

ЖИЗНЕННЫЙ ЦИКЛ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Постоев Дмитрий

Часть 7:

Управление качеством (тестирование)

Определения

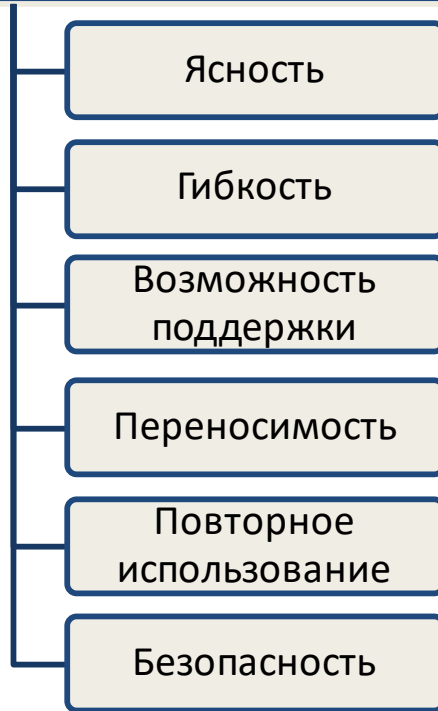
Качество (функциональное) – степень того, насколько система, компонент или процесс соответствуют заявленным требованиям, нуждам потребителей или их ожиданиям.^[1]

Гарантия качества (QA) – спланированный и систематичный порядок действий, необходимый для обеспечения достаточной уверенности в том, что изделие или продукт соответствуют установленным функциональным и техническим требованиям.

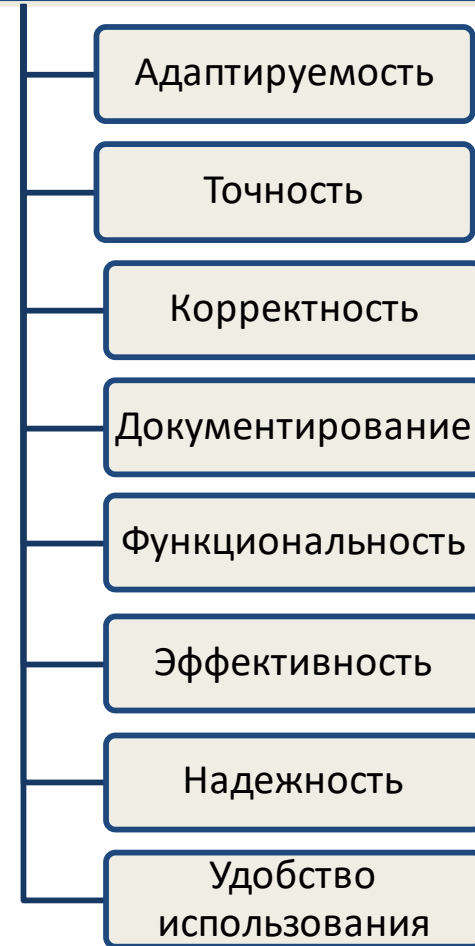
[1] IEEE 730

Характеристики качества

Структурные (внутренние)



Внешние

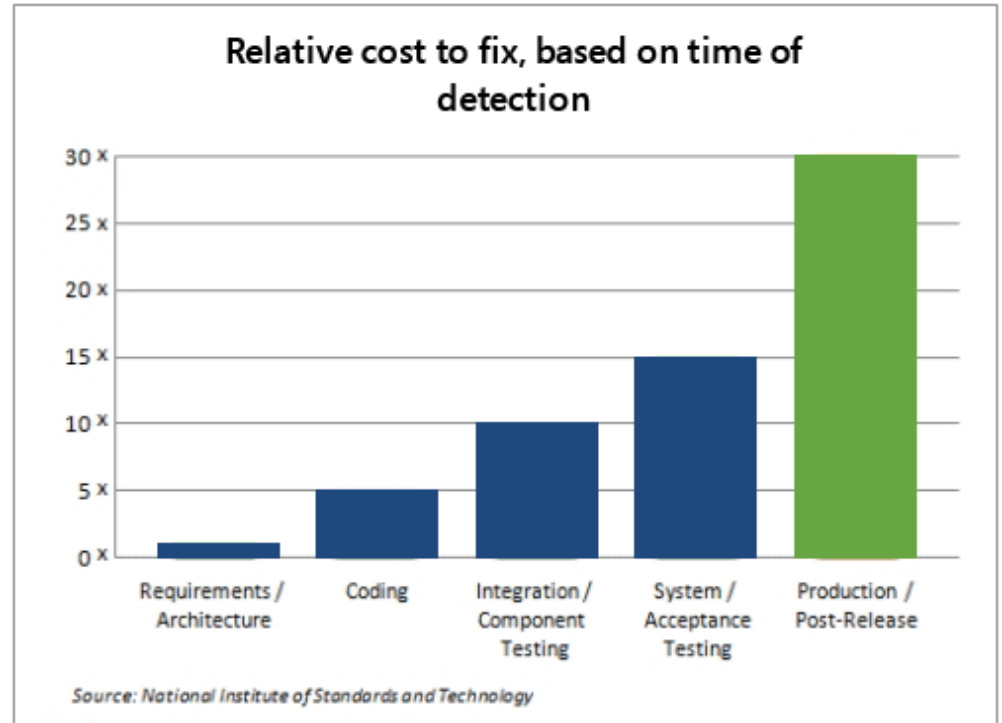


Гарантия и контроль качества

Признак	Гарантия качества	Контроль качества
Цель	Улучшение процесса разработки для избегания появления дефектов	Идентификация дефектов после того, как продукт был разработан, но до того как был выпущен
Как?	Создание хорошей системы управления качеством, оценка ее адекватности и периодический аудит	Поиск и устранение источников проблем качества с помощью инструментов, для удовлетворения требований
Ответственность	На всей команде, участвующей в создании продукта	Обычно на специальной команде, тестирующей продукт на дефекты
Примеры	Верификация	Валидация/Тестирование
Ориентация	На процесс создания	На итоговый продукт

Зачем?

