам для функциональных средств, чем группе, состоящей из 100 команд.

- > **Срок действия продукта.** Продукты, которые будут, вероятно, существовать еще тридцать лет, претерпевают медленные изменения под предлогом снижения риска. А продукты, которые существуют лишь несколько лет, должны изменяться быстрее.
- > **Степень компонентной или функциональной специализации.** Чем больше специализация, тем более крупные изменения вызывает внедрение команд для функциональных средств. Чтобы наглядно представить текущее состояние компонентной или функциональной специализации, начертите карту внедрения команд для функциональных средств.
- > **Количество мест для разработки.** Чем больше мест для разработки, тем труднее внедрение команд для функциональных средств. Это тем более справедливо, когда отдельные места специализируются на определенных компонентах или функциях. Подобная специализация служит препятствием для межкомпонентного или межфункционального обучения.

2. Выберите свою стратегию перехода

В общем, имеются три стратегии перехода, которые вкратце описываются ниже

Все и сразу. Эта стратегия применяется также при внедрении LeSS. А в LeSS Huge стратегия “все и сразу” менее распространена из-за объема требующихся организационных изменений. Тем не менее такая стратегия пригодна в том случае, если группа продукта относительно невелика, срок действия продукта короткий, специализация низкая, а разработка совмещена в одном месте. Характерной ошибкой для внедрения LeSS Huge во всей организации сразу является недооценка требующихся масштабов обучения и инструктирования.

Постепенное расширение обязанностей команд для компонентов. Текущее состояние своей организации можно нанести на карту внедрения команд для функциональных средств, наметив будущие цели для расширения рамок команд. Межфункциональное расширение достигается расширением Критерия Готовности. Подробнее об этом речь пойдет в главе 10.

Нам не раз приходилось наблюдать на практике применение подобной стратегии. Она вполне работоспособна, но ей присущи два крупных недостатка. Во-первых, она привносит недостатки обоих типов команд для функциональных средств и компонентов, не давая особых преимуществ. И, во-вторых, она затрудняет внедрение Областей Требований, ориентированных на клиентов, когда команды все еще ориентированы на разработку компонентов. Тем не менее такая стратегия перехода подходит для многоместной среды, где должно происходить масштабное обучение во многих местах.

Параллельная организация. При такой стратегии сохраняется существующая организация команд для компонентов, и рядом с ней постепенно строится параллельная организация команд для функциональных средств (рис. 4.11).

См. раздел “Наставление: поочередное создание Областей Требований” главы 3.

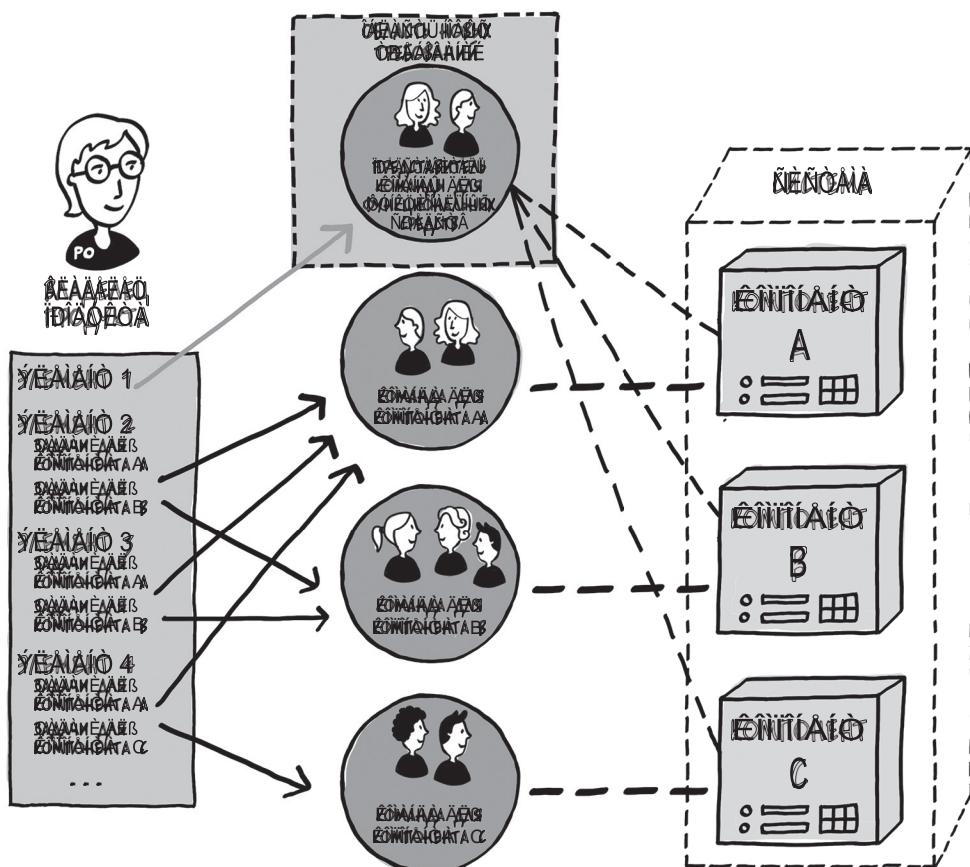


Рис. 4.11. Наращивание параллельной организации

Существующая организация команд для компонентов продолжает функционировать, как и прежде, за исключением того, что новые команды для функциональных средств будут вносить изменения в код, написанный командами для компонентов. Новые команды для функциональных средств берутся за разработку ценных, но трудных (из-за имеющихся зависимостей) функциональных средств, работая на границе между компонентами и внося в них непосредственные изменения. Напомним о необходимости искать добровольцев в рождающиеся команды для функциональных средств.

Такая стратегия носит постепенный характер, создает меньший риск и вполне пригодна для очень больших групп продуктов в LeSS Huge. Самый большой ее недостаток заключается в том, что она требует длительного времени.

Применяя данную стратегию, окажите полную поддержку молодым командам для функциональных средств, не ожидая от них большой отдачи. Ведь они должны преодолеть такие препятствия, как отличающиеся нормы практики в разных компонентах, разные структуры компонентов, разные применяемые инструментальные средства и

среды тестирования. Помимо этого, им придется овладевать навыками разработки новых компонентов и функциональных средств. Предоставьте им достаточно времени и окажите всяческую поддержку, поскольку они являются вестниками всех слабостей и нарушений в нормальной работе организации.

Наставление: организационная структура LeSS Huge

Масштабирование нередко сопровождается дополнительной организационной структурой. Прежде чем исследовать типичную в данном случае организационную структуру, следует подчеркнуть, что масштабирование совсем *не* означает потребность в дополнительной структуре. Ведь такая структура обычно сужает круг обязанностей, открывая путь для многих нарушений нормального функционирования и существующих правил в организации. Поэтому старайтесь сохранять простую организационную структуру.

С этой оговоркой организационная структура LeSS Huge строится на основании организационной структуры LeSS. Типичная схема организационной структуры LeSS Huge выглядит так, как показано на рис. 4.12.

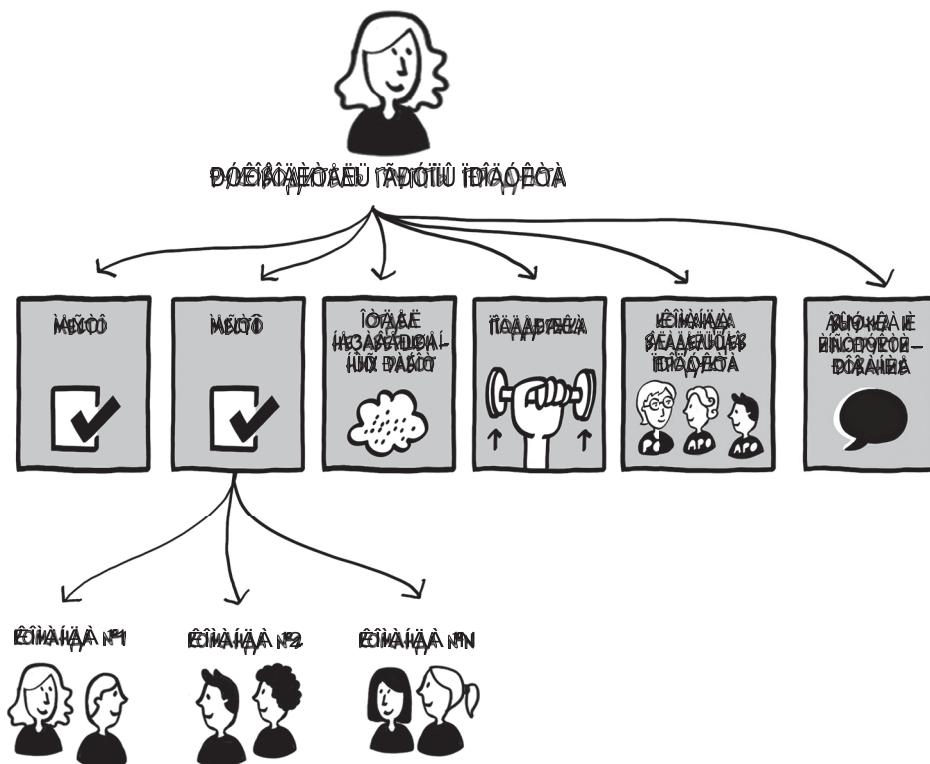


Рис. 4.12. Типичная организационная структура LeSS Huge

Обратите внимание на отсутствие проектно-программной организации либо отдела руководства проектом или программой. При внедрении Scrum и LeSS такие отделы просто перестают существовать. Рассмотрим те части, которые отличаются от организационной структуры LeSS.

Команды на местах. Внедрение LeSS Huge почти всегда происходит во многих местах, и организации обычно предпочитают сохранять локальной свою линейную организационную структуру. Поэтому старайтесь не приравнивать Области Требований к организационной структуре, поскольку это затрудняет изменения в них.

Команда Владельцев Продукта. Принципиально она такая же, как и в организационной структуре LeSS. Но эта команда крупнее, поскольку в нее входят Владельцы Продукта во всех Областях. В очень больших группах LeSS Huge каждая Команда Владельцев Продукта разделяется на подкоманды по Областям Требований.

Отдел незавершенных работ. И этот отдел принципиально такой же, как и в организационной структуре LeSS. В очень больших группах LeSS Huge отделы незавершенных работ зачастую оказываются более крупными, и поэтому для их упразднения требуется больше времени. Кроме того, отделы незавершенных работ в очень больших группах LeSS Huge будут иметь дополнительную структуру — возможно, даже с собственными устаревшими нормами практики управления проектами.

Поддержка. Этот отдел оказывает командам всяческую поддержку в организации среды разработки. Если в LeSS команды поддерживают друг друга без помощи отдельной группы, то в LeSS Huge, как правило, поддержка оказывается централизовано из-за очень большого объема работ. Тем не менее такой отдел должен быть как можно меньшим и работать по принципу “Чем можем помочь?”, а не по принципу “Делай так!”. Ведь группы поддержки нередко перебирают на себя обязанности команд и разростаются до огромных масштабов отвратительными подразделениями, контролирующими команды вместо их поддержки.

Поддержка в управлении конфигурацией служит типичным примером того, как группа поддержки превращается в группу *контроля*. Она берет на себя владение сборкой и создает все сценарии сборки. В итоге команды не имеют ни малейшего представления о том, что происходит, когда “сборка готова”, почему на это уходит 92 минуты и не чувствуют, что у них есть какие-то полномочия усовершенствовать сборку. Весь этот процесс оказывается для них в тайне и не контролируется ими.

Ограниченнное представление группы поддержки о сборке приводит к появлению узких мест, неэффективности, локальной оптимизации и лишению полномочий. Группа поддержки управления конфигурацией должна состоять из специалистов, которые помогают командам в усовершенствовании конфигурации, поясняют особенности сборки и обучают способам лучше собирать проекты, но при этом они *не* становятся их владельцами. Они могут работать в паре с членами команды, наблюдая за их действиями, чтобы совместно выработать пути для совершенствования. К числу других распространенных групп поддержки относятся группы лабораторного обеспечения, поддержки непрерывной системной интеграции и технологических процессов.

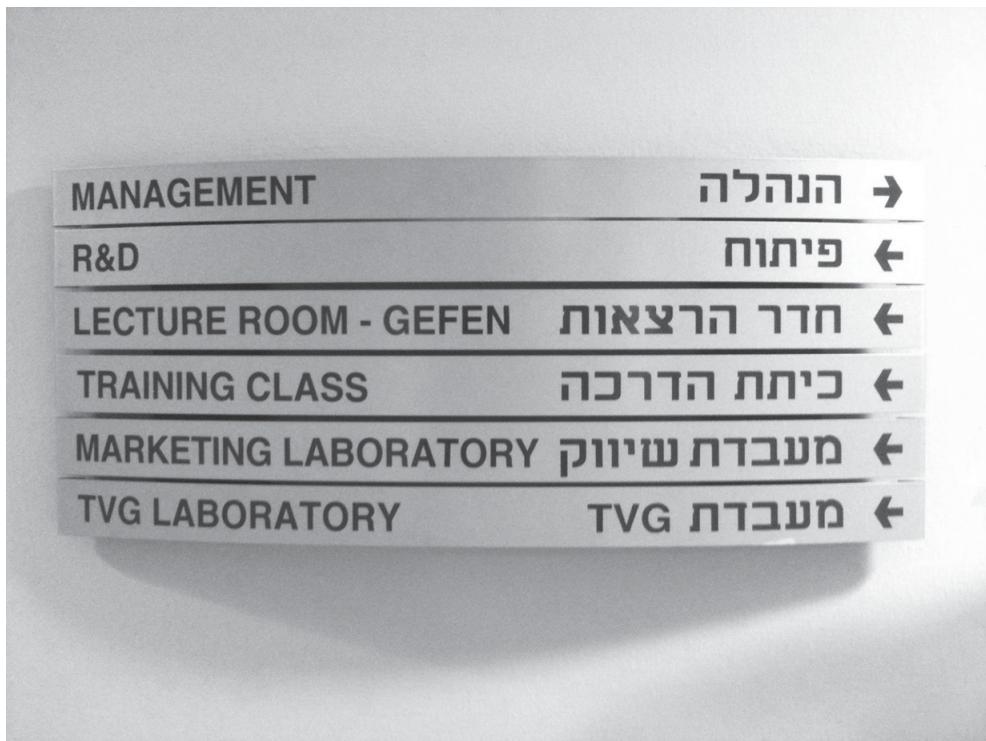
Выучка и инструктирование. Программное обеспечение создается людьми. Совершенствуясь сами, люди совершенствуют программные продукты. И хотя это кажется очевидным, нам редко приходилось видеть организации, действительно преданные делу неустанного обучения и инструктирования своих работников. Организации, практикующие LeSS Huge, выделяют целый отдел для обучения и инструктирования, что очень важно для непрерывного совершенствования.

Отдел обучения и инструктирования уделяет основное внимание следующему:

- > наблюдению (по принципу посещения рабочих мест);
- > обучению;
- > инструктированию.

В традиционных организациях запросы на обучение и инструктирование поступают от руководителей, которые даже не знают о существовании обучающей группы. И тогда они организуют оторванное от реальности обучение, напрасно тратя рабочее время подчиненных, что вряд ли можно приветствовать. Вместо этого обучающая и инструктирующая группа составляется из квалифицированных специалистов-практиков, активно посещающих рабочие места и наблюдающих за тем, как люди работают. Они работают в паре с людьми, чтобы выяснить их потребности в обучении. Самы люди не запрашивают обучение каким-то предметам, которых они не знают, или навыкам, о нехватке которых они не подозревают.

Особое место принадлежит инструктированию, поскольку это самый эффективный способ помочь командам совершенствоваться. Инструкторы работают с командами или в самих командах. Они наблюдают, изучают рабочую обстановку, работают в парах с членами команды и задают наводящие вопросы. Затем они делятся своими наблюдениями, отзывами и примерами, которым команда может следовать, чтобы совершенствоваться. Инструктирование происходит на трех уровнях: организационном, техническом, командном и Владельцев Продукта. Все эти уровни важны. Нам еще не приходилось наблюдать успешное внедрение LeSS без активного инструктирования.



Нарушение принципа разделения управления

Управление

Преобладает превратное мнение, будто хороший руководитель может быть таковым везде, не имея специальных знаний о производственном процессе, которым он управляет.
— У. Эдвардс Деминг

Scrum для одной команды

О руководителях в Scrum вообще не упоминается. Но Scrum изменяет стиль управления в большей степени, чем просто инфраструктура разработки. И это изменение обусловлено, главным образом, следующим элементами Scrum: самоуправляемыми командами, Владельцем Продукта и Scrum-мастером.

При наличии *самоуправляемой команды* ответственность Команды расширяется, включая управление процессом и контроль за его продвижением¹. Эта ответственность исключается из обязанностей руководителей.

Вся работа, выполняемая Командой, должна исходить от Владельца Продукта. Благодаря этому исключается обязанность руководителей выбирать Команды, с которыми они должны работать.

Scrum-мастер отвечает за нормальное функционирование Команды, Владельца Продукта и организации. Он разрешает конфликты, способствует совершенствованию, воспитывая стремление к размышлению и обучению, является наставником как для команды, так и для организации.

Традиционные обязанности руководителя сосредоточены на том, чтобы *следить* за тем, что и как делается. В целом все эти обязанности руководителей в организации, практикующей Scrum, больше не

¹ Команды в LeSS называются *самоуправляемыми*, а не *самоорганизующимися*. В литературе по Scrum оба эти термина употребляются попеременно. Самоуправляемые команды имеют ясно определение: команда, отвечающая за работу, управляющая процессом и контролирующая ход его продвижения. Это определение было дано его изобретателем, гарвардским профессором и исследователем команд Хакманом. С другой стороны, термин *самоорганизующиеся команды* нередко употребляется неоднозначно или несогласованно.

требуются. Таким образом, на смену командного и контролирующего стиля управления приходит *поддерживающий*.

Типичное затруднение при внедрении Scrum возникает в том случае, если руководители не отдают добровольно свои обязанности. Это приводит к конфликту между Командой, Владельцем Продукта, Scrum-мастером и руководством организации.

Какова в таком случае роль руководителей в организации, практикующей Scrum? Ответ на этот вопрос в Scrum не дается, но организациям предлагается самим найти его. А в LeSS предоставляется возможность начать нелегкое обсуждение изменений в роли руководителей организации, практикующей LeSS.

Управление в LeSS

Инфраструктура LeSS следует традиционной теории организации труда. Если требуется повысить организационную гибкость, для этого следует делегировать обязанности таким образом, чтобы принятие решений не замедляло ответную реакцию. Это приводит к совершенствованию организационной структуры и сокращению количества руководителей. Но внедрение LeSS в большинстве организаций происходит без сокращения количества руководителей. Так в чем же их роль? При масштабировании соблюдаются следующие принципы, связанные с управлением.

- > **Управление эмпирическим процессом.** Владеть тем, как работа должна быть сделана, должны те, кто ее выполняет. При этом они испытывают ответную реакцию и совершенствуются. Каким же образом переход владения процессом к командам изменяет управление?
- > **Ориентация на клиентов.** Основное внимание команд, работающих непосредственно с клиентами, сосредоточено на клиентах, что делает их работу более полноценной. А руководители больше не сотрудничают с клиентами и не выступают в качестве посредников.
- > **Непрерывное улучшение до полного совершенства.** Когда руководителям больше не нужно осуществлять ежедневное управление, они могут уделить больше внимания улучшению системы.
- > **Системный подход.** Организационная структура до внедрения LeSS нередко вызывала отщельническое умонастроение, которое должно сменить взгляды на всю систему и продукт в целом. Но смена взглядов — зачастую дело неизвестное, неудобное и требует значительного обучения.

Правила LeSS

Наличие руководителей в LeSS необязательно. Но если руководители все же присутствуют, то их роль, скорее всего, изменится. Их основное внимание переносится с ежедневного управления работой над продуктом на улучшение возможностей доставлять ценность в системе разработки продукции.

Роль руководителей заключается в совершенствовании системы разработки продукции. С этой целью они практикуют посещение рабочих мест, поощряют оперативное устранение недостатков и “экспериментирование” превыше соответствия”.

Наставление: представление о теориях Тейлора и Файоля

Управление — придуманное понятие. Очень важно знать его происхождение и контекст, чтобы приспособить и сделать его пригодным в нынешних условиях. Похожи ли задачи, которые оно некогда решало, на задачи, которые приходится решать сегодня? Без критического анализа и глубокого понимания этого вопроса непрерывное совершенствование невозможно, хотя эта непрерывность тянется еще с XIX века. Двумя ранними авторитетами в области управления являются Фредерик Тейлор и Анри Файоль.

Фредерик Тейлор родился в 1856 году и, работая инженером-механиком, увлекся проблемами производительности труда рабочих. Как мастер участка он успешно применял принципы научной организации труда своих рабочих. В итоге он открыл свою консультационную фирму, а его идеи послужили основанием для теории научной организации труда².

Анри Файоль — французский горный инженер. Он родился в 1841 году и с 19 лет работал во французской горнодобывающей компании. Его первой задачей стало повышение безопасности труда в горнодобывающей отрасли. Он никогда не покидал свою компанию и в конечном итоге стал ее управляющим директором. Под руководством Файоля компания расцвела и стала одной из самых крупных во Франции. Свои мысли об управлении он изложил и опубликовал в фундаментальном исследовании “Общее и промышленное управление”³.

Фредерик Тейлор ввел два следующих понятия, которые, к сожалению, господствуют до сих пор.

- > Имеется единственный самый лучший способ сделать работу, и его можно усовершенствовать научным путем. Как только такая норма “передовой практики” будет обнаружена, ее следует внедрить в организации.

² Называется также тейлоризмом.

³ Называется также файолизмом.

- > Работа по планированию и улучшению должна быть отделена от обычной работы. Работа по планированию и улучшению должна выполняться специально обученными высокообразованными людьми, тогда как обычная работа — главным образом, малообразованными. По мнению Тейлора: “Стоимость производства, несомненно, снижается благодаря тому, что работа по планированию и умственный труд как можно в большей степени отделяются от ручного труда”⁴.

Анри Файоль сформулировал 14 принципов управления, включая разделение труда, властные полномочия, единство руководства и цепочку команд. Он определил также пять обязанностей руководителей: планирование, организация, координирование, начальствование и контроль.

Корни многих так называемых “современных” теорий управления надо искать в идеях Тейлора и Файоля⁵. Именно они изменили работу компаний в частности и мир в целом.

Но современный мир сильно изменился со времен Тейлора и Файоля. В силу такого различия в контексте самые лучшие идеи прошлого становятся самыми худшими идеями настоящего. Ниже приведены характерные тому примеры.

- > Тейлор стремился в максимальной степени повысить производительность труда малообразованной рабочей силы. Но современные разработчики продуктов являются высокообразованными работниками интеллектуального труда. Отделение планирования и улучшения от производства приводит к дополнительной передаче работы между подразделениями, жесткой специализации и увеличению издержек.
- > Файоль стремился укрепить единство, совершая сообщение, поскольку для поездки из Франции в Соединенные Штаты требовалось 10 дней. Ныне на такую поездку уходит меньше 7 часов, а на сообщение — считанные секунды. Обширные иерархии, призванные укрепить единство и упростить сообщение, безнадежно устарели.
- > Научный анализ с целью выявить норму передовой практики и скопировать ее может пригодиться разве что при перемещении чугунных чушек в литейном цехе, как это делал Тейлор. И хотя идея применить научный анализ труда сама по себе замечательна, копирование без контекста никуда не годится. Не менее замечательна идея копировать нормы надлежащей практики, но копирование норм *передовой* практики противоречит непрерывному совершенствованию.

⁴ Цитата из труда “Управление фабрикой”, опубликованного Фредериком Уинслоу Тейлором в 1903 году.

⁵ К числу других солидных авторитетов в области управления относятся Макс Вебер и Мэри Паркер Фоллетт. И хотя их труды здесь не упоминаются, тем не менее, они вполне заслуживают изучения.

- > Централизованное руководство, укрепляющее *единство* путем планирования, координирования, начальствования и контроля, может оказаться вполне работоспособным после оптимизации в горном деле. Идея создания единства или концепции в организации сама по себе замечательна, но только не идея централизации планирования и контроля. Акцент на отдаче приказов и контроле приводит к ослаблению внимания на систематических улучшениях.

Изучите структуру, нормы практики и правила, принятые в вашей организации. Выясните, сколько их применяется только потому, что так было всегда. Откуда исходят эти мысли и действительно ли они отвечают современному контексту вашей организации.

Наставление: теории управления

Вспомним 1960 год, когда был изобретен лазер и оральный контрацептив. Через год была построена Берлинская стена, а через два года на экраны вышел первый фильм о Джеймсе Бонде. С тех пор вышло больше двадцати подобных фильмов и многое изменилось! А может быть, и нет?

В 1960 году Дуглас Макгрегор из школы управления имени Слоуна при Массачусетском технологическом институте опубликовал свой фундаментальный труд “Человеческая сторона предприятия” по теории управления. В этом труде он исследовал причины, по которым человеческий потенциал не полностью используется в организациях, и пришел к выводу, что большинство современных ему теорий и норм практики управления основывались на ряде базовых *неисследованных допущений*, которые он называл Теорией X. Подобные допущения о социальном поведении человека ограничивают нормы практики, модели и поведение при управлении, чтобы по-настоящему воспользоваться человеческим потенциалом.

Теория X

Теория X управления основывается на следующих допущениях.

- > Люди не любят работать и стараются отлынивать от работы.
- > Следовательно, людей нужно принуждать, контролировать, направлять и даже запугивать, чтобы извлечь максимум из их труда.
- > Людьми нужно руководить, поскольку они проявляют мало честолюбия и избегают ответственности.

И хотя об этих допущения мало говорят так просто и непосредственно, тем не менее, они до сих пор образуют основание многих (если не большинства) норм практики! Группы управления кадрами поддерживают стремление использовать труд наемных работников. Но, по иронии судьбы, большинство таких норм практики управления кадрами, как аттестация, достижение индивидуальных целей и системы поощрений, прочно опираются на допущения из Теории X. И в этом нет ничего удивительного! Какие в конечном счете допущения скрываются за термином *управление кадрами*?

Теория Y

Чтобы извлечь максимум из человеческого потенциала, придется изменить свои представления и нормы практики, опирающиеся на Теорию X, приняв допущения, основанные на выводах, сделанных в социальных науках. Это следующие допущения из Теории Y.

- > Люди затрачивают усилия, чтобы работать так же *естественно*, как они привыкли играть и отдыхать.
- > Люди прибегают к самоуправлению и самоконтролю, чтобы достичь целей, которых они обязались добиться⁶. Обязательство исходит в основном из внутреннего вознаграждения, связанного с самодостижением. Это означает вызов, обучение и целеустремленность.
- > При наличии подходящей среды люди ищут, а не избегают ответственности. Воображение, находчивость и творческие способности являются навыками, которыми обладают все люди.

Генри Форду приписывают следующий вопрос: “Почему, когда мне требуется только пара рук, мне дают в придачу еще и мозги?” Вероятно, Форд находился под влиянием теории научной организации труда и Теории X.

“Хорошие идеи — качественная продукция” — такой лозунг начертан на стене фабрики компании Toyota, где под сокращением TPS подразумевают не Toyota Production System (Производственная система компании Toyota), а *Thinking People System* (Система думающих людей). В компании Toyota сформировали основание для бережливого производства, и это было сделано под влиянием Теории Y.

Вся гибкая разработка, Scrum и LeSS также основываются на Теории Y. И объясняется это следующими причинами.

1. **Практическое применение Теории X затрудняет внедрение LeSS.** Большинство организаций просто насыщены нормами практического применения Теории X, где основное внимание уделяется индивидуально подотчетности и контролю со стороны руководства. А в организации, внедряющей LeSS, такой подход следует изменить на подотчетность команде и самоконтроль.
2. **Допущения из Теории X трудно изменить.** LeSS требует изменить стиль управления, т.е. манеру поведения и допущения руководителей. А для того чтобы изменить эти допущения, требуется переосмыслить *весь* предыдущий, а не только трудовой опыт. Культурные и семейные представления о работе особенно глубоко укоренены и с трудом поддаются изменению.

⁶ Понятие “люди обязываются” в Теории Y противоположно понятию “людей обязывают” в Теории X.

Многие трудности внедрения LeSS можно объяснить попытками применить Теорию Y управления на практике, не отказываясь от допущений из Теории X.

Итак, теперь время для аттестации! Много ли изменилось с 1960 года?

Наставление: руководители не обязательны

Руководители в LeSS не обязательны. В тех организациях, где имеются руководители, совсем не обязательно избавляться от них. Ведь они могут выполнять полезную роль. Но *вводить* руководителей для внедрения LeSS не стоит.

В мире наметилась важная тенденция: появление компаний *без* руководителей. В таких компаниях пересматривают допущения, лежащие в основе управления, выясняя те задачи, которые оно призвано решать, а также намечая пути разделения обязанностей и распределения власти⁷. Подобные эксперименты служат отличным источником для новых идей, нововведений и вдохновения, даже если в компании остаются руководители.

В большинстве крупных компаний нет недостатка в руководящих ролях и должностях. Эти роли нужно подвергнуть критическому анализу при внедрении LeSS. В организации, практикующей LeSS, предпочтение отдается переносу ответственности на команды вместо того, чтобы возлагать ее на руководящие роли.

Почему же многие компании насыщены руководителями? А потому, что в них принят стандартный подход к разрешению организационных проблем, который состоит в следующем.

1. Выявить проблему — пустяковую.
2. Создать новую роль — пустякового руководителя.
3. Назначить новую роль для решения проблемы.

Пустяковые руководители изобилуют в большинстве организаций! Примером тому служат ответственные за устранение отказов (поскольку имеются программные ошибки), проблемные руководители (ответственные за разрешение проблем), ответственные за функциональные средства (и координирование проблем), ответственные за качество (и проблемы качества) и т.д. В ведущих организациях имеются пустяковые отделы, начальники которых руководят пустяковыми руководителями и специалистами в течение всей их пустяковой карьеры.

См. раздел “Наставление: безопасность труда, а не роли” главы 3.

См. далее раздел “Наставление: список рекомендованной литературы по управлению”

⁷ К числу других солидных авторитетов в области управления относятся Макс Вебер и Мэри Паркер Фоллетт. И хотя их труды здесь не упоминаются, тем не менее, они вполне заслуживают изучения.

Ничего подобного не должно быть в организациях, практикующих LeSS, благодаря следующему.

- > **Системный подход.** Многие пустяковые проблемы являются системными (например, динамика развития команд для компонентов), и поэтому назначить роль для их решения, не изменяя систему, — значит упрощенно разрешить проблему на скорую руку. Истинное представление о динамике развития систем позволяет вносить в них изменения без дополнительных ролей.
- > **Организация на основе команд.** Для разрешения некоторых проблем действительно требуется назначить новую роль (например, для координирования действий со сторонней организацией). Решение подобных проблем лучше поручать обычным командам для функциональных средств, а не создавать дополнительные роли. Благодаря этому, во-первых, люди привлекаются к работе над улучшениями, во-вторых, улучшения основываются на реальности и, в-третьих, структура организаций упрощается, поскольку отсутствуют дополнительные роли.

Таким образом, руководители могут быть полезны. Но каковы их обязанности в организации, практикующей LeSS? Этот вопрос рассматривается в следующем наставлении.

Наставление: организация, практикующая LeSS

При бережливом подходе акцент делается на *гембе*. Этим японским термином обозначается реальное рабочее место, где создается потребительская ценность. Имеются следующие разновидности гембы.

- > Место, где применяется продукт. Это гемба потребления ценности.
- > Место, где создается продукт. Это гемба создания ценности.

В организации, практикующей LeSS, эти две разновидности гембы должны быть расположены как можно ближе друг к другу. Требования поступают потоком от пользователей к командам и Владельцу Продукта, а ценность — обратным потоком от команд к пользователям. Доставка ценности должна проходить через всю организацию без проталкивания вверх по иерархии.

Руководители не участвуют в принятии решений, связанных с доставкой ценности или направлением продукции. Так что же они делают? В организации, практикующей LeSS, они уделяют внимание системе разработки и, в частности, расширению возможностей организации доставлять ценность. *Их дело — улучшение!*

И хотя руководители могли бы способствовать улучшениям, инструктируя команды и помогая людям расти профессионально, сами они *не* выполняют работу по улучшению, поскольку это — прямая дорога назад к тейлоризму. Ведь нам не нужны работы, бездумно выполняющие технологические процессы, а требуется извлечь максимум из человеческого потенциала. Следовательно, руководители уделяют основное внимание

улучшениям в организации. В частности, они занимаются улучшениями системы разработки, которые нередко охватывают организационную структуру, решения и правила. Структура организации, практикующей LeSS, наглядно показана на рис. 5.1.

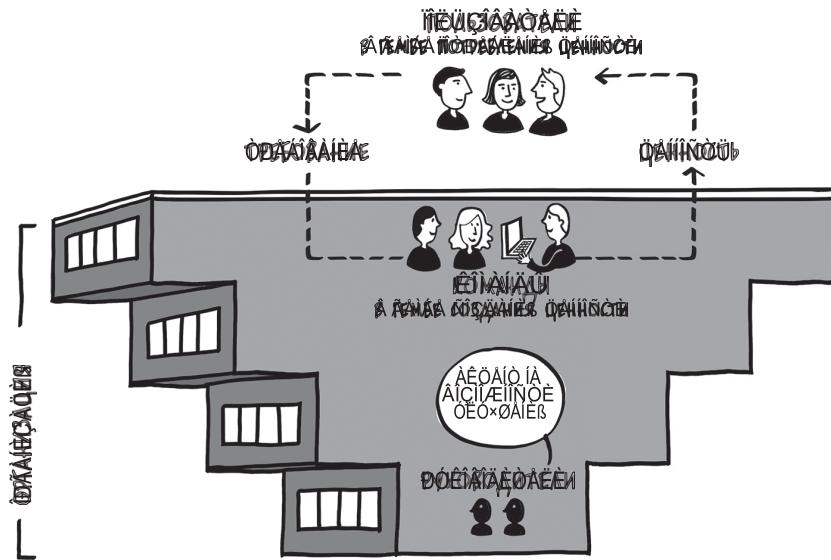


Рис. 5.1. Общая структура организации, практикующей LeSS

Разные роли требуют разного сосредоточения. Ниже перечислены три основные сферы деятельности для сосредоточения основного внимания.

- > Создание и доставка продукта.
- > Концепция и направление продукта.
- > Организационные возможности для улучшения.

Было бы ошибкой думать, что каждая роль точно соответствует одной сфере деятельности. На самом деле они *должны* перекрываться, поскольку роли действуют совместно. На рис. 5.2 наглядно показаны роли и обязанности в трех основных сферах деятельности.

Рассмотрим подробнее перекрывающиеся сферы деятельности.

- > **Команды – Владелец Продукта.** Владелец Продукта определяет направление или концепцию продукта, а команды должны принимать в этом участие. Команды должны *владеть* продуктом в такой же степени, как и Владелец Продукта. Это их продукт, и поэтому они тесно сотрудничают с пользователями, оказывая содействие Владельцу Продукта. В частности, команды вводят элементы в Задел Продукта и обсуждают с Владельцем Продукта назначение приоритетов.

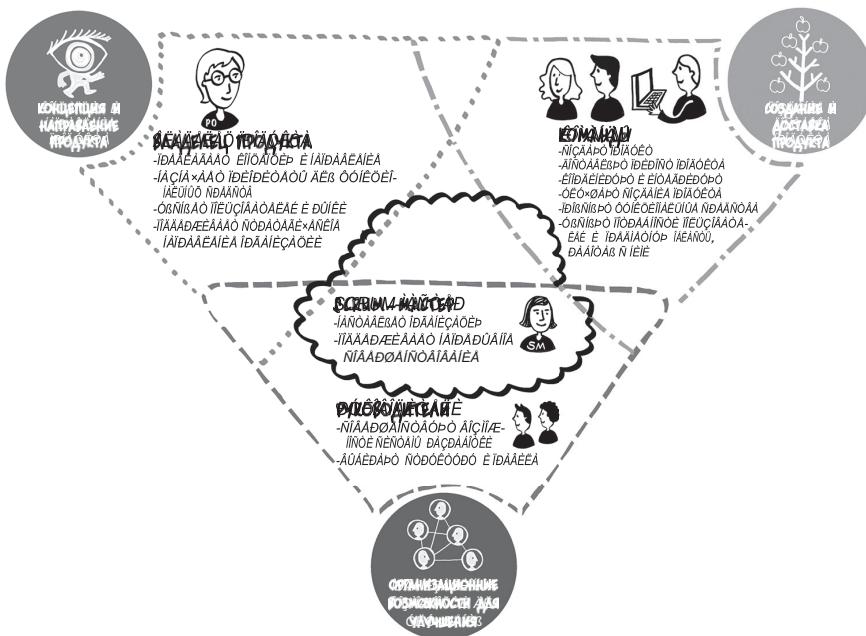


Рис. 5.2. Роли и обязанности в трех основных сферах деятельности

См. далее раздел
“Наставление:
посещение рабочих
мест”.

- > **Команды – Руководители.** Команды осуществляют улучшения, тогда как руководители уделяют основное внимание *возможности улучшения* и поддержке необходимых организационных изменений. Для улучшений нередко требуется изменение организационной структуры или правил, и зачастую команды не могут справиться с этим сами. Поэтому командам нужно работать вместе со Scrum-мастерами и руководителями над внедрением таких изменений. Например, команды совершенствуют автоматическое развертывание, тогда как руководители изменяют нормативные и организационные правила, связанные с развертыванием.
- > **Руководители — Scrum-мастера.** Руководители и Scrum-мастера уделяют основное внимание улучшениям и поэтому должны работать вместе. Если руководители сосредоточены в большей степени на организационных моментах, то Scrum-мастера — на динамике развития команд и отношений между ними. Например, Scrum-мастера обнаруживают и объясняют потребность в расширении межфункциональности команд, тогда как руководители вносят соответствующие организационные изменения, упраздняя, например, группы тестирования.

> **Руководители — Владелец Продукта.** Руководители нижнего звена мало пересекаются с Владельцем Продукта, поскольку они уделяют основное внимание *улучшению*. Вероятно, они могли бы побуждать команды вносить предложения по улучшению в Задел Продукта. Старшие руководители вырабатывают стратегическую перспективу, охватывающую многие продукты. Они должны тесно сотрудничать с Владельцами Продуктов над выбором *подходящих продуктов*, а с руководителями нижнего звена и Scrum-мастерами — над *подходящими улучшениями*. Например, работая вместе со всеми Владельцами Продуктов, они выявляют новые рынки для освоения и делают соответствующие организационные выводы.

См. раздел “Наставление: назначение продукта” главы 7.

В предыдущем разделе упоминалось сосредоточение основного внимания на продукте вместо организационных моментов. Выдвижение ролей в таких масштабах создает еще одно представление о ролях в LeSS, как показано на рис. 5.3.

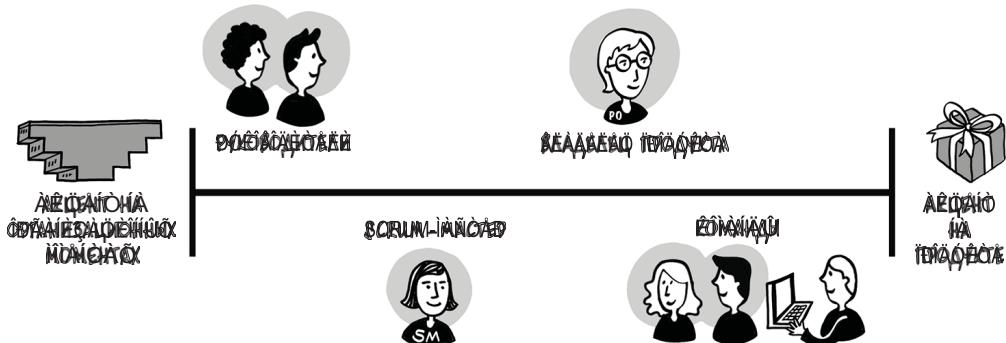


Рис. 5.3. Сосредоточение ролей в LeSS на продукте и организационных моментах

На этом рисунке наглядно показано, что основное внимание руководителей и Scrum-мастеров несколько отличается. Если руководители сосредоточены на организационных моментах, то Scrum-мастера — на командах и продуктах.

Наблюдение за сходством функций Scrum-мастеров и руководителей наводит на следующий вопрос: не следует ли сделать руководителей Scrum-мастерами? Нет, не следует, поскольку это неудачная идея. Если Scrum-мастер приобретет положение руководителя, это воспрепятствует самоуправлению в команде, даже если он является Scrum-мастером другой команды. Ведь у членов команды имеются глубоко укоренившиеся представления о руководителях, которые неявно изменяют их поведение. А этого лучше всего избежать, назначая меньше руководящих ролей.

И, наконец, что конкретно означает сосредоточение основного внимания на расширении возможностей системы разработки? В качестве ответа на этот вопрос далее рассматриваются следующие нормы практики.

- > Посещение рабочих мест.
- > Руководители не только учат, но и сами учатся.
- > Раскрытие возможностей как предметной области, так и технических.

Наставление: посещение рабочих мест

Посещение рабочих мест является едва ли не самым важным навыком руководителей в LeSS. На первый взгляд, такая норма практики кажется довольно простой: нужно посетить гембу — реальное рабочее место, чтобы увидеть, что происходит в действительности. Но на практике эту норму трудно понять и усвоить.

Рассмотрим сначала, чем посещение рабочих мест на самом деле не является. Это не мелочная опека на уровне микроуправления. Микроуправление можно описать таким образом: “пойти посмотреть, вмешаться и исчезнуть”. И в лучшем случае такая практика расхолаживает работников. Если подойти к посещению рабочих мест с точки зрения традиционного управления и контроля, то оно, вероятнее всего, превратится в микроуправление. Такого подхода следует всячески избегать.

Посещение рабочих мест, практикуемое руководителями, означает, что они должны выработать в себе привычку посещать гембу⁸, чтобы лучше понять реальные проблемы и на этом основании выработать решение о расширении возможностей организации. А что такое гемба? Для разработки продукции имеются следующие разновидности гембы.

- > Гемба создания ценности — команды, создающие продукт.
- > Гемба потребления ценности — пользователи, применяющие продукт.

Оба места следует посещать регулярно, чтобы получить ясное представление о реальном положении дел в гембе. Посещение рабочих мест преследует две важные цели:

- > расширить возможности решения проблем;
- > улучшить организационные решения.

См. далее раздел “Наставление: руководители в качестве учителей и учащихся”.

Расширение возможностей решения проблем. Исследуя гембу и вникая в повседневный труд команд, можно ясно понять проблемы, с которыми они сталкиваются. Но не следует спешить их разрешать, как бы этого ни хотелось! Руководителю в LeSS нужно, чтобы команды *сами* разрешили свои проблемы.

Если они не в состоянии этого сделать, то главная роль руководителя — научить команды и поспособствовать тому, чтобы они разрешили свои проблемы. Иными словами, руководитель должен расширить возможности команд решать свои проблемы.

Улучшение организационных решений. Трудности и проблемы, возникающие в гембе, можно разделить на следующие категории.

⁸Часто, а не в виде исключения.

1. Контекстные, характерные для команды.
2. Обусловленные решениями, принимаемыми за пределами команды.

В последнем случае проблемы обусловлены организационной структурой, принимаемыми решениями и принятыми правилами, причем они одинаковы для всех команд. Следовательно, руководитель посещает рабочие места в организации, практикующей LeSS, чтобы ясно понять рабочий контекст нескольких команд и получить ответную реакцию на важные управленческие решения. Такая ответная реакция, поступающая из реальных рабочих мест, способствует принятию более взвешенных организационных решений на основании реального положения дел.

Чем старше руководитель, тем важнее для него подобная норма практики, поскольку он, во-первых, дальше отстранен от гембы, и чтобы быть в курсе дел, ему приходится прилагать дополнительные усилия; а, во-вторых, принимаемые им решения имеют более серьезные последствия. Не посещая рабочие места, старшие руководители, вероятнее всего, будут принимать решения, оторванные от реального положения дел в гембах. Такие решения могут оказаться опасными и в конечном итоге способны привести к краху организации. Знакомая ситуация, не так ли?

Норма практики посещения рабочих мест внедряется с трудом. Ниже перечислены соответствующие причины.

Нехватка времени. Многие руководители, очевидно, не контролируют свое время, но большинство из них способны быстро реагировать. Их рабочие календари заполнены совещаниями, которые им трудно отменить. Их одолевают поручения провести всевозможные мероприятия. А кроме того, что может быть приятнее, чем ощущение, когда вовремя потушен очередной пожар? С другой стороны, практика посещения рабочих мест означает намерение зарезервировать для этой цели достаточно времени, разорвав бесконечный круг совещаний и обязательных к исполнению мероприятий.

Недопонимание. Посетить рабочее место — совсем не означает выразить сочувствие работникам или проверить, как продвигается дело. Это делается для того, чтобы ясно понять трудности, возникающие в гембе. Для этого недостаточно пообщаться с командой или пригласить ее на совещание, поскольку это дает лишь поверхностное представление об истинном положении дел. При посещении рабочих мест нужно наблюдать за командами и задать им много откровенных вопросов, проявив искренний интерес к их работе и нуждам. Если речь идет о разработке программных продуктов, то руководителю, скорее всего, придется просмотреть и обсудить исходный код.

Нетерпение. Многие руководители умеют отлично разрешать сложные проблемы, а следовательно, они могут разрешить и проблемы, которые были выявлены и уяснены при посещении рабочих мест. Но именно этого и нельзя делать! Команды должны совершенствоваться, сами решая свои проблемы на рабочих местах. Им не нужен руководитель, мимоходом разрешающий их проблемы. Но что, если команда не в состоянии разрешить затруднение, поскольку оно носит внешний, организационный и систематический характер? В таком случае нужно принять дальнейшие меры.

Отсутствие анализа. Уяснив проблемы в гембе, следует выяснить их причины: командные и контекстные или организационные и системные, хотя сделать это нелегко.

Многие руководители приписывают большинство проблем контексту, не распознавая их характер. Им безопаснее отнести проблемы к разряду контекстных, поскольку они не требуют от них никаких действий. Иные руководители усматривают во всем организационные причины и в конечном итоге принимают ужасные решения, исходя из ответной реакции и контекста только одной команды. Но выявить подлинные причины не так-то просто.

Посещение рабочих мест требует практики и еще раз практики! Оно требует также непредубежденности и естественного любопытства, чтобы уяснить во всех подробностях творческий характер работы над созданием продуктов.

Наставление: руководители в качестве учителей и учащихся

Зачастую встречаются две типичные разновидности руководителей.

- > **Формальные руководители**, т.е. такие, которые перестали учиться, как только стали руководить. Такие руководители не читают профессиональную литературу, не посещают никаких курсов и не смотрят никаких фильмов, кроме занимательных. Все свое рабочее время они занимаются решением таких административных задач, как составление отчетов или аттестация работников. В итоге они превращаются в бесполезных административных роботов.
- > **Профессиональные руководители**, т.е. такие, которые *не* перестали учиться, но читают только популярную литературу по управлению⁹. Тем не менее они полностью утратили представление о реальной работе над созданием продуктов. А зачем им вообще иметь такое представление, если они считают, что профессиональный руководитель способен руководить всем, чем угодно, — даже тем, чего он вообще не понимает. И это настолько глубокое заблуждение, что мы предпочли бы формальных руководителей — по крайней мере, вреда от них меньше.

В организации, практикующей LeSS, руководители постоянно *чему-то* учатся. Они в курсе новейших технологий и предметной области, чтобы понимать современную, а не ту реальность, которая существовала, когда они стали руководить. Очевидно, такое понимание требуется для посещения рабочих мест, но не только.

Руководители в LeSS соблюдают норму практики бережливого подхода, которая предполагает, что руководители должны действовать в качестве *учителей*. Это совсем не означает, что руководители должны быть самыми лучшими техническими специалистами и знатоками предметной области. Напротив, у них должно быть ясное представление как о предметной области, так и о текущих технических навыках, чтобы пользоваться им для обучения и наставления команд совершенствовать и расширять свои возможности в области разработки.

⁹ Иногда такие книги называют “аэропортовыми”. Многие шумные организационные кампании можно объяснить наличием большого выбора популярной литературы по управлению в залах ожидания аэропортов.

Как руководителю оставаться на современном уровне? С этой целью он может предпринять следующее:

- > периодически устранять ошибки вместо команды;
- > пользоваться продуктом и тестировать его;
- > участвовать в анализе исходного кода, не обязательно комментируя его;
- > посещать пользователей и наблюдать за ними;
- > реорганизовывать какой-нибудь код, а возможно, и отвергать его;
- > просматривать периодические издания в предметной области;
- > работать в паре с членами команды над выполнением какой-нибудь работы;
- > автоматизировать что-нибудь;
- > присоединиться к сообществу.

Национальная культура. В некоторых странах практиковать роль руководителей в качестве учителей намного труднее, поскольку для этого требуется намного больше усилий, чем в других странах. В одних национальных культурах делается более естественный акцент на технике и традиции грамотных руководителей, тогда как в других национальных культурах руководители редко соприкасаются с техникой, поскольку это уже пройденный этап в их карьере.

Наставление: возможности как предметной области, так и технические

Командам необходимо найти золотую середину между техническими навыками и пониманием предметной области. Нам часто приходилось наблюдать команды с превосходными техническими навыками, но недостаточными знаниями предметной области, и наоборот. Еще хуже дело обстоит в тех организациях, у которых имеется исторический опыт передачи работ по программированию на внешний подряд¹⁰.

В организации, практикующей LeSS, руководители должны регулярно оценивать навыки команд, чтобы определить, где именно следует сосредоточить свои усилия на расширении возможностей. Типичной ошибкой в данном случае является переоценка навыков, которыми обладает сам руководитель, а также недооценка чужих навыков. Страйтесь не попадать в эту ловушку.

При внедрении LeSS Huge обычно организуется несколько мест разработки. В этом случае для мест разработки характерна специализация в одной области, но не

См. раздел “Организация по потребительской ценности в LeSS” главы 4.

¹⁰ Прожив какое-то время в Азии (Крейг — в Индии, а Бас — в Китае и Сингапуре) и опираясь на немалый опыт программирования и работы в гембе, мы считаем идею перенесения работ по программированию за границу ошибочной и вообще неработоспособной. Люди, с которыми нам приходилось встречаться и которые утверждали, что эта идея вполне работоспособна, на самом деле не уделяли времени посещению гембы, чтобы ознакомиться с реальным положением дел.

в другой. Поэтому места разработки следует оценить с этой точки зрения и принять соответствующие меры.

Каковы эти меры? Диспропорция зачастую обусловлена недостаточными инвестициями в обучение в какой-то одной области. Как правило, это происходит из-за недооценки данной области. Например, в одном месте, где нам пришлось работать, См. раздел “Наставление: сообщества” главы 13. у разработчиков были отличные знания предметной области, но поразительно слабые технические навыки, поскольку в этом месте сложились такие культурные представления, будто программиривать может всякий.

Ниже приведены некоторые примеры надлежащих в данном случае мер.

- > Повысить осознание имеющейся диспропорции, обсудить ее с руководителями, Scrum-мастерами и командами.
- > Организовать обучение и инструктирование.
- > Организовать обмен опытом и примерами между командами.
- > Содействовать сообществам в обучении.
- > Поощрять команды выбирать работу, сокращающую диспропорцию.

Качественные продукты создаются только в тех командах, в которых соблюдается пропорция между техническими навыками и знанием предметной области и которые работают непосредственно с пользователями, разрешая их реальные проблемы.

Наставление: количественные показатели и цели в LeSS

У руководства, участвующего во внедрении LeSS, обычно возникает следующий вопрос: “Как нам это измерить?” Это любопытный, интригующий, но неверный вопрос, потому что он предполагает, что измеряемые показатели могут оказаться плохими или хорошими. Казалось бы, достаточно найти подходящие количественные показатели, установить правильные цели, и все пойдет нормально, но на самом деле это не так.

Количественные показатели сами по себе не важны. Намного важнее, во-первых, их *назначение*, а во-вторых, *кто* их устанавливает.

Нашим любимым примером является тестовое покрытие. Насколько это хороший или плохой количественный показатель? Вопрос неправомерный. Если тестовое покрытие задается руководителями как организационная цель, а еще хуже — как цель индивидуальной работы, то можно гарантировать вредное поведение. Безусловно, нам приходилось наблюдать крайние случаи дублирования и саботирования измерений, но мы предпочитаем тесты без проверок их непрохождения. И такой подход примечателен тем, что он позволяет достичь максимального тестового покрытия и сокращает затраты труда на сопровождение. Не достигают ли при этом работники своих целей, тогда как сократить затраты труда — мечта руководства?

Но если командам потребуется усовершенствовать свои возможности автоматизировать тестирование и измерять тестовое покрытие, чтобы узнать больше, то отлично! Вероятнее всего, это приведет к лучшему пониманию, усовершенствованиям, большему воздействию и владению. Но здесь речь не шла о количественных показателях.

Количественные показатели являются полезными инструментальными средствами, но, применяя их, необходимо избегать следующих ошибок:

- > иметь цели, не проясняя назначение;
- > ставить цели для команд;
- > измерять ради контроля;
- > измерять что-нибудь, не зная зачем;
- > наносить другим ущерб ради того, чтобы измерить.

В общем, необходимо иметь в виду следующее:

Основное внимание должно уделяться назначению, а не целям.

Цели без назначения — это не административно-командное управление, а диктатура.

Людей, пользующихся количественными показателями, необходимо учить устанавливать свои количественные показатели. Этим гарантируется правильное понимание количественных показателей и их назначения, а также исключается напрасный труд для достижения произвольной цели.

Наставление: список рекомендованной литературы по управлению

Руководителям организаций, практикующей LeSS, нужно постоянно изучать свою предметную область и технологию. Им нужно быть также в курсе современных тенденций в области управления. А для этого придется немало учиться, и поэтому ниже приводится список самой важной, на наш взгляд, литературы по управлению.

- > *Fifth Discipline*, Peter Senge. Классический труд по созданию обучающихся организаций и системному подходу. На наш взгляд, с ним должны ознакомиться руководители организаций, практикующих LeSS.
- > *Lean Manager* и *Lead with Respect*, Michael и Freddy Balle. Обе эти книги написаны в форме делового романа, в котором повествуется о студенте Энди, который изучает бережливое управление и которому требуется перейти от традиционного управления к бережливому. Норма практики посещения рабочих мест особенно удачно описана в книге *Lean Manager*.

- > *Workplace Management*, Taiichi Ohno. Тайити Оно является создателем производственной системы в компании Toyota, а его книга *Workplace Management* — классический труд по бережливому подходу и управлению. Особенno примечателен его подход к проблемам и сосредоточию основного внимания на посещении рабочих мест.
- > *Future of Management*, Gary Hamel. Нужны ли руководители? Гэри Хэмель считает, что нужны, но стиль руководства в будущем должен определенно измениться. В этом классическом труде поясняется, как это произойдет.
- > *Hard Facts, Dangerous Half-Truths & Total Nonsense: Profiting from Evidence Based Management*, Jeffrey Pfeffer и Robert Sutton. Если лишенные контекста нормы передовой практики создают вредную иллюзию, то это еще не означает, что мы не должны перенимать идеи друг у друга. Тем не менее слишком многие идеи, основанные на последних достижениях в области управления, были почерпнуты из “аэропортовых” книг. Авторы книги, Пфеффер и Саттон, подводят прочное научное основание под ответственныеправленческие решения.
- > *Reinventing Organizations*, Frederic Laloux. Действительно ли нужны руководители? Фредерик Лалу исследует в своей книге деятельность современных компаний, отказавшихся от традиционного управления. Они полностью организованы по принципам самоуправления, и зачастую в них совершенно упразднена роль руководителя. И хотя доводить организацию до такого состояния нет никакой нужды, тем не менее, в этой книге исследуются возможные замыслы и структуры будущих компаний.



Scrum-мастер проводит крупное мероприятие в Открытом Пространстве при внедрении LeSS

6

Scrum-мастера

Хороший Scrum-мастер может отекать несколько команд, а выдающийся Scrum-мастер только один.

— Майкл Джеймс

Scrum для одной команды

Scrum-мастер обучает методике Scrum работников организации и наставляет их в ходе непрекращающегося внедрения. Он овладел теорией Scrum, применяя свои глубокие знания Scrum, чтобы направлять всех работников к мысли внести наибольший вклад в создание наиболее ценного продукта.

Роль Scrum-мастера нередко понимается неверно и поэтому плохо выполняется из-за того, что люди пытаются свести эту *новую* роль к уже существующей. Но она к ней не сводится. Это не глава команды, не руководитель проекта гибкой разработки и не лидер команды.

Роль Scrum-мастера — служить одной из двух “петель обратной мета-связи”, чтобы выяснить, насколько успешно действует Scrum¹. Это поддерживающая роль, помогающая работникам организации размышлять и улучшать свою организационную концепцию совершенства. Scrum-мастер создает среду для успешной деятельности работников организации.

Роль Scrum-мастера в LeSS

Роль Scrum-мастера является *новой* и нередко недопонимается при внедрении Scrum. В конечном итоге “оставшихся людей” пытаются превратить в Scrum-мастеров. Возможно, это и прекрасные люди, но им зачастую не хватает нужных навыков, мотивации и знаний Scrum. Они видоизменяют роль в нечто другое, и она принимается в организации как роль Scrum-мастера. Их действия иногда направлены против внедрения Scrum, превращая их в нечто совершенно противоположное Scrum-мастерам.

