

МЕТОДИКА И ПОДХОДЫ ПОВЫШЕНИЯ КАЧЕСТВА РАЗРАБАТЫВАЕМЫХ ПРОДУКТОВ В ПРОДУКТОВЫХ КОМАНДАХ

Леухин Елисей Владимирович

Независимый исследователь

eles.leukhin@gmail.com

METHODS AND APPROACHES FOR IMPROVING THE QUALITY OF DEVELOPED PRODUCTS IN PRODUCT TEAMS

Ye. Leukhin

Summary. In the conditions of a rapidly changing market and the need to respond as quickly as possible to business initiatives, IT tasks include building processes that allow making large amounts of changes in the shortest possible time, without compromising on the quality of implemented solutions. An additional direction is to create conditions in which engineers are maximally immersed in the product, for an effective value delivery process. Product teams are designed to help solve these problems, the main goal of which is to improve metrics, study the user and influence business indicators.

Many companies work according to the scheme of product teams using flexible methodologies or working in a classical form, or combining them. This helps to develop products quickly, efficiently, taking into account the needs of users and achieving high business performance.

Depending on the volume of the product, the number of teams working on changes changes, as well as the number of interactions with external teams and systems changes.

The isolation of the teams may lead to the need for separate stages of assembling the entire solution and researching the results. This leads to an increase in the cost of development with an incremental-iterative approach and does not agree with the basic principles of the work of product teams, greatly increasing the lead time (the time of delivery of the solution to the industrial operation environment).

Keywords: methodology, quality improvement, development, products, team.

Для решения задачи повышения качества разрабатываемых продуктов в продуктовых командах предлагаю следующую методику, при использовании рекомендаций которой, будет обеспечиваться превентивное обеспечение качества и снижение затрат на внесение исправлений в процессе разработке цифровых продуктов.

Процесс работы над изменением продукта начинается с того, что продуктовая команда, прорабатывая пользовательскую историю. На этапе проработки беклога продукта или груминга беклога продукта. При работе с пользовательской историей команде необходимо использовать INVEST модель. Команда формирует основные критерии пользовательской истории, которые

Аннотация. В условиях быстро меняющегося рынка и потребности максимально быстро реагировать на бизнес инициативы, в задачи ИТ ложится выстраивание процессов, позволяющих делать большие объемы изменений в кратчайшие сроки, без компромиссов в качестве реализуемых решений. Дополнительным направлением является создание условий, в которых инженеры максимально погружены в продукт, для эффективного процесса доставки ценности. Помочь решить данные задачи призваны продуктовые команды, главной целью которых является, улучшение метрик, изучение пользователя и влияние на бизнес-показатели.

Множество компаний работают по схеме продуктовых команд используя гибкие методологии или работающие в классическом виде, либо комбинируя их. Это помогает развивать продукты быстро, качественно, учитывая потребности пользователей и достигая высоких бизнес-показателей.

В зависимости от объемов продукта меняется количество команд, которые работают над изменениями, так же, меняется количество взаимодействий с внешними командами и системами.

Изолированность команд может приводить к необходимости проведения отдельных этапов сборки всего решения и исследования результатов. Это ведет к увеличению стоимости разработки при инкрементальном-итеративном подходе и не согласуется с основными принципами работы продуктовых команд, сильно увеличивая lead time (время доставки решения в среду промышленной эксплуатации).

Ключевые слова: методика, повышение качества, разработка, продукты, команда.

должны выполняться и фиксирует приемочные критерии [3].

Команда формирует definition of done для пользовательской истории. При разработке необходимо обеспечивать выполнение прохождения этих критериев, первичный контроль прохождения приемочных критериев выполняется разработчиком.

Команда фиксирует набор артефактов, которые описывают критерии выполнения пользовательской истории. Артефакты используются при завершении этапа разработки, для проверки реализованной функциональности.

Артефакты результатов проведения исследования продукта сохраняются для возможности вести исторические наблюдения. Необходимо проводить ретроспективный взгляд на жизненный цикл продукта с изучением артефактов, которые были сохранены.

