

Agile и Scrum - обзор и анализ основных принципов

Mihai Valentin, Power IT, 2022

Введение

На сегодняшний день существует множество стратегий и методологий управления проектами. Из заметок отдельных менеджеров и управленцев они сформировались в отдельную область изучения, развития. Как только организация сталкивается с потребностью в создании четких процессов или улучшении текущих в ее поле зрения попадают самые популярные и простые решения.

На текущий момент такими являются Agile методологии и в частности Scrum. Однако перед внедрением новых процессов следует провести тщательный анализ и хорошо ознакомиться с их основными принципами и ценностями.

Данная работа посвящена решению этой задачи и являет собой краткий обзор и анализ принципов Agile и методологии Scrum в первом приближении.

Agile

Термин “Agile” описывает группу методологий по управлению разработкой проектов. Основные принципы и свойства присущее этой группе описаны в одноименном Agile манифесте. Его основными положениями являются:

- люди и взаимодействие важнее процессов и инструментов;
- работающий продукт важнее исчерпывающей документации;
- сотрудничество с заказчиком важнее согласования условий контракта;
- готовность к изменениям важнее следования первоначальному плану.

Гибкие методологии отлично подходят для управления разработкой программного обеспечения, однако могут применяться и в других сферах [1].

Scrum

Scrum является одним из многочисленных вариантов реализации манифеста Agile. Он был разработан в начале девяностых двумя ветеранами IT индустрии Джеффом Сазерлендом и Кеном Швабером. Они входят в число людей, которые подписали Agile манифест.

Данная методология описывает эмпирический и циклический процесс, где опыт прошлых итераций применяется для улучшения следующих. Итерации имеют одинаковую длительность и называются спринтами. Результатом спринта является инкремент - готовое обновление для проекта.

Вся теоретическая часть Scrum была описана создателями в Scrum Guide. Это описание является довольно строгим и находится на грани противоречий с Agile принципами [2].

Принципы в Scrum

Все практики и теория Scrum стоит на трех столпах (принципах): прозрачность, инспектирование, адаптация.

Прозрачность (или *transparency*) говорит о том, что все процессы, цели, успехи и неудачи должны быть открыты всем участникам команды. Такая прозрачность работы может быть достигнута при наличии должной откровенности по отношению к команде и себе.

Инспектирование (или *inspection*) предполагает постоянный анализ предыдущих результатов и текущего состояния проекта, что позволяет делать прогнозы и своевременно находить узкие места в рабочем процессе.

Адаптация (или *adaptation*) подразумевает применение опыта и результатов анализа для улучшения общего процесса и команды в частности [2].

Ценности в Scrum

Кроме принципов в Scrum есть набор ценностей, которые должна уважать и развивать команда: ответственность, фокусирование, открытость, уважение и храбрость.

Каждая ценность из списка говорит сама за себя, однако иногда их понимают не до конца верно. Так например, *ответственность* означает, что команда и каждый ее участник отвечают за достижение долгосрочных и краткосрочных целей. Это исключает ситуации, когда один разработчик отвечает лишь за его персональные задачи.

Фокусирование на целях, успехе, продукте и личном вкладе является весьма важным, так как только заинтересованная команда сможет создавать творческие и надежные решения.

Открытость предполагает готовность команды к экспериментам, изменениям и испытаниям, которые приносят непостоянство бизнеса. Также каждый участник команды должен быть открыт перед другими.

Очевидно, что *уважение* ко всем коллегам и самому себе необходимо в таких методологиях, ведь люди и коммуникация важны (согласно манифесту Agile).

Для применения практик Scrum и качественного продвижения к цели необходима *храбрость* и смелость. Это нужно для того, чтобы в трудный момент признать ошибку или попросить помощи; принять вызов вечно меняющегося бизнеса; найти силы взять на себя ту самую ответственность [2].

Цель и ее достижение

Для достижения цели ее необходимо обозначить. Постановка целей является очень важным элементом в разработке продукта.

В Scrum есть два основных типа целей: цель продукта и цель спринта.

Цель продукта, очевидно, является конечной целью, которую преследует команда разработки и заинтересованные лица.

Цель спринта - это более специализированная цель, достижение которой приближает всех к глобальной цели. Цели на спринт, как правило, не должны меняться в ходе спринта.

Цели считаются достигнутыми, когда они переходят в состояние завершения (*done*). Критерии, по которым определяется достижение этого состояния объединяются в "определение готовности" (*definition of done*). Критерии должны быть согласованы понятно и доступно для всей команды [2].

