

Экономика программной инженерии

Барышникова Марина Юрьевна
МГТУ им. Н.Э. Баумана

baryshnikovam@mail.ru

Лекция_11. Вместо заключения

Способы повышения эффективности разработки программного обеспечения или о чем говорит актуальная статистика реализации программных проектов. Информация к размышлению

Рекомендации компании Standish Group

- Посчитайте то, что поддается счету
- Улучшайте клиента и пользователя
- Снижайте затраты
- Улучшайте среду

Источник: *CHAOS Report*



Основные риски программных проектов

1. Неясные и неточные цели проекта (степень важности близка к 100%);
2. Неадекватные сроки реализации проекта и неадекватный бюджет (степень важности порядка 75%);
3. Непродуманные и плохо отлаженные коммуникации между заинтересованными лицами проекта (степень важности составляет примерно 79%);
4. Убытки из-за отсутствия коммуникаций с пользователями (степень важности порядка 83%);
5. Не отвечающие требованиям проекта знания и умения (степень важности примерно 57%);
6. Отсутствие эффективной методологии управления проектом (степень важности близка к 60%);
7. Недооценка требований проекта (степень важности оценивается в 92%);
8. «Золотое покрытие» (gold plating) – завышение аналитиком и/или менеджером требований проекта с целью сделать ПП лучше и удобнее (степень важности незначительная -12%);
9. Изменение требований в процессе разработки проекта (степень важности составляет 63%);
10. Ошибки (bag), допускаемые в процессе разработки (степень важности близка к 90%);
11. Субподрядчики;
12. Низкая производительность проектной команды;
13. Применение новых технологий в проекте;
14. Некачественное управление ожиданиями клиентов

Источник: Addison T., Vallabh S. Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers



Оценка успешности IT-проектов на основе традиционного подхода (OnTime, OnBudget, OnTarget)

	2011	2012	2013	2014	2015
успешные	39%	37%	41%	36%	36%
провальные	22%	17%	19%	17%	19%
спорные	39%	46%	40%	47%	45%

Успешные проекты – все цели проекта достигнуты в плановый срок и бюджет

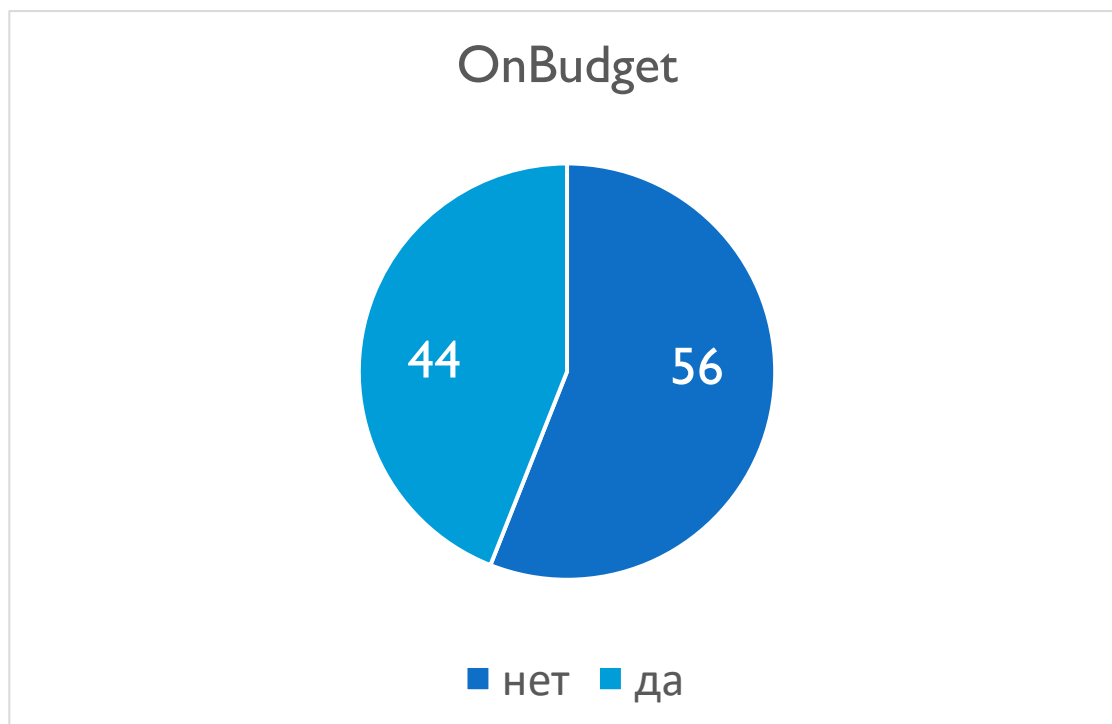
Провальные проекты– проекты, остановленные без получения результата

Спорные проекты – те проекты, которые были завершены либо с превышением сроков, либо стоили дороже, чем планировалось, либо достигли лишь части целей

