**Планирование в Agile методологии**

**Раздел планирования описан на примере планирования спринта для итерационных моделей разработки.**

**Планирование Спринта**

Задачи, над которыми будет трудиться команда разработки во время спринта, определяются на планировании спринта. План создается совместными усилиями всей скрам-команды.

Планирование спринта ограничено по времени. Для спринта длительностью один месяц планирование не должно занимать более 8 часов. Если спринт короче, то и планирование проводится быстрее.

Скрам-мастер убеждается, что событие состоялось, а участники понимали его цель. Скрам-мастер обучает скрам-команду соблюдать временное ограничение.

По результатам планирования спринта скрам-команда решает:

* каким будет инкремент в конце спринта;
* как организовать работу, чтобы получить готовый инкремент продукта.

**Тема первая: что будет сделано?**

Команда разработки прогнозирует объём функциональности, который будет разработан в течение спринта. Владелец продукта выносит на обсуждение два важных вопроса: бизнес-цели, которые должны быть достигнуты в спринте, и элементы бэклога продукта, необходимые для достижения цели спринта. На основании этих данных скрам-команда формирует единое понимание о всей работе в спринте.

Для проведения планирования спринта нужны: бэклог продукта, последний инкремент продукта, прогноз возможностей команды разработки в будущем спринте, статистика её прошлой производительности. При этом ,только команда разработки определяет количество элементов бэклога продукта, которые могут быть выполнены в спринте. Ей же принадлежит исключительное право оценивать объём работ, который по силам завершить в текущем спринте.

Во время планирования спринта скрам-команда также формирует цель спринта. Цель спринта служит необходимым ориентиром для реализации элементов бэклога продукта и помогает команде разработки лучше понять, для чего создается инкремент.

**Тема вторая: как будет выполнена работа?**

Когда цель спринта определена и выбраны элементы бэклога продукта, команда разработки решает, как реализовать эту функциональность в виде готового инкремента продукта в течение спринта. Выбранные элементы бэклога продукта и план их реализации называют бэклогом спринта.

Составление плана работ в спринте команда разработки обычно начинает с организации системы и работы, необходимых для трансформации бэклога продукта в полностью готовый инкремент.

Работа может отличаться объёмом и сложностью, поэтому команда разработки планирует достаточный объём задач, который, по её мнению, удастся завершить за предстоящий спринт. Часто к концу планирования спринт команда разработки более тщательно детализирует работу, которую будет выполнять в первые дни спринта. Для этого она разделяет работу на более мелкие задачи, обычно, длительностью не более одного дня.

Команда разработки самоорганизуется как во время спринта, так и во время его планирования при работе над бэклогом спринта.

Владелец продукта помогает прояснить смысл выбранных элементов. Если у команды разработки набирается слишком много или слишком мало работы, то владелец продукта может пойти на компромисс. Тогда команда разработки вместе с владельцем продукта корректируют количество и состав выбранных элементов бэклога продукта для достижения запланированной цели спринта. Чтобы получить дополнительную информацию в предметной̆ или технической областях, команда может пригласить сторонних экспертов для консультации.

К концу планирования спринта команда разработки должна уметь объяснить владельцу продукта и скрам-мастеру, как она намерена работать в рамках самоорганизации, чтобы достичь цель спринта и создать ожидаемый инкремент.

**Цель Спринта**

**Цель Спринта** – это установленный для спринта ориентир, который достигается через выполнение части бэклога продукта. Цель спринта формируется во время его планирования и объясняет команде разработки, для чего создается инкремент.

Цель спринта оставляет команде разработки некоторую гибкость в объёме функциональности, которую они разрабатывают в рамках спринта. Так выбранные элементы бэклога продукта могут реализовывать одну связанную функцию, которая является целью спринта. Или целью спринта может быть любая другая логическая связь, для достижения которой команда разработки будет работать совместно, а не разрозненно, над разными задачами.

Цель спринта – это ориентир для команды разработки. Чтобы его достичь, команда должна использовать технологии и реализовывать функциональность. Если объём работ становится отличным от ожиданий команды разработки, команда договаривается с владельцем продукта об объёме бэклога текущего спринта.

**ЧТО ТАКОЕ ПЛАНИРОВАНИЕ СПРИНТА?**

**Планирование спринта**— это ограниченная по времени встреча в начале спринта, на которой команда и владелец продукта (ВП) обсуждают и принимают решение о том, какая работа будет завершена в спринте.

Как правило, встреча делится на две части: «Что?» и «Как?».

**ЧТО?**

Владелец продукта приносит с собой список приоритетных пользовательских историй на встречу. В идеальном случае, производительность команды известна, поэтому ВП хорошо представляет, сколько работы войдет в cпринт. Идет командное обсуждение и разбивка историй, а затем команда договаривается и фиксирует работу, которая будет завершена в спринте.

**КАК?**

* Команда обсуждает согласованную работу и план ее завершения.
* Пользовательские истории разбиваются на задачи, уточняются технические детали.

Если команда практикует регулярные встречи по уточнению беклога продукта, то она уже имеет представление, о чем идет речь в истории.

