1. Что является объектом тестирования?  
   Разработанная система (или подсистема/компонент, в случае модульного тестирования)
2. Чем отличается коробочный продукт от заказного продукта?  
   Коробочный продукт - пишется на основе общих требований к продуктам такого класса, возможно - с учетом какого-то сегмента потребителей. Коробочный продукт потребляют много потребителей.  
   Заказной продукт - пишется с учетом нюансов и бизнес-процессов конкретной компании. Один потребитель.
3. Перечислите основные этапы процесса разработки ПП. Какова основная задача каждого из них?  
    Планирование - выбор методологии разработки, распределение ролей, опредедение основных майлстоунов.

Сбор и анализ требований. Требование - задокументированные желания заказчика. Задача - выявление не точностей, уточнение сложных моментов. Ответсвенный - аналитик.

Проектирование архитектуры - определение основных компонентов системы и процесса их взаимодействия. Ответственный - Архитектор

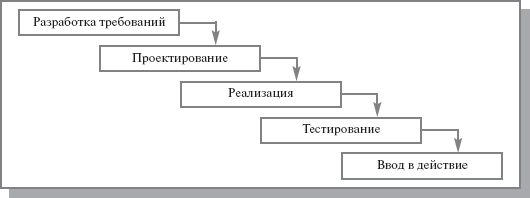
Кодирование. Ответственный - программист

Тестирование - выявление несоостветсвия ПП требованиям. Отвественный - тестировщик.

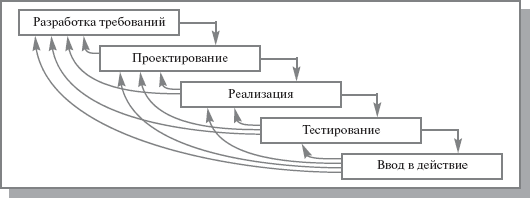
Документирование. Ответсвенный - технический писатель

Внедрение и сопровождение

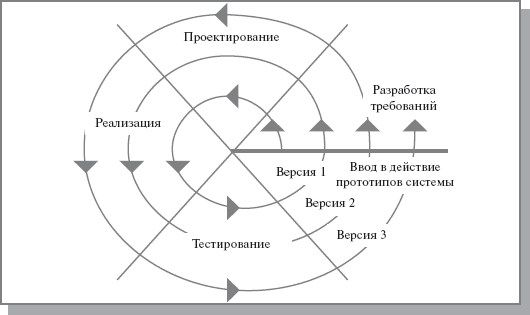
1. С какими процессами взаимодействует процесс тестирования?  
   Взаимодействует с процессом разработки (разработчики исправляют ошибки) и с процессом выдачи продукта заказчику (отдавать заказчику можно только качественный продукт)
2. Что такое проект? Перечислите основные роли в проекте по разработке ПО?  
   Проект - разработка уникального продукта с помощью ограниченных ресурсов и указанных требований к качеству.
3. Что такое жизненный цикл ПП?  
   Жизенный цикл = зарождение + процесс + смерть проекта, уход с рынка
4. Какие модели жизненного цикла ПП Вы можете назвать (дайте краткую характеристику каждой модели, когда ее можно применять, когда не следует)?  
    Waterfall



Итеративная модель



Данная модель является почти эквивалентной по алгоритму предыдущей модели, однако при этом имеет обратные связи с каждым этапом жизненного цикла, при этом порождает очень весомый недостаток: *10-ти кратное увеличение затрат на разработку*.   
 Спиральная модель (в отличие от итерационной - здесь присутствует этап прототипирования для большего учета рисков)



Преимущества:

* Быстрое получение результата
* Повышение конкурентоспособности
* Изменяющиеся требования — не проблема

Недостатки:

* Отсутствие регламентации стадий

1. Какую модель ЖЦ можно применять при условии частых изменений в требованиях? Основные принципы Agile методологий?  
   Если требования часто меняются - то Agile. Agile использует итеративную модель. Основной принцип - это гибкость требований, постоянное их изменений в результате взаимодействия с заказчиком, самоорганизующиеся рабочие группы. Методологии Agile: экстремальное программирование, Test-Driven Development, Lean, Scrum, Kanban
2. Каковы преимущества и недостатки каскадной модели?  
   Преимущества - продукт будет отвечать поставленным требованиям, обладать высокой надежностью и качеством  
   Недостатки - высокие затраты, высокая формальность процесса, низкая гибкость (нельзя менять требования без отката всего проекта назад)
3. Кто определяет цели и задачи тестирования в проекте?
4. Кто формулирует требования к продукту?  
   Требования формирует аналитик в результате общения с клиетом.
5. На что влияет качество существующих процессов?  
   На количество появляющихся дефектов?

