

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ SCRUM-ПОДХОДА В УПРАВЛЕНИИ IT-ПРОЕКТАМИ

Огнева Д.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Кашникова И.В. – канд. физ.-мат. наук

На сегодняшний день традиционные методы проектного управления устаревают, именно поэтому необходимо внедрять во многие сферы жизни современные гибкие подходы, одной из которых является Scrum, который позволяет повысить эффективность и продуктивность работы в целом. В данной статье представлены основное содержание Scrum-подхода, выявлены и проанализированы проблемы применения Scrum в управлении IT-проектами, дано практическое обоснование возникновения выявленных проблем.

Компании, занимающиеся внедрением информационных технологий или же связанные с IT-проектами, в своей деятельности должны учитывать и использовать передовые методы, знания и технологии для повышения конкурентного преимущества по сравнению с другими компаниями. Одними из таких технологий, дающих компании возможность успешно и качественно закрывать проекты, учитывая все пожелания заказчика, являются гибкие подходы к управлению проектами, Agile методологии, в число которых входит и фреймворк Scrum.

Scrum – это самый популярный фреймворк Agile, который помогает людям, командам и организациям создавать ценность с помощью адаптивных решений комплексных проблем [1].

Scrum прост для понимания, и для его применения есть подробное руководство – Scrum Guide, помогающее командам эффективно создавать сложные продукты с высокой ценностью и в условиях высокой неопределённости. При таком подходе команды работают по спринтам от 1 до 4 недель, состоят из 5-9 человек и самостоятельно распределяют задачи между собой. В начале каждого спринта разработчики планируют задачи, а в конце — демонстрируют готовую часть продукта. Заказчики дают обратную связь, и команда начинает делать следующую функциональность или дорабатывает текущий инкремент.

Scrum требует, чтобы Scrum Master способствовал возникновению среды, в которой [2]:

1. Product Owner упорядочивает работу по решению комплексной проблемы в Product Backlog.
2. Scrum Team в ходе Sprint превращает выбранную работу в Increment, несущий ценность.
3. Scrum Team и заинтересованные лица инспектируют результаты и вносят правки для следующего Sprint.
4. Повторение.

Цель данной работы – рассмотреть проблемы применения Scrum-подхода в управлении современными IT-проектами.

В настоящее время Scrum используется очень широко в рамках IT-проектов, при внедрении и использовании которого, часть компаний добивается блестящего успеха, а для других внедрение не приносит ожидаемых преимуществ [3]. Рассматривая на практике одну из самых известных Agile практик Scrum, можно понять, где он действительно не работает и почему это происходит. Такой анализ может помочь не допустить множество ошибок в начале пути и выстроить либо эффективный процесс, либо отказаться от идеи внедрения фреймворка Scrum.

Первоначальная проблема применения Scrum подхода заключается в том, что его используют за пределами его границ применимости, и, соответственно, в этом случае он не работает. Граница применимости Scrum заключается в степени неопределенности. Если степень неопределенности крайне мала, то такой подход в управлении не может быть использован эффективно. Рассмотрим системы, где применение Scrum подхода является совершенно бессмысленным и принесёт большую потерю в деньгах и времени [4]:

1. Упорядоченные простые системы. Они понятны, для их решения у команды есть опыт. Уже на старте понятно, что получится в результате, за какие деньги и в какие сроки. Нередко есть инструкция по выполнению этого действия.

2. Упорядоченные сложные системы. В этом случае заранее не понятно, как решать проблему. Задача не уникальна, однако именно ее команда ранее не решала. Показательным может быть пример создания типового сайта, у команды есть согласованное ТЗ от заказчика, шаблон, план работ, а изменения в процессе работы скорее всего не потребуются. И на выходе получится готовый продукт, по которому предлагаются корректировки и поддержка некоторое время.

3. Хаотичные системы. Здесь хорошо работает экспериментальный подход. Хаотичные – абсолютно новые задачи, которые никто и никогда не решал раньше. Попытка разобраться с такой системой – путь к инновациям. Любой способ решения (стабилизации системы) будет новым. Примером такой системы в реальной жизни является стартап.