**КАК ЭТО ВЫГЛЯДИТ?**

**ЧАСТЫЕ ПРОБЛЕМЫ**

* Владелец продукта сам определяет и решает, какая работа будет завершена.
* Беклог продукта не актуален, не приоритезирован или не готов к обсуждению.
* В конце планирования все слишком детализировано и вся работа уже распределена по исполнителям (эту проблему трудно преодолеть).
* Никто не понимает, что означает статус «Готово».
* Встреча слишком длинная.
* Встреча не включает участников в процесс.
* Некоторым людям сложно проявляться.
* Неподходящая среда, команда не чувствует поддержки или безопасности.
* Нет доверия или уважения с обеих сторон.
* Команда не понимает, для чего нужна эта встреча.

**ПЕРЕД ПЛАНИРОВАНИЕМ СПРИНТА**

**Чек-лист для подготовки к успешному планированию:**

* Мотивированная команда.
* Хороший контакт и доверие между ВП и командой.
* Если ВП не доверяет выводам команды, встреча быстро станет упражнением по избеганию, вместо диалога о работе. Команде важно видеть ценность встречи и преимущества планирования. Если есть какие-либо сомнения ‒ процесс рухнет.
* Встреча проходит в установленное время.
* Она не должна быть слишком длинной и утомительной, потому что тогда теряется ценность.
* Проведена подготовительная работа, часто, в формате встреч по уточнению беклога.
* ВП гарантирует, что существует здоровый, поддерживаемый и приоритезированный беклог.
* Истории в верхней части беклога, которые должны входить в следующий спринт, разбиты на небольшие части, чтобы команда могла их обсудить и запланировать.
* Скрам-мастер убедился, что участвуют все члены команды: владелец продукта, скрам-мастер, разработчики, тестировщики, администратор базы данных — каждый, кто является частью команды разработчиков. Другие заинтересованные стороны могут присутствовать только в качестве наблюдателей, не прерывающих процесс.
* Каждый участник должен чувствовать свою ценность на встрече, иметь возможность поделиться своим видением.
* Решения о работе принимаются командно.
* Если ВП принимает все решения о работе и ее выполнении, команда будет чувствовать, что это пустая трата времени.
* Консенсус по поводу метода оценки и разбивки работы. Story Points или Planning Poker — команде нужно договориться о методе оценки, чтобы работать согласованно.

**ДЛИТЕЛЬНОСТЬ**

Длительность встречи зависит от длины спринта, чем дольше спринт, тем больше времени нужно для его планирования. Для ориентира:

* Однонедельный спринт — 2 часа.
* Двухнедельный спринт — 4 часа.
* Спринт длинной в 1 месяц — 8 часов.

Длительность также очень зависит от зрелости и эффективности команды и владельца продукта, от объема предварительной подготовки.

**ЦЕЛИ**

**Задача встречи**— сформулировать цель спринта. Ее можно представить в форме беклога спринта. Беклог спринта — список приоритезированных задач, которые команда берется завершить до конца спринта. Здесь важно помнить о командных критериях готовности (Definition of Done).

Если Вы представили себе конечный результат спринта, то согласуйте его с владельцем продукта и Ваш путь станет гораздо проще. Команду очень мотивирует визуализация и мысленное представление этого результата.

Один из ключевых принципов гибкости — способность принимать изменения. Очевидно, что меньше всего информации у нас есть в начале проекта, во время работы часто происходят неожиданные вещи. Поэтому позвольте беклогу меняться в процессе и будьте готовы уточнять его при необходимости.

**СТРУКТУРА ВСТРЕЧИ**

Структура встречи нужна в виде предварительного высокоуровневого плана. Пример: рассказ владельца продукта, команда обсуждает задачи, сессия вопросов и ответов с ВП, беклог спринта уточнен, ретроспектива и встреча закрыты. Эта практика должна быть неизменной для каждой встречи, поскольку она поможет команде войти в хороший режим и получить от нее максимальную отдачу.

Производительность команды:

* Достаточно взять среднее последних 3 спринтов, как руководство.
* Обсудите часы доступности команды, отпуска, режим работы членов команды.
* Помните, что спринты не бывают одинаковыми!
* Не пытайтесь быть слишком детальными — это бесполезная трата времени, т.к. количество неизвестных слишком велико.
* Команда все равно согласовывает объем работы.
* Оставьте некоторое время для решения пока неизвестных вопросов и проблем. Так команда получает больше свободы действия.
* Проще добавить работу в спринт, если у вас хорошо проработан беклог, чем убрать ее.

Сотрудничайте, вместе планируйте работу над пользовательскими историями, не оценивайте объем ресурсов, не пытайтесь слишком оптимизировать или микроменеджить каждого. Продукты должна делать команда, а не группа людей, работающих над своими задачами.

