

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФГБОУ ВО «Воронежский государственный
технический университет»

составитель О.Г. Яскевич

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам по курсу

«Технология Scram»

Направление подготовки (специальность) 09.03.02 Информационные
системы и технологии, профиль (специализация) Информационные системы
и технологии цифровизации

Воронеж 2022

Лабораторная работа №1

Командообразование

Цель: генерация идей, сбор команды и выбор владельца продукта

Методика Scrum, которую разработали Джефф Сазерленд и Кен Швабер, призвана решить все эти проблемы. Scrum — это противоположность классическому поэтапному подходу, применяемому к реализации проектов. Важными характеристиками Scrum является ее гибкость и ориентированность на клиента, так как она предполагает его (клиента) непосредственное участие в процессе работы. Scrum не требует внедрения каких-либо дорогостоящих инструментов. Схему методики Scrum вкратце можно описать следующим образом:

1. Для начала необходимо выбрать «Владельца продукта» — человека, обладающего видением того, что вы собираетесь создать или достигнуть.

2. Затем нужно собрать «Команду», в которую войдут люди, непосредственно выполняющие работу. Они должны обладать навыками и знаниями, которые помогут воплотить идею владельца продукта в жизнь.

3. Нужно выбрать «Скрам-мастера» — того, кто будет следить за ходом реализации проекта, обеспечивать проведение коротких собраний и помогать команде устранять препятствия на пути достижения цели.

4. Приступая к работе, нужно создать максимально полный список всех требований, предъявляемых к продукту или цели. Пункты этого списка должны быть расставлены по приоритету. Список носит название «Бэклог продукта». Он может развиваться и изменяться на протяжении всего срока реализации проекта.

5. Участники команды должны оценить по своей системе оценок каждый пункт на предмет сложности и затрат, которые потребуются для его выполнения.

6. Затем участники, скрам-мастер и владелец продукта должны провести первое скрам-собрание, на котором они запланируют спринт — определенное время для выполнения части заданий. Продолжительность спринта не должна превышать один месяц. За каждый спринт команда зарабатывает определенное количество баллов. Команда должна постоянно стремиться к тому, чтобы превзойти в новом спринте количество заработанных баллов за предыдущий спринт, то есть ее цель — постоянно

превосходить свои собственные результаты — «наращивать динамику производительности».

7. Чтобы все участники были в курсе состояния дел нужно завести скрам-доску с тремя колонками: «Нужно сделать, или бэклог»; «В работе»; «Сделано». На доску участники клеят стикеры с заданиями, которые в процессе работы поочередно перемещаются из колонки «Бэклог» в колонку «в работе», а затем в «сделано».

8. Ежедневно проводится скрам-собрание. По выражению Джеффа Сазерленда «это пульс всего процесса Scrum». Суть его проста — ежедневно, на ходу, пятнадцать минут на то, чтобы все дали ответы на три вопроса: «Что ты делал вчера, чтобы помочь команде завершить спринт?», «Что ты будешь делать сегодня, чтобы помочь команде завершить спринт?», «Какие препятствия встают на пути команды?».

9. По завершении спринта команда делает его обзор — проводит встречу, на которой участники рассказывают, что сделано за спринт.

10. После показа результатов работы за спринт участники проводят ретроспективное собрание, на котором обсуждают, что команда делала хорошо, что можно сделать лучше, что можно улучшить прямо сейчас.

Владелец продукта должен не только понимать клиента, но и иметь концепцию ценности, которую Scrum-команда поставляет клиенту. Кроме того, владелец продукта определяет баланс между потребностями других заинтересованных сторон в организации.

Таким образом, владелец продукта должен учесть все эти входные данные и расставить приоритеты в работе. Пожалуй, это его самая значимая обязанность, поскольку противоречивые приоритеты и расплывчатые установки не только снижают эффективность команды, но и могут нарушить важные доверительные отношения между бизнесом и командой разработчиков.

В руководстве по Scrum выделяют следующие обязанности владельца продукта.

- Управление бэклогом Scrum. Это не означает, что только лишь владелец продукта помещает новые задания в бэклог продукта; но в конечном итоге именно он несет ответственность за бэклог, в соответствии с которым команда разработчиков поставляет продукт. Это значит, что владелец продукта должен знать обо всем, что находится в бэклоге продукта, а другие люди, добавляющие задачи в бэклог, должны обращаться к владельцу продукта.

