Тестирование и качество

Тестирование ПО

Тестирование программного обеспечения - это процесс анализа программного средства и сопутствующей документации, с целью выявления дефектов и повышения качества продукта.

В более широком смысле, **тестирование** - это одна из техник контроля качества, включающая в себя активности по **планированию работ** *(Test Management)*, **проектированию тестов** *(Test Design)*, **выполнению тестирования** *(Test Execution)* и **анализу полученных результатов** *(Test Analysis).*

Тестирование является одним из наиболее устоявшихся **способов обеспечения качества** разработки программного обеспечения и входит в набор эффективных средств современной системы обеспечения качества программного продукта.

Что представляет собой «качественное ПО»?

- ✓Его легко использовать.
- ✓Оно демонстрирует хорошую производительность.
- ✓В нем нет ошибок.
- ✓Оно не портит пользовательские данные при сбоях.
- ✓Его можно использовать на разных платформах.
- ✓Оно может работать 24 часа в сутки и 7 дней в неделю.
- ✓В него легко добавлять новые возможности.
- ✓Оно удовлетворяет потребности пользователей.
- ✓Оно хорошо документировано.
- ✓…

Что представляет собой «качественное ПО»?

Качественный – значит соответствующий ожиданиям того, кому этот продукт предназначается.

Для этого нужны требования к продукту.

В зависимости от того, какие это требования – соответственно, нужно выбирать подход, чтобы проверить как они соблюдаются.

Качество

Полнота свойств и характеристик продукта, процесса или услуги, которые обеспечивают способность удовлетворять заявленным или подразумеваемым потребностям *[согласно ISO 9000:2000. Система менеджмента качества].*

Качество программного обеспечения - это степень, в которой ПО обладает требуемой комбинацией свойств *[1061-1998 IEEE Standard for Software Quality Metrics Methodology]*.

Качество программного обеспечения - это совокупность характеристик ПО, относящихся к его способности удовлетворять установленные и предполагаемые потребности *[ISO 8402:1994 Quality management and quality assurance]*.

Модели качества ПО

Модель качества программного обеспечения - это упорядоченная система атрибутов, вместе и по отдельности значимых для заинтересованных сторон проекта разработки ПО (представители заказчика, пользователи, разработчики и т.д.)

