|  |  |
| --- | --- |
|  | **Кислюк Игорь Витальевич**  Год рождения: 1996  Факультет ИКТ, кафедра ПС, группа K4120  Направление подготовки: 11.04.02 – Программное обеспечение в инфокоммуникациях  e-mail: igorkislyuk@icloud.com |

УДК – 004.41

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ КАЧЕСТВА И СКОРОСТИ РАЗРАБОТКИ КОМАНДЫ ПРИ РАЗЛИЧНЫХ СПОСОБАХ ОРГАНИЗАЦИИ РАБОТЫ**

**И.В. Кислюк**

**Научный руководитель – старший преподаватель С.В. Одиночкина, Университет ИТМО**

В данной статье проведен обзор основных подходов к организации процесса разработки среди команд разработки программного обеспечения и проведен их анализ.

**Ключевые слова**: способ процесса разработки, команда разработки, качество и скорость разработки

Согласно исследованиям Standish Group итогов выполнения проектов по разработке ПО, доля полностью успешных проектов в среднем за 5 лет составляет 29%, что варьируется в пределах одной трети [1]. В области разработки ПО ключевым моментом для успешной реализации проекта является организация людей в команды. Грамотно организованная и сбалансированная команда способна решать поставленные бизнес-задачи. В современном мире немалую роль отводят гибким методологиям, которые ориентируются на человеческое взаимодействие. Независимо от Standish Group исследования Ambysoft доказывают преимущества гибких методологий разработки, что подтверждает важность взаимодействия, понимания и согласованности между работниками команды [6]. Хорошие показатели этих параметров определяются через качество и скорость разработки.

Качество разработки в первую очередь состоит из заявленного уровня качества результирующего продукта. Влияние оказывает **понимание и взаимодействие отделов** команды, **предоставленные человеческие ресурсы**, **размеры выпускаемого продукта**, наличие технического долга и **неизменность требований** в процессе работы. Перечисленные факторы зависят от правильного выбора способа организации работы команды. Параметр скорости разработки команды определяется как отношение количества решенных бизнес-задач ко времени и зависит также от необходимой **частоты выпуска версий** продукта, **важности поставки** программного обеспечения **в определенный срок**.

Первым способом организации выступает каскаднаямодель. Название данного способа описывает главный принцип организации. Ввиду последовательности этапов разработки, сотрудничество отделов не затруднено. Размеры команды и размер выпускаемого продукта неважны для данного способа, чего нельзя сказать относительно устойчивости требований и наличия технического долга. Отсутствует необходимость частой поставки обеспечения в конкретный срок и регулярного выпуска частично-работающих версий, что положительно сказывается на определении скорости в начале планирования разработки. Тип задачи для выбора данного способа – минимальная рабочая версия для получения оценки продукта на рынке.

Вторым способом организации выступает гибкая методология разработки Agile. Главная характеристика данной модели представлена в одноименном манифесте, а появление обусловлено постоянно меняющимися требованиями рынка программного обеспечения [4]. Стабильность требований становится менее важной, чем качество продукта, объемы предоставленных ресурсов и наличие технического долга. Итеративность, ключевая характеристика, позволяет повысить понимание и слаженность отделов команды. По отношению к каскадной модели разработки увеличивается важность скорости, о чем свидетельствует необходимость частой поставки версий продукта и поставка в определенный срок. Тип задач для выбора данного способа – весь жизненный цикл программного обеспечения от разработки до поддержки.

Третьим способом организации выступает следующий вид множества гибких методологий – Scrum. Основным отличием выступает «жесткая фиксированность временных рамок» [3] и упор на итеративность. На основе этого отличия возрастает важность качества продукта и человеческих ресурсов в рамках данного способа, наличие технического долга и размеры продукта отходят на второй план при выборе данного способа. Наличие обязательных ежедневных совещаний и собраний в отличие от предыдущего способа помогает вывести момент коммуникации участников команд на новый уровень. Параметр скорости достигает наибольшей значимости при выборе данного способа, поскольку частота и необходимость поставки продукта к конкретному моменту во времени выходят на первый план. Тип задач для выбора данного способа – решение сложных задач бизнеса и разработка продукта на протяжении нескольких лет.

В таблице 1 представлено сравнение ключевых показателей исследования после завершения проекта [2]. Наблюдается тенденция, что при выборе одной из представленных гибких методологий повышается качество с заметным понижением общей скорости разработки по сравнению с каскадной моделью. В рамках данного исследования при сравнении видов гибких методологий наблюдается большая «гибкость», представленная Agile, что заключается в качестве продукта и количестве обращений неудовлетворенных пользователей, а также сходство Scrum и каскадной модели разработки, выраженное в незначительном повышении показателей общей скорости разработки проекта с небольшим ухудшением качества. На основе приведенных показателей можно выполнить грамотный выбор необходимого способа разработки.

Таблица 1. Сводные параметры сравнения проектов при различных методологиях

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Используемая методология | Кодовое название проекта | Среднее количество найденных ошибок за все время тестирования (на 1000 строк кода) | Средняя частота обращений пользователей в службу поддержки (количество в месяц) | Среднее значение скорости разработки (1000 строк на разработчика в месяц) | Относительная точность оценки сроков выполнения проекта (планируемая к реальной) |
| Waterfall | Курасао (Curacao) | 9.85 | 67.50 | 7.65 | 0.91 |
| Agile | Хитра (Hitra) | 6.20 | 61.20 | 5.89 | 0.61 |
| Scrum | Венера (Venus) | 6.95 | 69.36 | 6.11 | 0.66 |

**ЛИТЕРАТУРА**

1. Standish Group 2015 Chaos Report [Электронный ресурс]. – Режим доступа: https://www.infoq.com/articles/standish-chaos-2015 свободный. Язык рус. (дата обращения 03.02.2018)
2. B. Lindström. A Software Measurement Case Study using GQM. – Software Engineering Institute, 2014.– 96 с.
3. Вольфсон Б. Гибкие методологии разработки.– СПб.: Питер, 2017.– 144 с.
4. Грин Д. Постигая Agile.– СПб. Манн, Иванов и Фербер, 2017.– 350 с.
5. David C. Young. Software Development Methodologies. – Alabama Supercomputer Center, 2013.– 10 с.
6. Agile vs Waterfall: Which Method is More Successful? [Электронный ресурс]. – Режим доступа: https://clearcode.cc/blog/agile-vs-waterfall-method/ свободный. Язык рус. (дата обращения 03.02.2018)