- Управление релизами. Спринт является не циклом выпуска, а циклом планирования. Это значит, что Scrum-команда может осуществлять поставку в любое время. В идеале они выполняют частую поставку на протяжении всего спринта, рассматривая во время обзора итогов спринта реальное использование и отзывы клиентов. Однако непрерывная поставка не всегда возможна и требует других моделей выпуска. Владельцу продукта важно знать, когда определенные функции могут и должны быть выпущены.

- Управление заинтересованными сторонами. Любой продукт имеет множество заинтересованных сторон, от пользователей, клиентов, администрации и до руководителей организации. Чтобы поставка ценности командой разработчиков была эффективной, владельцу продукта необходимо работать со всеми этими сторонами. Это может включать большой объем коммуникаций и работы по управлению.

Scrum-мастер — это роль, которая объединяет все элементы и обеспечивает эффективную реализацию методики Scrum. На практике это означает, что такой сотрудник помогает владельцу продукта определять ценность, команде разработчиков — поставлять ценность, а Scrum-команде — становиться лучше. Scrum-мастер — это лидер-слуга, который не только формирует лояльный стиль руководства, но и ежедневно его практикует.

Он помогает владельцу продукта лучше понимать и выражать ценность, управлять бэклогом, планировать работу с командой и разбивать эту работу для наиболее эффективного обучения. Команде разработчиков Scrum-мастер помогает с самоорганизацией, настраивает участников на достижение конечного результата и выполнение инкрементов, а также вместе с ними управляет блокерами. Кроме того, Scrum-мастер служит организации в целом, помогая ей понять суть методики Scrum и создать для нее благоприятную среду.

Scrum-мастер фокусируется на следующем.

- Прозрачность. Для проведения эффективного анализа и адаптации важно, чтобы нужные люди могли видеть происходящее. Но добиться этого гораздо сложнее, чем кажется. В задачи Scrum-мастера входит обеспечение прозрачности работы Scrum-команды. Для этого, например, создаются карты историй или вносятся в Confluence идеи по результатам ретроспективы.

- Эмпирический подход. Основная идея подходов Scrum и Agile состоит в том, что лучший способ планирования — работать и учиться на результатах. Эмпирический процесс непрост и требует, чтобы Scrum-мастер

обучил Scrum-команду разбивать работу на составные части, четко описывать и оценивать конечные результаты.

- Самоорганизация. Это означает, что команда разработчиков может самостоятельно наладить свой рабочий процесс. Самоорганизация приходит со временем, и для ее реализации требуется помощь и поддержка. Scrum-мастер стимулирует участников выходить из зоны комфорта и пробовать различные приемы и практики, такие как «покер делегирования». Этот прием помогает выявлять и оспаривать традиционные идеи о границах ролей и обязанностей.

- Ценности. В Scrum выделяют пять ценностей (смелость, сосредоточенность, ответственность, уважение и открытость) не потому, что они хороши, а по той причине, что они создают атмосферу безопасности и доверия. Эта атмосфера необходима для успешного развития гибкости. Каждый участник Scrum-команды обязан следовать этим ценностям, а Scrum-мастер активно поддерживает такие начинания и напоминает всем об их важности.

Scrum-мастер помогает владельцу продукта при планировании спринта и обзоре его итогов, чтобы в коллективе сохранялось четкое понимание ценности и направления. Он помогает команде разработчиков в повседневной работе по методике Scrum, следя за выполнением работы и устранением блокеров. Он также отвечает за работу с блокерами, которые команда не может устранить самостоятельно. Scrum-мастер следит за тем, чтобы Scrum-команда видела все возможности для развития, а ретроспектива давала набор четко сформулированных достижимых результатов.

Лабораторная работа №2

Спринт

Цель: планирование спринта на две недели

Спринт — это короткий временной интервал, в течение которого scrum-команда выполняет заданный объем работы.

«При использовании scrum продукт разрабатывается в ходе нескольких итераций с фиксированной продолжительностью, которые называются спринтами и разбивают большие сложные проекты на небольшие задачи», — говорит Меган Кук, менеджер группы товаров для Trello Software в Atlassian.