**РАЗБИВКА ЗАДАЧ**

* Определите критерии готовности (DoD). Это устраняет конфликты и дает прозрачность процесса. «Готово» должно значить потенциальную поставку продукта.
* Обсудите все задачи, чтобы получить представление о работе и ее выполнении: создание скриптов, рефакторинг, интеграция кода, тестирование и автоматизация, исправление багов, техническое обслуживание.
* Не слишком привязывайтесь к процессу оценки, ведь это всего лишь предположение, а не обязательство.
* Частая ошибка на этом этапе — хождение кругами.
* Не привлекайте отдельных членов команды к ответственности за оценку. Команда не должна бояться оценивать.
* Не стоит сравнивать относительные оценки с фактически затраченным временем (если нет существенных различий, но тогда это нужно вынести на командную ретроспективу).
* Вся команда владеет беклогом спринта, поэтому не распределяйте задачи по исполнителям.Если это происходит, тогда одни и те же люди постоянно получают одну и ту же работу. Это плохо, потому что тогда Вы развиваете «специалистов» в своей команде, а обмен знаниями и развитие становятся ограниченными. Совершенно нормально иметь специалистов, пока они обмениваются знаниями с командой, но нет ничего хуже, чем один человек, который несет знания и ответственность за конкретную часть кода, а затем он уходит — забирая эти знания с собой.
* Цели спринта достигает вся команда. Поскольку у вас есть список приоритетов, то, если одна задача выполнена, член команды может предложить помощь другим, если нужно, или перейти к следующей по приоритету задаче.
* Используйте Story Points как способ относительной оценки. Тут Вы сравниваете задачи по сложности, а не по времени. Разработчику проще сказать «эта задача в 3 раза сложнее, чем та», а не «эта задача займет у меня около 4 дней». Не смешивайте Story Points с часами, тогда люди просто пытаются конвертировать Story Points во время, а затем Story Points вообще теряют свою ценность.
* Согласовывайте размеры ваших задач. Хороши задачи, которые занимают не более 1 дня. Преимущества согласовывания размеров задач:
  + разработчикам проще планировать рабочий день;
  + повышается эффективность, если задачу можно начать и завершить в тот же день;
  + небольшие задачи дают более полное представление об объеме работы;
  + можно сократить «узкие горлышки» процесса;
  + атомарные задачи можно было делать параллельно. Задачи, зависящие друг от друга, будут вызывать проблемы и провоцировать «узкие горлышки».
* Создавайте только те задачи, которые требуют выполнения: разработка, тестирование, документация, демо и т. д. Не вносите в них работу владельца продукта или командные встречи.
* Если Вы не уверены в задаче, то создайте “зазор”. Он нужен для проведения исследования.
* Не планируйте 8-часовой рабочий день, даже если команда нанята на это время. В действительности, команда не работает 8 часов подряд. «Эффективность» хорошей здоровой команды — около 70% рабочего времени, поэтому стоит планировать хотя бы 6 часов в день, т.к. в течение дня происходит много всего: встречи, обеденный перерыв, выяснения деталей с ПО, короткие перерывы.

**РАБОЧАЯ СРЕДА**

Команде важно чувствовать себя в безопасности, понимать, что нормально не знать всего прямо сейчас. Agile весь состоит из адаптации, подстройки и обучения. Должно быть ощущение сотрудничества, поддержки, любопытства и драйва. Команда не должна бояться задавать вопросы, высказываться. Команде важно чувствовать уверенность в том, что они действительно могут поставить и какие взять на себя обязательства; и если у них есть какие-то сомнения или возникают риски, отнеситесь к ним всерьез. Очень важно, что последнее слово о поставке остается за командой, а не владельцем продукта.

**Автономия команды**

Я большой сторонник самоорганизующихся автономных команд. Не потому, что ленив …. А потому, что я видел результаты их работы. Автономные команды эффективны, потому что они владеют продуктом от начала до конца, они независимо принимают решения и это способствует росту и мотивации всех участников.

Как скрам-мастер, Вы должны позволить команде ставить собственные цели, обучить команду, как взять на себя ответственность, помочь им присвоить план и беклог спринта. Ни в коем случае не сдерживайте команду, помогите им удивляться и изобретать, чтобы превзойти самих себя. Позвольте им думать и пускай в комнате время от времени воцаряется тишина.

**Активное слушание**

В начале встречи особенно важно вовлечь команду в рассказ ВП, когда он объясняет и рассказывает о приоритетах. Хорошо, если команда активно слушает и задает вопросы на этом этапе.

**Взаимное уважение**

Я говорил о важности доверия, но уважение не менее важно. Я упомянул, что владелец продукта должен иметь доверие в команде, и важно, чтобы оно было взаимным. Команда должна уважать ВП и доверять решениям, которые он принимает. Это тот человек, который приоритезирует их работу, им важно знать, что их работа имеет ценность и хорошо продумана.

ВП отвечает за «почему» и команда должна позволить ему сосредоточиться на этом. ВП отвечает за управление заинтересованными сторонами и представляет голос бизнеса.  
Команда ответственна за «как», ВП слушает и, возможно, делится мнением, но никоим образом не диктует команде, как выполнять задачи, это их работа и они в этом профи. С обеих сторон нужно взаимное уважение и четкое определение ролей для каждого.

**Вопросы**

Не успокаивайтесь, если команда не задает никаких вопросов. Если вопросов нет, то кажется, что есть полное понимание, но такое бывает крайне редко. Вы можете договориться, что каждый должен задать хотя бы один вопрос. Обычно, когда возникает один вопрос, за ним появляются и другие.

**Процесс**

Если Вы используете физическую Kanban доску, предложите команде самостоятельно выписать свои стикеры, повесить их на доску и расставить приоритеты, опять же, это хороший способ развивать самоорганизацию. Используйте время встречи для обновления своего инструмента, будь то физическая Kanban доска, Jira или TFS. Во-первых, каждый сможет увидеть план, согласиться на него и начать работу. Во-вторых, хорошо, когда каждый понимает процесс. Тогда, если скрам-мастер болен, то члену команды несложно подменить его.

