

Тренинги Intel Delta Course

«Дополнительные главы по Software Engineering»



Методы разработки ПО. Понятие тестирования и качества продукта. Оценка качества

Виценко А.Ю., Крюков А.К., Моренов О.А., Пряхин И.В., Семенов Д.С.,
Чиликин Е.В. Intel



Модели Процессов Разработки ПО

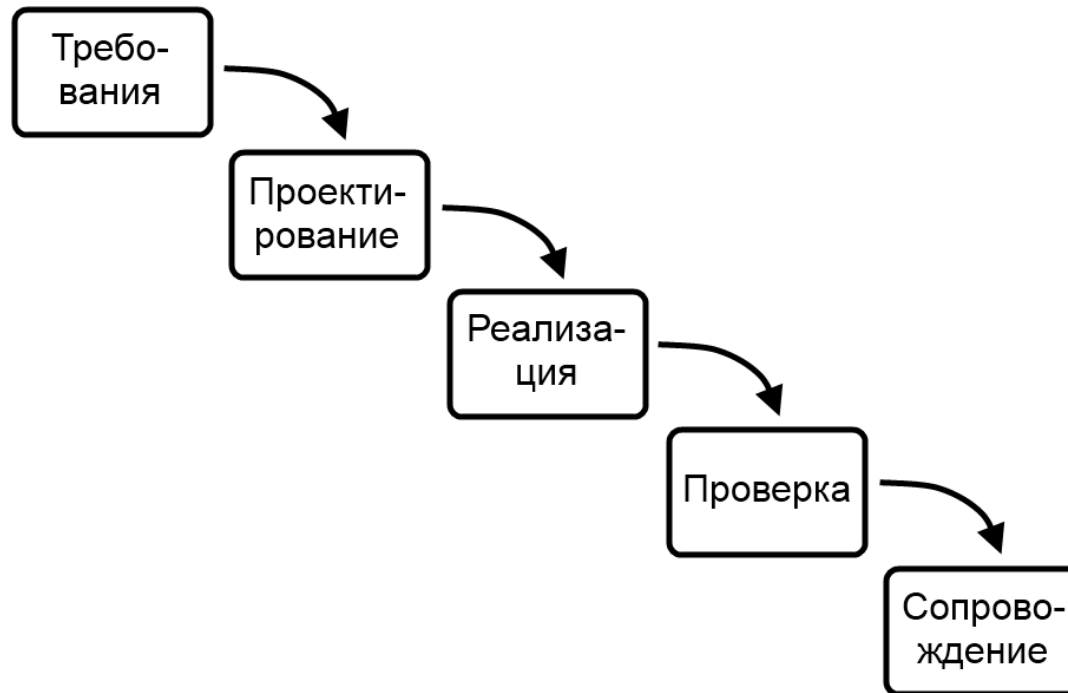
Понятие Качества Программного Продукта

Оценка Качества. Дефекты в ПО

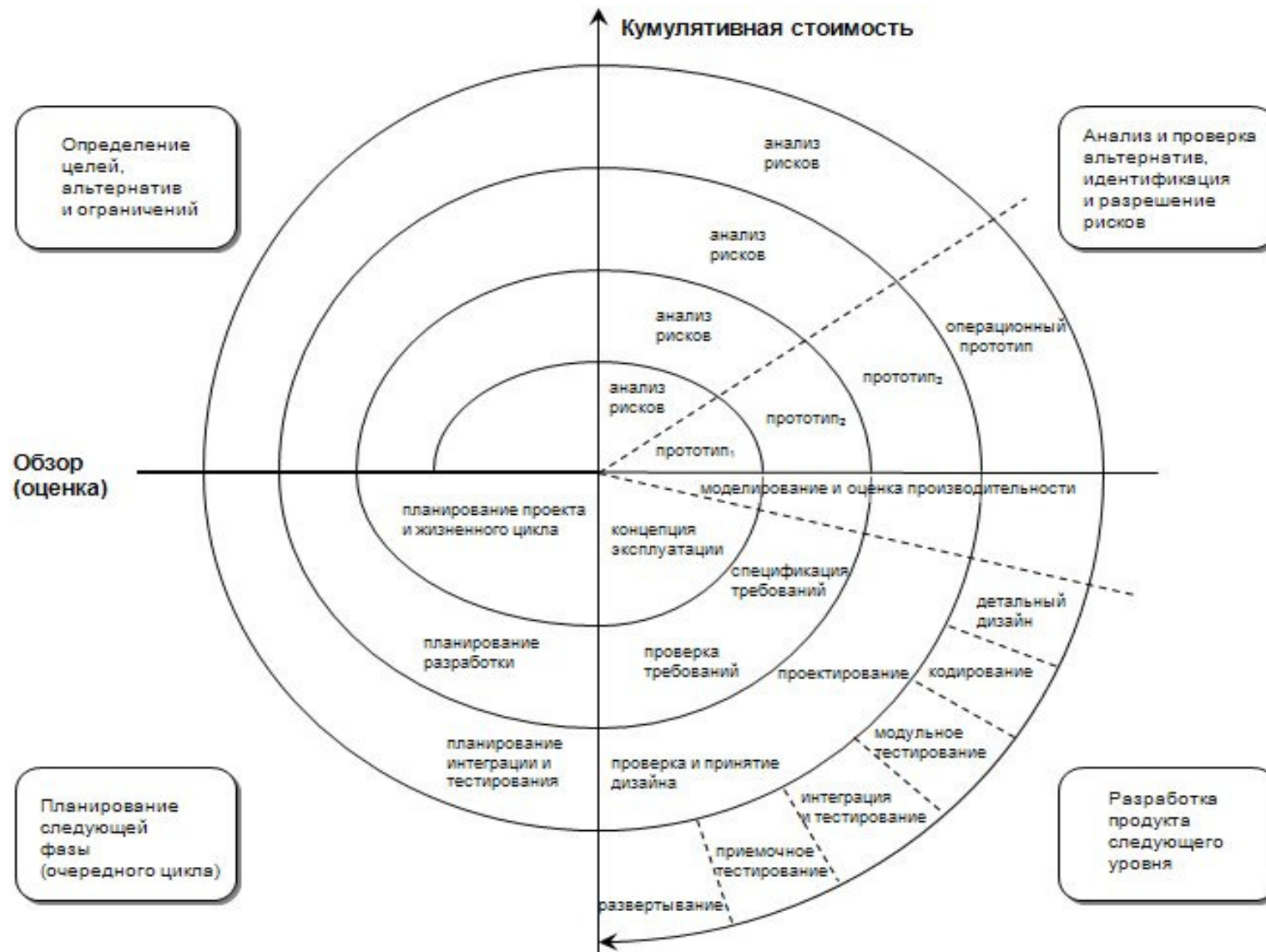
Понятие Тестирования

Виды Тестирования

Каскадная модель (Waterfall)