«Спринты повышают управляемость проектов, позволяют командам поставлять высококачественные продукты быстрее и чаще, а также обеспечивают большую гибкость при адаптации к изменениям».

Установки scrum — прозрачность, проверка и адаптация играют главную роль в концепции спринтов.

Чтобы запланировать предстоящий спринт, нужно провести собрание по планированию спринта. Планирование спринта — это мероприятие, на котором команда сообща отвечает на два основных вопроса: какую работу можно выполнить в этом спринте и как она будет выполняться?

Выбором подходящих рабочих задач для спринта занимаются совместно владелец продукта, Scrum-мастер и команда разработчиков. Владелец продукта определяет цель спринта и задачи из бэклога продукта, при выполнении которых она будет достигнута.

Затем команда создает план, согласно которому будут выполняться задачи бэклога, чтобы к окончанию спринта вся работа была завершена. Выбранные рабочие задачи и план по их выполнению называется бэклогом спринта. К концу совещания по планированию спринта команда готова приступить к работе. Для этого необходимо просто выбирать задачи из бэклога спринта и менять их статус с «В работе» на «Готово» по мере завершения работы.

В течение спринта команда собирается на ежедневные Scrum-совещания (стендапы), чтобы обсудить ход работы. Такие совещания нужны, чтобы выявить блокеры и проблемы, которые могут повлиять на достижение цели спринта.

По окончании спринта команда показывает выполненную работу на обзоре итогов спринта. Здесь можно продемонстрировать итоги работы заинтересованным сторонам и другим участникам команды до того, как они попадут в рабочую среду.

Завершите цикл спринтов на моем любимом собрании — ретроспективе спринта. Здесь команда может определить области, требующие улучшения в следующем спринте. С этими сведениями можно начинать следующий цикл спринта. Вперед!

Что нужно делать.

- Убедитесь, что команда понимает цель спринта и способ измерения успеха. Это важно, чтобы все участники настроились и двигались к общей цели.

- Убедитесь, что у вас есть четкий и понятный бэклог с приоритетами и зависимостями. Если бэклог ведется неправильно, он может превратиться в большую проблему и сорвать рабочий процесс.

- Убедитесь, что вы правильно оцениваете скорость работы команды и учитываете такие события, как отпуска и общие собрания.

- Используйте собрание по планированию спринта, чтобы расширить описание актуальных задач дополнительными подробностями. Поощряйте участников команды за наброски общего плана для всех историй, багов и задач, которые входят в спринт.

- Отбрасывайте задачи, в ходе которых команде не удастся решить вопросы с зависимостями. Такие задачи включают работу другой команды, дизайн и юридическое подтверждение.

- Наконец, после принятия решения или составления плана убедитесь, что есть человек, который фиксирует эту информацию в инструменте для управления проектами или совместной работы, например в заявках Trello. В таком случае позднее каждый сможет быстро прочесть о решении и его обосновании.

И если уж вы работаете над тем, чтобы стать сильным специалистом по Scrum, выполняя рекомендации, ознакомьтесь также с действиями, которые выполнять не следует.

В Scrum спринт — это фиксированный отрезок времени, за который выполняется вся работа. Но прежде чем начать действовать, спринт нужно спланировать. Необходимо определить продолжительность и цель спринта, а также отправную точку работы. На совещании по планированию спринта определяется круг основных задач и вектор развития спринта. Если совещание проводится правильно, команда покидает его мотивированной, заряженной решать амбициозные задачи и нацеленной на успех. Плохие планы спринтов могут загнать команду в тупик, если ей сообщают нереалистичные ожидания.

- **«Что».** Владелец продукта ставит основную задачу (или цель) спринта и рассказывает, какие задачи из бэклога нужно выполнить, чтобы достичь этой цели. Scrum-команда решает, что удастся выполнить за грядущий спринт и что для этого нужно сделать в течение спринта.

- **«Как».** Команда разработчиков составляет план работ, которые необходимо выполнить для достижения цели спринта. Итоговый план спринта согласуется между командой разработчиков и владельцем продукта с точки зрения создаваемой ценности и затрачиваемых усилий.

