

QA процесс, часть 2 (Подготовка,
тестирование, рапортование, финальная
фаза).

Урок 17.



План


1. С чего начать обеспечение качества.
2. Подготовка и планирование.
3. Тестирование.
4. Рапортование и финальная фаза.
5. Стандарты.
6. Инструментарий QA.



С чего начать обеспечение качества.

Создание организации по обеспечению качества начинается со следующих шагов. Их относительно легко реализовать, и они приносят действительно ощутимую пользу:

- Договоритесь об общих шаблонах
- Определите последовательность действий
- Убедитесь, что стандарты и процессы используются
- Проводите анализ выполненных проектов
- Анализируйте и учитесь, используя данные дефектов
- Используйте то, что Вы изучили



Для получения максимальной пользы, вам необходимо будет внедрить и постоянно улучшать как можно больше видов деятельности, направленных на улучшение качества.

Путь к хорошему качеству подразумевает некоторые затраты компании. Обычно нанимают человека или группу людей, которые будут нести ответственность за обеспечение качества, и делегируют им полномочий на проведение действий по обеспечению качества.


Обеспечение качества – это процесс обучения: изучение того, что работает не так и как это исправить; изучение того, что работает правильно и при каких обстоятельствах, а также как делать свою работу лучше с каждым новым проектом. Любая организация, вовлеченная в процесс Обеспечения Качества, постоянно обучается.

Самый первый шаг – это сделать Обеспечение Качества неотъемлемой частью разработки продукта.



Создание и Использование Шаблонов


У каждого разработчика есть свой любимый способ выполнения задачи, и практически каждый с радостью согласится обсудить те стандарты, которые он использует. Общие шаблоны предоставляют всем членам команды важную основу для сотрудничества. Когда каждый человек выполняет задачу своим способом, о сотрудничать не просто. Часто разработчик боится попросить помощи другого человека, потому что он может не согласиться с его подходом. А когда сотрудничества нет, такие различия в подходах могут препятствовать общему пониманию и накоплению знаний и опыта. Виды деятельности по Контролю Качества (анализ, рецензии и тестирование) принесут больше пользы и будут более продуктивны, если продукт был сделан, используя общую модель. Без их использования рецензенты и специалисты по тестированию просто будут пытаться отловить проблемы везде, где касалась рука разработчика. Такой бессистемный подход к Контролю Качества требует больше усилий и приводит к плохому покрытию и слабому обнаружению дефектов.



Общие шаблоны способствуют улучшению технической работы.

Разработчик, выполняющий задачи своим собственным способом, может с легкостью пропустить важные детали или информацию. Когда работа стандартизирована, не возникает вопросов, что проделанная работа должна в себя включать.

Стандарты должны применяться при написании тест планов, спецификаций, пользовательских интерфейсов, документации, тренинговых материалов и других продуктов, т.к. общее видение того, как проект должен быть сделан, может помочь обеспечить его качество.



Создание инструкций или определение последовательности действий.

Каждая организация использует установленные процессы для повседневных задач. QA специалист контролировать следующие вопросы:

1. Отвечают ли эти уставные процесс нуждам компании?
2. Часто ли их используют в соответствующих ситуациях?



Первый вопрос обращает внимание непосредственно на качество самих процессов.

Достигают ли они своей цели?

В зависимости от цели вы можете задать такой вопрос:

обеспечивают и стимулируют ли они необходимый уровень сотрудничества?

Способствуют ли они достаточному взаимодействию между командой разработчиков и заказчиками?

Поддерживают ли они лучшие наработки технических стандартов?

Помогают ли они достичь целей по качеству?

Часто люди недостаточно хорошо знают процессы, которые сами и используют. Например, процессы могут мешать взаимодействию людей, совершенствованию технического мастерства или просто не отвечать нуждам команды. Второй вопрос обращает внимание на качество следования установленным процессам.



Использование Стандартов и Процессов.

Чтобы получить пользу от использования внедренных стандартов и процессов, мы должны постоянно контролировать, что все делается согласно установленной договоренности, и мы получаем именно тот результат, который планировали. Все что регулярно не используется, рано или поздно перестает существовать. Это закон человеческого поведения. Интегрированная модель зрелости процессов программного обеспечения (CMMI - Capability Maturity Model Integration) реализует это при помощи аудитов (CMMI определяет аудит, как вид деятельности по Обеспечению Качества, потому что данная модель тестирует процессы, а не продукт). При использовании гибких (Agile) методик, например, Extreme Programming или SCRUM, для этой цели нанимают инструктора. Если принятый стандарт или процесс игнорируется, то необходимо выяснить, почему так происходит, потому что причины могут быть абсолютно разные.