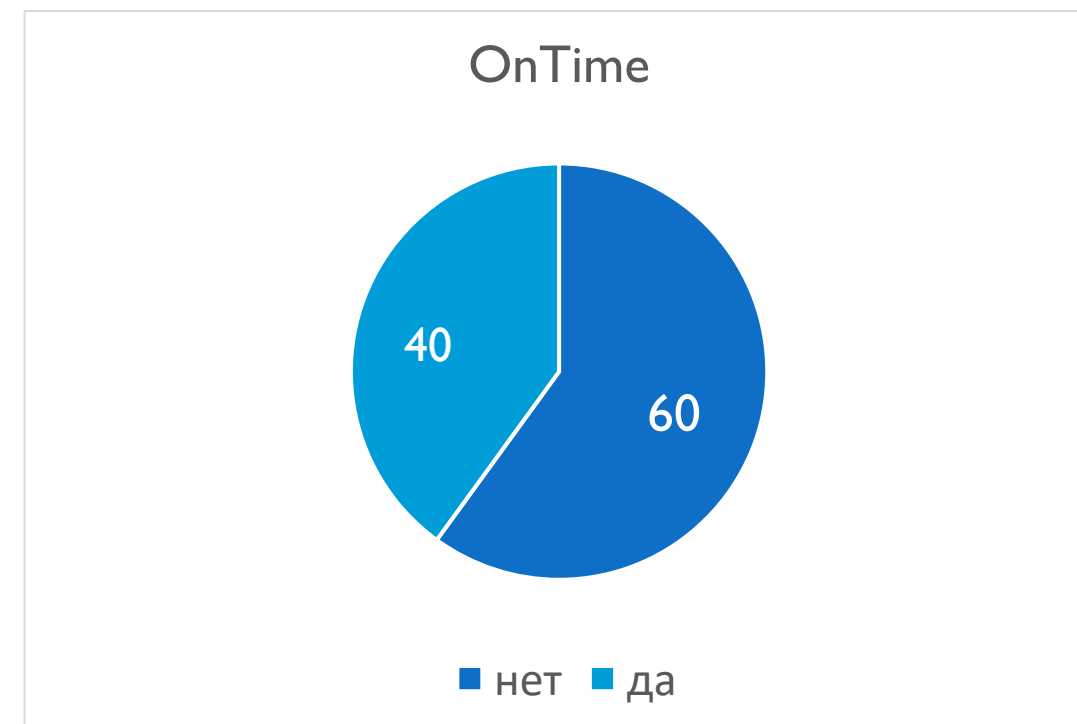
Примечание: Общее количество проанализированных программных проектов за 2011-2015 более 25 000, в среднем 5 000 в год



Декомпозиция результатов IT-проектов, реализованных в 2011 – 2015 гг.



Процент проектов, которые уложились в бюджет



Процент проектов, которые были выполнены в срок



Оценка успешности IT-проектов на основе современного подхода (OnTime, OnBudget и удовлетворенность клиентов)

	2011	2012	2013	2014	2015
успешные	29%	27%	31%	28%	29%
провальные	22%	17%	19%	17%	19%
спорные	49%	56%	50%	55%	52%

Степень удовлетворения клиента существенно выше, когда реализованные в рамках проекта функции не просто автоматически соответствуют запланированным, а отвечают именно его реальным потребностям. Автоматическое следование ТЗ часто приводит к тому, что реализованные функции программного обеспечения не используются, но увеличивают стоимость и степень риска проекта и сказываются на качестве: масштаб проекта растет, но не обязательно обеспечивается ценность для пользователя

Примечание: Произошло перераспределение между успешными и спорными проектами. В среднем с 2007 по 2016 годы показатель успешных проектов снизился на 8%, а показатель спорных – увеличился на 6%.



Оценка степени влияния размера проекта на его результат

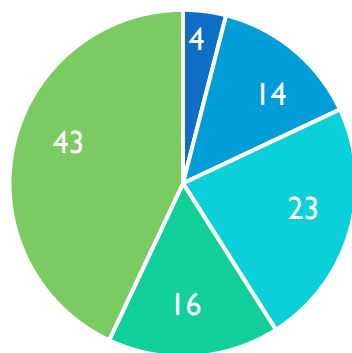
Результат Размер	успешный	спорный	провальный
Очень большой	6%	51%	43%
Большой	11%	59%	30%
Средний	12%	62%	26%
Умеренный	24%	64%	12%
Маленький	61%	32%	7%

Примечание: Т.е. только 6% из всех очень больших проектов оказались успешными, тогда как для маленьких проектов процент успеха составил 61%



Финансовая отдача от реализации программных проектов

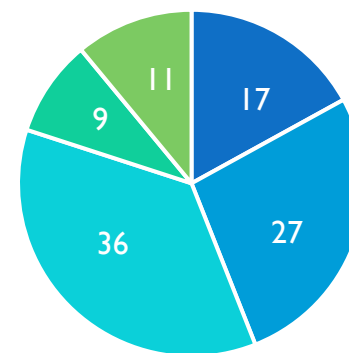
Ценность ("отдача") для организации от реализации больших проектов