¹ Другой петлей обратной мета-связи служит Ретроспектива.

Роль Scrum-мастера в LeSS по-прежнему называется Scrum-мастером, а не LeSS-мастером по ряду объективных причин. При масштабировании соблюдаются следующие принципы, связанные с ролью Scrum-мастера.

- > **Системный подход и сосредоточенность на продукте в целом.** Чем крупнее группа, тем труднее видеть целое. Scrum-мастер помогает людям расширить свой кругозор, чтобы они увидели систему в целом: взаимодействия в группе продукта, задержки, причины и следствия, потенциальные возможности. Он также напоминает каждому о необходимости уделять основное внимание всему продукту, поскольку отдельно действующая команда неспособна создать потребительскую ценность.
- > **Крупномасштабная инфраструктура SCRUM — это тоже SCRUM.** Scrum-мастера в LeSS приходится иметь дело со сложными, крупномасштабными проблемами, поэтому он должен воздерживаться от искушения разрешить их *такими же* сложными, крупномасштабными средствами. Вместо этого ему следует обратиться к самой сути Scrum, найти простые средства и вооружить ими людей для преодоления стоящих перед ними препятствий. Он должен экспериментировать, исследуя крупномасштабные простые решения.
- > **Прозрачность.** Scrum-мастера являются блюстителями прозрачности. Но большая часть крупномасштабной разработки постоянно покрыта туманом неизвестности. Рассеяние этого тумана до состояния прозрачности — нелегкое и неблагодарное занятие в политических джунглях организации.

Правила LeSS

Scrum-мастера отвечают за вполне работоспособное внедрение LeSS, уделяя основное внимание Командам, Владельцу Продукта, организации и нормам практики разработки. Scrum-мастер уделяет внимание не только команде, но и всей организационной системе.

Scrum-мастер уделяет своей роли все свое рабочее время.

Один Scrum-мастер может обслуживать 1–3 команды.

Наставление: основные сферы деятельности Scrum-мастера

Документ *Scrum Master Checklist*, составленный Майклом Джеймсом и доступный по адресу <http://scrummasterchecklist.org/>, служит отличным справочником для Scrum-мастера. В нем определены четыре основные сферы деятельности Scrum-мастера:

- > Команда
- > Владелец Продукта
- > Организация
- > Нормы практики разработки

Эти основные сферы деятельности вскрывают типичный для Scrum-мастера недостаток, когда он уделяет слишком много внимания Команде. Чрезмерное внимание Команде приводит к неглубокому внедрению LeSS, поскольку Scrum-мастеру отводится ключевая образовательная и аналитическая роль при внедрении LeSS. Причина данного недостатка заключается в том, что роль Scrum-мастера нередко берет на себя один из членов Команды, уделяя ей часть своего рабочего времени. Но роли Scrum-мастера в LeSS следует уделять *все* рабочее время, опекая 1–3 команды, поскольку Scrum-мастер отвечает за успешное внедрение LeSS и должен быть сосредоточен на *всех* основных сферах деятельности.

Перечисленные выше основные сферы деятельности помогают лучше понять роль Scrum-мастера в LeSS, особенно если наглядно представить типичное сосредоточение его основного внимания во времени, как показано на рис. 6.1. Рассмотрим подробнее это графическое представление и поясним, чем оно обосновано.

Правило: Scrum-мастер уделяет своей роли все свое рабочее время.

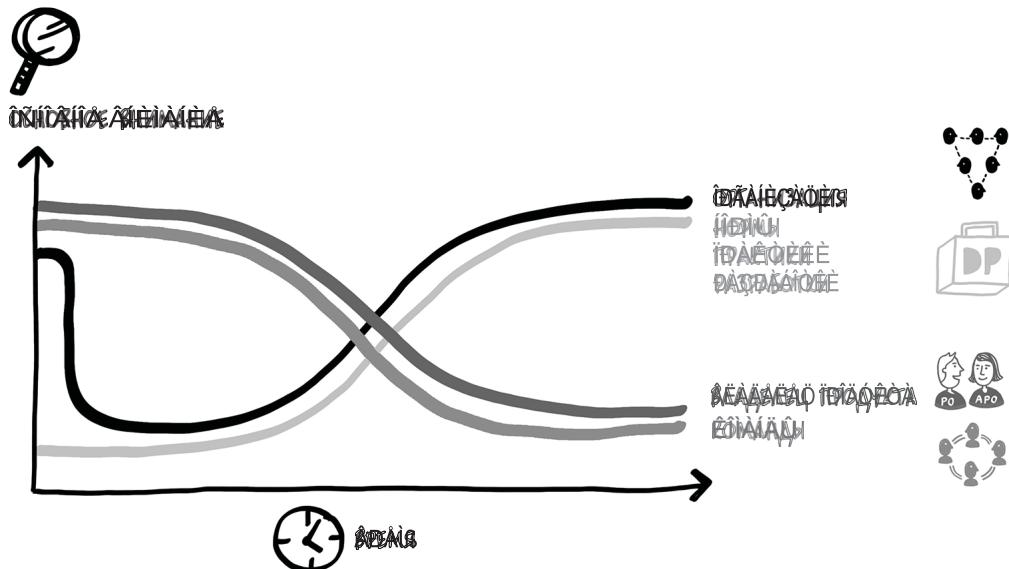


Рис. 6.1. Сосредоточение основного внимания Scrum-мастера во времени

Сосредоточение внимания на организации

Подробнее о структурных изменениях см. в главе 4.

Создание работоспособного и готового к поставке программного обеспечения вызывает доверие.

Совершенствование не прерывается, ведь мир не стоит на месте. Поэтому работа Scrum-мастера никогда не завершается.

Для внедрения LeSS требуется первоначальное структурное изменение, и поэтому основное внимание поначалу уделяется самой организации. Основное внимание на совершенствовании организации постепенно ослабевает, как только сформируется основная структура. А затем наступает черед команд добиться определенных результатов. Самый лучший способ изменить что-нибудь в организации — добиться определенных результатов (в данном случае — прироста готового к поставке продукта). А почему организация должна доверять Scrum-мастеру и его команде, если они не показали еще никаких результатов и ощутимой выгоды?

Очень важна следующая динамика: *со временем господствующие ограничения распространяются из команд на всю организацию*. Организационная структура и правила сдерживают производительность труда в командах. И тогда Scrum-мастер уделяет все больше внимания совершенствованию организации.

Сосредоточение внимания на командах

Первоначально Scrum-мастер уделяет немало внимания командам, но со временем его внимание ослабевает. Как только команды сформированы, Scrum-мастер прилагает немало усилий образованию и обучению команд самоуправлению, координированию их действий и повышению разделяемой ими ответственности. Со временем команды меньше полагаются на своего Scrum-мастера, беря на себя всю ответственность.

Созревание команд служит одной из причин, по которым во многих примерах внедрения Scrum роли Scrum-мастера уделяется лишь часть рабочего времени. Но в LeSS этой роли следует уделять все рабочее время. Как только созреет первая команда, Scrum-мастер может заняться другой, а фактически число опекаемых им команд должно быть не больше трех. Когда Scrum-мастер опекает несколько команд, его внимание автоматически смещается на общую картину в организации и на Владельца Продукта.

Сосредоточение внимания на Владельце Продукта

Подробнее о взаимоотношениях с Владельцем Продукта речь пойдет в главе 8.

Первоначально Scrum-мастер уделяет внимание обучению Владельца Продукта его роли. В частности, он обучает Владельца Продукта, как лучше всего пользоваться Заделом Продукта, устраивает его взаимодействие с командами и всегда готов помочь проанализировать сложившуюся ситуацию.

У Владельца Продукта имеется несколько взаимоотношений, в том числе и те, что показаны на рис. 6.2.

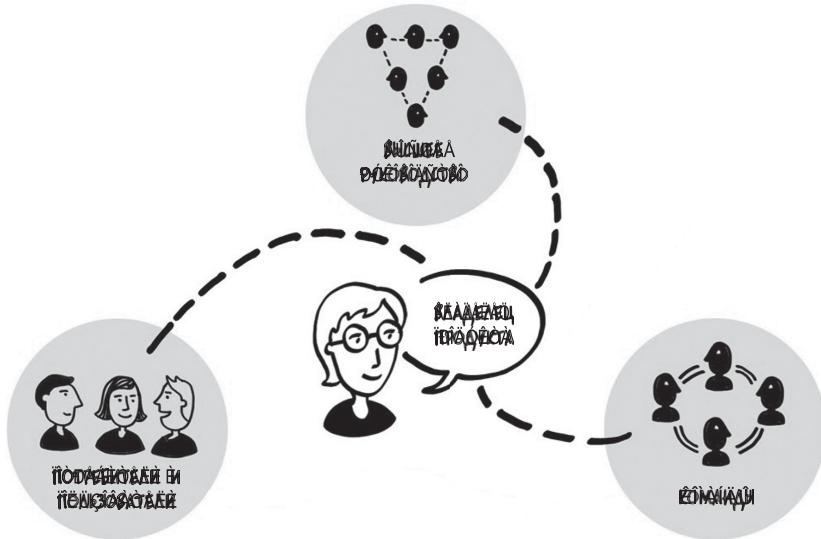


Рис. 6.2. Взаимоотношения Владельца Продукта

Внимание следует уделять *не только* взаимоотношениям Владельца Продукта и Команд. Scrum-мастер должен оказывать поддержку в установлении других взаимоотношений Владельца Продукта, в том числе следующих.

- > **Взаимоотношения Владельца Продукта с заказчиками.** Scrum-мастер помогает Владельцу Продукта налаживать более тесные связи с реальными пользователями и заказчиками. Владельцу Продукта нужна их ответная реакция, чтобы убедиться в правильности выбранного направления продукта. Если же Владелец Продукта неспособен выполнять свою роль, а такое случается, то Scrum-мастер должен помочь организации найти другого Владельца Продукта, который сможет лучше наладить тесные связи с пользователями и заказчиками.
- > **Взаимоотношения Владельца Продукта с высшим руководством.** Scrum-мастер должен помогать Владельцу Продукта в работе с высшим руководством, чтобы постоянно доводить до него состояние разработки, оказывать всяческую поддержку Владельцу Продукта и работать в направлении оптимизации влияния продукта.
- > **Взаимоотношения Владельца Продукта с командами.** Scrum-мастер помогает Владельцу Продукта установить взаимоотношения доверия, равноправия и сотрудничества. Это нелегкое дело, поскольку исторически для таких взаимоотношений характерны непрозрачность, взаимные упреки и недоверие.

Основное внимание Scrum-мастера на Владельце Продукта должно со временем ослабеть, как только Владелец Продукта лучше освоится со своей ролью в организации, практикующей LeSS.

Сосредоточение внимания на нормах практики разработки

Первоначально Scrum-мастер занят формированием рабочих команд, вполне способных совместно производить нечто *ценное*. Но как только основное внимание Scrum-мастера на командах и Владельце Продукта начнет постепенно ослабевать, он должен переключить свое внимание на помощи командам в усовершенствовании их норм практики разработки.

Scrum-мастер должен быть в курсе самых передовых норм практики разработки и содействовать их внедрению в командах. Как правило, внедрению LeSS сопутствуют массивные кодовые базы с немалым объемом архаичного устаревшего кода. Поэтому внедрение в командах таких норм передовой практики разработки, как разработка посредством тестирования, непрерывная разработка и автоматическое приемочное тестирование, вызывает немалые трудности. Большое внимание, уделяемое Scrum-мастером нормам практики разработки, не ослабевает и впредь, поскольку командам становится все труднее совершенствоваться.

Наставление: пять инструментальных средств Scrum-мастера

Прежде чем перейти к наставлениям, имеющим непосредственное отношение к LeSS, рассмотрим инструментальные средства, проясняющие особенности работы Scrum-мастера. Нам лично нравятся следующие инструментальные средства Scrum-мастера.

См. далее раздел “Наставление: список рекомендованной литературы для Scrum-мастера”.

- > **Спрашивать.** Scrum-мастер должен служить зеркалом для всех, кому требуется помочь найти в нем как в отражении пути к совершенствованию. Один из самых эффективных способов достичь этой цели — задавать много наводящих вопросов, помогающих людям самим найти ответы.
- > **Обучать.** Scrum-мастер должен глубоко разбираться в методике Scrum и помогать команде понять причины, по которым Scrum действует так, а не иначе. Одним из средств достичь этой цели служит обучение. Но при этом Scrum-мастер не должен переусердствовать, чтобы не отбить у людей охоту учиться. Избегая фанатизма, он должен сосредоточить обучение на *причинах*, сохраняя непредубежденность и внимательно слушая других.
- > **Содействовать.** Scrum-мастер показывает, как проводить мероприятия в LeSS и вести продуктивные дискуссии, содействуя им. Он создает обстановку прозрачности, выводя конфликты на поверхность и помогая командам разрешить их. Но он должен помнить: ему нужно, чтобы команда брала большую часть обязанностей на себя. Если Scrum-мастеру по-прежнему приходится устраивать совещание по Планированию Спринта на стадии десятого Спринта, значит, он не справляется со своей ролью.

См. далее раздел “Наставление: список рекомендованной литературы для Scrum-мастера”.

- > **Не делать ничего за других.** Scrum-мастер должен освободить место для того, чтобы люди взяли на себя ответственность. Как это сделать? Для начала не делать ничего самому за других. Если у одной или нескольких команд возникнут трудности, Scrum-мастер должен сначала понаблюдать за их поведением, чтобы выяснить, смогут ли они справиться с этими трудностями без его помощи. Благодаря этому освобождается место для дальнейшего роста.
- > **Вмешиваться.** Команды должны учиться сами, но когда дело не ладится, Scrum-мастер должен вмешаться во избежание непоправимого ущерба.

В приведенном выше перечне явно отсутствуют некоторые инструментальные средства, поддерживаемые и применяемые Scrum-мастерами на практике. Рассмотрим их и причины, по которым их лучше избегать.

Scrum-мастер не должен представлять команду. На некоторых мероприятиях в LeSS должны присутствовать представители команд, но их не должен представлять Scrum-мастер. Кто же тогда должен быть представителем команды? Это дело самой команды. Scrum-мастер уделяет своей роли в LeSS все свое рабочее время, и поэтому он не является членом Команды. Было бы, по меньшей мере, странно, если бы он представлял команду на каком-нибудь мероприятии.

Scrum-мастер не должен принимать решения за команды. Команда принимает свои решения самостоятельно, а несколько команд — совместно. Поэтому Scrum-мастер должен не принимать решения за команды, а помогать и содействовать им в принятии решений.

Scrum-мастер должен проявлять осторожность, давая советы. Его советы командам не всегда интерпретируются как таковые. Это особенно справедливо для молодых команд, которым требуется старший, который смог бы принять решение за них.

Scrum-мастер должен аккуратно устранять препятствия. Нам не раз приходилось наблюдать, как устранение препятствий служило оправданием для любого поведения. Большинство ежедневных препятствий должны устраниться самими командами. В то же время Scrum-мастер уделяет основное внимание созданию среды для успешной работы команд, а следовательно, он устраивает *организационные причины* для появления препятствий. И сделать это намного труднее.

Наставление: содействие крупным группам в проведении совещаний

Провести совещание продуктивно, эффективно и занимательно нелегко, особенно если на нем присутствует много людей. Ведь совещания не должны быть скучными. Содействие проведению совещаний в крупных группах является очень важным наработкой, который должен приобрести Scrum-мастер. В частности, он должен взять на вооружение такие методики, как Открытое Пространство и Мировое Кафе.

Для содействия в проведении крупных совещаний нередко применяются следующие методы.

- > **Децентрализация.** Сосредоточение всеобщего внимания на одном центральном вопросе замедляет ход совещания. Поэтому присутствующих на нем следует по возможности разделить на небольшие группы и провести параллельные мероприятия. Но для обмена мнениями и ради согласованности необходимо также принять меры к объединению.
- > **Установка большого количества белых досок и перекидных блокнотов.** Обсуждение будет намного более продуктивным, если его участники будут записывать и набрасывать на доске или бумаге свои мысли.
- > **Удаление лишней мебели.** Чтобы совещание не стало скучным, а обсуждение — сосредоточенным вокруг стола для председательствующих, достаточно убрать этот стол. И динамика проведения совещания сразу же изменится в лучшую сторону!
- > **Исключение компьютеров с проекторами.** Применение компьютеров — один из самых лучших способов подавить динамику проведения совещания. Они централизуют совещание, а тот, кто управляет компьютером, становится узким местом. Если без компьютера все же не обойтись, нужно постараться, чтобы он не стал центральной точкой совещания.
- > **Добровольное участие.** Людей не следует принуждать к обсуждению. Сначала нужно перечислить вопросы, предлагаемые для обсуждения, а затем распределить людей по группам для добровольного участия в обсуждении интересующих их вопросов.
- > **Постановка ясных целей.** Сначала нужно объяснить присутствующим, зачем их собрали на совещании, а затем его цель.
- > **Ретроспектива.** Совещание следует завершить ретроспективой, чтобы подготовить благоприятную почву для последующих совещаний.

Подробнее одобрительно о участии см. в главе 3.

Наставление: содействие в обучении и приобретении многих навыков

Суть непрерывного совершенствования в постоянном обучении, что особенно важно для разработки программных продуктов, которые производят не машины, а только люди. К сожалению, многие люди не стремятся включить обучение новым навыкам в свою работу или жизнь.

Недостаток организационного обучения служит главным препятствием для внедрения LeSS. Поэтому Scrum-мастеру нужно создать такую обстановку, в которой люди

хотели бы учиться и стать из узких специалистов в одной области специалистами во многих областях².

Ниже приведены некоторые рекомендации, которые можно дать Scrum-мастеру по поводу содействия обучению.

- > Будьте примером для других, овладевая новыми навыками.
- > Делитесь со своей командой тем, чему научились сами.
- > Организуйте библиотечку со справочной литературой рядом с рабочим местом.
- > Напоминайте членам команды, что они родились без навыков, и поэтому должны доказать свою способность овладевать новыми навыками.
- > Делитесь с командой статьями, но не только на тему гибкой разработки, Scrum, или LeSS.
- > Поощряйте учебные занятия, например, мини-курсы, просветительные беседы или обсуждения прочитанной литературы.
- > Напоминайте команде о том, что они наделены полномочиями планировать обучение в ходе Спринта.
- > Предлагайте команде анализировать имеющиеся навыки в ходе Ретроспективы.

Наставление: работать с сообществами

Сообщество — это группа добровольцев из команд, разделяющих общие интересы, стремящихся углубить свои знания, подискутировать или пообщаться с коллегами. Участие в сообществе *совершенно добровольно*.

См. раздел “Наставление: сообщества” главы 13.

Сообщества образуют неформальные связи между командами, что очень важно для обучения, координирования действий и непрерывного совершенствования. Scrum-мастер должен работать с сообществами, наставляя их поддерживать здоровую атмосферу в течение длительного времени. Особенно активного участия Scrum-мастера требуют сообщества LeSS и Scrum-мастеров.

Сообщество LeSS

Scrum-мастера содействуют взаимному обучению команд любым предметам, включая и LeSS. А поскольку команды работают в одном и том же контексте, то они, вероятнее всего, наталкиваются на сходные препятствия и могут многому научиться друг у друга. Такое сообщество Scrum-мастера могут создать общими усилиями. Ниже приведены некоторые рекомендации Scrum-мастерам по этому поводу.

² Мы не рекомендуем всем становиться универсальными специалистами и знать все на свете. Более подробно этот вопрос рассматривается в статье *Specialization and Generalization in Teams* (Специализация и универсализация в командах), доступной по адресу <https://www.scrumalliance.org/community/articles/2011/january/specialization-and-generalization-in-teams>.

См. раздел “Наставление: ОткрытоеПространство” главы13.

- > **Группа для обсуждения вопросов LeSS.** Организуйте группу в пределах группы продукта или даже в масштабах всей организации для обсуждения вопросов LeSS и обмена опытом.
- > **Внутренние собрания по вопросам LeSS.** Организуйте собрания для обмена опытом. Воспользуйтесь методикой Открытого Пространства, поскольку она основывается на самоорганизации, что вполне соответствует сути LeSS.
- > **Обсуждение вопросов LeSS за кружкой пива.** Запланируйте встречи в пабе за кружкой пива.
- > **Распространение историй в блогах, на вики-страницах или информационных рассылках.** Напишите истории о своих командах или убедите их сделать это самим.
- > **Наблюдение за другими командами.** Призывайте свои команды наблюдать за другими командами и обсуждать вместе с ними, как и почему они работают иначе.

Сообщество Scrum-мастеров

Нередко роль Scrum-мастера приносит разочарование. Но он в этом не одинок! Поэтому он может связаться с другими Scrum-мастерами, чтобы организовать свое сообщество. Другие могут наставить его на роль Scrum-мастера. Ниже приведены некоторые рекомендации Scrum-мастерам по этому поводу.

- > **Списки рассылки только для Scrum-мастеров.** Такие списки организуются аналогично спискам рассылки по вопросам LeSS, но только для приглашения участников к обсуждению. Они должны основательно разбираться в вопросах LeSS, чтобы обсуждение вышло далеко за рамки основ.
- > **Наблюдение за другими Scrum-мастерами.** Спросите у других Scrum-мастеров разрешения понаблюдать за их деятельностью. После этого пригласите их к открытому обсуждению своих наблюдений, анализу и выработке предложений по совершенствованию роли Scrum-мастера.
- > **Работа в паре с другим Scrum-мастером.** Устройте совещание и понаблюдайте вместе с ним за командой или проинструктируйте команду в паре с ним.
- > **Учебная группа.** Попросите каждого ее участника прочитать одну и ту же главу избранной книги, а затем обсудить ее вместе. Читайте по одной главе в неделю, обсуждая ее — возможно, за обедом.

Наставление: руководство Scrum-мастера по выживанию

Организации являются опасным местом, кишащим дислексическими зомби, тупыми начальниками, переключающими контекст вампирами, недоделанными

разработчиками и Scrum-антимастерами. Но это совсем не означает, что нужно впадать в панику. Ниже приведены некоторые рекомендации по выживанию Scrum-мастера в такой неблагоприятной среде.

Обращайте обвинение в конструктивное действие

К сожалению, обвинение является типичной реакцией, когда что-нибудь пойдет не так. И чем больше команд и мест разработки, тем хуже обстоит дело. Обвинение удобно, поскольку оно освобождает от ответственности, особенно если очевидно, что причиной всех бед служит другая команда!

Scrum-мастер не должен вообще участвовать в поиске виноватых. Вместо этого он должен помочь своим командам обратить обвинение в конструктивные действия, которые они могут предпринять. И это можно сделать, задав следующие вопросы.

- > Что можно сделать, чтобы изменить X в нашей среде?
- > Если сделать ничего нельзя, то следует принять то, что теперь нельзя изменить.
- А что можно сделать, чтобы исключить или уменьшить влияние X на нас?

В качестве примера ниже приведен следующий диалог.

Команда: Мы не можем тестировать, ведь для этого требуется доступ к среде на уровне администратора.

Scrum-мастер: Ладно, а что мы можем сделать, чтобы получить такой доступ?

Команда: Мы не можем получить такой доступ из-за организационных правил.

Scrum-мастер: Хорошо. А что мы можем сделать, чтобы показать, чего стоят эти правила, и в конечном итоге изменить их. А если принять отсутствие доступа на уровне администратора, то как нам лучше всего провести тестирование без него?

Не берите на себя роль координатора действий команд

В традиционных организациях имеется координатор (руководитель проекта), согласующий действия команд. А в организациях, практикующих LeSS, координирование действий команд входит в обязанности самих команд.

Многие команды настолько привыкли к роли координатора, что они надеются, что эту роль возьмет на себя Scrum-мастер. Но он не должен ни в коем случае этого делать. Вместо этого он может оказать помощь командам следующим образом.

- > Напомнить им, что это их обязанность и пояснить причины.
- > Представить команды друг другу.
- > Помочь им согласовать механизм координирования их действий.

Но сам Scrum-мастер ни в коем случае не должен брать на себя роль координатора.

Более подробно механизмы координирования действий рассматриваются в главе 13.

Объединяйте усилия для внесения изменений

Организационные изменения, требующиеся для создания условий, благоприятных для успешной работы команд, скорее всего, окажутся трудными, хотя и сходными для всех команд. Поэтому Scrum-мастерам нужно объединить свои усилия, поскольку вместе они сильнее.

Как это сделать? Необходимо обсудить самые действенные в настоящий момент изменения и создать для них прецедент, а затем собрать всю уместную информацию и внести предложение (старшему) руководству. Не следует, однако, ожидать, что они сразу же примут предложение, а лучше воспользоваться им, чтобы начать обсуждение, и набраться терпения.

Налаживайте партнерские отношения с руководителями

Подробнее об обязанностях руководителей см. в главе 5.

Обязанности руководителей и Scrum-мастеров сходны. И те и другие создают условия для создания наилучшего продукта. Scrum-мастера работают в тесной связи с командами, тогда как руководители заняты дополнительной организационной работой. Но для того чтобы внести изменения, им приходится работать вместе (рис. 6.3).

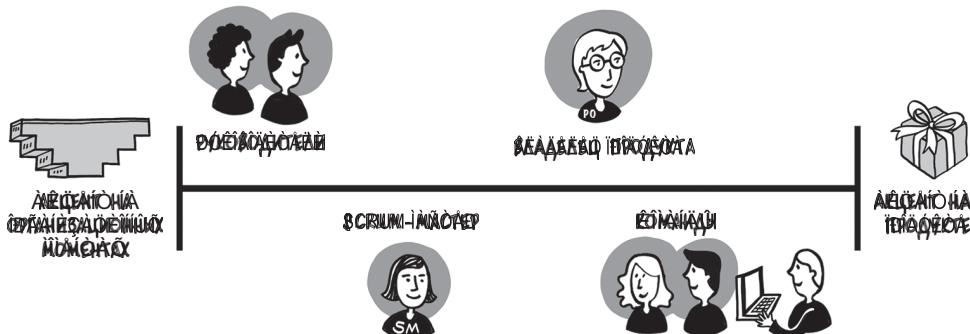


Рис. 6.3. Сосредоточение разных ролей на продукте и организационных моментах

Начать следует с регулярных еженедельных обсуждений, чтобы составить общее представление о текущих проблемах. Затем нужно выбрать самую важную из них, выработать ее решение, далее взять следующую по степени важности проблему и т.д.

Иногда партнерские отношения с руководителями налаживаются легко, потому что руководители сами стали Scrum-мастерами! Но это зачастую вызывает трудности в самоуправлении команд. В некоторых организациях для преодоления этих трудностей руководитель одной команды становится Scrum-мастером другой команды. И хотя такой подход иногда оказывается вполне работоспособным, он все же не рекомендуется по двум причинам. Во-первых, в некоторых иерархических средах разрыв между руководителями и подчиненными настолько велик, что руководителю в роли Scrum-мастера практически невозможно заслужить доверие команды. И, во-вторых, некоторые

руководители настолько заняты другой “организационной работой”, ценность которой не вполне очевидна, что они не могут уделять все свое рабочее время роли Scrum-мастера.

Сохраняйте здравомыслие

Роль Scrum-мастера состоит в том, чтобы изменить положение дел в организации, хотя у него нет на то *официальных* полномочий. И это совсем неплохо. Scrum-мастер должен убедить людей в необходимости перемен, поскольку они считают, что делают все правильно. Но влиятельные перемены в организациях непросты и нечасты, и как бы ни стараться их осуществить, они действуют в обратном направлении. И тогда встает вопрос, как выжить в подобных условиях? Чтобы выжить в организациях, требующих перемен, и при этом сохранить здравомыслие, нужны следующие качества.

- > **Терпение и минимальные ожидания.** Перемены в большинстве организаций происходят медленно. Поэтому ожидания (но не цели) должны быть минимальными, и при этом полезно не забывать, что работать в данном направлении придется не один год. В то же время следует приветствовать даже небольшие перемены.
- > **Настойчивость.** Не следует ожидать, что предложения относительно перемен будут приняты сразу. Вместо этого следует быть готовым пояснить их миллион раз (зачастую одним и тем же людям).
- > **Смелость.** Никакие перемены невозможны без смелости. Не следует бояться говорить о них с высшим руководством или вносить неудобные для себя предложения.
- > **Чувство юмора.** Чтобы убедить людей изменить хоть что-нибудь, пришлось потрудиться целый год. И они наконец согласились, но стало еще хуже. Что же делать? С одной стороны, отнестись к этому серьезно, а с другой — несерьезно и посмеяться. Ведь это единственный способ выжить.
- > **Открытость и скромность.** Перемены следует предлагать смело, настойчиво и терпеливо. Не принимайте всерьез глупые решения, губящие всю вашу работу. И все это нужно делать открыто и скромно, иначе ничему новому нельзя научиться. Может быть, правы были они, а не вы?

Не забыли ли мы о *терпении*? Без этого качества Scrum-мастера никак нельзя.

Наставление: список рекомендованной литературы для Scrum-мастера

Предполагается, что Scrum-мастер должен быть специалистом по Scrum. Но насколько он владеет Scrum? Овладение означает, что учиться больше нечему. И хотя мы уже давно занимаемся внедрением LeSS, тем не менее, продолжаем изучать LeSS. И Scrum-мастерам нужно постоянно совершенствоваться. Это можно, в частности, сделать, прочитав одну из рекомендуемых ниже книг.

- > *Leading Teams*, Richard Hackman. В этой книге подытожен более чем 30-летний опыт исследования команд, и поэтому она является едва ли не самой лучшей книгой по формированию самоуправляемых команд.
- > *The Skilled Facilitator*, Roger Schwarz. Превосходное пособие по совершенствованию навыков содействия, столь необходимых Scrum-мастеру.
- > *Co-active Coaching*, Kimsey-House и др. Эта книга — один из самых лучших источников для изучения особенностей наставничества.
- > *The Five Dysfunctions of a Team*, Patrick Lencioni. Это небольшая, но замечательная история о том, как команды работают (или не работают).
- > *Humble Inquiry*, Edgar Schein. Автор этой книги имеет 50-летний опыт организации разработки и консультирования организаций. Из своего солидного опыта он сделал такой вывод: нужно меньше рассказывать, а больше спрашивать.

Наставление: особое внимание Scrum-мастера

В некоторых сферах деятельности часто возникает немало осложнений, которые требуют особого внимания со стороны Scrum-мастера.

См. главу 8.

- > **Неверно функционирующие взаимоотношения между Владельцем Продукта и Командами.** Нередко между Владельцем Продукта и Командами исторически возникает недоверие. Причина тому — стремление Владельца Продукта принудить команды брать на себя обязательства, что в конечном итоге приводит к хаосу в разработке.
- > **Неверно функционирующие взаимоотношения между Командами и Владельцем Продукта.** На самом деле получать “готовые” функциональные средства в каждом Спринте кажется трудным делом для многих команд. Это, в частности, объясняется тем, что такие команды привыкли беспокоиться только об отдаленных сроках, а не о Спринте. Положение усугубляется тем, что Владелец Продукта старается быть “хорошим” для команды, принимая незавершенную работу, что приводит к еще большей халатности.
- > **Враждебные отношения.** От появления противостоящих групп, указывающих на вину друг друга, вместо того, чтобы предпринять конструктивное действие, теряют все. Примером тому служит противостояние Команд и Владельца Продукта, Команд и руководителей, одних и других мест разработки.
- > **Внедрение Scrum без перемен.** Scrum-мастер не должен быть помощником команд, приглашающим на совещание и кофе. И тем более он не должен быть Scrum-антимастером, назначенным на эту роль, но ничего, по существу, не делающим.

- > **Удаленный Scrum-мастер.** Scrum-мастер должен находиться в том же месте, где и его команды. Он должен наблюдать за тем, как работают команды, чтобы выяснить, как помочь им в частности и организации вообще.
- > **Scrum-мастер в роли руководителя проекта.** И хотя мы уже упоминали об этом, стоит повторить еще раз. Scrum-мастер *не является* руководителем проекта. Руководители проектов ведут проект, планируя работу, следя за ее продвижением, координируя действия и принимая меры, чтобы проект не выбивался из намеченного графика. Scrum-мастер *ничего* подобного не делает и *не* отвечает за проект. Он не является также контактным лицом ни для проекта, ни для команды. Вместо этого он уделяет основное внимание формированию отличных команд, созданию удобных организационных условий и обучению.

LeSS Huge

Роль Scrum-мастера в LeSS Huge, по существу, остается такой же, как и в LeSS. Следовательно, LeSS Huge не вносит никаких дополнительных правил, касающихся роли Scrum-мастера.

Наставление: исключение разобщенности Областей Требований

Самое распространенное затруднение в LeSS Huge состоит в отсутствии взаимодействия между Областями Требований. Такое затруднение чаще всего возникает при взаимно однозначном соответствии организационной структуры, организации мест и Областей Требований. Scrum-мастер должен оказывать помощь в разрешении подобного затруднения. Ниже приведены некоторые рекомендации по этому поводу.

- > Выберите одного Scrum-мастера, оказывающего помощь Команде Владельцев Продукта и предоставляющего им ответную реакцию на улучшения. См. главу 4.
Подробнее о Команде Владельцем Продукта см. в главе 8.
- > Назначьте одного Scrum-мастера для двух команд, принадлежащих разным Областям Требований.
- > Организуйте для упоминавшегося ранее сообщества LeSS мероприятия (например, внутреннее собрание) по Областям Требований.
- > Организуйте Ретроспективу и/или Подведение Итогов, охватив хотя бы две области.

Продукт в LeSS



Два продукта или один?

7

Продукт

Любой продукт, требующий руководства для работы, считается испорченным.
— Элон Маск

Scrum для одной команды

Поначалу понятие Scrum вызывало недоразумение и в большинстве ранних описаний обозначалось как инфраструктура для управления сложными *проектами*. Кен Швабер, один из создателей Scrum, написал книгу *Agile Project Management with Scrum* (Гибкое управление проектами средствами Scrum), которая начиналась так: “Я предлагаю вам Scrum — запутанный и парадоксальный процесс управления сложными проектами”. И это кажется странным, поскольку понятие Scrum всегда было связано с *продуктами*, а не с проектами. В самом деле, в инфраструктуре Scrum имеется Задел Продукта и Владелец Продукта, но не план выполнения проекта и не руководитель проекта.

Правда, это недоразумение было разрешено. В упоминавшемся ранее *Руководстве по Scrum* (*Scrum Guide*; <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>) теперь утверждается следующее: “Scrum — это инфраструктура для разработки и сопровождения сложных продуктов”. Из этого обозначения понятия Scrum исключены проекты¹. Имеет ли это какое-то значение? Имеет, и очень большое, как поясняется в этой главе.

Ведение работ как продуктов, а не проектов изменяет структуры, решения, поведение в разработке продукции.

¹ Если не считать, что каждый Спринт подобен мини-проекту.

См. раздел “Наставление: начало внедрения” главы 3.

А что собой представляет продукт? В упомянутом выше руководстве по Scrum ничего не говорится о том, что означает понятие *продукт* и каковы его пределы. Возможно, потому что это вполне очевидно? Но в крупномасштабной разработке определение продукта редко оказывается очевидным и относится к числу самых важных решений, которые приходится принимать.

Продукт в LeSS

Почему так важно определение продукта? Оно определяет уровень и размеры Задела Продукта, конечных потребителей и подходящего Владельца Продукта.

Оно делает инфраструктуру LeSS Huge более распространенной, чем принято ожидать. Чем шире определение продукта, тем больше, как правило, команд работают над этим продуктом. И в конечном счете внедрение LeSS превращается во внедрение LeSS Huge.

При масштабировании соблюдаются следующие принципы, связанные с определением продукта.

- > **Сосредоточение внимания на всем продукте.** Очевидно, что разные определения продукта приводят к разной степени внимания к нему. Какое же определение продукта может привести к широкой, но достаточно ясной перспективе, заслуживающей сосредоточения внимания?
- > **Системное мышление и непрерывное улучшение до полного совершенства.** Разные определения продукта приводят к более широкой или узкой перспективе. Если предпочтительны более широкие определения продукта, то какую динамику системы они вызовут? А если определение продукта является гибким, то может ли оно способствовать непрерывному совершенствованию?
- > **Ориентирование на клиентов.** Какое бы определение продукта ни было выбрано, оно должно оставаться ориентированным на клиентов. Okажется ли расширение определения продукта более или менее ориентированным на клиентов? И оказываются ли с этой точки зрения лучше или хуже разные направления, в которых расширяется определение продукта?
- > **Добиваться большего средствами LeSS.** Более широкие определения продукта приводят к тому, что инфраструктура LeSS Huge становится более распространенной, чем предполагалось первоначально. Усложняет ли это дело или же позволяет добиваться большего средствами LeSS?

Правила LeSS

Определение продукта должно быть широким и ориентированным на конечного пользователя до такой степени, чтобы стать практическим.

Со временем определение продукта может быть дополнено.

Предпочтительны более широкие определения продукта.

Наставление: назначение продукта

Владелец Продукта назначает приоритеты в Заделе Продукта, а Команды постепенно создают *продукт*, сосредоточивая свое внимание на продукте в целом. Но что собой представляет продукт? Должны ли его выпускать команды или текущий отдел? Является ли он компонентом, каркасом или платформой? Имеет ли это какое-то значение?

Имеет. Определение продукта устанавливает пределы Задела Продукта и подходящего Владельца Продукта. При внедрении LeSS оно устанавливает величину организационных изменений, которые следует ожидать, а также тех, кого они должны коснуться. На первый взгляд определить назначение продукта нетрудно, но на самом деле это не так. Это ответственный выбор.

Можно, например, остановить свой выбор на узком определении продукта. Допустим, что компонент, разрабатываемый двадцатью командами, является продуктом. Таким образом, для него назначается технический Владелец Продукта, не ориентированный на клиентов, а в Заделе Продукта появляются технические элементы. Плохо ли это? Плохо. Такой продукт нельзя продать, он не ориентирован на клиентов и не доставляет потребительскую ценность, что зачастую приводит к построению технически совершенного, но малопригодного или вообще не употребляемого продукта.

С другой стороны, можно остановить свой выбор на *широком* определении продукта. Чем шире определение продукта, тем больше он ориентирован на клиентов. Но если определение продукта оказывается слишком широким, то оно становится непрактичным, поскольку может охватывать менее дружественные отделы или даже разные компании. А кроме того, становится намного труднее выразить ясную и привлекательную концепцию продукта.

Так какое же определение продукта следует выбрать? Предпочтение в LeSS следует отдавать более широким определениям продукта, поскольку они приводят к следующему:

- > допускают более точное и ориентированное на клиентов назначение приоритетов, а в конечно счете — лучшее общее представление о разработке и продукте;
- > разрешают зависимости с помощью команд для функциональных средств;
- > позволяют обдумывать продукт вместе с заказчиком, уделяя больше внимания реальной проблеме и ее влиянию, чем запрашиваемым требованиям;

- > исключают дублирование функциональных возможностей;
- > создают более простые организации.

Более точное и ориентированное на клиентов назначение приоритетов. Узкие определения продуктов приводят к появлению многих разрозненных мелких Заделов Продукта. Как назначить среди них приоритеты? Как выяснить несогласованность приоритетов, если отсутствует общее представление? И что получается в итоге? В лучшем случае получается довольно грубое назначение приоритетов крупными партиями на уровне заделов, а не элементов. Но чаще всего это приводит к интригам с целью продвинуть свою излюбленную команду, которая с чрезвычайной продуктивностью и постоянством доставляет малоценные элементы. С другой стороны, при широком определении продукта все элементы находятся в одном и том же заделе, что позволяет точно назначить приоритеты и улучшить общее представление о разработке и продукте.

Разрешение зависимостей. Узкие определения продуктов обусловливают зависимости между *отдельными* продуктами. Рассмотрим в качестве примера продукта платформу с надстроенными над ней несколькими “прикладными” продуктами. Для управления зависимостями между ними создаются координирующие роли, на долю которых приходится немало дополнительного планирования. Эти так называемые продукты на самом деле являются лишь компонентами более крупного продукта, а методы для разрешения зависимостей между ними остаются такими же, как и в организациях с командами для компонентов. Но подобные зависимости могут быть разрешены иначе. При более широком определении продукта эти зависимости оказываются в пределах одного и того же продукта. Чтобы разрешить их, достаточно передать команде для функциональных средств ориентированное на конечных клиентов функциональное средство, охватывающее платформу и приложение. Благодаря этому исключаются дополнительные роли и усложнение.

Обдумывание продукта вместе с заказчиком. Узкое определение продукта ограничивает возможные решения реальных проблем заказчика текущими пределами продукта. Допустим, заказчик запрашивает возможность экспортieren данные в формате XML, чтобы другой “прикладной” продукт мог импортировать эти данные. Если определение продукта сужено до одного из приложений, то оно будет реализовано так, как и запрашивалось, поскольку требования (и решения) ограничиваются пределами продукта. А если определение продукта расширено, то увеличивается творческий простор для команд, которые могут исследовать лучшие решения того, чего пользователь пытается достичь. Что же касается экспорта данных, то команды могут найти способ связать приложения, избавив тем самым пользователя от необходимости выполнять ручной экспорт и импорт данных.

Дублирование функциональных возможностей. Узкое определение продукта может привести к появлению сходных продуктов или вариантов продукта. А это в конечном итоге приводит к появлению разрозненных отделов, имеющих или желающих иметь отдельные хранилища исходного кода. Если в нескольких продуктах требуются одинаковые или сходные функциональные возможности, то в конечном счете возникает одно из следующих событий.

1. Функциональные возможности переносятся из первого продукта в остальные, а для этого обычно требуется дополнительное координирование и уточнение кода, что редко оправдывается.
2. Одни и те же функциональные возможности повторно реализуются для разных продуктов.
3. Создается новый внутренний “компонентный” продукт, в который переносятся функциональные возможности со всеми связанными с этим организационными осложнениями.

А при более широком определении продукта разработка его вариантов ведется как единое целое через один Задел Продукта. Благодаря этому организуется одно общее хранилище исходного кода, несколько раз исключается потребность в повторной реализации одинаковых функциональных возможностей, а также упрощается структура организаций.

Создание более простых организаций. Узкое определение продукта приводит к появлению дополнительных организационных структур для координирования работы над отдельными продуктами. Например, при рассмотрении способов разрешения зависимостей выше упоминались роли *координаторов*. Еще одним примером такой дополнительной структуры служит *управление портфелем проектов* (или программ), которое сводится к назначению приоритетов для требований крупными партиями. Ведь работы над узко определенными продуктами должны каким-то образом распределяться по приоритетности и финансироваться. Следует, однако, иметь в виду, что очевидная, на первый взгляд, потребность в таком управлении портфелем проектов является следствием усложнения, возникающего из-за сужения определений продуктов!

С другой стороны, широкое определение продукта приводит к тому, что вся работа оказывается в одном и том же Заделе Продукта, а следовательно, все приоритеты назначаются через один Задел Продукта. Благодаря этому исключается потребность в *устаревшем* ныне управлении портфелем проектов, и таким образом упрощается структура организаций².

Масштабы усложнения организаций в LeSS сокращаются благодаря более широким определениям продуктов, исключая ненужные сложные решения, которые можно упростить иными путями.

Ограничивающие факторы, сужающие определение продукта

Итак, широкие определения продуктов более предпочтительны. Но если расширить их до крайности, то получится один Задел Продукта на все мир. Какие же ограничения можно наложить на определение продукта? Если говорить вкратце, то это общность и структуры.

² Руководство, управляющее портфелем *продуктов* на уровне компании, принимает решения относительно тех рынков, на которых компания должна присутствовать. Такие компании, вероятнее всего, останутся очень крупными.

Общность. У элементов в заделе должна быть какая-то общая причина для принадлежности одному и тому же продукту. Определение продукта ограничивают следующие основные виды общности.

- > **Концепция.** Общая концепция продукта стимулирует людей и способствует раскрытию их творческого потенциала. Слишком широкое определение продукта обобщает его концепцию до привычного состояния “делать свое дело”, и люди теряют интерес к продукту. Поэтому все элементы желательно нацелить на общую содержательную концепцию продукта.
- > **Клиенты или рынки.** Несколько относительно небольших продуктов могут служить для одних и тех же клиентов или рынков. Включив все эти небольшие продукты в определение продукта, можно упростить назначение приоритетов и стимулировать команды исследовать проблемы не обслуженных клиентов. Но слишком широкое определение продукта, учитывающее всех возможных клиентов, не позволяет выявить проблемы тех клиентов, которым следует уделить особое внимание. Поэтому желательно, чтобы все элементы были прояснены и соотнесены с рядом (обычно взаимосвязанных) клиентов.
- > **Предметная область.** Несколько продуктов в одной предметной области нередко разделяют значительную часть сходных функциональных возможностей (или реализаций), и поэтому они требуют знания этой предметной области. Таким образом, при расширении определения продукта исключается дублирование функциональных возможностей и допускается более точное назначение приоритетов. Но если определение продукта распространяется на многие предметные области и становится слишком широким, то специализация в отдельной предметной области не допускается, а следовательно, команды бесконечно осваивают новые предметные области, так и не создавая ничего конкретного. Таким образом, все элементы должны находиться в одной или нескольких вполне определенных предметных областях клиентов.

Вопросы общности приводят как к сужению, так и к расширению определения продукта.

Существующие структуры. Они также ограничивают широту определения продукта. Команды, берущие работу из задела, работают в одном и том же Спринте, вместе координируют работу и объединяют ее результаты, доставляя один интегрированный прирост продукта³. При внедрении LeSS необходимо изменить организационные структуры, но иногда существующие структуры препятствуют широкому определению продукта — по крайней мере, теперь. Двумя существующими структурами, которые могут препятствовать широкому определению, являются *компании* и *отделы*, вкратце рассматриваемые ниже.

³ Один интегрированный прирост продукта может состоять из нескольких поставляемых или даже продаваемых продуктов. Эта главная мысль здесь не развивается дальше.

- > **Компании.** Часть продукта может быть создана другой компанией. Это ограничивает определение продукта. Ниже перечислены три типичные разновидности ограничивающих структур, состоящих из нескольких компаний.
 - > **Наемные команды или разработка на внешнем подряде.** Команды из другой компании работают только над данным продуктом. Следовательно, они должны брать работу из одного и того же Задела Продукта в одном и том же Спринте. Но такое ограничение не должно долго накладываться на определение продукта. Не следует также передавать один компонент в работу другой компании, поскольку это приводит к плаванию структур, состоящих из команд для компонентов, со всеми их недостатками.
 - > **Специализированные компоненты.** Одна компания вносит специализацию в свой типичный продукт, который является компонентом другой компании. Они, вероятно, не могут брать работу из одного и того же Задела Продукта, поскольку у них много заказчиков. В таком случае нужно, по крайней мере, попытаться, чтобы они непрерывно интегрировали свой продукт и специальные компоненты в общую кодовую базу.
 - > **Обобщенные компоненты.** Одна компания создает обобщенный компонент, являющийся частью крупного продукта, или же пользуется обобщенным компонентом, созданным другой компанией. Но в любом случае команды, создающие компонент, вообще не будут брать работу из одного и того же Задела Продукта в одном и тот же Спринте.
- > **Отделы.** Внедрение LeSS обычно подразумевает структурные изменения. Но существующие структуры отделов могут оказывать влияние на величину вносимых изменений и ограничивать определение продукта. Допустим, приложение A выполняется на платформе X, которая построена для внутреннего употребления и над которой надстроено лишь несколько приложений. Если приложение A и платформа X организационно связаны вместе, они должны быть объединены при внедрении LeSS. Но ни приложение A, ни платформа X не являются отдельными продуктами, они — часть более широкого определения продукта. Но если разработкой этих продуктов руководит один общий руководитель, находящийся на пять уровней выше по иерархии (например, исполнительный директор), то решение по поводу их объединения может принять только он. А это было бы непрактично и помешало бы внедрению LeSS. Поэтому следует временно создать или продолжить пользоваться узкими определениями продуктов A и X, но попытаться расширить его со временем.

В предыдущем разделе упоминался следующий ключевой момент: определения продуктов могут изменяться со временем. Более подробно он рассматривается далее в отдельном наставлении.

См. далее раздел
“Наставление: расширение определения продукта”.

Наставление: определение продукта

Принятие решения относительно текущего определения продукта и потенциального его расширения в будущем является важным этапом внедрения LeSS. Как правило, это делается на стадии первоначального обсуждения, а иногда и совещания с более конкретной повесткой дня.

Подойдем к этому вопросу, исследовав сначала *расширяющие*, а затем *ограничивающие* факторы. Для этого необходимо предпринять следующие действия.

1. Расширить как можно больше определение продукта.

Необходимо выбрать тот продукт, который считается текущим, и задать следующие вопросы, касающиеся расширения его определения.

- > **Что ответят конечные потребители, если их спросить: “Что собой представляет ваш продукт?”** Этот вопрос исключает из рассмотрения внутренние технические продукты и усиливает сосредоточенность клиента на чем-то конкретном.
- > **Имеются ли у нас общие компоненты или аналогичные функциональные средства для текущих продуктов?** Этот вопрос побуждает к поиску семейств продуктов, которые могут рассматриваться как один.
- > **Является ли наш продукт частью другого продукта? Какую проблему конечных потребителей решает данный продукт?** Эти вопросы позволяют исследовать крупные продукты или системы, к которым относится рассматриваемый продукт.

2. Наложить практические ограничения на определение продукта.

Необходимо исследовать ограничивающие факторы, задав следующие вопросы.

- > **Какова концепция продукта? Кто его потребители? Какова предметная область потребителей данного продукта?** Эти вопросы позволяют исследовать общность, которая должна быть у продукта.
- > **Какая разработка ведется в нашей компании? Насколько практически структурные изменения в ней?** Эти вопросы позволяют исследовать структурные границы определения продукта.

3. Принять решение о первоначальном определении продукта.

Необходимо сравнить широкое определение продукта (результат выполнения п.1) с его практическим определением (результат выполнения п.2) и выяснить, каким должно быть подходящее определение продукта на будущее и какие изменения для этого потребуются. В конечном итоге должно быть сформулировано первоначальное определение продукта, а также высказаны идеи по поводу его расширения в будущем.

Примеры определений продуктов

Ниже приведены три примера определений продуктов. Однако они сильно упрощены, поскольку для подробного рассмотрения каждого из них потребовалась бы отдельная книга.

Финансовая торговля

Финансовая торговая группа зачастую организуется по типу финансового продукта (например, основным и производным ценным бумагам) и далее подразделяется на главное подразделение (для совершения сделок) и вспомогательное (для обработки финансовых операций). Каждое из этих подразделений занимается своим делом, а также поддерживает подразделение разработки.

В упрощенном виде жизненный цикл финансовой торговли выглядит следующим образом: ценообразование, регистрация сделки, проверка и обогащение, а также расчет. Для каждой стадии этого процесса имеется свой компонент (или приложение), например, компонент для ценообразования производных ценных бумаг или компонент для расчетов по основным ценным бумагам. Это, бесспорно, наполовину, а то и больше дублирует функции компонентов в разных типах продуктов.

Традиционная структура состоит из команд для компонентов, принимая во внимание, что компонент для расчетов по основным ценным бумагам является *продуктом* и т.д. и т.п. Но так ли это? Выясним это, задав следующие вопросы, *расширяющие и ограничивающие* определение продукта.

- > **Что ответят конечные потребители, если их спросить: “Что собой представляет ваш продукт?”** Вероятно, они ответят так: “Полноценное решение для финансовой торговли” и “Полноценное решение для торговли основными ценными бумагами”.
- > **Имеются ли у нас общие компоненты или аналогичные функциональные средства для текущих продуктов?** Да, имеются. В частности, компоненты *справочных и рыночных данных*, а возможно, и общие, но в настоящий момент разделенные компоненты с большой, хотя и скрытой степенью дублирования, например, компоненты для *финансовых расчетов, регистрации сделки* и т.д.
- > **Является ли наш продукт частью другого продукта?** Если заявлено, что компонент для расчетов по основным ценным бумагам является продуктом, то он относится к подразделению обработки финансовых операций, которое может входить в подразделение торговли основными ценными бумагами, а последнее — в более крупное подразделение финансовой торговли. Если пойти еще дальше, то продукт для финансовой торговли является частью системы финансовой торговли, включающей в себя несколько аналогичных продуктов и бирж.

Приведенные выше расширяющие вопросы указывают на определение такого продукта, который позволяет клиентам вести торговлю. Оно охватывает торговлю основными и производными ценными бумагами от главных до вспомогательных подразделений, а возможно, и еще больше. Каковы же ограничения? Для этого уделим внимание лишь следующим главным вопросам.

- > **Какая разработка ведется в нашей компании?** Вся разработка системы финансовой торговли ведется в нашей компании, но системы для бирж ценных бумаг — за ее пределами. Это решительно ограничивает определение продукта.
- > **Насколько практичны структурные изменения?** Подразделения основных и производных ценных бумаг находятся на самостоятельном балансе, а основные торговцы специализируются на одном финансовом продукте. Руководители этих подразделений не особенно озабочены эффективностью технологий, применяемых на уровне компании, если только их не принудит к этому исполнительный директор. Таким образом, было бы непрактично расширять определение за пределы отдельных типов финансовых продуктов. Технология разделения на главное и вспомогательное подразделения охватывает компоненты, которые, несомненно, должны входить в состав одного более обширного продукта, например, для торговли основными ценными бумагами от главного до вспомогательного подразделений. Но руководители отдельных подразделений окажут серьезное сопротивление, борясь за сохранение своих вотчин, и никто из высшего руководства организации пока еще не уделял внимания и не особенно интересовался их объединением. Таким образом, объединение главного и вспомогательного подразделений в настоящее время практически неосуществимо по политическим мотивам.

Из всего сказанного выше можно сделать вывод, что в большинстве ограничивающих ответов просматривается поддержка одного общего продукта для финансовой торговли, в состав которого не входят внешние биржи ценных бумаг. Но последний вопрос — *Насколько практичны структурные изменения?* — указывает на ограничения политического характера, еще больше сужающие определения продуктов. Следовательно, в качестве реалистичной отправной точки для определения продуктов могли бы послужить следующие продукты.

1. Торговля основными ценными бумагами в главном подразделении.
2. Торговля основными ценными бумагами во вспомогательном подразделении.
3. Торговля производными ценными бумагами в главном подразделении.
4. Торговля производными ценными бумагами во вспомогательном подразделении.

В будущем следующим шагом, вероятно, станет объединение продуктов для торговли основными ценными бумагами в главном и вспомогательном подразделениях в единый продукт для торговли основными ценными бумагами. Аналогичным образом может произойти объединение в единый продукт для торговли производными ценными бумагами.

Базовые станции телекоммуникационной сети

Базовая станция является составной частью телекоммуникационной сети, подключающей мобильный телефон к Интернету. Как правило, в разработке базовой станции принимают участие тысячи людей. Каждое поколение телекоммуникационной сети (2G-GSM, 3G-WCDMA, 4G-LTE, 5G) отличается своим вариантом базовой станции. У этих вариантов разные функциональные возможности работы на разном оборудовании. И в каждом поколении могут быть субварианты с отдельными технологиями для разных рынков. Например, для поколения LTE телекоммуникационной сети имеются субварианты TDD-LTE и FDD-LTE. Телекоммуникационная группа зачастую организуется вокруг таких субвариантов.

Один вариант базовой станции состоит из следующих главных компонентов: *платформы, приложения* и пр. Платформа является общей для нескольких вариантов и по традиции отделена от приложения.

На первый взгляд определение продукта кажется тривиальным. Но в результате исследования с помощью расширяющих и ограничивающих вопросов оно оказывается не таким уж и тривиальным. Ниже приведены лишь самые важные из этих вопросов.

- > **Что ответят конечные потребители, если их спросить: “Что собой представляет ваш продукт?”** Если конечным потребителем является оператор телекоммуникационной сети (например, AT&T), то он ответит так: “Один конкретный вариант базовой станции, например, TD-LTE”.
- > **Имеются ли у нас общие компоненты или аналогичные функциональные средства для текущих продуктов?** Да, имеются. Функциональные возможности вариантов сходны и разделяются общим компонентом платформы, при условии, что они составляют лишь один продукт.
- > **Какую проблему конечных потребителей решает данный продукт?** Отдельная базовая станция сама не доставляет никакой ценности или функциональные возможности без поддержки мобильного телефона и других компонентов телекоммуникационной сети. В действительности базовая станция является лишь компонентом крупной сети, а следовательно, расширенное определение продукта должно охватывать всю телекоммуникационную сеть.

Приведенные выше расширяющие вопросы наводят на мысль о том, что продуктом является вся телекоммуникационная сеть. Но специалистам в области телекоммуникации такая мысль показалась бы довольно странной. Поэтому выясним причины с помощью следующих ограничивающих вопросов.

- > **Какая разработка ведется в нашей компании?** Телекоммуникационная сеть потенциально состоит из элементов от разных производителей, а следовательно, ее нельзя рассматривать как единый продукт. Но все варианты базовых станций разрабатываются в одной компании, и поэтому имеет смысл определить продукт как *базовую станцию*, охватывающую все варианты.