вероятно, на затраты: оптимальные бизнес-процессы экономят врямя так как нет простоев, траты времени на ненужные дела, позволяют снизить врямя на исправление дефектов

1. Что влияет на выбор модели ЖЦ проекта?  
   Модель ЖЦ влияет на порядок пунктов ЖЦ
2. Что такое «тестирование»?  
   Процесс испытания и исследования ПП для получения информации о его соответствии требованиям
3. Цели и задачи тестирования?  
   Цель - получение информации о готовности заявленной функциональности ПП. Задачи - проверка сценариев использования системы в валидных и невалидных условиях использования, которые формируются наборами входных данных и состояний, операций или пеолучаемых тестируемым функционалом данных и набором выходных данных и состояний системы.
4. В чем отличие тестирования от отладки?  
   Отладка находит причину дефекта (строчку кода); тестирование
5. Что такое Верификация (verification) и что такое Валидация (validation)?  
   Валидация - “делаем ли мы правильный продукт”. Проверка системы или компонента на соответствие заявленным требованиям со стороны заказчика.  
   Верификация - “делаем ли мы продукт правильно”  
   Проверка системы или компонента на соответствие условиям/правилам принятым в начале фазы.
6. Какие стратегии\подходы тестирования Вы знаете? Перечислите их.  
   black box (только требования)

gray box (знаем БД, структуру, немного архитектуру)

white box (полностью знаем исходный код)

1. Какие уровни тестирования Вы знаете? Перечислите их.  
    1 Модульное тестирование

2 Интеграционное тестироание - проверяется взаимодействия между модулями системы.

3 Системное тестирование - тестирование всей системы в целом. (как взаимодействует продукт с другими приложениями)

4 Приемочное (акцептанс) тестирование

Первые 3 вида проводятся в рамках разработки, а 4ое - заказчиком

1. Какие виды тестирования Вы знаете? Перечислите их.  
   Полная классификация теста скалдывается из классификации по каждой группе

Группа 1

* Статические (вычитка кода, не запуская)
* Динамические (когда хоть что-то запускаем, например, юнит тесты или программу саму)

Группа 2

* ручные
* автоматизированные

Группа 3

* Функциональные тесты (проверяем, что описанные функции работают)
* GUI (тестирование графич. интерфейса, кнопки нажимаются и тд.)
* Тесты производительности - тестирование при определенных заданных средних нагрузках
* Нагрузочные тесты - тестирование при max нагрузке
* Стрессовые тесты - тестирование при нагрузке больше, чем max (проверяем, что данные сохраняются, норм. сообщения..)
* Тесты на отказоустойчивость
* Security тесты (идентификация и аутентификация...)
* Юзабилити тесты (удобство использования)
* Тесты на совместимость (compatibility) (с разными ос, браузерами...)
* Тесты локализаций (проверка, что все на одном языке)
* Расширяемость (импортируемость)