Необходимо изучить, случались ли проблемы в промышленной среде, что стало их причинами, какие действия проводились на этапах процессе создания продукта во время которых были допущены ошибки, что было отражено в артефактах. Как артефакты помогли разобрать проблему, которая была найдена в промышленной среде. Что необходимо добавить, чтобы при появлении ошибок из подобного класса, команда могла решить их быстрее. Что нужно сделать что бы избежать подобных ошибок.

Грумминг беклога продукта и проработка беклога, это встреча продуктовой команды, на которой сущности беклога продукта, прорабатываются, обсуждаются и проходит подготовка к следующему планированию спринта.

Грумминг критически важен в управлении продуктом, он направлен на поддержание беклога продукта актуальным и поддерживает готовность элементов беклога для предстоящих спринтов [7, с. 10].

Грумминг беклога иногда называют пред планированием. Владелец продукта и представители команды проводят его в середине спринта. В данном случае планирование и грумминг проводятся в один и тот же день недели. Это позволяет поддерживать эффективный ритм для всей команды [9, с. 53].

Основные преимущества проработки беклога продукта [10]:

- убрать неопределенности и неизвестные факты о пользовательских историях, что повышает эффективность продукта
- позволить избежать переработки решения на этапах разработки и тестировании
- позволить идентифицировать зависимости в команде и помочь снизить риски
- постоянное проведения грумминг встреч сохраняет время для команды разработки, для дальнейшего обсуждения в течении спринта, так как позволяет внести ясность разработчикам и тестировщикам о требованиях
- успешный грумминг приводит к более эффективному планированию спринта
- процесс грумминга дает владельцу продукта, менеджерам и бизнес аналитикам больше шансов улучшить требования, дополнительной информацией, если она требуется.

INVEST модель, акроним от independent, negotiable, valuable, estimable, small, testable. INVEST модель помо-

гает сформулировать пользовательскую историю таким образом, чтобы повысить эффективность работы над ней, во время проработки пользовательской истории команда контролирует выполнения требований модели.

Independent — пользовательская история должна быть независима от других историй. Позволяет избежать проблем с приоритизацией и планированием, которые могут возникнуть при большом количестве историй в одной области, которые зависят друг от друга. В случае если одна из них должна быть пересмотрена это может привести к эффекту домино для всех зависимых историй. Необходимо удерживать истории в максимально самодостаточном виде.

Negotiable — пользовательская история должна обсуждаться между заказчиками и разработчиками. Не должно быть полностью закрытых требований, они должны быть открыты для корректировки в случае необходимости [5, с. 587].

Valuable — основной приоритет пользовательских историй, критерий, который демонстрирует каким образом функциональность помогает клиентам и пользователям. В случае описания пользовательской истории, когда команда не может определить пользу или любую ценность истории, команда должна определить действительно ли нужна данная пользовательская история.

Estimable — возможность оценки, разработчики должны точно понимать, что требуется для создания функциональности. Пользовательская история должна быть точно определена и понятна для разработчиков, что бы они могли разделить ее на индивидуальные задачи и эффективно спланировать.

Small — пользовательская история не должна быть большой или избыточно документированной. Пользовательские истории должны быть короткие, лаконичные и легкие для понимания. Они не должны быть слишком большими, что сделает их сложными для планирования и должны быть достижимыми в течение 40 часов работы.

Testable — пользовательские истории должны быть тестируемы что бы подтвердить, что они готовы к внедрению. Пользовательские истории должны включать четкие критерии, которые позволяют измерить историю. У пользовательской истории должны быть критерии приемки.

Definition of Done, в методологиях разработки программного обеспечения это соглашение между продуктовой командой о наборе состояний, которые должны быть выполнены для заключения о том, что элемент беклога готов и выполнен.

Приемочные критерии, определены как состояния, которые должны быть достигнуты продуктом, пользовательской историей или инкрементом работы, чтобы они считались принятыми и завершенными.

Приемочные критерии определяют каким образом каждая функциональность будет использоваться с перспективы конечного пользователя. Они фокусируются на бизнес ценности, устанавливают границы набора функциональностей и направляют разработчиков. Они уникальные для каждой пользовательской истории и составляют основу приемочного тестирования пользовательской истории, которое используется как условие успешного выполнения задачи [2, с. 314].

Практикой, которая позволяет контролировать качество на самых ранних этапах разработки является DevTest.