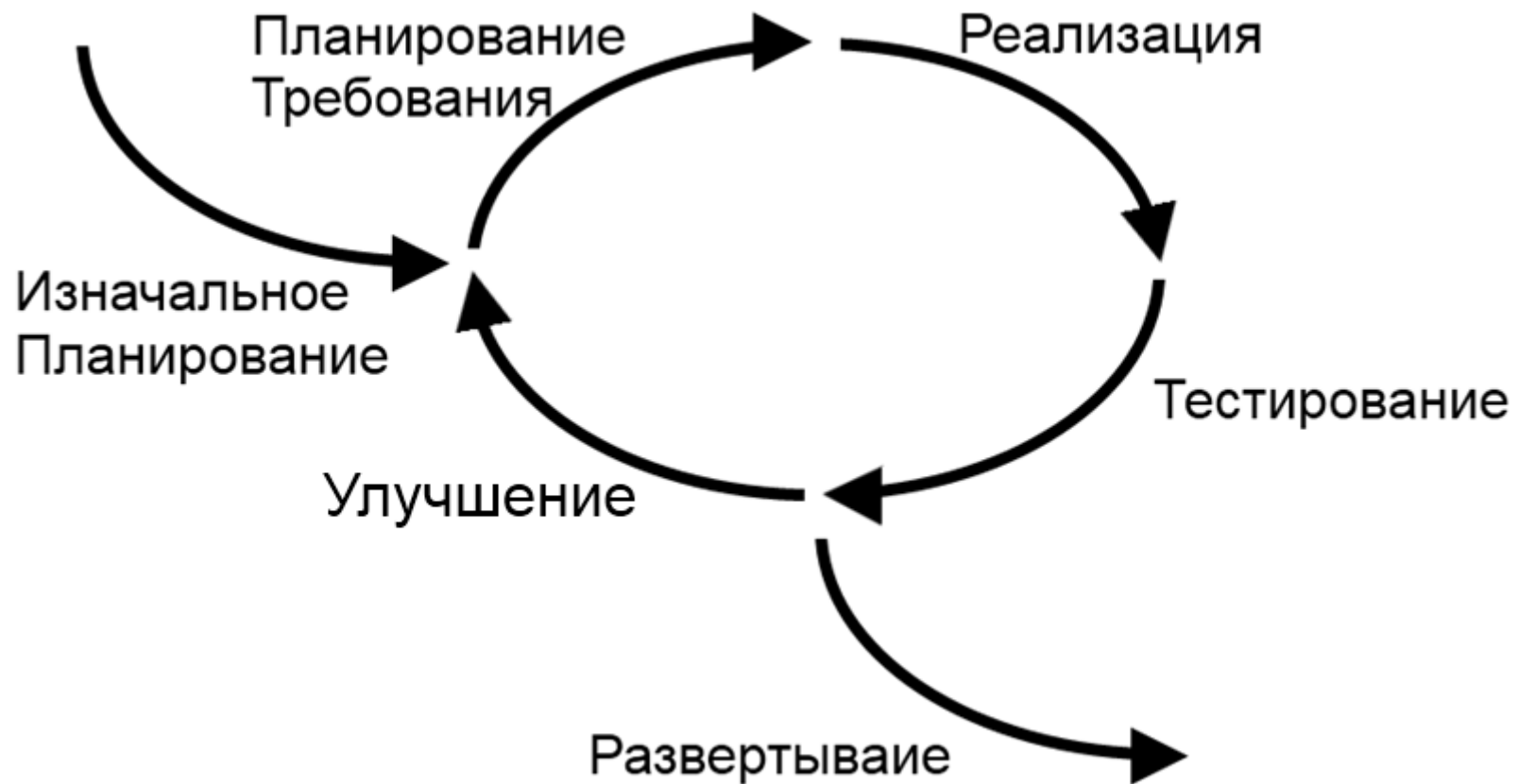


Спиральная модель (Spiral)



Итеративная модель (Iterative)

Планирование — Реализация — Проверка — Оценка
(*plan-do-check-act*)



Понятие качества продукта

Факторы качества:

- Портируемость
- Сопровождаемость
- Тестируемость

- Эффективность
- Согласованность
- Понятность, интуитивность
- Удобство использования
- Полнота
- Безопасность
- Надёжность

Стандарты качества по ISO 9000

Главное - это особая организация системы производства, которая называется "система качества".

Документированность процессов, влияющих на качество продукта.

В большинстве случаев, речь идет о всех процессах, "относящихся к производству", начиная от закупки материалов и комплектующих и заканчивая доставкой продукции потребителю

Регулярная оценка качества: внутренний контроль качества, аудит

Претензии заказчика регистрируются

Total Quality Management (TQM)

Quality Assurance (QA), Quality Improvement (QI)

Компания должна работать не только над **качеством продукции**, но и над

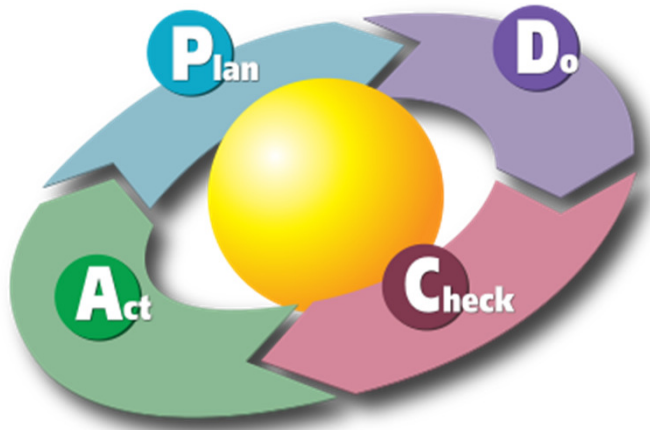
- **качеством организации процессов**
- **уровнем квалификации персонала**

Плюсы: быстрое эффективное развитие бизнеса

Показатели качества

- степень реализации требований клиентов,
- рост финансовых показателей
- повышение удовлетворенности служащих своей работой

Цикл качества Деминга



“Quality is not an act,
it is a habit.”

W. Edwards Deming