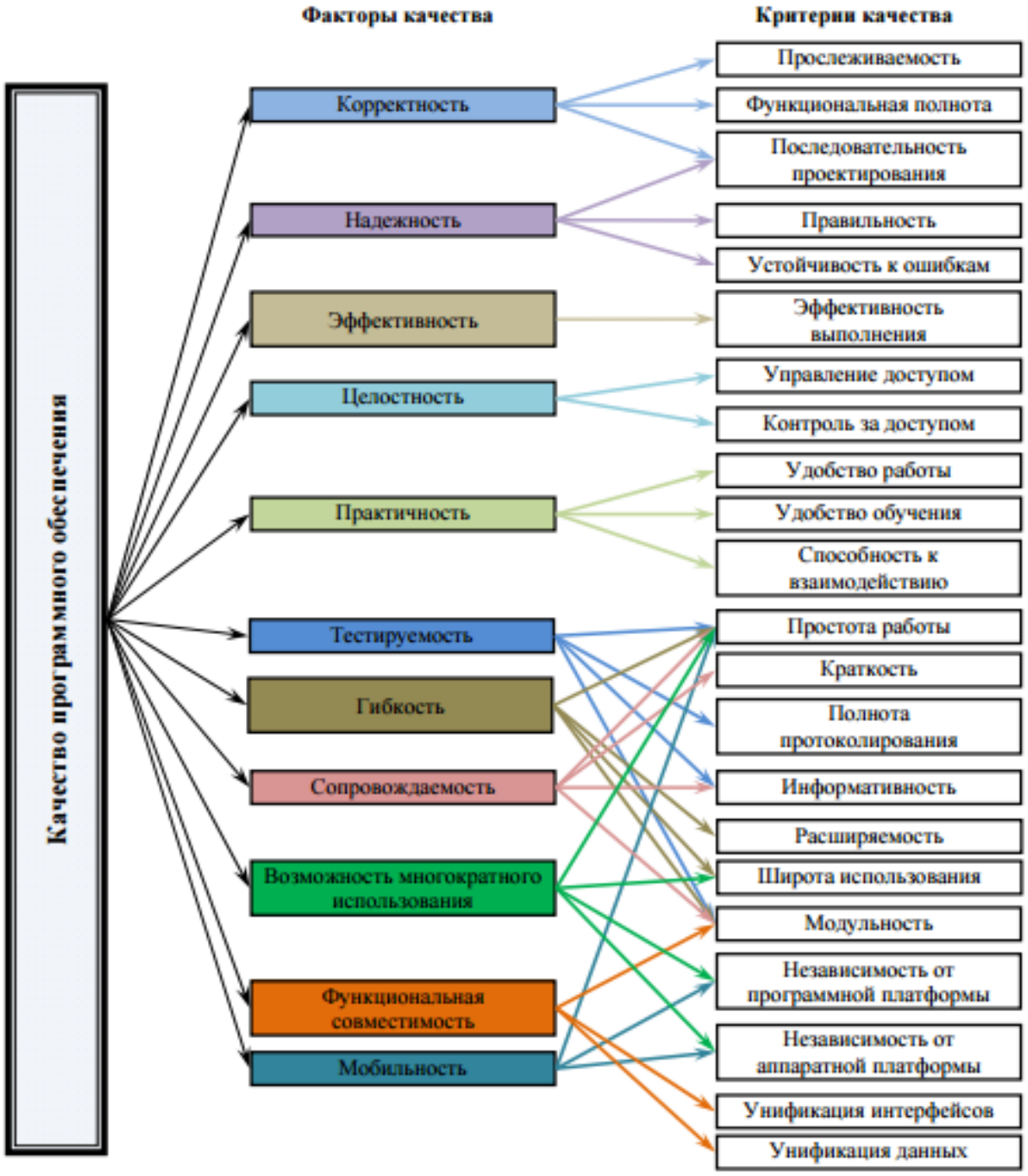
Наиболее известные модели качества:

- модель **МакКола** и др.
- модель **Боэма**
- модель **ISO 9126/ ISO 25000**

Качество ПО по МакКолу

Предложенная в 1977 г. Дж. МакКолом (J. McCall), П. Ричардсом (P. Richards) и Дж. Уолтерсом (G. Walters) модель качества ПО подразделяет атрибуты качества на три категории:

- ✓ **Факторы** *(factors)* – описывают ПО с точки зрения пользователя, определяются требованиями и группируются по видам деятельности заинтересованных лиц
- ✓ **Критерии** *(criteria)* – числовые уровни факторов: описывают ПО с точки зрения разработчика, задаются как цели при разработке
- ✓ **Метрики** *(metrics)* – служат для количественного описания и измерения качества (оценки от 0 до 10)


Факторы качества
Критерии качества
Качество программного обеспечения
Корректность
Надежность
Эффективность
Целостность
Практичность
Тестируемость
Гибкость
Сопровождаемость
Возможность многократного использования
Функциональная совместимость
Мобильность
Прослеживаемость
Функциональная полнота
Последовательность проектирования
Правильность
Устойчивость к ошибкам
Эффективность выполнения
Управление доступом
Контроль за доступом
Удобство работы
Удобство обучения
Способность к взаимодействию
Простота работы
Краткость
Полнота протоколирования
Информативность
Расширяемость
Широта использования
Модульность
Независимость от программной платформы
Независимость от аппаратной платформы
Унификация интерфейсов
Унификация данных

качества ПО

Факторы качества ПО. Треугольник МакКола

Метрики качества ПО по МакКолу

- ✓Удобство проверки на соответствие стандартам *(auditability)*
- ✓Точность управления и вычислений *(accuracy)*
- ✓Степень стандартности интерфейсов *(communication commonality)*
- ✓Функциональная полнота *(completeness)*
- ✓Однородность используемых правил проектирования и документации *(consistency)*
- ✓Степень стандартности форматов данных *(data commonality)*
- ✓Устойчивость к ошибкам *(error tolerance)*
- ✓Эффективность работы *(execution efficiency)*
- ✓Расширяемость *(expandability)*
- ✓Широта области потенциального использования *(generality)*

Метрики качества ПО по МакКолу

- ✓Независимость от аппаратной *платформы (hardware independence)*
- ✓Полнота протоколирования ошибок и других событий *(instrumentation)*
- ✓Модульность *(modularity)*
- ✓Удобство работы *(operability)*
- ✓Защищенность *(security)*
- ✓Самодокументированность *(selfdocumentation)*
- ✓Простота работы *(simplicity)*
- ✓Независимость от программной платформы *(software system independence)*
- ✓Возможность соотнесения проекта с требованиями *(traceability)*
- ✓Удобство обучения *(training)*

Качество ПО по Боэму

- В 1978 Боэм предложил свою модель, по существу представляющую собой расширение модели МакКола.

Атрибуты качества группируются **по способу использования ПО** *(primary use)*.

В дополнение к факторам МакКола атрибуты качества по Боэму включают следующие:

- ✓ ясность *(clarity)*
- ✓ удобство внесения изменений *(modifiability)*
- ✓ документированность *(documentation)*
- ✓ способность к восстановлению функций *(resilience)*
- ✓ понятность *(understandability)*
- ✓ адекватность *(validity)*
- ✓ функциональность *(functionality)*
- ✓ универсальность *(generality)*
- ✓ экономическая эффективность *(economy)*

Качество ПО согласно ISO 9126

В 1991 году в качестве стандартной была принята модель качества ПО ***ISO 9126***.

Эта модель не является прямым расширением ранее предложенных. В ней оценка качества ПО основана на трехуровневом рассмотрении.

- **Цели** *(goals)* – то, что мы ходим видеть в ПО
- **Атрибуты** *(attributes)* – свойства ПО, показывающие приближение к цели
- **Метрики** *(metrics)* – количественные характеристики степени наличия атрибутов

ISO 9126. Цели качества.

Выделено 6 целей:

- ✓**функциональность** *(functionality)*
- ✓**надежность** *(reliability)*
- ✓**практичность** или **удобство использования** *(usability)*
- ✓**эффективность** *(efficiency)*
- ✓**сопровождаемость** *(maintainability)*
- ✓**переносимость** или **мобильность** *(portability)*

Цели подразделяются на 21 атрибут качества.

ISO 9126. Атрибуты качества.

1. Функциональность

Эта характеристика отвечает за то, что ПО работает исправно и точно, функционально совместимо, соответствует стандартам отрасли и защищено от несанкционированного доступа.

Функциональные возможности — способность программного средства обеспечивать решение задач, удовлетворяющих потребностям заказчиков и пользователей в заданных условиях.

- ✓ **Функциональная пригодность** *(suitability)* - набор атрибутов, определяющих назначение, номенклатуру, основные, необходимые и достаточные функции программного средства, соответствующие техническому заданию и спецификациям требований заказчика или потенциального пользователя.

1. Функциональность

- ✓ **Точность** *(accuracy)* - способность выдавать нужные результаты.
- ✓ **Способность к взаимодействию** *(interoperability)* - способность взаимодействовать с нужным набором других систем.
- ✓ **Соответствие** стандартам и правилам *(compliance)* - соответствие ПО имеющимся индустриальным стандартам, нормативным и законодательным актам, другим регулирующим нормам.
- ✓ **Защищенность** *(security)* - способность предотвращать неавторизированный, т.е. без указания лица, пытающегося его осуществить, и неразрешенный доступ к данным и программам.

2. Надежность

Способность ПО поддерживать определенную работоспособность в заданных условиях.

- ✓ **Зрелость**, **завершенность** *(maturity)* - величина, обратная частоте отказов ПО. Обычно измеряется средним временем работы без сбоев и величиной, обратной вероятности возникновения отказа за данный период времени.
- ✓ **Устойчивость к отказам** *(fault tolerance)* - способность поддерживать заданный уровень работоспособности при отказах и нарушениях правил взаимодействия с окружением.
- ✓ **Способность к восстановлению** *(recoverability)* - Способность восстанавливать определенный уровень работоспособности и целостность данных после отказа, необходимые для этого время и ресурсы.
- ✓ **Соответствие стандартам надежности** *(reliability compliance)*

3. Практичность

Удобство использования

Способность ПО быть удобным в обучении и использовании, а также привлекательным для пользователей.

- ✓**Понятность** *(understandability)* - показатель, обратный к усилиям, которые затрачиваются пользователями на восприятие основных понятий программного обеспечения и осознание их применимости для решения своих задач.
- ✓**Удобство обучения** *(learnability)* - показатель, обратный усилиям, затрачиваемым пользователями на обучение работе с программным обеспечением.
- ✓**Удобство работы** *(operability)* - показатель, обратный усилиям, предпринимаемым пользователями для решения своих задач с помощью ПО.
- ✓**Привлекательность** *(attractiveness)* - способность программного обеспечения быть привлекательным для пользователей.
- ✓**Соответствие стандартам** удобства использования *(usability compliance).*

4. Эффективность (Производительность)

Способность ПО обеспечивать необходимую работоспособность с учетом используемых вычислительных ресурсов в определённых условиях.

Можно определить эффективность и как отношение получаемых с помощью программного обеспечения результатов к затрачиваемым на это ресурсам всех типов.

- ✓**Временная эффективность** *(time behaviour)* - способность ПО выдавать ожидаемые результаты, а также обеспечивать передачу необходимого объема данных за отведенное время.
- ✓**Эффективность использования ресурсов** *(resource utilisation)* - способность решать нужные задачи с использованием определенных объемов ресурсов определенных видов. Имеются в виду такие ресурсы, как оперативная и долговременная память, сетевые соединения, устройства ввода и вывода и пр.
- ✓**Соответствие стандартам производительности** *(efficiency compliance)*.

5. Сопровождаемость

Удобство сопровождения

Удобство проведения всех видов деятельности, связанных с сопровождение программ.

- ✓**Анализируемость** *(analyzability)* или удобство проведения анализа - удобство проведения анализа ошибок, дефектов и недостатков, а также удобство анализа необходимости изменений и их возможных последствий.
- ✓**Удобство внесения изменений** *(changeability)* - показатель, обратный трудозатратам на выполнение необходимых изменений.
- ✓**Стабильность** *(stability)* - показатель, обратный риску возникновения неожиданных эффектов при внесении необходимых изменений.
- ✓**Удобство проверки** *(testability)* - показатель, обратный трудозатратам на проведение тестирования и других видов проверки того, что внесенные изменения привели к нужным результатам.
- ✓**Соответствие стандартам удобства сопровождения** *(maintainability compliance).*

6. Переносимость (мобильность)

Способность программного обеспечения сохранять работоспособность при переносе из одного окружения в другое, включая организационные, аппаратные и программные аспекты окружения.

- ✓**Адаптируемость** *(adaptability)* - способность ПО приспосабливаться различным окружениям без проведения для этого действий, помимо заранее предусмотренных.
- ✓**Удобство установки** *(installability)* - способность ПО быть установленным или развернутым в определенном окружении.
- ✓**Способность к сосуществованию** *(coexistence)* - способность ПО сосуществовать с другими программами в общем окружении, деля с ними ресурсы.
- ✓**Удобство замены** *(replaceability)* другого ПО данным - возможность применения данного ПО вместо других программных систем для решения тех же задач в определенном окружении.
- ✓**Соответствие стандартам переносимости** *(portability compliance)*.

Методы контроля качества

Качество ПО контролируется с помощью процессов **верификации** и **валидации**.

Верификация обозначает проверку того, что программное обеспечение разработано в соответствии со всеми требованиями к нему, или что результаты очередного этапа разработки соответствуют ограничениям, сформулированным на предшествующих этапах.

Валидация — это проверка того, что сам продукт правилен, т.е. подтверждение того, что он действительно удовлетворяет потребностям и ожиданиям пользователей, заказчиков и других заинтересованных сторон.

Методы контроля качества

Методы контроля качества программного обеспечения можно классифицировать следующим образом:

- ✓ связанные с выяснением свойств программного обеспечения во время его работы (**все виды тестирования**)
- ✓ определения показателей качества на основе симуляции работы программного обеспечения с помощью моделей разного рода (**проверка на моделях** *(model checking)*, а также **прототипирование (макетирование).**
- ✓ нацеленные на выявление нарушений формализованных правил построения исходного кода программного обеспечения, проектных моделей и документации (**инспектирование кода**)
- ✓ обычного или формализованного анализа проектной документации и исходного кода для выявления их свойств (**методы анализа архитектуры** ПО).

Планирование тестовых испытаний

Планирование тестовых испытаний

В небольших проектах тестировщики ограничиваются заполнением стандартного шаблона, не стараются предположить проектные риски, не заботятся о критериях качества и т.д.

Приступать к тестированию следует уже на этапе сбора требований. И тестировать далее на последующих этапах вплоть до выхода продукта.

Часто продукт продолжают тестировать и на этапе поддержки.

Области ответственности тестировщиков:

Планирование тестов

- ✓ разработка методологии и плана тестирования
- ✓ участие в принятии стандарта качества
- ✓ разработка спецификаций тестов

Разработка и выполнение тестов

- ✓ создание ручных и автоматизированных тестов
- ✓ выполнение тестов
- ✓ управление билдами (оценка состояния проекта)

Отчеты по тестам

- ✓ сообщить проектной группе о качестве продукта
- ✓ отслеживать состояние дефектов

Планирование и тестовый план

Планирование - это процесс организации и установления процесса тестирования и согласование его со всеми ключевыми участниками проекта (менеджер проекта, лидер команды разработчиков, заказчик)

Тест план *(Test Plan)* - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения

Планирование и тестовый план. Задачи

1. Обеспечить ключевых участников проекта информацией о том, каким образом будет строиться процесс тестирования в проекте.

2. Убедиться, что тест-план реалистичен, полон и выполним.

Необходимые действия на стадии планирования

- ✓ Понять, что за продукт будет тестироваться, как он работает и для чего он нужен
- ✓ Понять, как продукт будет использоваться
- ✓ Хорошо изучить требования к продукту
- ✓ Решить какие части продукта будут тестироваться и как
- ✓ Убедиться, что выбранные области в полном объеме покрыты тестами
- ✓ Решить, какие из методов и техник тестирования наиболее эффективны для проверки нашего продукта
- ✓ Определить критерии качества продукта
- ✓ Определить и приоритизировать риски, то есть ситуации, которые приведут к ухудшению качества программного продукта. А также подумать о том, как предотвратить возникновение подобных ситуаций, и как из них выйти

Выводы по всем вышеперечисленным пунктам попадают в тест-план, который нужно утвердить и разослать всем заинтересованным лицам

Создать такое тестовое окружение, которое будет максимально приближено к условиям, в которых продукт будет эксплуатироваться *(production environment)*

Артефакты, создаваемые на стадии планирования

К **артефактам**, создаваемым на стадии планирования можно отнести:

- ✓ тестовый план;
- ✓ матрица конфигураций, которая может быть включена в тестовый план как отдельный раздел;
- ✓ запрос на выделение тестового оборудования.

