

ПРОБЛЕМЫ И СПОСОБЫ УЛУЧШЕНИЯ ГИБКОЙ МЕТОДОЛОГИИ РАЗРАБОТКИ SCRUM

Тепляков Н.Ю.

*Тепляков Николай Юрьевич - студент,
кафедра информационных систем,
Институт приборостроения, автоматизации и информационных технологий
Орловский государственный университет им. И.С. Тургенева, г. Орел*

Аннотация: *очень важно на стадии реализации ПО выбрать методологию разработки. Scrum, как один из методов гибкой методологии показывает себя очень эффективно, но имеет ряд недостатков. К ним можно отнести: не эффективна, если специалист не имеет должной квалификации; низкое внимание тестам на разных стадиях разработки программного проекта; узкая специализация методологии. Для улучшения Scrum были предложены следующие способы: учет сложности задач; переоценка приоритетов на каждой итерации спринта; декомпозиция задач; использование принципа: задача не является выполненной, пока не проведены все тесты.*

Ключевые слова: *гибкая методология разработки, Scrum.*

Процесс выбора методологии разработки программного обеспечения является очень важным на начальной стадии проектирования продукта. Разработка ПО, как и любая другая техническая дисциплина, имеет дело со следующими основными проблемами: качество, стоимость и надежность [2, с. 6]. В связи с этим правильная организация процесса разработки программного обеспечения является основой достижения лучшего результата в ожидаемые сроки, с хорошим уровнем качества и с запланированным бюджетом.

Одним из популярных и эффективных методологий разработки ПО является гибкая методология (англ. Agile software development) – серия подходов к разработке программного обеспечения, ориентированных на использование интерактивной разработки, динамическое формирование требований и обеспечение их реализации, в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля. Существует несколько методик, относящихся к классу гибких методологий разработки, в частности экстремальное программирование, DSDM, Scrum, FDD [1, с. 8].

Scrum (от англ. scrum – «толкучка») – методология управления проектами, активно применяется при разработке информационных систем для гибкой разработки программного обеспечения. Scrum делает акцент на качественном контроле процесса разработки. В основе управления процессами в Scrum заложены три главных принципа: прозрачность, инспекция и адаптация. Прозрачность означает, что все значимые аспекты процесса должны быть доступны тем, кто за него отвечает. Прозрачность требует, чтобы эти аспекты определялись общими стандартами, а все члены команды должны пользоваться общими понятиями и терминологией. Инспекции проводятся достаточно часто для своевременного выявления нежелательных отклонений от установленных целей. Однако инспектирование не должно быть настолько частым, чтобы мешать работе. Если по результатам инспекции выявлены отклонения, то необходимо как можно быстрее внести коррекцию в продукт во избежание распространения проблемы на другие части системы. Методология Scrum – это набор принципов, на которых строится процесс разработки, позволяющий в жестко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определен наибольший приоритет [3, с. 10].

Однако наряду со многими преимуществами методология Scrum обладает рядом проблем, среди которых можно выделить следующие:

1. Scrum не дает нужных результатов, если специалист не имеет должной квалификации. Для новых сотрудников, стажеров, специалистов с малым опытом работы, только пришедших в команду, данная методология может стать камнем преткновения.

2. Низкое внимание тестам на разных стадиях разработки программного проекта. Из-за слишком коротких интервалов спринта в проекте постепенно могут начать накапливаться разного рода ошибки (технические, бизнес-логики и т.д.).

3. Узкая специализация методов. Методология Scrum эффективно проявляет себя в технической сфере. Так, например, если использовать Scrum при разработке сайтов, этапы дизайна и контента уже будут выходить за рамки методологии и требовать совершенно иного подхода.

Зная основные недостатки Scrum, можем выделить следующие способы улучшения эффективности и устранения недостатков данной методологии:

1. Добавление задачам дополнительного свойства – сложности. Первенство по времени в осуществлении какой-либо деятельности – это залог быстро выполненного прототипа с основным функционалом, а соответственно и быстро выполненного проекта в целом [3, с. 50]. Если давать оценку сложности задачи, то появляется возможность распределять задачи в зависимости от уровня квалификации специалистов. Также, зная уровень сложности задачи, из двух задач с одинаковым приоритетом можно принять управленческое решение о выборе из этих задач более выгодной, опираясь на стоимость реализации этой задачи.

2. Использование многоуровневой иерархии задач. Подразумевается, что каждая задача может разбиваться на подзадачи, а каждая из подзадач может быть разделена еще на задачи и т.д. Данный метод называется декомпозицией. Декомпозиция задач – метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач, пусть и взаимосвязанных, но более простых.

3. Декомпозиция, как процесс расчленения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть расчленены на части. Данный метод дает возможность эффективного распределения и решения задач между разными подразделениями фирмы или отдельными сотрудниками. Также метод расширяет границы использования методологии SCRUM в других сферах, помимо технической.

4. Тестирование задач на каждом из уровней иерархии. Цель тестирования: продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям, выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации [5, с. 18]. Это предотвратит накопление ошибок, ведь порой на поиск и устранение на поздней стадии разработки тратится больше времени и ресурсов, чем, если бы эту ошибку исправили сразу. Если тестирование прошло удачно, задача помечается выполненной.

5. Переоценка приоритетов на каждой итерации спринта. Спринт в контексте методологии SCRUM – итерация, в ходе которой создается функциональный рост программного обеспечения [3, с. 16]. Переоценка приоритетов задач на каждой итерации спринта дает возможность более эффективного распределения ресурсов проекта. В течение периода очередной итерации, решаются задачи, обсуждаются новые идеи и некоторые задачи могут оказаться на втором плане.

Использование улучшений методологии разработки программных продуктов Scrum, позволяют устранить основные проблемы, перечисленные выше, повышая эффективность и пользу, которую может принести данная методология.

Список литературы

1. *Stephen Palmer John Felsing. A Practical Guide to Feature-Driven Development.*
2. *Rosenberg Doug, Stephens Matt u Collins-Cop Mark. Agile Development with the ICONIX Process: People, Process and Pragmatism.*
3. *Schwaber Ken. Agile Project Management with Scrum.*
4. *Поппендик М., Поппендик Т. Бережливое производство программного обеспечения. От идеи до прибыли.*
5. *Кристин Л., Грегори Д. Гибкое тестирование. Практическое руководство для тестировщиков ПО.*