|  |  |
| --- | --- |
|  | **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 – 007.3

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

**И.В. Кислюк**

**Научный руководитель – доцент, к.т.н Н.А. Осипов**

**Куратор – старший преподаватель С.В. Одиночкина**

В данной статье проведен обзор основных подходов к организации процесса разработки среди команд мобильной разработки, проведен анализ и перечислены достоинства и недостатки тех или иных подходов.

**Ключевые слова**: модель процесса разработки, команды мобильной разработки, анализ качества и скорости разработки

При любой разработке очень важно грамотно построить работу команды для достижения желаемых результатов за минимальное количество ресурсов. К ресурсам относится время, деньги, техническая экспертиза членов команды. Важным моментом становится именно организация людей в команду, за которую отвечают управленцы. Таким образом становится возможно определить показатель продуктивности управляющего звена – насколько слаженная команда и с какой возможной скоростью решаются бизнес-задачи. Согласно данному источнику [5], показатели продуктивности, это «показатели деятельности подразделения (предприятия), которые помогают организации в достижении стратегических и тактических (операционных) целей».

Возможно ли производить расчет показателя продуктивности отдельных работников для формирования общего показателя команды? В данном контексте многое зависит от задач работника. Например, для отдела продаж показателем продуктивности служит выполнение плана продаж в течении периода времени. Для менеджеров проекта – способность «удержать» заказчика в данной компании, удовлетворить все его потребности и решить его бизнес-задачи.

Немного сложнее выступает расчет показателя продуктивности для инженера-разработчика. Какие цели стоят перед ним? Во-первых, необходимо предоставить грамотное и надежное техническое решение для конкретной задачи клиента. Этот момент контролируется различным тестированием. Во-вторых, написанный код должен удовлетворять требованиям промышленной разработки. К таким требованиям можно отнести время исполнения, алгоритмическую сложность, понятность и логичность кода. Это стоит контролировать на этапе проведения код-ревью. Исходя из таких параметров становится возможным грамотное измерение показателя продуктивности инженеров-разработчиков. Для нетехнических специалистов это может быть сложной задачей и тогда следует опираться на опытных руководителей.

Показатель продуктивности является составляющей общего процесса организации команды. Под командой в сфере мобильной разработки понимается слаженная работа большого количества людей – менеджеров или иных управленцев, разработчиков визуальной части, инженеров-разработчиков разнообразных мобильных платформ, тестировщиков, участников отдела техподдержки. Именно в данный момент проблема организации большого количества людей в грамотный и направленный процесс для решения бизнес-задач становится наиболее остро.

В данной статье будут рассматриваться 3 наиболее известных подхода к разработке – каскадная и две гибких модели разработки. Первая ориентирована на проект, в то время как две других – на продукты. Примером области разработки выбрана развивающейся на данный момент сфера мобильной разработки. Почему? Все больше гигантов разработки программного обеспечения России уделяют этому больше и больше внимания. Стоит отметить, такие компании как Mail.Ru Group, Digital Design, EPAM Systems активно проводят организацию и сбор команд мобильной разработки. Будет проводиться сравнение качества и скорости разработки команды среди выбранных подходов. Для этого необходимо ввести данные понятия.

**Качество разработки** напрямую связано с качеством получаемого продукта через некоторый коэффициент. Качество разработки, умноженное на этот коэффициент, дает в результате качество продукта. В данном случае продуктом выступает разрабатываемое программное обеспечение. В коэффициент входят различные потери при работе, погрешности, неясности. Качество разработки складывается из следующих понятий – **скорость взаимодействия отделов** команды между собой, **частота закрытия технического долга**, **понимание со стороны руководителя**, где возникают проблемы и необходимая частота ретроспективы команды. Все эти понятия находятся под прямым влиянием от процесса построения работы команды. Поэтому грамотная организация команды ведет к качеству разработки. Вторым понятием является **скорость разработки** команды. Это **отношение** количества решенных **бизнес-задач к времени**.

Скорость разработки будет оцениваться через наличие итеративности, влияния её на процесс разработки, а также через наличие возможностей, с помощью которых легче производится оценка задач. Имея рамки стоимости задачи легче осознать рентабельность с точки зрения бизнеса. Качество разработки, в свою очередь будет оцениваться через понимание взаимодействия отделов, понимание руководителей в том, чем занимается команда в данный момент, понимание команды проблемы как таковой, и задач на проекте. Подходы к организации команды можно классифицировать по типу модели разработки. Модель разработки – это структура, согласно которой построена разработка программного обеспечения [4].