> **Насколько практичны структурные изменения?** Разные варианты базовой станции разрабатываются в отдельных подразделениях, а у отдельных хранилищ исходного кода имеется общий источник, хотя они разделены. Объединить эти подразделения в краткосрочной перспективе будет трудно по политическим мотивам, а для объединения хранилищ исходного кода потребуется немало труда. Компонент платформы, разделяемый разными вариантами, также разрабатывается в отдельном подразделении, и поэтому организация будет поначалу противодействовать включению платформы в один из вариантов. Все это серьезно ограничивает определение продукта.

Какой же вывод можно сделать из всего сказанного выше? К сожалению, *первоначальное* определение продукта должно охватывать один вариант базовой станции, хотя и с неприятной внешней зависимостью от компонента платформы, что увеличивает внутреннюю динамику развития команд для компонентов. Но со временем определение продукта должно включать в себя платформу и должно быть расширено, чтобы охватить больше вариантов.

Банковское обслуживание через Интернет

Несмотря на краткость данного примера, он рассматривается здесь потому, что из него следует один важный вывод. Банковская группа первоначально рассматривала банковское обслуживание через Интернет как свой продукт. Но анализируя расширяющие вопросы, они быстро поняли, что банковское обслуживание через Интернет само по себе не было вообще продуктом. Напротив, это был лишь один из путей к их реальному продукту — *базовым банковским услугам*. Как только они выявили, что банковское обслуживание через Интернет оказалось в действительности лишь компонентом, то поняли: в нем скрывается корень всех ненужных усложнений, в том числе синхронизации продуктов, управления программами и портфелем проектов.

К сожалению, ограничивающие факторы выявили трудности политического характера для объединения банковской группы в пределах реального продукта, а следовательно, первоначальное определение их продукта по-прежнему ограничивалось лишь банковским обслуживанием через Интернет. Но в данном случае это не было определением их продукта, которое следовало расширить. Вместо этого нужно было расширить определение продукта базовых банковских услуг таким образом, чтобы включить в него банковское обслуживание через Интернет.

Наставление: расширение определения продукта

В предыдущих наставлениях было установлено, что определение продукта — это выбор, который позволяет выяснить, насколько ограничивающие факторы способны сделать первоначальное определение менее совершенным. Этим открывается путь для применения определения продукта в качестве средства для непрерывного совершенствования.

В течение срока действия продукта организация должна постоянно задаваться следующим вопросом: что мешает нам расширить определение продукта? Ответы на этот вопрос позволяют выяснить мероприятия для будущих организационных улучшений. В этом смысле определение продукта играет такую же роль, как и Критерий Готовности, за исключением того, что расширить определение продукта труднее, поскольку организационные улучшения действительно оказывают более заметное влияние на различные подразделения, их собственные цели, доходы и расходы и принятые правила.

Наставление: продукт вместо проекта или программы

Вследствие широкой распространенности управления проектами в большинстве компаний, по-видимому, считают, что вся работа должна быть организована вокруг проектов или программ. Но что определяет проекты? Проект имеет конкретную дату начала, конкретную дату окончания, которая зачастую не соблюдается, а также нечто вроде жестко заданного объема работ. Решения, отслеживание состояния и составление бюджета основываются на краткосрочной цели проекта. Всякий проект завершается выпуском, и он готов тогда, когда действительно готов.

Но совсем другое дело — продукты!

Продукты, напротив, имеют более четкую дату начала, неясную дату завершения и ясное назначение с неясным и постепенно развертывающимся объемом работ. Они существуют дольше, чем можно было бы ожидать! У одного продукта имеется ряд выпусков, которые представляют собой лишь моменты времени, в которые продукт поставляется заказчику. Все примеры продуктов почти полностью состоят из разработки программного обеспечения. К их числу относятся не только такие очевидные продукты, как программное обеспечение для массового рынка или оперативное обслуживание через Интернет, но и менее очевидные *внутренние* продукты вроде торговых систем. Таким образом, продукты *переживают* проекты.

Люди настолько привыкли к проектами, что просто забывают обратить внимание на то, что проекты во многих компаниях, по существу, служат для построения и расширения продуктов. Но это ошибка! Управлению продуктами как проектами присущи серьезные недостатки, в том числе следующие.

1. Решения, принимаемые на краткосрочную, а не долгосрочную перспективу.
2. Нереальные зачастую бюджетные процессы.
3. Дополнительные издержки на начало и завершение проектов.
4. Временные команды или даже временные работники.

Управление продуктами недопустимо с помощью проектов или программ!

Управлять продуктами в LeSS предпочтительнее как *продуктами*. Это означает, что на весь срок действия продукта назначается один Владелец Продукта и один Задел Продукта. К преимуществам такого подхода относятся следующие:

1. Правильный взгляд на разрешение компромиссов в краткосрочной или долгосрочной перспективе.
2. Финансиование, исходя из будущей ценности продукта, а не конкретных функциональных средств.
3. Исключение проектных и программных структур и связанных с ними издержек.
4. Устойчивые в долгосрочной перспективе команды.

Об определении продукта в сравнении с проектом и его влиянии на организацию можно было бы сказать намного больше. Но для этого, вероятно, пришлось бы написать отдельную книгу.

LeSS Huge

Предпочтение, отдаваемое более широким определениям продуктов, приводит к более частым примерам внедрения LeSS Huge, чем следовало ожидать. В тех случаях, когда прежние более узкие определения продуктов оказываются хотя бы ориентированными на клиентов, эти мелкие продукты могут стать Областями Требований для одного крупного продукта. Например, в упоминавшемся ранее продукте для финансовой торговли Областями Требований могли бы стать торговые операции с основными и производными ценными бумагами.



Команда Владельцев Продукта назначает приоритеты в группе, практикующей LeSS Huge

8

Владелец Продукта

*Я лично всегда готов учиться, хотя мне
не всегда нравится, когда меня учат.
— Уинстон Черчиль*

Scrum для одной команды

Один Владелец Продукта отвечает за концепцию качественного продукта для клиентов и его оптимизацию (окупаемость инвестиций и прочее). Владелец Продукта непрерывно развивает Задел Продукта, вводя, удаляя и переназначая приоритеты (т.е. переупорядочивая элементы), а также опираясь на обучение и адаптацию к изменениям. И он поддерживает прозрачность Задела Продукта, сохраняя его доступность для высшего руководства, Команды и клиентов. Владелец Продукта сотрудничает с Командой и клиентами, чтобы прояснить элементы. Он адаптивно решает, какие именно элементы следует предложить в каждом новом Спринте, хотя только Команда решает, сколько их необходимо выбрать. Но только Владелец Продукта может дать работу Команде. А Scrum-мастер наставляет Владельца Продукта относительно его роли и обязанностей.

Роль Владельца Продукта в LeSS

При масштабировании соблюдаются следующие принципы, связанные с ролью Владельца Продукта.

- > **Сосредоточение внимания на продукте в целом.** При крупномасштабной разработке продукта нетрудно организовать “песочницы”, где каждому отводится своя роль. Один Владелец Продукта с одним Заделом Продукта поддерживает сосредоточенность на всем продукте.
- > **Бережливый подход с целью избежать перегруженности.** Как одному Владельцу Продукта равномерно распределить свою нагрузку, работая со многими командами? Решению этого вопроса посвящены многие наставления в этой главе, в том числе поручение командам большей части

работы по прояснению элементов из Задела Продукта, а также поддержание непосредственной связи команд с потребителями и пользователями.

- > **Крупномасштабная инфраструктура SCRUM — это тоже SCRUM; системный подход.** Реальное внедрение Scrum подразумевает окончание *игры в контракты* — традиционной модели, где внутренний контракт с жестко заданным объемом работ и сроками их выполнения сначала заключается между сторонами, относящимися к деловой части и разработке, после чего сторона, отвечающая за разработку, выполняет проект по контракту. Откровенно говоря, Scrum — не намного более эффективный механизм для выполнения внутреннего контракта. Это скорее смещение понятий в сторону сотрудничества с клиентами и адаптации, доставки ценности в каждом Спринте, а кто-нибудь из тех, кто отвечает за деловую часть контракта и принимает решения, берет на себя роль Владельца Продукта. При крупномасштабной традиционной разработке такая игра в контракты закрепляется в организационной структуре, где отдел управления программами отвечает за внутренний контракт и множество связанных с ним положений, правил и процессов. А при внедрении инфраструктуры LeSS в такой среде она неверно рассматривается как нечто, вписывающееся в текущую модель, а не заменяющее ее. И как следствие, возникает неверное представление, что роль Владельца Продукта должен выполнять руководитель программы или проекта.
- > **Добиваться большего средствами LeSS.** Роль Владельца Продукта в LeSS может эффективно выполнять один человек. Благодаря этому сокращается количество ролей, должностей и дело упрощается в целом.

Правила LeSS

Для всего готового к поставке продукта назначается один Владелец Продукта и один Задел Продукта.

Владелец Продукта не должен работать в одиночку над уточнением Задела Продукта, поскольку этим в основном занимаются многие Команды, непосредственно сотрудничающие с потребителями, пользователями и прочими заинтересованными сторонами.

Весь процесс назначения приоритетов (упорядочения) проходит через Владельца Продукта, но прояснение в как можно большей степени осуществляется между Командами и потребителями, пользователями и прочими заинтересованными сторонами.

Наставление: кто должен быть Владельцем Продукта

Где искать кандидата на роль Владельца Продукта в группе, внедряющей LeSS? С этой целью необходимо выполнить следующие действия.

1. Знать свой тип разработки

Место, где искать кандидата на роль Владельца Продукта, определяется типом разработки. Основные примеры таких мест приведены на рис. 8.1.

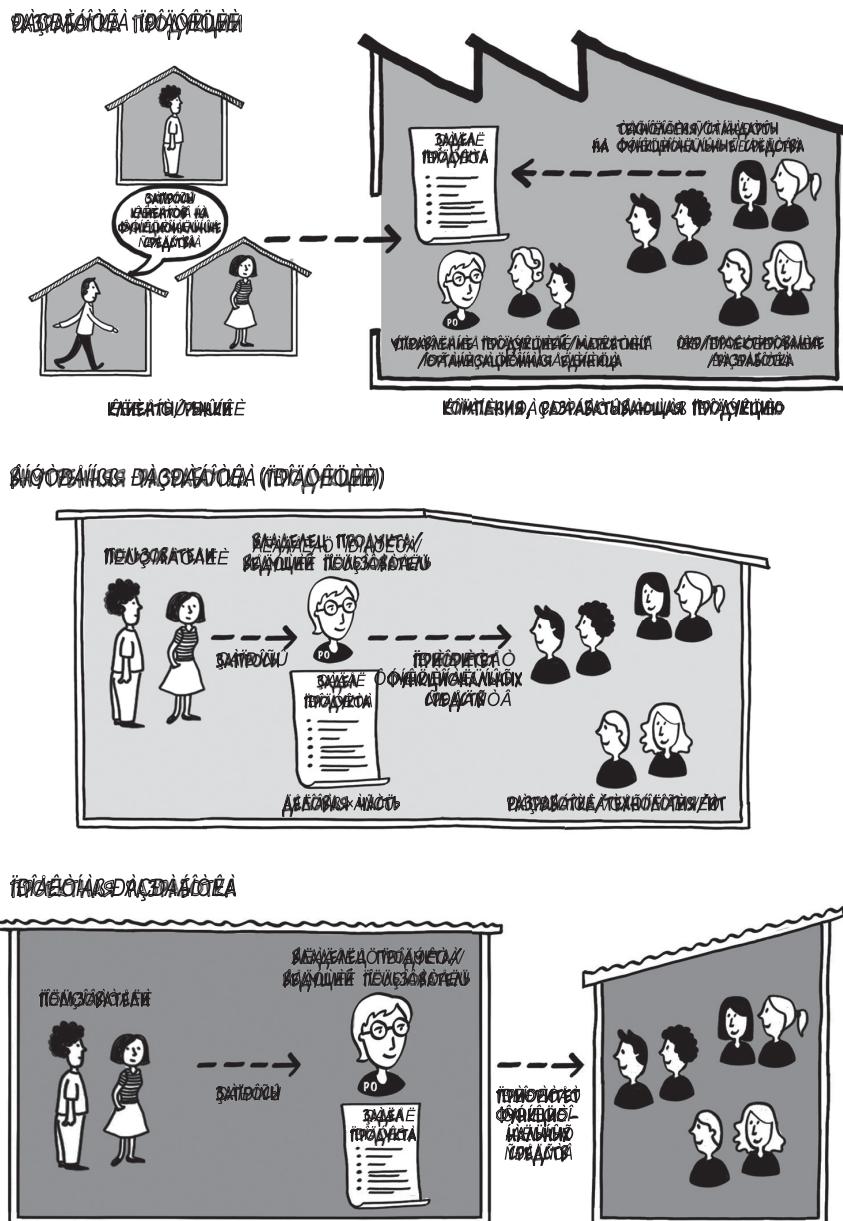


Рис. 8.1. Места для поиска кандидата на роль Владельца Продукта в зависимости от типа разработки

Разработка продукции ориентирована на внешних потребителей или рынки.

Внутренняя разработка (продукции) ориентирована на одну или несколько групп в компании. Группа разработки называется *ИТ* (Информационные технологии), *технологической* или группой *разработки систем*.

Дополнительные сведения см. также по адресу <http://agilecontracts.com/>

Проектная разработка, как правило, ориентирована на внешнего потребителя. Работа организована и ведется по контракту как некоторый проект, хотя это совсем не означает контракт на проект с жестко заданным объемом работ, сроками или затратами. Как правило, компания, разрабатывающая продукцию, является внешним подрядчиком или системотехническим предприятием. А в компании-заказчике имеются потребители и пользователи, которые оплачивают разработку, хотя они не всегда находятся в одном и том же подразделении.

2. Найти Владельца Продукта

Разработка продукции ведется в компании, где имеется, во-первых, организационная единица, выдвигающая инициативу по поводу разработки продукции (например, для банковского обслуживания мелких клиентов), а во-вторых, подразделение для разработки продукции. На традиционное управление продукцией возлагается ответственность за анализ потребительского рынка и конкурентов, выработку концепции продукции, грубый отбор функциональных средств и назначение приоритетов, составление графика выпуска продукции и пр. К этому не относится руководство работой традиционной группы разработки, поскольку это обязанность управления разработкой, где берутся обязательства на выполнение работ крупными партиями и в заданные сроки, а также координируются действия команд и пр.

Где же искать Владельца Продукта для группы, внедряющей LeSS? Если имеется подразделение для управления продукцией, то на эту роль вполне подходит *ответственный за выпуск продукции*. В противном случае это может быть представитель организационной единицы, выдвигающей инициативу по поводу разработки продукции.

Внутренняя разработка (продукции). Владелец Продукта, подходящий для LeSS, может быть выбран, во-первых, из той группы, где предполагается применять разрабатываемую систему, а во-вторых, он может быть человеком, вполне причастным и имеющим немалый практический опыт работы, которую данная система должна поддерживать. Такие кандидаты на роль Владельца Продукта очень близки к реальным пользователям. И как только они станут Владельцами Продукта, им потребуются серьезные полномочия, чтобы принимать независимые решения относительно продукта.

Проектная разработка (на внешнем подряде). Самое главное, что Владелец Продукта выбирается из компании, принимающей разрабатываемую систему. А при внутренней разработке это должен быть человек, причастный, имеющий немалый практический опыт и тесно связанный с пользователями.

Для внутренней и проектной разработки типичным является случай, когда система будет использоваться во *многих* подразделениях. В таком случае на роль Владельца Продукта вполне подойдет представитель одного из основных подразделений пользовательской организации, политически опытный и готовый взять на себя эту роль.

И, наконец, человек, вполне подходящий на роль Владельца Продукта, в любом случае должен иметь пристрастие к продукту, политический опыт и личное обаяние. Против такой кандидатуры вряд ли стоит возражать!

3. Определить полномочия и обязанности Владельца Продукта

Роль Владельца Продукта не нова для традиционного руководителя проекта или программы, отвечающего за выполнение заданного объема работ в указанные сроки по контракту, или для специалиста в предметной области. Тем не менее Владелец Продукта должен иметь независимые полномочия, чтобы принимать ответственные деловые решения, выбирать и изменять содержимое продукта, даты выпуска, приоритеты, концепцию продукта и т.д. И, конечно, он должен сотрудничать с заинтересованными сторонами, но окончательное решение должно оставаться за ним.

Рекомендация для многоместной разработки: быть ближе к клиентам, чем к командам

Если оставить в стороне случай выхода на глобальный массовый рынок, то для Владельца Продукта еще важнее быть ближе к потребителям и пользователям, чем к командам. Владелец Продукта должен находиться в одном месте с потребителями и пользователями, а не с командами, иначе его внимание будет сосредоточено внутри и меньше будет уделяться потребителям и пользователям.

Как следствие, Владельцу Продукта придется проводить многоместные совещания (например, по Первому Планированию Спринта) с удаленными командами. Такие совещания относительно эффективны при наличии инструментальных средств для видеоконференций, чему мы были неоднократно свидетелями.

Наставление: начало внедрения с временным заместителем Владельца Продукта

Как правило, инициатива внедрить LeSS принадлежит группе разработки. Допустим, эта группа решает найти сначала подходящего Владельца Продукта из деловых кругов, а затем приступить к внедрению. Такой подход может привести к следующим потенциальным осложнениям.

- > **Задержанное начало.** Представители деловых кругов, не являющиеся инициаторами изменений, обычно ссылаются на большую занятость, когда их просят принять участие в каком-нибудь крупном мероприятии. Кроме того, они не понимают никаких выгод и не знают, как действовать в роли Владельца Продукта. Поэтому на поиски подготовленного кандидата на эту роль придется потратить время.
- > **Неорганизованное начало.** Когда группа начинает свой первый (или второй) Спринт, результаты могут оказаться самыми разными: от слабой организации до полного краха. Все это может вскрыть немало проблем. В лучшем случае начинающий Владелец Продукта из деловых кругов разберется

в ситуации, набравшись терпения, а в худшем — просто обнаружит полный беспорядок. Какой же он сделает вывод? LeSS только усложняет дело, и он не понимает, зачем вообще ему участвовать на этой ранней стадии, когда еще ничего не сделано.

Таким образом, единственный выход в данном случае — быстро приступить к внедрению LeSS вместе с временным заместителем Владельца Продукта, хорошо понимающим, что происходит, и способным хотя бы в общих чертах выполнять свою роль. Но в то же время он не должен быть из деловых кругов и не должен иметь специальные знания в деловой сфере и отвечать за окупаемость инвестиций. Группа должна вместе с ним провести несколько Спринтов, чтобы преодолеть первые муки творчества, выдавая, что очень важно, подлинный (или хотя бы близкий к нему) прирост готового к поставке продукта в каждом Спринте. Почему это так важно? Дело в том, что, приглашая деловых людей принять участие в создании продукта вместе с настоящим Владельцем Продукта, разработчики могут показать им новые привлекательные возможности, сулящие весьма ощутимую коммерческую выгоду. И это действительно привлекает!

Крайне важно, чтобы все понимали, что временный заместитель не является настоящим Владельцем Продукта. Он должен быть замещен при первой возможности. Все дает основание назвать такого заместителя *ненастоящим Владельцем Продукта*.

Наставление: кем являются потребители и пользователи

Под *потребителями* подразумеваются те, кто приобретает или выбирает продукцию или же принимает непосредственное участие в принятии коммерческих решений относительно разрабатываемой продукции. Понятие *пользователей* оказывается несколько более сложным, особенно при крупномасштабной разработке, где организационная разобщенность не дает разработчикам точно знать, кто пользователи их продукции. Под *пользователями* обычно понимаются, хотя и не всегда, те, кто применяет разрабатываемую продукцию на практике. Кто же они и где находятся, а точнее, от кого исходят требования и кто должен утверждать создаваемые функциональные средства и давать свои отзывы о них? Ответы на эти вопросы сведены в приведенную ниже таблицу.

Тип разработки	Подтип разработки	Кто служит источником требований ¹	Кто утверждает и дает отзывы
Разработка продукции	Новаторская разработка под влиянием новых технологий и с соблюдением стандартов	Реальные пользователи или их заместители не предъявляют требования. Вместо этого требования исходят изнутри от тех, кто отвечает за выпуск продукции, включая Владельца Продукта, членов команды и т.д.	Псевдопользователи: Кандидаты в пользователи, внутренние добровольцы и пользователи прежних продуктов

Продолжение таблицы

Тип разработки	Подтип разработки	Кто служит источником требований ¹	Кто утверждает и дает отзывы
Разработка продукции	Разработка по требованиям потребителей и их массового рынка	Заместители пользователей вроде ответственных за выпуск продукции, членов команд и прочих специалистов, представляющих потребителей и ориентированных на рынок. В качестве источника требований следует использовать целевую группу кандидатов или существующих пользователей	Источник требований
Разработка продукции	Разработка по требованиям потребителей, число которых не превышает 50	Практические пользователи среди многих потребителей	Источник требований
Внутренняя разработка	Регулярная	Внутренние практические пользователи	Источник требований
Внутренняя разработка	Инициатива специальных изменений (например, в нормативах)	Источник специальных изменений (например, директивный или регламентирующий орган)	Источник требований
Проектная разработка		Практические пользователи из числа потребителей, оплачивающих разработку	Источник требований

¹ Это лишь иллюстративное представление, не претендующее на исчерпывающую полноту.

В большинстве перечисленных выше случаев цель LeSS — существенно укрепить непосредственное сотрудничество разработчиков с теми, кто являются *подлинным источником* требований, чтобы прояснить эти требования¹. Но для этого потребуются немалые перемены в умонастроении и поведении, ведущие к ломке сложившегося положения дел и процессов. Поэтому Владельцу Продукта необходимо заранее предусмотреть замену устаревших структур, чтобы служить *связующим звеном* между разработчиками и пользователями.

¹ Здесь и далее под *пользователями* понимаются люди, имеющие непосредственное отношение к практике, или их разновидности.

Наставление: назначение приоритетов вместо прояснения

С ролью Владельца Продукта в Scrum связаны следующие *информационные потоки*.

1. Адаптивный выбор *направления* для развития продукта и отражение этого выбора в назначении приоритетов в Заделе Продукта.
2. Выявление и прояснение *подробностей*, касающихся потребностей пользователей и элементов.

В первом потоке (выбор направления и назначения приоритетов) ищется и анализируется информация, связанная со стимуляторами прибыли, стратегическими потребителями, коммерческими рисками и т.д. А во втором потоке (подробности и прояснения) преследуется цель обнаружить точное поведение и свойства элементов, взаимодействие с пользователем и т.д.

Владелец Продукта должен уделить немало внимания осмыслианию выбора направления и назначения приоритетов, но в то же время поручить командам выявить как можно больше подробностей. Он должен поощрять и помогать командам идти на прямой диалог с пользователями, действуя в качестве *связующего звена*, а не посредника. Короче говоря, Владелец Продукта уделяет основное внимание назначению приоритетов, а не подробному прояснению, которое поручается командам.

Наставление: чего не должен делать Владелец Продукта

В связи с изложенным выше возникает следующий вопрос: может ли *один* человек эффективно работать над продуктом с шестью командами, многими требованиями и многочисленными заинтересованными сторонами?

Владелец Продукта может легко оказаться перегруженным. Он ведет разработку в соответствии с концепцией продукта и принимает активное участие в следующем.

- > Выбор направления и назначение приоритетов, чтобы определить дальнейшие пути развития.
- > Выработка концепции продукта, его развития и внедрение технологий, чтобы получить представление в долгосрочной перспективе.
- > Установление отношений и правил, чтобы все были довольны.
- > Оценивание и прогнозирование, чтобы оценивать рынки и конкурентов.

Это главные обязанности, которым Владелец Продукта должен уделить основное внимание. Но ему придется также уделить время следующему.

- > Прояснение, чтобы выявить подробности назначения элементов.
- > Административная работа, чтобы составлять отчеты и отслеживать количественные показатели.

- > Координирование действий разных подразделений, чтобы согласовать производство, продажи и пр.
- > Изучение рынков, технологий и конкурентов.

Эти обязанности желательно поручить командам. А сам Владелец Продукта не должен брать на себя следующие обязанности.

- > Управление зависимостями или координация действий команд.
- > Прогнозирование и планирование работы команд.
- > Анализ оценок.
- > А в более общем смысле — обмен информацией между людьми.

Наставление: помощники Владельца Продукта

Исходя из последнего наставления, Владелец Продукта должен делиться своей работой. Но с кем?

Команды. Прежде всего, он должен воспользоваться услугами команд. Ему нужно разрушить стены между разработкой и управлением продукцией, постепенно привлекая команды к деловой сфере. Он должен не только делиться своей работой, но и привлекать команды видеть целое. Так, если никто раньше не предлагал команде научиться исследовать рынок, Владелец Продукта должен попытаться это сделать! Это мероприятие он может запланировать, введя соответствующие элементы в Задел Продукта. И как упоминалось выше, Владелец Продукта может поручить командам прояснить элементы и встретиться с пользователями.

Ответственные за выпуск продукции. Если Владелец Продукта входит в состав группы управления продукцией, он может обратиться за помощью к другим ответственным за выпуск продукции.

Ответственный за выпуск или координатор. Поставкой крупных продуктов, в том числе подготовкой к поддержке потребителей, продажами и производством, могут заниматься самые разные подразделения. В традиционных крупных группах координацией подобных действий занимается ответственный за выпуск. Если при внедрении LeSS по-прежнему остаются незавершенные работы, включающие координацию действий подразделений, этим *не* должен заниматься Владелец Продукта. Если объем этих работ невелик, их можно поручить постоянному члену команды. Но если объем этих работ большой (например, поставка телекоммуникационного оборудования), то ими все свое рабочее время должен заниматься ответственный за выпуск — по крайней мере, до тех пор, пока не появится совершенный Критерий Готовности. А кроме того, ответственный за выпуск служит и оказывает поддержку Владельцу Продукта и командам, а не наоборот.

Если помощник занят все свое рабочее время (например, координатор выпуска телекоммуникационного оборудования), то он входит в состав Команды Владельцев Продукта. Основное

См. подраздел “2. Выявить действия, которые можно выполнять в каждом Спринте” главы 10.

См. далее подраздел “Команда Владельцев Продукта”.

значение термина *Команда Владельцев Продукта* раскрывается в инфраструктуре LeSS Huge, где он обозначает коллектив, состоящий из общего Владельца Продукта и Владельцев Продукта во всех Областях требований.

Внимание! В состав Команды Владельцев Продукта не должны входить аналитики, составители спецификаций, архитекторы, проектировщики взаимодействия с пользователем или пользовательского интерфейса, иначе текущие проблемы и структуры останутся под новой вывеской. Все подобные специалисты должны входить лишь в состав обычных команд для функциональных средств.

Наставление: пять взаимоотношений

Владелец Продукта должен уяснить пять основных взаимоотношений, существующих в крупномасштабных группах разработки (рис. 8.2).

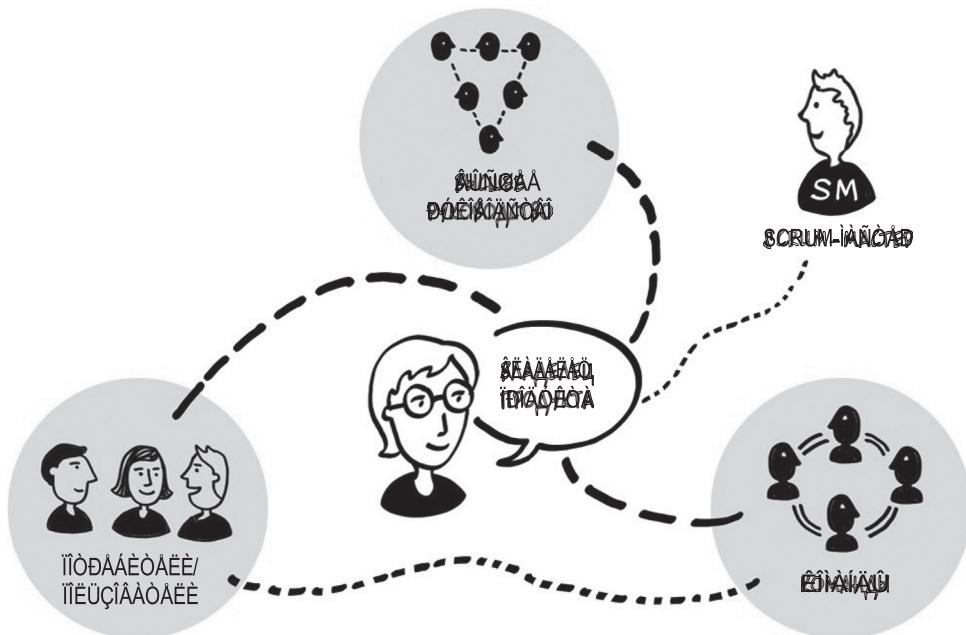


Рис. 8.2. Пять взаимоотношений с Владельцем Продукта

Многие группы, внедряющие LeSS, быстро усваивают взаимоотношения между Владельцем Продукта и Командами, но не особенно обращают внимание на другие

взаимоотношения, даже если они важны для успешной работы Владельца Продукта. Поэтому эти взаимоотношения поясняются в последующих подразделах.

Взаимоотношения Владельца Продукта с командами

Традиционные группы сильно разобщены. Нам приходилось работать с группами, в которых управляющим продукцией или коммерческой деятельностью, из числа которых назначается будущий Владелец Продукта, никогда не приходилось работать с разработчиками или даже видеть их. Вследствие этого возникает недостаток доверия или понимания с обеих сторон, когда они собираются вместе. В частности, Владелец Продукта должен знать, что разработчики гордятся созданием функциональных средств и с удовольствием этим занимаются. Они способны на многое, если понимают назначение продукта и непосредственно общаются с его пользователями.

От Владельца Продукта команды должны узнать о самом продукте и состоянии рынка, а также о том, что еще предстоит создать. А к Владельцу Продукта должна поступать информация о потребностях команд и той помощи, которую он может им оказать.

По поводу взаимоотношений Владельца Продукта с командами можно дать следующие рекомендации.

- > **Совместное владение.** Несмотря на то что роль Владельца Продукта выполняет один человек, структуры и культура правильно организованного коммерческого предприятия побуждают всех иметь внутреннее чувство владения и ощущение того, что это и их продукт. Владелец Продукта воспитывает подобное чувство, ища участия Команд во время общего Уточнения Задела Продукта и Подведения Итогов Спринта и обращаясь к ним за помощью в решении задач, носящих более коммерческий характер.
- > **Равноправие, а не слепое подчинение.** Если команды отчитываются прямо или косвенно перед Владельцем Продукта как подчиненные перед своим непосредственным начальником, то подобная организационная структура должна быть изменена таким образом, чтобы команды и Владелец Продукта сотрудничали как равные. Владелец Продукта не должен рассматривать команды как слуг, которым он дает поручения, но обязан воспитывать взаимоотношения сотрудничества с ними.
- > **Обращение к командам за помощью.** Если Владелец Продукта слишком обременен неотложными задачами управления продукцией, он может обратиться к командам, в которых найдется немало толковых людей, готовых ему помочь.

См. раздел “Наставление: общее Уточнение Задела Продукта” главы 11.

См. раздел “Наставление: организационная структура LeSS” главы 4.

См. раздел “Наставление: теории управления” главы 5.

См. главу 13.

- > **Построение доверительных отношений.** Основой доверия служит прозрачность, которая должна проявляться в поведении Владельца Продукта и состоянии Задела Продукта. В частности, Владелец Продукта должен пояснить назначение работы и мотивы ее приоритетности, но в тоже время разрешить их критический анализ. Он должен также пояснить сложившиеся стесненные обстоятельства, не перекладывая их разрешение на команды, и поинтересоваться, какая помошь им требуется. Это в большей степени будет способствовать установлению отношений доверия и доброй воли, чем попытка свалить всю работу на команды.
- > **Оказание помощи, но не всегда.** Оказание помощи командам, когда они просят о ней, позволяет разрешить затруднения и создает доверие. Но что, если команды попросят Владельца Продукта о координации действий, что входит в их обязанности в LeSS? (Между прочим, это весьма характерно для новых команд.) В таком случае Владелец Продукта должен отказать в помощи, пояснив причины при содействии Scrum-мастеров.
- > **Никакого микроуправления.** Во время Спринта команды переходят на самоуправление, чтобы достичь поставленных целей. Владелец Продукта не отслеживает их продвижение к цели, не поручает задачи людям и т.д. Хотя он может и оказывать им помощь.
- > **Ретроспектива.** Не следует рассматривать Общую Ретроспективу как необязательное событие для Владельца Продукта. Он должен участвовать в нем, обучаясь у других улучшению взаимоотношений.
- > **Посещение мест нахождения команд.** Периодически Владелец Продукта посещает места нахождения команд, участвуя в событиях, происходящих в течение Спринта. Помимо эффективности личного участия в совещаниях, у Владельца Продукта появляется больше возможностей для общения и расширения кругозора и согласования. Это будет также способствовать установлению отношений доверия и доброй воли, если Владелец Продукта не прибегает к микроуправлению. Посещая места нахождения команд, он сможет лучше понять сложившееся положение дел в удаленных командах, а они — общую ситуацию. Когда он уедет и в дальнейшем будет проводить совещания с удаленными командами в режиме видеоконференций и обмена сообщениями, он сможет лучше поддерживать связь с ними.

Взаимоотношения Владельца Продукта с потребителями и пользователями

Традиционным группам присуща сильная разобщенность и слабая ответная реакция. Нам приходилось работать с крупными группами, где новый Владелец Продукта раньше не общался непосредственно с пользователями, а если и общался, то делал это редко, не повторяя попыток и не особенно рассчитывая на ответную реакцию. Роль Владельца Продукта в LeSS предполагает поощрять пользователей к участию вместе с командами в циклах обучения, опираясь на частые поставки прироста продукта, прозрачность и обследование.

От Владельца Продукта потребители и пользователи должны узнать, когда и как работа над продуктом скажется на них (в хорошем смысле), а возможно, и об аргументации для назначения приоритетов. Их нужно привлекать, ничего не скрывая и сохранивая полную прозрачность отношений. А *самому Владельцу Продукта* нужно узнать у потребителей и пользователей их настоящие цели или трудности, возможно, даже заглянув дальше, чем они видят, а также получить сведения, которые помогут ему в назначении приоритетов.

По поводу взаимоотношений Владельца Продукта с потребителями или пользователями можно дать следующие рекомендации.

- > **Обучение.** Владелец Продукта должен пояснить, как и почему изменения в LeSS принесут выгоду потребителям и пользователям, и в этих изменениях они могут принять участие. К этому относятся все новые запросы, которые в конечном итоге должны доходить до Владельца Продукта, а не непосредственно до групп разработки через прежние сети заказчиков. Чтобы научиться своевременно сообщать об этом, Владелец Продукта должен обратиться за помощью к Scrum-мастерам.
- > **Участие пользователей.** Платящих заказчиков и практических пользователей следует привлекать к участию в Ретроспективах Спринтов, непосредственном обсуждении при Уточнении Задела Продукта вместе с командами и Владельцем Продукта.
- > **Поставки, по крайней мере, в каждом Спринте.** Ценные функциональные средства следует поставлять в каждом Спринте или еще чаще, если только добиться этого в настоящий момент невозможно или неуместно. Вместе с командами Владелец Продукта устраниет препятствия, которые мешают потребителям пользоваться новым приростом продукта в каждом Спринте.
- > **Повышение прозрачности.** Например, Владелец Продукта должен пояснить содержимое Задела Продукта и аргументацию для назначения приоритетов. Он должен быстро уведомить потребителей, когда изменения окажут влияние на них.

Взаимоотношения команд с потребителями и пользователями

Традиционные команды разработки в крупных группах редко взаимодействуют с платящими заказчиками и пользователями. Грамотный Владелец Продукта добивается, чтобы команды заботились о создании качественного продукта для заказчиков. Для этого требуется чуткость и непосредственная связь исполнителей с заказчиками. От потребителей и пользователей командам нужно узнать контекст и подробные сведения о разрабатываемых функциональных средствах без обмана и рассеяния информации. В идеальном случае команды вырабатывают решения вместе с потребителями, уясня их актуальные (а не кажущиеся) цели и нужды. А самим потребителям команды должны подтвердить, что они полностью уяснили их цели, нужды и требования, которые они прояснили вместе.

Даже наставляя группы в LeSS, мы, к сожалению, замечали старую привычку избегать непосредственного взаимодействия команд и пользователей. Иногда ненастоящая Команда Владельцев Продукта организуется из аналитиков, проектировщиков взаимодействия с пользователем и пользовательского интерфейса для прояснения вместе с пользователями неясных вопросов, хотя это создает еще больше трудностей при передаче работ. Почему так происходит? Помимо местничества и боязни узкого специалиста присоединиться к настоящей команде для функциональных средств, иногда командам неудобно прояснять вопросы вместе с заказчиками в силу прежнего отшельнического умонастроения и недостатка навыков. Еще одной причиной служит вера, что было бы эффективнее, если бы кто-нибудь один составил спецификацию, что открывает путь к локальной оптимизации. Имеются также опасения, что рамки проекта расплузутся в результате открытого обсуждения командами и пользователями. Иногда это объясняется тем, что Владелец Продукта умеет составлять спецификацию и не привык поручать эту работу другим.

Чтобы извлечь наибольшую выгоду из LeSS, очень важно избегать подобного поведения и активно устанавливать тесную связь команд с пользователями. По поводу взаимоотношений команд с потребителями и пользователями можно дать следующие рекомендации²:

- > **Роль связующего звена.** Владелец Продукта поощряет и настраивает потребителей и пользователей взаимодействовать непосредственно с командами при Уточнении Задела Продукта и во время Ретроспективы Спринта (когда они учатся пользоваться готовыми функциональными средствами), проводит исследования практической пригодности и обследования на рабочих местах³, посещает места установки продуктов, учебные занятия и т.д.

² В этих рекомендациях основное внимание уделяется улучшению Владельцем Продукта взаимоотношений команд с потребителями и пользователями. А другие рекомендации относительно взаимоотношений команд с потребителями и пользователями приведены в главах 11 и 13.

³ Когда Крейг поступил на свою первую работу в страховую компанию в 1970-е годы, от разработчиков программного обеспечения требовалось уделять время общению с практическими пользователями на их рабочих местах, помогая им в работе, чтобы лучше понять их контекст и нужды.

- > **Привлечение к деловой активности.** Владелец Продукта приглашает разработчиков посещать коммерческие подразделения, общаться с аналитиками деловой активности, маркетологами и т.д.
- > **Обучение общению с заказчиками.** Кто-нибудь может возразить, что разработчикам не следует общаться с заказчиками, поскольку они могут наговорить много глупостей. Это справедливо, но вполне устранимое опасение. Владелец Продукта должен организовать для разработчиков, если требуется, курсы “ликбеза” по общению с заказчиками.
- > **Партнерские отношения с группой по отношениям с клиентами.** Иногда организуется группа, которая считает, что в ее обязанности входит налаживание отношений с клиентами. Обучение и установление партнерских отношений с такой группой станет неплохим планом действий для Владельца Продукта, чтобы способствовать тесным связям команд с потребителями и пользователями. Если это происходит медленно, то не следует ожидать, что намеченные организационные меры помогут наверстать упущенное. Владелец Продукта должен сам установить тесную связь команд с потребителями и пользователями, преодолевая традиционные границы.
- > **Исключение посредников.** В традиционных коммерческих подразделениях привыкли пользоваться услугами подгрупп, состоящих из аналитиков деловой активности, проектировщиков взаимодействия с пользователем и пользовательского интерфейса или ответственных за изменения, которые собираются вместе и составляют требования. Роль этих специалистов *действительно* полезна, но только в качестве полноценных членов команд для функциональных средств, а не посредников. Вместе со Scrum-мастерами и поддерживающими руководителями Владелец Продукта должен обеспечить такие изменения организационной структуры для внедрения LeSS, которые позволили бы перейти к настоящим командам для функциональных средств, исключая подобные подразделения, группы и отдельные функции и тем самым упрощая структуру организации.

Взаимоотношения Владельца Продукта с высшим руководством

Как показывает наш опыт, в традиционных группах обычно наблюдается настоящая подотчетность и ответственность одного лица за успехи или неудачи при создании продукта. Так, получив из управления продукцией перечень требований год назад, отдел разработки реализовал не очень много из этого перечня, а отдел продаж выдал нереалистичные обещания, что в конечном итоге взбесило высшее руководство.

А в LeSS высшее руководство, помимо группы продукта (управляющих портфелем заказов, руководителей высшего звена и прочих), должно ясно и определенно представлять себе Владельца Продукта как полностью подотчетного и несущего окончательную ответственность за выпуск продукта. Когда взаимоотношения с высшим руководством наложены, Владелец Продукта получает необходимую поддержку, чтобы уделять основное внимание доставке качественного продукта.

Владелец Продукта отвечает за то, чтобы у высшего руководства сложилось ясное представление о состоянии разработки продукта. Он должен понимать (возможно, неявно) приказ высшего руководства, оптимизировать желательные воздействия (на окупаемость инвестиций, долю на рынке и т.д.). Опираясь на поддержку Scrum-мастеров, Владелец Продукта привлекает их к совершенствованию организационной структуры таким образом, чтобы группа продукта получила конкурентное преимущество благодаря гибкости ведения коммерческой деятельности.

Если же у высшего руководства не складывается впечатление, что Владелец Продукта подотчетен или ответствен за успехи в выпуске продукта, в таком случае у Владельца Продукта возникают следующие затруднения.

- > Он не надеяется организационными полномочиями принимать и исполнять трудные решения в отношении продукта.
- > Он лишается значительного влияния на распределение ресурсов: финансов, количества команд, мест разработки и т.д.

И хотя подобные трудности могут существовать и в компаниях, выпускающих узкую номенклатуру продукции, они более характерны для компаний, выпускающих широкую номенклатуру продукции. Почему? Допустим, на предприятии имеется пять групп продуктов и только в одной из них внедрена инфраструктура LeSS. В таком случае у группы формирования портфеля продуктов (исполнительного руководства и прочих) *навсегда* останется прежнее взаимодействие с четырьмя традиционными группами продуктов. Они ожидают от таких групп определенных количественных показателей, завершенных этапов и отчетов. Но с группой продукта, где была внедрена инфраструктура LeSS, им придется взаимодействовать иначе: через Владельца Продукта, сообщающего результаты работы и адаптации к продукту. Кроме того, инициатива внедрить LeSS могла исходить от самой группы, а не от высшего руководства. По существу, высшему руководству было предложено перейти на два совершенно других организационных принципа, хотя они могли еще не осознавать этого! Поэтому Владельцу Продукта очень важно уловить подобную динамику, если она имеет место, чтобы активно противостоять неверным предположениям и недоразумениям, которые могут и должны при этом возникнуть.

По поводу взаимоотношений Владельца Продукта с высшим руководством можно дать следующие рекомендации.

- > **Самооценка.** На роль Владельца Продукта лучше всего подходят те люди, которые способны к самооценке. Они, во-первых, налаживают прочные, устоявшиеся и авторитетные отношения с высшим руководством, во-вторых, являются активными и неутанными поборниками изменений, в-третьих, неравнодушны к продукту и заказчикам, в-четвертых, имеют или предполагают иметь серьезные полномочия для принятия решений, и, в-пятых, стремятся к владению продуктом.
- > **Обучение других и пропаганда своей роли.** Вероятно, роль Владельца Продукта может оказаться новой для организации. Если она непонятна другим, то они никогда не поймут ее, если Владелец Продукта не будет сам пропагандировать свою роль и ее преимущества. Поэтому он должен обучить высшее

руководство. В идеальном случае Владелец Продукта может справиться с этим сам, а иначе ему придется прибегнуть к помощи Scrum-мастеров.

- > **Общение.** Владелец Продукта должен быть готовым брать на себя ответственность за продукт и отвечать на запросы высшего руководства о состоянии разработки продукта. С этой целью он должен постоянно общаться с высшим руководством, укрепляя в нем уверенность, что на него можно положиться.

Взаимоотношения Владельца Продукта со Scrum-мастерами

Остальные взаимоотношения непосредственно связаны с владением продуктом, хотя они совершенно другие и связаны со знаниями и поведением Владельца Продукта. Если Владелец Продукта постоянно совершенствуется вместе с командами, то у группы появляется больше возможностей для оптимизации выгод, извлекаемых из LeSS. И это должны быть более приятные взаимоотношения!

От Владельца Продукта Scrum-мастерам нужно узнать нужды, трудности и препятствия, в преодолении которых они могут оказать помощь. И грамотный Scrum-мастер может всегда подставить дружеское плечо. А самому Владельцу Продукта нужно учиться у Scrum-мастеров и получать от них ответную реакцию на обучение. Кроме того, Scrum-мастера делают запросы, например, на инструктирование команд.

По поводу взаимоотношений Владельца Продукта со Scrum-мастерами можно дать следующие рекомендации.

- > **Умеренность.** Владелец Продукта тесно сотрудничает только с одним или двумя Scrum-мастерами.
- > **Желание учиться.** Владелец Продукта усваивает основные понятия, посещая курсы вместе со Scrum-мастерами, читая рекомендованную ими литературу и работая с ними в паре (например, с целью научиться назначать приоритеты для Задела Продукта), наблюдая за их действиями в качестве устроителей мероприятий в LeSS и т.д.
- > **Осмысление.** Владелец Продукта запрашивает ответную реакцию на свое поведение у команд и прочих заинтересованных сторон, предлагая им осмыслить сложившуюся ситуацию.

Наставление: сотрудничество с клиентами превыше всего

Непрерывное назначение приоритетов означает постоянное обновление существующих приоритетов и новых элементов в Заделе Продукта с целью оптимизировать различные последствия обучения. В идеальном случае ценность должна доставляться в каждом Спринте, чтобы увеличить прозрачность и получить ответную реакцию. Ответная реакция будет оказывать влияние на приоритеты.

Непрерывное назначение приоритетов зачастую требует существенных изменений в умонастроении и поведении людей, переходящих от традиционных групп к LeSS, поскольку они раньше *играли в контракты*, для чего LeSS совершенно не подходит. Но иногда игра в контракты все еще *продолжается*.

Игра в контракты

В традиционной (особенно крупной) группе разработки одна сторона (обычно организационная или административная единица) согласует с другой стороной (как правило, отделом разработки) условия выпуска продукции по контракту⁴ с жестко заданным объемом работ к конкретной дате (и зачастую с вполне определенными затратами). Такой контракт затем передается в отдел разработки, которому предписывается взять на себя обязательство и ответственность за выпуск продукции.

В силу сложности и изменчивости, присущей разработке продукции, возникает ложное представление о том, что оценки объема работ, требований или обязательств являются вполне определенными. Таким образом, выполнение вынужденных обязательств превращается в поиск виноватых — игру между группой, занимающейся коммерческой деятельностью или управляющей продукцией, и группой, разрабатывающей продукцию. Такая игра приводит к медленному, но неумолимому снижению качества продукции и возможностей организации. Как же это происходит?

Если кратко,⁵ то для выполнения вынужденных обязательств (внутреннего контракта) подобная игра ведется за минимум победу в краткосрочной перспективе. При этом возникают скороспелые реакции и выбираются кратчайшие пути, которые имеют относительно долго проявляющиеся последствия и долги. Люди, вынуждающие брать обязательства, редко остаются до второй игры, и поэтому они вообще не испытывают последствия подобных долгов в отдаленном будущем. Следовательно, когда начинается следующая игра в контракты, дело обстоит еще хуже, а затем оно начинает ухудшаться по спирали. В конечном итоге продукт переводится в разряд самых неудачных и “устарелых образцов разработки где-нибудь в Йемене”.

Внедрение LeSS означает отказ от иллюзии жестко заданного объема работ и игры в контракты, но использование от одного Спринта к другому информации с целью направить разработку продукции на доставку наибольшей ценности. Но в то же время это означает не отказ от долгосрочного планирования, а стремление не путать планы с действительностью. Это также означает обучение и реагирование на перемены, но только не придерживаясь плана.

Наставление: выпуск хотя бы в каждом Спринте

На самом деле выпуск продукции на рынок (или для внутреннего потребления) в каждом Спринте требует больших перемен в умонастроении и поведении крупных групп. Очевидно, что возможны случаи, когда практически невозможно выпускать продукцию в каждом Спринте, например, при разработке сложного оборудования. Но для исключительно программных продуктов такое *вполне* возможно. Следует также иметь в виду случаи, когда действительно неуместно и невозможно выпускать

⁴ Такой контракт является *внутренним* соглашением, а не внешним коммерческим контрактом.

⁵ Столь примечательная динамика развития системы была подробно проанализирована нами в главах “Управление продукцией” и “Устаревший код” книги *Practices for Scaling Lean & Agile Development*, а также в главах “Системный подход” и “Организация” книги *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*.

продукцию в каждом Спринте, например, в ожидании большого всплеска внимания к продукции, приуроченного к какому-нибудь маркетинговому мероприятию.

Но по возможности Владелец Продукта должен принять решение о выпуске в каждом Спринте или еще чаще. Ниже перечислены веские основания для такого решения.

1. Ранняя доставка ценности.
2. Ответная реакция на эффективность новых функциональных средств, чтобы лучше адаптироваться к будущим Спринтам.
3. Повышенная чуткость к изменениям деловых потребностей.
4. Глубокие перемены к лучшему в группе разработки, поскольку трения, препятствующие частым выпускам, становятся до боли очевидными и требуют устранения.
5. Повышение внутренней мотивации у команд от ощущения достижений и прогресса.
6. Повышение доверия у заинтересованных сторон из-за ощутимых результатов.

Еще одно преимущество, пропорциональное организованному сопротивлению переменам, которое усиливается в крупных группах, заключается в следующем.

Выпуск продукции говорит сам за себя красноречивее любых слов.

Сильно ощущимое влияние выпуска в каждом Спринте разбивает большую часть аргументации против перемен в крупных группах и быстро становится убедительным предлогом для доставки еще большей ценности средствами LeSS. И как только появится возможность для выпуска в каждом Спринте, Владелец Продукта может исследовать пути для еще частого или даже непрерывного выпуска функциональных средств в потоке доставки ценности потребителям.

Наставление: непреклонность

Допустим, в LeSS сформировалась новая группа, но она не вся новая! Некоторые ее члены научились заурядным привычкам. Их еще терпели и даже не замечали в прежней системе из-за длинных циклов выпуска и разобщенности.

Таким образом, новая группа продукта в LeSS редко сможет получать готовые элементы в конце Спринта. Понятно, что командам требуется время, чтобы стать настоящими командами и постоянно чему-то учиться. И хотя это вполне ожидаемый, но все же нежелательный результат.

Владелец Продукта играет ключевую роль в формировании ожиданий для групп. Допустим, команды приходят к Владельцу Продукта и сообщают, что готова только половина элементов. Грамотный Владелец Продукта может им посочувствовать, но не принять сложившуюся ситуацию, проявив непреклонность. Вместо этого он должен

дать командам очень ясно понять, что если какие-нибудь элементы не готовы, то он ожидает от команд улучшения их работы с целью довести эти элементы до состояния готовности.

Но это совсем не означает, что Владелец Продукта должен требовать готовности всех первоначально запланированных элементов в каждом Спринте. Ведь такое требование лишь приведет к нарушениям, присущим игре в контракты, включая уменьшение прозрачности, увеличение многословия во избежание наказания, снижение качества и сокращение обучения. Чтобы справиться с изменчивостью в разработке, вполне *допустимо* исключить элемент из Спринта и даже не начинать его. Тем не менее в данном наставлении речь идет о небрежно сделанной, незавершенной работе над элементами, которые команда была в состоянии завершить, но оставила наполовину готовыми. Ведь она годами не несла никакой ответственности за создание комплексных функциональных средств, которые можно действительно считать готовыми.

Нам нередко приходилось видеть уступчивых Владельцев Продуктов, принимавших элементы, которые оказывались незавершенными из-за заурядных норм практики и разобщенного умонастроения. Такая уступчивость приводит к тому, что команды постоянно недорабатывают и не справляются со своими обязанностями. Владелец Продукта должен ясно дать понять командам, что им необходимо усиливать и расширять, а не ослаблять Критерий Готовности.

См. раздел “Наставление: организация, практикующая LeSS” главы 5.

Как только послание дойдет до адресата, для организации очень важно быстро и эффективно оказать конкретную помощь в улучшении положения. Поступить иначе означало бы нарушить мотивацию и доверие команд.

Наставление: ослабление контроля

Непреклонность совсем не означает микроуправление. Эффективное внедрение LeSS подразумевает наличие самоуправления, совмещение команд для функциональных средств, выполняющих всю работу и координирующих свои действия с другими командами. Высокая степень прозрачности проистекает от способности (или неспособности) доставлять завершенный продукт в коротком цикле. Поэтому от привычек пытаться контролировать разработку во время Спринта следует избавиться.

Многие команды не искушены в самоуправлении, но эту слабость нельзя одолеть, сказав им, что нужно делать! Им требуется пространство и время, а опытный Scrum-мастер поможет им расти дальше.

Контроль со стороны Владельца Продукта в группе, практикующей LeSS, осуществляется легко и просто. Например, Владелец Продукта может действовать следующим образом.

1. Не обследовать команды и не запрашивать у них отчет о состоянии во время Спринта. И тем более этого не должны делать другие руководители. Уделить основное внимание заказчикам и подготовке к будущим Спринтам. Безусловно, если команды обратятся за помощью, оказать ее.

2. Воспользоваться продуктом при Подведении Итогов Сprintа, чтобы выяснить, что происходило во время Сprintа. Адаптивно поставить цель на следующий Сprint.
3. Обследовать и изучить на Общей Ретроспективе процессы, условия и виды поведения, которые затруднены или требуют помощи. Адаптивно решить вместе со своей группой поставить эксперимент по улучшению.

Если такой контроль окажется слабым или неэффективным, в качестве обычных контрмер следует сделать Сprintы короче, увеличить прозрачность, усовершенствовав Критерий Готовности, а также производить более частые выпуски.

Наставление: незавершенная работа не по вине Владельца Продукта

Если говорить кратко, то Критерий Готовности отличается от прироста потенциально готового к поставке продукта наличием *Незавершенной Работы*. Это особенно характерно для первоначальной стадии внедрения LeSS в крупной традиционной группе. Особенности Незавершенной Работы и вытекающие из этого последствия подробно поясняются в главе 10.

См. подраздел “2. Выявить действия, которые можно выполнять в каждом Сprintе” главы 10.

Владелец Продукта должен обеспечить ясное распознавание любой Незавершенной Работы, чтобы знать, как справиться со сложившимся положением дел и вместе с командами постараться исключить это явление. Зачем? А затем, что Незавершенная Работа представляет собой риск и задержку выпуска. Самый лучший способ борьбы с Незавершенной Работой — вообще *не* иметь ее и производить выпуск в каждом Сprintе.

Наставление: совещания в LeSS

Когда мы представляем LeSS, то нам нередко задают такой вопрос: “Как одному Владельцу Продукта справиться со всеми этими совещаниями с участием всех команд?” К счастью, этот вопрос возникает из-за недопонимания. Один Владелец Продукта не посещает разные совещания с участием каждой команды. Например, имеется лишь одно общее совещание по Первому Планированию Сprintа, на котором собираются все команды.

См. часть “Сprint в LeSS” данной книги.

Какие же совещания в LeSS должен посещать Владелец Продукта и какова их средняя продолжительность в типичном двухнедельном Сprintе?

1. Первое Планирование Сprintа — 1 час.
2. Общее уточнение Задела продукта, если такое проводится, — 1 час⁶.
3. Подведение Итогов Сprintа — 2 часа.
4. Общая Ретроспектива Сprintа — 1,5 часа.

⁶ Это необязательное, хотя и типичное для LeSS совещание. Подробнее о нем см. в разделе “Наставление: виды Уточнения Задела Продукта” главы 11.

Таким образом, общее время, уделяемое совещаниям, оказывается намного меньше, чем кажется новому Владельцу Продукта. На практике оно не превышает 6 часов в двухнедельном Спринте.

См. раздел “Наставление: поговорить” главы 13.

Разумеется, когда Владельцу Продукта нужно поговорить с командами, ему не стоит дожидаться совещаний. Он должен просто пойти и поговорить с ними!

LeSS Huge

При внедрении LeSS в крупных масштабах с ролью Владельца Продукта связаны следующие принципы.

- > **Сосредоточение внимания на продукте в целом.** Заделы Продукта в Областях настолько наводнены подробностями, что могут потопить способность Владельца Продукта видеть общую картину. А поскольку Владельцы Продукта получают немалую свободу действий в своих Областях для внедрения новых направлений и подробностей, то сохранить общее представление о продукте нелегко.
- > **Ориентирование на клиентов.** Чтобы добиться согласованного взаимодействия с пользователями или полноценного комплексного решения, когда крупное требование охватывает несколько областей, требуется дополнительное координирование действий.

Команда Владельцев Продукта

См. раздел “LeSS Huge” главы 4.

Общий Владелец Продукта и Владельцы Продукта в Областях образуют *Команду Владельцев Продукта* в LeSS.

Никаких аналитиков, составителей спецификаций, проектировщиков взаимодействия с пользователем и пользовательского интерфейса или архитекторов не должно быть в Команде Владельцев Продукта, иначе текущие проблемы и структуры останутся под новой вывеской. Все подобные специалисты должны входить лишь в состав обычных команд для функциональных средств.

Правила LeSSHuge

В каждой Области Требований имеется свой Владелец Продукта в данной Области.

Один(общий) Владелец Продукта отвечает за направление развития продукта и распределение работы среди команд по Областям Требований. Он тесно сотрудничает с Владельцами Продукта в отдельных Областях.

