

SCRUM-МЕТОДОЛОГИЯ РАЗРАБОТКИ ИНТЕРНЕТ-ПРОЕКТОВ

Бичахчян Варвара Мушеговна

студент 1 курса магистратуры, кафедра экономико-математического моделирования РУДН, РФ, г. Москва

Ревина Светлана Юрьевна

научный руководитель, канд. экон. наук, доц., кафедра экономико-математического моделирования РУДН, РФ, г. Москва

Впервые методология Scrum была предложена Джефом Сазерлендом и Кеном Швабером в 1990-х гг. в виде четкого формализованного документа, который назывался «руководство по Scrum». “Scrum” – это легкий и гибкий метод управления и контроля процессами разработки продукта и программного обеспечения в быстро развивающихся и меняющихся условиях [1, с. 9].

До того, как начать проект, участники разрабатывают глобальные цели и базовую стратегию для достижения данных целей. Интересы компании заказчика проекта представляет лицо, который описывает бизнес-функции настраиваемого или внедряемого программного обеспечения. Исполнитель определяет команду разработчиков и прочие ресурсы, после идет обсуждение между сторонами примерных рамок проекта и даты начала первой итерации. В «скраме» итерация называется спринтом, и его продолжительность составляет от двух до четырех недель. У всех спринтов относительно фиксированная длительность. Это значит, что если установлена запланированная дата завершения спринта, то в случае наступления этой даты все работы данного спринта вне зависимости от того, успели ли члены команды сделать задание или нет, должны быть завершены [3, с. 146].

В модели методологии “scrum” присутствуют 3 главных компонента: роли, артефакты и мероприятия. В свою очередь, роли также подразделяются на 3 составляющие: владелец продукта, scrum-мастер и scrum-команда.

- Владелец продукта зачастую является заказчик проекта, который координирует список требований к проекту. Именно владельцем продукта определяется приоритет требований, т.е. заказчик сам определяет функции программного обеспечения, которые являются на данный момент наиболее важными и срочными для организации;
- Scrum-мастер отвечает за то, чтобы все участники проекта понимали сущность всех «скрам»-процессов. У scrum-мастера в чем-то есть схожесть с традиционным менеджером проекта, но есть одна отличительная характеристика: scrum-мастер не устанавливает сроки для разработчиков и не отдает явных приказов, а, наоборот, помогает разрешать трудности и следит, чтобы работа происходила слажено;
- Scrum-команду составляют профессионалы, которые разрабатывают в конце каждого «спринта» потенциально готовую к выпуску новую версию продукта. Как правило, команду разработчиков и программистов составляют не более 8 человек. Scrum-команда перед началом каждой итерации ставит значимую для заказчика и достижимую для команды цель, а в ходе итерации все усилия направлены на достижений данной цели с соблюдением сроков.

Далее перейдем к рассмотрению артефактов в методологии “scrum”. Принято выделять журнал спринта, журнал продукта и диаграммы сгорания (рис. 1).



Рисунок. 1. Артефакты методологии “scrum”

Журнал продукта создается после анализа потребностей для бизнеса в виде списков требований к разрабатываемому программному обеспечению. Непосредственно создателем журнала продукта является владелец (заказчик) продукта. После чего каждое требование распределяется по степени важности для заказчика. В ходе всего проекта у владельца продукта есть право на добавление новых требований или модификацию старых в журнале продукта. Именно данная возможность дает этому подходу оставаться «гибким» ко всем изменениям [4, с. 373]. Требования, выбранные на определенный спринт, заносятся из журнала продукта в журнал спринта. Рассмотрим это схематически (рис. 2).



Рисунок 2. Формирование журнала спринта

Журнал спринта в отличие от журнала продукта является на время конкретной итерации зафиксированным и утвержденным командой документом. Из журнала спринта могут исключаться задачи только при экстренных ситуациях, когда решение данных задач не будет больше нацелено на достижение успеха всего проекта [5, с. 75].

Для ведения учета объема выполненных работ и объема предстоящих работ используют диаграммы сгорания, которая позволяет оценить в рамках определенного спринта, в правильном ли русле продвигается работа команды.