- **«Кто».** Планирование спринта невозможно без участия владельца продукта и команды разработчиков. Владелец продукта ставит цель, которая зависит от искомой ценности. Команда разработчиков должна понять, может ли она достичь этой цели (и как это сделать). Если одна из этих сторон отсутствует на мероприятии, планировать спринт становится практически невозможно.
- **Входные данные.** Прекрасной отправной точкой для планирования спринта является бэклог продукта, так как многое из него можно включить в состав текущего спринта. Команда также должна проанализировать текущую работу, которую она выполняет в рамках инкремента, и оценить собственные возможности.
- **Результаты.** Самое важное, с чем команда может покинуть совещание по планированию спринта, — это понимание того, что нужно достичь за спринт и как начать продвижение к этой цели. Отобразить это можно с помощью бэклога спринта.

Лабораторная работа №3 Бэклог продукта

Цель: эффективный бэклог продукта в Trello

Бэклог продукта — это перечень рабочих задач, расположенных в порядке важности, для команды разработчиков. Его составляют на основе дорожной карты и требований в ней. Наиболее важные задачи расположены в начале бэклога продукта, чтобы команда понимала, какую работу следует выполнить в первую очередь. Скорость, с которой команда выполняет задачи бэклога, не зависит от желаний владельца продукта, а он, в свою очередь, не оказывает давления на команду.

Храните все в одном трекере задач. Не используйте несколько систем для отслеживания багов, требований и рабочих задач по разработке. Есть задача для команды разработчиков? Тогда информация о ней должна быть в одном бэклоге.

Владелец продукта составляет из этих пользовательских историй единый список для команды разработчиков. Владелец продукта может упорядочить истории так, чтобы команда сначала выполнила один эпик полностью (слева). Как вариант, может быть важнее сначала протестировать бронирование билетов со скидкой, а для этого нужно реализовать истории из нескольких эпиков (справа). Оба варианта представлены ниже.

Что может повлиять на то, как владелец продукта расставляет приоритеты?

- Важность для клиента
- Необходимость в обратной связи
- Относительная сложность реализации
- Тесная взаимосвязь между рабочими задачами (например, сделать «Б» будет проще, если сначала сделать «А»)

Хотя расстановкой приоритетов занимается владелец продукта, в процесс вовлечены и другие стороны. Успешность бэклога зависит от вклада и обратной связи, предоставленной клиентами, дизайнерами и командой разработчиков. Совместными усилиями они должны добиться оптимальной рабочей нагрузки между всеми участниками и обеспечить поставку продукта.

После создания бэклога важно регулярно корректировать его по мере выполнения программы. Владельцы продукта должны пересматривать бэклог перед каждым собранием по планированию итерации, чтобы уточнить расстановку приоритетов и внести изменения на основе выводов, сделанных в результате последней итерации.

Когда бэклог становится достаточно большим, владельцам продукта приходится выделять в нем группы краткосрочных и долгосрочных задач. Краткосрочные задачи нужно досконально проработать, прежде чем присвоить им этот статус. Для этого нужно составить полноценные пользовательские истории, обсудить все детали совместной работы с дизайнерами и разработчиками и оценить сложность разработки. Долгосрочные задачи могут быть продуманы не до конца, однако если команда разработчиков даст им приблизительную оценку, это поможет расставить приоритеты. Ключевое слово здесь — «приблизительная». Оценки меняются, когда команда получит полное понимание долгосрочных задач и приступит к их выполнению.

Бэклог служит связующим звеном между владельцем продукта и командой разработчиков. Владелец продукта может в любое время менять приоритеты в работе на основе обратной связи от клиентов, более точных прогнозов и новых требований. И все же следует избегать изменений в ходе работы, потому что они мешают команде разработчиков, негативно влияя на концентрацию, рабочий процесс и моральный дух.

Кроме того, на основе бэклога продукта планируются итерации. В бэклог должны быть включены все рабочие задачи: пользовательские истории, баги, изменения в дизайне, технический долг, запросы клиентов, действия,

намеченные по итогам ретроспективы, и т. д. Так рабочие задачи каждого участника будут рассмотрены на общем обсуждении перед каждой итерацией. Затем участники команды и владелец продукта с полным пониманием объемов задач и учетом обоюдных интересов принимают решения до начала итерации.

