

Процессы в Scrum

JSP & Servlets
15 уровень, 5 лекция

ОТКРЫТА

Планирование спринта

Планирование спринта — это начальный этап в спринте Scrum. На нем определяется объем и способы выполнения работы во время спринта. В планировании участвует вся Scrum-команда.

Спринт — это четко определенный отрезок времени, во время которого нужно выполнить оговоренную часть работы. Перед началом работы спринт нуждается в планировании. Прежде всего, нужно определить длительность и цель спринта.

На рабочем совещании по планированию оговаривается перечень задач и цель спринта. Важно зарядить команду правильной мотивацией к работе, чтобы каждый ее участник был нацелен на успех.

Если же спринт плохо спланирован, то это может привести команду к неудаче. Разработчики не смогут справиться с возложенными на них ожиданиями, потому как задачи оказались нереальны.

Вопросы, которые необходимо решить при планировании спринта:

- Заказчик или владелец ПО оглашает цель спринта, попутно объясняя способы, как ее достичь. Scrum-команда выясняет, какие задания для достижения этой цели можно выполнить на будущем спринте.
- Разработчики распределяют между собой план работ, который согласовывается с заказчиком ПО.
- В составлении плана спринта всегда принимает участие заказчик (владелец) продукта. Он ставит цель, а команда программистов должна выяснить, можно ли ее достичь на спринте.
- В плане стоит использовать бэклог продукта, информацию из которого можно добавить в план.
- Участники команды должны закончить совещание по планированию, четко осознавая, что им нужно для достижения результата. Отобразить порядок будущих действий можно в бэклоге спринта.

На планирование нельзя отводить более двух часов в неделю. Скрам-мастер должен объяснить всем, что есть временные ограничения. Если все рабочие вопросы решены быстро, то собрание можно закончить раньше обычного. Минимальной продолжительности для подобного совещания не предусмотрено.

Оценка задач

Оценивая сложность работы не нужно переусердствовать. Процесс планирования нуждается не в точной, а хотя бы в приблизительной оценке сложности разработки. Команда должна не только понять цель спринта, но и сопоставить цель с возможностями своего коллектива.

Для оценки сложности можно использовать привычные всем размеры одежды (L, XL, XXL). Конечно, это никак не дает гарантию точности, но все же.

Чтобы оценка сложности была более точной, нужно взаимопонимание. Участники команды должны открыто делиться своим мнением и не бояться задавать вопросы владельцу продукта.

Критика по отношению к команде после завершения работы может привести к тому, что при планировании следующего спринта прогнозы будут менее оптимистичными. Это поможет команде избежать повторения ошибки и обезопасит ее от негативной оценки в будущем.

Оценка сложности в очках, баллах и часах

Обычно команды разработчиков оценивают сложность своей работы во времени. Но некоторые команды Agile выбирают оценку сложности в баллах или очках. Это лучше показывает общую сумму затрат, необходимых для реализации элемента бэклога или выполнения другой поставленной задачи.

Баллы начисляются исходя из сложности и объема работы. Кроме того, учитываются возможные риски. Начисление баллов по этому методу помогает эффективно разбить работу на мелкие этапы.

Регулярно используя метод начисления очков (баллов) при планировании, команды лучше и более точно понимают, сколько им понадобится времени на выполнение работы. Кроме того, здесь имеются и другие плюсы.

- Оценка времени не учитывает работу, которая прямо не относится к проекту, хотя она наверняка появится. Обсуждение рабочих вопросов через мессенджер, проведение совещаний — все это также отнимает время у членов команды.
- На выбор дат могут оказывать влияние эмоции. Начисление очков при оценке работы исключает этот фактор.
- Оценка сложности работы и, соответственно, скорость выполнения заданий, может быть разной у каждой из команд. Работа с учетом выполненных очков не может считаться каким-либо показателем скорости. То есть, нет психологического давления на команду.
- Правильно распределив трудовые затраты и сложность, можно быстро и без конфликтов разделить очки за выполненную работу между участниками.
- Количество баллов, полученных за выполнение задачи, зависит от ее сложности, а не от потраченного времени. Поэтому программисты будут думать об улучшении своей эффективности, а не о том, сколько на это уйдет времени.

Недостатком оценки сложности является то, что ее часто используют неправильно. Например, этот метод нельзя применять для оценки сотрудников.

Командам стоит применять систему начисления очков, чтобы лучше понимать объем заданной им работы и верно расставлять приоритеты.

Daily Scrum Meeting

Рабочие совещания имеют важное значение: на них участники команды делятся своим мнением, общаются и согласовывают дальнейшие действия. Ежедневные scrum-совещания также нужны для поднятия командного духа и оглашения текущих новостей.