Первой моделью процесса разработки является **каскадная модель** разработки. Задачи устанавливаются в рамках всего проекта. Согласно этому источнику [3], «модель процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки». Особенностью является отсутствие переходов между различными фазами проекта, т. е. переход к следующей фазе осуществляется только после завершения текущей фазы, а также не наблюдается обратных переходов. Отсутствие последних ведет к тому, что после завершения этапа невозможно повторно вернуться к проблеме и устранить недочеты и разногласия команды.

Данная модель положительно сказывается на скорости взаимодействия отделов между собой, на что напрямую влияет последовательность этапов. Критическим моментом стоит отметить полнейшее отсутствие технического долга. При учете отсутствия ретроспективы, руководителю команды сложно рассуждать об этих проблемах, сложно взглянуть на работу команды со стороны и предпринять какие-либо меры по устранение внутренних проблем. Типичным составом такой команды является команда из 1 разработчика на платформу, 1 дизайнера и 1 инженера качества продукта. Спектр задачи – минимальная рабочая версия приложения для получения оценки рентабельности продукта на рынке.

Вторая модель является гибкой методологией разработки. Согласно данному источнику [1] **Agile** – это модель процесса разработки, "ориентированная на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля". Существенное отличие от каскадной модели в итеративности. Главным пониманием работы команды есть наличие 4 принципов главных принципов Agile манифеста [2].

Вторая модель разработки отлично подходит для команд, которые работают над большими проектами внутри компании. Показатель качества команды гибкой разработки значительно выше, чем у предыдущей, на что позитивно влияет наличие итеративности. Наличие некоторого цикла процессов внутри компании увеличивает частоту проведения анализов, выявление недостатков работы команды, проблем, которые возникают на уровне межличностного взаимодействия. В данном контексте возможны более низкие показатели скорости команды [2], «вполне возможно, что финальная версия продукта выйдет гораздо позднее, чем при использовании каскадной модели разработки». Типичным составом такой команды является несколько разработчиков на платформу, до двух дизайнеров, такого же количества инженеров качества продукта. Целями команды служит разработка и развитие продукта.

Третья и последняя модель разработки является следующей ступенью развития гибких методологий. **Scrum** является разновидностью гибких методологий. Согласно данному источнику [6], это «набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами, предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет». Среди особенностей этого подхода к разработке относят короткие итеративные промежутки времени для выполнения задач, планирование, активное понятие технического долга. Весомым достоинством выступает итеративность, трансформировавшаяся в довольно жесткие рамки временной продолжительности под названием спринт. Важным моментом контроля являются регулярные собрания, что помогает актуализировать понимание менеджеров команды текущие проблемы и ситуации, которые требуют усиленного внимания и решения. Скорость взаимодействия отделов значительно возрастает при использовании меж-командных ежедневных совещаний. Касательно мобильной разработки, таким образом построены процессы работы в больших командах, которые в среднем состоят из 5-10 разработчиков для каждой платформы, нескольких дизайнеров, нескольких инженеров качества продукта. Организация работы команды в таком ключе позволяет решать и разрабатывать приложения общим сроком выполнения в несколько лет.

Процесс анализа ключевых методологий мобильной команды разработки показал, что для каждой задачи необходимо грамотно выбирать подходящий способ организации команды. Грамотный выбор определяется опытом руководителя и пониманием соотношения показателей работы команды –качества и скорости. Среди возможных улучшений качества работы команды наблюдается повышение технической экспертизы руководящий людей, контроль частоты смены кадров на проекты, а также привлечение сторонних экспертов.

**ЛИТЕРАТУРА**

1. Вольфсон Б. Гибкие методологии разработки.– СПб.: Питер, 2017.– 144 с.
2. Грин Д. Постигая Agile.– СПб. Манн, Иванов и Фербер, 2017.– 350 с.
3. Каскадные модели разработки [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Каскадная\_модель свободный. Язык рус. (дата обращения 16.01.2018)
4. Семь основных методологий разработки [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/company/edison/blog/269789/ свободный. Язык рус. (дата обращения 16.01.2018)
5. Ключевые показатели эффективности [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Ключевые\_показатели\_эффективности свободный. Язык рус. (дата обращения 16.01.2018)
6. Гибкая методология разработки Scrum [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Scrum свободный. Язык рус. (дата обращения 16.01.2018)
7. http://digest.mbltdev.ru/digests/176