Development testing, подход в разработке программного обеспечения который направлен на сближение фазы разработки и тестирования.

В традиционном процессе разработки программного обеспечения, разработка и тестирование два разных этапа, со своими специфичными функциями. Сложность такого подхода заключается в задержке между тем, когда код был написан и процессом, когда этот код будет протестирован.

В DevTest эти фазы намного больше интегрированы друг в друга, код будет написан и протестирован в автоматизированном или ручном виде. При использовании подхода проблемы будут найдены, изучены и проработаны намного быстрее и раньше.

Использование DevTest методологии позволяет получить множество преимуществ. Основные важные преимущества, следующие:

- повышенное качество кода в любой момент времени, так как новый коды непрерывно тестируется
- сокращение времени доставки новых функций в продуктивную среду

Следующий важный этап работы команды с требованиями, определение деталей реализации и полное понимание всех ветвлений разрабатываемой пользовательской истории. Активность 3 Амиго, на которой члены команды, нескольких ролей прорабатывают пользовательскую историю, которая будет реализовываться.

3 Амиго это процесс в гибких методологиях разработки, включающий активность, на которой аналитик/владелец продукта, разработчик и инженер по качеству, прорабатывают и обсуждают цели проекта или продукта, чтобы внести ясность по всем аспектам задачи.

Такая совместная работа позволяет членам команды быть в едином контексте и знать, что ожидается от каждой стадии разработки продукта.

Встреча может быть расширена представителями DevOps и другими.

В результате активности необходимо что бы все роли определили ключевые особенности реализации, какие могут быть сложности, какие части системы будут затронуты. Это необходимо для определения возможных объемов регрессий. Необходимо уточнить приемочные критерии (и сформировать тестовые сценарии), определить подходы с помощью, которых будут проводиться исследования (какие данные необходимы, какие состояния будут у системы), внести уточнения и дополнения в документацию [10].

Три основные цели, на которых необходимо фокусироваться во время митинга, это спецификации, тестовые сценарии и новые функции.

- Бизнес аналитик инициирует обсуждение требований к пользовательской истории. Обсуждается какую продуктовую проблему решает команда.
- Разработчик инициирует обсуждение функций продукта, обсуждая их пока команда не достигнет консенсуса по новым возможностям продукта. Обсуждается как команда будет строить, разрабатывать систему для решения этой проблемы.
- Специалист по качеству инициирует обсуждение дизайна спецификации, чтобы убедиться в высоком качестве и направляет разработчика в процессе создания новых пользовательских историй. Обсуждается как команда будет исследовать разработанное решение, какой негативный эффект для качества могут принести изменения [6, с. 280].

Этот подход позволяет построить общее понимание изменений, которые создаются в рамках данной итерации. Определить, не полностью проработанные зоны решаемой задачи, недопонимания, прояснить все неточности и неоднозначности на раннем этапе, это позволяет изучить то, что будет происходить в ближайшее время в процессе Деливери текущего инкремента [11, с. 54].

Подход позволяет выстроить изоляцию и концентрацию для определенного количества вовлеченных людей, в обсуждение конкретного программного инкремента.

Отсутствие практики три Амиго в продуктовых командах может приводить к тому, что каждый член команды имеет свое смешанное и отличающееся представление о том какой проект реализуется. Поэтому бизнес аналитик, разработчик и специалист по качеству должны совместно работать и экзаменовывать разработку продукта через призму того, что важно для их клиентов и бизнеса.

Практика Inner source. Стратегия разработки программного обеспечения, которая позволяет вносить изменения в код, владельцем которого является другая команда/проект/инженер и т.д. Позволяет выстроить культуру внутри организации в процессе разработки программного обеспечения.

Команды используют inner source для повышения прозрачности, усиления взаимодействия, и устранения разрозненности.

Для решения задач доработки и развития больших систем, командам необходимо использовать набор шагов и действий позволяющих повышать качество финального продукта.

По отдельным частям системы, микросервисами и наборам микросервисов объединённых одним функциональным доменом, необходимо формировать разделы документации, содержащие информацию о ролевом составе команды, технической документации, подходах к исследованию функциональности, подходов к внесению изменений, процессах верификации функциональности.