Scrum можно применить только в комплексной, сложной системе, что трактует само его определение: «фреймворк, который помогает людям, командам и организациям создавать ценность с помощью адаптивных решений комплексных проблем». Если проецировать систему на задачу, то в комплексных системах речь идет о непонятной, сложной задаче, но при этом команда с подобной задачей уже сталкивалась и имеет опыт ее решения. Например, у заказчика есть видение продукта, но команда ранее этого не делала.

Следующая проблема, с которой зачастую сталкивается проект, использующий Scrum, является раннее вовлечение заказчика. Однако «не всякая наполовину прожаренная булочка» стоит времени клиента. Scrum хорошо работает только в тех случаях, когда присутствует много пользовательской функциональности. Если проект преимущественно состоит из бэкэнд логики, то невозможно дать заказчику сервер, который умеет принимать данные; затем сервер, который умеет

отдавать данные; после чего сервер, который умеет принимать и отдавать данные и так далее, а в конечном итоге сервер, который работает эффективно и надежно. Заказчик будет ждать до тех пор, пока не будет получен нормальный сервер, промежуточные результаты не несут для него никакой ценности.

Решая использовать Scrum нужно быть готовым, что такой подход никогда не даст понять сроки. Scrum отвергает экспертные оценки и считает, что попытка прогнозировать сроки — это самообман. В соответствии со Scrum команда не должна пытаться придумать сколько будет длиться определенный объем работы, команда должна померить это по факту.

При применении Scrum, требования заказчика собираются и складываются в беклог. Единственный способ оценить продолжительность в Scrum — начать работать. Хотя бы 2–3 спринта. Тогда становится понятна «velocity» («мощность») команды – то, сколько задач и какой сложности команда может выдать за спринт. По этой «мощности» определяют — сколько еще понадобится времени, чтобы сделать все оставшиеся в текущем бэклоге задачи в случае, если он не будет изменяться. На старте, пока работа не началась, определить сроки невозможно, так как это запрещено подходом напрямую. То есть команда не имеет информации о velocity, а значит не может предоставить каких-либо оценок.

Однако отсутствие сроков при начале работы не является ключевой проблемой. Спустя несколько спринтов команда определит свое velocity, но сроков заказчик все равно не получит. Причина такой проблемы заключается в коллективной ответственности. Scrum считает продолжительность, оттолкнувшись от «мощности команды». Velocity не рассчитывают на человека, этот показатель относится только к команде. Учитывая то, что люди на проекте меняются: одни уходят в отпуск, другие пропускают по болезни, кто-то увольняется, а в команду вливаются новички. Каждая такая перестановка как-то влияет на velocity. Никто не знает, как.

Практика свидетельствует, что люди на проекте перемещаются каждые 2 месяца [5]. Из чего следует, что, используя Scrum, не может быть актуального значения «мощности» команды. Срок окончания проекта никогда не будет известен, так как velocity в очередной раз изменяется, и чтобы ее измерить требуется 2-3 спринта. Единственная метрика, позволяющая прогнозировать сроки в Scrum, не работает в реальной жизни. Что говорит о том, что, если есть жесткий срок окончания разработки, определяемый контрактом, Scrum работать не будет. Scrum будет работать только там, где сроки не важны.

Таким образом, чтобы правильно внедрить Scrum, важно следовать четким правилам и работать в рамках практик фреймворка. Scrum не содержит лишних правил или практик. Чтобы он работал как задумано, его нужно реализовывать целостно и в пределах его границ применимости, следуя тем правилам, что описаны в Guide. Частичное применение правил, описанных в Guide, равносильно отказу от Scrum.

Список использованных источников:

1. Отчет об исследовании Agile в России 2021 VersionOne. ScrumTrek. – URL : <https://scrumtrek.ru/blog/agile-scrum/7205/otchet-issledovanie-agile-v-rossii-2021/>
2. Сазерленд Д. Scrum. Революционный метод управления проектами. – М.: Манн, Иванов и Фербер, 2016.
3. Швабер К., Руководство по Scrum / К. Швабер, Д. Сазерленд. – URL : <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Russian.pdf>
4. Гепрепт Д. В., Agile mindset & frameworks / Higher School of economics, 2017.
5. Селиховкин И., Черная книга Скрам, 2018.