**ЗАВЕРШЕНИЕ ВСТРЕЧИ**

Планирование спринта ограничено во времени, заканчивайте его вовремя. Даже если фактически планирование еще не закончено, важно начать работу, продолжение встречи не поможет завершить пользовательские истории. Совершенно нормально не знать всех деталей, это развивает гибкость!

Иногда команде необходима смелость сказать «Ок, этого достаточно, мы не знаем всех деталей, но будем изучать оставшиеся и приспосабливаться по пути». Быть гибким — значит экспериментировать, изобретать, и старт работы дает команде шанс сделать это!

В конце встречи возьмите несколько минут, чтобы снова подумать о том, что получилось хорошо, чего не было в предыдущем спринте и убедитесь, что забираете удачные практики с собой, а все плохое не повторится. Это позволяет бросить последний взгляд на запланированную работу и начать свое путешествие.

**Capacity**

* Capacity прогноз — количество идеальных часов, доступное в следующем спринте.
* Понимание, сколько часов у нас есть на работу: на написание кода, тестирование, т.д.
* Как правило, участник проекта работает не более пяти часов в день.
* Эффективное распределение задач.
* Нет смысла планировать задачи на тех, кто будет в отпуске или занят другими активностями.
* Мало пользы принесет технический анализ задачи, выполненный участником проекта, который в следующем спринте будет отсутствовать.
* Аккуратное и точное планирование.
* Мы оцениваем задачи в часах и берем в спринт столько, сколько соответствует нашей capacity.

**Velocity Scrum**

Если представить себе движение автомобиля, то скорость в нем оценивается по спидометру и тогда всё предельно ясно. Однако, в жизни в целом и в каком-то конкретном деле нет спидометра и как оценить скорость объективно — вопрос. Если представить, что на автомобиле сломался спидометр, то скорость может быть оценена постфактум, то есть, когда человек будет ехать 60 минут и проедет, например, 100 км, затем опять пройдет 60 минут, но проедет 120 км, он будет видеть с какой скоростью он двигался – около 110 км/ч. При этом, если во время следующих 60 минут он остановится и потратит на заправку 12 минут, на обед 15 минут и проедет 55 км, то средняя скоростьза весь путь составит – 98 км/ч.

Как и в движении на автомобиле, скорость можно измерять и в Scrum, и называется это **Velocity (скорость)**. Расчет **Scrum Velocity**очень простой и также состоит из поставленных отметок, как через каждые 60 минут было какое-то количество километров.

Для примера, предоставим график Velocity, отображающий по горизонтальной оси количество Sprints, а по вертикальной — Story Points.

В таком графике, по сути, изображено Story Points и на основе этих показателей выстраивается среднее значение скорости. Однако, график Velocity может быть и иной, с простым отображением Story Points, на основе которых визуально видна тенденция.

Если быть кратким в сравнении с движением по дороги, то список ассоциаций, влияющих на скорость, может быть такой:

* Дорога — она же препятствия.
* Топливо — мотивация, что движет нами.
* Опыт водителя — знание / опыт / компетенция разработчик.
* Условия для автомобилей — DEV среда.
* Видимость — прозрачность проекта.
* Направление — цели проекта.
* Трафик / правила вождения — процессы.
* Пункт назначения – продукт.

Хочется, однако, отметить, что по поводу метрики Velocity в Scrum ходят споры, и кто-то считает данные графики не очень полезными и сложными в определении возможных проблем, да и самого разгона. В целом, все сходятся во мнении, что Velocity должен использоваться специалистом, как дополнительное наглядное отображение эффективности, которое как калька накладывается на другие данные, помогая скорректировать аналитическую работу Scrum Master.

**Трудозатраты** — количество рабочего времени, необходимого для выполнения работы (выражается в человеко-часах).

Перед выполнением каждого задания, возникают следующие вопросы:

* Как много времени понадобится на выполнение работы?
* Когда всё будет готово?
* Можно ли гарантированно выполнить работу к такому-то сроку?
* Каковы наиболее оптимистичный и пессимистичный прогнозы по времени?