- Очень высокая ценность
- Высокая ценность
- Средняя
- Низкая
- Очень низкая

Примечание: Т.е. практически 60% больших проектов не принесли ценности той организации, которая их реализовывала

Ценность ("отдача") для организации от реализации маленьких проектов



- Очень высокая ценность
- Высокая ценность
- Средняя
- Низкая
- Очень низкая

Примечание: Только 20% маленьких проектов не приносят ценности той организации, которая их реализует



Выводы и рекомендации

- ▶ Чем крупнее проект, тем не только меньше вероятность достижения успеха, но и тем скромнее его норма доходности (норма прибыли)
 - ▶ Во многих случаях крупные проекты никогда не окупают затраты организации, которая их реализует
 - ▶ Чем быстрее проекты идут в производство, тем быстрее они начинают окупаться
-
- ▶ Целесообразно разбивать большой проект по созданию программного обеспечения на несколько небольших проектов, с ранней доставкой результатов до потребителя, с целью более быстрой финансовой отдачи и большей удовлетворенности пользователей
 - ▶ Большинство программных проектов должно быть реализовано силами небольшой команды, которая привлекается на короткий срок для того, чтобы сформировать ценность для организации

Только в очень редких случаях проекты должны быть большими и длинными



Распределение программных проектов по отраслям экономики

Отрасли экономики	успешные	спорные	провальные
Банковское дело	30%	55%	15%
Финансы	29%	56%	15%
Государственные контракты	21%	55%	24%
Здравоохранение	29%	53%	18%
Производство	28%	53%	19%
Торговля	35%	49%	16%
Сервис / Сфера услуг	29%	52%	19%
Телекоммуникации	24%	53%	23%
Другие	29%	48%	23%

Примечание: Наиболее успешными были проекты в сфере розничной торговли (Retail): их показатель успеха 35%. Самый высокий уровень неудач (24%) показали проекты, реализуемые в рамках государственных заказов. Результаты проектов в финансовой сфере оказались наиболее спорными (56%)



Уровень удовлетворенности клиентов результатами программных проектов



Уровень удовлетворенности клиентов от реализации проектов в сфере розничной торговли



Уровень удовлетворенности клиентов от реализации проектов в финансовой сфере



Географическое распределение результатов программных проектов

География реализации проекта	успешные	спорные	провальные
Северная Америка	31%	51%	18%
Европа	25%	56%	19%
Азия	22%	58%	20%
Остальной мир	24%	55%	21%

Примечание: Одной из причин успешности американских проектов является наличие у менеджеров проектных команд эмоциональной зрелости – т.е. навыков, ориентированных на управление ожиданиями клиентов и достижение консенсуса всех заинтересованных сторон, что приводит к высокому уровню удовлетворенности



Анализ зависимости результата от типа проекта

Тип проекта	успешные	спорные	провальные
Разработка с нуля с использованием традиционных языков и методов	22%	61%	17%
Разработка с нуля с использованием современных методов	23%	54%	23%
Разработка с использованием готовых и/или покупных компонентов	24%	59%	17%
Приложение собрано из покупных компонентов	25%	59%	16%
Модификация покупных приложений	42%	37%	21%
Покупка и использование готовых приложений (коробочные продукты)	57%	28%	15%
Модернизация	53%	38%	9%
Другие	28%	47%	25%

Примечание: Самый высокий показатель успешности был у коробочных продуктов (57%). Проекты, которые разрабатывались с нуля с использованием традиционных языков и методов, показали самый высокий уровень спорности результатов – 61%.



Влияние метода разработки на результат проекта

Тип проекта	метод	успешные	спорные	провальные
Проекты любого размера	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Проекты большого размера	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Проекты среднего размера	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Проекты маленького размера	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Примечания: Agile проекты оказались почти в четыре раза более успешными, чем проекты, реализуемые по методу водопада. Водопадные проекты не очень хорошо масштабируются в отличие от гибких. Для маленьких проектов характерна небольшая разница в уровне успешности между гибким и водопадным процессами.

Вывод: Самое выгодное сочетание – это гибкий процесс и небольшие проекты, они имеют только 4% неудач



Соответствие цели или удовлетворенность потребителя?

Степень соответствия проекта первоначальной цели	успешные	спорные	провальные
Точное	22%	53%	25%
Примерное	23%	54%	23%
Свободное	27%	52%	21%
Расплывчатое (неопределенное)	38%	46%	16%
Очень отдаленное	34%	58%	8%

Основные выводы: Бизнес-цели практически любого проекта динамичны, в том числе, их может сильно изменить и ход его реализации. Многие из наиболее успешных проектов на старте имели высокий уровень неопределенности.

