

Scrum

Проблема

Очень много возникает проблем на этапе создания программного обеспечения. Желание заказчика постоянно вносить изменения, проваленные сроки, из-за не точной оценки времени выполнения задач программистами, неправильное понимание разработчиком задания, постоянное психологическое давление от начальства. А когда мы презентуем готовый проект заказчику, оказывается, что он хотел получить совсем другой продукт на выходе. А как это все получилось? Перенесемся на начальный этап. Мы встречаемся с заказчиком, выясняем, что ему от нас требуется и погружаемся с головой в разработку на длительный срок, при этом не общаясь с заказчиком. Эти и еще ряд проблем может помочь решить методология разработки ПО «Scrum».

Scrum - методология разработки программного обеспечения, на которой строится процесс создания ПО, позволяющий за некоторое количество итераций, называемые «Спринтами», предоставить заказчику продукт, с некоторым функционалом, который имеет наивысший приоритет. В целом скрам позволяет сделать процесс разработки очень гибким.

Фундамент скрама:

Все разработчики находятся в одном помещении.

Срок выполнения задачи задают разработчики.

Роли:

Скрам-мастер - по-сути это человек, который следит за правильным выполнением принципов скрама. Помогает правильно скорректировать процесс скрама.

Владелец продукта - заказчик или его представитель, который заинтересован в качественном конечном продукте.

Скрам-команда - команда разработчиков конечного продукта. Размер команды от 5 до 9 человек. Никто кроме команды не может вмешиваться в процесс разработки на протяжении спринта.

С чего начинается Скрам?

Планирование Спринта.

Мы встречаемся с заказчиком и составляем список требований к проекту (Клиент зашел сюда, тыкнул туда, открылось это), при этом присваиваем каждому элементу списка степень важности. Также в дальнейшем степень важности может меняться. Еще одной немаловажной вещью является определение цели спринта. (Админ панель, профиль). Разбить спринт на более мелкие подзадачи. (Создание дизайна, верстка страницы, создание таблиц в БД) Затем нужно определить длину спринта (2-4 недели). Далее идет оценка времени разработчиками. При оценке времени, так же учитывается фокус-фактор программиста. Т.е если рабочий день длится восемь часов, то в основном из этого времени он сфокусирован на работе только 5-6 часов. Так же, очень крутая вещь, которая мне понравилась - это Покер планирование. (Похоже я наконец-то понял, для чего в лабе лежат эти карты с числами фиббоначи) Разработчики садятся за стол, каждому разработчику дается колода карт с цифрами, кто-то объясняет задачу, и каждый разработчик для себя решает, сколько бы он потратил времени на выполнения данной задачи и кладет карту шапкой вверх, чтобы заранее никто не видел, какую карту он туда положил (В команде есть Team Lead, который оценил задачу в 1 день, а Junior думает Тим лид оценил задачу в 1 день, то и я тоже управлюсь в 1 день), затем все одновременно вскрывают карты, тем самым становится ясно, сколько именно времени нужно потратить на выполнение этой задачи, либо возможна такая ситуация, когда все выкинули два - три дня, а кто-то 6, это может означать, что этому

человеку непонятна задача или он видит в этой задачи те трудности, которые не увидели другие.

Запуск спринта

Все таски находятся на доске, они должны быть на виду. Задачи упорядочены по важности, берем приоритетные, каждый человек имеет взять право таск, т.к Скрам подразумевает, что почти каждый должен уметь заменить человека, если вдруг он заболел или уволился. (Например, если я всегда писал на Java, то мне может достаться таск «Написать модуль на Node.js»). Мне кажется, что это не совсем верно, например, если у нас в команде есть чудак, который уже 10 лет проектирует БД, а тут на тебе, пиши сервер на Джаве или взять дизайнера, как он сможет написать мобильное приложение? Быть может это как-то по-другому устроено.

Утренний скрам. (Ежедневные митинги)

Когда мы распланировали задачу, каждое утро начинается с того, что проводится утренний скрам, все встают у доски с задачами и отвечают на три главных вопроса:

Что делал вчера?

Что будешь делать сегодня?

Отвечая на эти два вопроса, мы будем всегда в курсе дела, как у нас продвигается работа над проектом, мы сможем точнее определить время.

Какие есть проблемы?

Программист может сидеть целый день и не знать решения возникнувшей проблемы. На утреннем скраме он может вынести эту проблему на общее обозрение, и скорее всего кто-то знает, как решается

эта проблема. И тот эффект, что люди не варятся в собственном соку, позволяет ускорить разработку.

А также в утренний скрам входят:

Переоценка задач

Получение новых задач

Выполнение задач:

На доске у задач 4 состояния:

Задачи, которые еще никто не взял

Задачи, которые кто то взял работает

Завершенные задачи

Завершение спринта

Заказчик обязателен или его представитель!

Все задачи готовы, проводится демонстрация, важный момент, за день скрам мастер делает рассылку, все собираются вместе, и проводится презентация, каждый разработчик показывает что он сделал, очень важно это мотивирует, т.е ты приходишь, показываешь что ты сделал, а оно сломалось, и тебе становится стыдно. Демонстрация нацелена на то, что ты готов все показать, ты проверяешь все заранее, дабы не было никаких косяков. Помогает улучшить качество кода и сконцентрировать программистов на определенную задачу лучше.

После демонстрации проводится **ретроспектива**.

Ретроспектива - двигатель самоорганизации.

Команда собирается вместе и задают друг другу 3 вопроса:

Что было хорошо?

Что было плохо?

Как улучшить процесс?

Тем самым, позволяя организовать внутренний процесс в команде.

Улучшить самоорганизацию.

Вывод

В целом скрам хорош тем, что нам известны сроки, очень хорошо отслеживается процесс разработки, все и у всех на виду. Повышается производительность, идет сплочение команды. Так же Scrum позволяет клиенту, без вреда для программистов вносить изменения в проект, путем постоянных встреч. Scrum прост в освоении. Ваша команда становится многофункциональной, но мне кажется с этим плюсом прирастают некоторые минусы, уходит больше времени на обучение, также возможны и некоторые затраты на это дело. Я уверен, что если ваша команда от 6 до 9 человек, то скрам при всех его достоинствах отлично подходит вам!