Группа 4

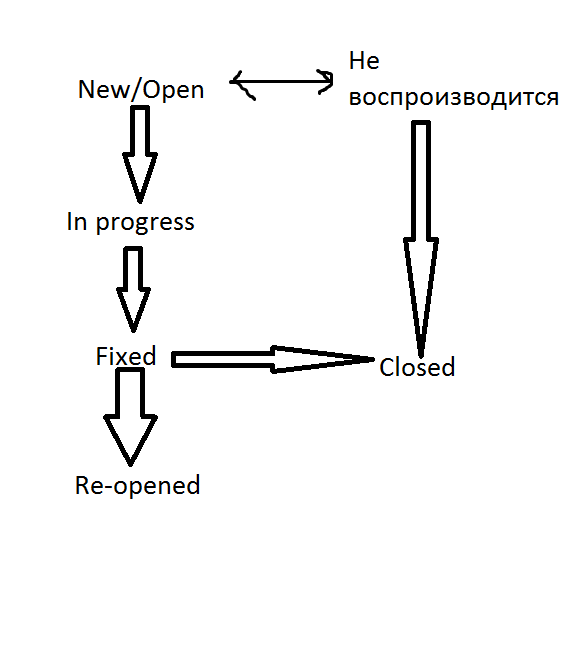
* регресионные тесты (все тесты становятся регрессиоными после того как первый раз успешно отработали) (проверка, что после внесения изменений в систему ничего не сломалось)
* smoke testing (на самые явные ошибки (выполняется прогером). если не проходит - нет смысла отдавать на более глубокое тестирование(например программа не инсталируется))

1. Перечислите основные этапы процесса тестирования.  
   1. Планирование (создается test план(цель, стратегия, роли и обязанности, приоритеты, подходы, уровни, виды, инструменты, тестовое окружение, для приемки(железо, софт, ОС, БД и тд.), план автоматизации(если есть), отчеты, метрики, критерии завершения тестирования))  
   2. Разработка текст-кейсов (цель - тестовая матрица) (+Берем у заказчика версию приемочных тестов)  
   3. Аудит тест-кейсов (скорее всего тут происходит проверка получившихся кейсов на непротиворечивось и того, что их не больше чем нужно(др тест-кейс может проверять это же))  
   4. Выполнение тест-кейсов (на определенной версии ПП, сравнения ожидаемого результата с реальным)  
   5. Верификация дефpектов (проверка, что ошибка исправлена(возможно изменяем статус тест-кейса))  
   6. Формирование отчетов
2. Что такое требования к ПО?  
   Документ, утверждающий качества и функции, которыми должно обладать ПО (т.е. формализованные ожидания заказчика)
3. Какие бывают типы требований?

* спецификация (подробная документация)
* Userstory (без технических подробностей, короткое описание)
* ТЗ

1. Что такое Test Case?  
   Test case - минимальная единица тестирования, тест, не обязательно автоматизированный, для нахождения единичной ошибки (т.е. не соотвтествия требованиям) (состоит из предусловия, действий, описания результата ожидаемого)  
   Тест-сценарий - последовательность тест-кейсов
2. Что должно быть в описании тест-кейса (приведите пример)?  
   Pre-condition, Actions, Expected results
3. Что такое Test Matrix (Тестовая матрица)?
4. Что является источником\основанием для создания тест-кейсов?  
   Требования
5. Какие техники создания тест-кейсов Вы знаете? Перечислите и опишите основную идею каждой из них.  
   1 разбиение на классы эквивалентности (анализ вхдных и выходныз данных, составляем 1 правильный для каждого коррктного значения и неправильные для каждого неккоректного значения)  
   2 анализ граничных решений (идентифицируем граничные значения у каждого входного значения: на границе, меньше, больше)  
   3 таблицы решений (определяем список возможных условий(в строки), определяем список результатов(ожидаемых действий системы)(в строки), определяем все комбинации, пишем тест-кейсы как минимум 1 для каждого стобца(если интервал то для верхней и нижней точки по тест-кейсу))  
   4 метод функциональных диаграмм (Разбить внешние спецификации на отдельные функции, которые будут тестироваться (декомпозиция фугкциональных требований); Идентифицировать явные и неявные причины (условия на входе) и присвоить каждой из них уникальный номер; Идентифицировать явные и неявные эффекты (действия на выходе) и присвоить каждому из них уникальный номер; Перевести семантику спецификации в граф «причина-следствие» (Boolean cause-effect graphing); Добавить информацию о невозможных комбинация причин\эффектов; Построить таблицу решений (бинарные значения); Записать тест кейс для каждого столбца)  
   5 предположение об ошибке (Этот метод в значительной степени является интуитивным. ;Тест инженер использует свои знания системы и способность к интерпретации спецификации на предмет того, чтобы "предугадать" при каких входных условиях система может выдать ошибку.;Перечислить в некотором списке возможные ошибки или ситуации, в которых они могут появиться, а затем на основе этого списка написать тесты.)  
   6 Requirements-Driven Testing (Проверяем каждое требование\запрос, которое описано или озвучено; анализ требований: выявление неоднозначностей, неточностей, пропущенной информации и т.п. (можно использовать функциональные диаграмма); Отслеживаем все требования и их покрытие тестами; список требований с идентификаторами и соответствующих тестов (Requirements Tracing Matrix); Для каждого требования должны быть разработаны тесты)  
   доп см лабу 1.
6. Что такое «ошибка» (дефект, «баг»)?  
   Несоответствие реального результата ожидаемому (=описанному в требованиях)
7. Какая информация обязательно должна быть в отчете об ошибке, чтобы ее можно было воспроизвести?  
   environment description, pre-conditions, actions, expected results, actual results.