Например:

- Человек просто забыл использовать какой-либо стандарт или процесс. Напомните ему
- Человек просто не знал о существовании стандарта или процесса или не знал, как именно его использовать. В таком случае необходимо улучшить коммуникации в команде или провести тренинг.
- Стандарт или процесс не подходит для данной задачи. Либо адаптируйте сам процесс, либо попробуйте найти альтернативный способ.
- Стандарт или процесс был неэффективным или слишком «громоздким» для данной ситуации. Необходимо упростить его так, чтобы он отвечал нуждам проекта.


Каждое нарушение стандарта или процесса – это возможность его изучить и улучшить, чтобы он соответствовал нуждам команды.



Анализ прошлых проектов

Изученные уроки или постпрограммы (Post Programs or Post Project Analysis) - это один из самых мощных инструментов упреждающего улучшения качества нашей работы.

Ретроспектива – это отдельно выделяемый отрезок времени, с целью обратить свой взгляд на проделанную работу, изучить полученный опыт и задать себе следующие вопросы: "Что было хорошо и как сделать так же в будущем?" и "Что было не так и как этого можно избежать?" Несмотря на то, что ретроспективы относят к лучшим практикам, используются они крайне редко. Две основные причины этого: «Сложно собрать всю команду на семинар по ретроспективе» и «Многие считают, что когда-то это делали, но это не приносило никакой пользы».



Первая причина возникает из-за того, что семинары по ретроспективе происходят в конце разработки проектов. Большинство членов команды уже работают на других проектах, а те, кто остались, заняты релизом проекта или его поддержкой. Гибкие методика решают эту проблему очень просто: не стоит делать всего лишь одну ретроспективу в конце проекта, необходимо делать ретроспективы на всем его протяжении. Преимущества: - Члены проекта всегда доступны, потому, что они все еще закреплены за проектом - Ежемесячный семинар по итогам, например, одной фазы проекта гораздо легче запланировать и он займет всего около часа (вместо целого дня). - Весь полученный опыт и наработки все еще свежи в памяти, и вряд ли что-то будет упущено - Полученные уроки можно применить к оставшейся части проекта. Вторая причина непопулярности ретроспектив – очень часто удается собрать много интересной и полезной информации, но нет никакой возможности использовать полученные данные на практике в будущих проектах.




Использование Данных Дефекта

Информация - это записи о просчетах в качестве, а это – список возможностей для улучшения качества на будущих проектах. Дефекты необходимо обязательно документировать.

Информация о дефектах, которая может быть полезна для улучшения качества, включает следующие вопросы:

- Что было не так? Решать нужно саму проблему, а не ее симптомы.
- Когда была создана эта проблема? Какое именно действие при разработке явилось ее источником? Это была проблема в требованиях? Проектировании системы? Коде? Тестировании?
- Когда проблема была выявлена? Может она и не была сразу же устранена, но что нас интересует, сколько она существовала до того, как мы ее обнаружили
- Каким образом была найдена эта проблема? Способ обнаружения можно внедрить в постоянно используемую практику.
- Можно ли было обнаружить ее раньше? Есть ли какой-либо процесс Контроля Качества,



- Сколько стоило устранение этой проблемы? Этот момент очень легко недооценить. Необходимо убедиться, что мы учли процесс диагностики проблемы и всю работу по ее устранению, которую пришлось сделать, включая ре-дизайн, переписывание кода, ре-компиляцию, переработку тестов, повторное тестирование, повторный релиз, выпуск заплатки, формирование отчета по дефекту, отчет по статусу проекта и т.д.

- Какого рода была эта проблема? Когда у нас огромное количество дефектов, их категоризация облегчает анализ и обучение.

- Когда мы анализируем информацию о дефектах, то необходимо искать те дефекты, которые обнаруживаются регулярно, и те, затраты на устранение которых высоки. Вот как раз таких дефектов и нужно избегать в будущем (или по крайней мере устранять их на более ранней стадии разработки), именно такая тактика гарантированно будет способствовать улучшению качества.