Plan: установка целей, **планирование** процессов и работ для достижения целей и удовлетворения потребителя, распределение ресурсов

Do: **выполнение** запланированных работ

Check: **проверка** результата на основе ключевых показателей эффективности (KPI), выявление и анализ отклонений

Act: **корректировка** в планировании, распределении ресурсов и др. для устранения причин отклонений от запланированного результата

Понятие Программы

Программа – набор инструкций и данных, исполнимых компьютером

Тогда, **Дом** – набор стройматериалов, собранных согласно чертежам

Дом – постройка, предназначенная для жилья людей
В таком определении есть цель и заинтересованные лица (**stakeholder**)

Заинтересованное лицо (**stakeholder**) - любой человек, на которого влияет:

- Успех или неудача проекта создания программы
- Действия или бездействие программы

Программа –

- Общение
- между людьми или компьютерами
- Разнесенными во времени и пространстве
- Содержащее инструкции, которые могут быть исполнены компьютером

Дефект

Дефект - атрибут программы, который:

- Уменьшает ценность для «важных» заинтересованных лиц
- Увеличивает ценность для «враждебных» заинтересованных лиц
- Без соответствующей компенсирующей пользы

Понятие Дефекта

Дефект – отличие поведения программы от

- Спецификаций продукта:

Калькулятор считает, что $2 \times 2 = 5$

- Ожиданий пользователя, не определенных спецификацией продукта:

Невозможность вывести документ на печать в программе обработки документов

Дефект или нет?