Владельцы Продукта в отдельных Областях выполняют роль Владельцев Продукта для своих команд.

Наставление: Владелец Продукта в LeSS Huge

Роли Владельца Продукта в LeSS Huge и более мелкой инфраструктуре LeSS в какой-то степени перекрываются, например, при определении концепции продукта и представлении о конкурентах. Но эти роли имеют и существенные отличия.

Владелец Продукта в более мелкой инфраструктуре LeSS уделяет время выбору элементов для предстоящего Спринта, встречаясь с командами на совещании по Первому Планированию Спринта и т.д. А Владелец Продукта в LeSS Huge этого не делает, за исключением таких особых случаев, как полезное посещение рабочих мест. Основное внимание он уделяет более крупным и организационным задачам, которые перечислены ниже.

- > Выявление и назначение приоритетов для крупных тем и очень больших требований, охватывающих продукт (например, здравоохранение или поддержка технологии FDD-LTE), не вдаваясь в подробности⁷.
- > Выявление тенденций в сфере коммерции и технологий, которые должны привести к изменениям в Областях Требований.
- > Ввод или удаление и расширение или сокращение Областей Требований.
- > Распределение команд по Областям Требований.
- > Поиск, выращивание и поддержка Владельцев Продукта в Областях.
- > Обследование и адаптация крупных приоритетных тем в каждой Области Требований.
- > Выбор стратегии для мест разработки вместе с высшим руководством.

⁷ И хотя обособленный Владелец Продукта, не интересующийся или не вникающий в детали, не рекомендуется, при разработке крупных продуктов он просто не в состоянии уделять внимание мельчайшим подробностям.

См. выше раздел “Наставление: пять взаимоотношений”.

Помимо пяти взаимоотношений в более мелкой инфраструктуре LeSS, в LeSS Huge появляется шестой вид взаимоотношений между общим Владельцем Продукта и Владельцами Продукта в отдельных Областиах.

Наставление: Владельцы Продукта в Областиах

См. выше раздел “Наставление: кто должен быть Владельцем Продукта”.

Поиск Владельцев Продукта в отдельных Областиах производится по такому же критерию, как и в предыдущем наставлении для поиска Владельца Продукта в более мелкой инфраструктуре LeSS. Например, при разработке продукции в качестве специалиста в Области Требований вполне подойдет *ответственный за выпуск продукции*.

Владелец Продукта в Области обладает такими же полномочиями, как и общий Владелец Продукта. У последнего имеются независимые полномочия выбирать общее направление развития продукта, принимать решение о сроках выпуска и постепенное сворачивание и разворачивание Областей Требований. Но насколько это возможно, Владелец Продукта должен передавать свои полномочия и обязанности по определению концепции продукта и назначению приоритетов Владельцу Продукта в отдельной Области. При этом необходимо принимать во внимание приведенные ниже обстоятельства.

Мелкие области приводят к неверной роли Владельцев Продукта в Областиах. В обычной Области Требований имеется не меньше четырех команд⁸. А что происходит с ролью Владельца Продукта в мелкой Области, где действует одна или две команды? Она превращается в роль своего рода аналитика или составителя спецификации для прояснения элементов, а не руководителя, уделяющего основное внимание выбору стратегии и извлечению прибыли в главном секторе рынка. И тогда общий Владелец Продукта сотрудничает не с несколькими предпримчивыми и сосредоточенными на выборе стратегии Владельцами Продукта в Областиах, а с большим числом аналитиков деловой активности или руководителей проектов, сменивших название своей должности на “Владельцев Продукта в Областиях”.

Гибкость и безопасность труда. Области Требований должны медленно, хотя и быстрее, чем в традиционной группе, развиваться во времени, отражая изменение возможностей в крупных масштабах. Если люди в сужающихся областях, включая и Владельца Продукта в Области, опасаются за свою работу, то они могут оказать сопротивление, что приведет к снижению прозрачности и повлияет на организационную гибкость. Естественно, что это потребует пересмотра правил безопасности труда.

⁸ В особом случае первоначальная область может разрастись с большой вероятностью до многих команд. Внедрение может начаться с одной *ведущей команды*, которая сначала рассеивает туман неизвестности, а затем обучает другие команды, появляющиеся в новой области. Подробнее об этом см. в разделе “Наставление: динамика областей Требований” главы 4.

Временный заместитель Владельца Продукта. В данном случае можно дать такую же рекомендацию, как и в упоминавшемся ранее аналогичном наставлении для более мелкой инфраструктуры LeSS. В связи с тем, что поиск *настоящего* Владельца Продукта в Области (например, среди опытных специалистов, ответственных за выпуск продукции) может отнять время, развитие новой Области Требований можно безотлагательно начать с помощью *ненастоящего* Владельца Продукта, который способен на определенные действия, хотя и не совсем разбирается в данной области или не имеет достаточных полномочий. Такого временного заместителя следует заменить на настоящего Владельца Продукта в Области как можно скорее.

См. выше раздел
“Наставление: начало внедрения с времененным заместителем Владельца Продукта”.

Наставление: помощь, оказываемая Scrum-мастером Команде Владельцев Продукта

Команда Владельцев Продукта должна научиться работать вместе в существующей инфраструктуре LeSS Huge. При коммерческой разработке продукции они уже могут быть ответственными за выпуск продукции, работающими в одной и той же группе со сложившимися нормами взаимодействия, но LeSS является для них новым контекстом. А при внутренней разработке менее вероятно, что у них имеется опыт совместной работы. И поэтому они должны развивать в себе привычку к ретроспективе и самосовершенствованию. С этой целью они должны найти Scrum-мастера, готового добровольно оказать им помощь. Он должен посещать совещания своей Команды Владельцев Продукта, устраивать и проводить регулярные ретроспективы и давать свои отзывы Команде Владельцев Продукта относительно их работы.

Общий Задел Продукта

Элемент	Область
BX (обобщение элементов B1 и B2)	Освоение новых рынков
C	Управление торговыми операциями
D	Освоение новых рынков

Задел Продукта в Области освоения новых рынков

Элемент	Предок
BX-1 (прежний элемент B-1)	BX
BX-2 (прежний элемент B-2)	BX
D	

Объединение в Заделе Продукта

Задел ПрОдукта

Как править страной, где насчитывается 246 сортов сыра?
— Шарль де Голль

Scrum для одной команды

Единственный Задел Продукта с упорядоченными (по приоритетам) элементами представляет собой хранилище требований к продукту. Владелец Продукта отвечает за содержимое и упорядочение Задела Продукта, а также за его доступность для Команды и заинтересованных сторон. В нем постоянно происходят изменения, регулярно вводятся, удаляются или переупорядочиваются элементы с целью довести до максимума окупаемость инвестиций, исходя из обучения в каждом Спринте. Элементы, расположенные ближе к вершине Задела Продукта, являются более уточненными и готовыми к реализации. А элементы, расположенные ближе к дну Задела Продукта, являются более крупными и неясными. В ходе Уточнения Задела Продукта в каждом Спринте элементы разделяются, проясняются и оцениваются.

В упоминавшемся ранее *Руководстве по Scrum* приведено следующее главное правило масштабирования, которое говорит о том, что имеется лишь *один* общий Задел Продукта для работы многих команд над одним продуктом:

[Они] зачастую совместно работают над одним и тем же продуктом. А для описания предстоящей работы над продуктом служит один Задел Продукта.

При масштабировании Scrum отсутствует отдельный задел элементов для каждой команды. Почему? Дело в том, что это привело бы к снижению общей прозрачности, ослаблению внимания ко всему продукту, усложнению и подавлению привычки команд гибко переключать свое внимание.

Задел Продукта в LeSS

Наставления по составлению нового Задела Продукта при внедрении LeSS приведены в главе 3. А при масштабировании с Заделом Продукта связаны следующие принципы.

- > **Крупномасштабная инфраструктура SCRUM — это тоже SCRUM.** Таким образом, имеется лишь один Задел Продукта, даже если над одним продуктом работают многие команды.
- > **Сосредоточение внимания на всем продукте.** Единственный общий задел усиливает сосредоточенность на общем продукте и его видимость, а также увеличивает способность видеть и оптимизировать целое.
- > **Ориентирование на клиентов.** При традиционной крупномасштабной разработке работа и выполняющие ее команды разделяются на задачи по техническому характеру, компонентам и отдельным функциям. А в LeSS элементы из задела сосредоточены на комплексных целях клиентов.

Правила LeSS

Имеется единственный Задел Продукта (и один Владелец Продукта) для всего поставляемого продукта.

Наставление: минимизация ограничений вместо управления зависимостями

Определение продукта оказывает влияние на то, что является внутренним, а что — внешним (см. главу 7).

Грамотно составленный Задел Продукт прост и обеспечивает ясное общее представление о ходе работ при разработке продукции. Но Задели Продуктов нередко усложняются, поскольку они применяются в качестве инструментального средства для управления зависимостями. Хотя такого быть не должно.

При разработке продукции различаются внутренние и внешние зависимости. В частности, внутренние зависимости возникают между командами в группе продукта, тогда как внешние — за пределами группы продукта или команд, не предназначенных для разработки функциональных средств в самой группе продукта, например, в отделе незавершенных работ.

Исключение внутренних зависимостей

Управлять внутренними зависимостями в LeSS нет никакой необходимости. Более подробно этот вопрос рассматривается в главах 4 и 13.

В отсутствие внутренних зависимостей нет никакой необходимости в их управлении со стороны команд для функциональных средств, которые пользуются общим кодом.

Команды могут извлечь выгоду из совместной работы над компонентами продукта, но в то же время они не будут зависеть от результатов работы других команд.

Дело в том, что любая команда для функциональных средств может работать с общей кодовой базой для своих элементов. И команды сами координируют свои действия, применяя такие методики, как непрерывная интеграция, сообщества, многокомандные семинары, а также обмен и разделение общих работ. Для группы, которая раньше состояла из команд для компонентов с частным кодом и традиционным управлением зависимостями (например, через интеграционную команду или крупные события планирования), это несложная, но огромная перемена в умонастроении.

Сведение ограничений к минимуму вместо управления внешними зависимостями

Допустим, что завершение работы над элементом А зависит от выпуска компонента, *внешнего* для группы продукта. Как правило, такой компонент представляет собой канал передачи данных, службу, измененный интерфейс, аппаратный модуль или библиотеку. Это типичный пример крупномасштабной разработки. По традиции Владелец Продукта разрешает подобную зависимость следующим образом.

1. Вводит в задел внешнюю зависимость от элемента А.
2. Предусмотрительно планирует Спринт на будущее, когда элемент А будет готов, синхронизируя его с выпуском внешнего компонента.
3. Вводит запланированный Спринт в Задел Продукта.

При разработке крупных продуктов такой подход применяется не только к элементу А, но и ко многим другим элементам. В этом случае выполняется прогнозирующее планирование с точками синхронизации в ряде будущих Спринтов. Такое планирование беспорядочно и только отнимает время. Кроме того, прогнозы не всегда сбываются, и поэтому приходится выполнять повторное планирование, тратя еще больше времени.

Этого не следует делать ни коем случае! Вместо того чтобы рассматривать зависимости как неизменяемые ориентиры, при планировании зависимости нужно пересмотреть, представив их как ограничения, которые можно нарушить, придерживаясь следующих принципов.

1. Не позволяйте, чтобы зависимости вынуждали вас прибегать к прогнозирующему планированию. Не пытайтесь управлять зависимостями с будущими точками синхронизации, поскольку это приводит лишь к утомительному прогнозирующему планированию.
2. Рассматривайте зависимости как ограничения, вызывающие негибкость и задержку.
3. Противодействуйте, сводите к минимуму и устраняйте ограничения при всякой возможности.

Рассмотрим значение термина *зависимость*. Он предполагает *беспомощность* вследствие зависимости от других. Но рекомендация свести к минимуму или устранить ограничения означает действие, варианты выбора, властные полномочия, поскольку ограничения находятся под контролем. А это оказывает влияние на содержимое и приоритеты в Заделе Продукта.

Рекомендации по устранению ограничений

Как же устраниТЬ или хотя бы свести к минимуму ограничения? Допустим, что работа группы над элементом А якобы зависит от внесения внешней группой изменений в интерфейс продукта X. Прежде всего эту зависимость можно пересмотреть следующим образом: имеется ограничение на доведение элемента А до состояния готовности, и оно состоит в изменении интерфейса. Каким образом можно устраниТЬ или свести к минимуму это ограничение? Ниже даются некоторые рекомендации на этот счет.

- > **Сделайте свою часть работы.** Возможно, стоит заключить соглашение с группой продукта X, чтобы изменить код в этом продукте в сочетании с такими методами контроля качества, как совместное совещание по проектированию и/или ежедневные проверки кода. С другой стороны, можно написать сначала код без согласования с группой продукта X, а затем продемонстрировать его работоспособность и попросить разрешение включить его в их кодовую базу в сочетании с некоторыми методами проверки качества.
- > **Сделайте их часть работы вместе с ними.** Попробуйте объединить усилия своей и другой группы, чтобы помочь им вместе завершить их работу.
- > **Упростите или разделите элемент А таким образом, чтобы изменения, вносимые другой группой, оказались минимальными.** Разделите элемент А на более мелкие варианты, например, таким образом, чтобы только небольшие и простые изменения требовалось постепенно вносить в интерфейс продукта X. Такой прием можно также рассматривать как *сокращение объема партии* внешних изменений. А сочетание небольших партий с непрерывной интеграцией по всем продуктам позволяет ослабить ограничения и усилить ответную реакцию.

- > **Разделите элемент А на элемент с заглушкой, а также на полностью интегрированный элемент.** Реализуйте элемент с заглушкой — упрощенной имитацией продукта X. Как только интерфейс продукта X будет готов, удалите заглушку.
- > **Разделите элемент А на элемент с упрощенным интерфейсом, а также на элемент с окончательным интерфейсом.** Реализуйте элемент с альтернативным (т.е. ручным) вариантом интерфейса. Как только появится окончательный вариант интерфейса, удалите его альтернативный вариант.
- > **Поясните ограничение.** Поделитесь с группой X последствиями, затратами и выгодами, чтобы повлиять на их назначение приоритетов.
- > **Обойдите ограничение.** Возможно, стоит переопределить элемент A, чтобы поработать с другими существующими интерфейсами — по крайней мере, теперь.
- > **Добейтесь результата другим путем.** Возможно, существует другое решение для достижения поставленной цели.

Пример изменений в Заделе Продукта

Выбранный замысел обычно отражается в заделе. Рассмотрим два характерных тому примера.

Разделение элемента на более простые варианты. Допустим, в продукте для управления финансовыми рисками используются данные из продукта для управления торговыми операциями. Допустим также, что для элемента A из первого продукта требуется 30 новых элементов данных из второго продукта, но для их извлечения потребуется немало труда. В таком случае, можно попытаться разделить элемент A на следующие элементы, каждый из которых будет иметь значение для пользователей.

- > Элемент A1 с 10 элементами данных, наиболее важными для анализа рисков.
- > Элемент A2 с остальными элементами данных.

Разделение элемента А на элемент с заглушкой и полностью интегрированный элемент. Например, элемент A разделяется на два новых элемента в заделе следующим образом.

- > Элемент A с заглушкой.
- > Завершенный (или просто) элемент A.

Элемент A с заглушкой означает, что группа будет пользоваться (обычно простой) программной имитацией или заглушкой незавершенной части продукта X, как если бы она была полностью завершена. А завершенный элемент A подразумевает, что работа над продуктом X завершена, заглушка удалена и произведена полная интеграция обоих продуктов вместе с достоверными по-прежнему тестами, написанными при создании элемента с заглушкой.

Назначение приоритетов в ожидании завершения работы другой группой

В нежелательном случае, когда для завершения элемента А приходится ждать до тех пор, пока не закончит свою работу группа Х, по-прежнему требуется прогнозирующее планирование с точками синхронизации. Вместо этого рекомендуется сделать следующее.

1. Повысить приоритет элемента А с заглушкой, чтобы завершить его как можно раньше, а приоритет завершенного элемента А оставить низким. В этом случае отпадает необходимость прогнозировать, в каком именно Спринте должен быть завершен элемент, хотя для этого очень важно, чтобы завершенный элемент А оставался как можно более мелким, а следовательно, легко реализуемым в течение Спринта.
2. Добавить столбец “Сведения об ограничениях” в Задел Продукта. Записать в нем заслуживающие внимания подробности (например, *предполагаемую* дату выпуска) для тех элементов, на которые наложены ограничения в другой группе.
3. Пояснить группе Х последствия, затраты и выгоды, чтобы повлиять на их назначение приоритетов.
4. В дальнейшем, когда группа Х известит об окончании своей части работы, произвести адаптацию, повысив приоритет завершенного элемента А таким образом, чтобы он был полностью готов в следующем Спринте. Именно в этом и заключается гибкость разработки!

Наставление: надкусывание

См. далее раздел “Наставление: обращение с очень большими требованиями”.

В крупномасштабной области очень больших требований (даже во многих так называемых гибких вариантах внедрения) требуются месяцы, прежде чем требование появится в Заделе Продукта. Почему? Дело в том, что команды разработчиков *не* сумеют обращаться со столь крупным по масштабам требованием. Поэтому многие группы анализа, архитектуры или системного проектирования тратят месяцы на анализ очень больших требований, их разделение на части, составление спецификаций или изучение их выполнимости.

Традиционное умонастроение и поведение состоит в том, чтобы стремиться разделять и анализировать большую часть очень крупного элемента, прежде чем приступить к его реализации. Приступить к реализации требования нельзя, не уяснив это требование полностью, а также его влияние. Но что, если в дальнейшем обнаружится что-нибудь важное? Это произойдет неизбежно в отсутствие возможности путешествовать во времени!

А каковы затраты на столь раннюю излишнюю обработку? По иронии судьбы одна из статей затрат состоит именно в *задержанном обучении* из-за большого объема работы по предварительному анализу и прогнозирующему проектированию.

Ведь дело в том, что блок-схемы не нарушаются,
а документы не выполняются.

Помимо этого, группа существует на горе незавершенных работ, на которой нагромоздились скрытые риски и дефекты, потери от передачи работ и отложенные выгоды от доставки ценности. Время от времени с этой горы скатывается булыжник и прибивает кого-нибудь насмерть.

Но так жить нельзя! В организации не должно быть отдельных групп анализа, системного проектирования и составления спецификаций. И начинать так рано нужно не анализ, а разработку.

Но как это сделать? Если кратко, то одна Команда должна разделить нечто очень большое на несколько кусков, *надкусить* один из них и пережевывать вплоть до реализации, как показано на рис. 9.1. Это, по существу, означает, что мелкий элемент следует отделить от крупного, прояснить все его подробности на стадии уточнения и приступить к его реализации!

См. раздел "Уточнение Задела Продукта в команде по мелким кусочкам" главы 2.

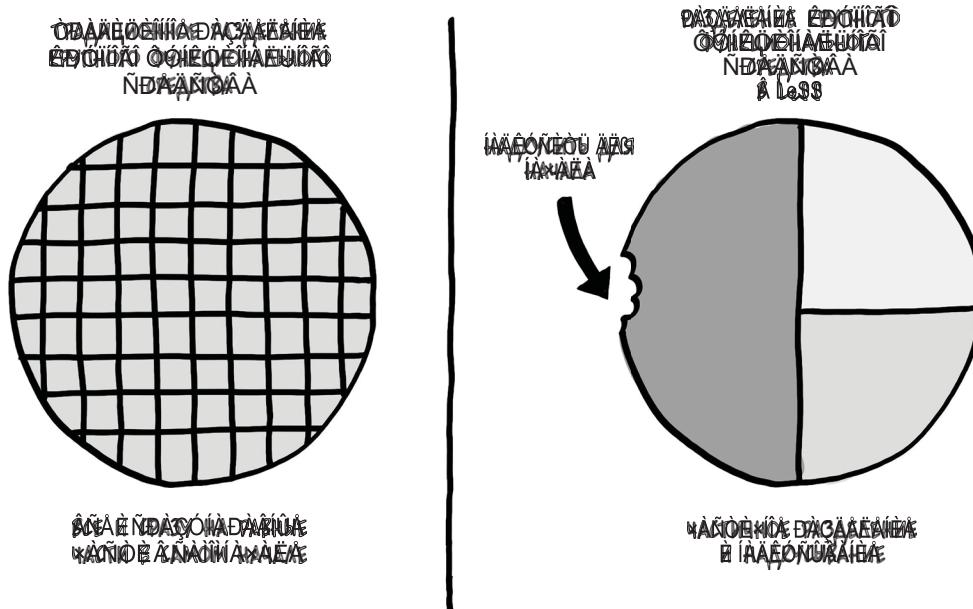


Рис. 9.1. Надкусывание в ведущей команде

Это делается по следующим причинам.

- > Начинайте пораньше! Чтобы завершить реализацию раньше, нужно и начинать ее раньше, надкусив небольшой кусочек, изучив и адаптировав. Путь к гибкости пролегает через ограничение незавершенных работ и повышение степени обучения с петлями более жесткой обратной связи.
- > Привлекайте команды к разделению и анализу крупного элемента, поскольку реализовать придется небольшой его кусочек. Благодаря этому повышается степень обучения и сокращается потребность в передаче работ.
- > Используйте начальную команду от начала и до конца, чтобы сократить потребность в передаче работ и утрату знаний.
- > Усильте ответную реакцию и обучение, чтобы выяснить, сколько кусочков и фрагментов можно переварить — возможно, не все кусочки, чтобы ощутить довольство. А после полной реализации небольшого кусочка выявите следующий самый вкусный кусочек, который стоит откусить.
- > Как можно раньше поймайте себя на мысли, что замысел продуман не до конца, и только после этого начинайте что-то делать.

Привычным умонастроением является “полное понимание” требования перед его реализацией. Но ирония состоит в том, что уклонение от ранней реализации препятствует пониманию. Это сильно упрочившаяся привычка, подкрепляемая наличием отдельных групп анализа, которые не занимаются реализацией. Такую привычку можно побороть мелким и частым надкусыванием. Отсутствие группы анализа совсем не означает отсутствие самого анализа, поскольку он выполняется теми же командами, которые реализуют элементы.

Наставление: обращение с родительскими элементами

Крупные элементы, которые должны быть разделены на более мелкие элементы, весьма распространены в крупномасштабной разработке. Если такой элемент разделяется, то что происходит с исходным, родительским элементом? Например, когда элемент расчетов по сделкам разделяется на элементы расчетов по покупкам и расчетов по продажам, то что делать с расчетами по сделкам? Для этого имеются следующие альтернативы: удалить или сохранить родительский элемент. Рассмотрим их компромиссы и применимость.

Удаление родительского элемента

Удаление родительского элемента из Задела Продукта аналогично делению клеток (цитокинезису). При этом родительский элемент заменяется новыми элементами. В чем преимущества такого решения? Первое преимущество заключается в простоте: структура задела остается простой и не требует дополнительных усилий для связывания родительских и рожденных элементов. А второе, менее заметное преимущество заключается вследующем:



Приоритеты для новых элементов естественно или очевидно назначаются независимо друг от друга, а также от их родительского элемента.

Что же все это означает? При традиционной крупномасштабной разработке *все* требования, подчиненные более крупному требованию, имеют одинаковый с их родителем приоритет, и они перемещаются одной партией по всему процессу разработки. Это приводит к затратам времени и средств на менее ценные элементы наряду с более ценными, задержке выпуска мелких, но очень ценных элементов, замедленной ответной реакции и запоздалому снижению рисков.

Но для гибкой разработки очень важна *независимость элементов*. Каждый новый элемент должен быть независимым от других элементов и своего родителя. И хотя простое решение удалить родительский элемент не гарантирует переход к такому умонастроению и поведению, тем не менее, оно позволяет создать задел, в котором вполне естественным и очевидным оказывается независимое упорядочение элементов. Допустим, что перед разделением элемента для расчетов по сделкам Задел Продукта был упорядочен следующим образом:

1. X
2. Расчеты по сделкам
3. Y

После разделения и удаления родительского элемента порядок расположения элементов в заделе, определенный Владельцем Продукта, стал следующим.

1. Расчеты по покупкам
2. X
3. Y
4. Расчеты по продажам

Очевидно, что элементы для расчетов по покупкам и по продажам упорядочены независимо друг от друга. Благодаря этому поддерживается ранний выпуск только самых ценных элементов, а также гибкость и удобство. Это, конечно, замечательно, но требует изменить умонастроение коренным образом.

А есть ли недостатки у такого подхода? Есть, и это потеря контекста и ассоциаций, которые могли бы оказаться полезными при уточнении элементов или же в том случае, если бы потребовалось определить тему для выпуска из связанных вместе элементов.

В небольшом заделе продукта или предметной области, где люди близко знакомы с требованиями, удаление родительского элемента вряд ли вызовет какие-нибудь осложнения. А поскольку это простое решение, то оно оказывается более предпочтительным для простых случаев.

Сохранение родительского элемента

Когда же следует сохранять сведения о родительском элементе? Это следует делать тогда, когда имеется крупный Задел Продукта, содержащий обширный ряд элементов или настолько сложный, что трудно запомнить (или выявить) взаимосвязи между новыми порожденными элементами и их родителями. А насколько полезными могут быть сведения о родительском элементе? Они могут оказаться полезными в следующих случаях:

- > для крупномасштабного контекста, помогающего лучшему пониманию или принятию решений;
- > в качестве источника вдохновения для порождения новых элементов;
- > для выявления тем выпусков;
- > в LeSS Huge, чтобы помочь вести Задел Продукта в Области как отдельные артефакты.

См. далее раздел
“Наставление: заделы
в областях”.

А где следует хранить сведения о родительском элементе? С этой целью следует добавить столбец “Родитель” в Задел Продукта и ввести в него необходимые сведения, как показано в приведенном ниже примере.

Номер по порядку	Элемент	Заслуживающий внимания прямой или непрямой родитель
1	Расчеты по продажам	Расчеты по сделкам
2	X	
3	Y	
4	Расчеты по покупкам	Расчеты по сделкам

Сохраняя родительский элемент, необходимо иметь в виду следующее.

- > Нельзя допускать углубления иерархии на многие уровни родительских элементов. (Подробно об этом — ниже, в разделе “Наставление: максимум три уровня”.)
- > Сведения о родительском элементе необязательны, поэтому ими следует пользоваться лишь тогда, когда они *заслуживают внимания*.
- > Регистрируемый родительский элемент совсем не обязательно должен быть *прямым* предком вновь порожденного элемента. Он может быть и *далеким* предком, если первоначальный очень крупный элемент порождает многочисленное потомство. В таком случае сохранение сведений о родительском элементе более предпочтительно, поскольку оно позволяет легче прослеживать связи между далеким предками.
- > В таблице из приведенного выше примера необходимо обратить внимание на следующее очень важное, хотя и незаметное на первый взгляд обстоятельство: элементы упорядочены слева направо — от порожденного к родительскому, а не от родительского к порожденному. Это обстоятельство отражает перемену в умонастроении и назначении приоритетов в связи с независимостью порожденных элементов.

Наставление: обращение со специальными элементами

Помимо функциональных средств клиентов, в Заделе Продукта могут присутствовать другие элементы. К их числу относятся элементы для дефектов, улучшений, нововведений и специальных исследований.

Элементы для дефектов

Обычно в Scrum рекомендуется регистрировать дефекты, сообщаемые заказчиком, в виде элементов Задела Продукта. Это превосходная рекомендация при наличии не более десяти дефектов, и поэтому ею следует пользоваться, когда это уместно.

Большой список программных ошибок. Но об этой рекомендации следует забыть, когда список программных ошибок насчитывает 714 обнаруженных дефектов, поскольку все эти дефекты уже зарегистрированы инструментальным средством для отслеживания дефектов. Если переместить их в Задел Продукта, что само по себе не-легко и чревато ошибками, то они просто засорят его. Это весьма характерная ситуация для крупных продуктов, в которых годами накапливались дефекты, и только затем была внедрена инфраструктура LeSS. Следовательно, когда Задел Продукта создается в первый раз при наличии большого числа дефектов, пользоваться инструментальным средством для отслеживания дефектов следует до тех пор, пока количество дефектов не уменьшится до такой степени, чтобы их было удобно занести в Задел Продукта. И в этом случае на вершине Задела Продукта следует ввести элемент “Количество дефектов = N”, чтобы легче было отслеживать дефекты и быстро довести их до нуля. Все особые дефекты необходимо как можно скорее зарегистрировать в Заделе Продукта, чтобы *не* упустить их из виду и *не* забыть о них. Их должны видеть все заинтересованные стороны и соответственно реагировать на них.

Полное устранение дефектов. Как довести 714 дефектов до нуля на первоначальном этапе? В общем, следует применить принцип “остановиться и устраниТЬ” бережливого подхода, уделив основное внимание дефектам. Почистите список программных ошибок, отбросив все лишнее. Выделите одну или несколько команд для устранения дефектов, возможно, сменяя их по кругу. Можно даже организовать в группе рабочее совещание по разрешению затруднений, попросив разработчиков подготовить и провести эксперименты по устранению программных ошибок.

Неотложные новые дефекты. Если дефекты известны при Планировании Спринта, их устранение следует запланировать. А как насчет неотложных дефектов, возникающих в течение Спринта и требующих быстрого реагирования? С этой целью можно, в частности, назначить обычную команду для функциональных средств в качестве команды *быстрого реагирования*, передавая по очереди ее обязанности другой команде в каждом Спринте. Такие команды берут на себя все прерывания и отклонения от привычного ритма работы, а остальные команды не отвлекаются от своих прямых обязанностей. Еще одно преимущество, в отличие от прерывания работы и выбора той команды, которая способна быстро устранить возникший дефект, заключается в том, что команды могут изучать малознакомые области.

Элементы для совершенствования команд

Многие элементы должны быть выполнены командами для их совершенствования. Это могут быть организационные улучшения, но зачастую они носят технический характер или имеют отношение к рабочей среде. Элементы для совершенствования команд должны, как правило, появляться в результате ретроспектив (на уровне команд или системы) или совещаний в сообществах (архитектурных, тестовых и прочих).

А где должны регистрироваться элементы для совершенствования команд?

Крупные улучшения в Заделе продукта. Если в проект вкладываются большие средства, элементы для совершенствования команд лучше разместить в Заделе Продукта. Это дает ряд следующих преимуществ.

1. Работа команд видна и контролируется в одном месте.
2. Владелец Продукта может решить, в какие крупные улучшения следует вкладывать средства в первую очередь.
3. Непрерывное совершенствование происходит в обычном порядке выполнения работ.

Регистрируя элементы для совершенствования команд, рекомендуется сделать следующее.

Выразить крупные элементы для совершенствования команд с точки зрения выгоды для коммерческой деятельности и Владельца Продукта.

Если командам требуется переделать главный компонент, то какие выгоды это принесет?

Мелким улучшениям *не* место в Заделе Продукта. Почему? А потому что в крупной группе будет столько улучшений для всех команд, что, введя их в Задел Продукта, можно просто засорить его, ослабив внимание на главной цели: функциональных средствах заказчика. Множество мелких элементов для совершенствования команд только затруднит ведение задела и назначение приоритетов, что приведет к микрокуправлению и микроулучшениям, уничтожающим дух самоуправления, атмосферу доверия и стремление к непрерывному совершенствованию.

См. раздел “Наставление: теории управления” главы 5.

Что же тогда делать? Можно, например, согласовать следующее правило.

1. Вводить в Задел Продукта только те элементы для совершенствования команд, которые превышают определенную величину “Х”.
2. Предложить каждой команде уделять в каждом Спринте до 20% своего рабочего времени мелким улучшениям, которые отсутствуют в Заделе Продукта.

Это упрощает дело и способствует самоуправлению и доверительным отношениям.

Элементы для нововведений и необычных исследований

Рассмотрим следующие случаи, типичные для разработки крупной продукции.

- > Альтернативные наборы микросхем или сторонние программные компоненты.
- > Нововведения.
- > Анализ конкурентов.
- > Анализ перспективных технологий.

Это типичные крупные задачи со множеством вариантов. И поэтому Владельцу Продукта и Командам потребуется дополнительная информация, чтобы принять решение или назначить приоритеты. Как же справиться с этим в LeSS? С этой целью рекомендуется сделать следующее.

- > Вести исследование или нововведение в Задел Продукта.
- > Наложить определенные ограничения на подобные ничем не ограниченные мероприятия, чтобы они не отнимали все время, отведенное на Спринт (например, максимум 50 человеко-часов).
- > Воспользоваться услугами обычной команды для функциональных средств, а не какой-нибудь исследовательской группы.
- > Пользоваться *надкусыванием* везде, где это только возможно, а не долгим исследованием.
- > Уделить внимание изучению информации или рекомендаций для Владельца Продукта и Команд, чтобы оказать им помощь в принятии решений.

- > Поделиться советом по поводу следующих действий на Ретроспективе Спринта.
- > Если речь идет о нововведении, постараться быстро создать экспериментальные функциональные средства для продукта и получить ответную реакцию на них от *реальных* пользователей.

Такого рода необычные исследования или изыскания не имеют никакого отношения к обычным аналитическим, проектным или архитектурным работам. А с другой стороны, следует избегать *ненастоящих* исследований, следуя такой рекомендации.

Ни в коем случае не создавайте исследовательские элементы для обычных и повторяющихся мероприятий по анализу или проектированию, включая анализ деловой активности, архитектуры или взаимодействия с пользователем, проектирование пользовательского интерфейса или других компонентов системы.

Внимание! Ни в коем случае не создавайте специальную группу, дав ей право включиться в разрешение крупного затруднения. Уж лучше взять большую пачку денег и поджечь ее — по крайней мере, можно согреться!

Наставление: инструментальные средства для ведения крупных заделов продуктов

“Мы не действуем гибко! Аналитики составляют прецеденты использования и сценарии в окружающем мире, регистрируют их в SharePoint и сообщают командам по электронной почте, где им искать информацию”.

“А теперь мы действуем гибко! Владельцы Продуктов пишут эпические и обычные истории, регистрируют их в сборном заделе и сообщают командам в уведомлениях, где искать информацию”.

Разумеется, было бы заблуждением считать, что употребление новых терминов и инструментальных средств под новыми вывесками означает, что произошли какие-то значительные перемены.

Гибкость заключается не в инструментальных средствах, а в организационном поведении.

Нам приходилось видеть немало крупных и многокомандных групп, которые успешно вели свои Заделы Продуктов в виде простых электронных таблиц, привязанных (например, через Google) к вики-страницам с подробной информацией. В действительности группам лучше поступать именно таким образом.

Каким же инструментальным средством следует пользоваться для ведения Задела Продукта при крупномасштабной разработке? Оно должно быть не сложнее электронных таблиц и вики-страниц.

Почему? А потому что применению так называемых “гибких” инструментальных средств присущи следующие недостатки.

- > Сосредоточение основного внимания на инструментальных средствах, а не на систематических затруднениях. Это отвлекает внимание и не позволяет сосредоточить его на том, что важнее всего: изменении поведения и системы. Такие инструментальные средства не разрешают настоящие затруднения.
- > Подобные инструментальные средства содержат функциональные средства для составления отчетов и способствуют их употреблению, только укрепляя традиционные виды поведения, присущие административной отчетности и контролю.
- > Они служат лишь фасадом улучшений или внедрения методик гибкой разработки, за которым не происходит никаких существенных перемен. Так называемые “гибкие” инструментальные средства не имеют никакого отношения к настоящей гибкости.
- > Они нередко навязывают командам неудобную терминологию и технологические процессы, отнимая у них владение процессом и ограничивая совершенствование.
- > Задел нередко оказывается скрытым для большинства заинтересованных сторон, поскольку требует затратной учетной записи.
- > Подобные инструментальные средства усложняют, а не упрощают дело.

Безусловно, подобные недостатки могут обнаружиться и в электронных таблицах, если сделать их чрезмерно сложными. И этого следует всячески избегать.

Отслеживание состояния работ

Отслеживание в Спринте

Ниже приведены цитаты с начальной страницы, рекламирующей известные инструментальные средства “гибкого” управления.

“Мгновенно отслеживает ход выполнения работ в команде”, “Составляет отчеты о состоянии работ”, “Подготавливает отчетность по вашим проектам”, “Более чем наполовину оснащено гибкими количественными показателями и отчетами” и прочая тошнотворная реклама.

В инструментальных средствах так называемого “гибкого” управления основное внимание уделяется функциям отслеживания и отчетности, которые показывают

руководителям индивидуальные или командные задачи и Заделы Спринтов, а также состояние работ, что полностью противоречит таким принципам гибкой разработки, как доверие к людям и самоуправление в командах. В этой связи известный исследователь команд Ричард Хакман поясняет: “В самоуправляемых командах ответственность за отслеживание состояния работ возлагается на команду”.

Таким образом, у руководства не должно быть обязанности или причин для отслеживания хода выполнения работ в командах во время Спринта. Упомянутые выше инструментальные средства ориентированы на отчетность, а не на достижение поставленной цели, улучшение, совершенствование потока доставки ценности или команд, владеющих своими процессами и улучшающих их.

Заделы Продукта и Спринта в Scrum разделены и имеют разное назначение. Если Задел Продукта предназначен для ведения элементов, ориентированных на клиентов, то Задел Спринта — для самоуправления в Команде и решения задач, стоящих перед ней в течение Спринта. Он не предназначен для Владельца Продукта и внешнего отслеживания. Об этом ясно говорится в *Руководстве по Scrum*: “[Задел Спринта] при надлежит только Команде Разработчиков”. Таким образом, Команде нужно выбрать подходящее инструментальное средство для ведения Задела Спринта, чтобы иметь возможность изменять свой выбор. И разные Команды могут пользоваться разными инструментальными средствами. В связи с этим необходимо иметь в виду следующее.

Для ведения Задела Продукта и Задела Спринта нельзя
пользоваться одним и тем же инструментальным средством.

Несмотря на то что для ведения Задела Спринта может подойти любое инструментальное средство, мы постоянно наблюдали, что команды, просто использовавшие карточки на стене, в намного большей степени оказывались *настоящими*, дружно работавшими и активно совершенствовавшимися.

Отслеживание состояния работ по Спринтам

См. раздел “Наставление: непреклонность” главы 8.

Полезно иметь общее представление о состоянии работ над ориентированными на клиентов элементами по Спринтам. А требуется ли группе для этого какое-нибудь специальное, гибкое инструментальное средство? Нет, нетребуется.

Прозрачность и простота отслеживания заметно увеличиваются, когда основное внимание сосредоточивается на *готовых* элементах. В конце каждого Спринта элементы отмечаются как готовые или неготовые, но не как “почти” или “на 90%” готовые. В Заделе Продукта необходимо отслеживать только состояние работ над готовыми элементами. И для этого достаточно самых простых средств.

Если требуется *график выполнения работ*, сначала нужно выяснить и проанализировать, зачем он требуется. В частности, почему заинтересованные стороны отсутствуют на Подведении Итогов Спринта? Выполняет ли так называемый Владелец Продукта лишь функции руководителя проекта или программы под новой вывеской?

Если график выполнения работ действительно требуется, для его построения можно воспользоваться соответствующей функцией инструментального средства вроде электронных таблиц, где ведутся записи в Заделе Продукта.

Наставление: больше конечных и меньше непосредственных результатов

Однажды нам пришлось работать с крупной группой продукта, где старший начальник сделал такое заявление: “За последние 12 месяцев мы затратили 1,3 млн человеко-часов на продукт. Все отлично потрудились!” Так называемые “достижения” до сих пор измеряются объемом затраченного труда и количеством поставок (т.е. готовых элементов). Даже такая распространенная мера, как скорость, служит для определения затрат труда на выпуск функциональных средств, и в этом все дело. Ведь все эти трудозатраты и поставки могут иметь малое отношение или вообще не иметь никакого отношения к конечным результатам. Если заказчику требуется новое инструментальное средство для управления производственными процессами с конкретным набором функций, то каковы его цели? Способны ли эти функции сократить на 25% среднюю продолжительность производственного цикла?

Как правило (особенно в крупных группах), основное внимание уделяется *непосредственным*, а не *конечным* результатам по следующим причинам.

- > Возникает сильное искушение управлять непосредственными результатами, поскольку их легче измерить.
- > В традиционном годовом бюджетном процессе требуется перечень оцененных по затратам функциональных средств (т.е. непосредственные, а не конечные результаты).
- > Крупный Задел Продукта превращается в место свалки сотен запросов функциональных средств без ясной связи с конечными результатами.

Один из принципов LeSS гласит: добиваться большего средствами LeSS. В данном контексте это означает следующее.

Больше конечных и меньше непосредственных результатов.

Какими же методами можно сместить акцент на конечные результаты? Эти методы рассматриваются в последующих подразделах.

Составление элементов в виде конечных результатов или целей, а не решений

Крупная служба доставки пакетов в Норвегии регулярно получала жалобы на недобство пользования ее веб-сайтом. И тогда в этой службе решили составить новый элемент следующим образом.

Показывать все подробности варианта доставки на одной веб-странице.

Такой элемент ориентирован на решение, поскольку в нем предполагается решение возникшего затруднения. Возможно, это было бы и отличное решение, но его назначение неясно. Поэтому предпочтение следует отдавать элементам, ориентированным на конечные результаты или цели, как, например:

служба-отправитель сможет найти все варианты поставки в верхнем квартile менее, чем за 1 секунду.

В таком элементе, ориентированном на конечный результат, приветствуется больше вариантов и замыслов, а также повышается мотивация благодаря творческому вызову команде.

Оценка воздействия

*Оценка воздействия*¹ — это коллективная, быстрая и наглядная методика для группы, позволяющая, во-первых, определить конечный результат (например, сократить количество ошибок при совершаемых сделках), во-вторых, определить меру успеха и, в-третьих, выработать альтернативные идеи, чтобы воздействовать на конечный результат. Эта методика наглядно представлена на рис. 9.2.

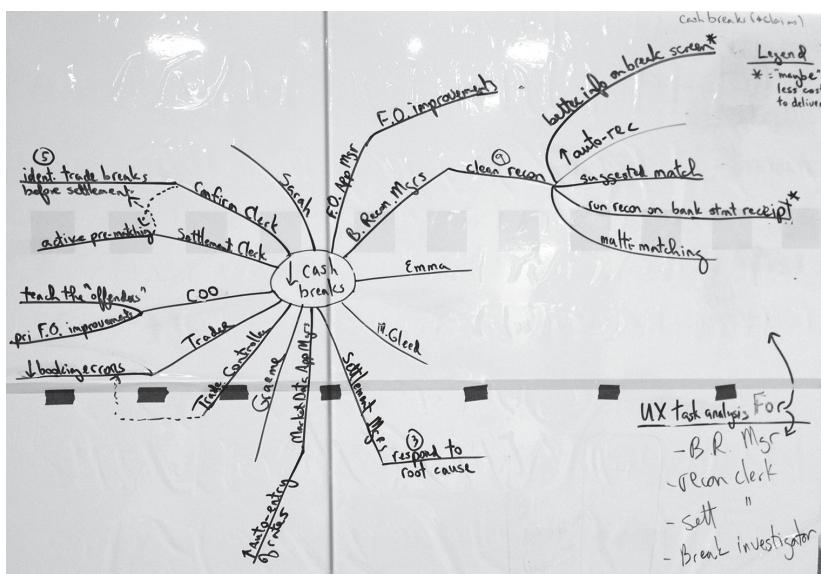


Рис. 9.2. Методика оценки воздействия способствует сосредоточению основного внимания на конечных, но не непосредственных результатах, а также альтернативным действиям, которые позволили бы достичь конечных результатов

¹ Подробнее об этом см. в книге *Impact Mapping* и на веб-странице по адресу <https://www.impactmapping.org/>.

Чем может помочь методика оценки воздействия? Во-первых, она содействует сотрудничеству с акцентом на конечные результаты, во-вторых, помогает сосредоточить основное внимание на нескольких альтернативных вариантах воздействия, и, в-третьих, связывает воздействия с конечными результатами.

LeSS Huge

При крупномасштабной разработке соблюдаются приведенные ниже принципы, связанные с Заделом Продукта.

Сосредоточение внимания на продукте в целом и прозрачность. Эти принципы соблюдаются, когда Задел Продукта разделяется на Заделы в отдельных Областях. Как же сохранить представление и основное внимание на общих целях и приоритетах, не вдаваясь в подробности?

Правила LeSSHuge

Существует единственный Задел Продукта. Каждый его элемент принадлежит только одной Области Требований.

На каждую Область Требований приходится единственный Задел Продукта в данной Области. Этот задел принципиально является более точным представлением единственного Задел Продукта.

Наставление: заделы в областях

Прежде всего следует заметить, что Область Требований представляет собой крупную совокупность элементов, логически связанных вместе с точки зрения *клиента*. В этой связи необходимо иметь в виду следующее.

Каждая Область Требований является совокупностью элементов с точки зрения клиента, а не с технической точки зрения.

Области Требований служат в качестве методики масштабирования крупных групп. В каждой Области Требований предполагается от четырех и больше команд.

Принципиально область требований является атрибутом, вводимым в единственный Задел Продукта. И каждый его элемент относится к одной и только к одной категории, как показано ниже.

Элемент	Область Требований
B	Освоение новых рынков
C	Управление торговыми операциями
D	Обслуживание активов
F	Освоение новых рынков
...	

А Задел в Области принципиально является *представлением* единственного Задела Продукта в одной Области Требований (например, в области освоения новых рынков), как показано ниже.

Элемент	Область Требований
B	Освоение новых рынков
F	Освоение новых рынков

Подробнее о роли Владельца Продукта и Владельцев Продукта в отдельных Областях см. в главе 8.

Для Владельца Продукта в Области и команд, назначенных для этой области, их Задел в Области выглядит и функционирует как обычный Задел Продукта. Но наивысший приоритет для Задела в Области не должен быть наивысшим приоритетом в Заделе Продукта. Если произойдет нечто подобное, Владелец Продукта должен определить, насколько сильно отличаются приоритеты, чтобы обосновать перемещение команд в другие области.

Задел в Области служит не для одной или двух команд, а для Области Требований, где обычно имеется не меньше четырех команд.

Заделы в Областях можно реализовать двумя способами: как представления и как отдельные артефакты.

Заделы в Областях как фильтрованные представления

Заделы в Областях проще всего реализовать, наложив фильтр на единственный Задел Продукта, чтобы создать его представление. Сделать это в электронных таблицах проще простого. Такой способ применяется тогда, когда имеется немного (например, три) областей Требований, а глубина разделения элементов невелика². Безусловно, это слишком расплывчатые рекомендации. И хотя переломный момент для перехода к отдельным артефактам зависит от конкретной ситуации, его можно распознать, когда

² Это также называется *менее крупной* группой продукта в LeSS Huge.

он появится. А начинать лучше всего с более простого способа: фильтрованных представлений.

Назначение приоритетов по областям. Владельцы Продукта в Областях назначают приоритеты в Заделах своих Областей более или менее независимо. Поэтому в каждой области имеется свой первый элемент, затем второй элемент и т.д., как показано в приведенном ниже примере.

Элемент	Область Требований
B	Освоение новых рынков
F	Освоение новых рынков
C	Управление торговыми операциями
M	Управление торговыми операциями
...	

The diagram shows three arrows originating from the first element in each row of the table. The first arrow points to the header 'Область Требований'. The second arrow points to the first element 'B' in the first row. The third arrow points to the first element 'F' in the second row.

Заделы в Областях как отдельные артефакты

Когда имеется немало Областей Требований и множество разделенных элементов, простой способ фильтрованных представлений начинает причинять неудобства³. При этом единственный Задел Продукта становится непомерно большим, детализированным и заполняемым при разделении множеством очень мелких, но подробных элементов из всех областей.

В таком случае альтернативой служат отдельные артефакты (например, отдельные электронные таблицы) как для Заделов Продукта в Областях, так и для общего Задела Продукта. Как поясняется далее, такому способу присущи иные недостатки, чем у более простого способа фильтрованных представлений.

Разделение по конкретным областям

Допустим, заделы начинаются так, как показано в таблицах на рис. 9.3. А теперь допустим, что элемент B (Освоение новых рынков) разделяется на элементы B-1 и B-2, как показано в таблицах на рис. 9.4. При разделении на отдельные артефакты описываемым здесь способом общий Задел Продукта остается без изменений, а Задел в Области освоения новых рынков изменяется.

³ Это также называется *очень крупной группой* продукта в LeSS Huge.

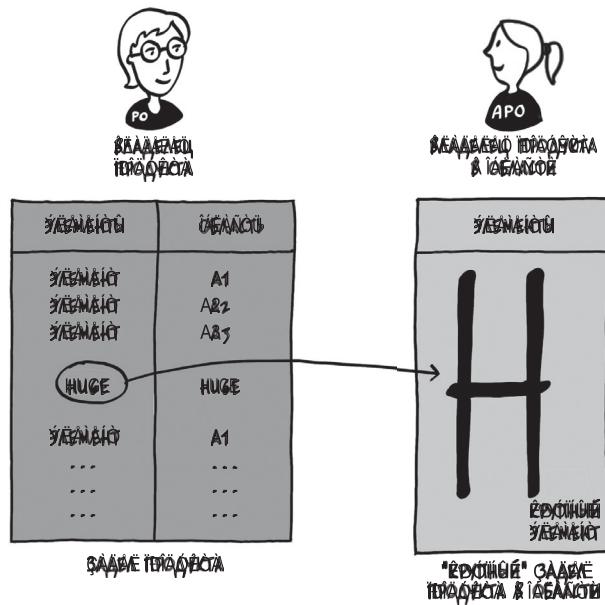
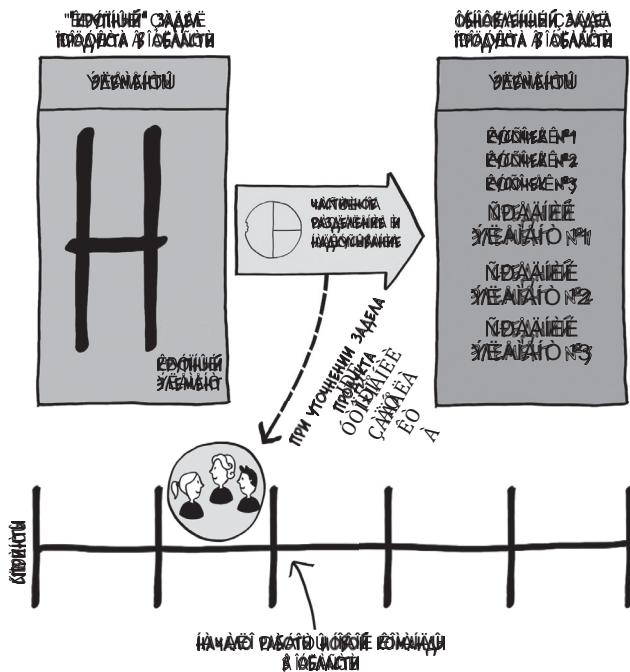


Рис. 9.3. Задел в Области до разделения

Рис. 9.4. Задел в Области после разделения элемента B.
При этом Задел Продукта остается без изменений

Назначение приоритетов по конкретным областям

Наличие отдельных артефактов позволяет Владельцу Продукта работать на более высоком уровне детализации, чем Владельцам Продукта в отдельных Областях, но в то же время оно делает менее прозрачным положение дел для Владельца Продукта. Дело в том, что для Задела Продукта в Области приоритетным является решение, принимаемое Владельцем Задела Продукта в данной Области, а приоритеты разделенных элементов не должны следовать приоритетам в общем Заделе Продукта. В приведенном ниже примере одни части элемента B имеют более высокий приоритет, чем у элемента D, а другие — более низкий.

Общий Задел Продукта

Элемент	Область
B	Освоение новых рынков
C	Управление торговыми операциями
D	Освоение новых рынков

Задел Продукта в Области освоения новых рынков

Элемент	Предок
B-1	B
D	
B-2	B

Как правило, различие в приоритетах невелико и не вызывает особых затруднений. Но иногда затруднения все же возникают, как показано в приведенном ниже примере.

Общий Задел Продукта

Элемент	Область
B	Освоение новых рынков
C	Управление торговыми операциями
D	Освоение новых рынков

Задел Продукта в Области освоения новых рынков

Элемент	Предок
B-1	B
B-2	B
D	

В данном случае одни части элемента B (B-1 и B-2) имеют высокий приоритет, тогда как другие его части (B-3 и B-4) его не имеют. Этот случай отражается в содержимом Задела в Области, но он не виден для Владельца Продукта. А это приводит к недоразумениям и последующим осложнениям. Например, Владелец Продукта может

прийти к выводу, что элемент В не будет готов до тех пор, пока не будут готовы все его части, хотя это и не отражает приоритеты Владельца Продукта в Области.

Чтобы правильно отразить *большое* различие в приоритетах, когда мелкими элементами можно пренебречь, поскольку они не создают особых трудностей, Владельцу Продукта в Области нужно снова *объединить* элементы в общем Заделе Продукта. Такое объединение означает создание нового обобщенного и крупного элемента из ряда мелких элементов, как показано в таблицах из примера на рис. 9.5. Благодаря этому различие в приоритетах правильно отражается в общем Заделе Продукта. А поскольку элементы В1 и В2 объединены (или обобщены) в один крупный элемент ВХ, то общему Владельцу Продукта не нужно вникать в подробности.

Общий Задел Продукта

Элемент	Область
ВХ (обобщение элементов В1 и В2)	Освоение новых рынков
C	Управление торговыми операциями
D	Освоение новых рынков
E	Освоение новых рынков
F	Освоение новых рынков
ВY (обобщение элементов В3 и В4)	Освоение новых рынков

Задел Продукта в Области освоения новых рынков

Элемент	Предок
ВХ-1 (прежний элемент В-1)	ВХ
ВХ-2 (прежний элемент В-2)	ВХ
D	
E	
F	
ВY-1 (прежний элемент В-3)	ВY
ВY-2 (прежний элемент В-4)	ВY



Рис. 9.5. Объединение (или обобщение) нескольких элементов

Преимущества и недостатки фильтрованных представлений в сравнении с отдельными артефактами

Фильтрованные представления обладают следующими преимуществами.

1. Простота.
2. Отсутствие трудностей при синхронизации.
3. Удобство и проста сохранения представления.

В то же время фильтрованным представлениям присущи следующие недостатки.

1. Фильтры затрудняют назначение приоритетов.
2. Владелец Продукта видит все области во всех подробностях. На первый взгляд этот недостаток может показаться преимуществом, но на самом деле, вникая во все подробности, Владелец Продукта может поддаться искушению прибегнуть к микроуправлению приоритетами в области, создав тем самым конфликт в распределении обязанностей между Владельцем Продукта и Владельцами Продукта в Областях.

Отдельные артефакты обладают следующими преимуществами.

1. Общий Задел Продукта остается на более высоком уровне детализации, и поэтому Владельцу Продукта не нужно вникать в подробности.
2. Владелец Продукта в Области может легко назначить приоритеты в своем заделе.
3. Обеспечивается четкое распределение обязанностей между Владельцем Продукта и Владельцами в Областях.

В то же время отдельным артефактам присущи следующие недостатки.

1. Потребность в синхронизации разных заделов.
2. Различия в приоритетах не видны в общем Заделе Продукта.
3. Увеличивается вероятность проявления отщепленческого умонастроения в каждой области вместо заботы Владельцев Продукта в Областях о сосредоточении основного внимания на всем продукте.

Наставление: максимум три уровня

В упоминаемом здесь наставлении по *разделению* рекомендуется добавить столбец “Предок” в Задел Продукта. Естественно, что эта рекомендация распространяется и на общий Задел Продукта в LeSS Huge, когда применяется способ разделения на отдельные артефакты, как показано в приведенном ниже примере.

См. раздел “Наставление: разделение” главы 11.

Элемент	Предок	Область
XA	X	Управление торговыми операциями
XB	X	Управление торговыми операциями
...		

Самое главное, что в данном случае создаются *два* уровня. И соответственно в Заделе Продукта в Области управления торговыми операциями появляется столбец “Предок”, как показано в приведенном ниже примере.

Элемент	Предок
XA-1	XA
XA-2	XA
...	

Обратите внимание на то, что предок XA не только передает родительскую информацию, но и обеспечивает связь между общим Заделом Продукта и Заделами в Областях. А самое главное, что при этом создаются всего *три* уровня в заделах. В данном примере это уровни разделения на элементы XA-1, XA и X.

Можно, конечно, ввести и больше уровней, но этого не следует делать, остановившись максимум на *трех* уровнях.

А почему именно три уровня? Мы обратили внимание на то, что группы, делающие записи на многих вложенных уровнях разделенных элементов, попадают в ловушку, *не* определяя требования, ориентированные на клиентов. Вместо этого они начинают определять ненастоящие требования, которые, по существу, являются техническими действиями или задачами. А кроме того, они сохраняют информацию, которой не пользуются, что только усложняет дело, не принося никаких выгод. Если же придерживаться лишь трех уровней разделения элементов, то такое разделение помогает сохранить простым Задел Продукта и сосредоточить основное внимание на клиентах.

Наставление: новая область для очень большого требования

Типичным затруднением в *крупных* группах продукта, практикующих LeSS Huge, являются очень большие требования, для реализации которых потребуются многие человеко-года. Чтобы как-то справиться с такими требованиями, в LeSS Huge они обычно вводятся в Область Требований и распределяются по командам. Когда требование оказывается действительно большим, для его реализации потребуются новые команды, а следовательно, область разрастается. В конечном итоге Область Требований становится слишком крупной и требует разделения.

