УДК 008.5

А.С. Коляда, аспирант,
С. Н. Ковешников, руководитель IT-проектов,
В.Д. Гогунский, д.т.н., проф.,
Одесский национальный політехнический университет

ЭФФЕКТИВНОСТЬ ИСПОЛЬЗОВАНИЯ АДАПТИВНЫХ ПОДХОДОВ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.С. Коляда, С.М. Ковешников, В.Д. Гогунський. **Ефективність використання адаптивних підходів при розробці програмного забезпечення.** У статті розглядається ефективність методологій розробки програмного забезпечення заснованих на ітеративному підході. Показані переваги, а також обмеження адаптивних методологій на прикладі процесу під назвою Скрам.

A.S. Kolyada, S.M. Koveshnikov, V.D. Gogunsky. **Efficiency of using Agile software development**. The article discusses the effectiveness of software development methodologies based on iterative approach. The advantages and limitations of agile methodologies are shown by the example of a process called Scrum.

Введение. Адаптивные методологии разработки программного обеспечения (ПО) не составляют очередной набор методов, а скорее являются культурой разработки приложений.

Под методологией понимается набор моделей, методов и правил, которых надо придерживаться, а культура – определяет множество укоренившихся концепций, принятых некоторым сообществом исполнителей и руководителей проектов, которые разделяют общие взгляды относительно поведенческих компетенций команды проекта или программы проектов.

С помощью методологии люди узнают как себя вести, культура же – определяет человеческую потребность вести себя определённым образом.

Анализ предыдущих исследований. Как следствие эволюционного развития итеративных методологий разработки ПО были разработаны адаптивные методологии. В 2001 г. 17 разработчиков ПО (Кент Бек, Майк Бидл, Эйри ван Беннекум, Алистер Кокбёрн и др.) собрались в городке Сноуберд в штате Юта, чтобы обсудить новые методологии разработки. В результате был разработан и принят документ «Манифест адаптивных методологий разработки» (Agile Manifesto) [1]:

- *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.

Наиболее популярной адаптивной методологией разработки ПО является Скрам (Scrum), который мы возьмём за основу для демонстрации преимуществ этого новаторского подхода. «Подход водопадной модели к разработке продукта… может противоречить целям максимальной скорости и гибкости. Вместо этого, целостный подход адаптивной модели, где команда пытается пройти дистанцию как единое целое, продвигаясь назад и вперед - может лучше служить сегодняшним конкурентным требованиям» [2].

Скрам – один из процессов адаптивных методологий, который фокусируется на поставке наиважнейших, с точки зрения бизнеса клиента, ценностей в наикратчайшие сроки.

Процесс разработки ПО представляется в виде ограниченных по времени итераций равной длительности, которые называются спринтами. Как правило, длительность каждого спринта фиксируется в начале проекта и колеблется от 1

до 4 недель в зависимости от специфики проектных целей. Скрам позволяет быстро и регулярно осматривать реально работающее программное обеспечение. В конце каждого спринта команда демонстрирует заказчику действующую версию ПО, обладающего функциональностью оговоренной перед стартом спринта. Заказчик расставляет приоритеты.

Команды самоорганизовываются и определяют лучший способ, чтобы в первую очередь реализовать функционал имеющий наивысший приоритет, а значит являющийся критическим с точки зрения бизнеса клиента.

С регулярностью от одной недели до месяца все могут видеть реально работающий программный продукт, и решить выпускать его как он есть либо продолжить улучшение в следующем спринте, расширив его дополнительными функциональными возможностями.

В среде проектных менеджеров ПО появились новые термины, которые отображают срецифику разработки проектов ПО:

- *владелец продукта (Product owner)*– это человек, ответственный за приоритезацию требований и часто за их создание;

- *скрам-мастер* – член команды, который дополнительно отвечает за процессы, координацию работы и поддержание социальной атмосферы в команде;

- *команда* – 7±2 человек, реализующие требования владельца продукта;

- *беклог продукта (Product Backlog)* – приоритезированный список требований с оценкой трудозатрат – обычно он состоит из бизнес-требований, которые приносят конкретную бизнес-ценность и называются элементами беклога;

- *беклог спринта (Sprint Backlog)* – часть беклога продукта, с наиболее высокой важностью и суммарной оценкой;

- *скрам-митинг (Scrum meeting)* – собрание членов команды проекта (с приглашением владельца продукта) для синхронизации деятельности команды и обозначения проблем. Каждый член команды отвечает на три вопроса: « Что было сделано после предыдущего скрам-митинга? Какие есть проблемы? Что будет сделано к следующему скрам-митингу?»;

- *планирование спринта* - команда выбирает из Бэклога продукта требования, которые они могут реализовать за спринт и согласовывает его с владельцем продукта; создается Бэклог спринта, учитывая высокоуровневую архитектуру приложения;