В ролевом составе команды необходимо отразить контакты и роль инженеров, отвечающих за часть системы, их зону ответственности и экспертизу.

В технической части необходимо зафиксировать ссылки на требования и артефакты части системы являющиеся наиболее важными и критичными при внесении доработок.

В подходах к исследованию необходимо зафиксировать ссылки на приемочные критерии, definition of done, критичные сценарии использования.

В подходах к внесению изменений необходимо отразить процессы, специфичные для данной части системы, которые используются при внесении изменений.

В процессах верификации необходимо отразить специфичные для части системы активности направленные на обеспечение качества.

Практика позволяет построить подход, в рамках которого любые изменения вносимые в части системы, владельцами которой является другая команда или инженер, проходят дополнительный этап верификации, со стороны инженеров, которые обладают максимальной экспертизой по данной части системы. Дополнительные разделы документации позволяют погрузиться в особенности реализации компонент или частей системы для того, чтобы вносимые изменения не приводили к деградации работоспособности внедренных функциональностей.

Практика направлена на выполнения условий конвенций, процессов исследования и использовании необходимых наборов инструментов, данных для процесса контроля качества.

Подход позволяет снизить количество времени, затрачиваемого на погружение инженеров в предметную область, при их введении в команду.

В ходе инкремента разработки, команда, которая делает изменения, знакомится с документацией и ключевыми подходами по работе в части системы, в случае необходимости обращается к необходимым ролям для уточнения деталей.

В ходе внесения изменений код проходит ревью со стороны владельцев компонента. Команда, вносящая изменения, в дальнейшем, передает все артефакты проделанной работы и выполненных изменений владельцу компоненты. Дальнейшее сопровождение функциональности будет производиться силами владельца компоненты. Владелец компоненты заинтересованы в том, чтобы внесенные изменения полностью соответствовали всем критериям качества, выставляемых к коду и реализации [12, с. 441].

Матрица покрытия требований. Подход описывающий документ, содержащий соответствие функциональных требований и сценариев исследования, которые используются для контроля качества.

Подход используется для того, чтобы показать, что требования были полностью покрыты проверками, матрица должна иметь прослеживаемость по всем артефактам, включенным в нее [7].

Двунаправленная прослеживаемость — это возможность отслеживания в прямом, например, от требования к сценарию исследования и обратном направлении, например, от сценария исследования к требованию.

Связанность матрицы позволяет проследить зависимость от требований к исходному коду, сценариям исследований, проведенным тестированиям и проблемам. От проблем обратно к требованиям. Необходимо иметь возможность проследить путь от требований к бизнес-целям или задачам, чтобы ответить на вопрос, почему существует данное требование.

Матрица покрытие требований позволяет формировать необходимый и достаточный объем сценариев исследований.

При внесении изменения в систему, опираясь на объем измененных требований, формируется объем сценариев использования необходимых для процедуры контроля качества.

Это позволяет командам контролировать ухудшение качества функциональных возможностей, минимальным количеством сценариев исследования.

Матрица покрытие требований позволяет давать более точные оценки на реализацию пользовательских историй, на основе объемов сценариев исследований участвующих в контроле качества. Большое количество сценариев исследования, необходимое для контроля качества, говорить о высокой сложности текущей реализации, внесение изменений в которую будет требовать повышенного внимания к регрессиям [11, с. 51].

Quality gate — практика, обеспечивающая соблюдение политик качества на проекте. Практика позволяет отвечать на вопрос, готов ли продукт к выпуску.

Чтобы ответить на этот вопрос, определяется набор условий, по которым производится оценка. Например, нет новых проблем уровня блокирующий, покрытие кода сценариями тестирования более 80%.

Практика использует определенные вехи в производственном процессе, во время которых продукт подвергается проверкам. Проверки направлены на выявление соответствия продукта критериям качества.

Продукт переходит на следующую фазу, только при выполнении политик качества. Практика является важным инструментом управления продуктом, используемым в крупных командах.

Необходимо использовать Quality Gate на всех этапах жизненного цикла программного обеспечения, таких как, проработка требований продукта, реализация продукта, тестирование продукта, стабилизация релиза продукта, приемочные испытания продукта.