Классификация целей по масштабу

Для более удобного гранулирования целей, условно, их можно представить классическими для типами задач:

1. *scope* - это общая область в разработке - цель продукта;
2. *epic* - это часть области, которая имеет более узкую направленность и может рассматриваться независимо;
3. *user story* - это результат разбиения *epic* на более мелкие самодостаточные части, которые описывают какую-то одну полезную функциональность;
4. *task* - описывает конкретную, как правило, техническую задачу, которая является подзадачей для *user story*.

Таким образом, достижение всех целей более низкого уровня автоматически приводит к достижению родительских целей. В процессе декомпозиции не стоит забывать об основных принципах Agile и ценностях Scrum. Это позволит избежать строгого распределения ресурсов в отдельные подпроцессы разработки (команда должна работать совместно) [2].

Команда

Scrum команда должна быть относительно маленькой, чтобы количество связей между ее участниками не превращало работу в бесконечное общение. Рекомендуется использовать команды до десяти человек.

Такая команда должна состоять из сильно мотивированных профессионалов, которые готовы помогать друг другу.

Ядром команды являются разработчики. Они определяют (в большей степени) что и как будет реализовано. Они отвечают за координацию и анализ текущего состояния (выявления трудностей). У разработчиков должен быть адекватный уровень автономности.

Владелец продукта (*product owner*) исполняет роль аналитика, который получая различные входные параметры (обновление от заказчика, просьбы команды, анализ рынка ...) составляет список целей и сортирует их по приоритету. Он должен быть фанатом своего продукта и иметь четкое представление о его целях, чтобы проецировать это на команду. Владелец продукта отвечает за общую направленность команды.

Для отладки и поддержки процессов, а также для организации дополнительных коммуникаций существует Scrum мастер (*Scrum master*). По большому счету, он является наблюдателем, который поддерживает стабильность Scrum. Также он должен быть ментором и тренером в одном лице, чтобы помогать решать конфликты и проводить эксперименты, тем самым, развивая команду [2].

Артефакты

Для организации работы Scrum существуют некоторые артефакты: бэклог продукта, бэклог спринта и инкремент.

Бэклог продукта (или *product backlog*) - это список обновлений, улучшений или функций необходимых для достижения цели продукта. Этот список составляет владелец продукта. Также он оценивает приоритет всех элементов и сортирует список, согласуя все с командой.

Бэклог спринта (или *sprint backlog*) - это часть бэклога продукта, которую планируется реализовать за спринт. Его формирует вся команда исходя из показателей производительности и прошлого опыта. Данный список отражает основные цели на спринт.

Инкремент (или *increment*) - является результатом работы за последнюю итерацию. Этот результат должен быть полезен в рамках продукта и

соответствовать определению готовности. Каждый инкремент добавляется к остальным представляя обновление продукта и приближение к глобальной цели [2].

Оценка работы

Для анализа проделанных работ и прогнозирования будущего продвижения (*inspection*) с последующим улучшением и развитием (*adaptation*) в Scrum существует система оценок.

Трудоемкость отдельных целей оценивается в так называемых стори поинтах (*story points*). Эта мера является субъективной для команды и основывается на предыдущем опыте разработки. Это позволяет проводить анализ в единицах отвлеченных от времени.

Однако не стоит злоупотреблять и оценивать каждое действие, ведь стори поинты и их вес должны использоваться для более комфортной оценки. Не стоит считать каждую сотую долю затрат - лучше оценить более высокий уровень работ. Оценки работы должны быть честными (*transparency*) и согласованными.

События

Для приведения всего механизма в действие существует набор событий, которые проводятся регулярно и помогают организовать работу команды.

Спринт (или *sprint*) - это основное событие, которое является контейнером для всех остальных. Именно в его рамках проводятся работы для создания нового инкремента. Спринты могут быть различной длительности, но рекомендуется рассматривать промежутки от недели до месяца. Продолжительность выбирается в начале работ и не изменяется (как правило).

Планирование спринта (или *sprint planning*) - это первый этап спринта, где команда анализирует бэклог продукта и выделяет задачи в бэклог спринта. Это событие является самым продолжительным, его длительность не должна превышать восьми часов для месячного спринта.

Ежедневная синхронизация команды (или *daily scrum, daily stand-up*) - самое короткое, но одно из важнейших событий в Scrum. Его длительность не должна превышать пятнадцати минут. Основной задачей стендапа является синхронизация команды. Также тут команда рассказывает про возникшие трудности, чтобы в дальнейшем устранить их во время спринта или в следующих итерациях (в зависимости от значимости проблем). После проведения такой встречи каждый участник получает представление о текущем

положении дел. Стоит отметить, что главными во время ежедневного скрама являются разработчики. Остальные участники должны быть пассивными слушателями.

