Вестник Евразийской науки / The Eurasian Scientific Journal https://esj.today

2018, №3, Том 10 / 2018, No 3, Vol 10 https://esj.today/issue-3-2018.html

URL статьи: https://esj.today/PDF/75ECVN318.pdf

Статья поступила в редакцию 26.06.2018; опубликована 18.07.2018

Ссылка для цитирования этой статьи:
Гугаев К.В. Границы применимости компонентов Scrum // Вестник Евразийской науки, 2018 №3, https://esj.today/PDF/75ECVN318.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ.

For citation:
Gugaev K.V. (2018). Restrictions on Scrum components usage. *The Eurasian Scientific Journal*, [online] 3(10). Available at: https://esj.today/PDF/75ECVN318.pdf (in Russian)

УДК 338

ГРНТИ 06.81.19

Гугаев Кирилл Валерьевич
НОЧУ ВО «Московский финансово-промышленный университет «Синергия», Москва, Россия
Аспирант
E-mail: gugaevkirill@gmail.com
РИНЦ: https://elibrary.ru/author_profile.asp?id=930145

Границы применимости компонентов Scrum

Аннотация. Большое количество компаний испытывает сложности с адаптацией и внедрением в свои проекты гибких методологий несмотря на их высокую популярность. Такие фреймворки, как Scrum, предлагают комплексный подход к решению большинства проблем традиционных подходов к разработке в виде набора взаимодополняющих компонент. Однако ввиду чрезвычайного разнообразия IT-проектов некоторые компоненты становятся неприменимыми для конкретных проектов, чьи бизнес-процессы подчинены специфике разрабатываемого продукта.

В статье исследуются составные части фреймворка Scrum в отдельности, исследуются их плюсы и минусы, а также границы применимости к конкретному проекту. Исследуются факторы успешности использования данных компонент и причины, и последствия неудач, допущенных в процессе внедрения Scrum. Помимо компонент методологии исследуются некоторые практики разработки программных продуктов, рекомендованные к использованию методологией Scrum. В статье приводится верхнеуровневое описание поэтапного процесса внедрения Scrum, позволяющий избежать некоторых типовых ошибок внедрения гибких методологий в целом.

Область исследования ограничивается IT-проектами, реализуемыми предприятиями малого и среднего бизнеса ввиду наличия эффектов масштаба, проявляющихся в организации работы над проектами крупных предприятий. Большинство описанных в статье проблем возникает так же и на крупных проектах, однако, могут быть решены при помощи методов и механизмов, недоступных маленьким компаниям. Таким образом, сделано заключение о целесообразности выбора ограниченного набора компонент гибких методологий для внедрения в проекте предприятий малого и среднего бизнеса в зависимости от текущего уровня зрелости команды и процессов разработки.

Ключевые слова: управление проектом; гибкие методологии; Scrum; agile; компоненты гибких методологий; внедрение Scrum; практики разработки

Введение

Гибкие методологии (agile) получили широкое распространение в современном IT-сообществе как замена традиционным каскадным моделям. Существует несколько «фреймворков» гибкой разработки ПО, самым популярным из которых является Scrum, отличительной особенностью которого является акцент на точный контроль процессов разработки.

Scrum призван решить многие проблемы традиционных моделей, однако сотрудники многих компаний, приступающие к внедрению методологии, отмечали ряд негативных эффектов, возникающих в процессе перехода, снижение уровня управляемости командой, или общую неэффективность методологии [6].

Scrum состоит из ограниченного набора хорошо описанных компонент [4], однако состав этих компонент не детерминирован и может изменяться в зависимости от текущей ситуации в проектах, работающих по Scrum [5]. Так же, процесс внедрения этих компонент не регламентирован и потенциально опасен для команды и компании, т. к. совершенные в процессе внедрения ошибки могут привести к распаду команды, уходу сотрудников или потере эффективности команды.