- *обзор спринта* – показ владельцу продукта (и заинтересованным лицам) работающего функционала продукта, сделанного за спринт – основная задача проведения обзора спринта заключается в получении обратной связи;

- *ретроспектива* – периодический пересмотр того, что работает, а что нет; ретроспективу традиционно проводят после обзора спринта;

- *скрам-мастер* – собирают всю команду для обсуждения результатов спринта; рекомендуется на ретроспективу приглашать владельца продукта для получения дополнительной обратной связи

Материал и результаты исследования. Определяющим свойством адаптивных методологий разработки ПО является ориентированность на удовлетворение потребностей клиента, а не на решение конкретных прикладных задач или удовлетворение амбиций менеджеров или команды разработчиков. Заказчик находится в непрерывной конкуренции на своём собственном рынке. Чтобы выжить и процветать в жестких, конкурентных условиях современного рынка, ему приходится гибко и оперативно реагировать на любые потенциальные возмож-

ности укрепления позиций своего бизнеса и экспансии в смежные области еще не занятые конкурентами. Это напрямую влечёт за собой постоянную адаптацию внутренних процессов и, как следствие, непрерывное изменение требований к создаваемому ПО. Сместив фокус на поддержку этих меняющихся процессов, мы становимся для клиента вложением средств в собственный бизнес, а не постоянными расходами на управление требованиями в вечном противостоянии заказчиков и разработчиков ПО.

Перед появлением итеративных методологий разработки использовалась «Водопадная модель», в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки (рис. 1).

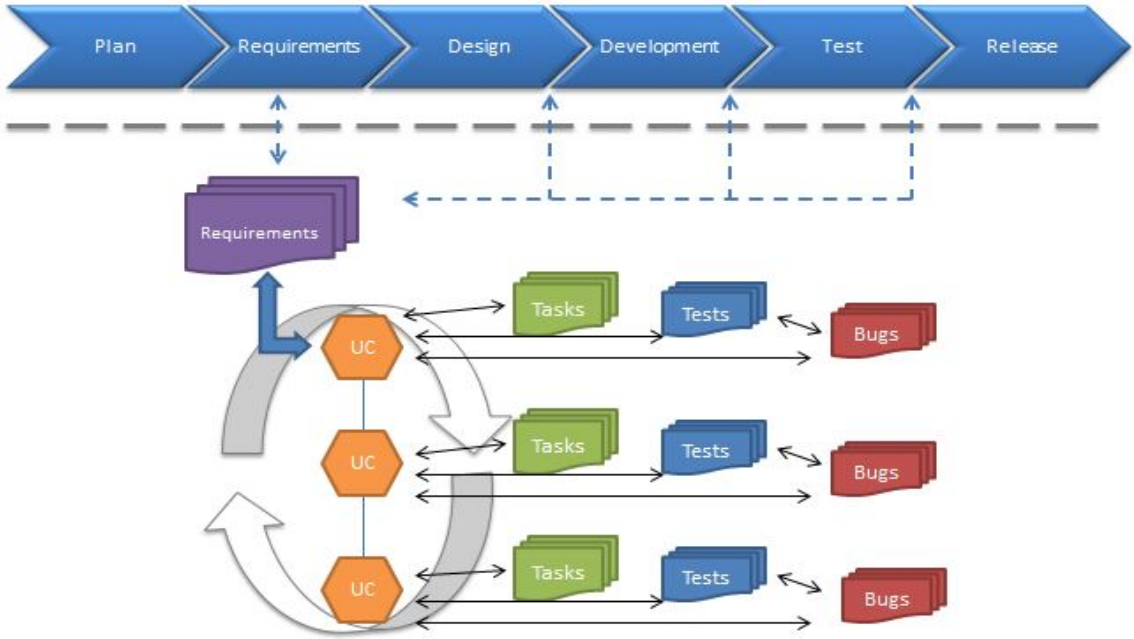

Рис. 1. Водопадная модель разработки ПО в сравнении с адаптивной

Выполнение каждой стадии строго последовательно является главным недостатком этой модели, из которой вытекают и остальные, такие как: сложность и трудоемкость внесения изменений в плане; нерациональный расход времени (фаза дизайна не может быть начата, пока не закончена фаза анализа требований); тестирование выполняется на последних стадиях, хотя более эффективно выполнять его как можно раньше. Методику «Водопадная модель» критикуют за недостаточную гибкость и объявление самоцелью формальное управление проектом в ущерб срокам, стоимости и качеству (табл. 1).