Слишком жесткое следование изначальной стратегии может препятствовать достижению высоких показателей удовлетворенности клиентов и, как следствие, - успеху. Проектные группы должны сбалансировать усилия по контролю бизнес-целей, чтобы поощрять продвижение инноваций



Влияние уровня компетентности персонала на результаты проектов

Качество персонала в команде проекта	успешные	спорные	провальные
Одаренный	38%	45%	17%
Талантливый	31%	53%	16%
Компетентный/квалифицированный	28%	53%	19%
Способный	24%	54%	22%
Неподготовленный/низкой квалификации	17%	60%	23%

Пять основных принципов обеспечения компетентности персонала:

- определить необходимые компетенции и альтернативные навыки
- обеспечить хорошую, непрерывную программу обучения персонала
- нанимать как внутренних, так и внешних сотрудников для обеспечения баланса опыта
- обеспечить стимулы для мотивации персонала
- убедиться, что персонал ориентирован на проект

Вывод: Когда проект обеспечен людьми, имеющими соответствующий уровень профессиональной компетентности и обладающими навыками командной работы, он может рассчитывать на успех даже в самых тяжелых условиях реализации



Факторы, влияющие на успех проекта

Фактор	Баллы
Заинтересованность менеджера проекта и поддержка руководства организации	15
Вовлеченность пользователя в процесс разработки	15
Оптимизация, т.е. декомпозиция большого проекта на несколько малых	15
Эмоциональная зрелость: управление ожиданиями и достижение консенсуса	15
Улучшение профессиональных навыков	10
Agile процессы, т.е. отказ от классической разработки (waterfall) в пользу гибких методологий реализации проекта	7
Ясные цели проекта	4
Готовность привлекать внешнюю экспертизу	6
Опыт управления проектами	5
Инструменты и инфраструктура проекта	8



Рекомендации по повышению эффективности проектов

- ▶ Создание документа, в котором заказчиком коротко и ясно формулируются цели и задачи проекта, описываются пользователи и предполагаемый продукт
- ▶ Привлечение независимых экспертов для идентификации и оценки рисков
- ▶ Декомпозиция большого проекта на несколько малых проектов, т.е. создание портфеля проектов (project portfolio) с назначением менеджера портфеля, отвечающего за успешное завершение данных проектов
- ▶ Наличие предварительных грубых оценок: погрешность оценки срока реализации ИТ-проекта и бюджета на этапе инициации не должна превышать 50%, а на этапе планирования - 10–15%
- ▶ Применение современных гибких методологий в процессе разработки ИТ-проекта
- ▶ Создание фокус-групп из потенциальных пользователей с целью тестирования инкрементов продукта в процессе реализации ИТ-проекта



Отчёт Standish Group «Chaos Report-2020» – За гранью бесконечности

В 2020 году, накопив информацию за 25 лет по более чем 50 000 проектов в своей базе данных, компания Standish Group пришла к следующим выводам. Для достижения успеха в разработке программного обеспечения вы должны:

- ▶ Прекратить использовать Waterfall и начать использовать Agile
- ▶ Прекратить использовать проекты
- ▶ Прекратить использовать менеджеров проектов
- ▶ Прекратить использовать инструменты управления проектами

«Пришло время перестать заниматься «проектами» по разработке и внедрению программного обеспечения – и начать просто разрабатывать и внедрять программное обеспечение. Мы знаем, что собираемся потратить деньги на разработку и внедрение программного обеспечения. Это данность. Что нам нужно прекратить делать, так это искусственно разбивать программное обеспечение на «проекты» со всеми их накладными расходами и пустой тратой времени и денег»

Standish Group Chaos Report (2020) Beyond Infinity



Почему?

Программное обеспечение следует рассматривать как продукт с полным жизненным циклом создания, роста и обслуживания. Зачем для его разработки использовать проект, который по определению является временным и уникальным?

Недостатком использования проектов является то, что они создают большие продукты. Даже если большой проект разбивается на части, все равно остаются пакеты требований с планом выполнения этой работы в рамках определённого бюджета и сроков. Разработка ведётся в строгое соответствие с планом, в конце этот пакет новых требований запускается в производство, проходит приёмочное тестирование пользователей и только тогда разработчики узнают о функциях, которые действительно нужны пользователям