- FAT32 не может работать с файлами размером больше 4ГБ
- Не можем перекодировать фильм за долю секунды
- Программа не удаляет за собой временные файлы
- «Стандартный» калькулятор не умеет считать синусы

Цена Дефекта ПО

Чем раньше найден дефект – тем меньше цена его исправления

Дефект стоит денег:

- Возможное уменьшение продаж продукта
- Возможное возмещение убытков покупателям
- Увеличение расходов на службу поддержки
- Судебные издержки
- Потеря имиджа компании

Дефект стоит времени:

- Поиск причин проблемы
- Исправление кода
- Повторное тестирование продукта

Примеры «Дорогих» Ошибок

- Взрыв на советском газопроводе в результате диверсии ЦРУ (1982)
- Ошибка в управляющем коде рентгеновского аппарата «Therac-25» привела к смерти 5 пациентов (1980)
- «Пинг смерти». Неверно сформированный пакет пинга приводил к «падению» операционной системы (1995/96)
- Веерное отключение электричества в Северной Америке из-за ошибки в управляющем ПО (2003)
- Использование целых чисел в расчетах механизма детонации ракет «Патриот»
- Все ключи сгенерированные версиями OpenSSL, выпущенными 2006-2008, уязвимы из-за ошибки в генераторе случайных чисел
- Зависание бортовой аппаратуры израильских самолетов при полетах над Мертвым Морем
- Патч онлайн игры «Eve Online» удалил файл boot.ini на компьютерах всех пользователей (2007)

Зачем тестировать?

- Демонстрация отсутствия ошибок в программе?
- Демонстрация корректности исполнения программой предусмотренных сценариев?
- Убедиться, что программа выполняет свое предназначение?
- Убедиться, что у программы нет «побочных эффектов»?
- Понять ценность программы для потребителя?
- ...

Что такое хороший тест:

Тест, подтверждающий отсутствие ошибки?

Тест, находящий ошибку?

Понятие Тестирования

Тестирование это:

- «Процесс поиска всех мыслимых ошибок или недостатков в рабочем продукте» [Майерс, 1979]
- Проверка работоспособности продукта
- Оценка качества продукта
- Информирование о качестве продукта для управления рисками

Тестирование – **повторяемый, систематизированный, документированный** процесс исследования программы с целью нахождения в ней угроз качеству продукта

«Тестирование может быть использовано для демонстрации наличия ошибок, но никогда для их отсутствия» [Дейкстра]

Примеры Целей Тестирования

Обнаружение ошибок в программе

Локализация и диагностика обнаруженных искажений результатов

Тестирование правильности сделанных **корректировок (устранения ошибок)**

Артефакты. Верификация и Валидация

Артефакт жизненного цикла ПО

Элемент информации, используемый или порождаемый в процессе разработки ПО

Верификация

Соответствие артефактов между собой, например, код правильно реализует архитектуру

Валидация

Соответствие артефактов нуждам потребителя

Полное Тестирование

Возможно ли обнаружить все ошибки в программе?

Пример: проверим перемножение 2 целых чисел

Число комбинаций - $2^{32} * 2^{32}$

Если один тест идет 0.000001 сек, то все тесты идут 213503982 дня

Невозможно создать набор тестов гарантирующий отсутствие ошибок

Экономическая целесообразность создания полного набора тестов



**В общем случае невозможно найти все ошибки в программе.
Но это не означает, что к этому не нужно стремиться.**

Принципы Тестирования (Майерс)

1. Необходимая часть тестового сценария – определение ожидаемого результата
2. Вдумчиво изучайте результаты каждого теста
3. Програмист (организация!) должен избегать тестирования собственных программ
4. Тестовые сценарии должны разрабатываться для некорректных входных данных, так же как и для правильных и ожидаемых
5. Определение что программа делает то что должна – лишь половина дела. Другая половина – проверка что программа не делает того чего не должна
6. Избегайте бездумного исключения тестовых сценариев
7. Не планируйте тесты в предположении, что ошибки не будут найдены
8. Вероятность нахождения ошибок в секции программы прямо пропорциональна количеству уже найденных там ошибок
9. Тестирование – исключительно творческая и интеллектуальная задача

Мифы О Тестировании

- Цель тестирования – демонстрация отсутствия ошибок в продукте
- Есть код, который нет нужды тестировать
- Тестирование – это просто. Все способны тестировать
- Тестирование – случайный, не систематизируемый процесс
- Тестированию не нужно учиться
- Тестирование = отсутствие карьерного роста
- Продукт может быть протестирован полностью
- Тестирование позволяет найти все дефекты продукта
- Все дефекты должны быть исправлены
- Автоматизированный тест эквивалентен аналогичному выполненному человеком

Обзор Типов Тестирования

По области:

Инсталляционный. Функциональный. Документационный.