**Оценка трудозатрат**

* Любая оценка лучше её отсутствия. Даже если область предстоящей работы для Вас совершенно нова, даже если Вы ошибётесь в своей оценке на порядок, Вы как минимум получите опыт, который сможете использовать в будущем при возникновении подобного рода задач.
* Оптимизм губителен. Как правило, люди склонны недооценивать сложность незнакомых задач, что приводит к занижению оценки трудозатрат. Но, даже при достаточно точном определении самих трудозатрат, люди без опыта выполнения оценки склонны рассматривать предстоящую работу как некую изолированную деятельность, забывая о том, что на протяжении любого рабочего дня «чистую производительность труда» будут снижать такие факторы, как переписка по почте, участие в собраниях и обсуждениях, решение сопутствующих технических вопросов, изучение документации и обдумывание сложных частей задачи, форс-мажорные обстоятельства (неотложные дела, проблемы с техникой и т.д.). Таким образом, обязательно стоит учитывать, что в реальности Вы сможете заниматься поставленной задачей не 100 % рабочего времени, а меньше (насколько меньше — зависит от конкретной ситуации, в среднем, принято считать, что на поставленную задачу из каждых восьми рабочих часов Вы сможете потратить не более шести). Учитывая этот факт, стоит сделать соответствующие поправки в оценке общего времени, которое понадобится на выполнение работы (а именно оно чаще всего интересует постановщика задачи).
* Оценка должна быть аргументирована. Это не значит, что Вы всегда должны пускаться в подробные пояснения, но Вы должны быть готовы объяснить, почему Вы считаете, что та или иная работа займёт именно столько времени. Во-первых, продумывая эти аргументы, Вы получаете дополнительную возможность лучше оценить предстоящую работу и скорректировать оценку. Во-вторых, если Ваша оценка не соответствует ожиданиям постановщика задачи, Вы сможете отстоять свою точку зрения.
* Простой способ научиться оценивать — оценивать. В специализированной литературе приводится множество технологий, но первична сама привычка выполнять оценку предстоящей работы. В процессе выработки этой привычки Вы, естественным образом, встретитесь с большинством типичных проблем и, через некоторое время, научитесь делать соответствующие поправки в оценке даже не задумываясь. Что оценивать? Что угодно: Сколько времени у Вас уйдёт на прочтение новой книги? За сколько времени Вы доедете домой новым маршрутом? За сколько времени Вы напишете курсовую или дипломную работу? — и так далее. Не важно, что именно Вы оцениваете, важно, что Вы повторяете это раз за разом, учитывая накапливающийся опыт.

**Алгоритм обучения формированию оценки:**

* Сформируйте оценку. Ранее уже было отмечено, что нет ничего страшного в том, что полученное значение может оказаться очень далёким от реальности. Для начала, оно просто должно быть.
* Запишите полученную оценку. Обязательно именно запишите. Это застрахует Вас как минимум от двух рисков: забыть полученное значение (особенно, если работа заняла много времени), соврать себе в стиле «ну, я как-то примерно так и думал».
* Выполните работу. В отдельных случаях, люди склонны подстраиваться под заранее сформированную оценку, ускоряя или замедляя выполнение работы — это тоже полезный навык, но сейчас такое поведение будет мешать.  
  Однако, если Вы будете тренироваться на десятках и сотнях различных задач, Вы физически не сможете «подстроиться» под каждую из них и начнёте получать реальные результаты.
* Сверьте реальные результаты с ранее сформированной оценкой.
* Учтите ошибки при формировании новых оценок. На этом этапе очень полезно не просто отметить отклонение, а подумать, что привело к его появлению.
* Повторяйте этот алгоритм как можно чаще для самых различных областей жизни. Сейчас цена Ваших ошибок крайне мала, а наработанный опыт от этого становится ничуть не менее ценным.

**Полезные идеи по формированию оценки трудозатрат:**

* Добавляйте небольшой «буфер» (по времени, бюджету или иным критическим ресурсам) на непредвиденные обстоятельства. Чем более дальний прогноз Вы строите, тем большим может быть этот «буфер» — от 5–10 % до 30-40 %. Но ни в коем случае не стоит осознанно завышать оценку в разы.
* Выясните свой «коэффициент искажения»: большинство людей, в силу особенности своего мышления, склонны постоянно или занижать, или завышать оценку. Многократно формируя оценку трудозатрат и сравнивая её впоследствии с реальностью, Вы можете заметить определённую закономерность, которую вполне можно выразить числом. Например, может оказаться, что Вы склонны занижать оценку в 1.3 раза. Попробуйте в следующий раз внести соответствующую поправку.
* Принимайте во внимание не зависящие от Вас обстоятельства. Например, Вы точно уверены, что выполните тестирование очередного билда за N человеко-часов, Вы учли все отвлекающие факторы и т.д. и решили, что точно закончите к такой-то дате. А потом, в реальности, выпуск билда задерживается на два дня, и Ваш прогноз по моменту завершения работы оказывается нереалистичным.
* Задумывайтесь заранее о необходимых ресурсах. Так, например, необходимую инфраструктуру можно (и нужно!) подготовить (или заказать) заранее, т.к. на подобные вспомогательные задачи может быть потрачено много времени, к тому же, основная работа часто не может быть начата, пока не будут завершены все приготовления.
* Ищите способы организовать параллельное выполнение задач. Даже если Вы работаете один, всё равно какие-то задачи можно и нужно выполнять параллельно (например, уточнение тест-плана, пока происходит разворачивание виртуальных машин). В случае если работа выполняется несколькими людьми, распараллеливание работы можно считать жизненной необходимостью.
* Периодически сверяйтесь с планом, вносите корректировки в оценку и уведомляйте заинтересованных лиц о внесённых изменениях заблаговременно. Например, Вы поняли (как в упомянутом выше примере с задержкой билда), что завершите работу, как минимум, на два дня позже. Если Вы оповестите проектную команду немедленно, у Ваших коллег появляется шанс скорректировать свои собственные планы. Если же Вы в «час икс» преподнесёте сюрприз о сдвигах срока на два дня, то создадите коллегам объективную проблему.
* Используйте инструментальные средства — от электронных календарей до возможностей вашей систем управления проектом: это позволит Вам как минимум не держать в памяти кучу мелочей, а как максимум — повысит точность формируемой оценки.

**Оценка с использованием структурной декомпозиции**

**Структурная декомпозиция** — иерархическая декомпозиция объёмных задач на всё более и более малые подзадачи с целью упрощения оценки, планирования и мониторинга выполнения работы.