Обзор спринта (или *sprint review*) - событие предназначенное для демонстрации результатов полученных по истечению спринта. Здесь команда показывает всем заинтересованным лицам полученный инкремент и получает отзывы.

Ретроспектива (или *sprint retrospective*) - после очередной презентации результатов и перед началом следующей итерации проводится анализ всех успехов и неудач. Это необходимо для адаптации и улучшения процесса. Результатом ретроспективы является обновление приоритетов, введение новых операций и отмена бесполезных этапов [2].

Средства анализа

Для более детального и наглядного анализа используются специальные показатели и графики. Все эти индикаторы основываются на оценках в стори поинтах. Основными показателями являются:

- мощность команды (или *velocity*) - показывает количество работ которое может быть выполнено в рамках спринта;
- график сгорания задач (или *burndown chart*) - демонстрирует сценарий идеального решения спринта и его актуальное развитие;
- график сгорания проекта (или *burnup chart*) - график, который отражает идеальное развитие проекта и его актуальное положение.

Стоит учитывать тот факт, что данные показатели хорошо работают в довольно стабильных командах и не являются исчерпывающими [3][4].

Минусы

Как и у любого инструмента - у Scrum есть минусы. При близком рассмотрении инструкций можно вовсе найти противоречия между Scrum Guide и Agile манифестом.

Во-первых, Scrum не является “серебряной пулей” - он не подходит абсолютно всем командам и проектам. Если четко следовать гайду, то на заказчика, команду и сам проект накладываются довольно серьезные и жесткие требования, которые часто бывают невыполнимыми в реальности.

Во-вторых, Scrum не подойдет для продуктов, которые невозможно разрабатывать частями при условии, что каждая часть является полезной отдельно (*increment*).

В-третьих, в нестабильных командах инструменты прогнозирования и оценки становятся необъективными, так как они полностью зависят от состояния команды и ее производительности.

В-четвертых, исполнение всех требований Scrum делает рабочую атмосферу довольно стрессовой, что может вызвать выгорание и демотивацию команды, а в худшем случае - провал всего проекта.

В-пятых, хоть Scrum и позиционируется как методология с малой долей планирования - это не так. Количество времени на планирование обратно пропорционально длительности спринтов. Так в недельном спринте, примерно, тридцать процентов времени будет потрачено на планирование (полтора дня из пяти рабочих).

Это не исчерпывающий список слабых мест данной методологии, поэтому следует провести качественный анализ всех ресурсов перед внедрением Scrum [5][6].

Вывод

Scrum является одной из самых популярных гибких методологий, однако его требования довольно строгие. Слепое следование Scrum Guide может привести к катастрофе для неподготовленных команд. Отклонение от правил не позволяет называть полученные процессы Scrum. Также не все проекты и команды будут вписываться в такие сценарии как Scrum. Поэтому стоит воспринимать такие методологии критически, анализировать имеющиеся ресурсы для полезного применения и реального улучшения процессов.

Scrum будет хорошим выбором для начала внедрения гибких методологий если команда сплоченная и стабильная, продукт может поставляться полезными частями, а сфера деятельности позволяет соответствовать концепциям методологии. В идеальных условиях Scrum способствует развитию команды и организации в целом.

Источники

1. BECK, K., MARTIN, R. C., SUTHERLAND, J. *Manifesto for Agile Software Development* [online]. 2001 [цитирован 11.03.2022]. Доступен: <https://agilemanifesto.org>
2. SUTHERLAND, J., SCHWABER, K. *The 2020 Scrum Guide* [online]. 2020 [цитирован 12.03.2022]. Доступен: <https://scrumguides.org/scrum-guide.html>
3. SCRUMTREK. *Словарь терминов Scrum* [online] [цитирован 13.03.2022]. Доступен: <https://scrumtrek.ru/blog/agile-scrum/scrum-glossary/3791/velocity>
4. SCRUM INC. *Sprint Burndown Chart* [online] [цитирован 13.03.2022]. Доступен: <https://www.scruminc.com/sprint-burndown-chart>
5. СЕЛИХОВКИН, И. *История ошибок менеджера, черная книга скрам* [online]. 2018 [цитирован 13.03.2022]. Доступен: http://scrumblackbook.pmllead.ru/ScrumBlackBook_v2.pdf
6. АЛИЗАР, А. *Где Agile ужасен, особенно Scrum* [online]. 2018 [цитирован 13.03.2022]. Доступен: <https://habr.com/ru/post/430890>