Сложности планирования

- К сожалению невозможно всё предусмотреть и всё запланировать.
- Всегда будут возникать какие-то сложности или непредусмотренные ситуации во время работы. Будут возникать ситуации, когда будет **трудно расставить приоритеты** той или иной деятельности: например, *какие тесты выполнить в первую очередь, а какие выполнять необязательно, (или выполнить, если останется время); какие виды тестирования более приоритетны, а какие – менее; какая функциональность более важна, а какая – менее.*
- Также будут **возникать различные ограничения**, которые невозможно контролировать или на которые невозможно повлиять.
- Однако, всегда можно попросить помощи у менеджера, заказчика, группы разработчиков.
- Также, если видно, что что-то будет мешать тестированию, следует проинформировать об этом менеджера и заказчика и добиться понимания ситуации от них, а также согласовать действия, которые будут предприняты в данной ситуации.

Риски

Риск – сочетание вероятности наступления события и последствий, вызванных этим событием.

Думая о рисках, следует думать о том, какие риски могут возникнуть на проектном уровне.

Эти риски на первый взгляд могут не касаться тестирования, но на самом деле их последствия могут напрямую влиять на работу команды тестировщиков.

Группы риска

- **Человеческие** – недостаток квалификации, нехватка сотрудников, внутренние взаимоотношения, больничные листы (10%-20%), форс-мажор, отпуска, увольнение
- **Программно-аппаратные** – нехватка оборудования, выход из строя, устаревшее оборудование, поздняя постановка ПО, форс-мажор, отключение электричества (Internet)
- **Финансовые** - деньги

Тестовый план

Тестовый план – документ проектной документации, который описывает:

- **процесс тестирования конкретного продукта в конкретном проекте;**
- **что, когда, кем и как будет тестироваться;**
- **компоненты тестирования;**
- **команду тестировщиков;**
- **стратегию и методы тестирования;**
- **критерии качества и риски тестирования;**
- **график работ.**

Тестовый план. Назначение

- служит для поиска ошибок
- облегчает управление работами и контроль хода их выполнения
- облегчает организацию технических аспектов тестирования
- помогает организовать и скоординировать усилия сотрудников, разрабатывающих и тестирующих программный продукт
- повышает эффективность и полноту тестирования

Документация должна быть не объемной, а эффективной. Любые составляющие плана, не помогающие в поиске ошибок и организации тестирования, являются пустой тратой ресурсов

Тестовый план. Составление

Создание Test Plan

Тестовый план. Виды

1. **Мастер Тест План** *(Master Plan or Master Test Plan).*
   Создается в двух случаях:
   - ✓*если продукт имеет множество релизов или итераций, между которыми сохраняется общая информация, которую нет смысла повторять;*
   - ✓*если различные тестовые команды работают над одним продуктом, выполняя различные задачи, которые необходимо объединить в рамках одного документа.*
2. **Тест План** *(Test Plan)* - детальный Тест План.
   Составляется на каждый релиз/итерацию или для каждой команды в рамках проекта.
3. **План Приемочных испытаний** *(Product Acceptance Plan)*
   Документ, описывающий набор действий, связанных с приемочным тестированием

Тестовый план. Секции

Секции тестового плана включают в себя:

- Перечень работ
- Критерии качества и оценка качества процесса
- Оценка рисков
- Документация и письма
- Тестовая стратегия
- Ресурсы
- Метрики
- Расписание
- Ключевые точки

Тестовый план. Секции.

Перечень работ *(scope of work)*

Включается перечень функциональных областей приложений, которые будут подвергаться тестированию.