С другой стороны, теоретически можно создать новую область при появлении очень большого требования. Такое требование не следует вводить в существующую область, но вместо этого сразу же выяснить, что работать над этим требованием придется более чем четырем командам. В итоге создается новая Область Требований и новый Задел в данной Области с единственным элементом. Затем в эту область перемещается только одна команда, временно нарушая правила, связанные с размерами областей, но учитывая, что данная область будет разрастаться.

А зачем это делать? Наличие очень большого требования в другой области и постепенное его разделение способно привести в полный беспорядок Задел в Области, который будет содержать многие разделенные элементы очень крупного требования вперемешку с другими элементами. С другой стороны, создание области как можно раньше позволяет сразу же сосредоточить основное внимание Владельца Продукта в Области и первоначальной Команды на данном очень крупном требовании.

Иногда первоначальные предположения бывают неверными и область вообще не разрастается за пределы двух команд, поскольку требование оказывается менее впечатляющим, чем предполагалось. В таком случае данную область следует объединить с другой областью, чтобы избавиться от слишком мелких областей.

Наставление: обращение с очень большими требованиями

В этой главе, а также в главе 11 представлен ряд методов для обращения с очень большими требованиями. В главе 2 описывалась история группы, имеющей дело с очень большими нормативными требованиями. А в этом наставлении рассматривается случай, иллюстрирующий ряд методов совместной работы над очень большими требованиями.

Традиционное обращение с очень большими требованиями

Прежде чем перейти к новой истории для сравнения и противопоставления контекста поделимся своим опытом, как с такими очень большими требованиями традиционно обращаются в довольно крупных группах. Очень большое требование, называемое далее просто Большим Требованием, поступает где-нибудь на крупное предприятие и кто-нибудь (старший аналитик, ответственный за выпуск продукции, системный архитектор или инженер-системотехник) анализирует его месяцами, составляя спецификацию на сотни страниц. Затем он передает спецификацию другим аналитикам и архитекторам, которые берут ту ее часть, которая относится к их сфере деятельности. В конечном счете составленные спецификации спускаются в качестве входных данных группе последующей разработки, где из них извлекаются элементы, из которых формируется Задел Продукта. Элементы поступают в задел в течение продолжительного периода времени от полугода до двух лет (нам приходилось наблюдать и такое) после того, как очень крупное требование появится на предприятии. И все это приводит к многочисленным передачам работ, рассеиванию и утрате больших объемов информации.

Обращение с очень большими требованиями в LeSS

См. раздел “Наставление: динамика Областей Требований” главы 4.

Как только Большое Требование поступит на предприятие, Владелец Продукта сразу же вводит его в Задел Продукта. Он выясняет, что данное требование на многие годы, а также решает, что оно является или будет важным, создает новую Область Требований для очень большого требования и находит подходящего Владельца Продукта в этой Области, знакомого с данным конкретным требованием. Владелец Продукта в Области создает Задел в данной Области только с одним элементом, как показано на рис. 9.6.

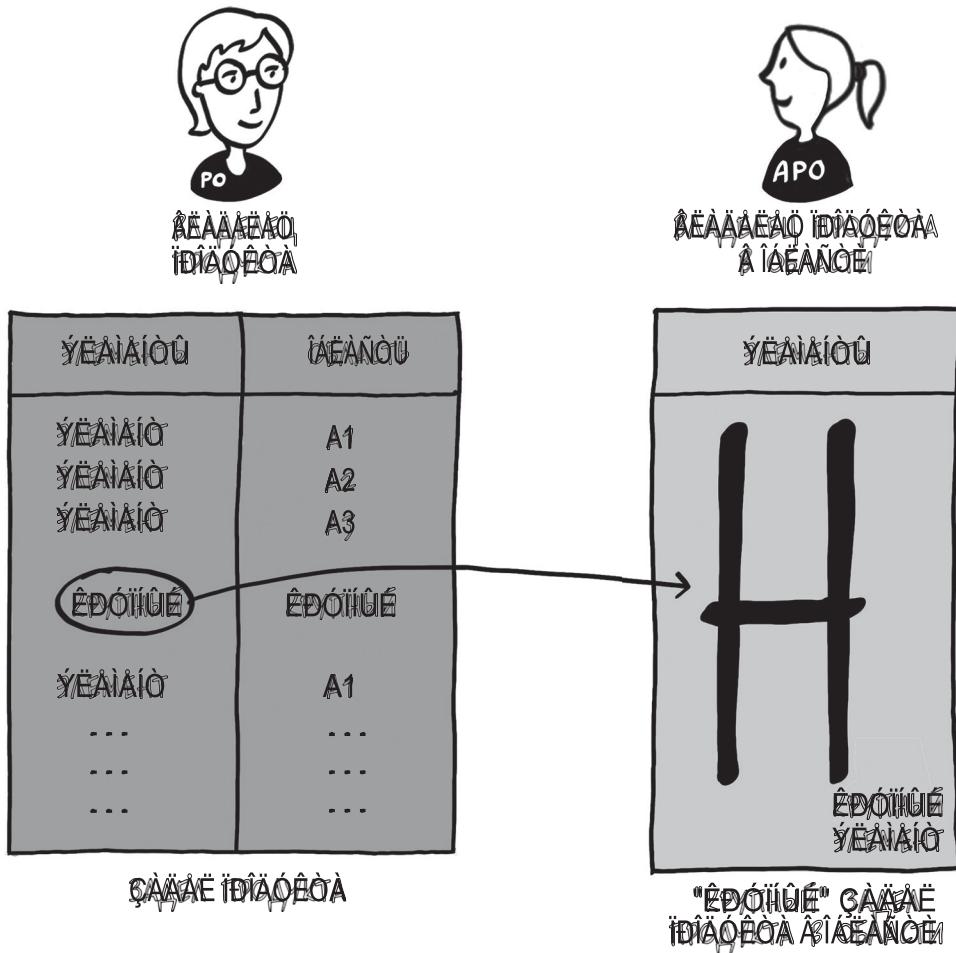


Рис. 9.6. Новая область с единственным элементом

Команда Владельцев Продукта находит среди существующих команд именно ту, которая обладает наибольшим опытом и знаниями в области данного Большого Требования, перемещая ее в новую область. Перед своим первым Спринтом в новой области эта Команда проводит совещание по уточнению Задела Продукта, где единственный элемент частично разделяется методом **надкусывания**, как показано на рис. 9.7.

См. раздел “Наставление: разделение” главы 11.

См. выше раздел “Наставление: надкусывание”.

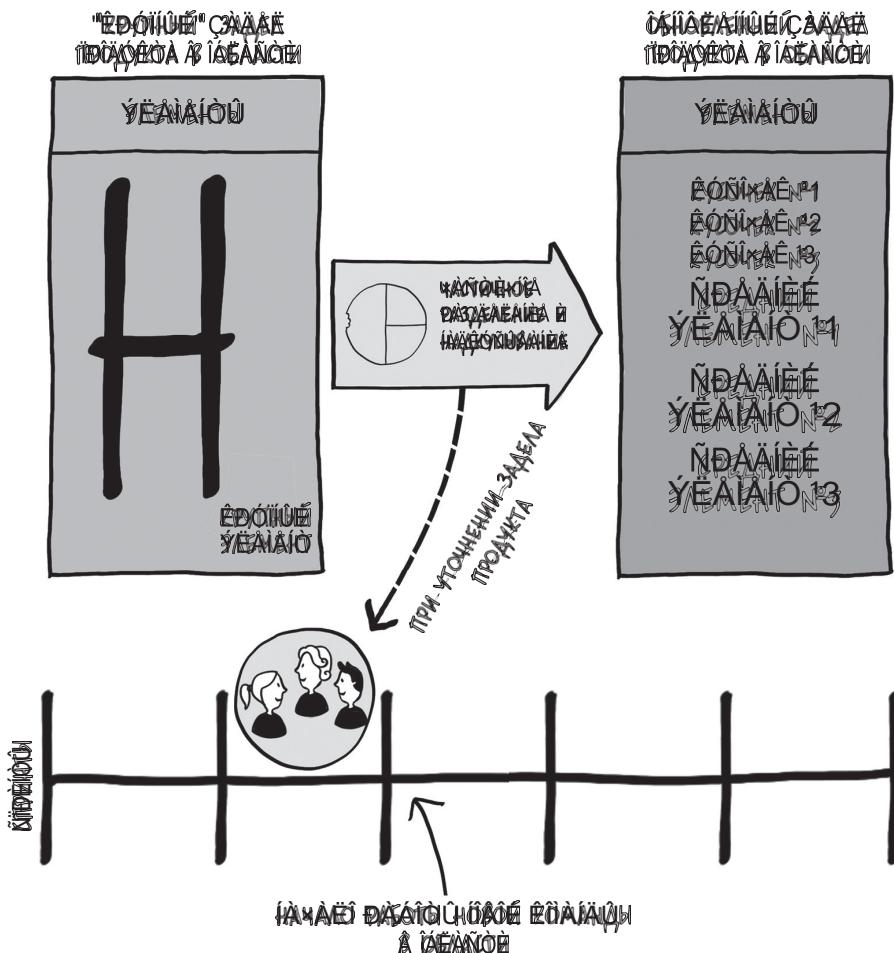


Рис. 9.7. Частичное разделение и надкусывание во время Уточнения Задела Продукта перед первым Спринтом

В первом Спринте команда реализует мелкий кусочек. Кроме того, она уделяет 50% своего рабочего времени уточнению Задела Продукта для последующих Спринтов, где члены команды постепенно разделяют элемент дальше (рис. 9.8).

Следует иметь в виду, что если команда выпустит первый кусочек в виде работоспособного программного обеспечения, то можно считать, что через месяц после того, как Большое Требование поступит на предприятие, уже появилось нечто ценное, а следовательно, достигнут определенный прогресс. Команда продолжает в том же духе и в последующих Сprintах, постепенно рассеивая туман анализа и реализации и уделяя основное внимание обучению через доставку ценности и ответную реакцию на нее.

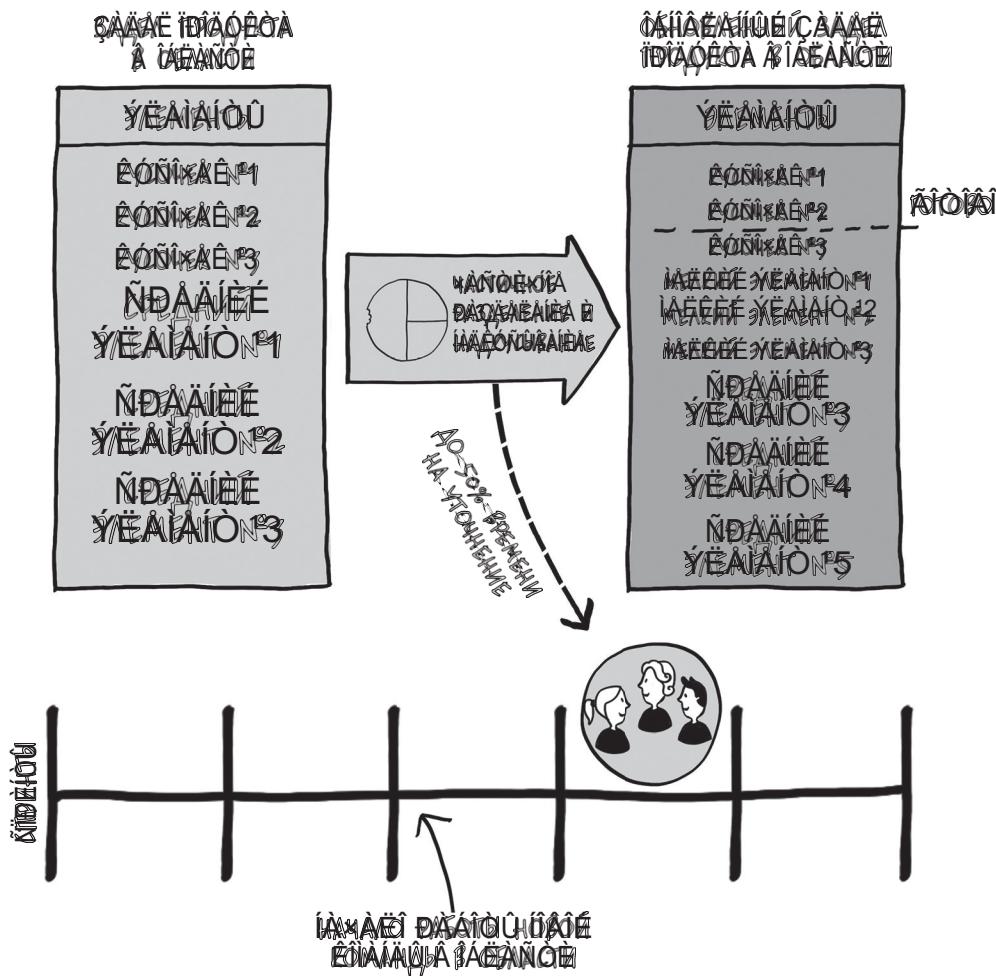


Рис. 9.8. Реализация мелкого кусочка при одновременных затратах около 50% рабочего времени на Уточнение Задела Продукта

Как только туман неизвестности заметно рассеется и появится сильная потребность в распределении очень большого объема работ, Команда Владельцев Продукта решает постепенно переместить дополнительные команды в данную область. Новые команды присоединяются к первоначальной команде на совещании по многокомандному

Уточнению Задела Продукта, где они совместно уточняют задел и дальше изучают Большое Требование. Когда новые команды присоединяются к первоначальной команде, ей отводится особая роль *ведущей команды*, обучающей и наставляющей новые команды и сохраняющей общее представление о рассматриваемом Большом Требовании, что особенно важно для согласованной интеграции всех частей разрабатываемого продукта. Первоначальная команда остается в данной области до тех пор, пока Большое Требование не будет полностью реализовано. Благодаря этому исключается передача информации, а одна и та же команда следит за Большим Требованием до самого конца (рис. 9.9).

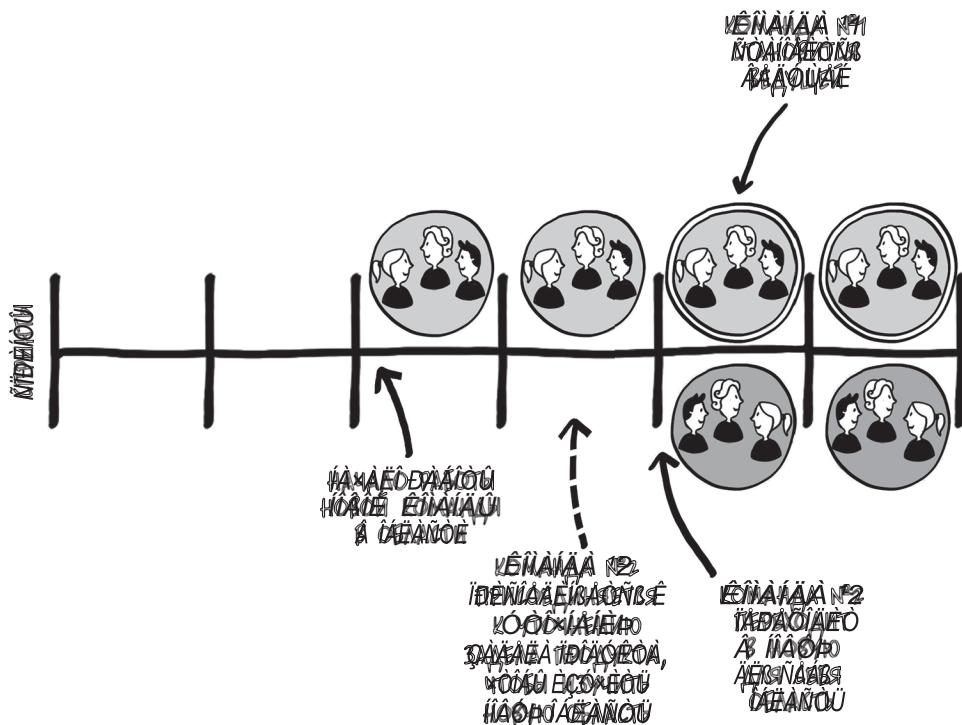


Рис. 9.9. Новая команда присоединяется к Уточнению Задела Продукта перед тем, как перейти в новую для себя область, а первоначальная команда становится ведущей

Таким образом, в данном случае были использованы следующие методы.

- > Создание новой Области Требований для очень большого требования.
- > Выбор поначалу самой опытной команды, поскольку не все команды равнозначны.
- > Частичное разделение и надкусывание.

- > Реализация кусочка при одновременных затратах около 50% времени Спринта на уточнение.
- > Постепенное разрастание новой Области Требований.
- > Применение многокомандного Уточнения Задела Продукта в целях обучения.
- > Превращение первоначальной команды в ведущую с дополнительными обязанностями в наставлении и поддержании общего представления о разрабатываемом продукте.



Готово!

10

Критерий Готовности

*Te, кто говорят, что этого нельзя сделать,
не должны мешать тем, кто это делает.*
— Джордж Бернард Шоу

Scrum для одной команды

Знакомый нам разработчик определил понятие готовности как *завершение набора исходного текста программы*, чем поверг нас в полное замешательство. Большинство разработчиков страдают от беспорядка, когда заявляют, что у них “почти” все готово. Ведь это “почти” не дает никакого представления о продвижении к состоянию готовности, которое само по себе является неопределенным конечным состоянием.

Прозрачность требуется и создается в Scrum. Один из методов увеличить прозрачность состоит в том, чтобы формально определить понятие готовности как Критерий Готовности. Состояние работ над продуктом определяется двоичной логикой: элемент готов или не готов.

В идеальный Критерий Готовности включается все, что команда должна сделать для элемента в Спринте, чтобы продукт по-прежнему оставался готовым к поставке для конечных пользователей и содержал новый готовый элемент. Поставлять продукт в каждом Спринте или еще чаще относительно просто в Scrum с одной командой. Если же команды еще не в состоянии достичь такого идеального Критерия Готовности, они определяют критерий готовности как подмножество идеального критерия. И тогда их целью становится совершенствование Критерия Готовности до тех пор, пока он не станет идеальным, а они смогут поставлять продукт в каждом Спринте или еще чаще.

Критерий Готовности¹ представляет собой согласованный перечень действий, которые Команда выполняет над каждым элементом в Заделе Продукта. Когда все соответствующие действия завершены, элемент считается готовым.

¹ Выразить Критерий Готовности можно и по-другому: во-первых, как состояние элемента в Заделе Продукта, а во-вторых, как состояние прироста продукта с присутствующими в нем элементами. Критерий Готовности, выражаемый по отдельным элементам, способствует непрерывной доставке ценности.

Не следует путать Критерий Готовности с критериями приемки. Последние являются условиями, которым должен удовлетворять конкретный элемент, чтобы считаться поставляемым. Удовлетворение всех критериев приемки обычно включается как условие в Критерий Готовности.

Готовность в LeSS

Ныне группа продукта, состоящая из одной команды, должна быть в состоянии выработать идеальный Критерий Готовности и даже непрерывно доставлять ценность в течение Спринта. Но для многих крупных групп продуктов идеальный Критерий Готовности кажется недостижимым, хотя они по-прежнему измеряют периоды стабилизации в месяцах. Бас вспоминает, как его удивило, что он получил премию за какой-то код, который он написал два года назад. А дело в том, что к тому времени продукт был окончательно выпущен.

При масштабировании соблюдаются следующие принципы, связанные с Критерием Готовности.

- > **Прозрачность.** В традиционных крупных группах предпринимается попытка обеспечить прозрачность путем установления дополнительного административного контроля и отчетности. А у групп, практикующих LeSS, имеется четкий общий Критерий Готовности и интегрированный продукт — по крайней мере, в конце каждого Спринта. Благодаря этому создается подлинная и до боли ясная прозрачность.
- > **Непрерывное улучшение до полного совершенства.** Что же следует улучшать? Постепенное расширение Критерия Готовности задает направление для улучшений и предпринимаемых с этой целью мер.

Правила LeSS

Один общий Критерий Готовности на весь продукт и все его элементы.

У каждой команды может быть свой более строгий Критерий Готовности, расширяющий общий критерий.

Цель совершенства — улучшить Критерий Готовности таким образом, чтобы получать поставляемый продукт в каждом Спринте или еще чаще.

Наставление: составление Критерия Готовности

Подробнее о первоначальном Уточнении Задела Продукта см. в главе 11.

Первоначальный Критерий Готовности должен быть согласован перед началом первого Спринта. И, как правило, это делается на первом рабочем совещании по Уточнению Задела Продукта.

Составить Критерий Готовности можно, предприняв следующие шаги.

1. Определить действия, необходимые для поставки продукта конечным потребителям.
2. Выявить действия, которые можно выполнять в каждом Спринте.
3. Решить, что делать с Незавершенной Работой.
4. Наметить первые улучшения для расширения Критерия Готовности.

Рассмотрим перечисленные выше шаги более подробно.

1. Определить действия, необходимые для поставки продукта конечным потребителям

В связи с этим возникает следующий вопрос: какие действия требуются в настоящий момент, чтобы поставить продукт? Для этого нужно напомнить всем, кого это касается, следующее.

- > Поставка означает доставку продукта конечным пользователям, а не отправку его из отдела разработки. Все должны иметь общее представление о том, что именно требуется для поставки продуктов.
- > Следует критически проанализировать потребность в промежуточных артефактах или вспомогательных задачах. Действительно ли требуется техническая документация? Нужно ли обновлять *всю* техническую документацию? Как используется техническая документация? Подобные артефакты и задачи унаследованы из традиционных способов организации проектных работ, где они передавались из одной специализированной группы в другую.

На данном этапе требуются разные роли, чтобы все видели *полную* картину, т.е. их должно быть больше, чем только роли Команды и Владельца Продукта. Руководители, принимающие участие во внедрении LeSS, также должны быть привлечены, поскольку Критерий Готовности — важное инструментальное средство для проведения организационных улучшений.

См. далее раздел “Наставление: развитие Критерия Готовности”.

Команды, Владелец Продукта и прочие заинтересованные стороны намечают требующиеся действия, записывая их на листах клейкой бумаги для заметок, строят диаграммы связей или перечисляют эти действия на доске с перекидными листами. К числу таких действий обычно относятся программирование, тестирование и составление пользовательской документации, а кроме того, организация поддержки клиентов, проектирование оборудования и даже правовая работа. Тестирующие действия обычно разделяются на разные уровни, включая модульное тестирование, тестирование или верификацию системы. Такой перечень действий обозначает *потенциальную готовность к поставке* и является *идеальным Критерием Готовности*.

Как показывает наш опыт, участников данного процесса нередко удивляет, что такой перечень оказывается более коротким, чем они ожидали, хотя он может быть

довольно длинным. Дело в том, что лишь некоторые из них имеют общее представление о том, что нужно сделать, чтобы доставить продукт.

См. раздел “Наставление: организационная концепция совершенства” главы 3.

Полученный результат нередко включается в организационную концепцию совершенства для группы продукта. В крупных группах аппаратно-программных продуктов на составление идеального Критерия Готовности могут уходить годы и даже десятки лет улучшений. А в небольших совмещенных группах только программных продуктов для этого может потребоваться лишь несколько Спринтов.

2. Выявить действия, которые можно выполнять в каждом Спринте

В связи с этим возникает следующий главный вопрос: что может быть завершено в каждом Спринте, исходя из текущего контекста и имеющихся возможностей? Такое подмножество составляет первоначальный Критерий Готовности. Если это лишь небольшое подмножество, то Критерий Готовности считается *нестрогим*. А если оно почти равно Потенциально Готовому к Поставке Продукту, то Критерий Готовности считается *строгим*.

Критерий Готовности составляется группированием листов клейкой бумаги для заметок или подчеркиванием тех действий, которые являются его частью, как показано на рис. 10.1.

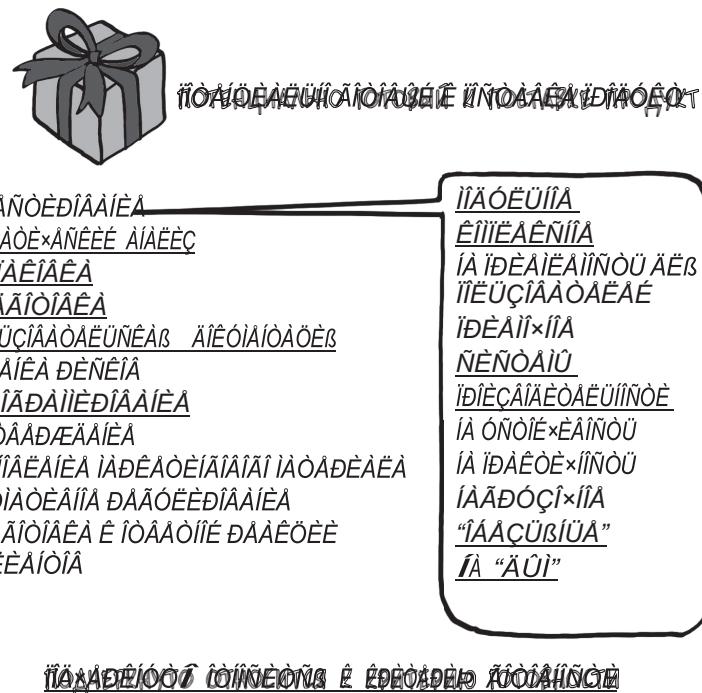


Рис. 10.1. Потенциально Готовый к Поставке Продукт и первоначальный Критерий Готовности

Разность между Критерием Готовности и Потенциально Готовым к Поставке Продуктом называется *Незавершенной Работой*. Спринт планируется в соответствии с Критерием Готовности, исключая Незавершенную Работу, чтобы не планировать работу, оставляемую не доведенной до конца. Чтобы не путать эти термины, ниже ясно описано их различие.

Математическое толкование готовности:

$$\begin{aligned} \text{Потенциально Готовый к Поставке Продукт} &= \\ &= \text{Критерий Готовности} + \text{Незавершенная Работа} \end{aligned}$$

$$\text{Работа в Спринте} = \text{Элементы из Задела Продукта} \times \text{Критерий Готовности}$$

Потенциально Готовый к Поставке Продукт — это все действия, которые должны быть выполнены, прежде чем продукт может быть поставлен конечным потребителям. Этот перечень действий зависит *не* от навыков команд или организационной структуры, а *только* от продукта.

Критерий Готовности — это соглашение между командами, Владельцем Продукта и руководителями о тех действиях, которые следует выполнить в течение Спринта. Критерий Готовности считается идеальным, если он равен Потенциальному Готовому к Поставке Продукту.

Незавершенная Работа — это разность между Критерием Готовности и Потенциально Готовым к Поставке Продуктом. Если Критерий Готовности оказывается идеальным, то Незавершенная Работа отсутствует. В противном случае организация должна решить, во-первых, что делать с Незавершенной Работой и, во-вторых, какие улучшения требуются, чтобы в будущем свести к минимуму объем Незавершенной Работы.

Неготовый или незавершенный элемент — это такой Элемент из Задела Продукта, который был начат в течение Спринта, но не был завершен. Его часто путают с Незавершенной Работой. Если еще не готовый Элемент из Задела Продукта был начат, но не доведен до состояния готовности перед окончанием Спринта, то Незавершенная Работа вообще не планировалась. Если у команды имеется элемент, который не был завершен, т.е. он готов частично, то они должны озабочиться этой ситуацией и обсудить действия по ее улучшению во время своей Ретроспективы.

Не начатый элемент — это такой Элемент из Задела Продукта, который был запланирован во время Спринта, но так и не был начат. В таком случае он просто возвращается в Задел Продукта. Тем не менее команда должна выяснить причины, по которым элемент так и не был начат, обсудив сложившееся положение во время своей Ретроспективы.

3. Решить, что делать с Незавершенной Работой

На данном этапе необходимо найти ответ на следующий вопрос: кто и когда должен сделать Незавершенную Работу? Имеется несколько способов выполнить Незавершенную Работу, но прежде рассмотрим последствия Незавершенной Работы на ряде примеров.

Как показано на рис. 10.2, в соответствии со своим Критерием Готовности команды завершили двадцать Элементов из Задела Продукта. Тем не менее у них еще осталось немало Незавершенной Работы (например, тестирование на устойчивость и составление пользовательской документации) из-за недостаточной строгости их Критерия Готовности. Поэтому команды продолжают работать над элементами в течение двух очередных Спринтов.

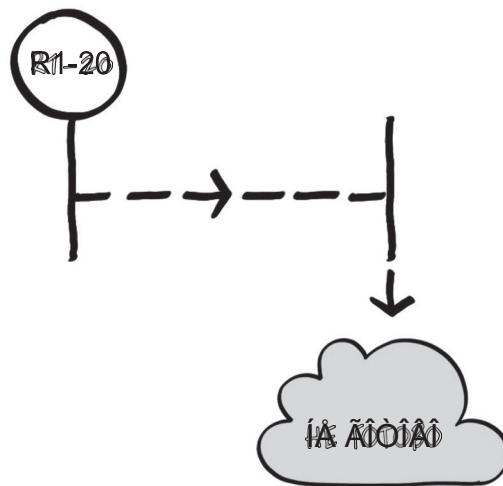


Рис. 10.2. Незавершенная Работа, появляющаяся из-за неидеального Критерия Готовности

Как показано на рис 10.3, команды завершили работу над шестьюдесятью Элементами из Задела Продукта в соответствии со своим нестрогим Критерием Готовности в течение трех Спринтов. Объем Незавершенной Работы непомерно вырос, вызвав ложное ощущение продвижения вперед. Увлекшись рыночным потенциалом продукта, Владелец Продукта решает, что у продукта достаточно функциональных средств и *теперь* самое время поставить продукт на рынок.

Тем не менее они не могут поставить продукт. И хотя командам кажется, что они довели свой продукт до состояния готовности, у них накопилось немало Незавершенной Работы вследствие нестрогого Критерия Готовности. И эта Незавершенная Работа становится причиной задержки и недостаточной прозрачности, таящей в себе немалые риски, как показано на рис. 10.4

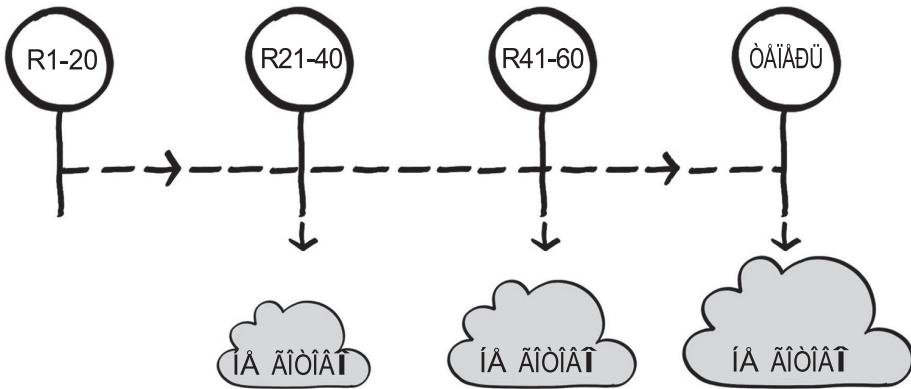


Рис. 10.3. Накапливание Незавершенной Работы

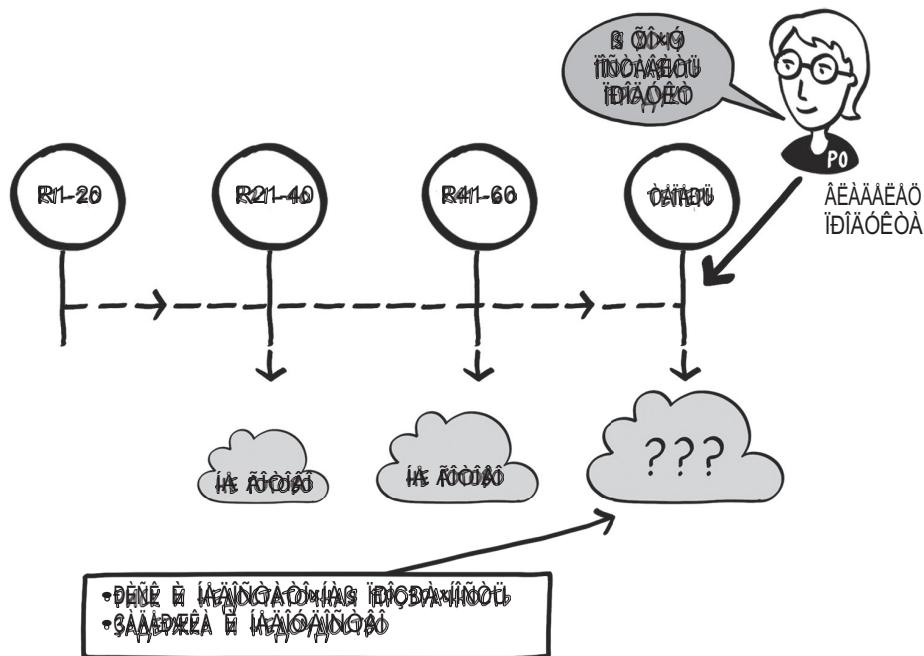


Рис. 10.4. Незавершенная Работа становится причиной задержки и немалых рисков

Задержка. Незавершенная Работа доставляет Владельцу Продукта немалые неудобства. В частности, он не может непосредственно реагировать на потребности рынка из-за тех неудобств, которые ему доставляет накопившийся объем Незавершенной Работы. Дело усугубляется еще и тем обстоятельством, что очень трудно спрогнозировать, сколько труда потребуется, чтобы выполнить Незавершенную Работу.

Риск. Незавершенная Работа становится причиной недостаточной прозрачности. Она задерживает реализацию риска. Так, если тестирование производительности не доведено до конца, риск нефункционирующей системы остается скрытым вплоть до ее выпуска, когда риск отрицательно скажется на всех, как только он станет реальностью.

Как справиться с Незавершенной Работой

Самый лучший и *единственный* правильный способ борьбы с Незавершенной Работой — предотвратить ее появление, имея строгий Критерий Готовности. Если же это пока *еще* невозможно, то справиться с Незавершенной Работой можно тремя описываемыми ниже временными мерами.

Выпускные Спринты. Это один или несколько Спринтов, проводимых перед выпуском продукта, когда Команды работают *не* над новыми функциональными средствами, а над Незавершенной Работой (рис. 10.5).

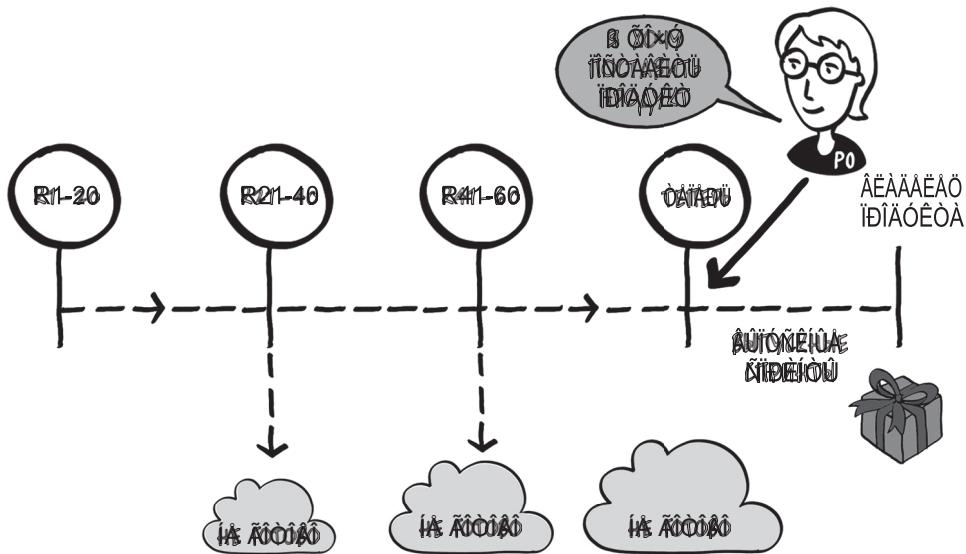


Рис. 10.5. Неудачная идея выполнять Незавершенную Работу в выпускных Спринтах

Выпускные Спринты являются неудачной идеей, но иногда они оказываются неизбежным злом до тех пор, пока Команды не расширят свой Критерий Готовности. Чаще всего к выпускным Спринтам прибегают для борьбы с коллективной бюрократией, возникающей вокруг развертывания продукта. Такую бюрократию нужно в конечном итоге преодолеть, но для этого потребуется время.

В выпускных Спринтах не следует выполнять тестирование или исправление программных ошибок. Если команды способны выполнять подобные виды работ в выпускных Спринтах, значит, они должны быть способны выполнять их в течение обычных Спринтов. Вместо этого лучше расширить Критерий Готовности.

Завершение работ в отделе незавершенных работ. В таком отделе работают люди, специализирующиеся на выполнении Незавершенных Работ после того, как Команды доведут до состояния готовности все элементы, требующиеся для выпуска продукта (рис. 10.6).

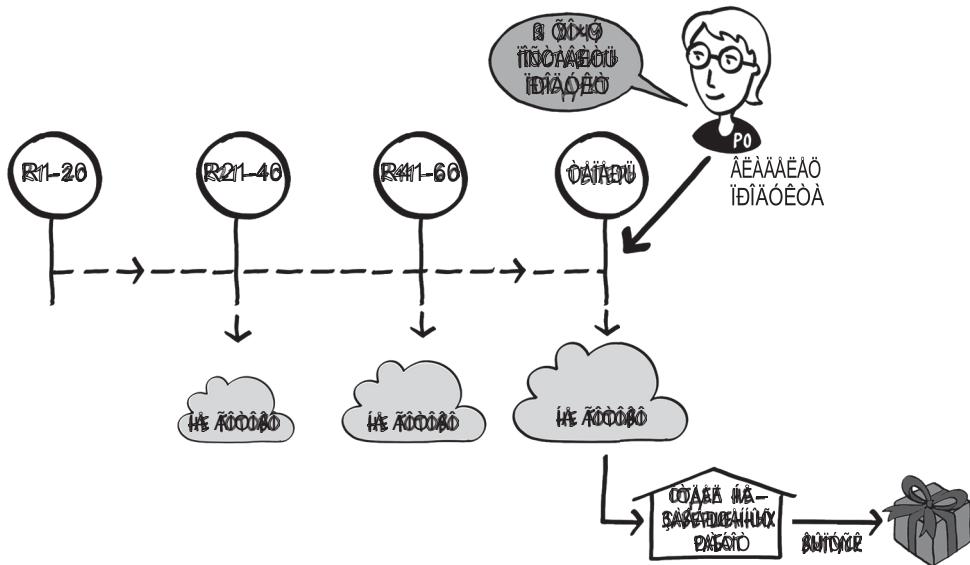


Рис. 10.6. Неудачная идея поручить Незавершенную Работу отделу незавершенных работ

Большинство отделов незавершенных работ являются пережитками прошлого и служат временной мерой до тех пор, пока Команды не расширят Критерий Готовности. Чаще всего отделы незавершенных работ служат для тестирования, которое еще не было автоматизировано или выполнено Командами из-за ограниченности их рамок. Управление отделами незавершенных работ зачастую осуществляется традиционным руководством проектов или по методике Канбан, поскольку в инфраструктуре Scrum наличие отдела незавершенных работ не имеет никакого смысла.

Цель каждого внедрения LeSS заключается в том, чтобы команды поставляли что-нибудь ценное в каждом Спринте или еще чаще. С этой целью следует упразднить все отделы незавершенных работ, поскольку они вызывают дополнительную передачу работ, задержку, вмешательство, риск и сокращенное обучение. Кажущиеся преимущества от специализированных функциональных групп того не стоят.

Передача работы по конвейеру в отдел незавершенных работ. В конце каждого Спринта все Команды передают Незавершенную Работу в отдел незавершенных работ таким образом, чтобы Незавершенная Работа не накапливалась, как показано на рис. 10.7.

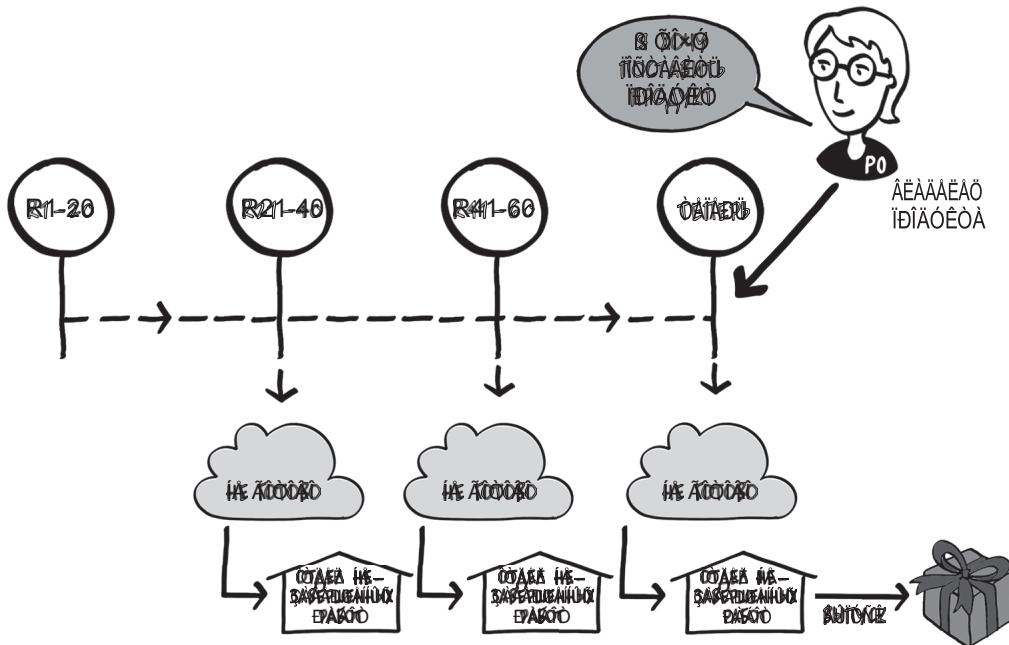


Рис. 10.7. Неудачная идея передавать работу по конвейеру в отдел незавершенных работ

На первый взгляд, идея передавать работу по конвейеру в отдел незавершенных работ кажется трезвой, но на самом деле она такая же неудачная, как и все предыдущие, поскольку это лишь временная мера для команд с ограниченной областью действия. Чтобы не прибегать к подобной мере, следует расширить Критерий Готовности и организовать настоящие команды для функциональных средств на уровне продукта. Передача работы по конвейеру чаще всего применяется, во-первых, когда область действия команд по-прежнему ограничена пределами компонентов, а следовательно, некоторые виды тестирования не вписываются в рамки команд; и, во-вторых, для тестирования требуется специальное оборудование, совместное пользование которым затруднено для команд. Последнее обстоятельство чаще служит оправданием для появления специального отдела, чтобы не думать о том, как организовать для команд совместное пользование специальным оборудованием.

Передача работы по конвейеру никогда не действует как следует. Отделу незавершенных работ требуется знать, когда Команды передадут ему Незавершенную Работу. А это, в свою очередь, потребует вмешательства в работу команд в их следующем Спринте, вызывая конфликт между отделом незавершенных работ и Командами.

Как показывает наш опыт, передача работы по конвейеру всегда служит оправданием не упразднять специализированные функциональные группы или расширять область действия Команд. По мере совершенствования групп продуктов передача работы по конвейеру должна постепенно исчезнуть.

4. Наметить первые улучшения для расширения Критерия Готовности

На данном этапе необходимо найти ответ на следующий вопрос: что мешает расширить Критерий Готовности? Критерий Готовности определяет *степень* текущей *гибкости* группы продукта, а Незавершенная Работа выявляет возможности для улучшений. А на что способны улучшения?

Рассмотрим следующие улучшения.

- > **Автоматизация.** Большая часть Незавершенной Работы по традиции выполнялась вручную, и поэтому она должна быть автоматизирована.
- > **Гармонизация.** Группы продуктов нередко разрешали одно и то же затруднение самыми разными способами (например, в четырех сходных средах тестирования). Поддерживать во всех командах тесты с помощью многих сходных, но все-таки разных технологий редко оказывается полезным, а следовательно, Команды должны согласовывать какие-то общие нормы для проведения тестов.
- > **Среда.** Некоторыми средами трудно пользоваться, особенно совместно (например, тестовым оборудованием). Поэтому совместное использование следует улучшать и согласовывать в Командах. С другой стороны, следует сократить зависимость от сред, повысив степень виртуализации.
- > **Распараллеливание.** Иногда команды предполагают, что определенная работа должна быть выполнена в определенной последовательности (например, начать тестирование по завершении всего кода). Такое предположение зачастую оказывается неверным, а работа может быть распараллелена, если подойти к ее выполнению иначе.
- > **Межфункциональность.** Некоторые виды Незавершенной Работы требуют навыков, которых у команд пока еще нет (например, навыков составления технической документации). Повышение межфункциональности достигается путем взаимного обучения или привлечения специалистов нужной квалификации. Чаще всего людей перемещают с этой целью из отдела незавершенных работ в команды.

Незавершенную Работу следует анализировать, выявляя возможности для улучшений и расширения Критерия Готовности. Когда команды будут работать над этими улучшениями, они войдут как отдельные элементы в Задел Продукта.

Наставление: развитие Критерия Готовности

Критерий Готовности многогранен и требует строгого контроля и развития. Цель достижения идеального Критерия Готовности заключается в том, чтобы *организация* оказалась способной поставлять продукт в каждом Спринте или еще чаще. В разных ролях Критерий Готовности рассматривается под разными углами зрения, как поясняется ниже.

Руководители. Несмотря на неидеальность Критерия Готовности, он является *главным инструментальным* средством для контроля и управления организационными изменениями. Расширение Критерия Готовности приводит к организационным изменениям, стратегическим решениям и обычно к определенным обязанностям руководства.

Допустим, что группа продукта состоит из пяти мест разработки, в двух из которых имеется группа, специализирующаяся на проверке работоспособности системы из-за большой стоимости испытательного оборудования. Расширение Критерия Готовности

могло бы привести к приобретению навыков проверки работоспособности системы на всех местах, исключая применение для этих целей отдельных групп и выясняя, как организовать совместное пользование испытательным оборудованием в нескольких географических местах. И это довольно непростое изменение!

Руководители должны поощрять группы совершенствовать и расширять их Критерий Готовности. Ведь благодаря этому легче расширить в дальнейшем Критерий Готовности всего продукта.

См. раздел “Наставление: посещение рабочих мест” главы 5.

Следует избегать одностороннего расширения Критерия Готовности, особенно если нет представления о том, что происходит на рабочих местах. Ведь результат окажется далеко не самым лучшим.

Команды. Каждый Спринт представляет собой цикл совершенствования путем обследования и адаптации, а Критерий Готовности предоставляет источник для поиска улучшений рабочих методов, применяемых в Команде. Каждая команда может расширить Критерий Готовности независимо от Критерия Готовности на уровне продукта в целом.

Так, в приведенном выше примере проверки работоспособности системы одна Команда может совершенствоваться путем овладения навыками проверки работоспособности системы или исследования разных способов совместного пользования дорогостоящим испытательным оборудованием.

Владелец Продукта. Нестрогий Критерий Готовности вызывает риск и задержку, что мешает Владельцу Продукта получать наибольшую ценность и решать, когда ее следует доставлять. Грамотный Владелец Продукта вкладывает средства в улучшения с целью повысить гибкость организации.

Так, в предыдущем примере проверки работоспособности системы Владелец Продукта, вероятно, испытает неприятные задержки вследствие проверки работоспособности системы. Для улучшения ситуации он может вложить средства в испытательное оборудование или обсудить с командами, какие Элементы из Задела Продукта им потребуются, чтобы усовершенствовать их Критерий Готовности.

Scrum-мастера. Отсутствие расширения Критерия Готовности служит явным признаком отсутствия совершенствования. Scrum-мастера отвечают за организацию команд, способных к самоуправлению и непрерывному совершенствованию, а также за оказание помощи в совершенствовании всей организации.

Так, если в предыдущем примере проверки работоспособности системы в командах отсутствует обсуждение путем совершенствования Критерия Готовности, то могут возникнуть вопросы вроде следующего: что мешает командам совершенствовать их навыки проверки работоспособности системы?

Критерий Готовности и способы его достижения Командой дают важную информацию для того, чтобы оценить состояние внедрения Scrum. Решение о расширении Критерия Готовности нередко принимается на следующих мероприятиях.

- > **Обсуждения и совещания руководителей.** На подобных мероприятиях руководители задают следующий главный вопрос: как расширить Критерий Готовности? Улучшение организационных возможностей по доставке ценности является основной обязанностью руководителей, и главным инструментом для исполнения этой обязанности служит Критерий Готовности.
- > **Ретроспективы.** Ретроспективы, проводимые как на уровне команд, так и на общем уровне, служат для выявления элементов, предназначенных для целей совершенствования. Критерий Готовности на уровне всего продукта является общим для всех команд, но каждая команда поощряется для совершенствования в этом.
- > **Сообщества.** Обсуждение в сообществе служит идеальным местом для анализа организационного поведения и разрешения системных затруднений. Кроме того, в сообществах удобно выяснить способы расширения Критерия Готовности. Особенно подходящим для этой цели местом служит сообщество Scrum-мастеров, поскольку они вместе с руководителями отвечают за перемены в организации, совместно обеспечивая устранение обнаруженных недостатков.

См. раздел “Наставление: сообщества” главы 13.

Прекращает ли организация совершенствоваться, достигнув идеального Критерия Готовности? Нет, поскольку совершенствованию нет предела, а улучшения никогда не прекращаются. К числу дополнительных улучшений, которые могут быть сделаны, относятся следующие.

- > Проводить более короткие Спринты.
- > Делать неоднократные выпуски во время Спринта.
- > Расширить Критерий Готовности за пределы потенциально готового к поставке продукта, а также включить успешный выход на рынок в Критерий Готовности. В этом случае элемент считается неготовым до тех пор, пока не удастся выяснить, каким образом потребители пользуются им на практике. В инфраструктуре бережливого запуска это называется *утвержденным обучением*.

LeSS Huge

В отношении Критерия Готовности для LeSS Huge особых правил или наставлений не существует. Один общий Критерий Готовности распространяется на весь продукт и все Области Требований.

Сprint в LeSS



Рабочее совещание по многокомандному уточнению общего Задела Продукта в LeSS

Уточнение Задела Продукта

Я не обязательно соглашаюсь со всем, что я говорю.
— Маршалл Маклухан

Scrum для одной команды

Прежде всего обратите внимание на то, что эта глава, посвященная Уточнению Задела Продукта, предшествует главе, посвященной Планированию Спринта. Но Уточнение Задела Продукта происходит *не* перед самым Планированием Спринта, а задолго до него — как правило, посередине какого-нибудь предыдущего Спринта. Такая организация глав данной книги объясняется тем, что с точки зрения потока требований все дело начинается именно с Уточнения Задела Продукта.

Чтобы выбрать элемент из Задела Продукта во время Планирования Спринта, Команде достаточно оценить, в состоянии ли она действительно реализовать “готовый” элемент в течение Спринта.

Следовательно, в каждом Спринте требуется постоянное Уточнение Задела Продукта, чтобы подготавливать и уточнять элементы для будущих Спринтов. С этой целью выполняются такие действия, как прояснение, детализация и оценка. Не отходя от истинной сути управления эмпирическим процессом, в Scrum ничего не говорится о том, как проводить Уточнение Задела Продукта, хотя и предполагается, что Команда должна затрачивать на него не больше 10% времени, отводимого на Спринт. И, как правило, Уточнение Задела Продукта происходит посередине Спринта.

Уточнение элементов *не* проводится отдельно Владельцем Продукта, Командой Владельцев Продукта или отдельной группой аналитиков деловой активности, ответственных за выпуск продукции, или проектировщиков пользовательского интерфейса, поскольку это увеличивает потери на передачу работы, запасы или незавершенные работы и прочее, а также снижает чуткость, проявляемую Командой к потребителям или пользователям, и понимание их нужд. Вместо

этого подобную работу выполняет *вся* Команда, а не отдельные ее члены (например, специалисты по анализу деловой активности или взаимодействию с пользователем), поскольку в Scrum не предусмотрены подгруппы, специально выделяемые для конкретных предметных областей вроде анализа деловой активности или взаимодействия с пользователем. Как поясняется в *Руководстве по Scrum*:

[Уточнение Задела Продукта] является непрерывным процессом, в ходе которого Владелец Продукта и Команда разработчиков совместно выясняют подробности, касающиеся элементов из Задела Продукта. Никаких подгрупп в Команде разработчиков для этих целей в Scrum не предусмотрено, независимо от конкретных предметных областей, которые должны быть рассмотрены подобно тестированию или анализу деловой активности. И никаких исключений из этого правила не делается.

Уточнение Задела Продукта в LeSS

При масштабировании соблюдаются следующие принципы, касающиеся Задела Продукта.

- > **Сосредоточение внимания на продукте в целом.** Если каждая команда уточняет разные элементы только по отдельности (для локальной оптимизации), то в конечном итоге ограничиваются знаниями предметной области, уменьшается гибкость и затрудняется координирование действий. Поэтому очень важно найти способы борьбы с этим явлением.
- > **Ориентирование на клиентов.** В традиционной организации так называемые требования нередко оказываются техническими или функциональными задачами для разрозненных групп, а не истинными целями клиентов. Следовательно, при внедрении LeSS многие разработчики не будут ознакомлены со всеми требованиями клиентов, их терминологией и предметной областью, не говоря уже о сотрудничестве с ними для решения их проблем вместо того, чтобы выбирать заранее выработанные решения.
- > **Бережливый подход и теория массового обслуживания.** В традиционной организации уяснением и определением требований занимается несколько функциональных групп, и для этого им придаются аналитики деловой активности и взаимодействия с пользователями, проектировщики пользовательского интерфейса, ответственные за выпуск продукции и прочие специалисты. Это приводит к многочисленным потерям и очередям, заполняемым промежуточными незавершенными документами. И хотя такой подход локально выглядит эффективным, он не позволяет уяснить истинные затраты и затруднения. В итоге происходит так называемое внедрение Scrum или гибкой методики, при котором подобная динамика сохраняется, но под новой вывеской вроде “Команды Владельцев Продукта”, “Команды для написания историй” и т.д. А потери и очереди все равно остаются.

Правила LeSS

Уточнение Задела Продукта проводится в каждой команде для тех элементов, которые ей, вероятнее всего, придется реализовывать в будущем. Для повышения уровня общего понимания и использования возможностей координирования действий при наличии тесно связанных элементов или потребности в более широком участии или обучении следует проводить многокомандное Уточнение Задела Продукта.

Владелец Продукта должен заниматься уточнением Задела Продукта не в одиночку, а при поддержке многих Команд сотрудничать непосредственно с потребителями или пользователями и другими заинтересованными сторонами.

Все назначение приоритетов проходит через Владельца Продукта, но прояснение — как можно больше в непосредственной связи Команд с потребителями или пользователями и другими заинтересованными сторонами.

Наставление: виды Уточнения Задела Продукта

Уточнение Задела Продукта в LeSS представляет собой рабочее совещание, где команды проясняют предстоящие элементы вместе с пользователями и другими заинтересованными сторонами, разделяют крупные элементы, оценивают или переоценивают отдельные элементы. Конкретный образец Уточнения Задела Продукта зависит от следующих факторов.

- > Элементы не назначаются заранее для отдельных команд, поскольку это снизило бы степень гибкости, уровень обучения, а также устойчивость команд. И зачастую желательно, чтобы группа команд уточняла ряд элементов вместе, но пока еще не решала, какой именно команде следует реализовывать конкретный элемент, поскольку это расширяет знания, усиливает координирование действий и повышает гибкость.
- > Если все команды будут уточнять все элементы, то для этого придется приложить слишком много усилий, что может сделать неэффективными совещания по уточнению. Кроме того, очень трудно поддерживать всеобщий интерес к прояснению, когда команда знает, что ей не придется реализовывать конкретный элемент.

Эти факторы принимаются во внимание благодаря наличию разных видов Уточнения Задела Продукта для разных случаев. Имеются четыре следующих вида совещаний по Уточнению Задела Продукта.

- > **Общее Уточнение Задела Продукта.** Сосредоточено на продукте в целом и проводится перед многокомандным или однокомандным Уточнением Задела Продукта. Общее Уточнение Задела Продукта предназначено для того чтобы выяснить, какие именно команды могут уточнить конкретные элементы, а также повысить уровень обучения и согласования.
- > **Многокомандное Уточнение Задела Продукта.** Все члены двух или больше команд уточняют вместе ряд элементов, но пока еще не решают, какие из этих команд будут реализовывать конкретные элементы.
- > **Однокомандное Уточнение Задела Продукта.** Все члены одной команды уточняют элементы, которые им, вероятнее всего, придется реализовывать. Это такое же совещание, как и в Scrum.
- > **Первоначальное Уточнение Задела Продукта.** Проводится при внедрении LeSS и только один раз в течение срока действия продукта. При первоначальном Уточнении Задела Продукта все команды создают первый Задел Продукта и уточняют такое количество элементов, которого должно быть достаточно, чтобы начать первый Спринт.

В приведенной ниже таблице проясняется назначение различных уточняющих совещаний.