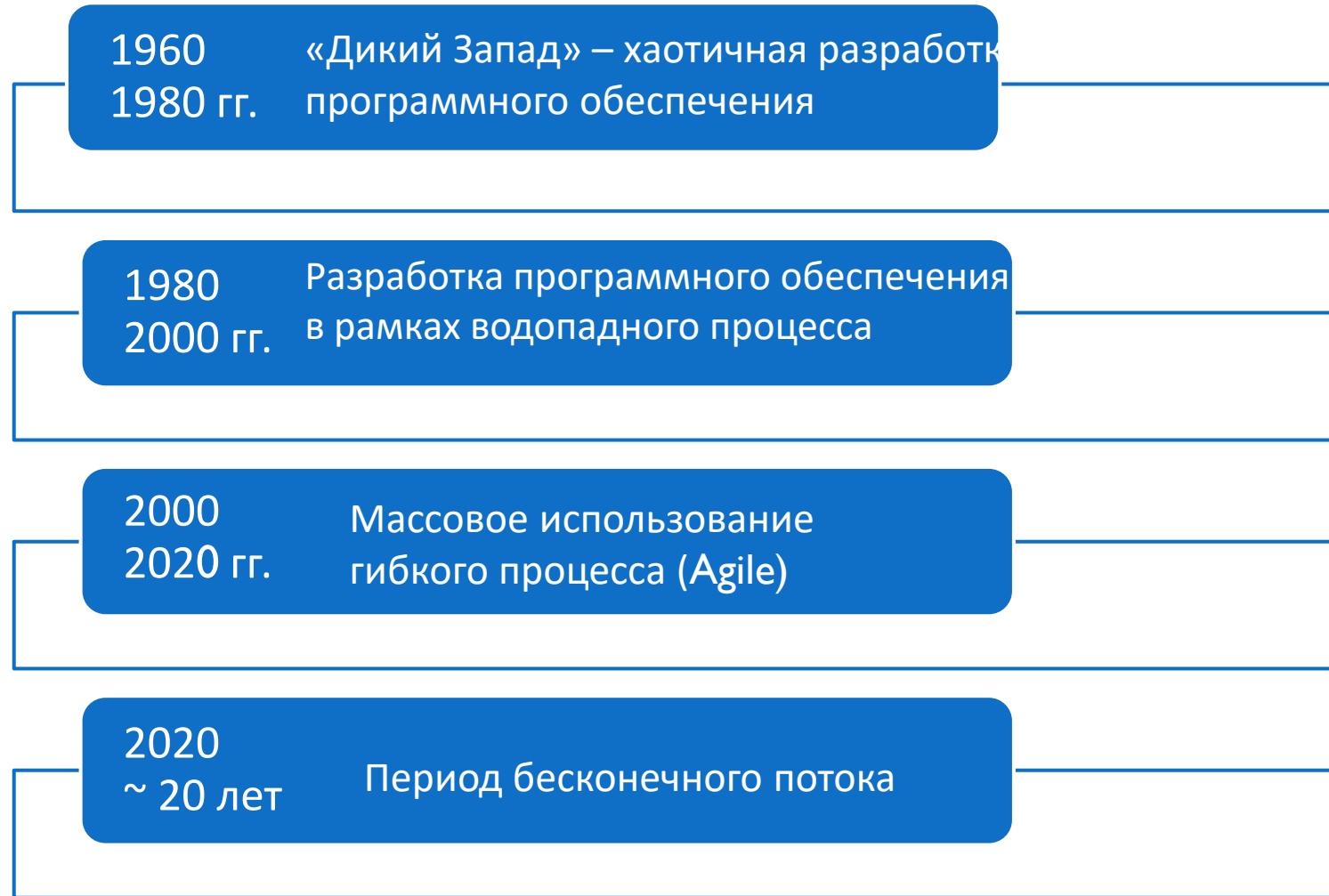
Что лучше работает в программном обеспечении, так это небольшие, постепенные улучшения, осуществляемые непрерывно. Эти небольшие улучшения легче тестируются и осваиваются конечными пользователями

«Настоящий прорыв произошёл, когда мы поняли, что программное обеспечение бесконечно, в то время как проекты конечны. Это подход к разработке программного обеспечения, который разорвёт цепи, сдерживающие наши достижения. Сделайте 2020 год концом программных проектов...»

Standish Group Chaos Report (2020) Beyond Infinity



60 лет разработки программного обеспечения



Предпосылки для перехода на конвейерную разработку

Причины неудач большинства проектов – это размер и сложность

Размер определяется в первую очередь трудозатратами. Трудозатраты определяются стоимостью нормированного труда, количеством человек и общим размером команды. Также учитывается количество функций, строк кода и другие факторы

Рекомендации по оценке размера		
Факторы для определения размера		Размер
Менее \$1 миллиона на оплату труда	6 или менее членов команды/месяц	маленький
От \$1 миллиона до \$3 миллионов	От 7 до 12 членов команды/месяц	умеренный
От \$3 миллионов до \$6 миллионов	От 13 до 24 членов команды/месяц	средний
\$6 миллионов до \$10 миллионов	От 25 до 50 членов команды/месяц	большой
Свыше \$10 миллионов	Свыше 50 членов команды/месяц	очень большой



Влияние сложности на успех проекта

Сложность часто вызвана размером, противоречивыми целями и большими бюджетами

Сложность создаёт издержки и путаницу

Для борьбы со сложностью Standish Group представила теорию поглощения, которая рассматривается как способность организации успешно воспринимать бизнес- и технические изменения без сбоев. Для этого предлагается перейти на еще более мелкие проекты – нанопроекты, так как они фокусируются на узком наборе функций и требований, которые пользователям легче понять и усвоить. Чем быстрее вы сможете внести небольшие изменения, тем они более приемлемы и тем проще в реализации

Оценка степени влияния сложности проекта на его результат			
Результат Сложность	успешный	спорный	провальный
Очень сложный	9%	60%	31%
Сложный	13%	58%	29%
Средний	27%	54%	19%
Простой	35%	46%	19%
Очень простой	34%	48%	18%