Используйте полученные знания

Многие виды деятельности по Обеспечению Качества, предоставят нам просто невероятный объем информации о возможностях улучшить качество создаваемого продукта. Но эта информация сама по себе не гарантирует желаемого качества. Мы должны особым образом применять на практике все то, чему мы научились. Например, если процесс разработки дизайна неэффективен для использования на определенных видах проектов, то тогда мы должны разработать альтернативный процесс и использовать его на всех будущих проектах такого типа. Регулярно рассматривать новые возможности улучшения стандартов, процессов и методов и вносить необходимые корректировки. Планируйте для этого специальное время и назначайте ответственных людей, иначе данный вид деятельности просто не будет иметь смысла. В начале разработки нового проекта необходимо обернуться назад и извлечь уроки из предыдущих проектов. Просмотреть базу знаний накопленного опыта и историю дефектов, и определить, что должно быть улучшено, основываясь на анализе прошлых проектов. Какие действия можно предпринять в этот раз, чтобы обеспечить лучшее качество продукта, чем то, которое было достигнуто до этого? Причем, это нужно делать при инициировании каждого проекта, иначе это будет пропускаться под натиском новых проектов.



2. Подготовка и планирование

Этапы подготовки и планирование в процессе QA занимают очень важную роль.

Планирование качества – процесс определения требований и стандартов качества для проекта и продукта, а также документирования того, каким образом будет продемонстрировано и подтверждено соответствие продукции проекта и самого проекта установленным требованиям и стандартам. Результатом этого процесса является план управления качеством, который включает:

- цели и критерии обеспечения качества;
- методы достижения целей по качеству;
- описание превентивных мер по обеспечению качества в проекте.

Планирование качества - это определение стандартов и критериев качества, на которые нацелен проект, и средств удовлетворения этих стандартов. Это один из ключевых вспомогательных процессов планирования и он должен осуществляться регулярно и параллельно с другими процессами планирования. Так, желательное качество процесса может потребовать корректировки стоимости или расписания работ, а желательное качество продукта - детального анализа рисков потенциальных проблем



3. Тестирование.

Процесс QA на этапе тестирования предполагает:

Регулярное тестирование

- прогон тестов
- анализ результатов
- регистрация дефектов
- предоставление отчетов

Контроль качества

- контроль параметров качества
- контроль исполнения плана качества

Управление качеством

- улучшение процессов
- выработка рекомендаций

Обучение команды.

Поддержку тестовой базы.



4. Рапортование и финальная фаза.

На этапе эксплуатации и сопровождения процесс QA предполагает:

- Регулярное тестирование
- Приоритезация дефектов полученных от пользователей
- Разработка дополнительных тестов на проблемные области
- Управление версиями
- Управление качеством
- Обучение команды

На этапе финальной фазы процесс QA предполагает:

- Архивирование тестовой инфраструктуры
- Анализ успешности проекта и документирование его результатов



5. Стандарты

Quality Assurance (Гарантия качества) - обязательство, что работа выполнена с соблюдением соответствующих стандартов.

Стандарты:

ISO 9000 CMM/CMMI PMBoK




6. Инструментарий QA.

Инструменты поддержки совместной разработки кода.

Инструменты: CVS, TeamWare(SCCS), SubVersion, VCS ...

Функции:

- Несколько человек могут править один и тот же файл одновременно
- Поддержка версионности на уровне файлов и всего продукта
- Поддержка change-sets
- Мультиплатформенность
- command-line и GUI интерфейсы




Инструменты для регистрации дефектов

Инструменты: Bugzilla, Mantis, Jira...

Функции:

- Учет и регистрация найденных дефектов или заданий для разработки
- Классификация дефектов
- Поддержка жизненного цикла дефекта
- Сохранение истории коммуникации по дефектам




Тестовая база Функции:

- Структурированное хранилище тестов и всего необходимого для их исполнения
- Тесты объединены в сюты
- Позволяет определить специфические условия исполнения для каждого теста или для всей сюты



Тестовый отчет

- Версия продукта
- Дата прогона
- Метрики качества по отношению к предыдущему тестированию, а также к тестированию эталонного компилятора
- Список новых падений
- Список известных падений
- Список новых тестов
- Список прошедших тестов



База данных результатов тестирования Функции:

- Хранение результатов тестирований за длительный срок
- Сравнение произвольного набора тестирований между собой
- История жизни любого теста из тестовой базы
- Использование общей статистики для оптимизации тестовых прогонов

Диаграмма связей — это инструмент управления качеством, основанный на определении логических взаимосвязей между различными данными. Применяется этот инструмент для сопоставления причин и следствий по исследуемой проблеме.