По нагрузке:

Smoke. Бизнес-цикл. Стресс. Объем. Ресурс

Регрессия:

Регрессионный. Нерегрессионный

По стадии:

Прототип. Unit. Компонент. Интеграция компонентов.
Системный.

По требованиям:

Стабильность. Правильность данных. Законность.
Производительность. Безопасность. Доступность.
Восстановление после сбоя. Удобство. Конфигурация.
Совместимость.

Тест представляет из себя комбинацию типов.

Обзор Типов Тестирования

По области:

Инсталляционный. **Функциональный**. Документационный.

По нагрузке:

Smoke. **Бизнес-цикл**. Стресс. Объем. Ресурс

Регрессия:

Регрессионный. Нерегрессионный

По стадии:

Прототип. Unit. Компонент. Интеграция компонентов.

Системный.

По требованиям:

Стабильность. Правильность данных. Законность.

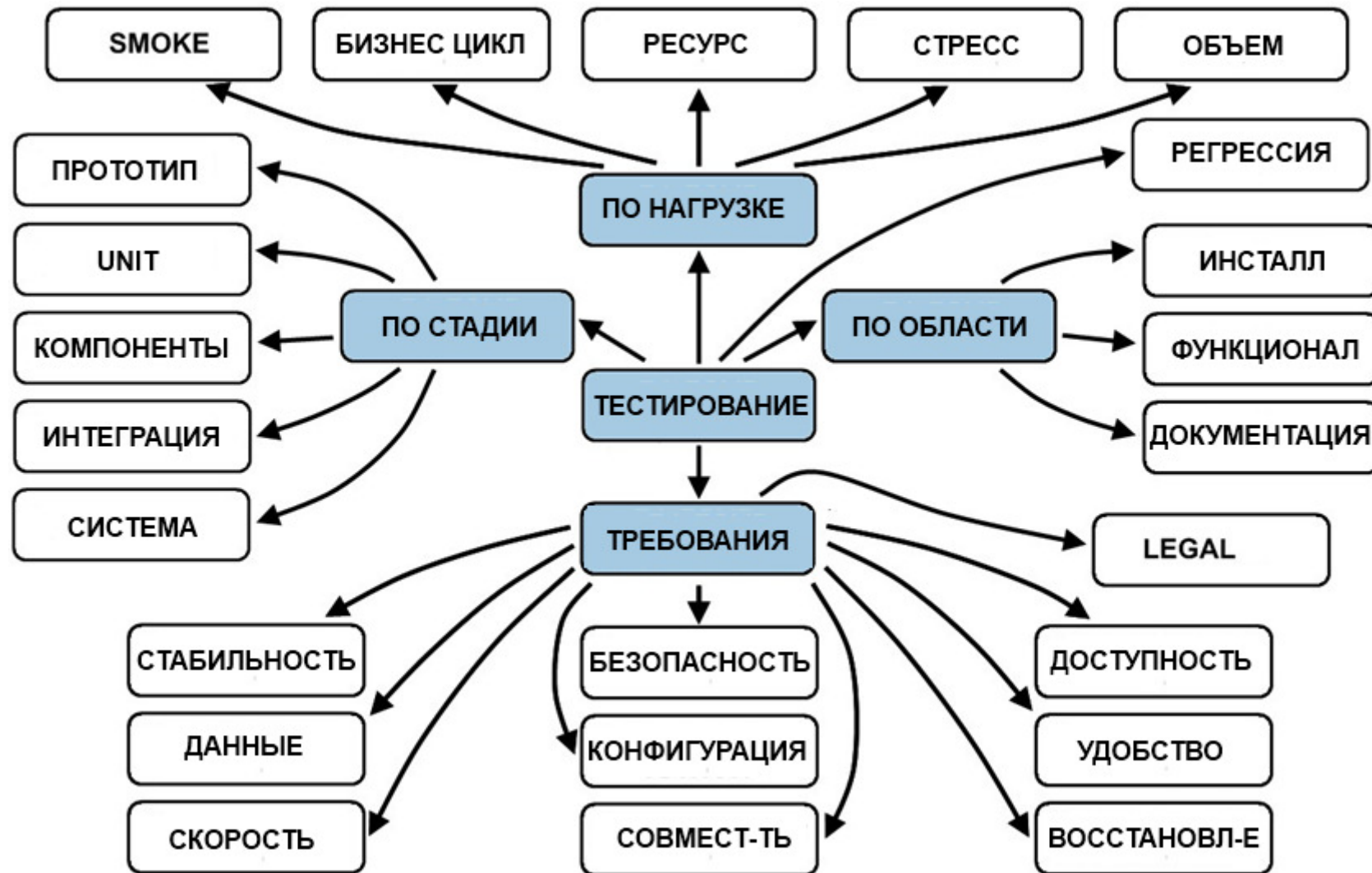
Производительность. Безопасность. Доступность.

Восстановление после сбоя. Удобство. Конфигурация.

Совместимость.

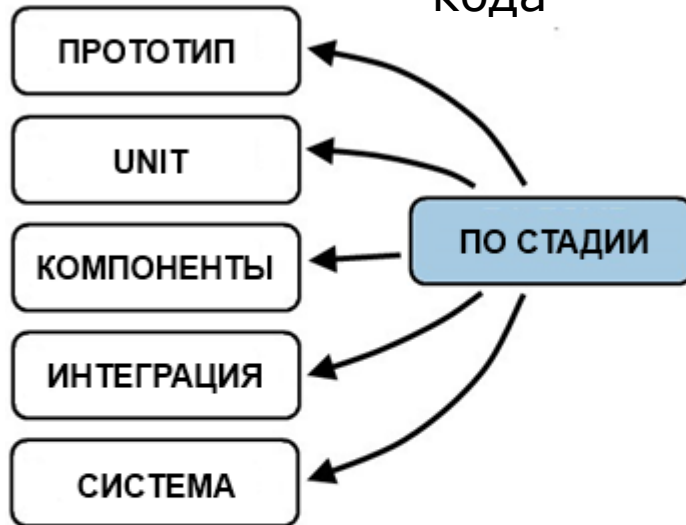
Функциональный регрессионный бизнес-цикл тест системы на соответствие требованиям стабильности

Обзор Типов Тестирования



Обзор Типов Тестирования

Прототип, концепт: поиск изъянов дизайна, которые могут быть обнаружены до написания кода



Unit: тестирование части кода продукта (например, вызовы функции с различными параметрами)

Компоненты: протестировали код – протестируем компоненту, скомпилированную из него

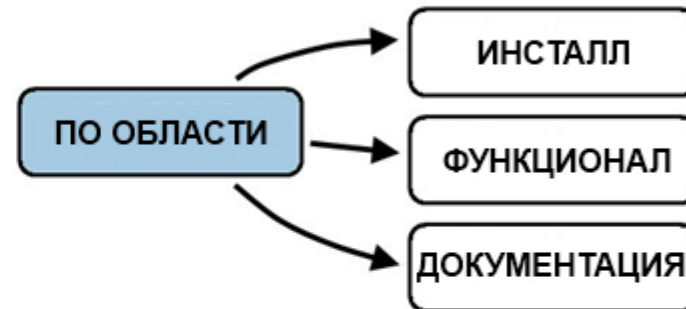
Интеграция: (восх., нисх.) тестирование нескольких компонент с целью поиска ошибок в их взаимодействии