Уточнение Задела Продукта	Общее	Много-командное	Одно-командное	Первоначальное
Участие членов	Всех команд	Двух и больше команд	Одной команды	Всех команд
Участвует ли Владелец Продукта?	Определенно	Зависит от обстоятельств	Редко	Определенно
Участвуют ли потребители или пользователи?	Редко	Возможно	Возможно	Определенно
Выбираются ли отдельные команды для работы над конкретными элементами?	Да (предпочитательно выбрать ряд элементов для группы команд)	Нет	Уже сделано	Нет
Уровень прояснения	Поверхностный	Доскональный	Доскональный	Доскональный
Продолжительность	Довольно короткая	От полдня до целогодня	От полдня до целогодня	Не менее двух дней
Типичная последовательность	В каждом Спринте	В большинстве Спринтов	В большинстве Спринтов	Один раз

Как правило, группа, состоящая из 2-3 команд, проводит лишь одно совещание по Уточнению Задела Продукта, где Владелец Продукта, пользователи и все члены команд совместно уточняют досконально все элементы. По существу, такое совещание сочетает в себе общее и многокомандное Уточнение Задела Продукта.

Группа продукта, состоящая из трех и больше команд, обычно проводит общее Уточнение Задела Продукта, а затем многокомандное и однокомандное Уточнение Задела Продукта. Следует избегать однокомандного Уточнения Задела Продукта, если только нет полной уверенности, что отдельная команда сумеет реализовать конкретные элементы. В общем, предпочтение следует отдавать многокомандному Уточнению Задела Продукта в группе команд для уточнения ряда элементов. Общий образец Уточнения Задела Продукта приведен на рис. 11.1.

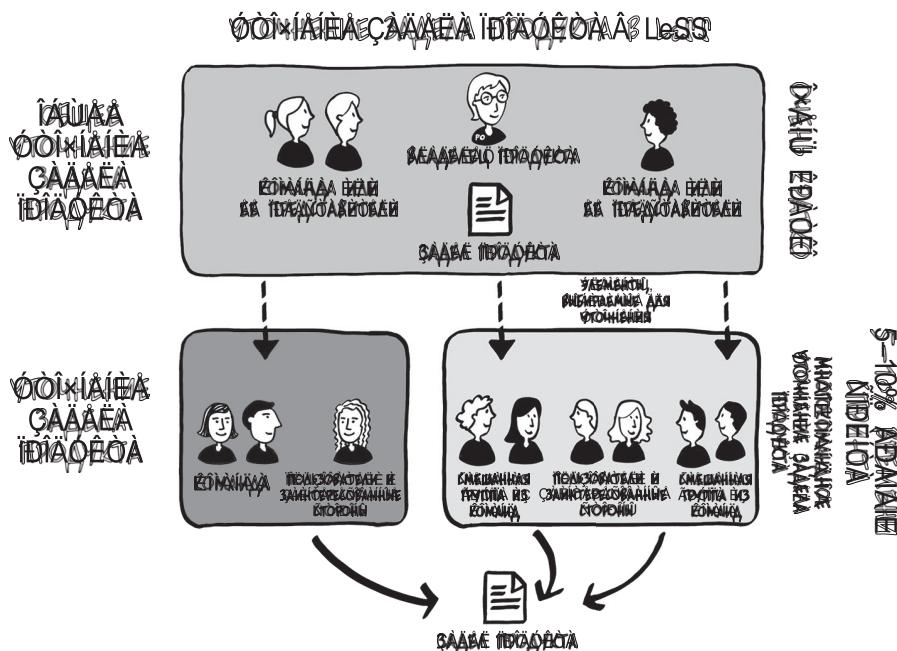


Рис. 11.1. Виды Уточнения Задела Продукта в LeSS

Наставление: общее Уточнение Задела Продукта

При общем Уточнении Задела Продукта принимается решение, требуется ли дальнейшее доскональное уточнение в группе команд, уточняющих ряд элементов, что было бы идеально, или же в отдельных командах. Но иногда становится очевидно, какие именно команды смогут лучше всего уточнить элементы, опираясь на свой предыдущий опыт работы или последние интересы, и тогда общее Уточнение Задела Продукта можно опустить.

Общее Уточнение Задела Продукта проводится “коротко и ясно”, например, за один час в двухнедельном Спринте. Его посещают Владелец Продукта и представители всех команд, если команды крупные, или же все команды в полном составе. На этом совещании проводятся следующие мероприятия.

- > Обсуждение направления и концепции продукта вместе с Владельцем Продукта.
- > Обсуждение уточняемых элементов.
- > Отбор команд и элементов для последующего досконального Уточнения Задела Продукта в команде.
 - > Для повышения уровня обучения, степени гибкости и устойчивости команд предпочтение следует отдавать уточнению ряда элементов в группе команд, а не в отдельных командах, что приводит также к многокомандному Уточнению Задела Продукта.
 - > Выявление сильно связанных элементов, которые предполагают возможности для сотрудничества и координации, что приводит также к многокомандному Уточнению Задела Продукта.

При общем Уточнении Задела Продукта группа может также выполнить следующие действия:

См. далее раздел
“Наставление: оценивание в крупных масштабах”.

- > разделить крупный элемент, что приводит к последующему обсуждению и обучению;
- > оценить элементы, что также приводит к последующему обсуждению и обучению, а также помогает синхронизировать оценки в командах;
- > прояснить элемент, хотя и не досконально;
- > например, прояснение может быть проведено в определенных временных рамках (10 минут) или контексте (двух примеров).

Что касается *представителей* команд, то в LeSS постоянно рекомендуется посыпать представителей на любые совещания, смения их со временем. Благодаря этому расширяются перспективы, укрепляются многие навыки в членах команд и сокращаются недостатки отдельных людей.

А что касается *выбора* отдельных элементов, то общее Уточнение Задела Продукта позволяет командам, а не Владельцу Продукта, выбрать те элементы, которые в дальнейшем будут перенесены на последующее многокомандное или однокомандное Уточнение Задела Продукта. Это способствует самоорганизации и сокращает объем работы, которую приходится выполнять Владельцу Продукта. И как упоминалось выше, предпочтение следует отдавать уточнению ряда элементов в группе команд, а не в отдельных командах.

Наставление: многокомандное Уточнение Задела Продукта

При многокомандном Уточнении Задела Продукта все члены двух или больше команд уточняют вместе ряд элементов, пока еще не решая, какая из команд будет

реализовывать конкретный элемент. Таким образом, они откладывают решение относительно реализации элементов отдельными командами до следующего Планирования Спринта. Благодаря этому повышается степень организационной *гибкости* (т.е. легкость реагирования на изменения), а более обширные знания всего продукта способствуют самоорганизованному координированию действий. Кто же участвует в этом совещании? Помимо членов всех команд, в многокомандном Уточнении Задела Продукта обязательно принимают участие потребители или пользователи и прочие заинтересованные стороны. По окончании многокомандного Уточнения Задела Продукта на смену ему приходит однокомандное Уточнение Задела Продукта.

Естественно, что одного присутствия в помещении двух или трех команд для уточнения явно недостаточно, чтобы волшебным образом расширить общее понимание и т.д. Поэтому многокомандное Уточнение Задела Продукта *должно* включать в себя ряд “смешанных” методов, в том числе следующие.

1. Смешение команд. Сначала формируются группы, временно смешанные из представителей отдельных команд. Например, из двух команд формируются две смешанные группы.

> После всех последующих методов следует рассмотреть возможность формирования новых смешанных групп для следующего цикла, чтобы повысить разнообразие и взаимодействие.

2. Уточнение по кругу. Сначала каждая смешанная группа уточняет разные (или одинаковые) элементы по отдельности на разных рабочих участках в одном помещении, стоя, например, у разных белых досок, вывешенных таблиц и автоматизированных проекторов. По истечении 30-минутного регламента все группы переходят по кругу к следующему рабочему участку и связанному с ним уточняемому элементу, оставляя на прежнем участке двух или трех людей, чтобы ввести пришедшую группу в курс дела и ускорить понимание текущего уточнения. К числу тех, кто обычно остается на месте, относятся потребители или пользователи и прочие заинтересованные стороны, которые способны оказать наибольшую помощь группе, уточняющей элемент.

3. Циклы расхождения–схождения. Группы уделяют немного времени сначала работе на различных участках помещения для уточнения разных (или одинаковых) элементов, а затем собираются вместе, обмениваясь мнениями, задавая вопросы и находя другие возможности для координирования действий.

Ниже перечислены основные причины для проведения многокомандного Уточнения Задела Продукта.

> **Повышение степени организационной гибкости.** Многокомандное Уточнение Задела Продукта увеличивает количество команд, которые могут реализовать ряд элементов. А с другой стороны, оно *задерживает* решение относительно того, какая именно команда реализует конкретный элемент. Следовательно, Владелец Продукта может изменить порядок следования

элементов другими способами, реагируя на изменившиеся факторы, но не накладывая следующего жесткого ограничения: конкретный элемент может реализовать только одна определенная команда. И таким образом, повышается степень гибкости!

- > **Повышение степени сосредоточенности на продукте в целом и расширение знаний.** Команды, совместно выполняющие многокомандное Уточнение Задела Продукта, расширяют свои знания предметной области, во-первых, знакомясь с более разнообразными элементами и, во-вторых, привлекая людей и знания из других команд. Благодаря этому повышается их степень гибкости, чтобы понимать, видеть и сосредоточиваться на *целом*.
- > **Улучшение координаирования действий.** При многокомандном Уточнении Задела Продукта команды узнают во всех подробностях, что знают и делают другие команды. Благодаря этому усиливается координаирование действий и способность выполнять общую работу.

Наставление: многоместное Уточнение Задела продукта

Общее или многокомандное Уточнение Задела Продукта может быть многоместным. Общие рекомендации по проведению многоместных мероприятий приведены в разделе “Наставление: межкомандные совещания” главы 13. А в этом наставлении основное внимание уделяется рекомендациям, связанным с Уточнением Задела Продукта.

Разделение

См. далее раздел
“Наставление: разделение”.

При разделении крупного элемента нередко полезно начертить древовидную схему на белой доске. Аналогично на многоместном совещании используется инструментальное средство для построения диаграммы связей в каком-нибудь общем пространстве (например, в окне браузера), поскольку такие средства оптимизированы на создание древовидных структур. Благодаря этому участники совещания в разных местах могут одновременно видеть и видоизменять диаграмму связей.

Прояснение

*Спецификация по образцу*¹ — это отличный метод прояснения и изучения элемента в группе путем обсуждения конкретных *примеров*. Спецификация по образцу уже давно поощряется в LeSS. А как осуществить ее при многоместном Уточнении Задела Продукта? С этой целью следует воспользоваться электронной таблицей в каком-нибудь общем пространстве (например, в окне браузера), поскольку многие примеры естественно вписываются в табличный формат. А люди на всех местах могут легко внести изменения в такую таблицу.

¹ Подробнее об этом см. в книгах *Bridging the Communication Gap via Specification by Example*.

Оценивание

Во-первых, следует избегать инструментальных средств так называемого “гибкого” планирования, поскольку они вынуждают людей уделять основное внимание самому инструментальному средству, а не друг другу, тогда так физические средства вроде карточек лучше побуждают и поощряют людей к действию. И, во-вторых, в LeSS допускается любой метод оценивания, но при этом предполагается применение метода, называемого покером планирования вследствие его широкой распространенности.

См. далее раздел “Наставление: оценивание в крупных масштабах”.

См. раздел “Наставление: инструментальные средства для ведения крупных заделов продуктов” главы 9.

Применение веб-камер при голосовании карточками в покере планирования или руками. Участники многоместного совещания пользуются большими карточками с крупными цифрами на них, чтобы их было хорошо видно в веб-камеру. Но они могут подавать сигналы, показывая разные оценки кулаком и пальцами.

Общая интерактивная переписка. Все участники оснащены устройствами и инструментальными средствами для ведения общей интерактивной переписки. Когда модератор просит показать оценки, все вводят соответствующие числа.

Наставление: первоначальное Уточнение Задела Продукта

Группе продукта, внедряющей LeSS, перед первым Спринтом необходим Задел Продукта с достаточно понятными элементами для работы команд над ними. И в LeSS такая предварительная подготовка вполне благоразумно называется *первоначальным Уточнением Задела Продукта*. Безусловно, данное наставление можно было бы привести еще в главе 2, но поскольку оно имеет непосредственное отношение к Уточнению Задела Продукта в целом, то рассматривается именно в этой главе.

Причины для беспокойства

В группе продукта, внедряющей LeSS, может возникнуть следующий вопрос: “А зачем об этом вообще беспокоиться? Ведь у нас уже имеется задел, и наши люди разобрались в требованиях”. На самом деле эти два предположения могут оказаться ложными, а для первоначального Уточнения Задела Продукта имеются другие причины, в том числе следующие.

- > **Существующий задел не является Заделом Продукта, полезным для LeSS.** Когда мы инструктируем группу, приступающую к внедрению LeSS, то обычно задаем следующий вопрос: “Есть ли у вас готовый задел?” И неизменно получаем такой ответ: “Конечно, у нас имеется объединенный список разрабатываемых элементов!” В частности, нам пришлось работать с группой, в первоначальном заделе которой насчитывалось 508 элементов! После активного обсуждения в течение двух часов этот задел превратился в Задел Продукта, состоящий из 23 элементов вместо 508 и пригодный для LeSS. Почему? А потому что большинство прежних “элементов” были

функциональными задачами, предназначенными для команд, выполняющих какую-нибудь одну функцию (анализ, проектирование, тестирование и прочее), или же компонентными задачами, рассчитанными на команды, реализующие отдельные компоненты, и т.д., и все предполагало наличие прежней организационной структуры. Они не имели ни малейшего представления о новой структуре LeSS, состоящей из команд для функциональных средств.

- > **Люди не понимают реформированные элементы.** Исходя из предыдущей причины, следует сказать, что элементы должны быть выражены комплексно как действительно ориентированные на клиентов и уяснены именно таким образом. В силу прежней разобщенности команд нарождающимся командам для функциональных средств придется многому научиться, чтобы формировать элементы по-новому.
- > **Ограниченнное понимание концепции продукта, ориентированной на клиентов.** Даже если прежние элементы были выражены ориентированным на клиентов образом, разрозненные прежде специалисты сосредоточены на решении узких задач, и поэтому они не до конца понимают концепцию продукта, ориентированную на клиентов.
- > **Отсутствие оценок реформированных элементов, слабые или недостаточные оценки.** Вновь сформированные элементы требуют новых оценок. И даже если элементы не требуют реформирования, их оценки нередко исходят из какой-нибудь другой группы, а не вновь сформированных команд для функциональных средств. И оценки всех или почти всех элементов могут потребоваться Владельцу Продукта для поддержки более долгосрочного планирования.
- > **Новое, более широкое определение продукта.** Как поясняется в разделе “Наставление: назначение продукта” главы 7, при внедрении LeSS границы продукта могут быть расширены. Поэтому несколько существующих заделов, возможно, придется преобразовать в новый, более широкий задел. И такое изменение подразумевает создание и сообщение новой, более широкой концепции продукта, восполнение многих пробелов в знаниях и знакомство друг с другом многих незнакомых прежде людей, которым вскоре предстоит вместе доставлять ценность.
- > **Отсутствие общей концепции продукта.** Каким бы обширным ни был новый продукт, нередко его концепция известна лишь немногим в силу разобщенности традиционных групп, даже если такая концепция и существует! Поэтому первоначальное Уточнение Задела Продукта дает возможность сформировать, сообщить и приступить к согласованию общей концепции.

Основные предпосылки

Ниже перечислены основные предпосылки для первоначального Уточнения Задела Продукта.

1. Назначен Владелец Продукта.
2. Принято решение о формировании команд для функциональных средств.
3. Имеется достаточно сведений для уточнения многих элементов, подготавливаемых к первому Спринту. Такие сведения в идеальном случае предоставляются потребителями или пользователями и заинтересованными сторонами, хотя они могут быть также извлечены из существующих документов и заделов.

Какова продолжительность первоначального Уточнения Задела Продукта? Как правило, два дня или еще дольше.

Кто участники данного совещания? Все, в том числе Владелец Продукта, все члены команд, потребители или пользователи, специалисты в предметной области, Scrum-мастера и руководители, оказывающие всяческую поддержку.

Где проводится такое совещание? Первоначальное Уточнение Задела Продукта проводится в одном месте, где могут собраться все участники, даже если речь идет о многоместной группе. Для этой цели обычно подходит большой зал заседаний, семинаров или совещаний.

Преследуемые цели

Основная цель первоначального Уточнения Задела Продукта — уточнение элементов до такой степени, чтобы все команды могли продуктивно работать в первом Спринте, реализуя элементы до тех пор, пока они не будут доведены до состояния готовности, а в конечном итоге будет создан готовый к поставке продукт. К числу других целей относятся следующие.

1. Формирование общей концепции продукта и ее уяснение.
2. Выработка предложений по нововведениям.
3. Первоначальная постановка главных целей.
4. Долгосрочное планирование.

Эти цели более или менее важны в зависимости от текущего состояния группы и продукта. Например, в устоявшейся на одном месте небольшой группе, работающей над зрелым продуктом, такие цели могут быть уже достигнуты. И совсем иначе дело обстоит с двухлетним продуктом, испытывающим бурный рост на “горячем” рынке, при наличии трех новых мест для его разработки.

Основная цель: уточнение элементов в достаточной степени

Достижению этой цели обычно уделяется большая часть времени при первоначальном Уточнении Задела Продукта. Для этого придется решить следующие вопросы.

- > **Как добиться цели?** Рекомендации относительно совместной работы всех команд приведены выше, в разделе “Наставление: многокомандное Уточнение Задела Продукта”. Не отступая от сути управления эмпирическим процессом, в LeSS не дается никаких предписаний по поводу уточнения элементов, хотя и рекомендуются такие распространенные методы, как *гибкое моделирование и спецификация по образцу*.
- > **Сколько потребуется элементов для подготовки к первому Спринту?** Как поясняется далее в разделе “Наставление: разделение”, команды следует нацеливать на элементы, мелкие до такой степени, чтобы одна команда могла реализовать около четырех элементов в одном Спринте. Так, если имеется пять команд, это означает, что при первоначальном Уточнении Задела Продукта следует подготовить 20 (4×5) элементов. Но ведь, как правило, для прояснения и подготовки к реализации первоначально неясных и неисследованных элементов в среднем требуются *два* Спринта. В таком случае при первоначальном Уточнении Задела Спринта придется подготовить около 40 элементов. Такого задела должно быть достаточно для первоначальных Спринтов, тогда как в первом Спринте команды приступают к уточнению элементов для последующих Спринтов.

Цель: формирование общей концепции продукта и ее уяснение

На каждом совещании по Уточнению Задела Продукта предоставляется возможность сформировать концепцию продукта и расширить общее представление о нем. Но вся группа может впервые сосредоточиться на этом лишь при первоначальном Уточнении Задела Продукта. Например, в прежней традиционной организации формирование концепции продукта, возможно, было уделом только ответственных за выпуск продукции, тогда как программисты, тестировщики и прочие просто должны были реализовывать то, что им спускалось в приказном порядке. Ничего подобного не должно быть в новой группе, внедряющей LeSS.

Как добиться цели? Прежде всего, для проведения подобных мероприятий требуется грамотный устроитель рабочих совещаний, роль которого просто неоценима! И хотя для этой цели подойдет любая методика, мы рекомендуем коллективные, занимательные и верные методики, описанные в литературе, в том числе в книгах *Innovation Games* и *Gamestorming*.

Цель: выработка предложений по нововведениям

Как и при формировании концепции продукта, на каждом совещании по Уточнению Задела Продукта предоставляется возможность высказывать свои соображения и вырабатывать предложения по поводу нововведений. Но первоначальное Уточнение Задела Продукта идеально служит первым шагом для того, чтобы привлечь всех к нововведениям.

Как добиться цели? И в этом случае подойдет любая методика, но для начала рекомендуется ознакомиться с методиками, описанными в упомянутых выше книгах *Innovation Games* и *Gamestorming*.

Цель: первоначальная постановка главных целей

Каждое совещание по Уточнению Задела Продукта служит удобным местом не только для предложения нововведений, но и для рассмотрения новых или альтернативных целей. Но самое время и место сделать это, а также опробовать новые методики на практике — именно при первоначальном Уточнении Задела Продукта.

Как добиться цели? Для уяснения и постановки целей обычно рекомендуются две методики: *оценка воздействия* и *построение карты историй*. И не удивительно, что изучению и применению этих методик посвящены рекомендуемые книги *Impact Mapping* и *User Story Mapping* соответственно. Применение одной из этих методик наглядно показано на рис. 11.2.



Рис. 11.2. Оценка воздействия при первоначальном Уточнении Задела Продукта

Цель: долгосрочное планирование

Прежде всего, для формального отсутствия в Scrum понятия “планирование выпуска” имеется веское основание. Дело в том, что главная цель совершенствования заключается в том, чтобы доставлять ценность хотя бы в каждом Спринте. Столь честолюбивая цель гибкой разработки дает немалые выгоды. Эта же цель преследуется и в LeSS. Благодаря этому отпадает потребность в планировании крупными партиями и эффективно проявляется гибкость реагирования на изменения.

Но, безусловно, возникают обстоятельства (чаще всего при крупномасштабной разработке), когда особое значение приобретает долгосрочное планирование. Как правило, оно требуется для согласования сроков с внутренними группами (например, рекламной кампании вместе с отделом маркетинга), заказчиками (например, развертывания новых радиомачт) или другими заинтересованными сторонами (например, таких

событий, как демонстрации). И самое удобное время и место приступить к долгосрочному планированию — именно при первоначальном Уточнении Задела Продукта.

При долгосрочном гибком планировании необходимо иметь в виду следующее важное обстоятельство.

Планировать объем работ над элементами на определенный срок иногда просто необходимо. Но планировать элементы на конкретные Сprintы не следует, поскольку это уничтожает гибкость.

Как добиться цели? Независимо от применяемой методики возникает вопрос: как часто следует выполнять долгосрочное планирование? Первоначальное Уточнение Задела Продукта служит первым удобным моментом для долгосрочного планирования, а каждое последующее совещание по Уточнению Задела Продукта и каждый Спринт могут и, вероятно, должны быть использованы для повторного долгосрочного планирования, если это так важно.

Главной особенностью долгосрочного планирования является согласование сроков, и для этой цели при первоначальном Уточнении Задела Продукта выполняется определенного рода оценивание. Более подробно этот вопрос рассматривается далее, в разделе “Наставление: оценивание в крупных масштабах”, а до тех пор можно указать на следующий важный момент:

Выбирать следует самую простую методику, которая соответствует поставленной цели и способствует обсуждению и обучению.

Еще одной важной особенностью долгосрочного планирования является *упорядочение* — зачастую целей или тем. Естественно, что для этого в LeSS не предписывается никакой методики, хотя и рекомендуются такие упоминавшиеся ранее методики, как оценка воздействия и построение карты историй.

Архитектурное проектирование и первоначальное Уточнение Задела Продукта

См. раздел “Наставление: многокомандное совещание по проектированию” главы 13.

Результаты первоначального Уточнения Задела Продукта могут подразумевать ряд архитектурных изменений, которые командам следует принять во внимание перед первым Спринтом. Обсуждается и решается ли этот вопрос при первоначальном Уточнении Задела Продукта? Нет, поскольку Уточнение Задела Продукта служит для уяснения и изучения продукта с точки зрения заказчиков, а не для обсуждения проектного задания. Вместо этого после первоначального Уточнения Задела Продукта организуется одно или несколько рабочих совещаний, на которых команды исследуют проектное задание.

Наставление: разделение

Крупномасштабная разработка является сферой очень больших требований, и поэтому нам часто приходится слышать следующее: “Наша группа вряд ли сможет выполнить стоящие перед нами требования в двухнедельных Спринтах. Их никак нельзя сделать более мелкими, ориентируясь в то же время на клиентов”. Поэтому мы приглашаем того, кто пояснит нам самый крупный элемент, который вроде бы невозможно разделить на мелкие, ориентированные на клиентов элементы, и вместе с ним мы пробуем разделить его, делая наброски на белой доске. Как правило, на это уходит около пяти минут. Следовательно, это не такое уж и трудное дело, хотя оно и требует другого подхода. Поучившись немного, и вы можете стать настоящим мастером разделения!

Ниже перечислены причины для разделения крупных элементов.

- > Ранняя доставка элементов, представляющих большую ценность или немалый риск, повышает выгоды, ускоряет ответную реакцию и снижает риски. С этим связано то обстоятельство, что мелкие элементы увеличивают прозрачность и усиливают контроль Владельца Продукта над тем, что действительно важно и что следует делать дальше.
- > “Вертикальное” разделение с ориентированием на клиентов помогает распределить и распараллелить ценную работу среди многих команд, которые по-прежнему выполняют все свои действия в соответствии с Критерием Готовности.
- > Элемент должен быть полностью реализован в течение Спринта, чтобы получать в каждом Спринте некоторый прирост готового к поставке продукта, уменьшая объем незавершенных работ.

В этом наставлении поясняется, как научиться разделять требования.

Как научиться

Учиться помогает рассмотрение подробно поясняемых примеров и рассказывающих историй. Поэтому в приведенном ниже примере повествуется настоящая история разделения элемента. Имеется немало других примеров, достойных обучения этому мастерству. Чтобы стать настоящим мастером разделения, воспользуйтесь следующими ресурсами.

- > В книге *Practices for Scaling Lean & Agile Development*, посвященной LeSS, имеется (на стр. 247) раздел “Try... Split Product Backlog Items” (Попробуйте разделить элементы из Задела Продукта), где приводится немало подробно поясняемых примеров.
- > В книге *Fifty Quick Ideas to Improve Your User Stories* имеется наставление на 30 страниц по обучению разделению элементов.
- > На веб-сайте Ричарда Лоуренса на данную тему имеются следующие страницы: *Patterns for Splitting User Stories* (Шаблоны для разделения

пользовательских историй; <http://agileforall.com/patterns-for-splitting-user-stories/>) и *How to Split a User Story* (Как разделять пользовательские истории”; <http://agileforall.com/resources/how-to-split-a-user-story/>).

Как разделять

Чтобы научиться разделять крупные элементы, очень важно уяснить сначала *перспективы разделения*, а затем пробовать их разделять. Некоторые из этих перспектив описываются в приведенной ниже таблице.

Прецедент использования	Основные потоки работ или прецеденты использования; прецеденты использования CRUD (Создание, Чтение, Обновление, Удаление)	Конфигурация	Различная конфигурация, например, тип операционной системы
Сценарий	Конкретная последовательность действий в прецеденте использования	Роль пользователя, личность	Нападающий, защитник, опытный пользователь, начинающий пользователь
Тип	Различные типы или виды деятельности, например, виды торговли	Формат данных	XML, разделяемый запятыми и т.д.
Внешняя интеграция	С несколькими внешними элементами, например, биржами	Информационная часть	Подмножество многих элементов данных, которое может быть полезным
Операция/ сообщение	Системная операция/ сообщение; например, HTTP GET , SWIFT MT304	Нефункциональные требования	Средняя или высокая пропускная способность, наличие или отсутствие восстановления и т.д.
Канал ввода-вывода	Канал ввода или вывода, например, графический пользовательский интерфейс или командная строка	Заглушка	Простая имитационная реализация какого-нибудь компонента

Приведенный ниже пример поможет вам научиться разделять элементы с помощью перспектив.

Пример разделения: обработка депозитарных операций на кенийском рынке

Этот пример взят из крупного продукта для торговли ценными бумагами, где имеется элемент “Обработка депозитарных операций на кенийском рынке”. Его контекст состоял в том, что раньше подобная работа выполнялась в наполовину ручном

режиме. Но поскольку объем операций на данном рынке вырос, то торговой группе потребовалось полностью автоматизировать данный процесс.

С этой целью было проведено рабочее совещание по Уточнению Задела Продукта, на котором присутствовали члены команд и некоторые практические пользователи, которые хорошо знали требования, поскольку они сами занимались обработкой депозитарных операций в наполовину ручном режиме.

Можно ли прекратить разделение? Конечно, можно, если элемент уже оценен как достаточно мелкий и на один Спринт для одной команды приходится четыре аналогичных элемента. В таком случае потребность в дальнейшем разделении отпадает. И по мере определения новых разделяемых элементов их необходимо оценивать, чтобы решить, требуется ли дальнейшее их разделение. В данном примере исходный элемент был оценен как очень крупный.

Перспектива: разделение по прецедентам использования. После общения с пользователями на совещании по Уточнению Задела Продукта стало ясно, что данное требование связано с обработкой операций. Как правило, такие требования разделяются по основным *прецедентам использования*², и в данном случае они были разделены на следующие прецеденты использования.

1. Расчеты по сделкам.
2. Обработка корпоративного действия (например, дробление акций) по неоплаченной сделке.
3. Прочие.

Пользователи хотели обсудить и выработать *все* прецеденты использования. Поэтому мы попросили их внимания и сказали: “Сделаем постепенное прояснение. Какой из этих прецедентов использования встречается чаще всего?” Они ответили: “Расчеты по сделкам производятся чаще всего и в первую очередь, поскольку это позволяет быстро сократить затраты и ошибки”. Следовательно, мы решили сосредоточиться сначала на расчетах по сделкам. Нас не интересовало выявление *всех* возможных прецедентов использования. Вместо этого мы набросали на белой доске три прецедента использования, чтобы наглядно показать следующие основные подразделяемые элементы, приведенные также на сопутствующем рисунке:

- > Расчеты по сделкам.
- > Обработка корпоративного действия по неоплаченной сделке.
- > Все остальное для обработки операций на кенийском рынке.



² Термин и модель “прецедентов использования” знакомы и употребляемы пользователями.

Это *частичное разделение*, где элемент “Все остальное” играет важную роль заполнителя при разделении. Он сокращает излишнюю обработку и объем незавершенных работ и позволяет группе сосредоточиться на разработке мелкими партиями. Этот крупный элемент будет присутствовать в Заделе Продукта в качестве заполнителя элементов, которые предстоит выявить в дальнейшем.

На чем следует сосредоточиться дальше? На направлении разделения, чтобы ответить на следующие вопросы: что заставило нас сосредоточиться на элементе “Расчеты по сделкам” и на каком направлении нужно сосредоточиться дальше? Ниже приведены некоторые рекомендации по выбору направления для дальнейшего разделения.

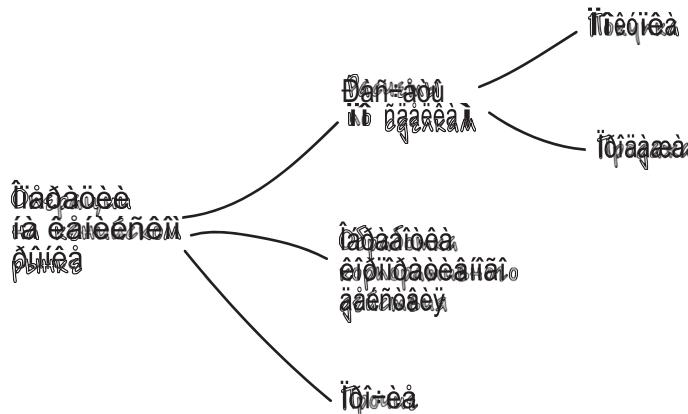
- > **Разделение по ценности или влиянию.** Например, увеличение доходов, доли на рынке или сокращение расходов.
- > **Разделение с целью изучить:**
 - > **предметную область** — например, неизвестные производные ценные бумаги;
 - > **технологию** — например, неизвестный протокол;
 - > **масштабы** — например, общий элемент.
- > **Разделение для снижения риска.** Например, разделение с целью прояснить и доставить элемент, предотвращающий штрафы, или же разделение для создания или оценки новой технологии.
- > **Разделение ради прогресса.** Иногда достаточно разработать хотя бы *что-нибудь*, чтобы приобрести уверенность, что требование может быть выполнено.

Можно ли прекратить разделение? Присутствующие на совещании непосредственно убедились, что рассматриваемый элемент все еще велик, и поэтому мы продолжили разделение.

Разделение по типу. Далее мы спросили: “Имеются ли другие типы оплачиваемых сделок?” Ответ был таков: “Да, имеются. Это покупки и продажи”. Такой ответ предполагает, что сделки можно дальше разделить на покупку или продажу ценных бумаг, как показано на сопутствующем рисунке. Но сначала нам пришлось задать еще один важный вопрос по поводу разделения.

Сокращает ли разделение трудозатраты? Если кому-то и удастся теоретически разделить требование на расчеты по покупкам и расчеты по продажам, то это еще не означает *разделение трудозатрат*, связанных с данным требованием. Иногда такие варианты разделения одинаково обрабатываются в одном и том же коде. И тогда подобное разделение не приносит никакой пользы, поскольку не сокращает трудозатраты. Поэтому мы спросили: “Являются ли логика, бизнес-правила, обработка и прочее одинаковыми для расчетов по покупкам и продажам?” Специалисты ответили: “О, нет, они совершенно разные”. Итак, мы выяснили, что разделение по типу приносит определенную пользу. И в конечном итоге у нас получилось разделение на следующие элементы.

- > Расчеты по сделкам.
- > Расчеты по покупкам.
- > Расчеты по продажам.
- > Обработка корпоративного действия по неоплаченной сделке.
- > Все остальное для обработки операций на кенийском рынке.



На чем следует сосредоточиться дальше? Рассматриваемое здесь требование автоматизировать обработку операций было обусловлено стремлением сократить затраты и ошибки при ручной обработке. Если затраты приравниваются к разным типам операций, то частота появления разных типов операций указывает, где можно извлечь наибольшую выгоду.

Далее мы спросили: “Какова доля в процентах, приходящаяся на сделки по покупкам?” Нам ответили: “80%”. Поэтому мы сосредоточились на элементе “Расчеты по покупкам”.

Можно ли прекратить разделение? Итак, мы спросили: “Являются ли покупки крупным элементом?”, на что нам ответили: “Нет, он все еще крупный”.

Наводящие вопросы и выявление разных вариантов. До сих пор мы пользовались своим опытом при обсуждении разделения, выясняя, какие именно существуют precedents использования, если таковые вообще имеются. Но не менее важно задать наводящие вопросы, поскольку опыт не всегда приводит к следующему умело сделанному шагу.

Задавая наводящие вопросы при обсуждении разделения, мы старались, в частности, выяснить наличие различных вариантов в рассматриваемом требовании. Дело в том, что поиск вариантов или альтернатив имеет *решающее значение* для выявления возможностей разделения и более глубокого понимания.

Итак, мы попросили разъяснить нам подробнее элемент “Расчеты по покупкам” и обнаружили два главных типа процессов расчетов: *бесплатно и поставка против платежа*. Следовательно, дальнейшее разделение оказалось следующим.

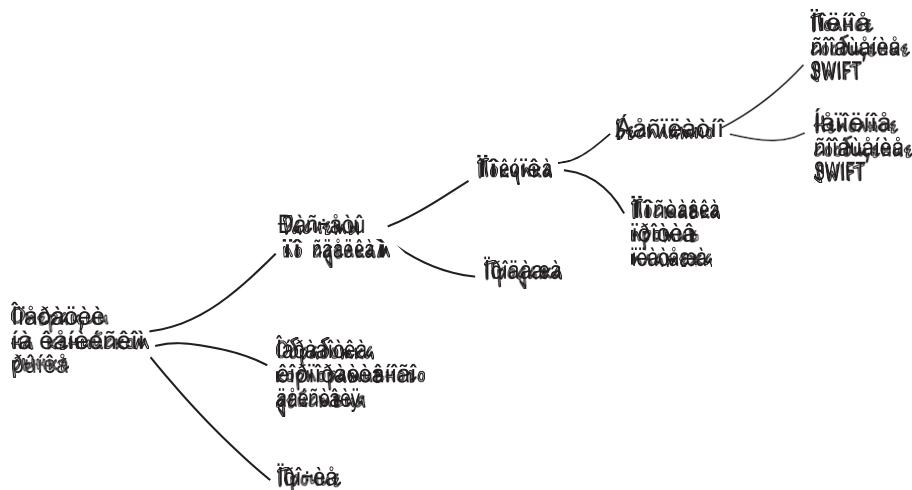
- > ...Расчеты по покупкам.
- > Расчеты по покупкам бесплатно.
- > Расчеты по покупкам с поставкой против платежа.

Затем мы выяснили, что элемент “Расчеты по покупкам с поставкой против платежа” предпочтительнее выпустить первым, поскольку такие операции выполняются довольно часто. Но этот элемент все еще оставался крупным.

И тогда нам пришлось задать дополнительные наводящие вопросы, попросив, в частности, дополнительно разъяснить элемент “Расчеты по покупкам с поставкой против платежа”. После этого мы обнаружили, что такой прецедент использования инициируется входящим сообщением SWIFT, и в зависимости от характеристик этого сообщения предпринимаются разные действия по обработке. А это подпадает под разделение по *типам* (или характеристикам) сообщений. В ходе последующего обсуждения мы обнаружили следующие подчиненные элементы, приведенные также на сопутствующем рисунке.

- > ...Расчеты по покупкам с поставкой против платежа.
- > Расчеты по покупкам бесплатно; все подробности о сторонах сделки уже имеются во входящем сообщении SWIFT (“полный вариант”).
- > Расчеты по покупкам бесплатно; некоторые подробности о сторонах сделки отсутствуют во входящем сообщении SWIFT (“неполный вариант”).

Для реализации последнего *неполного* варианта необходимо написать немало кода, чтобы извлечь и заполнить отсутствующие подробности о сторонах сделки. Но для реализации *полного* варианта этих сведений нужно было сделать не так уж и много. И тогда группа посчитала элемент “Расчеты по покупкам бесплатно; все подробности о сторонах сделки” достаточно мелким и не требующим дальнейшего разделения.



Так что же фактически зарегистрировано в Заделе Продукта? В нем зарегистрированы следующие элементы.

- > Расчеты по покупкам бесплатно; полное сообщение SWIFT.
- > Расчеты по покупкам бесплатно; неполное сообщение SWIFT.
- > Расчеты по покупкам с поставкой против платежа.
- > Расчеты по продажам.
- > Обработка корпоративного действия по неоплаченной сделке.
- > Все остальное для обработки операций на кенийском рынке.

Предки. Обратите внимание на то, что все промежуточные “предки” удалены из задела. И хотя это изящный и простой способ, иногда все же требуется сохранить какие-нибудь свидетельства о родительских элементах. В таком случае см. раздел “Наставление: обращение с родительскими элементами” главы 9. Итак, разделение можно пока что считать завершенным!

Разделение на тонкие комплексные элементы

А теперь рассмотрим обнаруженный ранее подчиненный элемент “Расчеты по покупкам бесплатно; полное сообщение SWIFT”. Он представляет собой полноценное комплексное, но *тонкое* функциональное средство, “вертикально” ориентированное на клиентов. К нему можно привязать лишь несколько приемочных тестов. И это наглядно показывает следующее крайне важное правило разделения.

Разделять следует на тонкие комплексные “вертикальные” требования.

**НЕ РАЗДЕЛЯЙТЕ ЭЛЕМЕНТЫ НА СТАДИИ
ВНУТРЕННЕГО ПРОЕКТИРОВАНИЯ!**

Каковы эти стадии? Разработчики привыкли разделять процесс разработки на отдельные стадии в соответствии с логическим алгоритмом внутреннего проектирования. Например, реализацию элемента “Расчеты по покупкам” можно разделить на следующие стадии.

1. Определение типа сообщения SWIFT.
2. Синтаксический анализ сообщения.
3. Извлечение сведений о сделке, связанной с сообщением, из базы данных.
4. Прочее...

Элементы не следует разделять на стадии обработки по алгоритму внутреннего проектирования. Например, для стадии “Определение типа сообщения SWIFT” не следует выделять отдельный элемент по следующим причинам.

- > В такой элемент нельзя ввести автоматизированные приемочные тесты, ориентированные на клиентов, поскольку не реализованы комплексные функциональные возможности, ориентированные на клиентов.
- > Такой элемент нельзя использовать на стадии эксплуатации, и поэтому он переходит в разряд незавершенных работ со следующими типичными недостатками: отсутствие полезной ценности и ответной реакции, а также скрытые дефекты и риски.
- > Такой элемент вносит динамику и затруднения, подобные тем, что возникают в командах для компонентов. Что здесь имеется в виду?

Нередко, особенно в архитектурах, создававшихся в то время, когда действовали команды для компонентов, одна стадия обработки оказывается связанной с каким-то одним программным компонентом. Например, стадия определения типа сообщения связана с компонентом идентификатора сообщения. В таком случае может произойти следующее.

Допустим, что каждая стадия обработки определяется в виде отдельного элемента (например, элемента “Определение типа сообщения SWIFT” и т.д.). И тогда проявляется тенденция определять и вносить изменения во все варианты пользовательских требований, включающих компонент, связанный с данной стадией. Например: “сделать всю работу в компоненте идентификатора сообщения, чтобы определить типы сообщений и больше не касаться этого вопроса”.

Но это возвращает нас обратно к динамике и затруднениям, возникающим в организации с командами для компонентов, даже если в ней, на первый взгляд, действуют команды для функциональных средств. Ведь на самом деле они работают над однокомпонентными задачами, скрывающимися под вывеской требований к стадии обработки.

С другой стороны, элемент “Расчеты по покупкам бесплатно; полное сообщение SWIFT” оказывается полноценным. Он очень тонкий и не допускает никаких других вариантов для расчетов по покупкам, кроме одного полноценного потока операций. Его можно интегрировать, поставлять, применять, а сам он может доставлять ценность и вызывать ответную реакцию. Кроме того, отпадает необходимость изменять автоматизированные приемочные тесты такого элемента.

Первое или последнее средство? Сначала неудачный исход

Иногда удобным местом для разделения, как ни странно, служат ошибочные (правильные) сценарии. Как-то нам пришлось участвовать в реализации стандарта HSDPA в продукте для телекоммуникационной сети 3G. Команда, разрабатывавшая этот продукт, приступила к разделению, пытаясь упростить сценарий удачного исхода. При этом они обсуждали следующий вариант.

Сделать вызов по стандарту HSDPA в как можно более простой конфигурации сети, пренебрегая всеми ошибочными случаями.

Но они обнаружили, что и такой вариант слишком крупный. Следовательно, вместо того чтобы рассматривать сценарии удачного исхода, они приступили к разделению, начиная со сценария неудачного исхода, а таких в телекоммуникационной сети встречается немало. Сначала они произвели разделение по самому простому сценарию неудачного исхода, а затем постепенно реализовали другие сценарии неудачного исхода. После двух Спринтов результатов, накопившихся после сценариев неудачного исхода, оказалось достаточно, чтобы приступить к работе над сценарием удачного исхода.

Почему такой подход принес пользу? Разделяя по случаям неудачного исхода, они постепенно создавали функциональные средства, постоянно рассматривая их с точки зрения клиентов. Кроме того, они рано начали устранять некоторые риски и повышать уровень обучения. Разумеется, наличие случаев неудачного исхода не доставляет практически никакой ценности.

Не меньше четырех элементов на Спринт

До какой степени мелкости следует разделять элементы? Естественно, элементы должны быть мельче Спринта, чтобы получить *прирост*, а делать элемент таким же крупным, как и целый Спринт, нежелательно. Дело в том, что один крупный элемент может быть не завершен полностью из-за большой изменчивости, присущей проектно-конструкторским работам. И тогда команда не сможет ничего выпустить в готовом виде. В итоге ничего полезного не будет доставлено, ответная реакция окажется слабой, а уровень обучения и степень адаптации понизится.

Таким образом, на степень мелкости разделения оказывает влияние следующая рекомендация: одна команда должна выбрать хотя бы четыре элемента на один Спринт. Ниже поясняется, почему четыре элемента являются золотой серединой между слишком крупными и слишком мелкими элементами.

- > При наличии крупных элементов увеличивается вероятность, что они не будут доведены до состояния готовности из-за изменчивости или доступности ограничивающих ресурсов (например, лабораторного оборудования).
- > Из приведенного выше пояснения следует такой вывод: если в конце Спринта накапливается много наполовину готовых (незавершенных) элементов, то у Владельца Продукта фактически окажется меньше вариантов выбора в следующем Спринте, поскольку завершение неготовых элементов почти всегда является обязательным.
- > Крупные элементы способствуют применению норм небрежной практики в водопадном стиле, заваливая команды множеством подробностей.

На первый взгляд, распределение по десять элементов на каждую команду при крупномасштабной разработке может показаться вполне допустимым, но такому распределению работ присущи следующие недостатки.

1. Издержки столь мелкого разделения.
2. Издержки ведения и уяснения крупного Задела Продукта, заполненного множеством мелких элементов.
3. Трудности сохранения комплексного и ориентированного на клиентов характера элементов.
4. Упрочение прежней тенденции завершать элемент в одиночку, а не всей командой с общей ответственностью.

Наставление: оценивание в крупных масштабах

В крупных группах к вопросам оценивания относится простая проблема синхронизации единиц оценивания среди команд и более губительная проблема несоответствия назначения оценок затрачиваемым усилиям и применяемым методикам.

Согласование трудозатрат и назначения оценок

В крупных традиционных группах приверженцы Тейлора и его теории научной организации труда переносят нормы передовой практики в группы, пренебрегая контекстом и допуская, что точные оценки всегда оправданы, несмотря на затраты и недостатки.

Оценки совсем не обязательно должны быть точными. Они должны быть полезными, а польза зависит от их назначения.

Например, наиболее частой причиной попыток улучшить точность оценивания (свести отклонение к конкретным затратам труда) служит стремление увеличить прогнозируемость. Но если рассматривать прогнозируемость с точки зрения гибкой методики разработки, то при 100%-ной точности оценок, которой достичь невозможно, прогнозируемость нельзя гарантировать из-за неизбежного появления новых элементов. Мы иногда напоминаем своим клиентам, что без изменений бывают только те продукты, у которых нет потребителей. Следовательно, мы должны *реагировать* на изменения, а не *следовать* намеченному плану.

Так как же оценивать? Ниже даются некоторые рекомендации по этому поводу.

- > **Приоритетность окупаемости инвестиций.** Если требуется получить большую отдачу от вложенных средств, следует оценивать как *отдачу*, так и *вложенные средства*.
- > **Согласование сроков:**
 - > с *внутренними группами* — например, для развертывания рекламной кампании с отделом маркетинга;
 - > с *заказчиками* — например, при поставке нового оборудования оператору телекоммуникационной сети необходимо организовать проект развертывания;
 - > с *событиями* — например, для проведения торговой выставки.

- > **Оценивание рисков в связи с обещаниями выпуска.** В худшем случае, когда обещания выпуска в срок при жестко заданном объеме работ все же делаются, оценка (и переоценка) помогает определить риски и потребность в адаптации. Вариант больших обещаний выпуска служит признаком безрассудства на рынке проектов, выполняемых по внешнему подряду с ограниченными или жестко фиксированными затратами и особенно — жестко заданным объемом работ. Оценки позволяют определить ощутимую прибыль и оценить регулярные риски для прибыли и поставки.
- > **Обучение путем исследования или выявления отличий.** Совместное оценивание позволяет лучше изучить элементы, включая сосредоточение основного внимания на дальнейшем прояснении или разделении. И если люди не согласны с оценками, то им нужно больше учиться. Однако в данном случае речь идет об *оценивании*, а не об *оценках*, что полезнее.

См. раздел “Наставление: сотрудничество с клиентами превыше всего” главы 8.

Относительные трудозатраты конкретных методик

Управление эмпирическим процессом является движущей силой LeSS, и поэтому никакой конкретной методики оценивания в LeSS не предписано. Для этой цели можно пользоваться любыми методиками: от покера планирования до параметрических моделей. Но еще важнее следующее.

Выбирать следует самую простую методику, которая отвечает поставленной цели и способствует обсуждению и обучению.

Согласование относительных единиц очкового оценивания среди команд

Естественно, что в LeSS не указано, какими единицами оценивания следует пользоваться, но в то же время весьма распространены такие относительные единицы оценивания, как очки за историю. А зачем пользоваться ими в крупномасштабном контексте? Объясняется это, в частности, тем, что такие единицы оценивания формируются относительно быстро и просто, обнаруживая в то же время отличия и возможности обучения. И все это увеличивает вероятность *обновления оценок*, что крайне редко делается в крупных группах со многими элементами, если единица или методика оценивания оказывается обременительной. А зачем вообще обновлять оценки? Это важный стимул для дальнейшего обучения и управления эмпирическим процессом создания продукта и выполнения процессов.

А почему бы не воспользоваться относительными единицами оценивания? Используя такие абсолютные единицы оценивания, как человеко-дни, можно избежать согласования исследуемой далее проблемы. А другая единица оценивания может быть

широко усвоена или использована, требуя меньше переобучения или преобразования. Кроме того, некоторые группы злоупотребляют относительными оценками и искажают их (например, привязывая их к человеко-дням, личным показателям и премиям), после чего они становятся бесполезными и неэффективными.

Но когда очки применяются в крупномасштабном контексте, возникает следующее затруднение: они относительны, например, оценка в 5 очков не имеет абсолютно независимого значения. В двух командах такая оценка может трактоваться по-разному. Если же оценки используются для принятия решений или оценивания достигнутого прогресса, то возникают трудности в связи с их несогласованностью. Но если группа пришла к общему согласию относительно величины очковых оценок, то она получит следующие преимущества.

- > Согласованная оценка масштабов, помогающая назначить приоритеты для окупаемости инвестиций и повышающая гибкость распределения элементов среди команд.
- > Скорость работы на уровне продукта, помогающая прогнозированию.

Как же выполняется согласование оценок? Это можно сделать следующим образом.

- > **Произвести калибровку по готовым элементам.** В частности, команды могут производить калибровку или сравнение по уже готовым элементам в Заделе Продукта. И таких элементов должно быть довольно много, чтобы они были известны многим людям. Тем самым гарантируется эффективность такого способа.
- > **Согласовать оценки при многокомандном или общем Уточнении Задела Продукта.** Когда две команды или более совместно уточняют Задел Продукта, они согласуют общее смысловое значение относительных очковых оценок. Аналогично оценки согласуются, когда оценивание производится при общем Уточнении Задела Продукта представителями всех команд.

LeSS HuGe

Уточнение Задела Продукта проводится в LeSS HuGe по каждой Области Требований таким же образом, как и в более мелкой инфраструктуре LeSS. Так, совещание по общему Уточнению Задела Продукта проводится лишь в одной Области Требований, а не для всего продукта. Для LeSS Huge не предусмотрено никаких особых правил, касающихся Уточнения Задела Продукта.



Первое Планирование Спринта в LeSS

Планирование Спринта

Единственным местом, где работа и движение означают одно и то же, является зоопарк, куда люди приходят и платят за то, чтобы посмотреть на животных в движении.
— Тайти Оно

Scrum для одной команды

Планирование Спринта охватывает два вопроса, которые сводятся к тому, что и как нужно делать. Первый вопрос сосредоточен на выборе элементов и обсуждении неразрешенных проблем, которое должно быть кратким, поскольку элементы были прояснены ранее при уточнении Задела Продукта. А второй вопрос сосредоточен на первоначальном замысле и плане работы для получения готовых элементов. Рассматриваемые элементы и задачи составляют Задел Спринта. И хотя Владелец Продукта определяет порядок следования элементов, только Команда решает, сколько элементов нужно выбрать для Спринта. Выбранные элементы составляют не обязательство или обещание выполнить определенный объем работ, а прогноз, который Команда считает реалистичным.

Планирование Спринта в LeSS

При масштабировании соблюдаются следующие принципы, связанные с Планированием Спринта.

- > **Сосредоточение внимания на продукте в целом.** При наличии многих команд увеличивается вероятность того, что каждая из них пойдет своим путем и не будет работать вместе с другими. И эта тенденция усиливается или ослабляется в зависимости от того, как выполнено планирование.
- > **Управление эмпирическим процессом и непрерывное совершенствование.** Для крупномасштабной разработки характерны различные контексты и потребности в улучшении, поэтому особенно важно, чтобы проведение

совещаний по Планированию Спринтов в LeSS было оставлено на усмотрение команд.

- > **Добиваться большего меньшими средствами.** Традиционный подход к планированию при крупномасштабной разработке отличается большой сложностью и необходимостью управлять многими зависимостями. А в LeSS планирование упрощается благодаря тому, что команды для функциональных средств сами координируют свои действия.

Правила LeSS

Планирование Спринта делится на две части: Первое Планирование Спринта, являющееся общим для всех команд, а также Второе Планирование Спринта, обычно проводимое отдельно для каждой команды. Многокомандное Второе Планирование Спринта следует проводить в общем пространстве для тесно связанных элементов.

Совещание по Первому Планированию Спринта посещают Владелец Продукта, Команды или их представители. Совместно они предварительно выбирают те элементы, над которыми предстоит работать каждой команде в следующем Спринте. Команды находят возможности для совместной работы и окончательно проясняют насущные вопросы.

У каждой Команды имеется свой Задел Спринта.

Второе Планирование Спринта проводится для того, чтобы Команды решили, как они будут реализовывать выбранные элементы. Как правило, такое планирование включает в себя замысел и создание их Заделов Спринтов.

Наставление: Первое Планирование Спринта

См. раздел “Наставление: общее Уточнение Задела Продукта” главы 11.

Что же еще происходит при Первом Планировании Спринта в LeSS, помимо сосредоточения внимания на главном вопросе: *что делать?* Прежде чем ответить на этот вопрос, необходимо напомнить приведенную ранее рекомендацию в отношении Уточнения Задела Продукта: группировать команды по ряду элементов. Это способствует улучшению общего понимания и повышению степени гибкости команд, а также подразумевает, что при Первом Планировании Спринта еще не вполне очевидно или разграничено, что конкретно будет делать каждая Команда.

Итак, Команды и Владелец Продукта должны решить, как распределить между собой элементы. Кроме того, Команды должны найти возможности для совместной работы и обсудить, как это сделать. А поскольку работать со многими командами и

элементами сложно, то при Первом Планировании Спринта Владелец Продукта и Команды обсуждают и непосредственно адаптируют свои решения относительно приоритетов и распределения элементов.

Что касается продолжительности, то Первое и Второе Планирование двухнедельного Спринта длится максимум два часа. А для Спринтов другой продолжительности длительность планирования изменяется пропорционально.

Кто участвует в Планировании Спринта

К Первому Планированию Спринта почти не должно остаться никаких вопросов о предстоящих элементах, поскольку они были предварительно прояснены при уточнении Задела Продукта. Кто же тогда должен участвовать в Первом Планировании Спринта? Только Владелец Продукта и Команды или их представители. Но при Первом Планировании Спринта нередко возникают мелкие вопросы, оставшиеся неразрешенными из-за пробелов в знаниях малоопытных команд, особенно на ранней стадии внедрения LeSS. В таком случае следует пригласить других специалистов, способных оказать помощь в разрешении на месте мелких вопросов (например, ответственных за выпуск продукции, потребителей или пользователей и т.д.). Преследуемая цель — улучшить положение дел таким образом, чтобы эта временная мера не потребовалась.

Сколько же членов каждой Команды должно участвовать в Планировании Спринта? Их количество колеблется от одного представителя до Команды в полном составе. Следует, однако, иметь в виду, что Scrum-мастер не является членом Команды и тем более ее представителем. Если предполагается выделить лишь несколько представителей, то их количество должно соответствовать числу проблем, которые могут возникнуть при передаче работ. Следует также принять во внимание размеры помещения, где необходимо создать ощущение общности. Представителей команды нужно периодически сменять по очереди. В данном мероприятии должен принять участие хотя бы один Scrum-мастер, чтобы давать инструкции по проведению Планирования Спринта в LeSS и оказывать помощь в его улучшении.

Выбор элементов

Требуется ли предварительное распределение? Должна ли группа принимать решение о распределении элементов среди команд до или во время Первого Планирования Спринта? Вероятно, во время данного мероприятия, поскольку принять решение как можно позже означает отложить его до тех пор, пока не накопится достаточно информации, чтобы сделать осознанный выбор. Это не только повышает степень организационной гибкости, открывая больше возможностей, но и способствует сосредоточенности на продукте в целом, поскольку командам требуется более широкое представление.

Что решает Владелец Продукта? Должен ли Владелец Продукта сам принять решение относительно распределения элементов среди команд? Вероятно, нет.

Такое решение лучше отдать на усмотрение команд, поскольку это снижает нагрузку на Владельца Продукта, поддерживает самоорганизацию, повышает степень гибкости и уровень обучения, когда команда выбирает менее известные ей элементы, а также вызывает усиление у команд чувства владения продуктом. Свобода в принятии

самостоятельных решений и ее поощрение усиливает посылы к самоуправлению, доверию вместо микроуправления, принятию решений, опираясь на знания, а также к необходимости дорожить обучением, особенно во вновь сформированных командах.

Нужно ли соперничать за элемент? Что, если разные команды соперничают за интересующий их элемент? В этом нет ничего страшного, поскольку команды проявляют заинтересованность. В таком случае опытный Scrum-мастер предлагает командам разные способы принятия решений: от рукоборья до развязывания спорного узла Владельцем Продукта. В конечном счете Владелец Продукта может сам решить, какой команде работать над конкретными элементами, исходя из контекста некоторых критических или рискованных вариантов выбора элементов. И, конечно, Владелец Продукта не может решать вместо команд, сколько элементов им следует выбрать. Таким образом, если у него возникает потребность нацелить команды на отдельные элементы, это, вероятно, свидетельствует о более глубоких затруднениях.

Пример Первого Планирования Спринта

Рассмотрим пример проведения Первого Планирования Спринта, обратив внимание на применяемые методики и назначение данного мероприятия. Чтобы провести такое Планирование Спринта, достаточно выполнить следующие действия.