1. Сдерживание затрат
2. Управление рисками



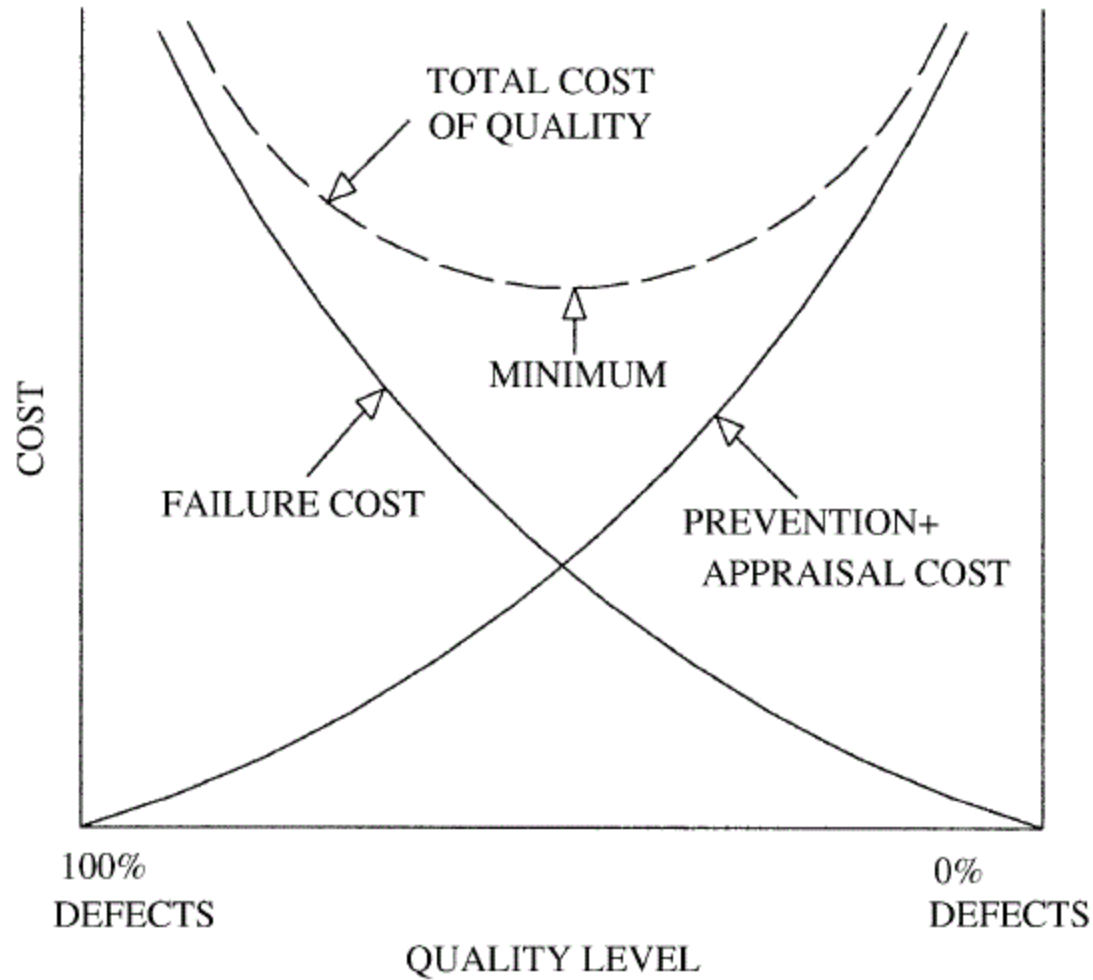
Всеобщее управление качеством

- общеорганизационный метод непрерывного повышения качества всех организационный процессов.

Принципы:

- Устранить необходимость в массовом контроле за счет повышения качества продукции
- Постоянно совершенствовать систему производства и обслуживания, улучшать качество и производительность, а, значит, снижать затраты
- Избавиться от лозунгов, призывов к повышению производительности и устранению дефектов
- Создайте программу самосовершенствования работников
- Все должны быть вовлечены в трансформацию процессов и заинтересованы в ее успехе

Оптимальное качество



Роли и обязанности

- Клиенты
- Архитекторы
- Разработчики
- Тестировщики
- Менеджеры



Отдел управления качеством

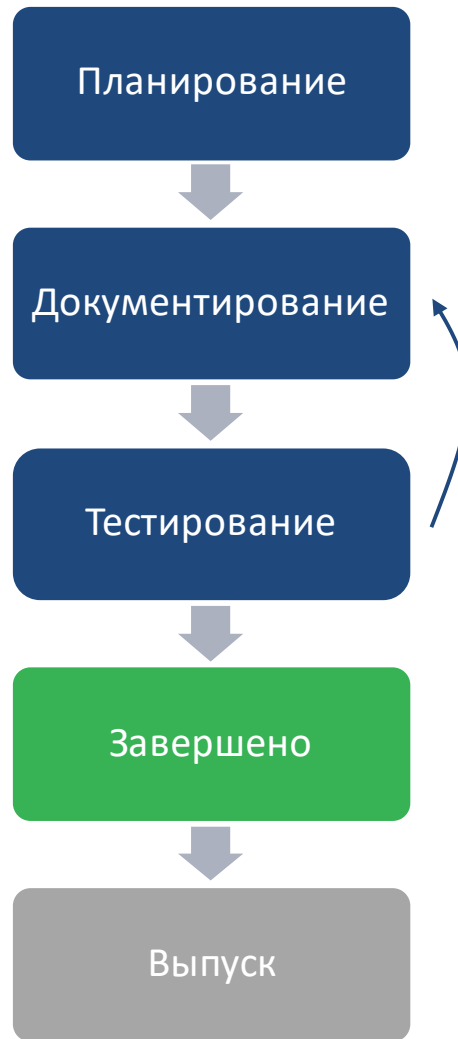
Обязанности:

- Мониторинг процесса разработки с целью выявления проблем
- Тестирование
- Отчетность о состоянии проекта, ответы на вопросы и предоставление информации для принятия решений

Роли:

- SQA Аналитик
- SQA Инженер
- QA Менеджер

Процесс управления качеством



Планирование

1. Определение политик и стратегии
2. Определение ролей и обязанностей
3. Анализ затрат и прибыли, рисков
4. Выбор типов, глубины и количества тестов, выявленный в ходе анализа
5. Определение потребностей в ресурсах
6. Планирование объектов и оборудования
7. Планирование мероприятий

Документирование

Фаза планирования:

- Обсуждение и принятие *плана тестирования*

Фаза тестирования:

- *Test Cases*: тесты, которые будут выполняться
- *Test Logs*: когда, кем, в каком окружении и каким результатом тесты были выполнены
- Дефекты/ошибки/улучшения

Фаза завершения:

- *Test Summary*: анализ логов и заключение о готовности к выпуску

Пример Test Case

Признак	Значение
Идентификатор набора тестов	TS001
Идентификатор тестового кейса	TC001
Описание	Проверить, что клик по кнопке «Сгенерировать» создает монеты в верном количестве
Связанное требование	RS001
Предварительные условия	<ol style="list-style-type: none">1. Пользователь аутентифицирован2. Баланс монет доступен
Тестовая процедура	<ol style="list-style-type: none">1. Выбрать номинал монеты в поле «Номинал»2. Ввести количество монет в поле «Количество»3. Нажать на кнопку «Сгенерировать»
Тестовые данные	<ol style="list-style-type: none">1. Номиналы: 0.05, 0.1, 0.25, 0.50, 1, 2, 52. Количество: 0, 1, 5, 10, 20

Пример Test Case (продолжение)

Признак	Значение
Ожидаемый результат	<ol style="list-style-type: none">1. Монеты заявленного номинала должны быть произведены, если Количество верно (1, 5, 10)2. Сообщение «Введите верное количество от 1 до 10» должно выводиться, если количество неверно
Полученный результат	<ol style="list-style-type: none">1. Если количество верно, то результат соответствует ожидаемому2. Если количество неверно, ничего не происходит, ожидаемое сообщение не появляется
Статус	Не пройден
Комментарий	Это пример тестового кейса
Кем проводился	Иванов Иван
Когда проводился	01/04/2017
Тестовая среда	ОС: Windows 7 SP1 Браузер: Chrome 56.0.2924.87

Техники тестирования

- Ручное и автоматизированное тестирование
- Функциональное (black-box)
- Структурное (white/gray-box)
- Ad-hoc (monkey-testing)
- Симуляция, моделирование
- Дымовое
- Регрессионное
- Производительности



Уровни тестирования



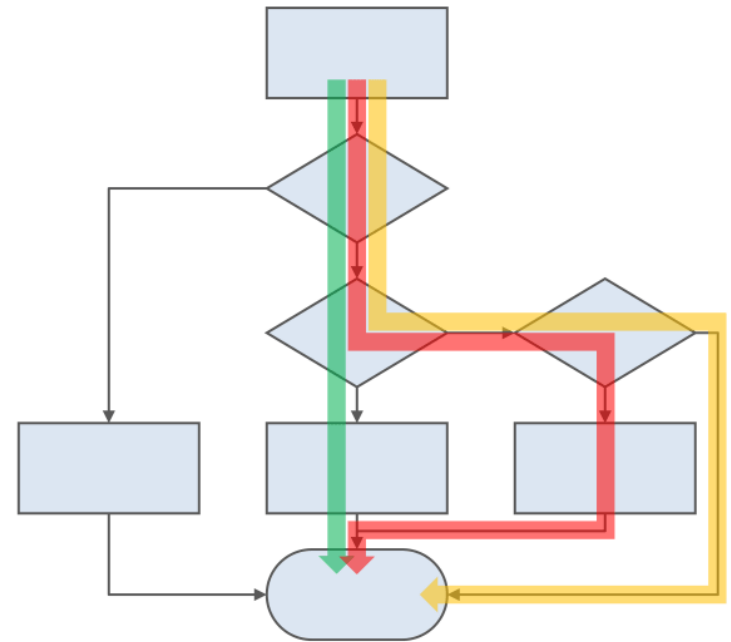
Тестовое покрытие (покрытие кода)