Стендап — это краткая встреча ключевых участников проекта: владельца ПО, программистов и скрам-мастера. Структура стендапа состоит из трех вопросов.

- Что мы смогли сделать вчера?
- Над чем мы будем сегодня работать?
- Что мешает нам достичь результата?

Постановка этих вопросов стимулирует развитие и помогает выявлять проблемы в команде. Когда каждый участник сообщает, как он помогает в достижении общей цели, это улучшает взаимопонимание внутри команды.

Важно помнить, что нет единого шаблона, по которому нужно проводить стендапы. Каждая команда проводит собрания по своей модели, исходя из особенностей коллектива.

А теперь давайте обсудим, что нужно для идеального стендапа и познакомимся с примерами эффективных стендапов.

Для начала нужно выбрать время, которое подходит всем. Обычно стендапы для команд из одного офиса проводятся в начале рабочего дня — между 9 и 10 часами утра. Это дает время лучше спланировать расписание на день. Если же участники команды работают в разных регионах, тогда выбирается время, которое устроит всех. К примеру, если некоторые члены команды живут в Калифорнии и Сиднее, то тогда стендап начинается в 15:30 по калифорнийскому времени. Разумеется, стендап после обеда не каждому удобен, но зато это дает возможность поддерживать связь с коллегами по другую сторону океана.

Следите за продуктивностью стендапа. Не проводите собрание слишком долго — концентрация внимания должна оставаться на высоте. Если это возможно, проводите стендапы длительностью не более 15 минут.

Используйте мяч. Его можно бросать друг другу по очереди. Так каждый будет вовлечен в обсуждение. Эта игра хорошо помогает поддерживать внимание в группе. Применяйте командную ретроспективу. Стендапы используются во многих Agile-методиках, это никак не мешает обсуждать эффективность стендапов на ретроспективах. Кто-то собирается каждый день, другие команды — пару раз в неделю. Если команде трудно извлечь пользу из стендапа, найдите причины этого и что-то поменяйте.

Sprint review

Spring review проводится на завершающем этапе спринта. Он необходим для проверки инкремента продукта и адаптации бэклога. В ревью итогов спринта принимает участие вся scrum-команда и все заинтересованные лица. Собрание проводится в непринужденном формате для улучшения взаимодействия участников проекта.

В обзор результатов спринта входят такие элементы:

- Владелец ПО показывает, что из бэклога было выполнено, а что нет.
- Программисты обсуждают, что получилось хорошо, а где появились сложности, и как они были устранены.
- Команда разработчиков показывает результаты своей работы во время спринта, и какой инкремент продукта они получили.
- Владелец продукта делится своим мнением о бэклоге в нынешнем виде. Также он дает прогноз на следующую цель и срок ее выполнения.
- Все обсуждают, что лучше всего делать дальше на основании оценки рынка и интересов пользователей.
- Происходит обмен мнениями насчет сроков, бюджета и перспектив дополнения бэклога.

Итогом становится обновленный бэклог с новыми целями для последующих спринтов. Бэклог может быть изменен, если этого потребует возникшая ситуация.

Sprint Retrospective

Sprint Retrospective — это рабочее совещание, на котором обсуждается, как улучшить рабочий процесс. Также на нем создается план улучшений для последующего спринта. Совещание обычно проводится уже после обзора итогов спринта и занимает не больше трех часов. Ведет совещание scrum-мастер.

В основные цели Sprint Retrospective входит:

- Анализ спринта (работа участников, результаты и проблемы).
- Обсуждение возможных решений для улучшения рабочего процесса в последующих спринтах.
- Создание плана по внедрению участниками команды улучшений в процессе реализации проекта.

Scrum-мастер приглашает участников команды вносить свои предложения, как улучшить эффективность разработки. Команда обсуждает предложения и предлагает те или иные способы и приемы их реализации.

Завершая Sprint Retrospective, команда должна выделить несколько предложений по улучшению, чтобы внедрить их в следующем спринте. Поступившие предложения можно реализовать в любое время, но Sprint Retrospective дает возможность глубже проанализировать их возможную адаптацию с точки зрения команды.

Здесь мы закончим обсуждения методологии Scrum. Более подробно с ней вы можете ознакомиться в тематической документации или на своем первом рабочем месте.

< Предыдущая лекция

Следующая лекция >

− +14 +

Комментарии

популярные

новые

старые

JavaCoder

Введите текст комментария



У ЭТОЙ СТРАНИЦЫ ЕЩЕ НЕТ НИ ОДНОГО КОММЕНТАРИЯ

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ