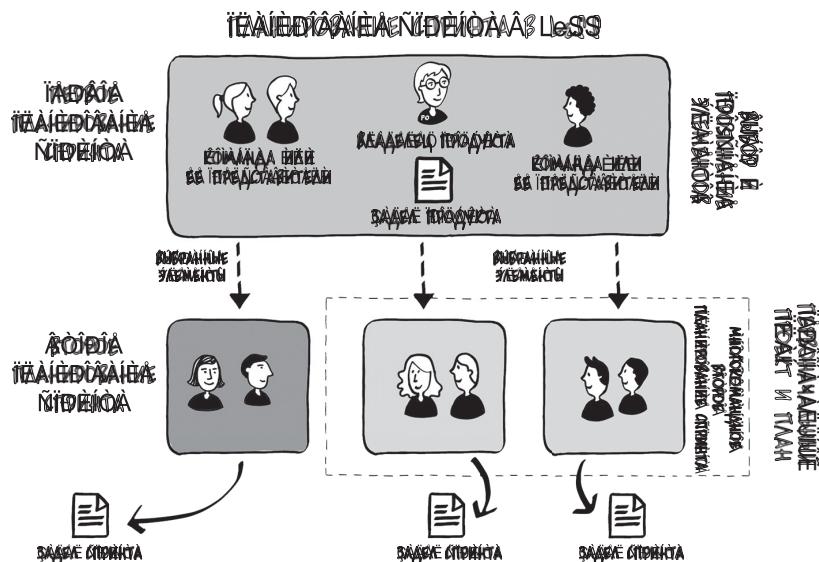
- 1. Положить карточки на стол.** Владелец Продукта пользуется карточками для оценивания предстоящих элементов, раскладывая их на столе в том порядке, в каком они расположены в Заделе Продукта. Члены команд обсуждают, решают, выбирают и, вероятно, меняют элементы местами.
- 2. Распределить высокоприоритетные элементы?** Допустим, команда А выбирает элементы, расположенные в порядке [1, 2, 3, 4], а команда В — в порядке [5, 6, 7, 8]. Во время Спринта команда А пропускает элемент 4 (не важно, по какой причине). В итоге высокоприоритетный элемент 4 оказывается неготовым, хотя его могла бы, вероятно, реализовать команда В. И если это серьезное затруднение, то Владелец Продукта должен попытаться распределить высокоприоритетные элементы среди команд. Хотя это не самое лучшее решение, поскольку оно может противоречить намерениям команды, желающей выбрать элементы, тесно связанные с данным.
- 3. Разойтись, чтобы прояснить?** В идеальном случае элементы подготовлены к реализации без накопившихся вопросов. Но иногда такие вопросы остаются. Если имеется только две команды, то они должны обсудить эти вопросы вместе. А если имеется семь команд, то обсуждение накопившихся вопросов может затянуться надолго. В таком случае команды могут разойтись, чтобы члены каждой команды перешли в другое место помещения и записали вопросы, которые требуется прояснить. К ним на помощь приходят Владелец Продукта, члены других команд (особенно те, что участвуют в многокомандном Уточнении Задела Продукта) и прочие заинтересованные стороны. Ответы на проясненные вопросы следует записать, чтобы их могли в дальнейшем прочитать отсутствующие члены команд.

4. Найти возможности для совместной работы. Команды выполняют общую работу, имеют общий код, им нужно создать один интегрированный продукт, а иногда работать над сильно связанными элементами. В связи с этим Первое и Второе Планирование Спринта служит удобным моментом для обсуждения и нахождения возможностей совместной работы и координирования действий. Поэтому предпочтение следует отдавать многокомандному Второму Планированию, которое рассматривается в следующем наставлении. А в качестве его дополнения или альтернативы можно предложить совместное обсуждение ближе к концу Первого Планирования Спринта.

Многоместное Планирование Спринта. Такое мероприятие проводится в режиме видеоконференции в виртуальном общем пространстве, где элементы предлагаются к рассмотрению. Если возникают какие-нибудь вопросы, их проще всего обсудить вместе. А если в данном мероприятии принимают участие многие команды, которым нужно обсудить немало вопросов, можно опробовать методику расхождения, когда для интерактивной переписки выделяется по одному окну на каждый обсуждаемый элемент.

Требуется ли согласование по окончании Планирования Спринта? После совещаний по Первому и Второму Планированию Спринта некоторые группы предпочитают проводить короткие собрания по согласованию с представителями всех команд, чтобы обучиться и адаптироваться к новым вопросам. Такая потребность возникает, например, в том случае, если команда пропустила какой-нибудь элемент во время Второго Планирования Спринта.

На приведенном ниже рисунке наглядно показана схема Планирования Спринта в LeSS.



Наставление: многокомандное Второе Планирование Спринта

В простейшем случае Второе Планирование Спринта проводится отдельно и более или менее параллельно в каждой команде с участием всех ее членов. На этом мероприятии обсуждаются вопросы проектирования и составляется план, воплощаемый в Заделе Спринта. Для планирования двухнедельного Спринта требуется не больше двух часов.

См. раздел “Наставление: многокомандное Уточнение Задела Продукта” главы 11.

Но зачастую более предпочтительным оказывается многокомандное Второе Планирование Спринта с участием двух и более команд в одном помещении. Это может быть в интересах той же самой группы, которая проводила многокомандное Уточнение Задела Продукта по связанным вместе элементам.

В общем, многокомандное Второе Планирование Спринта следует проводить при наличии элементов, сильно связанных вместе по требованиям и проектным решениям. И совместными усилиями можно улучшить обсуждение, проектное решение, координация действий и порядок выполнения общей работы.

Пример Второго Планирования Спринта

Рассмотрим пример проведения Второго Планирования Спринта. Чтобы провести такое Планирование Спринта, достаточно выполнить следующие действия.

- 1. Ставить вопросы и находить на них ответы всей группой.** Это делается с целью рассеять туман и наметить определенные пути.
- 2. Обсудить всей группой проект и общую работу в режиме совещания.** Общие проекты обсуждаются и намечаются совместно или в режиме совещаний, проводимых по образцу схождения-расхождения. При этом выявляются общие задачи для многих команд и принимается решение об их сотрудничестве. Следует, однако, иметь в виду, что на Второе Планирование Спринта отводится не более двух часов, поэтому требуется краткость, чтобы уложиться в этот срок.
- 3. Разойтись для обсуждения проекта и планирования в одной команде.** Команды перемещаются по разным участкам помещения, проводя свое Второе Планирование Спринта. Свои действия они постоянно координируют с другими командами современной разновидностью метода “поговорить”, которая в LeSS называется “прокричать”. На этой стадии проводится большая часть Второго Планирования Спринта.
- 4. Снова сойтись, когда требуется.** Это необходимо для решения вопросов, касающихся всех команд.

См. раздел “Наставление: поговорить” главы 13.

Возможности для общей работы в сравнении с зависимостями между командами

Что такое общая работа? Допустим, что у двух или более команд имеются элементы, требующие решения какой-нибудь общей задачи. Это и есть общая или разделяемая работа.

Группа с традиционным мировоззрением мыслит категориями зависимостей между командами и управления этими зависимостями. А в LeSS гордиев узел такого взгляда на данную проблему и ее разрешение разрубается следующим образом.

Между командами не возникает никаких зависимостей, а только имеются возможности для выполнения общей работы.

Как же выполняется совместная работа? Допустим, что в ходе многокомандного Второго Планирования Спринта команды А и В обнаруживают, что у них имеется общая задача Х. В процессе обсуждения выясняется, что команда А планирует взяться за решение этой задачи, и тогда она просто вводится в Задел Спринта данной команды. Если же при последующем проведении Спринта обнаружится, что задача Х нужна в первую очередь команде В, эта команда сама решит ее и затем известит об этом команду А, проявив простую гибкость, чтобы адаптироваться к сложившейся ситуации.

Группе, самоорганизующейся из команд для функциональных средств, нетрудно выявить или найти возможности для общей работы при многокомандном Втором Планировании Спринта. Но это требует коренных перемен в умонастроении и практике тех групп, которые привыкли иметь дело с зависимостями между командами, возникающими вследствие строгого владения задачами. И это характерно для индивидуального владения кодом, команд для компонентов, задач, поручаемых руководителями проектов, или отдельной команды, специально предназначеннной для управления зависимостями и координирования действий.

Наставление: ведение Задела Спринта без программных средств

Задели Продукта и Спринта ведутся раздельно и служат разным целям. В частности, Задел Продукта предназначен для ведения элементов, тогда как Команда ведет Задел Спринта самостоятельно, не прибегая к внешней помощи или услугам Владельца Продукта. Как подчеркивается в *Руководстве по Scrum*, Задел Спринта принадлежит *только* Команде. Поэтому Команда должна быть в состоянии выбирать подходящее инструментальное средство для ведения Задела Спринта, сменяя его по мере необходимости. Одним и тем же инструментальным средством *нельзя* пользоваться для ведения Задела Продукта и Задела Спринта.

См. раздел “Наставление: инструментальные средства для ведения крупных заделов продуктов” главы 9.

Если для ведения Задела Продукта вполне возможно применение цифрового инструментального средства, то для ведения Заделов Спринтов рекомендуется следующее.

Не пользоваться никакими программными средствами, а только физическими средствами наглядного управления, например, карточками на стене.

Ниже перечислены веские на то основания.

- > **Усиление взаимодействия команд с информацией.** Если внимательно присмотреться к поведению команд, как мы это делали неоднократно, сравнивая и противопоставляя команды, пользующиеся для ведения Задела Спринта карточками на стене и программными средствами, то нетрудно заметить существенное отличие во взаимодействии и сотрудничестве команды, действующей как коллектив, а не как группа отдельных личностей. Карточки на стене стимулируют команды, а карточки на экране компьютерного монитора — отдельные личности. Кроме того, простота употребления и замены, а также наглядность представления информации на карточках способствует тому, что команды активно и охотно взаимодействуют с информацией в их Заделе Спринта, представленном на стене. Совсем другая ситуация наблюдается на практике, когда команды пользуются программными средствами для ведения Задела Спринта.
- > **Усиление взаимодействия при Втором Планировании Спринта.** Применение только карточек на стене подразумевает, что при Втором Планировании Спринта компьютеры вообще не используются, что и рекомендуется делать. Нам не раз приходилось наблюдать, как компьютеры уничтожают атмосферу сотрудничества при Втором Планировании Спринта.
- > **Предотвращение слежения и микроуправления.** Любопытно понаблюдать за происходящим, когда команды вводят информацию о Заделе Спринта в программное средство. В частности, у руководителей возникает искушение вернуться к старой привычке следить за действиями команд, сравнивать их достижения и прибегать к микроуправлению. И даже Владелец Продукта может прибегнуть к микроуправлению Заделами Спринтов. Такое отступление от нормальной деятельности приходилось наблюдать нам (и, скорее всего, многим из вас) почти во всех случаях, когда применялись программные средства.

LeSS HuGe

Планирование Спринта проводится по Областям Требований. Никаких особых правил для этого в LeSS HuGe не предусмотрено.

Наставление: совещание Команды Владельцев Продукта

Каждый Владелец Продукта в Области относительно независим в принятии решений, и поэтому существует риск потерять сосредоточенность на всем продукте в целом или согласованность при выборе тем и элементов в отдельных областях. В качестве меры противодействия этому явлению рекомендуется провести перед следующим Спринтом совещание Команды Владельцев Продукта. На этом совещании Владельцы Продукта в Областях делятся сложившимся у них положением дел и предстоящими целями, а также обсуждают возможности для согласования своих действий. А общий Владелец Продукта может дать руководящее указание.

На этом совещании можно также обсудить результаты предыдущих совещаний по Подведению Итогов Спринта в каждой Области Требований как предпосылки для планирования. К участию в таком совещании можно привлечь представителей команд для дальнейшего обучения и получения ответной реакции и хотя бы одного Scrum-мастера для оказания помощи в критическом анализе и улучшении сложившегося положения дел.



Совещание в Открытом Пространстве для координации действий в LeSS

КООрдинирОвание дейСтвий и интеграция

Сначала запишите задачу, затем хорошенько обдумайте ее и запишите решение.

*— Алгоритм Фейнмана,
описанный Марри Гелл-Маном*

Scrum для одной команды

В непринужденном и постоянном взаимодействии члены команды стремятся к своей общей цели. Непрерывная интеграция всей работы в виде продукта постепенно нарастает в течение Спринта. Это существенные характеристики координации действий и интеграции в хорошо организованной инфраструктуре Scrum для одной команды. О хорошей команде можно судить по постоянной молве, порождающейся сотрудничеством. Только и слышно, что об их интеграции!

Каким образом координация действий и интеграция поддерживаются в Scrum? Решающую роль в этом играют следующие основные элементы: самоуправление, общая ответственность, общая цель и управление эмпирическим процессом. Этому также способствуют такие вспомогательные нормы практики, как Планирование Спринта, ведение Задела Спринта и Ежедневная Летучка. Они помогают создать мнение об отличной команде.

Как же добиться непринужденного самоорганизующегося координации действий и интеграции во *многих* командах? В этом-то и состоит главная трудность внедрения LeSS.

Координация действий и интеграция в LeSS

А почему речь идет о *координации действий и интеграции*? А потому что при непрерывной интеграции пути координации

действий и интеграции заметно пересекаются. В частности, для интеграции требуется координирование действий, а для координирования действий — интеграция.

При масштабировании соблюдаются следующие принципы, связанные с координированием действий и интеграцией.

- > **Крупномасштабная инфраструктура SCRUM — это тоже SCRUM.** В однокомандной группе продукта Команда сама координирует свои внутренние действия. А при наличии многих команд обязанности по внутреннему координированию действий распределяются среди команд, поскольку все они разделяют общую цель: создать прирост готового к поставке продукта. Но большинство команд не знают, как координировать и объединять свои действия с другими командами. Им известно только, как это делать в самой команде, и раньше крупное координирование действий осуществлялось отдельной управленческой группой.
- > **Системный подход и сосредоточение внимания на всем продукте.** Традиционно разобщенные команды не отвечают ни за весь продукт, ни за координирование действий по его поставке, ни за представление о работе системы как единого целого. Командам, только начинающим практиковать LeSS, будет трудно видеть целое.
- > **Управление эмпирическим процессом и непрерывное улучшение до полного совершенства.** Методы координирования действий должны особенно подбираться и приспосабливаться к конкретной ситуации в крупных группах, потому что контексты организации и продукта довольно сложны и переменчивы. А для того чтобы усилить ощущение владения и заинтересованности команд в методах координирования действий, командам нужно определить и уточнить их.

Правила LeSS

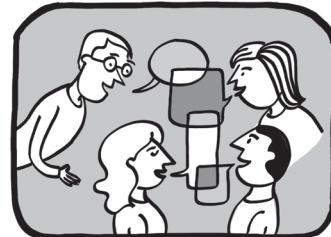
Решение о координировании взаимных действий принимают сами команды. Предпочтение следует отдавать децентрализованному и неформальному координированию действий над централизованным.

Каждая Команда проводит свою Ежедневную Летучку.

Наставление: поговорить

Работая годами с крупными группами и наблюдая массу методов координирования действий многих команд, мы обнаружили один самый действенный метод. Он предполагает выполнение следующих действий.

1. Вы осознаете необходимость скоординировать свои действия с другой командой.
2. Встаньте и пойдите к другой команде.
3. Приветствуя их, скажите: “Нам нужно поговорить”.



Такой метод называется “поговорить”. На первый взгляд это похоже на глупую шутку, но именно это мы имели в виду. Ведь нам не раз приходилось наблюдать следующий образец поведения: чем более формальными оказываются методы координирования, тем меньше координирования происходит, поскольку людям кажется очень важным следовать поциальному пути координирования действий. Например, вы осознаете, что возникает вопрос координирования действий, а затем вспоминаете о предстоящем завтра совещании в режиме Схватки над Схватками. Поэтому вы решаете, что лучше подождать до завтра и поставить данный вопрос на совещании, чем поднимать его теперь.

Если принять, что самый лучший путь для координирования действий команд — *просто поговорить*, то возникает потребность пересмотреть вопрос координирования. Ведь вопрос состоит не в том, какие методы координирования действий следует применить, а в том, как командам узнать, что им нужно скоординировать свои действия и поговорить. Как же им обнаружить, что поговорить нужно именно *теперь*, и с *кем* следует вести разговор?

Трудность крупномасштабного координирования действий состоит не в самом методе координирования, а в осознании потребности скоординировать свои действия и найти того, с кем можно об этом поговорить.

В каждом из приведенных далее наставлений рассматривается эксперимент по координированию действий, преследующий главную и конкретную цель и имеющий в качестве побочного эффекта создание сети неформальных координационных связей и обмена информацией. Именно эта сеть неформального обмена информацией позволяет членам команд осознать, что им нужно скоординировать свои действия и просто поговорить.

Например, роль спутника отводится работнику, переходящему в каждом Спринте из одной команды в другую, чтобы быть именно в том месте, где он нужен больше всего. Его основная функция — делиться и обучать конкретным знаниям, а в качестве

побочного эффекта он переносит нормы практики из одной команды в другую и создает неформальные связи между ними.

Ниже перечислены основные рекомендации относительно координации действий.

- > Поговорить.
- > Поддерживать связь через код.
- > Общаться в сообществах.
- > Проводить межкомандные совещания.
- > Общаться в открытом пространстве.
- > Привлекать наставников по компонентам.
- > Назначать спутников.
- > Назначать разведчиков.
- > Ориентироваться на ведущие команды.

Прежде чем выяснять, как поддерживать связь через код, рассмотрим основание, которое следует заложить для организации среды, удобной для координации действий.

Наставление: удобная среда для координации действий

Насколько широко и неформально разрешить перетекание информации, чтобы команды осознали потребность в координации действий? Какая удобная среда требуется для координации действий? Ниже перечислены особенности такой среды.

- > Координация действий принадлежит командам.
- > Предпочтение отдается децентрализованному и неформальному координации.
- > Команды для функциональных средств преследуют общую цель.
- > Основное внимание сосредоточено на продукте в целом.
- > Удобные физические и технические условия.

Координация действий принадлежит командам

Координация действий и интеграция в LeSS принадлежит командам, а не отдельной группе вроде команды управления проектами или интеграционной команды. Дело в том, что каждая команда отвечает за координацию своих действий с другими командами, чтобы обеспечить единый интегрированный прирост продукта, который можно поставить хотя бы один раз за Спринт. Это важно по следующим причинам.

- > Ответственность за принимаемые решения и предпринимаемые действия в отношении координирования действий и интеграции возлагается на тех, кто выполняет практическую работу.
- > Командам, владеющим процессами, оказывается поддержка, способствующая их совершенствованию.
- > Сокращаются задержки и передачи работ.
- > Уменьшается организационная сложность, поскольку не требуются специальные роли.

Предпочтение отдается децентрализованному и неформальному координированию

Централизованные методы координирования действий представляют собой запланированные совещания с представителями всех команд, например, Схватки над Схватками, собрания коллектива сотрудников организации или совещания по состоянию проекта, которые нередко проводятся под вывеской гибкой методики, хотя ничем не отличаются от традиционных методов. Каковы недостатки таких методов? Они усиливают узкие места в обмене информацией, задержку, передачу работ, отчужденное поведение вроде “это не мое дело”. А кроме того, могут быть невероятно скучными и надоедливыми!

Децентрализованные методы координирования действий не требуют общих сознаний или групп и охватывают сети взаимодействующих людей, в том числе команды, работающие и общающиеся в общем пространстве, или многоместные команды, ведущие обсуждение в режиме интерактивной переписки. Такие методы исключают появление узких мест, задержек и передач работ. Но им присущи также некоторые недостатки. В частности, труднее получить общее представление, чтобы выявить оставленные без внимания вопросы, а также недостаточно широкий и постоянный обмен информацией обо всей системе.

В пользу децентрализованных методов свидетельствует то обстоятельство, что они подкрепляют более *непредвиденное поведение* для координирования действий. И в LeSS такая *непредвиденность* поощряется потому, что в крупных системах централизованные и укоренившиеся методы координирования действий могут препятствовать, во-первых, управлению эмпирическими процессами и непрерывному совершенствованию, а во-вторых, появлению у команд чувства владения этими процессами.

Но здесь нет места для ложной дилеммы. Обе разновидности методов координирования действий полезны, хотя предпочтение следует все же отдавать децентрализованным методам.

Необходимо способствовать непредвиденному поведению для координирования действий. И в этом помочь оказывают децентрализованные методы.

Команды для функциональных средств преследуют общую цель

Зависимости между командами для компонентов носят асинхронный характер, а у команд нет общей цели. Так, для работы над функциональным средством F могут потребоваться команды А и В. В текущем Спринте команда А работает над своей частью функционального средства F, и эта работа имеет для ее членов наивысший приоритет. Команда В приступает к работе над своей частью функционального средства F несколько Спринтов спустя, а затем пытается объединить свою работу с тем, что делает команда А. Но ей не удается этого сделать, и поэтому команда В пытается скоординировать свои действия с командой А. В итоге возникает конфликт, поскольку основное внимание команды А сосредоточено совсем на другом, а попытка скоординировать действия только мешает ей. Такой конфликт между командами можно охарактеризовать как назойливое вмешательство вследствие зависимости, и разрешить его не так-то просто.

При наличии команд для функциональных средств и общего кода все взаимодействие команд преследует общие цели. Например, в одном и том же Спринте команды для функциональных средств работают над разными элементами, внося изменения в общий код, и поэтому у них имеется некоторая общая или разделяемая работа. Вся эта общая работа и ее координирование связываются и согласуются в одном и том же Спринте. Команды для функциональных средств кровно заинтересованы работать вместе, поскольку координирование их действий дает возможность всем командам извлечь выгоду из работы над общим приростом продукта.

Основное внимание сосредоточено на всем продукте

Для сотрудничества команд требуется *общая цель*. С другой стороны, команды, планирующие собственные Спринты на будущее, ведут по отдельности свои заделы и вследствие этого уделяют основное внимание своей части работы, а не всему продукту. В итоге координирование действий между командами затрудняется, и поэтому поступать так нельзя. Вместо этого необходимо усилить основное внимание на всем продукте, чтобы команды разделяли общие цели получить общий прирост продукта в конце общего Спринта.

Многие элементы LeSS способствуют сосредоточению основного внимания на всем продукте. К их числу относится общий Владелец Продукта, общий Задел Продукта, одно Первое Планирование Спринта, одно Подведение Итогов Спринта и один интегрированный прирост продукта.

Удобные физические и технические условия



К физическим условиям относятся следующие.

- > Наилучшим условием для сотрудничества команд служит физическое нахождение вместе без отдельных кабинок или конторок. Каждая команда работает за одним столом в своем пространстве со многими плоскими поверхностями, эффективно используемыми в наглядном управлении производством.
- > Чтобы достичь как можно большего совмещения, следует сократить количество этажей, зданий и мест.
- > Для проведения больших совещаний в LeSS и рабочих совещаний по проектированию требуются большие белые доски, а также просторные, минимально меблированные помещения. Особенно удобны для этих целей помещения, оснащенные панелями с белыми досками от пола до потолка и принадлежностями для рисования на них.
- > Для расширения пределов взаимодействия, обучения, по-настоящему дружеских или партнерских отношений с высоким уровнем доверия и понимания очень важны регулярные переезды членов команд из одного места (или здания) в другое.
- > Для отдыха, обучения и непринужденного обмена информацией следует выделить пространства, общедоступные для команд. Примером тому могут служить кофейные и обеденные уголки с большими общими столами, а также библиотеки с удобными креслами и крупными собраниями книг, которые каждый может свободно взять и почитать.
- > Для коллективного труда следует предоставить удобные средства. К их числу относятся скамейки для парной работы, самоклеющиеся листки для заметок, белые доски формата А4 для набросков в парном режиме, листы, крепящиеся к белой доске и позволяющие начать обсуждение в любом удобном

месте, а также любые физические, перекидные и переносные приспособления, стимулирующие коллективный труд.

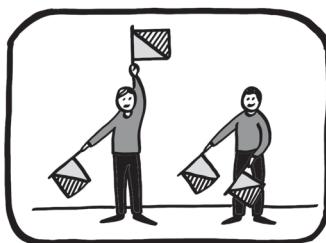
- > К техническим условиям относятся следующие.
- > Общие *информационные* пространства, в том числе вики-страницы, документы Google и пр. Вследствие больших потоков информации в крупной группе полезно выделить (хотя бы на условиях неполной занятости) роль *библиотекаря*, заботящегося об открытости информации, организуя, выделяя и помечая ее.
- > Общие пространства для *общения*, в том числе дискуссионные группы, списки рассылки, средства уведомления, средства для ведения видеоконференций и интерактивной переписки, особенно групповой (например, служба Slack).
- > Пространства для совместного и социального программирования, чтобы люди могли писать код вместе, будучи разделены во времени и пространстве. К их числу относятся средства для общего доступа к экрану (например, Screenhero), а также средства социального программирования (например, GitHub/GitLab), упрощающие общение по поводу кода.

Наставление: поддержание связи через код

Что требуется командам для координирования своих действий? Зачастую *интеграция*. Например, у команд может возникнуть потребность в общей работе над компонентом. По традиции они попытаются обнаружить такую потребность, прежде чем

вносить изменения, чтобы управлять своими зависимостями, исключить дублирование кода и конфликтов при его слиянии.

Все это наводит на следующий главный вывод: пути координирования действий отражают линии интеграции! Но при непрерывной интеграции такой вывод можно пересмотреть и тем самым обнаружить потребности в координировании действий по линиям интеграции.



По традиции координирование действий всегда поддерживало интеграцию, но и интеграция может поддерживать координирование действий.

Норма практики выявления потребностей в координировании действий через непрерывную интеграцию кода называется *поддержанием связи через код*. Как этого добиться? Допустим, член команды много раз на день вносит чужие изменения в свою локальную копию кода. Всякий раз он быстро проверяет изменения, и если обнаруживает, что другая команда работает в это же время над тем же самым компонентом, то

решает *поговорить* с ними о совместной работе, чтобы извлечь наибольшую взаимную выгоду из этой работы.

С этим связана еще одна, простая норма практики, которая состоит в том, чтобы вводить уведомления в систему контроля версий исходного кода. Это дает возможность подписаться на изменения в конкретном компоненте или файле, представляющем особый интерес.

Внимание: избегайте ветвлений!

Поддержание связи через код делает ветвление еще *большим* злом, чем оно было до того. И это особенно проявляется при масштабировании.

Ветвление не только задерживает интеграцию, но и препятствует координированию действий и сотрудничеству между командами.

Поддерживать связь через код нельзя без непрерывной интеграции.

Наставление: интеграция непрерывно

Всякий раз, когда мы посещаем группу, непременно спрашиваем: “Выполняете ли вы непрерывную интеграцию?” И в ответ неизменно слышим следующее: “О, да! Мы установили Jenkins!” (или эквивалентное инструментальное средство для сборки и непрерывной интеграции). Но как себя ведут разработчики в той же самой группе? И мы видим, что разработчики:

- > дни ожидают, пока не будет проверен код;
- > работают в отдельных ветвях (или локальных хранилищах Git).

Можно иметь сколько угодно систем сборки вроде Jenkins, но если разработчики задерживают объединение своего кода, то ни о какой непрерывной интеграции не может быть и речи.

Непрерывная интеграция подразумевает, что интегрировать следует непрерывно! Непрерывная интеграция — это поведение разработчиков, а не инструментальное средство.

Именно поэтому данное наставление призывает интегрировать *непрерывно* вместо непрерывной интеграции. Тем самым подчеркивается следующая важная особенность: непрерывная интеграция — это поведение разработчиков, а не инструментальное средство или система сборки. Что же в действительности подразумевает непрерывная интеграция?

Непрерывная интеграция означает поведение разработчика, стремящегося поддерживать работоспособность наращиваемой системы, внося в нее мелкие изменения посредством очень частой интеграции по основной ветви, поддерживаемой системой непрерывной интеграции с помощью автоматизированных тестов.

Ниже приводятся дополнительные пояснения такого обозначения понятия непрерывной интеграции.

- > **Поведение разработчика.** Непрерывная интеграция — это норма практики, которой разработчики пользуются все время. Но поскольку изменить основное поведение трудно, то наблюдается повальная имитация непрерывной интеграции. А поведению, способствующему подлинно непрерывной интеграции, препятствуют принятые правила. Так, во многих крупных группах устанавливается следующее правило: нарушать сборку нельзя, а те, кто не соблюдает это правило, подвергаются осуждению. И в конечном итоге интеграция, безусловно, задерживается! И тогда создается иллюзия передачи сборки. Такие правила непрерывной интеграции на самом деле мешают, а не помогают разработчикам. Каков же выход? Избавиться от опасений в обвинении и суждении, выполнять работу посредством тестирования и постоянную реорганизацию кода с целью сделать его чистым, поощрять частую интеграцию и прививать культуру оперативного устранения ошибок, когда сборка нарушается. В таком случае непрерывная интеграция превращается в норму практики, которая быстро извещает разработчиков о возникающих затруднениях и потребности в координировании действий. Если разработчики говорят, что непрерывная интеграция доставляет хлопоты, а не помогает им, значит, что-то не так.
- > **Мелкие изменения.** Крупные изменения в устойчивой системе способны нарушить ее устойчивость и работоспособность самыми разными способами. Чем крупнее изменение, тем больше (а иногда довольно много) времени требуется, чтобы сделать систему снова устойчивой. В связи с этим следует избегать крупных изменений, а вместо этого разделить каждое такое изменение на более мелкие, как предполагает понятие мелких партий при бережливом подходе. В этом случае каждое микроизменение легко интегрируется в систему.
- > **Нарашиваемая система.** *Нарашивание* системы подразумевает ее взращивание и развитие. Придерживаясь поведения, способствующего непрерывной интеграции, разработчик непрерывно интегрирует результаты своего труда. Он не ждет завершения всего функционального средства, а интегрирует при всякой возможности небольшой объем выполненной работы, не нарушая систему.

- > **Очень часто.** Как часто следует выполнять непрерывную интеграцию? *Непрерывно*. До какой степени часто должна интегрировать крупная группа, и будет ли всеобщая интеграция каждую секунду идеальным решением? Возможно, это и не достижимое, но верное направление, которое, впрочем, ограничивается следующими факторами.
- > **Умение людей разделять крупные изменения.** Специалист по разработке посредством тестирования (TDD) обычно способен разделять изменение вплоть до пятиминутного цикла для перехода в следующее устойчивое состояние.
- > **Скорость интеграции.** Ее можно увеличить, если, во-первых, вносить изменения очень мелкими партиями (например, за один пятиминутный цикл TDD), во-вторых, применять современные быстродействующие средства контроля версий исходного кода и, в-третьих, исключить правила, вносящие задержку (например, правило “просматривать код перед входным контролем”).
- > **Продолжительность цикла ответной реакции.** Его можно сократить, во-первых, выполняя быстрые тесты на быстродействующих компьютерах, во-вторых, распараллеливая операции и, в-третьих, поэтапно выполняя в несколько стадий подмножества тестов, которые быстро не проходят.
- > **Возможность работать в главной ветви.** Это означает отказ работать в порожденных ветвях.

Наставление: сообщества

В организации с межфункциональными командами группа по-прежнему должна проявлять внимание к таким насущным вопросам взаимодействия команд, как функциональные и прочие навыки, стандарты, инструментальные средства и проектные решения. Самым лучшим решением этих вопросов является создание сообществ.

Сообщество — это группа добровольцев от команд, разделяющих общие интересы или темы и стремящихся углубить свои знания или жаждущих действий через обсуждение или взаимодействие с равными себе. Участие со сообществом совершенно добровольно.

Сообщество не является командой и не реализует элементы. Как правило, сообщество собирается для выработки функциональной (например, проектной или архитектурной) нормы практики, но оно может собираться и по любым другим интересам, включая инструментальные средства инфраструктуры, поддержание связи, деятельность Scrum-мастеров и т.д. Область действия сообщества может охватывать лишь несколько команд (например, по одному сообществу на каждую Область Требований в LeSS Huge), продукт, место или предприятие.



Сообщества должны быть *динамичными*. Организовать сообщество может кто угодно. Но когда в сообществе пропадает взаимный интерес и желание что-то делать или оно становится неработоспособным, сообщество прекращает свое существование. Иногда это медленная агония!

Цели и полномочия сообществ

Сообщества преследуют следующие главные цели.

- > **Обучение.** Сообщества уделяют основное внимание распространению знаний, обучению и совершенствованию навыков. К этой категории относятся сообщества норм практики, например, сообщество чистого кода или тестовое сообщество.
- > **Соглашения между командами.** На уровне продукта или предприятия у команд возникают насущные вопросы, которые требуют внимания. Примером тому служат архитектурные нормативы, стандарты пользовательского интерфейса или нормы практики автоматизации тестов. Еще одним, менее очевидным примером соглашения между командами служит обсуждение в архитектурном сообществе развития архитектуры, которое должно направлять отдельные команды к правильным проектным решениям.

Многие сообщества преследуют обе упомянутые выше цели и зачастую достигают их. Например, тестовое сообщество может рекомендовать соглашение относительно автоматизации тестов.

Если сообщества вырабатывают соглашения, то могут ли они принимать решения, которые должны принимать команды? Нет, не могут.

Сообщества не могут принимать решения вместо команд. Но они могут произвести нечто такое, что команды решат принять.

Так, если сообщество хочет, чтобы результаты его трудов были приняты, ему лучше добиваться широкого участия в этом всех команд.

Рекомендации для сообществ

Процветающее сообщество извлекает выгоду из упреждающей организации и внимания. Удачное сообщество отличается следующими качествами:

- > имеет *координатора сообщества*, активно стремящегося или желающего культивировать сплоченность и заинтересованность в членах сообщества; на эту роль желательно выбрать активного практика;
- > активно пытается привлечь к участию в нем членов большинства команд;
- > находится на виду и легко доступно для всех, кто желает узнать о действующих сообществах и правилах вступления в их члены;

- > предпочтительно уделяет основное внимание решению конкретных задач, стремясь сделать обучение практическим и конкретным;
- > пришло к согласию о порядке своей работы и принятии решений;
- > может иметь своего Scrum-мастера, оказывающего помощь в его работе и совершенствовании и устраивающего его собрания и рабочие совещания;
- > пользуется вики-страницами, дискуссионными группами и групповой интерактивной перепиской;
- > регулярно собирается;
- > всячески поощряется в организации, где все знают, что сообщества полезны, и поэтому ожидается, что сотрудники организации будут вступать в сообщества и принимать активное участие в их деятельности.

Ниже перечислен ряд действий, которые только ставят палки в колеса, дезорганизуя сообщество.

- > Отказаться и игнорировать координатора сообщества (такое нередко происходит, если кого-нибудь назначают на эту роль).
- > Проводить частые собрания ради самих собраний.
- > Иметь большинство членов сообщества, не входящих в команды для функциональных средств.
- > Относиться к сообществу как к чему-то второстепенному и пренебрегать участием в нем, ссылаясь на “большую занятость”.

Рекомендуемые сообщества. Следует иметь в виду, что некоторые сообщества должны непременно существовать и процветать для того, чтобы деятельность группы была успешной. К их числу относятся сообщества человека-машинного интерфейса, проектирования, архитектуры и тестирования.

Виды деятельности и их результаты. Сообщество не является командой и не реализует элементы. Что же оно делает и какую пользу приносит? Ниже перечислены основные виды деятельности сообщества.

- > **Обучение.** К этому виду деятельности относится, например, обучение членов своих групп основополагающим принципам проектирования.
- > **Организация образования или обучения.** К этому виду деятельности относится, например, организация курсов по проектированию современного человека-машинного интерфейса.
- > **Предложение рекомендаций, норм или стандартов.** К этому виду деятельности относится, например, предложение рекомендации по проектированию современного человека-машинного интерфейса.
- > **Определение конкретной работы.** Например, такое определение, как “нам нужна намного более быстродействующая шина передачи сообщений”.
- > **Исследование.** Например, проведение всплеска с целью изучить что-нибудь.

- > **Изучение и обмен информацией.** К этому виду деятельности относится, например, организация собраний с блиц-докладами, где члены сообщества обмениваются информацией друг с другом.
- > **Проведение рабочих совещаний по проектированию.** К этому виду деятельности относится, например, совместное обсуждение у белой доски проектных замыслов.

См.раздел“Наставление: обращение со специальными элементами” главы9.

Сообщество может определить потребность в исследовании чего-нибудь конкретного. Крупные исследования должны проходить через задел продукта, но менее масштабные исследования могут проходить непосредственно через сообщества. Исследования и другие мероприятия могут потребовать дополнительной работы, например, “создать новый каркас”. Это работа не сообщества, а команды. Если дополнительная работа может быть сделана обычной командой для функциональных средств, ее следует зарегистрировать в Заделе Продукта, чтобы согласованно поручить ее командам для функциональных средств.

Сообщества на уровне организации. Помимо сообществ в одной группе, нередко требуются сообщества, охватывающие организацию в целом, например, для согласованного взаимодействия с пользователями разных продуктов. Традиционная организационная структура предполагает четкое распределение функций среди групп, тогда как в межфункциональной организации из членов команд для функциональных средств образуется сообщество, в которое входят представители сообществ различных продуктов, например, их координаторы.

См. раздел“Наставление: культура следует структуре” главы3.

Ложные сообщества. Традиционные крупные группы структурированы как однофункциональные команды, например, архитектурные, тестовые и т.д. Большинство организаций неявно оптимизировано таким образом, чтобы исключить должностями специалистов и ответственных за текущее положение дел, а также властные структуры. Следовательно, в них наблюдается появление ложных сообществ, которые на самом деле оказываются однофункциональными командами, переименованными в “архитектурные” и прочие сообщества с целью создать ложное впечатление, будто что-то изменилось, хотя на самом деле ничего не изменилось, и это весьма прискорбно.

Наставление: межкомандныесовещания



Естественно, что под термином *межкомандное совещание* подразумевается мероприятие, в котором участвуют члены хотя бы двух команд (все команды в полном составе или их представители). Межкомандные совещания в LeSS относятся также к категории много-командных, общих и прочих. Для конкретных категорий подобных совещаний в LeSS имеются отдельные

наставления, которые подытожены в приведенной ниже таблице и снабжены рядом примеров и рекомендаций далее в этом наставлении.

Многокомандное совещание	Общее совещание	Прочее
Многокомандное Уточнение Задела Продукта (см. раздел “Наставление: многокомандное Уточнение Задела Продукта” главы 11)	Первое Планирование Спринта (см. раздел “Наставление: Первое Планирование Спринта” главы 12)	Разные участники
Многокомандное Второе Планирование Спринта (см. раздел “Наставление: многокомандное Второе Планирование Спринта” главы 12)	Общее Уточнение Задела Продукта (см. раздел “Наставление: виды Уточнения Задела Продукта” главы 11)	Собрания сообществ
Многокомандное рабочее совещание по проектированию (см. далее раздел “Наставление: многокомандное совещание по проектированию”)	Подведение Итогов Спринта (см. раздел “Наставление: базар при подведении итогов” главе 14)	Рабочие совещания по изучению текущей архитектуры (см. далее раздел “Наставление: рабочее совещание по текущей архитектуре”)
Общая Ретроспектива (см. раздел “Наставление: Общая Ретроспектива” главы 14)		

Крупные совещания

Устроитель. Большинство совещаний оказываются более эффективными, если они проводятся умелым устроителем, но особенно это касается *крупных* собраний или рабочих совещаний. Подходящим кандидатом на роль устроителя служит Scrum-мастер, но его не следует просить устраивать подобное совещание, поскольку для этого требуется соответствующее образование и подготовка, особенно если речь идет об устроении совещаний в крупных группах.

Циклы расхождения–схождения. Собранные на обычных или рабочих совещаниях способствуют общему пониманию, согласованности и меньшей рассеянности информации, что, безусловно, является преимуществом. Но к недостаткам совещаний в крупной группе относится то обстоятельство, что многие люди не вовлекаются в общее обсуждение или их просто не слышат, а кроме того, подобные мероприятия не отличаются особым разнообразием и обилием идей. Поэтому такие совещания рекомендуется проводить по образцу циклов расхождения–схождения, когда команды или смешанные группы время от времени расходятся сначала по отдельным участкам помещения (например, на 30 минут), а затем сходятся вместе для проведения общего мероприятия или обсуждения. Очень важно объединить их усилия, но при этом нужно помнить следующее.

Централизация подавляет энергию, а децентрализация порождает ее.

Многоместные совещания

Во многих группах, практикующих LeSS, разработка ведется во многих местах. И нам не раз приходилось быть свидетелями проведения многоместных межкомандных совещаний. По поводу многоместных совещаний можно дать следующие рекомендации.

- > **Непосредственное общение как предусловие для чуткости.** Сотрудничество связано с доверием и чуткостью. Чтобы культивировать подобную атмосферу, необходимо задействовать свои чувства, а для этого мы должны видеть своих коллег и задействовать перечисленные ниже средства.
- > **Разнообразные инструментальные средства для проведения видеоконференций.** Нам приходилось наблюдать клиентов в противоположных ситуациях, когда, с одной стороны, требовалась дорогостоящая система видеоконференций в отдельном помещении, а с другой стороны, использовались разнообразные инструментальные средства для проведения видеоконференций и недорогие видеопроекторы. В итоге разнообразные инструментальные средства применялись постоянно и повсеместно.
- > **“Облачные” инструментальные средства для пользования общими документами.** Сравните следующие два варианта: обновить электронную таблицу в Excel и поместить ее файл в общую папку или же предоставить ссылку на общую электронную таблицу в Google, которую можно наблюдать и редактировать во время обсуждения. Второй вариант, безусловно, лучше.
- > **Циклы расхождения-схождения.** Более подробно эта методика описывается в предыдущем разделе. А в данном случае группы естественным образом разведены по отдельным местам, что заманчиво, хотя и является искусством возможного.

Наставление: многокомандное совещание по проектированию



Когда нескольким командам требуется коллективно обсудить предполагаемое проектное решение функциональных средств, компонентов или крупномасштабных архитектурных элементов, следует провести многокомандное рабочее совещание по проектированию средствами *гибкого моделирования*.

Когда проводится. Многокомандное рабочее совещание по проектированию может, а зачастую должно проводиться при многокомандном Втором Планировании Спринта, когда рассматриваются элементы, которые требуется реализовать. Но если новизна или сложность предстоящих элементов велика или же имеется возможность неспешно сформировать больше альтернативных вариантов и важно проявить творческую инициативу, то рабочее совещание по проектированию лучше провести в предыдущем Спринте. Хотя это менее желательно из-за возможного увеличения убытков, но все же может потребоваться потому, что Второе Планирование Спринта проводится в краткие и ограниченные сроки. Многокомандное рабочее совещание по проектированию в предыдущем Спринте сродни применению метода *надкусывания*.

См. раздел “Наставление: многокомандное Второе Планирование Спринта” главы 12.

См. раздел “Наставление: надкусывание” главы 9.

Для чего проводится. Для всего, что угодно! Рабочее совещание по проектированию может быть проведено для обсуждения предполагаемого проектного решения человека-машинных интерфейсов, моделей данных, алгоритмов, крупномасштабных компонентов, служб, взаимодействий и “архитектуры”.

Кем проводится. По определению в многокомандном рабочем совещании по проектированию принимают участие все члены двух или более команд. Важным и распространенным его вариантом является рабочее совещание по проектированию в архитектурном сообществе, в котором принимают участие представители многих команд.

Как проводится. Гибкое моделирование подразумевает создание всей группой простых и приемлемых моделей, способствующих творчеству, общению, наглядному представлению и быстрому изменению. Кredo гибкого моделирования состоит в следующем.

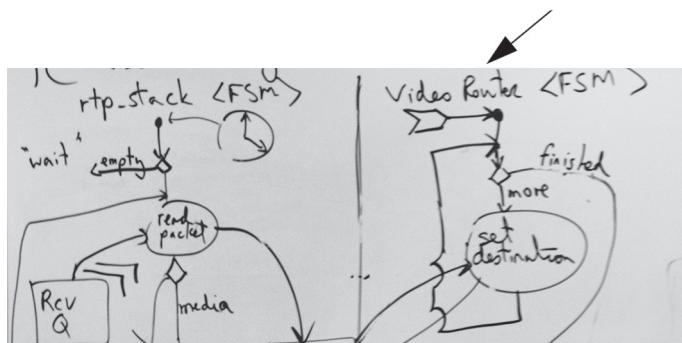
Мы моделируем, чтобы общаться.

Какие инструментальные средства применяются для гибкого моделирования? Прежде всего следует всячески избегать программных средств, поскольку они уничтожают сотрудничество, течение мысли и выработку идей. Вместо этого следует уделить больше внимания простым физическим средствам, включая листки клейкой бумаги и белые доски.

Ниже приведены некоторые рекомендации по проведению подобных рабочих совещаний.

- > **Обширные пространства в виде белых досок.** Успешное проведение рабочего совещания пропорционально площади рабочего пространства! И вместо стандартных белых досок все пространства стен следует покрыть специальным пластмассовым материалом, похожим на белые доски, например, листами типа “Wizard Wall” (Волшебная стена).
- > **Многоместное моделирование.** В этом случае применение программных средств неизбежно. В сочетании с видеоконференциями рекомендуется коллективное составление набросков в приложениях, имитирующих белую доску на планшетах или в окнах браузеров. Еще один метод, который мы практиковали для проведения рабочих совещаний по проектированию одновременно в двух местах, состоит в применении физических белых досок, хорошо видных в веб-камеру, в сочетании с циклами расхождения–схождения.
- > **Регистрация.** Важными составляющими гибкого моделирования являются общение, творчество и растущая согласованность, а не только записи на листках клейкой бумаги или наброски на белых досках. Если группе требуется запомнить или поделиться результатами своих трудов на рабочем совещании по проектированию, то его участники могут сделать фотографии и видеозаписи.

Это предназначено для общения и размышления,
а не для документирования



Модели не предназначены для документирования. Гибкое моделирование служит для составления приемлемых моделей, вдохновляющих на последующие действия (например, программирование), но не предназначенных для долгосрочного документирования, и поэтому они должны быть простыми. Если же группе требуется составить документацию, то для этой цели служит следующее наставление.

Наставление: рабочее совещание по текущей архитектуре



Межкомандное рабочее совещание по изучению текущей архитектуры проводится с целью ознакомиться с архитектурой, уже реализованной в коде. Такое изучение особенно важно для групп, лишь недавно внедривших LeSS и переходящих от команд для компонентов к командам для функциональных средств и общему коду. Дело в том, что раньше члены подобных групп очень мало знали общую архитектуру или же более подробную архитектуру других компонентов. А ведь такие знания очень важны для организации совместной работы команд, предназначенных для разработки функциональных средств. Инициаторами проведения подобных рабочих совещаний нередко являются архитектурные сообщества или кураторы отдельных компонентов.

Ниже приведены некоторые рекомендации по проведению подобных рабочих совещаний.

- > **Изучение архитектуры в существующем состоянии.** Участники данного рабочего совещания должны уяснить, что они изучают текущую архитектуру в существующем ее состоянии. Основное внимание на этом совещании должно быть уделено изучению и обучению. А для обсуждения предполагаемой архитектуры лучше проводить упоминавшиеся ранее рабочие совещания по проектированию.
- > **Наброски различных архитектурных представлений.** Для наглядного показа нескольких архитектурных представлений рекомендуется модель проекций “4+1”, включая логическую проекцию крупных компонентов, проекцию развертывания оборудования, организацию сети и процессов и т.д.
- > **Сценарии разыгрывания ролей.** Чтобы помочь участникам совещания лучше изучить и запомнить текущую архитектуру, можно составить занимательные сценарии, в которых они выполняют роли отдельных компонентов, передавая друг другу управление.
- > **Составление технических записок.** В подобных записках можно набросать краткое описание необычных и заслуживающих внимания архитектурных элементов или проектных решений, например, “причины, по которым мы пользуемся механизмом бизнес-правил Drools”, или “зачем и почему конечные автоматы являются нашей главной темой”.
- > **Выделение времени для вопросов и ответов с видеозаписью.** Помимо расхождения по отдельным участкам помещения для создания модели проекций “4+1” архитектурных представлений и составления технических

записок, участники данного рабочего совещания должны периодически собираться вместе и уделять время вопросам и ответам по каждому представлению и записке, обсуждая подробности. Все это должно сопровождаться видеозаписью.

Самое главное, что, находясь вместе, люди учатся. А кроме того, видеозапись, которую нетрудно организовать, делается для тех, кто сможет впоследствии посмотреть данное рабочее совещание.

Наставление: кураторы компонентов

Командам для функциональных средств приходится работать в незнакомых областях кода. Как же помочь им в изучении этих областей, особенно в тех случаях, когда программный компонент хрупок и уязвим. На помощь им может прийти *куратор компонента*, который:

- > является обычным членом команды для функциональных средств, но уделяет время кураторству;
- > проводит рабочие совещания по изучению текущей архитектуры, чтобы обучить других курируемому им компоненту;
- > находит и наставляет отдельных разработчиков во многих командах, чтобы они узнали больше о курируемом им компоненте;
- > обучает в режиме парного или коллективного программирования;
- > организует сообщество курируемого им компонента;
- > организует или принимает участие в рабочих совещаниях по проектированию, когда предполагается значительное воздействие на курируемый им компонент;
- > проверяет код и дает рекомендации по его улучшению;
- > побуждает разработчиков улучшать код и вводить тесты;
- > контролирует работоспособность курируемого им компонента в долгосрочной перспективе;
- > находит и наставляет других кураторов.

См. выше раздел “Наставление: интеграция непрерывно”.

Кураторы компонентов ничего *не* утверждают, они лишь обучаются и курируют свои компоненты. Кроме того, они *не* проверяют качество, поскольку это существенно задержало бы интеграцию, а следовательно, помешало бы координированию действий и сотрудничеству. В организации, где команды для компонентов разделяют общий код, принято следующее оптимистическое правило: разработчики подтверждают код, интегрируя его без задержек. Таким образом, вклад в общий код вносится следующими двумя способами.



- > **Прямое подтверждение кода.** В этом случае разработчики подтверждают и вносят свой код в общее центральное хранилище (в “голову ветки”) без задержки. Это оптимистическое поведение по умолчанию. А когда обнаруживается *ненормальный* код, куратор обучает разработчиков, прибегая к этому все реже по мере их совершенствования.
- > **Запрос на включение кода.** Куратор компонента не проверяет качество кода, но иногда (например, в том случае, когда кто-нибудь изменяет незнакомый код) разработчик *добровольно* направляет куратору запрос на включение кода, чтобы тот просмотрел код и дал свой отзыв, прежде чем производить слияние кода. Естественно, что разработчики делают прямые подтверждения знакомого кода или запросы на включение незнакомого кода, извлекая наибольшую выгоду из того и другого.

Кураторы *не* устраниют ошибки в компонентах. Весьма заманчиво направить программные ошибки в компоненте его куратору для локальной оптимизации. И это делается довольно часто и умело! Но тогда это уже не кураторы компонентов.

Разделение обязанностей наставничества особенно требуется на ранней стадии перехода к общему коду и командам для функциональных средств, когда куратор может оказаться перегруженным. В таком случае ему в помощь следует назначить дополнительных кураторов.

Наставление: Открытое Пространство

В Открытом Пространстве¹ регулярно проводятся совещания по обучению, координированию действий или иным причинам. Открытое Пространство — это метод проведения совещаний с присущей самоорганизацией. С этой целью выполняются следующие действия.

1. Совместно составляется повестка дня по насущным вопросам, разрешаемым в параллельном режиме.
2. Ответственные за созыв совещания приглашают желающих принять участие в параллельном обсуждении насущных вопросов в разных местах помещения. При этом соблюдается следующий закон двух футов: если кто-нибудь не учится или не участвует в обсуждении, он переходит в другое место.



Ниже приведены некоторые примеры применения метода Открытого Пространства в LeSS.

¹ Подробнее об Открытом Пространстве см. в книге *Open Space Technology* Харрисона Оуэна (Harrison Owen).

- > Регулярное (например, один раз в две недели) совещание по обучению и сотрудничеству. Проводить совещание в Открытом Пространстве рекомендуется с едой и напитками.
- > Совещание в формате Общей Ретроспективы для анализа ситуаций и экспериментов по усовершенствованию проектов.
- > Ежеквартальное однодневное совещание по углубленному изучению и укреплению социальных сетей.
- > Совещание в формате собрания сообщества.

Собрания, похожие на совещание в Открытом Пространстве

Совещания в Открытом Пространстве относятся к категории известных колективных мероприятий. Это собрания в формате “Мировое кафе”, “Бережливое кофе” и аналогичном формате “Бережливое пиво”.

Наставление: спутники



Однажды нам пришлось работать с группой продукта, в которой имелась пара самых опытных технических специалистов. Эта группа сформировала команды для функциональных средств, специально подобрав ее членов, но никак не могла решить, в какую из команд ввести самых редких своих специалистов, знания которых имели решающее значение для всех команд. (Попутно следует заметить, что разрозненность знаний проявилась как слабое место в результате внедрения LeSS.) В конечном итоге оба самых опытных технических специалиста стали *спутниками*.

Спутники работают как обычные члены команды в течение одного Спринта. Они разделяют общую ответственность за всю работу команды. Но еще важнее то, что они преследуют второстепенную цель: свести к минимуму зависимость команды от них — как правило, путем обучения. Следует также иметь в виду, что команды, у которых отсутствует спутник в текущем Спринте, действуют самостоятельно и должны сами решать, как достичь своих целей без постоянной помощи знающего спутника.

Чтобы подчеркнуть последнее обстоятельство, следует также заметить, что спутник старается избегать помочи другим командам во время Спринта. Это очень важная особенность его поведения, поскольку он должен подстегивать команду больше учиться у него в течение Спринта, чтобы устранить недостаток требующегося опыта.

Наличие специалиста по “узким местам” в качестве спутника на ранней стадии внедрения LeSS считается вполне очевидным. И хотя спутником может быть кто угодно, тем не менее, некоторым людям нравится работать именно таким образом. Более крупные группы могут извлечь особую пользу из спутников, поскольку они вносят не мало неформальных сведений и создают рабочие отношения между командами, а ведь это именно те элементы, которые ослабевают по мере укрупнения групп. Спутники

могут укрепить сеть каналов неформальной координации действий и повысить согласованность некоторых знаний или норм практики среди команд.

Следует, однако, иметь в виду, что в LeSS преобладает качество долговечного и устойчивого членства в каждой команде, поскольку команде требуется немало времени, чтобы превратиться в спаянный коллектив и двигаться в направлении высокой производительности труда. Спутники используются в LeSS не для того, чтобы превратить организацию в структуру матричного управления с недолговечными проектными командами.

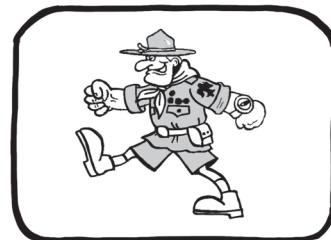
Когда и как спутники решают перейти в конкретную команду на один Спринт? Вероятно, при Первом Планировании Спринта. А поскольку речь идет о самоорганизующейся группе, то решение принимают спутник и команды, а не Владелец Продукта.

Спутники могут быть временными и постоянными, но для того чтобы стать спутником, нужно желание команды принять его в свои ряды на один Спринт. Сами спутники не навязывают себя командам. Таким образом, общая группа сама регулирует потребность в спутниках, которые не могут быть постоянными, если команды больше не принимают их. И тогда или в любое другое время спутник может найти себе пристанище.

Наставление: разведчики

Самый простой метод организовать совместную работу команд — отправить *разведчика*, а не Scrum-мастера в другие команды, чтобы он узнал что-нибудь и, вернувшись, доложил обстановку. Это довольно простой способ обучения, когда требуется просто с кем-нибудь поговорить по делу.

Наиболее благоприятным временем и местом для вылазок разведчика служат Ежедневные Летучки в других командах, где он присутствует в качестве молчаливого наблюдателя. Какие же команды он должен посетить? Вероятно, те, с которыми его команда проводила многокомандное Второе Планирование Спринта или многокомандное Уточнение Задела Продукта.



Наставление: стоит ли проводить Схватку над Схватками

Схватка над Схватками — это совещание, подобное Ежедневной Летучке среди представителей команд, а не Scrum-мастеров или руководителей. Такое совещание обычно проводится три раза в неделю, но поскольку оно формально централизовано, то ему *не* следует отдавать предпочтение.

Впрочем, Схватки над Схватками иногда действительно полезны, и тогда их стоит проводить! Но чаще всего команды, только начинающие осваивать принципы масштабирования, испытывают потребность проводить подобные совещания из-за неверного представления о масштабировании. И они проводят их, хотя пользы от них на самом деле немного. В таком случае следует отказаться от такой практики и лучше опробовать другие методы координации действий.

Наставление: ведущая команда

Ведущей называется такая команда, которая имеет дополнительные обязанности по доставке функционального средства или ряда связанных с ним функциональных средств. Такая команда уточняет и реализует элементы функционального средства, но дополнительно уделяет внимание общему представлению о всем ряде функциональных средств. Как правило, в обязанности ведущей команды входит обучение и координирование действий (зачастую с внешними группами).

См. раздел “Наставление: новая область для очень большого требования” главы 9.

Обучение. Ведущая команда в большей степени связана с рядом функциональных средств, чем другие команды, вероятно, благодаря своему богатому профессиональному опыту или же потому, что это первая команда, работающая в данной области.