- мера, показывающая насколько код проекта покрыт тестами.

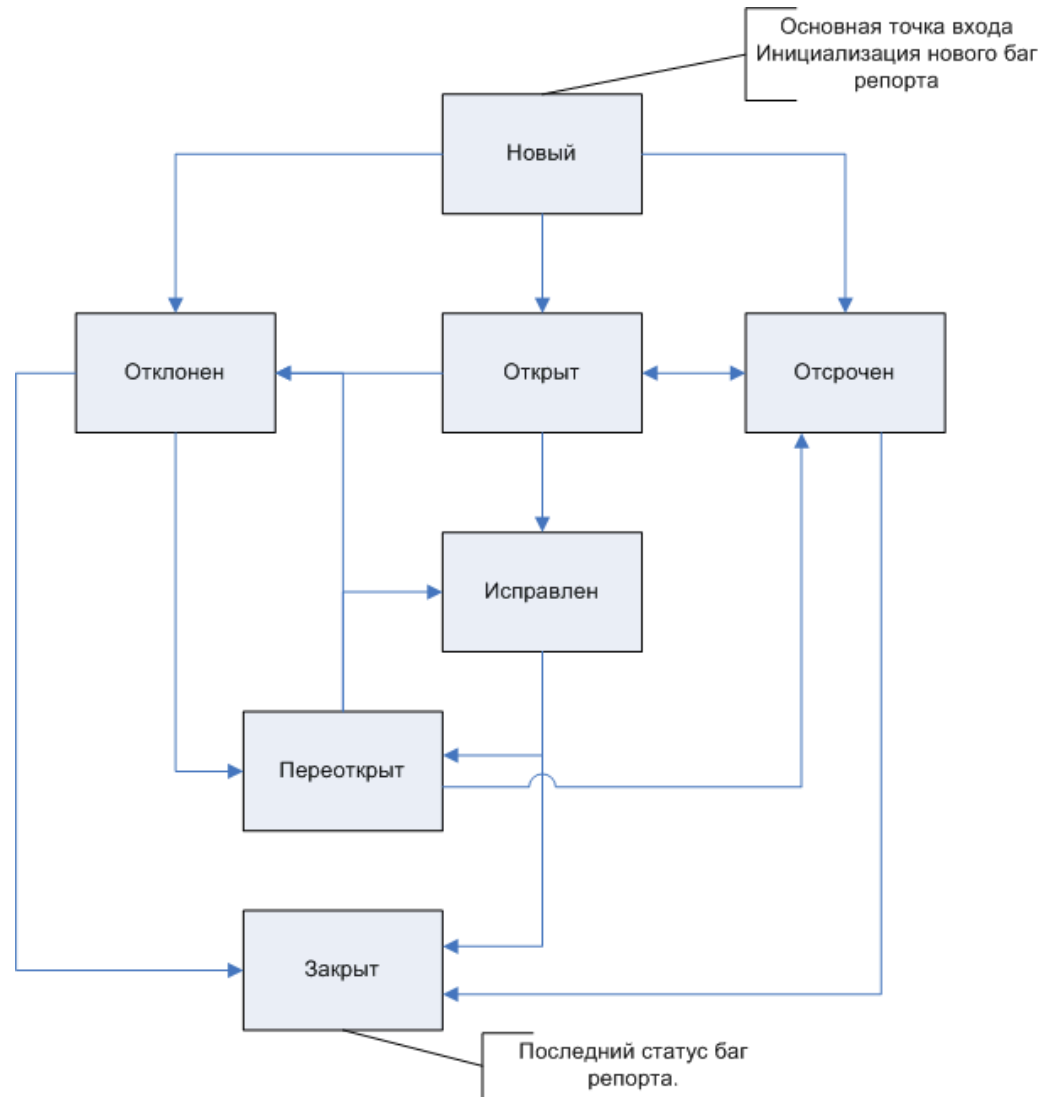
Критерии:

- Покрытие значений операторов
- Покрытие операторов
- Покрытие условий
- Покрытие путей
- Покрытие функций

```
...  
...  
...  
If ( powerOn ) {  
  if ( temp > 80 )  
    InitializeAC;  
  else if ( temp < 60 )  
    InitializeHeat;  
  } else {  
    DisplayMessage;  
  }  
...  
...  
...
```



Жизненный цикл бага



Bug Report

- документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Признак	Значение
Описание	Короткое описание проблемы, явно указывающее на причину и тип ошибочной ситуации.
Проект	Название тестируемого проекта
Номер версии	Версия на которой была найдена ошибка
Серьезность	<ol style="list-style-type: none">1. Блокирующий (Blocker)2. Критический (Critical)3. Значительный (Major)4. Незначительный (Minor)5. Тривиальный (Trivial)
Приоритет	<ol style="list-style-type: none">1. Высокий (High)2. Средний (Medium)3. Низкий (Low)

Bug Report (продолжение)

Признак	Значение
Статус	Статус бага. Зависит от используемой процедуры и жизненного цикла бага
Автор	Создатель репорта
Назначен на	Имя сотрудника, назначенного на решение проблемы
Окружение	Информация об окружении, на котором был найден баг: операционная система, сервис пак, для WEB тестирования - имя и версия браузера
Шаги воспроизведения	Шаги, по которым можно легко воспроизвести ситуацию, приведшую к ошибке.
Фактический результат	Результат, полученный после прохождения шагов к воспроизведению
Ожидаемый результат	Ожидаемый правильный результат
Дополнения	Файл с логами, скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на способ решения проблемы

1. Bug Tracking

Open-source: Bugzilla, Redmine, Github

Proprietary: JIRA, Microsoft TFS

2. Покрытие кода

Примеры: Visual Studio Enterprise, BullseyeCoverage, OpenCover.UI (расширение), CovTool,

3. Автоматизация действий пользователей

Примеры: Selenium, TestComplete, IBM Rational

4. Test Report

Примеры: TestLink, JIRA Zephyr, HP Quality Center, Microsoft Test Manager (TFS)

Литература

1. «The Art of Software Testing, 3rd Edition», Glenford J. Myers, Sandler, Badgett
2. «Testing Computer Software, 2nd Edition», Kaner, Falk, and Nguyen
3. <http://softwaretestingfundamentals.com/>

Домашнее задание

1. **Руководитель проекта** должен назначить задачу **Тестировщикам** на формирование Плана тестирования.
2. **Тестировщики** согласно Спецификации требований программного обеспечения подготавливают План тестирования.
3. **Аналитик** помогает **Тестировщикам** в подготовке Плана тестирования.
4. Согласно принятому Плану тестирования **Тестировщики** создают тестовое окружение.
5. **Тестировщики** по принятому Плану тестирования начинают проведение тестов для компонент.
6. В случае возникновения ошибок, по *Bug Report* **Руководитель проекта** создает *Issue* для исправления и назначает на ответственного за этот модуль **Разработчика**.

Примечание: Задачи назначаются и оформляются на Github