Матрица Размер-сложность

		Сложность				
		Очень простые	Простые	Средней сложности	Сложные	Очень сложные
Размер	Маленький	100	250	400	550	625
	Умеренный	175	325	475	625	775
	Средний	250	400	550	700	850
	Большой	325	475	625	775	925
	Очень большой	400	550	700	850	1000

Примечание: Матрица Размер-сложность использует 5-балльную шкалу как для размера, так и для сложности. Проект с наименьшим баллом – это простой небольшой проект, его оценка – 100 баллов. Самый крупный и сложный проект оценён в 1000 баллов. Зелёный означает низкий риск и усилия, жёлтый означает средний риск и усилия, а красный означает высокий риск и усилия



Параметры сложности

Окружающая среда (условия реализации)	Баллы
Разнообразные категории пользователей	1
Распределённая команда	1
Распределённые заказчики	1
Сложности во взаимоотношениях в команде («несговорчивые» коллеги)	2
Сложности во взаимоотношениях с заказчиками («несговорчивые» заказчики)	3
Масштаб	Баллы
Множество требований – большой масштаб	1
Неоднозначная базовая область применения	1
Нечёткие требования	1
Разнообразные и многосторонние цели	2
Высокий уровень инновационности	3

Примечание: Если проект набирает менее 3 баллов, то он будет простым. Если набирается от 4 до 7 баллов, проект будет средним, в то время как от 5 до 9 баллов проект будет сложным. Если проект набрал более 10 баллов, то он будет очень сложным



Раздел 10 «Chaos Report-2020»: «Мифы и иллюзии»

- ▶ Успешные проекты имеют высококвалифицированного менеджера проекта
- ▶ Инструменты управления проектами способствуют успеху проекта
- ▶ Все проекты должны иметь чёткие бизнес - цели
- ▶ Неполные требования приводят к оспариванию и провалу проектов

Новые факторы успеха

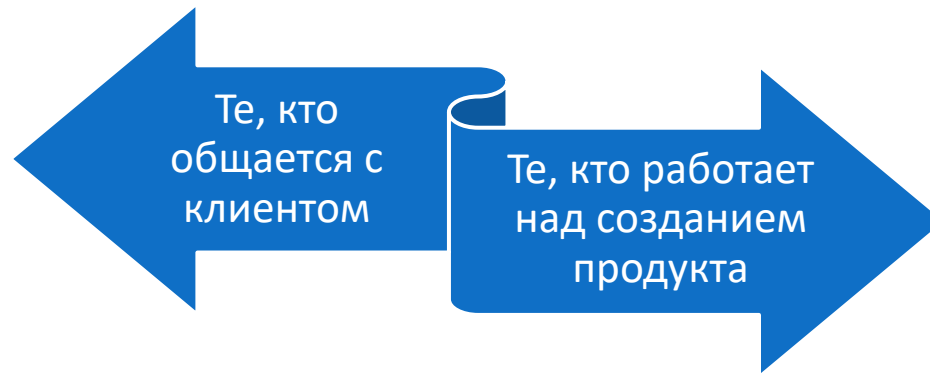
- ▶ хороший куратор
- ▶ хорошая команда
- ▶ хорошая среда

В период потока не будет бюджетов проектов, планов проектов, менеджеров проектов или Scrum-мастеров. Для потока будет предусмотрен бюджет, который представляет собой чистую прямую стоимость продукции. Текущие накладные расходы будут снижены на целых 90%, что будет достигнуто за счёт сокращения и ликвидации большинства текущих мероприятий по управлению проектами. Функциональное описание работы войдёт в конвейер и выйдет из конвейера полностью пригодным для использования. Изменения будут происходить непрерывно, но небольшими шагами, что позволит сохранить все актуальное, полезное и более приемлемое для пользователей, а не пугать их результатом «большого взрыва»



Навыки, которые нужны ИТ-специалисту

В традиционных подходах к реализации ИТ-проектов всегда существовало чёткое разграничение между людьми, которые управляют проектом, взаимодействуют с бизнесом, и разработчиками, которые просто выполняют поставленную задачу



Источник:

<https://tproger.ru/articles/10-soft-skills-for-it-specialist/>

В современных реалиях эта модель не всегда применима и очень часто не успешна

Основные причины: очень часто итоговый результат отличается от того, что на самом деле было нужно клиенту или пользователям, плюс срыв сроков и превышение бюджета

Как было сказано выше при разработке ПО все чаще используются гибкие методологии (Agile), что привело к размыванию границ между ролями и увеличению объемов коммуникации с бизнесом и внутри проектной команды



10 soft skills, которые нужны ИТ-специалисту

1. Понимание ценности для клиента	6. Переговорные навыки
2. Навыки коммуникации	7. Гибкость и креативность
3. Эмоциональный интеллект	8. Проактивность
4. Командная работа	9. Навыки презентации
5. Тайм-менеджмент	10. Менторинг