Вообще отчет об ошибке состоит из заголовка(кратко что не так), описания(предустовие, шаги, полученный результат, ожидаемый результат), номер версии(где нашли баг), тестовое окружение(ОС, браузеры, иногда тех. описание), автор(отчета), исполнитель(кому направляем), ссылка на требование, статус дефекта.

1. Приведите пример жизненного цикла ошибки.  
   
2. Что такое Тест План?  
   Тест-план - главный документ, которым руководствуется QA инженер. Состоит из:
   1. объект тестирования (тестируемый продукт)
   2. цели тестирования и скоупы работ
   3. стратегию тестирования (как будет производится тестирование, какие инструменты будут применяться)
   4. приоритеты проекта и роль тестирования в процессе
   5. процесс тестирования (когда можно начинать тестирование, как следует интерпретировать результаты и т.д.)
   6. тестовую среду
   7. собственно тестовые сценарии (test cases)
   8. для приемки(железо, софт, ОС, БД и тд.)
   9. план автоматизации(если есть),
   10. отчеты, метрики
   11. критерии завершения тестирования
3. Что такое «Качество программного продукта»?  
   Качество ПО = соответствие ПП требованиям заказчика
4. Как можно описать качество ПП?  
   Качество ПП можно описать (количественно) с помощью атрибутов качества и количества дефектов (критических и некритических)
5. Назовите характеристики (атрибуты) ПП?
   1. Отказоустойчивость
   2. Удобство пользования (~соответствие макету)
   3. Функциональность
   4. Безопасность
   5. Надежность
   6. Производительность (время отклика при конкретном железе)
   7. Переносимость (железо)
   8. Совместимость (софт)
6. Что такое «внешнее качество»?  
   Внешнее качество - оценивающееся при эксплуатации ПП. Это то, что видит заказчик, с чем он будет работать (в том числе интерфейсы вплодь до цветов)
7. Что такое «внутреннее качество»?  
   Внутреннее качество - к программному коду и требованиям. Отвечает за легкость поддержания, ресширяемости и тд.
8. Что такое Quality Assurance (Обеспечение качества)?  
   Quality Assurance - применяется к процесса; осуществляет превентивные меры для уменьшения появляющихся дефектов
9. Каковы основные задачи, цели и Quality Control?  
   QC - применяется к продукту. Цель - контроль качества ПП.
10. К чему относится тестирование: Quality Assurance или Quality Tracking?  
    Quality Tracking - тестирование информирует о найденных дефектах.
11. Кто отвечает за качество продукта в проекте?  
    отвечает вся команда. Но за Project Manager последнее решение. Он берет на себя риски.
12. Что такое «стоимость качества»?  
    стоимость поиска, исправления и предотвращения дефектов
13. Какие системы управления качеством вы знаете?  
    Excel
14. Что такое «Комплексная модель производительности и зрелости процессов» (Capability Maturity Model Integration)?  
    <http://megamozg.ru/post/1098/>
15. Что такое автоматизация тестирования? Каковы основные этапы процесса автоматизации тестирования?  
    Автоматизация тестирования - написание скриптов для проверки тест-кейсов. Применяется к системному уровню.  
    Автоматизвция увеличивает стоимость, но уменьшает время.  
    1. Планирование (составляется тест-план для автоматизации)  
    2. Разработка + ревью  
    3. Запуск и анализ результата  
    4. Поддержка  
    5. Отчетность  
    Занимаются: тестировщик, разработчики, автоматизатор, automation test lead