Таблица 1

Ценности адаптивной модели в противовес водопадной

Адаптивная модель	Водопадная модель
Люди и их взаимодействие	Процессы и инструменты
Готовый продукт	Документация
Сотрудничество с заказчиком	Жесткие контрактные ограничения
Реакция на изменения	Следование плану

Итеративные методологии меняют философию ведения проектов. Мишель Слигер в своей статье [3] по сравнению водопадной и адаптивной моделей указывает, что в философии водопадной модели «священная корова» – это сам план (plan driven), а в итеративных методиках «священная корова» – это максимальное удовлетворение заказчика в рамках фиксированного бюджета и сроков (value driven). Эта методологическая разница в подходах демонстрируется следующей схемой с помощью треугольника «сроки-деньги-объем работ» (рис. 2).

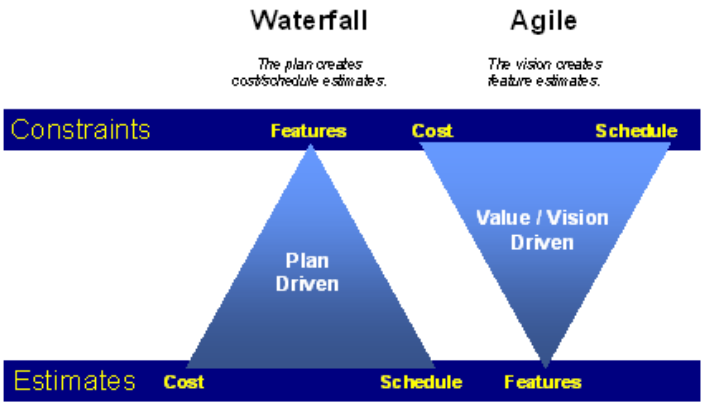

Рис. 2. Треугольник «сроки-деньги-объем работ» для водопадной и адаптивной моделей

Преимущества адаптивных технологий можно сформулировать так: возможность нивелирования воздействия серьезных рисков на ранних стадиях проекта, пока это еще можно сделать с минимальными затратами; возможность организовать плодотворную обратную связь с будущими конечными пользователями для создания системы, реально отвечающей их потребностям; акцент усилий на наиболее важные и критичные направления проекта; непрерывное итеративное тестирование конечного продукта, позволяющее оценить успешность всего проекта в целом; раннее обнаружение несоответствий между требованиями, моделями и программным кодом; более равномерная загрузка участников проекта; эффективное использование накопленного опыта; реальная оценка текущего состояния проекта и, как следствие, большая уверенность заказчиков и непосредственных участников в его успешном завершении; знания о предметной области и технических решениях распределены равномерно между участниками команды.

Скрам также имеет некоторые ограничения. Он подходит не для всех проектов, а для тех, где Заказчик (или его представитель) может постоянно уделять время команде и готов мобильно изменять требования к системе, чтобы повысить свой уровень удовлетворения, а также удержать проект в сроках и бюджете.

Еще одно ограничение относится к команде, которая должна быть кросс - функциональной, состоящей из специалистов разных профилей: тестировщиков, архитекторов, аналитиков, программистов и т. д. Команда саморганизовывается – в идеале нет специальных ролей. Перед внедрением Скрам-процесса команда должна быть зрелой, то есть достигнут уровень, при котором она может эффективно работать для достижения поставленных целей в условиях комфортной атмосферы, взаимной поддержки и взаимопомощи.

Выводы. Скрам – один из процессов адаптивных методологий, который позволяет фокусироваться на поставке наиважнейших, с точки зрения бизнеса, ценностей в наикратчайшие сроки.

Успешная реализация Скрама имеет много преимуществ как для команд, так и для менеджеров. Хорошо функционирующий Скрам представит функциональность с самым высоким приоритетом в первую очередь и позволит избежать создания функций, которые никогда не будут использоваться заказчиком. Индустриальные данные показывают, что около половины разработанных функций программного обеспечения никогда не используются, разработка может быть завершена в 2 раза быстрее, избегая траты времени или ненужной работы. Хорошо работающий Скрам достигает эффекта Toyota: в 4 раза выше средняя продуктивность труда и в 12 раз выше качество [1]. И наконец, совсем не обязательно применять все практики Скрам. Это можно делать постепенно либо вообще использовать одну наиболее подходящую проекту практику.

Література

1. Лайкер, Д. Практика дао Toyota: Руководство по внедрению принципов менеджмента Toyota [Текст] / Джеффри Лайкер, Дэвид Мейер; пер. с англ. – М. : Альпина Бизнес Букс, 2006. – 588 с.

2. Такеучи, Х. The New New Product Development Game. [Текст] / Хиротака Такеучи и Икуджиро Нонака // Harvard Business Review. – Harvard, 1986.

3. Sliger, M. Relating PMBoK Practices to Agile Practices підходу [Текст] / Michele Sliger – Part 2 of 4.

4. Kniberg, H. Scrum and XP from the Trenchers. How do we do scrum підходу [Текст] / Henrik Kniberg. – C4 Media, 2007. – 142 p.