

Лекция 1

1) Понятие видов деятельности в жизненном цикле ПО

Набор действий, направленных на решение одной задачи или группы тесно связанных задач в рамках разработки и сопровождения ПО

2) Примеры видов деятельности

Анализ предметной области
Выделение и описание требований
Проектирование
Разработка кода
Тестирование
Управление конфигурациями
Развертывание.

3) Понятие артефакта

Различные информационные сущности, документы и модели, создаваемые или используемые в ходе разработки и сопровождения ПО

4) Понятие верификации

Соответствие одних создаваемых в ходе разработки и сопровождения ПО артефактов другим, ранее созданным или используемым в качестве исходных данных, а также соответствие этих артефактов и процессов их разработки правилам и стандартам

5) Понятие валидации

Соответствие любых создаваемых или используемых в ходе разработки и сопровождения ПО артефактов нуждам и потребностям пользователей и заказчиков этого ПО, с учетом законов предметной области и ограничений контекста использования ПО

6) Понятие жизненного цикла

Весь интервал времени от момента зарождения идеи о том, чтобы создать или приобрести программную систему для решения определенных задач, до момента полного прекращения использования последней ее версии

Лекция 2

1) **Укажите составляющие характеристик качества ПО**

Функциональность – способность к взаимодействию, функциональная пригодность, защищенность, точность

Надежность – зрелость, способность к восстановлению, устойчивость к отказам

Удобство использования – удобство обучения, привлекательность, удобство работы, понятность

Производительность – временная эффективность, эффективность использования ресурсов

Удобство сопровождения – удобство внесения изменения, удобство проверки, анализируемость, стабильность

Переносимость – адаптируемость, удобство замены, способность к сосуществованию, удобство установки

2) **Точность является составляющей характеристики**

Функциональность

3) **Соответствие стандартам является составляющей характеристики**

Функциональность

Надежность

Удобство использования

Производительность

Удобство сопровождения

Переносимость

4) **Установите соответствие между характеристиками качества ПО и их составляющими**

Функциональность – Защищенность

Надежность – Зрелость

Удобство использования – Понятность

Производительность – Эффективность использования ресурсов

Удобство сопровождения – Стабильность

Переносимость – Удобство установки

5) **Привлекательность является составляющей характеристики**

Удобство использования

6) **Анализируемость является составляющей характеристики**

Удобство сопровождения

7) **Установите соответствия**

Эффективность - способность решать задачи пользователей с необходимой точностью при использовании в заданном контексте

Продуктивность - способность предоставлять определенные результаты в рамках ожидаемых затрат ресурсов

Безопасность - способность обеспечивать необходимо низкий уровень риска нанесения ущерба жизни и здоровью людей, бизнесу, собственности или окружающей среде

Удовлетворение пользователей - способность приносить удовлетворение пользователям при использовании в заданном контексте

Лекция 3

1) **Укажите составляющие характеристик качества ПО**

Функциональность – способность к взаимодействию, функциональная пригодность, защищенность, точность

Надежность – зрелость, способность к восстановлению, устойчивость к отказам

Удобство использования – удобство обучения, привлекательность, удобство работы, понятность

Производительность – временная эффективность, эффективность использования ресурсов

Удобство сопровождения – удобство внесения изменения, удобство проверки, анализируемость, стабильность

Переносимость – адаптируемость, удобство замены, способность к сосуществованию, удобство установки

2) **Соответствие стандартам является составляющей характеристики**

Функциональность

Надежность

Удобство использования

Производительность

Удобство сопровождения

Переносимость

3) **Установите соответствие между характеристиками качества ПО и их составляющими**

Функциональность – Защищенность

Надежность – Зрелость

Удобство использования – Понятность

Производительность – Эффективность использования ресурсов

Удобство сопровождения – Стабильность

Переносимость – Удобство установки

4) **Установите соответствия**

Эффективность - способность решать задачи пользователей с необходимой точностью при использовании в заданном контексте

Продуктивность - способность предоставлять определенные результаты в рамках ожидаемых затрат ресурсов

Безопасность - способность обеспечивать необходимо низкий уровень риска нанесения ущерба жизни и здоровью людей, бизнесу, собственности или окружающей среде

Удовлетворение пользователей - способность приносить удовлетворение пользователям при использовании в заданном контексте

5) **Шкала, используемая для измерения реальных физических величин, называется**

Интервальной

6) **Шкала, позволяющая ранжировать некоторые характеристики программного продукта путем сравнения с опорными значениями, называется**

Порядковой

7) **Шкала, которая характеризует только наличие рассматриваемого свойства или признака у программы без учета градации по численным значениям, называется**