Здесь же может быть перечень компонентов или функциональности, которые не будут тестироваться по тем или иным причинам.

Например, эту функциональность реализует другая фирма. Или в проекте используются какие-то уже готовые компоненты. Если есть какие-либо ограничения тестирования, их тоже можно здесь перечислить.

Критерии качества и оценка качества процесса *(quality criteria and process assessment)*

Отражается перечень критериев качества, на основании которых будет приниматься заключение об уровне качества продукта и возможности передачи продукта заказчику.

Критерии качества относятся к качеству продукта.

В тестовых планах могут быть записаны и критерии качества самого процесса тестирования (и разработки), с целью последующей оценки качества процесса.

Это необходимо, чтобы в случае необходимости существовала возможность оценить, насколько грамотно был построен процесс тестирования, были ли какие-либо проблемы, и, если были, разобраться, почему, а также выработать рекомендации по их предотвращению в будущем.

Тестовый план. Секции.

Оценка рисков *(risk assessment)*

Описываются риски (негативные ситуации), которые могут возникнуть в процессе работы над проектом и помешать или негативно повлиять на тестирование продукта.

Кроме описания самого риска, следует указать вероятность его возникновения и степень влияния на проект. Производная этих двух величин даёт показатель, который может рассматриваться как степень серьёзности риска. Чем выше этот показатель, тем больше внимания нужно уделить этому риску: обдумыванию последствий, составлению плана его предотвращения или выхода из ситуации, если риск наступит. Если риск случился (т.е. ситуация наступила), возникает реальная проблема («issue», «hot issue»), которую необходимо решить.

Документация и письма

(test documentation and deliverables)

В этой секции размещается перечень артефактов (результатов деятельности).

Например:

- ✓тест план;
- ✓тестовые сценарии;
- ✓тестовые автоматические скрипты;
- ✓дефект-репорты;
- ✓отчеты о результатах тестирования.

Также указывается, кто будет высылать данный артефакт, как часто, каким способом, кому и т.д.

Тестовый план. Секции.

Тестовая стратегия *(test strategy)*

Здесь расписывается стратегия тестирования, методы и типы тестов, каким образом будет выполняться тестирование, почему именно так и т. д.

Конкретное содержание этого раздела зависит от фирмы-разработчика, проекта, заказчика и т. д.

Например, этот раздел может включать подразделы:

- критерии приемки билдов;
- методы тестирования;
- типы тестирования;
- уровни тестирования;
- отслеживание ошибок;
- использование метрик.

Тестовый план. Секции.

Ресурсы *(resources)*

Человеческие ресурсы: перечень ключевых людей на проекте (менеджер проекта, представители заказчика, лидер команды разработчиков и т.д.), список тестировщиков с их ролями на проекте, а также с зонами ответственности.

Аппаратные ресурсы *(hardware):* сюда входит перечень тестовых серверов и рабочих станций, инструментов, используемых для тестирования или для вспомогательных работ, описание тестового окружения.

Программные ресурсы *(software):* операционные системы, СУБД, серверы приложений, веб-серверы и т.д.

Тестовый план. Секции.

Метрики *(metrics)*

- Это числовая характеристика, позволяющая оценить тот или иной аспект программного продукта или процесса в целом.

Например: количество дефектов, найденных за неделю в тех или иных областях продукта. Эта метрика позволяет оценить, как много дефектов в той или иной функциональности. Что, в свою очередь может позволить сделать немало выводов или, например, подтолкнуть к разбору причин такого количества багов.

Расписание и ключевые точки
(schedule and milestones)

- Описывается график тестирования в согласовании с графиком выпуска билдов и проектным планом, который разрабатывается менеджером проекта.

Сюда же включаются основные даты (*milestones*): *например*, даты окончания промежуточных фаз работы над проектом.