На этапе стабилизации выводимого в промышленную среду продукта Quality Gate будет сформированный необходимый и достаточный объем тестовых сценариев, который будет выполняться. Объем тестовых сценариев должен быть сформирован на основе матрицы покрытия.

Для каждого релиза Quality Gate будет уникальным, в связи с разницей вносимых изменений. Что позволит максимально эффективно расходовать ресурсы команды.

Автоматизация приемочных сценариев, позволяет командам, получить повторяемость результатов проверок на всех этапах разработки.

Автоматизация дает максимальный эффект, когда используется на разных уровнях тестирования: юнит, компонентное, интеграционное, системное, E2E и UI.

При автоматизации приемочных сценариев, на ранних этапах, команды получают готовые проверки, артефакты и критерии качества, по которым будут оцениваться результаты работы.

Автоматизированные проверки будут расширять набор проверок входящих в Quality Gate.

Автоматизированные проверки позволяют повысить воспроизводимость результатов на этапе подготовки релиза продукта, а также увеличить покрытие функциональности продукта тестовыми сценариями для дальнейшего использования.

Метрики, ключевой компонент в эффективном менеджменте качества и его планировании, эти измерения помогают убедиться в том, что клиенты получают приемлемый продукт или поставку. Метрики должны трансформироваться в цели для достижения.

Метрики качества позволяют ответить на вопросы чем команда может управлять для оценки своей работы, а также при принятии решения о релизе.

Командам необходимо использовать в своей работе метрики качества продукта, для определения соответствия текущего уровня качества целевому уровню. Основными метриками, на которые команда должна опираться, являются следующие:

- соотношение выполненных тестов к общему количеству валидаций
- соотношение выполненных тестов к количеству пройденных
- количество дефектов после рефакторингов
- изменение числа дефектов при переходе между не промышленными средами
- дефекты и их серьезность в промышленной среде
- количество инцидентов в промышленной среде
- down time, время простоя системы в целом и отдельных компонент
- recovery time, время на восстановление после сбоя
- first pass yield, процент модулей, соответствующих спецификации без какой-либо доработки
- defect per unit, среднее значение, получаемое из общего количества дефектов и модулей
- fill rate, процент модулей заказанных к внедренным, к доставленным в промышленную среду
- shipping errors per shipment, общее количество дефектов поставок, по отношению к общему количеству поставок
- total test duration, общее время на тестирование, в разрезе ручного и автоматизированного прохождения
- requirements coverage / test cases by requirement, покрытие требований, какие функции валидиру-

- ются и как много проверок относится к конкретной юзер стори или требованию
- percentage of tests passed or failed, отношение числа пройденных проверок к проверкам, по которым появились замечания, из общего количество планируемых к проведению валидаций
- number of defects found in testing, количество дефектов, найденных в процессе контроля качества
- percentage automated test coverage of total coverage, процент автоматизированных проверок к общему количеству валидаций
- useful vs irrelevant results, отношение релевантных результатов к ложнопозитивным/ложнонегативным полезный результат, не зависит от финального статуса passed или failed, с причиной именно в дефекте не релевантный результат, результатом выполнения, является причина, связанная с изменениями в коде валидаций или проблемы с не промышленной средой
- percentage of broken builds, процент билдов не прошедших quality gate на этапе контроля качества.

При внедрении данных методик и подходов, в трех компаниях в сфере финтех и фудтех, были получены высокие результаты работы.

Компании получили рост качества выпускаемых продуктов до 70 %. Сокращение времени вывода в промышленную эксплуатацию за счет снижения количества доработок во время разработки продукта. Была повышена

точность выдаваемых командами оценок на 30 %, за счет точного определения сложности внедряемых изменений. Сокращено количество обращений пользователей.

Достигнут рост первичного качества продуктов, снижен lead time. Повышена погруженность всех членов команды в требования к юзер стори. Появилась прозрачность в объеме изменений подвергаемых валидациям. Командам разработки стало проще планировать внедрения новых функциональностей. Концентрация на определенном инкременте всеми ролями позволяет быстрее доводить изменение до среды промышленной эксплуатации не меняя контекст.