Целью данной статьи является изучение состава компонентов Scrum и процесса их внедрения на предприятиях МСБ (малого и среднего бизнеса) для выявления границ применимости компонент в зависимости от факторов внешней среды и внутренних факторов, таких, как уровень зрелости текущих процессов разработки в компаниях.

Методология Scrum

Scrum – итерационная методология разработки, позволяющая за фиксированные и обычно небольшие промежутки времени поставлять в продукт законченный функционал, представляющий наибольшую ценность. Возможности по реализации функционала предполагаются изменяющимися от итерации к итерации, однако объем работ внутри каждой итерации (спринта) строго зафиксирован и определяется исключительно на планировании в начале спринта. Таким образом достигается предсказуемость и прозрачность хода работ в условиях изменяющихся требований к продукту.

Методология был впервые описали Такэути Хиротака и Икуджиро Нонака в статье «The New Product Development Game» в 1986 г. [10]. Через 10 лет метод был полностью задокументирован и представлен сообществу на конференции OOPSLA’96. В 2001-м году вышла первая книга по методологии Scrum «Agile software development with SCRUM».

Компоненты Scrum

Под методологией Scrum понимается набор необходимых и рекомендуемых компонент, ролевая модель команды и набор практик и подходов к разработке. При этом в процессе адаптации методологии допускается исключение некоторых компонент и выбор оптимального для компании состава методологии и способов взаимодействия между компонентами. Ниже приводится список базовых компонент методологии с указанием факторов, влияющих на эффективность выбора данных компонент в зависимости от ситуации в компании:

Итерация («спринт») – ключевое понятие методологии Scrum, являющиеся атомарным и неделимым отрезком времени длительностью не более 6 недель (по скрам-стандарту Nokia), во время которого выполняется планирование, разработка, тестирование и демонстрация

75ECVN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

функционала. Весь процесс работы над проектом разбивается на такие итерации и все задачи по проекту распределяются в один или несколько спринтов.

Беклог – приоритезированная очередь, в которую помещаются возникающие в процессе работы над проектом [6]. Методология Scrum предполагает обязательное наличие беклога и его поддержание в актуальном состоянии.

Планирование – процесс отбора задач на следующую итерацию из беклога, обязательства по исполнению которых принимает на себя команда проекта [6]. Задачи приоритезируются и оцениваются по времени, количество задач выбирается исходя из производительности команды, показанной в предыдущих спринтах.

Процесс планирования может быть неэффективным в двух случаях:

1. Плохая детализация задач, большая команда и сложная специфика проекта, таким образом задачи тяжело оценить по срокам и объему работ. Методология Scrum предполагает декомпозицию таких задач на более мелкие, объемом не более 12 часов [9]. В случае сложной специфики проекта команда может принять решение о проведении дополнительных встреч по уточнению требований задач перед началом планирования.
2. Излишняя детализация задач на небольшом проекте, требование руководства о формальном соблюдении регламентов методологии. Во избежание излишнего формализма команда должна быть наделена полномочиями самостоятельно выстраивать внутренние процессы взаимодействия на основе предыдущего опыта и ретроспектив.

Ежедневные митинги («стендапы») – обязательные утренние встречи, на которых каждый член команды вкратце рассказывает о планах на текущий день, результатах предыдущего дня и проблемах, стоящих перед ним [9]. Длительность стендапа обычно не превышает 15-ти минут. Таким образом поддерживается синхронизация между членами команды. На небольшом проекте возможны более редкие стендапы, формат встреч и их периодичность должны определяться командой.

Демонстрация результатов итерации («демо») – встреча с заинтересованными в продукте лицами, на которой производится показ разработанного в течение итерации функционала [9]. Недоработанные функции к показу не допускаются. Цель демонстрации – синхронизация ожиданий от продукта между командой и стейкхолдерами и сбор обратной связи по результатам работ. В случае, когда компания – заказчик не работает по Scrum и не ожидает от команды демонстрации незавершенного функционала, демонстрация может быть отменена.