В процессе выполнения структурной декомпозиции большие задачи делятся на всё более и более мелкие подзадачи, что позволяет:

* описать весь объём работ с точностью, достаточной для чёткого понимания сути задач, формирования достаточно точной оценки трудозатрат и выработки показателей достижения результатов.
* определить весь объём трудозатрат как сумму трудозатрат по отдельным задачам (с учётом необходимых поправок).
* от интуитивного представления перейти к конкретному перечню отдельных действий, что упрощает построение плана, принятие решений о распараллеливании работ и т.д.

Если абстрагироваться от научного подхода и формул, то суть такой оценки сводится к следующим шагам:

* декомпозиции требований до уровня, на котором появляется возможность создания качественных чек-листов;
* декомпозиции задач по тестированию каждого пункта чек-листа до уровня «тестировощных действий» (создание тест-кейсов, выполнение тест-кейсов, создание отчётов о дефектах и т.д.);
* выполнению оценки с учётом собственной производительности.

Оценка трудозатрат (помимо получения данных непосредственно о необходимом количестве персонала для тестирования) также позволяет определить:

* Приблизительную стоимость проведения тестирования.
* Сроки тестирования.
* Примерный график работ.

Существуют различные методы для проведения оценки. Некоторые из которых более серьезны в применении и требуют использования специальных математических расчетов, но часть из методов также является более простыми и может даже неосознанно применятся в планировании.

**Метод «пальцем в небо»**

Характеризуется тем, что оценивание осуществляется с учётом некоторого прошлого опыта или же и вовсе без такового на основании предположений и догадок. Является полностью неточным и содержит значительный процент погрешности.

**Экспертная оценка**

Название метода полностью отображает его суть в том, что оценка осуществляется на основании работы с предыдущими проектами либо же для работы привлекаются эксперты определённой области или специалисты, знакомые с тестируемым приложением.

**Специальный метод**

Оценивание трудозатрат осуществляется на основании предполагаемых временных рамках. Учитывая, что при таком подходе не берутся во внимание даже предыдущий опыт, погрешность достаточно велика.

**Структура декомпозиции работ**

Расчет количества заданий, выполнение которых ожидается от команды на этапе тестирования, осуществляется на декомпозиции проекта на определённые логические более мелкие части (например: модули –-> подмодули —> функциональности). И уже после проведения декомпозиции оценивается объем работ каждой небольшой части проекта.

Например, протестировать авторизацию. Буква «о» не проходит – можно сходу сказать 5 мин для себя. Это минимальная единица. Такое может определить даже человек, который работает всего пару месяцев. Плюс в таком методе – Вам сложнее что-то забыть.

**Метод Дельфи**

Основывается на том же методе декомпозиции работ, что и Структура декомпозиции работ с тем дополнением, что ожидаемые к выполнению задания распределяются на каждого отдельного члена команды, который самостоятельно оценивает временные затраты на их выполнение.

Данный метод характеризуется значительной точностью.

**Метод определения трудозатрат в процентном отношении к разработке**

Оценка основывается на предположении, что трудозатраты на тестирование являются прямо пропорциональными от таковых на разработку.

**Метод процентного распределения**

Использование метода исходит из того, что все этапы разработки программного продукта выражаются через процентное значение трудозатрат для каждого отдельно этапа. При этом, непосредственно этап тестирования также делится на его составляющие (планирование, проектирование тестов, выполнение тестов, анализ результатов), каждому из которого присваивается свой процент трудозатрат.

При осуществлении оценки трудозатрат принимаются во внимание также следующие факторы:

* Уровень мастерства команды в целом.
* Наличие и качество проектной документации.
* Применение автоматизации.
* Используемые инструменты и средства при тестировании.

Обратите внимание на диаграмму: здесь приведена статистика для сайтов от полугода – года. Программирование занимает от 20% до 40% разработки, это не тоже самое, что 20-40% от проекта, а это, в среднем, 15% от проекта. Тестирование никогда не занимает 15% от продукта. Если у вас не закладывают столько время для тестирования, то закладывайте хоть сколько-нибудь. Желательно, выясните статистически, какой процент от проекта занимает тестирование и это применимо, если у Вас стабильные версии релизов, постоянный объем продуктов один и тот же.

**Planning Poker (Scrum Poker)**

Planning Poker или Scrum Poker, пожалуй ,одно из важнейших мероприятий в методологии Scrum или любой гибкой технологии разработки. Практически всегда перед командой встает вопрос: **как оценить эту задачу?**

Оценка трудозатрат будет влиять на целую цепочку зависимостей. От сложности работы зависит количество баллов, начисляемых в рейтинг, сроки сдачи заказа и количество денег, которые должен будет заплатить заказчик. Пожалуй, каждый из членов Scrum Team может оценить ту или иную задачу лучше других, особенно, если она лежит в области его профессиональной деятельности. Сама методология Scrum, в выполнении той или иной работы, уводит нас из области личной ответственности в область коллективной. Логично при этом считать, что и оценивать ту или иную задачу, за которую несет ответственность вся команда, должна вся Scrum Team. Более того, такой подход поможет более точно определить реальные сроки, которые конкретный человек может себе искусственно завысить по разным причинам.

**Что собой представляют карты для Planning Poker / Scrum Poker**

На самом деле, таких вариантов карт очень много и каждый может придумывать свои, например, означающие количество дней на разработку.