Данная комбинация подходов и практик, используемых в продуктовых командах, позволяет выстроить производственный процесс, который фокусирует внимание команды на каждом из этапов жизненного цикла программного обеспечения, для увеличения производительности и влияния на продукт. Обеспечивая превентивное воздействие на качество продукта и на качество самого процесса создания ценности. Подходы и практики используются в компаниях и ИТ департаментах, для создания среды, в которой возможны быстрые и эффективные решения задач. Компании используют методики в динамично развивающихся проектах и продуктах, помогая бизнесу получать решения и реализации в максимально короткие сроки с выполнением всех политик качества. Методики позволяют максимизировать использование бюджетов на развитие продуктов и систем.

ЛИТЕРАТУРА

1. Абаев А.Л., Алексунин, В.А. Международный маркетинг. учебник и практикум для бакалавриата и магистратуры / А.Л. Алексунин В.А. Абаев. — Люберцы: Юрайт, 2016. — 362 с.
2. Акканина Н.В. Маркетинг в агропромышленном комплексе: учебник и практикум для академического бакалавриата / Н.В. Акканина [и др.]; под редакцией Н.В. Акканиной. — Москва: Издательство Юрайт, 2019. — 314 с.
3. Бабкин А.В., Буркальцева, Д.Д., Костень, Д.Г., Воробьев, Ю.Н. Формирование цифровой экономики в России: сущность, особенности, техническая нормализация, проблемы развития // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Экономические науки. — 2017. — № 3.
4. Безрутенко Ю.В. Маркетинг в социально-культурном сервисе и туризме: Учебное пособие, 2-е изд. / Ю.В. Безрутенко. — М.: Дашков и К, 2016. — 232 с.
5. Благовещенский Д.И., Козловский В.Н. Комплексные программы улучшений в машиностроении: монография. Тула: Изд-во ТулГУ, 2021. 587с.
6. Вовченко Н.Г., Галазова, С.С., Сопченко, А.А. Влияние экономики пандемии на мировые тренды цифровой трансформации // Интеллектуальные ресурсы — региональному развитию. — 2020. — № 2. — С. 275–283.
7. Гагарина С.Н. Аутсорсинг как метод повышения эффективности управления закупками организации / С.Н. Гагарина, В.С. Жукова // Экономика и бизнес: теория и практика. — 2021. — № 11-2 (81). — С. 40–43. — DOI 10.24412/2411-0450-2021-11-2-40-43.
8. Гагарина С.Н. Интервальное прогнозирование объемов спроса на услуги субъектов естественных монополий с учетом неопределенности информации / С.Н. Гагарина, Ю.Е. Гагарин // Вестник университета. — 2013. — № 22. — С. 101–110.
9. Гагарина С.Н. Форсайт как инновационный инструмент предвидения будущего / С.Н. Гагарина, М.А. Леонова, Т.Е. Раецкая // Вестник Калужского университета. — 2020. -№4 (49). — С. 52–55.
10. Джабраилов Ш.В. Сравнительный анализ методологий разработки ПО Agile и Waterfall // Молодой ученый. — 2020. — № 52 (342).
11. Кондрашова Н.Г., Авраменко М.А. Управление временем как эффективное средство менеджмента // Экономика и бизнес: теория и практика. — 2020. — № 11-2 (69). — С. 51–54.
12. Постигая Agile: ценности, принципы, методологии / Эндриу Стеллман, Дженнифер Грин / Пер. с англ С. Пасерба. — 3-е изд. — М.: Манн, Иванов и Фербер, 2019. — 441 с.

13. Ручкина Г.Ф., Березин, М.Ю., Демченко, М.В. Внедрение и практическое применение современных финансовых технологий: законодательное регулирование. — М., 2020.
 14. Сысоева Л.А. Подходы к использованию методологии Scrum для управления проектами по автоматизации сквозных бизнес-процессов // International journal of advanced studies in computer engineering. — 2020. — № 1. — С. 21–30.
 15. Ткаченко И.Н., Сивокос К.К. Использование гибких технологий Agile и Scrum для управления стейкхолдерами проектов // Управленец. — 2017. — №4 (68). — С. 85–95.
-

© Леухин Елисей Владимирович (eles.leukhin@gmail.com)
Журнал «Современная наука: актуальные проблемы теории и практики»