Ретроспектива – командная встреча в конце спринта, целью которой является анализ выполненной на протяжении итерации работы, выявление проблемных мест и точек роста внутри команды [9]. В ходе встречи вырабатываются решения по улучшению процесса работы команды с целью устранения выявленных недочетов. Ретроспективный анализ спринта – ключевой компонент методологии Scrum, предполагающий синхронизацию целей команды и целей компании [2].

Невыполнение командой принятых на ретроспективе решений сводит на нет большую часть пользы от методологии, поэтому важно, чтобы во встрече принимали активное участие все члены команды. Неисполнение решений команды может происходить из-за:

1. Низкого уровня командного духа, отсутствия корпоративной культуры [5].
2. Отсутствие лиц, ответственных за исполнение каждого принятого решения [5].

75ECVN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

3. Личная неудовлетворенность сотрудников.

Обязанностью руководителя команды является выявление и своевременная работа с этими проблемами.

Практики разработки

По завершению итерации в Scrum предполагается, что в продукт будут внесены рабочие и протестированные изменения. Таким образом на каждой итерации в рабочем продукте могу появляться дефекты [1] и возрастать объем технического долга. Для предотвращения риска появления дефектов, методология предполагает внедрение на проекте практики *обязательного кодревью*, *Continuous integration* и введение *юнит и функционального тестирования* компонент перед релизом. Таким образом не покрытый тестами функционал не может считаться завершенным и быть выпущенным в продукте в конце итерации.

Так же предполагается, что несколько членов команды смогут в случае необходимости вносить изменения в модули проекта. Не допускается дефрагментация знаний о проекте и вводится ротация специалистов, разрабатывающих модули проекта, практика *совместных брейнштормов* и *парного программирования* [7], что обеспечивает совместное владение кодом проекта.

Процесс внедрения Scrum

Неграмотное планирование процесса внедрения Scrum может привести к снижению эффективности команды. Часто допускаются следующие ошибки [7]:

1. Отсутствие понимания принципов работы методологии перед началом использования у руководства, либо у команды [4].
2. Внедрение всех без исключения компонент без адаптации к сложившимся бизнес-процессам.
3. Подробное планирование всех 100 % времени разработчиков.
4. Ввод санкций и мер наказания за просрочки задач на этапе обучения команды [8].

Вышеуказанные ошибки влекут за собой следующие последствия:

1. Появление внутреннего сопротивления изменениям [6].
2. Излишняя бюрократизация и рост объемов административной работы.
3. Невозможность выполнения незапланированных задач, отсутствие взаимопомощи в команде разработки ввиду нехватки времени, утрата командного духа и уход ключевых сотрудников [8].
4. Невозможность обучения команды на собственных ошибках, снижение эффективности методологии Scrum [5].

Правильный процесс внедрения может быть разбит на следующие этапы:

1. Подготовка компании к трансформации [9].

[1] ISO 9001:2000. Quality Management Systems Requirements, 2001.

Цель этапа: сбор и анализ информации о компании, знакомство сотрудников с принципами Agile.

Состав этапа:

- Анализ и описание текущих бизнес-процессов.
- Сбор информации о проектах компании.
- Тренинги и деловые игры для команды.
- Тренинги для скрам-мастеров.
- Обучение владельцев проектов.

2. Знакомство команды с ключевыми компонентами Scrum.

Цель этапа: переход к новому процессу разработки.

Состав этапа:

- Создание беклогов проектов.
- Старт первых тестовых спринтов.
- Проведение ежедневных митингов, демонстрации результатов и ретроспективы под руководством тренера [4].
- Проведение стабилизационных спринтов при необходимости [9].

3. Адаптация методологии к бизнес-процессам компании.

Цель этапа: отработка процессов планирования, выбор и внедрение практик разработки по Scrum.

Состав этапа:

- Проведение планирования спритнов исходя из user-stories [4].
- Внедрение процессов обеспечения качества [3].
- Внедрение функционального тестирования и механизма continuous-integration.