График тестирования нужен для того, чтобы четко понимать, когда и что следует делать, ничего не пропустить, ничего не забыть и т.д. Он же упрощает контроль за ходом работ по тестированию, а также позволяет оценить текущую ситуацию, определить, все ли выполнено из того, что было запланировано.

Характеристики хорошего тест плана

Тест план должен быть:

- Полным
- Корректным
- Недвусмысленным.

Характеристики хорошего тест плана

В тест плане должны быть:

- ✓ определены цели тестирования, тестовый подход, стратегия, методы, виды тестирования. Запланированный подход должен быть реально выполним
- ✓ установлены реалистичные критерии качества.
- ✓ определены критерии прохождения приёмочного теста и условия прекращения тестирования.
- ✓ определены все артефакты, подлежащие сдаче, поставке или распространению (заказчику, проекту и т.д.)
- ✓ перечислены тестовые ресурсы (люди, оборудование) с указанием ролей, назначений, ответственности.

Характеристики хорошего тест плана

Также:

- ✓ Должно быть определено и описано тестовое оборудование, окружение, программное обеспечение.
- ✓ Должен быть определён график тестирования, он должен быть реалистичен и выполним.
- ✓ Тест-план должен соответствовать принятому в компании шаблону, если на проекте не решено иначе: например, использовать шаблон заказчика.

Шаблоны тест-планов

Для облегчения жизни тестировщикам, существуют несколько шаблонов тест-планов (**IEEE, RUP**).

http://www.protesting.ru/documentation/test_plan_template_rup.zip

http://www.protesting.ru/documentation/test_plan_template_ieee_829.zip

Тестовый план. Итак…

1. **Что надо тестировать?** (описание объекта тестирования: система, приложение, оборудование)
2. **Что будем тестировать?** (список функций и компонентов тестируемой системы)
3. **Как будем тестировать?** (стратегия тестирования, а именно: виды тестирования и их применение по отношению к тестируемому объекту)
4. **Когда будем тестировать?** (последовательность проведения работ: подготовка *(Test Preparation)*, тестирование *(Testing)*, анализ результатов *(Test Result Analisys)*, учет зависимостей тестовых активностей от задач разработки и смежных групп)
5. **Критерии начала тестирования:**
   - готовность тестовой платформы (тестового стенда);
   - законченность разработки требуемого функционала?
   - наличие всей необходимой документации.
6. **Критерии окончания тестирования:**
   - результаты тестирования удовлетворяют критериям качества продукта;
   - требования к количеству открытых багов выполнены.

Тестовый план. Также + …

После того, как написан черновик тест плана, необходимо дополнить его следующими пунктами:

1. **Окружение тестируемой системы** (описание программно-аппаратных средств)
2. **Необходимое для тестирования оборудование и программные средства** (программы для автоматизированного тестирования и т.д.)
3. **Риски и пути их разрешения**

Тестовый план. В заключении…

В самом общем виде – это документ, отражающий что и как делать, чтобы оттестировать программный продукт.

Тестовый план необходим для того, чтобы планировать работу, мобилизовать и распределить человеческие, временные и технические ресурсы.

В некоторых компаниях Тест План выглядит как график и расписание работ. Где-то это сугубо технический документ.

Нет общепринятого стандарта.

Есть тенденции, есть типовые главы Тест Плана.

Тест План должен содержать те главы, которые адекватны конкретно задаче и видению этой задачи.

Контрольные вопросы:

1. Что представляет собой качество ПО?
2. Модели качества ПО.
3. Модель качества МакКола.
4. Качество ПО Боэма.
5. Качества ПО согласно ISO 9126.
6. Цели и атрибуты качества ISO 9126.
7. Понятия верификация и валидация.
8. Методы контроля качества.
9. Основные задачи этапа планирования.
10. Что такое риски. Какие группы рисков существуют?
11. Что представляет собой тестовый план.
12. Описать основные секции тестового плана.
13. Перечислить характеристики хорошего тест-плана.