Есть несколько вариантов карт, которые пользуются большой популярностью.

**1 вид популярной колоды для Planning Poker:**

Карточки представляют собой последовательность чисел Фибоначчи: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.

**2 вид популярной колоды для Planning Poker:**

Данный вид имеет следующие значения: 0, ?, 1, 2, 3, 5, 8, 13, 20, 40, 100, «?», «Чашка кофе». Знак вопроса означает, что «игрок» не понял до конца смысл обсуждаемого или не обладает достаточно информацией, чтобы оценить её. Чашка кофе, в свою очередь, означает: «Я устал, давайте передохнем».

**Как проходит Scrum Poker / Planning Poker**

Один человек является ведущим и он не участвует в «игре». На обсуждение выносятся поочередно пункты, которые необходимо оценить. Каждый пункт позволено обсудить и провести обзор без оценочных данных. После этого каждый член команды выбирает карточку и кладет её рубашкой вверх. После того, как все положили карты – они вскрываются. Идеальным состоянием считается, если разброса в значениях практически нет. Как можно догадаться, такое бывает не всегда. Так или иначе в выброшенных картах будут наименьшие и наибольшие значения. Людям, выбросившим такие карточки, дают слово и они высказывают свое мнение, почему оценка была именно такой. Это позволяет получить больше информации всей остальной команде и задуматься, услышав доводы, либо объяснить выбросившим высокие или низкие позиции свою точку зрения.

После этого карты выбрасываются снова и обычно разрыв уже сокращается, однако, если этого не произошло, то цикл повторяется. В данном случае, рекомендуется ввести таймер на цикл и поставить ограничения по циклам, но, в большинстве случаев, после третьего раза показатели примерно одинаковые. Если имеются небольшие расхождения, то в приоритете показатель человека, который непосредственно будет в разработке этой задачи.

**Основные проблемы в использовании Planning Poker**

Как и любая методология или технология должна иметь четкие инструкции в использовании, так и Planning Poker имеет четкие предписания, которые не позволяют делать ошибки и сводить на нет внедрение этого усовершенствования рабочего процесса.

**Эффект привязки в Scrum Poker**

Главной проблемой всегда был эффект привязки, который может проявлять себя по-разному. Главной ошибкой, вызывающей этот эффект, является открытое обсуждение оценок. Если тот, кто начинает обсуждение, говорит примерно следующее: «Я считаю, что данное задание займет 18 часов разработки», то, так или иначе, все будут акцентированы на сроке в 18 часов, и тот, кто считал, что задача будет решена за 2 дня, может подумать, что на самом деле и 18 часов будет достаточно, а тот, кто думал про 5 часов, может подумать, что недооценил. С одной стороны, консенсус достигается быстрее, но с другой стороны, он не будет эффективным, а эффективность – это то, для чего мы всё это делаем. В такой ситуации больше в результат войдет мнение одного человека, а не команды.

**Не выделяться из толпы**

Второй знаменитой проблемой бывает ситуация, когда оценки выставляются не одновременно. В такой ситуации кто-то, конечно, выскажет свое мнение, но, с другой стороны, человек сомневающийся решит бросить карту, которая ближе к тем, что есть. К примеру, опять кто-то решил, что задача займет 18 часов, а до него двое выкинули по 5 часов, и логично предположить, что данный человек быстро среагирует, что оценил не так и так выделяться не стоит и бросит не то, что хотел изначально.

**Story Points**

Одна из самых важных сторон методологии Scrum – так называемые **Story Points**. Эта сторона очень плотно интегрирована в Scrum совместно с технологией Planning Poker.

Самая частая проблема в работе Scrum Team – это неумение правильно оценивать сложность работы и временные затраты на её выполнение.

Для многих действительно тяжело выработать правильную шкалу оценок и здесь нужен опыт или хитрый подход. Для максимально эффективного и быстрого внедрения **Story Points**в жизнь команды нужно взять уже отработанные задачи с прошлых проектов и провести их полный анализ. В данный анализ должны входить названия задач, которые выполнялись, и их продолжительность. Дальше всё проще простого: необходимо расположить эти задачи в порядке возрастания, отсортировав по времени. Разделить на группы с одинаковыми показателями или близкими и проставить оценки из вашей колоды карт Scrum Poker.

Рано или поздно использование такого подхода сделает из Вас классную и прогрессирующую команду, что и есть единственно верное направление. Постоянно модернизирующаяся команда, всегда будет разгонять свой Velocity и радовать своих заказчиков успехами.

**Что же происходит на самом деле при Story Points?**

Когда Scrum Team способна оценивать свою работу, ведет график Velocity, следит за Диаграммой сгорания задач, рано или поздно (или с самого начала работы) абсолютно все оценки будут вестись в Story Points.

Например, по графику Velocity можно увидеть, что динамика команды по среднему разгону за каждый Sprint идет вверх и среднее значения за последние две итерации составили 23-30 Story Points (зависит от выбранной системы оценок). Глядя на эти показатели Product Owner может видеть, на какие задачи потратить доступные очки, чтобы заполнить Backlog.

Правильная оценка и использование Story Points делает действительно чудеса, ведь она связывает между собой работу всей команды: Development TeamScrum Master ощущает, на что она способна. Product OwnerBacklog видит, есть ли проблемы у команды, и что можно еще улучшить, а у Sprint не возникает вопросов ,сколько и какие задачи вкладывать в на текущий .