Заключение

При грамотном использовании методология Scrum позволяет повысить прозрачность и прогнозируемость работ над проектами, снизить риски и стабилизировать скорость разработки в условиях внешней среды [1]. В процессе внедрения методологии могут быть допущены существенные ошибки, для предотвращения которых требуется грамотное планирование внедрения и помощь опытных специалистов. Скорость внедрения методологии зависит от уровня зрелости текущих процессов разработки и уровня осведомленности команды о принципах Agile [7].

Состав компонент методологии должен быть определен в процессе адаптации фреймворка к текущим бизнес-процессам компании. Допускается исключение некоторых компонент в зависимости от размера команды, объема и специфики проекта, при этом финальный набор компонент определяется как руководством компании, так и непосредственно командой проекта.

75ECVN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

ЛИТЕРАТУРА

1. Гугаев К.В. Управление рисками инновационных IT-проектов // XIII Международный научный конгресс. 2018.
2. Гугаев К.В., Хабаров В.И. Современные подходы к целеполаганию в технологической компании // Успехи современной науки и образования. 2017. Т. 6. С. 31-34.
3. Гугаев К.В., Хабаров В.И. Обеспечение качества программных продуктов в Agile-разработке // Вестник Академии. Изд. Московская академия предпринимательства при Правительстве Москвы. 2017. Т. 6. С. 79-83.
4. Кон М. Scrum: гибкая разработка ПО: Вильямс, 2011.
5. Рубин К. Основы Scrum: практическое руководство по гибкой разработке ПО: Вильямс, 2016. 544 с.
6. Сазерленд Д. Scrum. Революционный метод управления проектами: Манн, Иванов и Фербер, 2018. 272 с.
7. Rosenberg D., Stephens M., Collins-Cope M. Agile development with ICONIX process: people, process, and pragmatism: Apress, 2005. 261 c.
8. Safiullin A., Tronin V. Cooperation and research networks in the knowledge economy // Probl. theory Pract. Manag. 2015. T. 2.
9. Schwaber K., Sutherland D. The Definitive Guide to Scrum: The Rules of the Game: Creative Commons, 2016.
10. Takeuch H., Nonaka I. The New New Product Development Game // Harward Bus. Rev. 1986.

75ECVN318
Издательство «Мир науки» \ Publishing company «World of science» http://izd-mn.com

Gugaev Kirill Valerievich
Moscow university for industry and finance «Synergy», Moscow, Russia
E-mail: gugaevkirill@gmail.com

Restrictions on Scrum components usage

Abstract. Huge amount of companies are experiencing negative effects of agile methodologies adaptation and embedding, regardless its high popularity. Flexible methodologies like Scrum offers complex solution for traditional approach problems as a set of different linked components, each complement other. However, some components are found not applicable for current company projects due to high variance of project mechanics in IT-industry and the fact, that IT-companies subordinate their business-processes to the product specific.

Article explores scum framework parts apart, their pros and cons, and their application restrictions for standalone project. Also, these components success usage factors, Scrum embedding process fault reasons and consequences are described. Besides Scrum framework components, some effective development practices, recommended for usage by Scrum authors are explored in the Article. Article contains brief description of Scrum gradual embedding process, dedicated to avoiding some typical Scrum and agile embedding faults.

Article research scope is bounded by small and medium-sized enterprise projects due to high-scale effects, witnessed in large companies approaches for projects organization. Most of problems, explored in the article, are situated not only on small startups, but also on huge enterprise projects, but large companies can handle these problems using methods and mechanisms, not allowed for small business. Thus, article provides a conclusion about limited amount of Scrum framework components usage expedience for embedding in concrete it-project, developed by the small and medium-sized enterprises due to current project command skills and company development-processes level.

Keywords: project management; flexible development methodologies; Scrum; agile; agile components; Scrum embedding; development practices