Владельцы продукта определяют важность рабочих задач в бэклоге, в то время как команда разработчиков определяет скорость работы над ними. Новым владельцам продукта, которые привыкли торопить команду, такой подход может оказаться не по душе. Подробнее см. в нашей статье о лимитах объема незавершенной работы и рабочем процессе.

Лабораторная работа №4

Совещания

Цель: совещание по результатам спринта

Для совещания важны критерии «готовности» работы. В эффективных командах выполнение каждой рабочей задачи подчиняется четко сформулированным методикам и принципам разработки. Чтобы оценить свой процесс выполнения работы и убедиться, что он идеально подходит команде, ответьте на следующие вопросы.

- Хорошо ли прорабатывают пользовательские истории владелец продукта, дизайнер и команда разработчиков, прежде чем начать работу над ними?
- Все ли понимают ценности и принципы разработки, принятые в команде, и следуют им?
- Установлены ли для проверки кода, автоматических тестов и непрерывной интеграции понятные критерии и требования, на основе которых формируется устойчивая культура разработки?
- Много ли обнаруживается багов после того, как команда завершает работу над историей? Другими словами, действительно ли завершена работа, отмеченная как «завершенная»?

Подходы к обеспечению качества и выполнению работы, сложившиеся в команде, должны стоять во главе каждой пользовательской истории, рабочей задачи по разработке и бага. Через призму этих подходов формируется отношение команды к созданию ПО и его поставке.

Четко сформулированные критерии готовности помогают командам сосредоточиться на конечной цели по каждой рабочей задаче. Когда владелец продукта добавляет работу в бэклог команды, в его обязанности входит и

определение критериев приемки. Какой должна быть завершенная пользовательская история? Команда Trello прописывает критерии приемки и примечания к тестированию там же, где находится описание пользовательской истории в Trello. Так все участники команды получают полное представление о ходе выполнения каждой задачи. Что такое критерии приемки и примечания к тестированию?

- **Критерии приемки** — это метрики, по которым владелец продукта может определить, удовлетворяет ли результат работы над пользовательской историей его требованиям.

- **Примечания к тестированию** — это краткие, предметные инструкции от команды по контролю качества, следуя которым специалист по разработке может написать более качественный функциональный код и автоматические тесты.

Когда задачи четко сформулированы, любой может успешно выполнить свою работу. В Trello можно без труда добавить дополнительные поля.

Обзоры итогов по смыслу отличаются от ретроспектив, поэтому нужно проводить их после итерации и перед ретроспективой. Это совещание открыто для посторонних, но обычно в нем участвуют владелец продукта, команда разработчиков в полном составе и Scrum-мастер. В рамках этого совещания рекомендуется уделять каждой итерации от 30 минут до часа.

Они предназначены в первую очередь для объединения команды. В них нет элемента конфронтации, поскольку это не экзамен, а коллективное мероприятие для всей команды. На нем сотрудники рассказывают о своей работе, отвечают на вопросы и узнают мнение других.

Если обзор итогов спринта в команде проходит недостаточно конструктивно, это может быть сигналом о следующих проблемах.

- Команда берет на себя слишком много работы и не успевает выполнить ее за итерацию.

- У команды накопился технический долг.

- При разработке функций не учитывается долгосрочная перспектива, поэтому в базу кода могут проникнуть новые баги.

- Подходы команды к разработке не отлажены до конца.

- Владелец продукта имеет привычку менять приоритеты в течение итерации, поэтому изменения объема работы выбивают команду разработчиков из колеи.

Следует учитывать, что неудачные итерации случаются у каждой команды. Определите причины изменения итерации в ходе ретроспективы

команды и составьте план действий, которые предупредили бы возникновение этих проблем в следующем спринте.

Лабораторная работа №5

Задачи Scrum-мастера

Цель: оценка эффективности Scrum-мастера

Scrum-мастера — это фасилитаторы scrum, легковесной agile-платформы, которая фокусируется на сгруппированных по времени итерациях, называемых спринтами. Как и фасилитаторы, scrum-мастера выступают в качестве коучей для остальной части команды. Руководство по scrum называет их «лидерами-служителями». Хорошие scrum-мастера преданы основам и ценностям метода scrum, но остаются гибкими и показывают команде возможности для улучшения рабочего процесса.