Система: тестирование всего продукта, где проверяется работа в целом

Обзор Типов Тестирования

Функциональное тестирование: покрытие тестами функциональности продукта

Тестирование инсталляции:
Установка, активация удаление,
модификация продукта



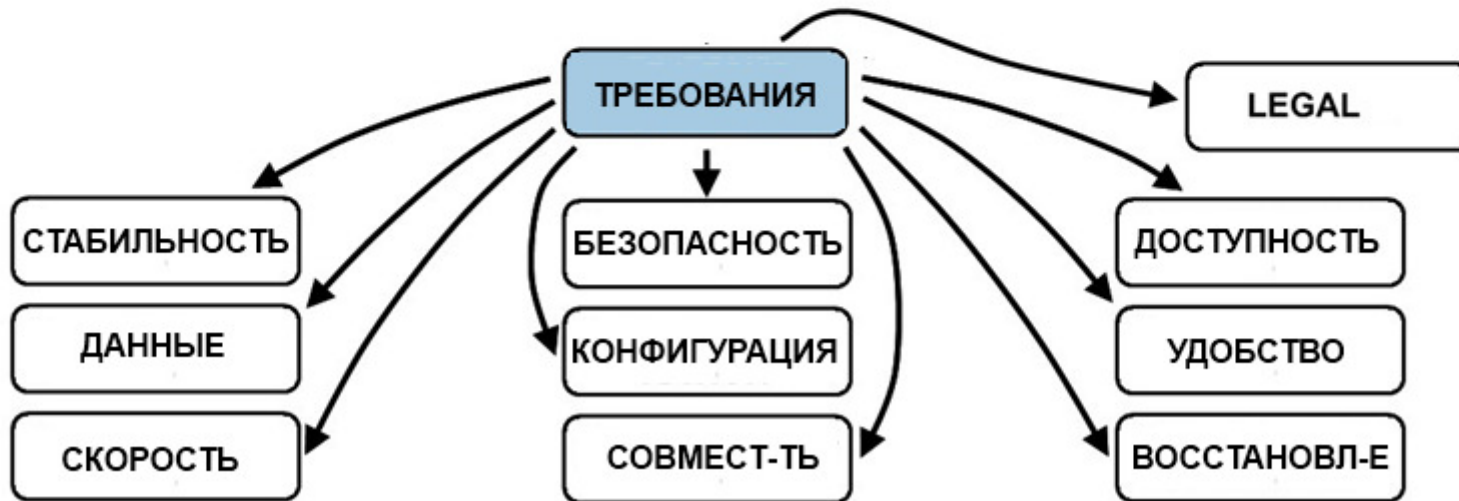
Тестирование документации: Проверка полноты документации, целостности и правильности представленной информации

Обзор Типов Тестирования

Стабильность: тестирование на предмет падений, зависаний, ошибочных сообщений пользователю. Задекларированная функциональность должна работать

Правильность данных: продукт должен возвращать *корректные* данные

Скорость: тестирование скорости работы продукта на соответствие заявленным целям

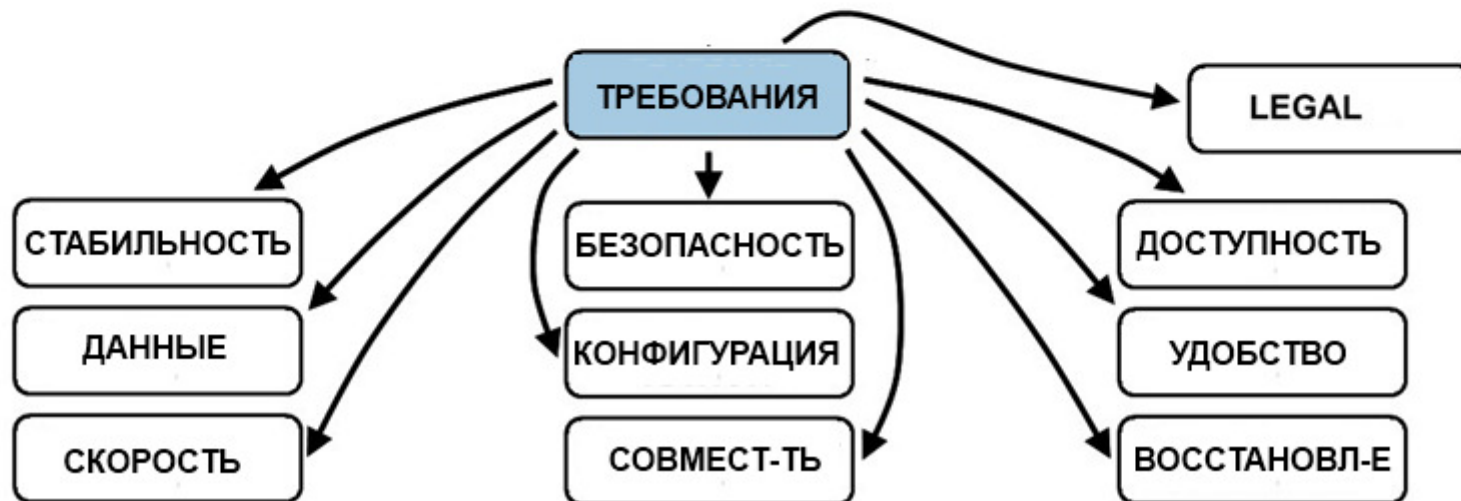


Обзор Типов Тестирования

Конфигурационное тестирование: проверка работы при различных конфигурациях системы (архитектура, железо, драйвера, операционные системы)

Совместимость: проверка сосуществования и корректности взаимодействия продукта с другими продуктами