В методологии «scrum» кроме спринта существуют еще 4 мероприятия: планирование спринта, ежедневные совещания, рассмотрение (обзор) итогов спринта и совещания в ретроспективе. До начала спринта выполняется его планирование. После того, как заканчиваются процессы планирования, начинается непосредственно спринт. Scrum-мастер каждый рабочий день проводит короткое совещание, которое не превышает 15 минут. Такие ежедневные совещания нацелены на то, чтобы предоставлять участникам команды возможность поделиться своими достижениями и возникшими затруднениями. Каждый из команды должен ответить на 3 вопроса [2, с. 182]:

1. Что сделано после прошлого совещания?
2. Что запланировано сделать к завтрашнему совещанию?
3. Какие есть трудности, тормозящие выполнение заданий?

В случае возникновения тех или иных проблем в конце совещания scrum-мастер стремится сделать все возможное, чтобы разрешить и устранить их. В конце каждого спринта осуществляется обзор его итогов. Полнота и качество выполненного задания определяется владельцем продукта, тем самым происходит налаживание обратной связи с разработчиками проекта (рис. 3).

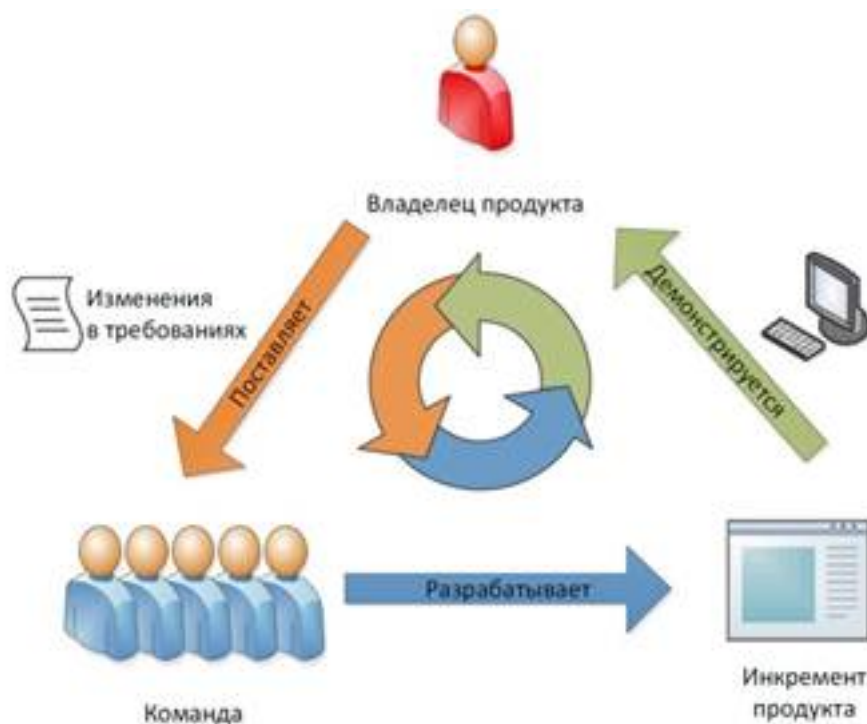


Рисунок 3. Организация обратной связи в рамках обзора итогов спринта

Ретроспективное совещание проводят также в конце спринта, но в такой встрече участвуют только scrum-команда, scrum-мастер и владелец продукта (для вспомогательной обратной связи).

Как правило, продолжительность совещания в зависимости от длительности спринта и количества обнаруженных проблем составляет от тридцати минут до четырех часов. Схематически общий scrum-процесс может быть представлен следующим образом (рис. 4):



Рисунок 4. Общая схема Scrum

Таким образом, можно прийти к выводу, что благодаря регулярному анализу выполненных работ и возможности корректировать направления проекта между спринтами методология "scrum" способствует более продуктивному достижению результатов, что, в свою очередь, позволяет намного качественнее создавать программное обеспечение.

Список литературы:

1. Ауэр К., Миллер Р. Экстремальное программирование: постановка процесса. С первых шагов и до победного конца. – СПб.: Питер, 2013. – 368 с.
2. Бычков И. В., Веденеев В. С. Применение экстремального программирования при разработке научных приложений // Математические структуры и моделирование. – 2014 г. – № 4(32). – С. 180-184.
3. ДеМарко Т., Листер Т. Человеческий фактор. Успешные проекты и команды. – М.: Символ-Плюс, 2016. – 256 с.
4. Лайкер Д., Майер Д. Практика дао Toyota. Руководство по внедрению принципов менеджмента Toyota. – М.: Альпина Пабlisher, 2016. – 586 с.
5. Физерс М. К. Эффективная работа с унаследованным кодом. – М.: Вильямс, 2016. – 400 с.