1. **Стендапы.** Практикуйте ежедневные стендапы, или scrum-совещания, по мере необходимости.

2. **Собрания по планированию спринтов или итераций.** Берегите команду от чрезмерной нагрузки и расширения области работы. Участвуйте в оценке работы и разбивайте ее на подзадачи.

3. **Обзоры итогов спринтов.** Принимайте участие в совещаниях и фиксируйте обратную связь.

4. **Ретроспективы.** Отмечайте, что можно улучшить, и планируйте конкретные действия на следующий спринт.

5. **Администрирование досок.** Станьте администратором доски scrum. Поддерживайте карточки в актуальном состоянии, а инструмент scrum (например, Trello Software) — в рабочем.

6. **Встречи один на один.** Разговаривайте индивидуально с каждым участником команды и заинтересованными сторонами, если необходимо. Улаживайте разногласия по поводу процессов и стиля работы в команде. Многие специалисты scrum против встреч один на один, поскольку убеждены, что все должно обсуждаться во время стендапов. Однако участники многих, особенно новых, команд предпочитают регулярно встречаться с определенными коллегами один на один. Scrum-мастер может признать, что такие личные встречи важны для развития команды и знакомства ее участников друг с другом.

7. **Внутренние консультации.** Scrum-мастерам следует быть готовыми к обмену мнениями с другими участниками команды и внутренними

заинтересованными сторонами, чтобы вместе решать, как лучше организовывать работу scrum-команды.

8. **Блокеры.** Scrum-мастер устраняет внешние блокеры и помогает команде справиться с внутренними препятствиями, улучшая рабочие процессы.

9. **Интенсивная работа.** Если scrum-команда не погружена в рабочий процесс, это сигнал для scrum-мастера. Возможно, стоит починить сломанные компьютеры, передвинуть столы или, может быть, просто отрегулировать температуру в офисе. Scrum-мастер должен быть готов создавать условия для работы команды любыми подходящими способами — даже принести кофе и что-нибудь перекусить, если это решит проблему.

Любой scrum-тренер учит, что в scrum-команде должен быть scrum-мастер. Без него вы будете работать не совсем так, как предполагает подлинный scrum. Подобную ситуацию часто называют «скрам-бат» (scrum-but).

Когда вы только начинаете работать по методу scrum, очень полезно взять на роль scrum-мастера человека, который уже видел, как ведется такая работа. Еще лучше, если он видел множество примеров того, как это работает. Именно поэтому scrum-мастеров часто нанимают как консультантов, а не штатных сотрудников.

Но все scrum-команды разные. Многие опытные команды выполняют все перечисленные обязанности, действуя как единое целое. Они гордятся и наслаждаются совместным управлением процессом. Роль scrum-мастера дрейфует между участниками команды, которые проводят стендапы и ретроспективы по очереди.

Для других команд будет лучше, если один и тот же человек изо дня в день выполняет эту роль.

К сожалению, неправильное понимание роли scrum-мастера часто приводит к тому, что существующие менеджеры начинают считать эти обязанности своими. Чтобы лучше понять, почему это может вылиться в проблему, давайте сравним роль scrum-мастера с ролями, не относящимися к scrum, которые, возможно, уже есть в вашей организации, и обсудим, почему важно не смешивать эти роли.

При найме scrum-мастера стоит принять во внимание еще одно очень важное соображение. Делайте это только в случае, если организация активно настроена на работу по методу scrum и инвестирует в этот процесс. Все вышеперечисленные роли могут управлять командой разработчиков

множеством способов, но scrum-мастер эффективен только при абсолютной приверженности методу scrum. Этим все сказано.

Благодаря scrum-мастеру, который помогает каждой команде управлять процессом, организация в целом может получить несколько серьезных преимуществ. Помимо регулярной поставки ценного продукта клиентам (основная цель scrum), участники команды и менеджеры смогут сосредоточиться на том, что они делают лучше всего. Менеджеры по продукту могут сфокусироваться на стратегии, разработчики получают возможность написать свой лучший код, а Кайл из отдела продаж наконец-то совершит звонок, до которого не доходили руки. На что это похоже? Это похоже на высокофункциональный scrum, музыку для наших ушей.

<https://www.atlassian.com/ru/agile/scrum/roles>