**Ретроспектива спринта**

**Ретроспектива спринта**— это возможность для скрам-команды провести инспекцию, направленную на себя, и создать план улучшений командной работы в следующем спринте.

Ретроспектива спринта проводится после обзора спринта и перед планированием следующего спринта. Максимальная продолжительность ретроспективы – 3 часа (для спринта длительностью один месяц). Для более коротких спринтов, как правило, отводится меньше времени.

Скрам-мастер убеждается, что событие проходит позитивно и продуктивно, обучает всех участников укладываться в отведенное на событие время. Он принимает участие в ретроспективе наравне с другими членами команды, но продолжает нести ответственность за скрам-процесс.

Цели проведения ретроспективы спринта:

* инспекция прошедшего спринта применительно к людям, отношениям, процессам и инструментам. Обнаружение и упорядочение того, что прошло хорошо и того, что нуждается в улучшении;
* создание плана внедрения улучшений в процесс работы скрам-команды.

Скрам-мастер побуждает скрам-команду улучшать процесс разработки и практики в рамках скрам-фреймворка. Это необходимо, чтобы в следующем спринте повысить её эффективность и получать больше удовлетворения от своей работы. Каждую ретроспективу спринта скрам-команда планирует действия для улучшения качества продукта, совершенствуя рабочий процесс или адаптируя критерий готовности, если это необходимо, и не противоречит спецификации продукта и стандартам организации.

К концу ретроспективы, скрам-команда должна запланировать конкретные улучшения, которые она реализует в следующем спринте. Реализация этих улучшений — и есть адаптация скрам-команды. Работать над улучшениями можно в любое время, ретроспектива спринта – формальная возможность сконцентрироваться на инспекции и адаптации.

**Sprint Review Meeting**

Окончание каждого Sprint в Scrum знаменуется значительным приростом функционала продукта. Более того, это означает, что команда полностью написала код, провела полноценное тестирование и выдала готовую к употреблению часть программного продукты, или целый продукт.

Sprint Review Meeting проводится в конце каждого спринта и носит обзорный характер. На встрече команда оценивает то, что она сделала, и ,чаще всего, это выглядит в виде демонстрации новых возможностей.

Не стоит относится к Sprint Review Meeting как к формальной четко поставленной встрече с развернутыми докладами. Sprint Review Meeting это всё же просто логическое завершение Sprint. На подготовку к этой встрече не позволительно готовиться более 2 часов и запрещено использование слайдов типа PowerPoint.

В данной встрече обычно участвует Product OwnerDevelopment Team, Scrum MasterManagement, , , заказчики и разработчики из других проектов.

В ходе Sprint Review Meeting проект оценивается в отношении цели спринта, которая была определена во время планирования. В идеале, команда выполнила все задачи из Product Backlog, помещенные в Sprint, но это не самое важное, а важно то, что была достигнута цель спринта.

**Более подробно, на что смотрит заказчик на Sprint Review Meeting:**

* Команда передала законченный продукт.
* Работа команды завершена.
* Показатели проекта (завершенность кода и тд.).
* Работоспособность выполненных задач.
* Обзор приоритетов (для следующих итераций / спринтов).

**Время Sprint Review Meeting**

Время данного обзора строится по следующей формуле: за каждую неделю Sprint набегает 1 час обзора. То есть, если Sprint был четырехнедельным, то Sprint Review Meeting будет длится 4 часа. Проводится эта встреча в последний день спринта.

**Пример Sprint Review Meeting**

Один созданный пример на всю базу знаний можно и упомянуть здесь. Разработка интернет магазина, описываемая в разных статьях нашей Info Base, само собой разумеется, имеет спринты, а спринты имеют Sprint Review Meeting.

**Sprint Review Meeting при разработке интернет магазина**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тематика | Название | Описание | Статус | Оценка | Релиз |
| Управление каталогом | Добавление продукта | Разработка формы создания продукта, которая содержит фотографию, название, цену, скидку или её отсутствие… | В работе | 2 | Релиз 1 |
| Управление каталогом | Удаление продукта | Удаление продукта как из страницы редактирования, так и списком | В работе | 2 | Релиз 1 |
| Заказ | Оплата | Наложенный платеж | В работе | 10 | Релиз 1 |
| Заказ | Оплата | Оплата с помощью карт Visa и Mastercard | В работе | 10 | Релиз 1 |
| Заказ | Оплата | Оплата с помощью системы Яндекс Деньги | В работе | 10 | Релиз 2 |
| Заказ | Вход | Регистрация с помощью Facebook | В работе | 1 | Незапланировано |
| Заказ | Вход | Регистрация с помощью Google+ | В работе | 1 | Незапланировано |
| … | … | … | … | … | … |

Имея данный Product Backlog, в Sprint были добавлены задачи из первого релиза. А точнее, мы имеем задачи:

1. Добавление продукта.
2. Удаление продукта.
3. Оплата — наложенный платеж.
4. Оплата — оплата с помощью карт Visa и Mastercard.

На Sprint Review Meeting, соответственно, будут продемонстрированы возможности по добавлению продуктов в базу интернет магазина и возможность их удалять. Также продемонстрирована работа корзины с возможностью оплаты заказа по карте или выбором наложенного платежа.