А поскольку другие команды в конечном итоге соединяют свои усилия в работе над большим числом связанных вместе функциональных средств, то таким присоединяющимся командам требуется обучение (например, предметной области и развитию решения), и эту обучающую роль берет на себя ведущая команда. Например, во время многокомандного Уточнения Задела Продукта члены ведущей команды поясняют предпосылки или подробности создания элементов, над которыми они работали, чтобы помочь другим командам уяснить новые элементы. Кроме того, они проводят рабочее совещание по изучению текущей архитектуры, на котором рассматриваются новые основные элементы.

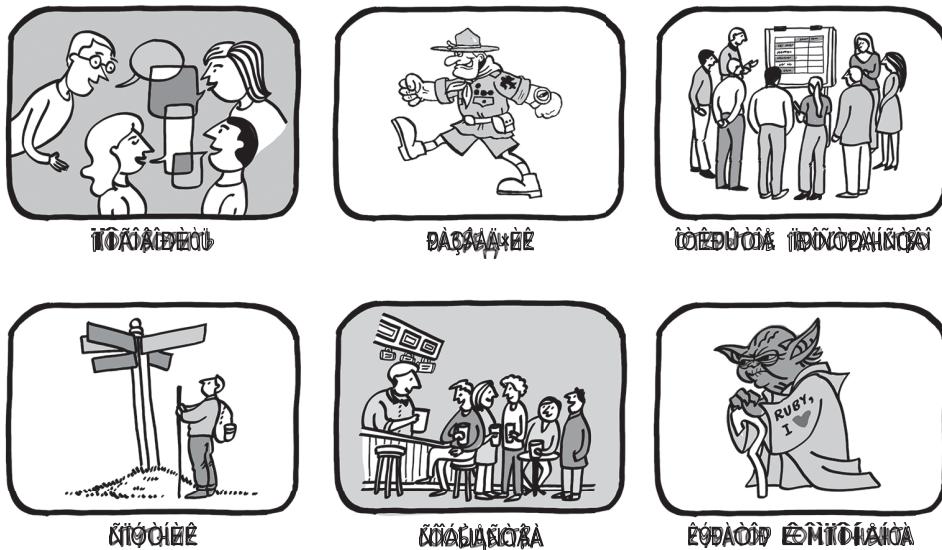
Координирование действий. Для реализации крупного функционального средства или ряда подобных средств ведущая команда нередко берет на себя обязанности по координированию действий с *внешними группами*, к которым относится внешняя группа, создающая компонент или отдел незавершенных работ. А координирование действий между внутренними командами следует предоставить в как можно большей степени самим командам. Ниже перечислены разновидности такого координирования действий.

- > **Координирование действий с внешними группами для компонентов.** Крупные продукты нередко содержат компоненты, созданные другой группой, — по крайней мере, тогда, когда только начиналось внедрение LeSS. Поэтому определенная работа должна быть сконцентрирована. И ее должна выполнять ведущая команда, а не многие команды или отдельная управленческая группа. И хотя такое координирование действий осуществляется ведущей командой, *прояснение* должно происходить непосредственно между конкретной командой и внешней группой во избежание потерь на дополнительную передачу работ.
- > **Координирование действий с отделом незавершенных работ.** Ведущие команды берут на себя всю ответственность довести функциональное средство до поставки в продукте. При наличии нестрогого Критерия Готовности требуется координирование действий и поддержка отдела незавершенных работ, когда проводятся завершающие мероприятия, чтобы довести

функциональное средство до подлинно готового к поставке состояния. Следует, однако, иметь в виду, что ведущая команда берет на себя обязанности, которые по традиции возлагались на руководителей проектов и ответственных за выпуск продукции.

Наставление: сочетание разных методов

Многие методы, рассмотренные в этой главе, могут подкреплять друг друга и применяться совместно. Ниже приведены характерные тому примеры.



- > **Сообщество для компонентов.** Куратор компонента становится координатором в сообществе для компонентов. В таком сообществе обсуждаются вопросы устранения ошибок при сборке, анализе кода и т.д. Оно собирается регулярно, и время от времени ему требуется...
- > **Открытое Пространство в сообществе для компонентов.** Обнаружив потребность в тщательном обсуждении и обучении, сообщество для компонентов решает организовать совещание в Открытом Пространстве. Разумеется, на такое совещание приглашают не только членов данного сообщества, но и всех желающих. Понимая и обсуждая насущные вопросы, они в конечном итоге обнаруживают, что им требуется...
- > **Многокомандное рабочее совещание по проектированию с куратором компонента.** Предвидя немалые изменения в компоненте, куратор компонента организует рабочее совещание по проектированию. В ходе этого совещания куратор компонента выявляет те элементы, к реализации которых он предлагает присоединиться как...

- > **Спутник и куратор компонента.** Многим новым командам для функциональных средств недостает специальных знаний куратора компонента, чтобы внести изменения в компонент. Поэтому он становится спутником тех команд, которым больше всего требуется его помочь. Чтобы не почувствовать себя одиноким путником, он решает вступить в...
- > **Сообщество спутников.** Все спутники делятся в своем сообществе своим опытом работы с командой в течение одного Спринта, обсуждают насущные вопросы и учатся друг у друга. И в конечном счете они решают организовать...
- > **Открытое Пространство в сообществе спутников.** На совещание в Открытом Пространстве, безусловно, приглашают не только спутников! В нем принимают участие члены сообщества Сообществ, чтобы перенять опыт организации открытых пространств в сообществах и поделиться информацией с другими командами.

Чем больше методов сочетается, тем более эффективными они становятся.

LeSS Huge

См. выше раздел “Наставление: интеграция непрерывно”.

Большая часть приведенных выше наставлений распространяется и на LeSS Huge. Некоторые из них (например, наставление *интегрировать непрерывно*), по существу, являются нормами практики в Областях Требований. Специальные правила в LeSS Huge не предусмотрены.

Опираясь на упоминавшееся ранее наставление *поговорить*, следует черпать вдохновение из его мотивации и подхода к поощрению *неформального децентрализованного* сообщения между Областями Требований.



Базар при Подведении Итогов Спринта в LeSS

Подведение итогов и ретроспектива

Конституция должна быть краткой и непонятной.
— Наполеон Бонапарт

Scrum для одной команды

В основу Scrum положено управление эмпирическим процессом как для самого *продукта*, так и для его *создания*. Сначала создается небольшой поставляемый прирост продукта, а затем проводится тщательное обследование и адаптация того и другого. По существу, в этом и состоит главная цель Подведения Итогов Спринта и Ретроспективы.

При Подведении Итогов Спринта пользователи или потребители и прочие заинтересованные стороны узнают о продукте от его Владельца и Команды. Пользователи на практике исследуют новые элементы, а все остальные участники данного мероприятия — состояние рынка вместе с пользователями. И, наконец, что не менее важно, они обсуждают, что делать дальше. А на Ретроспективе Спринта Команда анализирует приобретенный опыт и исследует простые пути доставки впечатляющего прироста продукта, улучшающего условия существования и облегчающего жизнь, в том числе и самим разработчикам. С этой целью они подготавливают эксперимент, чтобы опробовать его в следующем Спринте, стремясь к такому трудно достижимому совершенству.

Подведение Итогов Спринта и Ретроспективы в LeSS

Приведенные далее наставления охватывают Подведение Итогов и Общую Ретроспективу, но не Ретроспектины в отдельных командах. При масштабировании соблюдаются следующие связанные с этим принципы.

- > **Ориентирование на клиентов.** Зачем привлекать пользователей или потребителей к участию в Подведении Итогов каждого Спринта? Вследствие разобщенности прежние группы вообще не привыкли к совместному обучению. Нам встречалось слишком много команд, которые вообще не общались с пользователями и опасались привлекать их к подведению итогов, поскольку это означало бы полную прозрачность их деятельности.
- > **Прозрачность.** Руководители признают благотворные преимущества прозрачности, но с опаской наблюдают за тем, что происходит, когда в группе, только начинающей практиковать LeSS, слишком откровенно проявляется подлинная прозрачность. Многие группы непрозрачны и боятся раскрыть фактически неблагополучное положение дел. И преодолеть такие опасения очень трудно.
- > **Непрерывное улучшение до полного совершенства.** Нам довелось иметь дело с клиентами, которые проводили один раз в год *посмертные разборки*, чтобы запланировать восхитительно сказочные улучшения на следующий год. А работая во многих крупных группах, нам приходилось не раз слышать, что “дела, в общем, обстоят совсем не плохо”. Эти примеры свидетельствуют об отсутствии *внутреннего* желания к совершенствованию.
- > **Управление эмпирическим процессом.** Во многих крупномасштабных организациях имеется централизованный процесс или группа управления проектами, на которые возложены задачи совершенствования и внедрения “улучшений” в команды, опираясь на культуру тейлоризма. Наделять такими полномочиями или привлекать для этой цели специальных работников не имеет никакого смысла. Ведь понятие управления эмпирическим процессом для создания продукта и организации работы над ним в *каждом* Спринте не имеет ничего общего с подобными закоренелыми привычками.
- > **Сосредоточение основного внимания на продукте в целом и системный подход.** Крупные группы с разобщенными командами не имеют привычки или свойство рассматривать целое, отвечать за все и мыслить системно.

Правила LeSS

На каждый продукт приходится лишь одно Подведение Итогов Спринта. Оно является общим для всех команд. На этом мероприятии должны присутствовать соответствующие заинтересованные стороны, чтобы поделиться информацией, необходимой для эффективного обследования и адаптации.

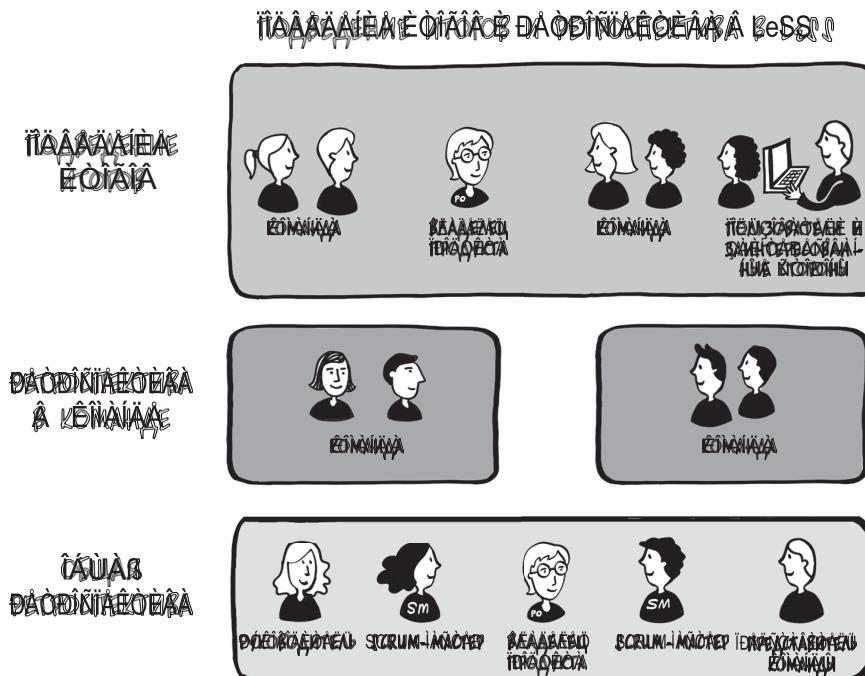
Каждая Команда проводит свою Ретроспективу Спринта.

Общая Ретроспектива проводится после Ретроспектив в отдельных командах с целью обсудить насущные вопросы среди команд и на системном уровне, а также подготовить эксперименты по улучшению. Данное мероприятие посещают Владелец Продукта, Scrum-мастера, представители Команд и руководители, если таковые имеются.

Наставление: ранняя и частая адаптация продукта

Если организация насчитывает всего девять человек, то вряд ли в ней практикуется составление ежегодного плана и графика выполнения работ, которому нужно строго следовать, чтобы сдавать заказчикам в срок готовые части продукта крупными партиями. Но в крупной группе подобные мероприятия практикуются по причинам, которые слишком сложны и здесь не рассматриваются, чтобы не раскрывать целый ящик Пандоры. Следовательно, когда крупная группа переходит к LeSS, вполне возможно, что она переносит старый багаж прогнозирующего планирования и приемочного контроля в Ретроспективу Спринта, которая в итоге превращается в мероприятие с целью выяснить, укладывается ли группа в сроки и готовы ли элементы к приемке.

Этого нельзя ни в коем случае допустить, но лучше опробовать гибкую методику и совместное обучение. При Подведении Итогов Спринта следует искать новую информацию о побуждающих факторах роста прибыли, стратегических клиентах, коммерческих рисках, конкурентах, новых трудностях и возможностях, чтобы адаптироваться и выбрать направление дальнейшего развития продукта на следующий Спринт. А кроме того, нужно обсудить вместе новые элементы, чтобы каждый чему-нибудь научился. В этом, собственно, и заключается главная перемена в умонастроении и поведении крупных групп. На приведенном ниже рисунке дается наглядное представление о подведении итогов и ретроспектива в LeSS и участниках этих мероприятий.



Подведение итогов и ретроспективы в LeSS

Наставление: базар при подведении итогов

Базар при Подведении Итогов Спринта аналогичен ярмарке знаний и умений. Данное мероприятие проводится в большом помещении, где представители команд распределются по многим участкам для исследования и обсуждения разрабатываемых элементов вместе с пользователями и прочими заинтересованными сторонами. Характерный тому пример приведен на иллюстрации в самом начале этой главы.

Следует, однако, иметь в виду, что базар охватывает не все Подведение Итогов. Ведь после такого базара очень важно провести дополнительное обсуждение и решить, что делать раньше.

Для проведения такого базара его участникам необходимо выполнить на макроуровне следующие действия.

1. Разойтись для исследования отдельных элементов в базарном стиле.
2. Сойтись для совместного обсуждения вместе с Владельцем Продукта следующих действий. Для этого следует зарезервировать достаточно времени.

Ниже перечислены действия, предпринимаемые в конкретном примере проведения базара.

1. Подготовьте участки для исследования разных видов элементов. Такие участки необходимо снабдить устройствами, на которых можно выполнять разрабатываемый продукт. Члены команд обсуждают элементы на каждом участке с пользователями, представителями других команд и прочими заинтересованными сторонами. Обучение происходит с обеих сторон! Для записи заслуживающих внимания моментов и вопросов следует предоставить бумажные карточки для отзывов.
2. Пригласите людей, включая членов других команд, посетить любые другие участки базара.
3. Запустите таймер на краткий период времени (например, на 15 минут), в течение которого проводится исследование элементов. Таймер задает размеренный ритм для перехода с одного участка на другой.
4. Обсудите совместно и запишите заслуживающие внимания моменты на карточках в ходе практического исследования элементов.
➤ **Совет:** избегайте демонстраций, поскольку они отвлекают пользователей и не побуждают к проявлению глубокой ответной реакции. Вместо этого поощряйте применение пользователями исследуемых элементов на практике. В этом случае члены команд могут ответить на вопросы пользователей или дать им соответствующие наставления.
5. Пригласите участников по окончании короткого цикла исследований сменить участки или остаться на следующий цикл. Такие мини-циклы способствуют более широкому и разностороннему исследованию всех элементов.

После базара предпринимаются следующие совместные действия.

1. Участники сортируют свои карточки с отзывами и вопросами, чтобы Владелец Продукта выбрал среди них самые важные.
2. Когда все соберутся вместе, Владелец Продукта проводит обсуждение карточек с отзывами, как показано на рис. 14.1.



Рис. 14.1. Владелец Продукта проводит обсуждение карточек с отзывами

3. Владелец Продукта проводит обсуждение рынка и клиентов, предстоящей ответной реакции деловых кругов и рынка на разрабатываемый продукт, а еще шире — внешней обстановки.
4. Для всего подведения итогов в целом самое главное — обсуждение, а возможно, и решение относительно выбора направления на следующий Спринт.

Как проводить базар сразу в нескольких местах? С этой целью можно, в частности, организовать базар в каждом месте (если допускается разница во времени), но отзывы и вопросы должны непременно поступать Владельцу Продукта. А для совместного обсуждения после базара можно воспользоваться инструментальными средствами, предназначенными для проведения видеоконференций.

Еще одним дополнением базара может стать экспериментирование с функциональными средствами на устройствах там, где они установлены. Для регистрации ответной реакции вместо карточек можно воспользоваться программными средствами, выделив, например, по одному окну на каждый элемент в интерактивной переписке.

Наставление: Общая Ретроспектива

“Мы не можем выпускать продукты непрерывно из-за правил развертывания”. “У нас слишком много мест разработки”. “Наш код никуда не годится”. “Чтобы узнать требования государственных регулирующих органов, требуется целая вечность”. “Мы продвигаемся очень медленно”. “Пользователи не принимают никакого участия”. “Отдел кадров нам не позволяет”. “Поставщики не задействованы”.

Эти и многие другие заявления нам приходилось не раз слышать в группах, вне-драйвящих LeSS. Их объединяют общие проблемы как для отдельных команд, так и для всей системы, начиная с общего замысла и кончая наличными средствами.

Удобное время и место для разрешения подобных системных проблем и обдумывания возможностей усовершенствования системы предоставляет Общая Ретроспектива. Кто же принимает в ней участие? Владелец Продукта, представители команд, Scrum-мастера и руководители. А почему именно они? А потому что они являются неотъемлемой частью *системы* и заинтересованы в ее усовершенствовании. Во-первых, они обсуждают и изучают разные особенности системы, во-вторых, подготавливают эксперимент по усовершенствованию системы в следующем Спринте и, в-третьих, анализируют результаты эксперимента, намеченного в предыдущей ретроспективе, пользуясь этим для изучения и адаптации.

Одним из принципов LeSS является *непрерывное улучшение до полного совершенства*. Как-то раз мы посетили очень крупную группу, рассматривавшую возможность внедрения LeSS, и ее руководитель сказал нам: “Мы получаем прибыль и имеем постоянную клиентуру. Зачем нам вообще совершенствоваться?” На своем опыте мы познали, что преодолеть такое отношение едва ли не труднее всего на ранней стадии внедрения LeSS, поскольку в предыдущей системе слишком много людей было отчуждено от потребителей и результатов коммерческой деятельности. Установление тесных связей с настоящими потребителями и пользователями и привлечение их к владению продуктом имеет решающее значение для культивирования внутреннего желания улучшений до полного совершенства. А в чем оно состоит? На этот вопрос нельзя дать однозначный ответ, но можно привести следующие убедительные примеры.

См. раздел “Наставление: организационная концепция совершенства” главы 3.

- > Продукт необыкновенно популярен и прибылен, лишен дефектов, а его функциональные средства легко создаются.
- > Организация действует гибко. Она способна изменить направление без особого труда или существенных затрат, быстро вернув с прибылью каждую затраченную копейку.
- > Все обладают обширными и глубокими знаниями, проявляют большую заинтересованность в клиентах и продукте и вполне удовлетворены результатами своего труда.

Все это позволяет группе продолжать хотя бы какое-то время совершенствоваться! Ниже приведены некоторые рекомендации относительно проведения Общей Ретроспективы.

- > Проанализировать результаты последнего эксперимента.
- > Как подчеркивается в следующем наставлении, уделить основное внимание системе в целом.
- > Проводить Общую Ретроспективу пораньше в следующем Спринте, поскольку последний день Спринта обычно посвящается Подведению Итогов и

Ретроспективам в Командах, а следовательно, людям будет трудно провести еще и это мероприятие в тот же день.

- > Включить хотя бы еще две следующие важные стадии.
 1. Анализ каких-нибудь систематических проявлений.
 2. Разработка эксперимента для систематического усовершенствования.
- > Подготовить только один новый эксперимент, действуя сосредоточенно и последовательно.
- > Не забывать, что на эксперимент может уйти немало недель, а то и месяцев, особенно в крупномасштабных системах, поэтому новый эксперимент должен быть тесно связан с предыдущим.

Многоместную Общую Ретроспективу можно попробовать провести в режиме видеоконференции и циклов расхождения–схождения. Например, сначала в каждом месте проводится отдельный анализ пяти главных причин или системное моделирование проблемы, затем участники в каждом месте обмениваются результатами, далее в каждом месте осуществляется коллективная выработка контрмер, а после этого участники со всех мест обмениваются своими предложениями и подготавливают эксперимент. Но некоторые вопросы (например, среды и культуры) носят местный характер, поэтому их лучше решать, проводя отдельные ретроспективы на местах. Пример проведения многоместной Общей Ретроспективы приведен на рис. 14.2.



Рис. 14.2. Проведение многоместной Общей Ретроспективы на стадии расхождения

Еще одну возможность в LeSS предоставляет *многокомандная* Ретроспектива, в которой принимают участие члены всех команд. Потребность в проведении подобного мероприятия может возникнуть у некоторых команд, когда, например, они работают в

тесном сотрудничестве. Но оно ни коим образом не заменяет Общую Ретроспективу, где основное внимание уделяется всей системе.

Наставление: совершенствование системы

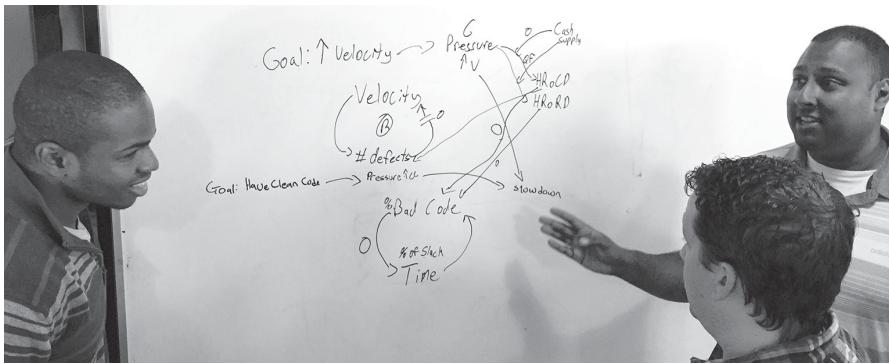
Мы все безотчетно делаем ошибку, размыщляя о локальных проблемах и оптимизации. Как ни странно, при проведении Общей Ретроспективы об этом свидетельствует накапливание результатов ретроспектив, проводимых на уровне команд, в качестве отправной точки для общего анализа. Но при таком методе восходящего анализа упускается из виду следующий важный вывод из системного подхода: *система не является простой суммой ее составных частей*. Именно поэтому следует всячески избегать восходящего анализа, хотя это совсем не означает, что следует пренебрегать вопросами, назревшими во всех командах. Они также требуют внимания. В данном случае имеется виду следующее менее очевидное обстоятельство.

Чтобы понимать и совершенствовать систему,
нужно уделять внимание системе в целом.

А что такое *система*? Это все, что и кто угодно: от общего замысла вплоть до наличных средств, включая всю ее динамику во времени и пространстве. Люди, организационная структура, физические и технические условия и прочее являются частью системы, и все они тесно связаны во взаимодействии.

В известной мере мысль распознать систему кажется тривиальной и бесполезной. Но это не совсем так, поскольку эволюция человека еще не достигла такого уровня мышления, чтобы он задумывался над нелинейной динамикой задержек в своей организации. Его мышление пока еще остается на уровне удовлетворения обычных потребностей вроде наличия шоколадки в нужный момент. И такая локальная перспектива только усиливается в крупных старых организациях, где группы специализируются на чем-нибудь одном, что приводит к утрате системной перспективы. В частности, группа анализа деловой активности озабочена решением собственных задач и необходимостью действовать эффективно, хотя и локально. Ее члены даже не знают, а возможно, и не собираются знать о других перспективах. Короче говоря, существуют биологические, структурные, культурные и условные факторы, позволяющие увидеть *часть*, но не *целое*.

Понимание. Как же применить системный подход и как понять систему, а точнее, как обсудить и обдумать модель системы? Это делается с помощью *системной модели*, иначе называемой циклической *диаграммой причинно-следственных связей*. Для построения системной модели теперь используется язык визуального моделирования и обозначения, но прежде выясним, что происходит при обсуждении, показанном на приведенном ниже рисунке.



На первый взгляд, его участники рисуют диаграмму в определенном обозначении, хотя здесь важна не форма, а содержание. Одни обдумывают и обсуждают систему и динамику, применяя *системный подход*. Более того, и этого не стоит недооценивать, они демонстрируют следующее кредо грамотного моделирования.

Мы моделируем, чтобы общаться и в конечном итоге добиться общего понимания, а не создать модель.

Когда группа совместно набрасывает системную модель в ходе Общей Ретроспективы, ее члены исследуют взаимное понимание и мнение о системе в ее текущем состоянии. Сначала они делают сложные и невидимые заметки в своем уме, а затем высказывают их, например, таким образом: “А теперь я понимаю, что вы думаете о текущей системе. Верно ли это?”

Большее понимание на ранней стадии внедрения. В этом наставлении подчеркивается, насколько важно применять системное моделирование не только во время ретроспектив, но и на начальной стадии внедрения LeSS, которая называется “0. Обучить всех” и поясняется в главе 3.

См. раздел “Наставление: начало внедрения” главы 3.

Действие. Общая Ретроспектива включает в себя вторую важную стадию разработки эксперимента по усовершенствованию системы: действие! На этой стадии можно также применять системное моделирование. Например, группа может обдумать будущую модель системы, обсудить и исследовать последствия ее внедрения. А кроме того, ее члены могут обсудить и смоделировать динамику внедрения и конкретного эксперимента для внесения изменений в систему в ее текущем состоянии. Что же может при этом произойти? Мы, конечно, не можем предсказывать будущее, но можем обдумать возможные его варианты. Помимо эксперимента для выполнения очевидного действия, следует обратить внимание на то, что может произойти нечто менее очевидное: представления людей меняются по мере того, как они лучше узнают модель системы, а это естественно приводит к улучшению поведения или процесса принятия решений в будущем независимо от какого-то конкретного действия.

Первые шаги в изучении системной модели

Для системного моделирования имеется нетривиальный язык, поскольку сами системы нетривиальны. Этот язык не особенно сложен, но очень удобен для обсуждения системной модели. Ниже приведены его основные элементы.

- > **Переменная** — нечто, выражаемое в величинах, поддающихся измерению, например, скорость (частота доставки) функциональных средств или качество кода.
- > **Причинно-следственная связь** — определяет взаимное влияние переменных. Так, если увеличивается количество функциональных средств, то увеличиваются и убытки, и наоборот.
- > Однако для системного подхода крайне важно уметь мыслить категориями взаимодействий и причинно-следственных связей. И тем более это касается моделирования крупномасштабных систем, поскольку они обширны во времени и пространстве, а динамика взаимодействий между многочисленными их частями обычно полна *скрытых*, но очень важных фактов и решающих факторов.
- > **Обратное действие** — такое действие может иметь причинно-следственную связь. Так, если возрастаёт доля в процентах слабых разработчиков, то снижается качество кода, и наоборот.

Набросок, приведенный на рис. 14.3, демонстрирует обозначение переменных, также прямых и обратных причинно-следственных связей. Набросок системной модели рекомендуется начертить на белой доске, снабдив переменные наклейками с комментариями, чтобы их можно было легко перемещать.

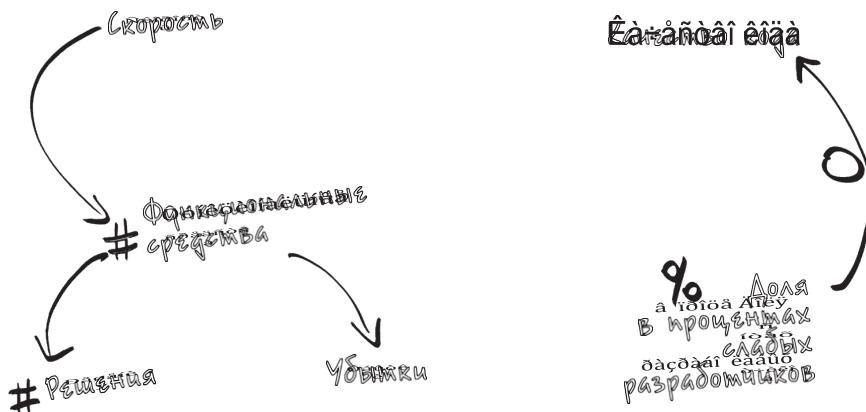


Рис. 14.3. Обозначение переменных, причинно-следственных связей и обратных действий на наброске системной модели

Ниже перечислен ряд других полезных понятий и связанных с ними обозначений.

- > **Задержка.** Одна из главных причин ложных представлений о поведении системы заключается в том, что влияния могут происходить с задержками. Причины и следствия не замкнуты ни во времени, ни в пространстве при крупномасштабной разработке. А задержанные последствия (например, потери информации) могут скрываться в эффектах взаимодействия групп. Поэтому люди испытывают трудности в выявлении и изучении их динамики. Например, поскольку руководители вынуждены увеличивать скорость работ, они принимают *сиюминутное* решение нанять много дешевых, а в конечном счете — слабых разработчиков. В краткосрочной перспективе это дает видимость увеличения скорости работ, но в долгосрочной перспективе возникают задержанные последствия снижения качества кода, приводящие к замедлению работ одиннадцать месяцев спустя.
- > **Представление.** Обсуждение представлений является еще одной важной нормой практики в системном моделировании. Одно дело — сделать набросок, утверждать, подразумевать или допускать, что руководители могут оценивать труд разработчиков, не особенно вникая в их код, и другое дело — признать, что может сложиться представление, а не факт. Если мы моделируем, чтобы общаться, то системное моделирование — это удобный момент для того, чтобы обсудить и осознать наши представления, сделать их видимыми и подвергнуть критическому анализу.

Практически каждая причинно-следственная связь или переменная дает возможность изучить и обсудить представления. Например, является ли скорость подходящей переменной для включения в системную модель? К чему приведет ее измерение? Действительно ли слабые разработчики создают некачественный код? Что имеется в виду под утверждением “чем больше функциональных средств, тем больше убытков”?

Набросок, приведенный на рис. 14.4, содержит обозначение задержки двумя линиями, перечеркивающими причинно-следственную связь, а также неформальные заметки, фиксирующие результаты обсуждения. Разумеется, само обозначение не имеет никакого значения при наличии общего понимания в группе. Данный пример системной модели не поучительный, он только демонстрирует, что мы моделируем, чтобы общаться!

Обучение. Чтобы узнать больше о системном подходе и моделировании, рекомендуется обратиться к следующим первоисточникам.

- > Глава, посвященная системному подходу, книги *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, Larman and Vodde. Эта глава доступна также на веб-странице по адресу http://less.works/less/principles/systems_thinking.html.
- > Очень важный классический труд *The Fifth Discipline*, Peter Senge.
- > *Thinking in Systems*, Donella H. Meadows and Diana Wright.
- > *Systemantics*, John Gall.



Рис. 14.4. Обозначение задержек и неформальных заметок на наброске системной модели с целью прояснить обсуждение

LeSS Huge

Правила LeSSHuge

Для подведения итогов и ретроспектив в LeSS Huge не предусмотрено никаких особых правил. Универсальное утверждение “Все правила LeSS распространяются на каждую Область Требований” подразумевает, что Подведение Итогов Спринта и Общая Ретроспектива проводятся по отдельности в каждой Области Требований. Но к проведению совещаний, охватывающих продукт в целом, не предъявляется никаких особых требований.

Наставление: подведение итогов и ретроспектизы во многих областях

Подведение итогов. Это мероприятие охватывает все области (от двух и больше, т.е. продукт в целом) и, безусловно, проводится тогда, когда группа испытывает в нем потребность, хотя оно и не является обязательным.

Почему же Подведение Итогов Спринта на уровне продукта обязательно в LeSS Huge? Ведь если не провести это мероприятие, то ослабнет внимание и способность видеть целое. Прежде всего, группа *может* подводить итоги на уровне продукта. Но

каждая область зачастую имеет достаточно отличий, чтобы сделать какие-нибудь существенные выводы из такого мероприятия — по крайней мере, не в каждом Спринте. И это особенно справедливо в масштабах LeSS Huge, где, помимо необходимости собирать целую Команду Владельцев Продукта, разработка может вестись в десяти местах по всему миру, что существенно затрудняет подведение итогов на уровне продукта. Следовательно, если и возникают потребности в таком мероприятии, то, вероятно, не в каждом Спринте.

Ретроспектива. И в данном случае не существует правил, требующих проведения Ретроспективы на уровне продукта, но, безусловно, могут появиться веские основания для проведения Ретроспективы во многих областях, поскольку преследуется главная цель — усовершенствование системы, а организационная структура охватывает все области. Ретроспектива во многих областях более вероятна, когда совместная работа в областях не налажена или же в ряде областей возникают сходные затруднения. Кроме того, командам из разных Областей Требований полезно также работать вместе в одном и том же физическом месте, чтобы улучшить отношения и обмен знаниями.

Больше или меньше LeSS

Что делать дальше

Но это еще не конец. Это даже не начало конца, а, возможно, конец начала.
— Уинстон Черчилль

Поздравляем вас с тем, что вы наконец-то добрались до конца начала. Что же дальше? Надеемся, это будет практический опыт, подкрепляемый идеями, почерпнутыми из данной книги. Не забывайте экспериментировать. В двух других написанных нами книгах — *Scaling Lean & Agile Development* и *Practices for Scaling Lean & Agile Development* — можно найти немало примеров тех экспериментов, которые стоит опробовать на практике.

В процессе написания данной книги мы организовали трехдневные курсы для подготовки сертифицированных специалистов, практикующих LeSS. Эти курсы охватывают материал данной книги, а также дополнительные эксперименты, истории, прецеденты использования и примеры. Они проводятся сертифицированными инструкторами, имеющими практический опыт внедрения LeSS. Кроме того, каждый год проводится конференция для обмена опытом работы в LeSS.

Мы постоянно ведем и обновляем веб-сайт, посвященный LeSS и доступный по адресу <http://less.works>. На этом сайте вы можете проверить, насколько верно вы разбираетесь в LeSS, пройдя тест в диалоговом режиме на странице по адресу <http://less.works/less/test/index.html>. Мы постоянно дополняем данный веб-сайт новым содержимым, видеоматериалами, прецедентами использования и прочими учебными материалами. На этом сайте представлено также постоянно растущее собрание отчетов об опыте внедрения LeSS. Каждый такой отчет содержит интересные знания и идеи, которые стоит изучить и опробовать.

Если вы желаете описать свой опыт внедрения LeSS и поделиться им, непременно дайте нам знать! Мы постоянно ищем опыт, накопленный другими, чтобы самим учиться и опробовать новые идеи. Если вы имеете отношение к существующему прецеденту использования и являетесь сертифицированным специалистом,

практикующим LeSS, можете поделиться своим опытом на упомянутом выше веб-сайте, составив соответствующий отчет. Ведь накопленный опыт открывает широкие перспективы для того, чтобы всегда учиться.

Принимая во внимание упомянутые выше ресурсы для обучения, мы искренне надеемся, что вы получите максимум удовольствия от создания продуктов, оказывающих заметное влияние на то, чтобы добиваться большего средствами LeSS.



Рекомендованная литература

Мы решили не включать в данную книгу обширную библиографию. Вместо этого ниже приводится перечень всей рекомендованной литературы, упоминавшейся в книге. В двух предыдущих наших книгах — *Scaling Lean & Agile Development* и *Practices for Scaling Lean & Agile Development* — можно найти более обширную библиографию по LeSS.

Adzic, G., *Specification by Example: How Successful Teams Deliver the Right Software*, Manning Publications, 2011.

Adzic, G., *Impact Mapping: Making a Big Impact with Software Products and Projects*, Provoking Thoughts, 2012.

Adzic, G., Evans, D., *Fifty Quick Ideas to Improve Your User Stories*, Neuri Consulting, 2014.

Balle, M., Balle F., *The Lean Manager: A Novel of Lean Transformation*, Lean Enterprise Institute, 2009.

Deemer, P., Benefield, G., Larman, C., Vodde, B., *The Scrum Primer*, 2012. См. по адресу <http://scrumprimer.org> и <http://less.works>.

Gall, J., *Systemantics, The Systems Bible: The Beginner's Guide to Systems Large and Small*, General Systemantics, 2003.

Gray, D., Brown, S., Maconufo, J., *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers*, O'Reilly Media, 2010.

Hackman, R., *Leading Teams*, Harvard Business Press, 2002.

Hamel, G., *The Future of Management*, Harvard Business Review Press, 2007.

Hohmann, L., *Innovation Games: Creating Breakthrough Products Through Collaborative Play*, Addison-Wesley, 2006.

James, M., *The Scrum Master Checklist*, 2010. См. по адресу <http://scrummasterchecklist.org>.

Kimsey-House, H., Kimsey-House, K., *Co-active Coaching: Changing Business, Transforming Lives*, Nicholas Brealey America, 2011.

Laloux, F., *Reinventing Organizations*, Nelson Parker, 2014.

- Larman, C., Vodde, B., *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, Addison-Wesley, 2008.
- Larman, C., Vodde, B., *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*, Addison-Wesley, 2010.
- Larman, C., Vodde, B., *Feature Team Primer*, 2011. См. по адресу <http://featureteams.org>.
- Larman, C., Fahmy, A., *How to Form Teams in Large-Scale Scrum? A Story of Self-Designing Teams*, 2013. См. по адресу <http://www.scrumalliance.org/community/articles/2013/2013-april/how-to-form-teams-in-large-scale-scrum-a-story-of>.
- Lawrence, R., *Patterns for Splitting User Stories*, 2009. См. по адресу <http://agileforall.com/patterns-for-splitting-user-stories/>.
- Lencioni, P., *The Five Dysfunctions of a Team: A Leadership Fable*, Jossey-Bass, 2002.
- McGregor, D., *The Human Side of Enterprise*, McGraw-Hill Education, 1960.
- Meadows, D., Wright, D. [редактор], *Thinking in Systems: A Primer*, Chelsea Green Publishing, 2008.
- Ohno, T., *Workplace Management*, Productivity Press, 1988.
- Owen, H., *Open Space Technology: A User's Guide*, Berret-Koehler Publishers, 2008.
- Patton, J., *User Story Mapping: Discover the Whole Story, Build the Right Product*, O'Reilly Media, 2014.
- Pfeffer, J., Sutton, R., *Hard Facts, Dangerous Half-Truths and Total Nonsense: Profiting from Evidence-Based Management*, Harvard Business Review Press, 2006.
- Schein, E., *Humble Inquiry: The Gentle Art of Asking Instead of Telling*, Berret-Koehler Publishers, 2013.
- Schwaber, K., *The Scrum Guide*, 2013. См. по адресу <http://scrumguides.org>.
- Schwarz, R., *The Skilled Facilitator: A Comprehensive Resource for Consultants, Facilitators, Managers, Trainers, and Coaches*, Jossey-Bass, 2002.
- Senge, P., *The Fifth Discipline: The Art & Practice of the Learning Organization*, Doubleday, 2006.
- Vodde, B., *Specialization and Generalization in Teams*, 2011. См. по адресу <http://www.scrumalliance.org/community/articles/2011/january/specialization-and-generalization-in-teams>.
- Веб-сайт, посвященный LeSS, см. по адресу <https://less.works>.

Б

Правила LeSS

Правила LeSS определяют инфраструктуру LeSS, поэтому их обязательно следует знать. Причины этого поясняются на веб-странице “Why LeSS” (Почему LeSS), доступной по адресу <http://less.works/less/framework/why-less.html>.

Правила инфраструктуры LeSS

Инфраструктура LeSS применяется к продуктам, над которыми работают от двух до восьми команд.

Структура LeSS

Структура LeSS подчиняется следующим правилам.

- > Структура организации опирается на использование настоящих команд в качестве основных конструктивных блоков.
- > Большинство команд предназначены для функциональных средств и сосредоточены на клиентах.
- > Scrum-мастера отвечают за работоспособное внедрение LeSS. Они уделяют основное внимание Командам, Владельцу Продукта, организации и нормам практики разработки. Scrum-мастер сосредоточивает свое внимание не только на одной команде, но и на всей организационной структуре в целом.
- > Scrum-мастер уделяет своей роли все свое рабочее время.
- > Один Scrum-мастер может обслуживать 1–3 команды.
- > Руководители в LeSS не обязательны, но если они все же имеются, то их роль, вероятнее всего, изменится. Основное их внимание переходит с повседневного руководства производством продукции к расширению возможностей системы разработки продукции доставлять ценность.
- > Роль руководителей состоит в том, чтобы совершенствовать систему разработки продукции, практикуя посещение

рабочих мест, поощряя оперативное устранение недостатков или ошибок и “эксперименты превыше соответствия”.

- > В группе продукта с самого начала устанавливается полноценная структура LeSS, что очень важно для внедрения LeSS.
- > В крупной организации внедрение LeSS за пределами группы продукта происходит эволюционным путем регулярного посещения рабочих мест и создания такой организации, где экспериментирование и совершенствование является нормой.

Продукт в LeSS

Создание продукта в LeSS подчиняется следующим правилам.

- > Для организации выпуска полноценного продукта имеется Владелец Продукта и единый Задел Продукта.
- > Владелец Продукта не должен работать в одиночку над уточнением Задела Продукта. Ему оказывают посильную помощь многие Команды, работающие непосредственно с потребителями или пользователями и прочими заинтересованными сторонами.
- > Все назначение приоритетов проходит через Владельца Продукта, но прояснение осуществляется в как можно большей степени непосредственно Командами, потребителями или пользователями и прочими заинтересованными сторонами.
- > С практической точки зрения определение продукта должно быть как можно более широким и сосредоточенным на клиентах. А со временем определение продукта может быть расширено. Предпочтение следует отдавать более широким определениям продуктов.
- > Для всего продукта в целом имеется единый Критерий Готовности, который является общим для всех команд.
- > У каждой команды может быть свой более строгий Критерий Готовности, расширяющий общий критерий.
- > Цель совершенства — улучшить Критерий Готовности таким образом, чтобы получать готовый к поставке продукт в каждом Спринте (или еще чаще).

Спринт в LeSS

Проведение Спринта в LeSS подчиняется следующим правилам.

- > На уровне продукта проводится один Спринт, а не разные Спринты для каждой Команды. Каждая Команда начинает и завершает Спринт одновременно. В результате каждого Спринта получается интегрированный целый продукт.

- > Планирование Спринта разделяется на следующие две части: Первое Планирование Спринта, которое обычно проводится с участием всех команд, тогда как Второе Планирование Спринта — как правило, в каждой команде в отдельности. Многокомандное Второе Планирование Спринта проводится в общем пространстве для подготовки тесно связанных вместе элементов.
- > Первое Планирование Спринта посещают Владелец Продукта и Команды или их представители. Совместно они предварительно выбирают элементы, над которыми предстоит работать каждой команде в течение Спринта. Команды выявляют возможности для совместной работы и проясняют итоговые вопросы.
- > У каждой Команды имеется свой Задел Спринта.
- > Второе Планирование Спринта проводится для того, чтобы Команды решили, что им делать с выбранными элементами. Это, как правило, подразумевает проектирование и создание командами их собственных Заделов Продукта.
- > Каждая Команда проводит свою Ежедневную Летучку.
- > Команды сами координируют свои действия. Предпочтение отдается децентрализованному и неформальному координированию действий над централизованным. Особое внимание уделяется методу “поговорить” и неформальным сетям поддержания связи через код, межкомандным совещаниям, кураторам компонентов, спутникам, разведчикам и открытым пространствам.
- > Уточнение Задела Продукта проводится каждой командой для анализа тех элементов, над которыми им, скорее всего, придется работать в будущем. Многокомандное и/или общее Уточнение Задела Продукта проводится для укрепления общего понимания и исследования возможностей для координации действий при наличии тесно связанных элементов или потребности в более широком участии и обучении.
- > Подведение Итогов Спринта должно быть общим для всех команд. К этому мероприятию должны привлекаться соответствующие заинтересованные стороны, готовые вносить информацию, необходимую для эффективного обследования и адаптации.
- > Каждая Команда проводит свою Ретроспективу Спринта.
- > Общая Ретроспектива проводится после ретроспектив в отдельных Командах с целью обсудить насущные вопросы как среди команд, так и на уровне системы, а также подготовить эксперименты по усовершенствованию. Это мероприятие посещают Владелец Продукта, Scrum-мастера, представители Команд и руководители, если таковые имеются.

Правила инфраструктуры LeSS Huge

Инфраструктура LeSS Huge применяется к продуктам, над которыми работают больше восьми команд. Ее не следует применять в более мелких группах продуктов, поскольку соблюдение ее принципов потребует больших издержек и локальной оптимизации. Все правила LeSS распространяются и на LeSS Huge, если не указано иное. А каждая Область Требований действует подобно основной инфраструктуре LeSS.

Структура LeSS Huge

Структура LeSS Huge подчиняется следующим правилам.

- > Требования, которые заказчик считает тесно связанными, группируются в Областях Требований.
- > Каждая Команда специализируется в одной Области Требований. Команды остаются в одной области на долгое время. Когда в других областях появляется большая ценность, команды могут изменить Область Требований.
- > В каждой Области Требований имеется один Владелец Продукта в данной Области.
- > В каждой Области Требований действует от четырех до восьми команд. Эти пределы нельзя нарушать.
- > Внедрение LeSS Huge, включая структурные изменения, осуществляется постепенным эволюционным путем.
- > Следует постоянно помнить, что для внедрения LeSS Huge требуются месяцы, а то и годы, бесконечное терпение и чувство юмора.

Продукт в LeSS Huge

Создание продукта в LeSS Huge подчиняется следующим правилам.

- > В каждой Области Требований имеется свой Владелец Продукта в данной Области.
- > Один (общий) Владелец Продукта отвечает за назначение приоритетов на уровне продукта, а также за распределение команд по отдельным Областям Требований. Он тесно сотрудничает с Владельцами Продукта в отдельных Областях.
- > Владельцы Продукта в Областях выполняют роль Владельцев Продукта для своих команд.
- > Имеется единый Задел Продукта. Каждый его элемент относится к конкретной Области Требований.
- > В каждой Области Требований имеется свой Задел Продукта в данной Области. Принципиально он служит более мелким представлением единого Задела Продукта.

Спринт в LeSS Huge

Проведение Спринта в LeSS Huge подчиняется следующим правилам.

- > На уровне продукта проводится один Спринт, а не разные Спринты в каждой Области Требований. В результате каждого Спринта получается интегрированный целый продукт.
- > Общий Владелец Продукта и Владельцы Продукта в отдельных Областях часто согласовывают свои действия. Перед Планированием Спринта они принимают меры, чтобы Команды работали над самыми цennыми элементами. После Подведения Итогов Спринта они принимают дополнительные меры для адаптации на уровне продукта.

Предметный указатель

L

Less

перечень рекомендованной литературы 347

LeSS

- внедрение
 - выбор наставника или инструктора 75
 - главные принципы 71
 - добровольное участие 73
 - наращивание, рекомендации 86
 - особенности 70
 - поддержка руководства 72
 - порядок действий 75
 - с ненастоящим Владельца Продукта 187
- главные принципы 28
- другие источники для изучения 25
- количественные показатели, применение 145
- многоместная работа, организация 115
- наглядное представление 27
- назначение 20
- общие элементы 31
- организационная структура, особенности 112
- основные элементы 31
- особенности методики 24
- правила, перечень 349
- предпосылки 25
- примеры внедрения 25
- составные части 27
- структура организаций на основе команд 95
- что делать дальше 345

LeSS Huge

владельцы продукта в областях, роль 53

внедрение

подходы 87

поочередное создание Областей Требований 88

построении параллельной организации 88

эволюционным путем 87

общие элементы 31

организационная структура, особенности 123

основные элементы 55

правила, перечень 352

разделение на Области Требований 52

S

Scrum

внедрение, особенности 69

главный принцип 20

достиоинства и недостатки 69

как понятие 167

назначение 69

нормы практики 20

основные роли 23

особенности 19

поддерживающий стиль управления 129

прозрачность 23

управление эмпирическим процессом 23

Scrum-мастер

инструментальные средства 154

основные обязанности 149

поддержка взаимоотношений 153

помощь командам владельцев продукта 207

работа с сообществами LeSS 157

Scrum-мастеров 158

рекомендации по выживанию 159

- рекомендованная литература 161
 роль
 в LeSS Huge, особенности 163
 в LeSS, особенности 150
 в Scrum, особенности 149
 содействие
 в проведении совещаний,
 методы 155
 обучению 157
 сосредоточение основного
 внимания 151
 сферы деятельности
 основные 150
 требующие особого внимания 162
- Б**
- Базар при подведении итогов
 действия по окончании 333
 многоместный, проведение 333
 назначение 45, 332
 проведение 45, 332
 Большие требования
 обращение
 в LeSS 238
 традиционное 237
- В**
- Владелец продукта
 взаимоотношения
 с Scrum-мастерами 199
 с владельцами продукта в
 областях 206
 с высшим руководством 197
 с командами 193
 с потребителями и
 пользователями 195
 в Области
 поиск кандидатуры 206
 полномочия и обязанности 206
 контроль, осуществление 202
 непреклонность 201
 обязанности
 излишние 191
- основные 190
 поручаемые 190
 полномочия и обязанности 187
 помощники 191
 посещаемые совещания 203
 решение о выпуске в каждом
 Спринте, основание 201
 роль
 в LeSS 183
 в LeSS Huge 205
 в Scrum 183
 информационные потоки 190
 критерии подбора кандидата 184
 Выпускные спринты, назначение 252
- Г**
- Гемба
 определение 136
 посещение руководителями 140
 разновидности 136
- З**
- Зависимости
 внутренние, исключение 212
 как ограничения
 пересмотр, принципы 213
 устранение, рекомендации 214
 Задел
 продукта
 внесение изменений 215
 в области, назначение 53, 230
 единственный, общий 211
 инструментальные средства для
 ведения 224
 многокомандное уточнение,
 проведение 41
 назначение 226
 независимость элементов 219
 обновление 51
 обращение со специальными
 элементами 221
 общее уточнение, проведение 41, 48
 однокомандное уточнение,

- проведение 42, 49
определение 211
соблюдаемые принципы 262
сохранение родительского элемента 220
удаление родительского элемента 219
уточнение по кругу, проведение 50
спрингта
ведение без программных средств 295
инструментальные средства для ведения 226
назначение 226
- Законы**
двух футов, соблюдение 321
организационного поведения Лармана 80
- И**
- Игра в контракты
последствия 200
традиционная модель 184
- Интеграция
в LeSS
силами команд 304
соблюдаемые принципы 302
в Scrum, поддержание 301
непрерывная
в командах 40
назначение 37, 309
особенности 310
непрерывно, принцип 309
- Истории
в LeSS
поток
команд, назначение и реализация 33, 46
элементов, назначение и реализация 47, 51
в LeSS Huge
многоместные команды 63, 66
- новая область требований 56, 62
- К**
- Команды
быстрого реагирования, назначение 222
ведущие
координирование действий 324
обязанности 324
роль 240
взаимоотношения с потребителями и пользователями 196
- владельцев продукта
организация 124
помощь со стороны Scrum-мастера 207
формирование 204
в областях функциональных средств, роль 53
- для компонентов
зависимости, неразрешимые 104
модель организации 99
недостатки 100
преимущества 99
- для функциональных средств
зависимости, разрешимые 104
карты внедрения
назначение 106
помощь в принятии решений 110
применение 107
модель организации 101
назначение 97
недостатки 103
переход
в LeSS Huge, стратегии 121
в LeSS, стратегии 120
преимущества 98, 101
примеры организации 107
специализация в предметной области клиентов 110
- многоместные, разновидности и роли 63

самоорганизующиеся,
формирование 74
требования при формировании 78

Концепция совершенства
бережливого, классическая 82
в LeSS 82
организационная, выработка 82

Координирование действий
в LeSS
 силами команд 304
 соблюдаемые принципы 302

в Scrum, поддержание 301

децентрализованное и
 централизованное, отличия 38,
 305

мероприятия 39

метод ‘поговорить’ 303

поддержание связи через код 308

по линиям интеграции 308

потребность 38

рекомендации 304

сочетание разных методов 325

технические условия, удобные 308

удобная среда, особенности 304

физические условия, удобные 307

Критерий Готовности
в LeSS
 идеальный, достижение 246
 определение 247

в Scrum
 идеальный, достижение 245
 определение 245

порядок составления 246

расширение
 проводимые мероприятия 257
 с точки зрения разных ролей 255

соблюдаемые принципы 246

строгий и нестрогий,
 особенности 77, 248

Кураторы компонентов
 обязанности 320
 роль 320

Л

Локальная оптимизация
на основе неоспоримых
допущений 84

функциональная 84

Н

Надкусывание
назначение и метод 49, 217
причины 218
проведение 49

Незавершенная работа
определение 115
отделы незавершенных работ
 назначение 115, 253
 упразднение 115, 253
 передача по конвейеру 254

Незавершенная Работа
меры борьбы 252
накапливание 250
определение 249

Непрерывное совершенствование,
рекомендации 84

О

Области требований
владельцы продукта, роль 53
динамика развития 119
заделы продукта
 как отдельные артефакты,
 реализация 231
 как фильтрованные
 представления, реализация 230
назначение 117, 229
определение 117
основные свойства 52
поочередное создание 88
составные элементы 117
структуря 118

Области Требований
очень больших
 методы ведения 241

- назначение 236
- Общая работа**
- возможности для выполнения 295
 - определение 295
- Определение продукта**
- в LeSS, более широкое, преимущества 169
 - накладываемые ограничения 171
 - порядок действий 77
 - примеры 175, 178
 - расширение 179
 - расширяющие и ограничивающие факторы 174
 - соблюдаемые принципы 168
 - узкое, недостатки 170
- Организационные термины**, перечень 16
- Организация по потребительской ценности**
- в LeSS Huge, особенности 116
 - в LeSS, основные принципы 94
 - в Scrum, основные роли 93
- Оценивание в LeSS**
- единицы оценивания, согласование 285
 - рекомендуемый порядок 284
- Оценка воздействия, методика** 228
- П**
- Планирование спрингта**
- в LeSS
 - соблюдаемые принципы 289
 - схема 293
 - в LeSS Huge, проведение 296
 - в Scrum, решаемые вопросы 289
 - второе
 - многокомандное, проведение 36, 294
 - однокомандное, проведение 36
 - пример проведения 294
 - многоместное, проведение 293
 - первое
 - выбор элементов 291
- пример проведения 292
- проводение 33, 290
- Планирование Спрингта в LeSS**
- участники и место проведения 291
- Подведение итогов спрингта**
- в Less
 - проводение 331
- в LeSS**
- соблюдаемые принципы 329
- в LeSS Huge, проведение** 340
- в Scrum, назначение** 329
- в командах, проведение** 46
- общее, проведение** 44
- Поддержание связи через код** 37
- Пользователи, характеристики** 188
- Потребители, характеристики** 188
- Правила**
- инфраструктуры
 - LeSS Huge, перечень 352
 - LeSS, перечень 349
- Р**
- Разведчики, функции** 39, 323
- Разделение элементов**
- литература, рекомендованная для изучения 275
 - на тонкие комплексные элементы 281
 - перспективы 276
 - причины 275
 - с помощью перспектив, пример 276
 - степень мелкости 283
 - частичное 278
- Разработка, разновидности** 186
- Ретроспектива**
- в LeSS, проведение 331
 - многокомандная, проведение 335
 - общая
 - в LeSS Huge, проведение 340
 - в LeSS, соблюдаемые принципы 329

- многоместная, проведение 335
проводение 38
рекомендации по проведению 334
системное моделирование 336
участники 334
спринта
в Scrum, назначение 329
проводение 38
- Руководители**
- в LeSS
 - в качестве учителей и учащихся 142
 - необязательность 135
 - обязанности 136
 - посещение рабочих мест 140
 - роль 131
 - в Scrum, роль 130
 - пустяковые, явление и меры борьбы 135
 - разновидности 142
- С**
- Системное моделирование
назначение системной модели 336
построение системной модели 339
рекомендованная литература 339
элементы языка 338
- Смешанное спаривание, методика 73
- Совещания**
- в LeSS, посещаемые владельцем продукта 203
 - в открытом пространстве, проведение 321
 - крупные, проведение 315
 - межкомандные
 - назначение 314
 - разновидности 315
 - многоместные, проведение 316
 - по проектированию,
 - многокомандные, проведение 37, 317
- по текущей архитектуре, рабочие, проведение 319
- схватка над схватками, потребность в проведении 323
- Сообщества**
- добровольное участие 157, 311
 - ложные 314
 - на уровне организации 314
 - определение 311
 - отличительные качества 312
 - разновидности и организация 39
 - рекомендуемые, характеристики 313
 - цели и полномочия 312
- Спецификация по образцу, метод 268
- Спутники**
- роль 303
 - функции 322
- Сфера деятельности**
- Scrum-мастера 150
 - основные 137
 - перекрывающиеся 137
- У**
- Улучшения
дополнительные 257
разновидности 255
- Управление**
- в LeSS, основные принципы 130
 - в Scrum, поддерживающий стиль 130
 - продуктами
 - в LeSS как продуктами, преимущества 180
 - как проектами, недостатки 179
 - рекомендованная литература 145
 - теории
 - X и Y, особенности 133
 - Тейлора и Файоля, достоинства и недостатки 132
 - Утвержденное обучение, определение 257

- Уточнение задела продукта
в LeSS
 проведение 263
 разновидности совещаний 263
в LeSS HuGe, проведение 286
в Scrum, проведение 261
многокомандное
 причины для проведения 267
 проводение и методы 266
многоместное, проведение
 и методы 268
- общее, проведение и
мероприятия 265
- первоначальное
 основные предпосылки 271
 порядок проведения 271
 преследуемые цели 271
 причины для проведения 269