Категорийной

Лекция 4

- 1) **Формулирует ли проектные документы все проектные решения точно и полно проверяют**
При верификации проектных решений

- 2) **Укажите составляющие характеристик качества ПО**

Функциональность – способность к взаимодействию, функциональная пригодность, защищенность, точность

Надежность – зрелость, способность к восстановлению, устойчивость к отказам

Удобство использования – удобство обучения, привлекательность, удобство работы, понятность

Производительность – временная эффективность, эффективность использования ресурсов

Удобство сопровождения – удобство внесения изменения, удобство проверки, анализируемость, стабильность

Переносимость – адаптируемость, удобство замены, способность к сосуществованию, удобство установки

- 3) **Способна ли система работать в том окружении, в котором она нужна пользователям, проверяют**

При верификации самой работающей системы

- 4) **Соответствует ли описанное в документации поведение реальному поведению системы проверяют**

При верификации пользовательской документации

- 5) **Насколько выбранные формы организации, планы и методы выполнения работ соответствуют задачам, решаемым в рамках проекта, и ограничениями по сроками бюджету проверяют**

При верификации организационных документов

- 6) **Содержит ли документация полное, точное и непротиворечивое описание поведения системы проверяют**

При верификации пользовательской документации

- 7) **Осуществимость требований с помощью технологий, взятых на вооружение в процессе и в рамках выделенных на проект ресурсов проверяется**

При верификации описания требований

Лекция 5

- 1) **Методы изоляции ошибок относятся к**
Обеспечение устойчивости к ошибкам
- 2) **Ошибки программирования являются**
Внутренним источником угроз надежности
- 3) **Объектный код программы является**
Объектом уязвимости, влияющим на надежность ПС
- 4) **К обработке сбоев аппаратуры относится**
Предупреждение отказов питания
Повторное выполнение операций
Восстановление памяти
- 5) **Динамический вычислительный процесс обработки данных является**
Объектом уязвимости, влияющим на надежность ПС

Лекция 6

1) *Искажения в каналах телекоммуникации информации, передаваемой потребителям являются*

Внешними дестабилизирующими факторами

2) *Сбои и отказы в аппаратуре являются*

Внешним источником угроз надежности

3) *Методы, позволяющие свести сложность программы к минимуму относятся к*

Предупреждению ошибок

4) *Информация, выдаваемая потребителям*

Объектом уязвимости, влияющим на надежность ПС

5) *Методы отступления или сокращенного обслуживания относятся к*

Обеспечение устойчивости к ошибкам

Лекция 7

1) Модель переходных вероятностей

Основа на марковском процессе

2) Оценка надежности больших многомодульных программ, основанная на использовании имитационной модели, программа которой «засоряется» ошибками, выполняется по

Модели Миллса

3) Аналитическое моделирование надежности включает

4 шага

4) Проведение тестирования двумя группами программистов предусмотрено в

Простой интуитивной модели

5) Модель Шика-Волвертона относится к

Динамическим моделям дискретного времени

Лекция 8

1) **Дублирование накопленных исходных и промежуточных данных, обрабатываемых комплексом программ, образует**

Информационную избыточность

2) **Для оперативной защиты при отказовых ситуациях наибольшее внимание уделяется**

Временной избыточности

3) **Для реализации фильтров, обеспечивающих повышение надежности функционирования программ и защиту вычислительного процесса программно-алгоритмическими методами, используется**

Три вида избыточности

4) **Использование некоторой части производительности ЭВМ для контроля исполнения программ и восстановления вычислительного процесса образует**

Временную избыточность

5) **Контроль и обеспечение достоверности наиболее важных решений по управлению и обработке информации образует**

Программную избыточность

Лекция 9

1) Существует

6 видов отказовых ситуаций

2) Существует следующее количество состояний, характеризующих процесс функционирования комплекса программ на однопроцессорной ЭВМ

5

3) Версии комплекса программ можно разделить на

Эталонные и пользовательские

4) Повышения интенсивности искажений исходных данных, а также увеличение загрузки комплекса программ выше нормальной образуют группу

Форсированных методов определения показателей надежности

5) При введении избыточности, сокращение ресурсов ЭВМ, доступных для выполнения основных функций системы, характеризуется

Коэффициентом простоя

Лекция 10

1) *Ошибки разработчика на любом из этапов разработки ПО, которые могут содержаться в исходных или проектных спецификациях, текстах кодов программ, эксплуатационной документации и т.п., являются*

Дефектами.