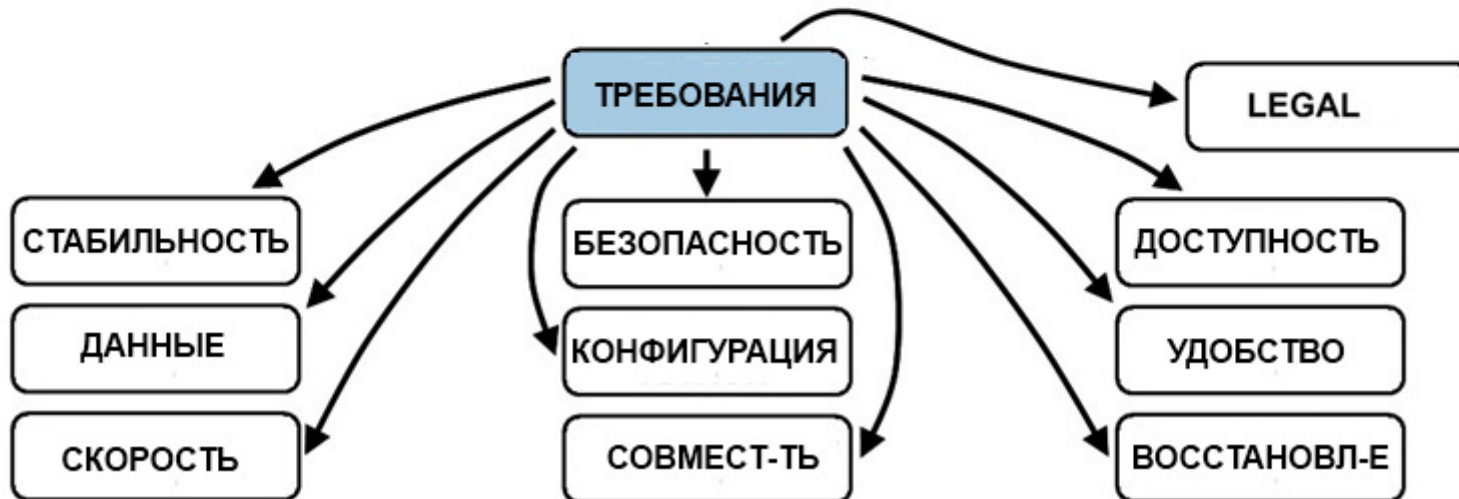
Безопасность: отсутствие вредоносного кода, защита персональных данных, поиск уязвимостей



Обзор Типов Тестирования

Законность (Legal): проверка того, что продукт не содержит вещей, конфликующих с законодательством и не противоречит законным интересам производителя

Устойчивость к сбоям и восстановление при ошибках: отключение питания, сбой сетевого соединения, остановка сервиса или процесса должны вызывать минимальные последствия



Обзор Типов Тестирования

Удобство использования: работа с продуктом должна быть интуитивной, интерфейсы – единообразными

Доступность: насколько использование продукта может быть доступно людям с ограниченными физическими возможностями



Обзор Типов Тестирования



Нагрузочное тестирование: поиск ошибок в системе при различных запросах

Smoke: легкая нагрузка с целью поиска простых но серьезных дефектов

Бизнес-цикл: оценка способности продукта отвечать требованиям реальных пользователей. Моделирование нагрузки, ожидаемой при обычном использовании продукта

Обзор Типов Тестирования



Ресурсное тестирование: продукт должен быть стабилен на протяжении определенного временного периода при заданной нагрузке

Стресс тестирование: оценка способности продукта работать в более тяжелых условиях, чем рассчитано

Объемное тестирование: оценка способности продукта работать с большими объемами данных, например, билд большого продукта, распаковка "тяжелого" архива, сбор гигабайт данных ...

Домашнее Задание

Разработать тесты для программы по курсу Многопоточность C++.

Хотя бы один тест на одну область. Покрыть:

- Минимум 8 областей на соответствие требованиям (стабильность, данные ...)
- Минимум 4 нагрузочных области (бизнес-цикл, стресс ...)

Всего 12 областей. Должно получиться от 8 до 12 тестов. В ответе указать:

Область – Тест

или

Область 1, Область 2 – Тест

Материалы и источники

1. The Art of Software Testing. Glenford J. Myers
2. Methods of software Quality Improvement. Amir Sejminovic, Farshid Atachi
3. Блог Майкла Болтона (<http://www.developsense.com/blog>)
4. Блог «255» ступеней (<http://www.developsense.com/blog>)
5. Википедия:
 1. http://en.wikipedia.org/wiki/List_of_software_bugs
 2. http://en.wikiversity.org/wiki/Software_testing/history_of_testing
6. <http://www.exforsys.com/tutorials/testing/software-testing-myths.html>
7. TestingGeek <http://www.testinggeek.com/what-is-not-software-testing-exploring-myths>
8. Alan Myrvold blog <http://testapprentice.com/2011/02/21/software-testing-myths/>
9. <http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all>