«Существует заблуждение, что гибкие навыки нужны в ИТ только тем, кто хочет стать менеджером. Нет, гибкие навыки пригодятся любому. Благодаря им, вы сможете легко доносить свои мысли до коллег, грамотно объяснять свои действия и логику. Сотруднику с хорошими «софт скиллами» доверяют интересные сложные проекты, коммуникацию с заказчиками и много дополнительных задач, которые разнообразят его рабочий день и делают работу увлекательней»



Понимание ценности для клиента

- ▶ В современном мире IT разработчику нужно понимать бизнес заказчика. Только так можно сделать по-настоящему хороший продукт
- ▶ Наличие этого навыка повышает уровень доверия между разработчиком и заказчиком. Если разработчик действительно понимает, зачем он делает то, что делает, он сможет выбрать правильный подход и решение и вовремя подскажет, что задачу можно выполнить иначе (либо не делать вообще)
- ▶ Понимание ценности для клиента позволит избежать переделок. Если разработанный продукт в итоге не решает проблему заказчика или не интересен пользователям, его придётся переделывать (клиент потратит больше денег и засомневается в уровне вашей компетентности)

Как сформировать навык: Всегда вначале думать о проблеме, а потом уже о задаче. Часто разработчик начинает выполнять задачу, не задумываясь о том, какую проблему решает. Возможно, задача не решает никакой проблемы или решает, но не ту



Навыки коммуникации

- ▶ Коммуникация - один из базовых социально-психологических навыков, который пригодится в любой сфере жизни и работы. Это умение слушать и слышать, убеждать и аргументировать, вести переговоры и организовывать работу команды, поддерживать отношения и работать в команде
- ▶ Коммуникация – один из важнейших навыков, необходимых для взаимодействия с командой проекта. Каким бы профессионалом вы себя ни ощущали, всегда найдётся задача, которую вы не в состоянии решить в одиночку
- ▶ Навыки коммуникации нужны для обмена опытом и знаниями. Если тем, что умеете вы, будут владеть и другие члены команды, то от этого выиграют все. Это работает и в обратную сторону — если в вашей команде есть человек, обладающий какими-то уникальными навыками, он может поделиться этим с вами. Это бесконечный цикл, в котором выигрывает каждый: команда, проект и, конечно же, каждый член команды в отдельности

Как сформировать навык: Будьте открытыми и идите навстречу общению. Помогайте другим и не бойтесь просить о помощи. Ну и конечно больше общайтесь! Только так можно сформировать этот навык, никакие курсы и книги тут не помогут



Эмоциональный интеллект

- ▶ Под эмоциональным интеллектом понимают эмпатию, способность видеть и распознавать скрытые эмоции людей, навык управления собственными эмоциями при принятии практических решений:
 - ▶ Первая сторона эмоционального интеллекта — это понимание чувств и эмоций других людей. Нужно учиться понимать чувства и мотивацию своих коллег, заказчиков и партнеров, уметь с ними взаимодействовать
 - ▶ Вторая сторона эмоционального интеллекта — это умение управлять своими эмоциями. Часто мы совершаем спонтанные действия под влиянием эмоций, например от злости или раздражения. Если мы понимаем свои эмоции и контролируем их, то можем чаще принимать осознанные и взвешенные решения

Как сформировать навык: Для понимания других людей учитесь навыку **активного** слушания, будьте вовлеченными в диалог, задавайте вопросы.

Старайтесь анализировать своё и чужое поведение **объективно**.

Don't be toxic — не позволяйте себе токсичного поведения. Не критикуйте и не жалуйтесь, особенно публично, потому что подобные действия деструктивны — они не несут пользы вам и окружающим, не создают ценности, а только ухудшают настроение и микроклимат в коллективе



Командная работа

- ▶ Как бы хороши вы ни были, есть задачи и проекты, с которыми вы не можете справиться в одиночку, не потратив на это кучу времени. Есть проекты, на которых задействовано от 100 до 1000 человек и даже больше. И вовсе не потому, что нужно создать много рабочих мест, а потому, что проект действительно масштабный, и с меньшим количеством участников его не реализовать
- ▶ Каждый человек обладает уникальным набором навыков, и только собрав вместе таких людей, можно получить хороший результат

Как сформировать навык: Решайте совместно с кем угодно, какие угодно задачи — будь то студенческий проект, домашние дела или что-то связанное с хобби — главное, делать это не одному



Тайм-менеджмент

- ▶ Вопреки общему заблуждению, мы не можем управлять временем. Его у нас ровно 24 часа каждый день
- ▶ Искусство грамотно управлять своим временем — один из самых важных навыков, так как помимо основных задач на проекте, у вас постоянно появляются новые. При неверном планировании работа становится непродуктивной
- ▶ Тайм-менеджмент — это не про время, а про задачи и приоритеты. О том, как управлять делами, чтобы сохранять продуктивность в периоды высокой нагрузки, избавлять себя от отвлекающих факторов и соблюдать баланс между работой и жизнью

Как сформировать навык:

- Держать все задачи в одном месте. Сегодня мы получаем информацию отовсюду — почта, мессенджер, звонки, что-то устно, что-то записано в блокноте или на стикерах, это затрудняет работу, поэтому важно держать всё в одном месте — будь то блокнот или мобильное приложение
- Разбивать большие задачи на мелкие, вплоть до таких, как «сходить», «написать», «позвонить». Человеческий мозг боится больших задач, потому что не понимает, с чего начинать, и мы начинаем откладывать задачу снова и снова, до тех пор, пока не придёт дедлайн
- Планировать, что за чем делать, когда большая задача разбивается на маленькие



Переговорные навыки

- ▶ Во-первых, это нужно, чтобы «продавать» свои идеи и навыки
- ▶ Во-вторых, важно уметь убеждать собеседников. Переговорные навыки необходимы даже на кадровом собеседовании. Каждое собеседование — это переговоры: у вас есть своя позиция и условия, у работодателя — свои. Каждая сторона защищает свои потребности и пытается найти компромисс

Как сформировать навык:

- Первое — готовьтесь к переговорам. Они пройдут хорошо, если не будут для вас неожиданностью. Найдите информацию о собеседнике, подумайте, чего хотите достичь — поставьте цель
- Практикуйте переговоры в повседневной жизни



Гибкость и креативность

- ▶ Любому разработчику нужно быть готовым к изменениям и уметь принимать решения в нестабильной ситуации. Именно гибкость ума, креативность и нестандартное мышление — это главные принципы Agile-подхода

Как сформировать навык:

- старайтесь избегать думать стандартно, по шаблону, руководствоваться стереотипами. Часто мы сталкиваемся с ситуацией, когда на наши предложения мы получаем ответ «в этой организации так не принято делать». Остановитесь и подумайте, почему здесь так заведено и что будет, если вы сделаете иначе, может быть, получится что-то новое и интересное
- Смело беритесь за новые задачи — мы все делаем хорошо то, что делаем постоянно, и неохотно берёмся за что-то незнакомое, чего ещё ни разу не делали. Причина одна — мы боимся неудачи. Но неудачи — это часть обучения и развития
- Занимайтесь творчеством — оно развивает гибкость мышления



Проактивность

- ▶ Проактивность — это полезный навык для управления своей жизнью и карьерой, чтобы не плыть по течению, а самостоятельно задавать нужный путь и траекторию
- ▶ Если вы хотите выделяться из массы, быть лучшим, нужно быть проактивным — значит, делать больше, чем от вас ожидают

Как сформировать навык:

- Во-первых, забудьте фразу «это не моя работа». Есть вещи, которые напрямую не относятся к вашим обязанностям, но иногда возникают ситуации, когда ваша помощь необходима — в Agile люди помогают друг другу, берут на себя смежные функции: такой подход помогает команде быстрее и эффективнее выполнять задачи, а вам — расти
- Прежде чем идти к более старшим товарищам с вопросом — постарайтесь самостоятельно найти решение проблемы. В команде очень ценится подход, когда человек подходит не с «голым» вопросом, а уже владеет какой-то информацией, вариантами решения и спрашивает не «как это сделать», а «как сделать лучше и правильнее»



Навыки презентации

- ▶ Это очень нужно для то, чтобы демонстрировать результаты своей работы и делиться опытом с коллегами, а также помогает обучать других
- ▶ Полезно записывать свое выступление на камеру, чтобы посмотреть на себя со стороны. Многие начинающие спикеры совершают много произвольных движений, что мешает им и слушателям. Именно видео поможет вам заметить такие нюансы и попытаться избавиться от них

Как сформировать навык:

- Смотрите, как это делают другие. Просматривайте хорошие публичные выступления, запоминайте то, что вам показалось интересным и зацепило. Если есть возможность где-то выступить публично — делайте это, не стесняйтесь
- Тяните руку на лекциях и конференциях, выходите к доске, в круг, из ряда. Не прячьтесь за спинами других людей!



Менторинг

- ▶ Менторинг необходим, чтобы обмениваться знаниями с коллегами и помогать адаптироваться новичкам
- ▶ Помимо этого, обучение других укрепляет собственные знания

“В любой деятельности наступает момент, когда для того, чтобы вырасти как специалист, уже недостаточно развивать только свои собственные знания и навыки — необходимо передать их другим. Процесс обучения позволяет глубже заглянуть в самую суть своей работы, вычленить самое важное и систематизировать это до такого уровня, чтобы можно было понятно объяснить это другим”

Как сформировать навык:

Если вы умеете что-то, чего не умеют другие — пробуйте себя в роли репетитора. Это тренирует терпение и другие необходимые навыки



Вывод

- ▶ Возможно, для успеха в карьере нам достаточно только soft skills?
Конечно нет. В первую очередь, любые IT-компании ищут талантливых профессионалов, проверяют и оценивают ваши навыки специалиста (hard skills)
- ▶ Но только когда soft и hard skills гармонично сочетаются в одном человеке, он вырастает выдающимся профессионалом





Спасибо за внимание!