2) *Состояние программы, при котором выдается неправильные результаты, причиной которых являются изъяны в операторах программы или в технологическом процессе ее разработки, что приводит к неправильной интерпретации исходной информации, а, следовательно, и к неверному решению, называется*

Ошибкой

3) *Тестирование минимально возможного для тестирования компонента, например, отдельного класса или функции называется*

Модульным тестированием

4) *Тестирование интегрированной системы на ее соответствие исходным требованиям называется*

Системным тестированием

5) *Тестирование интерфейсов и взаимодействий между компонентами, называется*

Интеграционным тестированием

6) *Отклонение программы от функционирования или невозможность программы выполнять функции, определенные требованиями и ограничениями, рассматриваемыми как событие, способствующее переходу программы в неработоспособное состояние называются*

Отказами

Лекция 11

1) *Если разработчик теста имеет доступ к исходному коду программ и может писать код, который связан с библиотеками тестируемого ПО, то такое тестирование называется*

Тестирование «белого ящика»

2) *Тестирование, выполняемое после внесения изменений в очередную версию программы, называется*

Регрессионным тестированием

3) *Тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям называется*

Системным тестированием

4) *Если тестирующая сторона имеет доступ к ПО только через те же интерфейсы, что и заказчик или пользователь, то такое тестирование называется*

Тестированием «черного ящика»

5) *Тестирование ПО в целях проверки способности ПО в определенных условиях решать задачи, нужные пользователям называется*

Функциональным тестированием

6) *Тестирование ПО с целью оценки удобства использования продукта называется*

Среди предложенных вариантов нет верного

Лекция 12

1) **Процесс формализованного описания функциональных и нефункциональных требований называется**

Спецификацией требований

2) **Процесс проверки правильности спецификаций требований на их соответствие, непротиворечивость, полноту и выполнимость, а также на соответствие стандартам называется**

Среди предложенных вариантов нет верного

3) **Отметьте методы сбора требований**

Наблюдение за работой действующей системы с целью отделения ее проблемных свойств от тех, которые обусловлены структурой кадров

4) **Отвечьте нефункциональные требования к системе**

Несанкционированный доступ к системе

Лекция 13

1) Провести классификацию

В разделе «Что надо тестировать?» приводятся	Описания объекта тестирования
В разделе «Что будет тестироваться?» приводятся	Описание тестируемой системы Описание компонент тестируемой системы в отдельности Список функций тестируемой системы
В разделе «Как будет тестироваться?» приводятся	Описания видов тестирования и их применения по отношению к тестируемому объекту
В разделе «Когда будет тестироваться?» приводятся	Описание последовательности проведения и перечень работ по тестированию

5) Критерии начала тестирования (все, кроме выдержки)

Готовность тестовой платформы
Наличие всей необходимой документации
Законченность разработки требуемого функционала

6) Критерии окончания тестирования (две выдержки)

Выдержан определенный период без изменения исходного кода приложения
Выдержан определенный период без открытия новых багов.

Лекция 14

1) *Покрытие решений*

303 211
5010 311

2) *Покрытие операторов*

205

3) *Провести классификацию*

К стратегиям «белого ящика» относятся

Покрытие операторов
Покрытие решений
Покрытие условий
Покрытие решений/условий
Комбинаторное покрытие условий

К стратегиям «черного ящика» относятся

Эквивалентное разбиение
Анализ граничных значений
Анализ причин и следствий
Предугадывание ошибки
Исчерпывающее тестирование

4) *Расположите стратегии тестирования в порядке повышения качества тестирования*

Покрытие операторов
Покрытие решений
Покрытие условий
Покрытие решений/условий

5) *Покрытие условий*

204 111

Лекция 15

1) *Покрывание операторов*

Нет правильного ответа

2) *Покрывание решения*

303 211

5010 311

3) *Покрывание условий*

204 111

4) *Соедините*

$A = 2, B = 0, X = 4$ - 1, 5

$A = 2, B = 1, X = 1$ - 2, 6

$A = 1, B = 0, X = 2$ - 3, 7

$A = 1, B = 1, X = 1$ - 4, 8

Лекция 16

1) **Важности дефекта**

Блокирующий
Критический
Значительный
Незначительный
Тривиальный

2) **Приоритеты дефекта**

Высокий
Средний
Низкий

3) **Градации важности дефекта**

Блокирующая ошибка – ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работы с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы

Критическая ошибка – неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, пришедшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

Значительная ошибка – часть основной бизнес логики работает некорректно. Ошибка некритична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

Незначительная ошибка – ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

Тривиальная ошибка – ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная по средствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

4) **Градация приоритета дефекта**

Высокий – ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

Средний – ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

Низкий - ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.