

1. Какой из перечисленных этапов не является частью процесса создания программного обеспечения?

=====

#

Обслуживание

=====

Анализ требований

=====

Проектирование

=====

Разработка

+++++

2. Какой этап разработки программного обеспечения включает определение целей программного продукта и изучение требований пользователей?

=====

#

Анализ требований

=====

Тестирование

=====

Внедрение

=====

Оптимизация

+++++

3. Какой из этапов разработки программного обеспечения предполагает создание детального плана архитектуры и дизайна программы?

=====

#

Проектирование

=====

Разработка

=====

Тестирование

=====

Внедрение

++++

4. Какой этап разработки программного обеспечения включает написание и отладку исходного кода программного продукта?

=====

#

Разработка

=====

Обслуживание

=====

Тестирование

=====

Внедрение

++++

5. Какой этап разработки программного обеспечения включает проверку программного продукта на соответствие требованиям и исправление обнаруженных ошибок?

=====

#

Тестирование

=====

Проектирование

=====

Обслуживание

=====

Анализ требований

++++

6. Какой из этих этапов разработки программного обеспечения включает установку программного продукта на компьютеры конечных пользователей?

=====

#

Внедрение

=====

Проектирование

=====

Разработка

=====

Обслуживание

+++++

7. Что такое качество программного обеспечения?

=====

#

Совокупность свойств и характеристик программы, отвечающих требованиям заказчика и пользователя

=====

Количество строк кода в программе

=====

Количество функций в программе

=====

Степень сложности алгоритмов в программе

+++++

8. Какой из нижеперечисленных факторов не относится к атрибутам качества программного обеспечения?

=====

#

Количество строк кода

=====

Надежность

=====

Скорость выполнения программы

=====

Эффективность использования ресурсов

+++++

9. Что подразумевается под термином "надежность" в контексте качества программного обеспечения?

=====

#

Возможность программы работать без сбоев в течение продолжительного времени

=====

Способность программы сохранять конфиденциальность данных

=====

Удобство использования интерфейса программы

=====

Скорость выполнения программы

+++++

10.Какой атрибут качества программного обеспечения обеспечивает соответствие программы спецификациям и требованиям пользователя?

=====

#

Правильность

=====

Понятность

=====

Гибкость

=====

Тестируемость

+++++

11.Какой из перечисленных атрибутов качества программного обеспечения означает удобство использования программы пользователем?

=====

#

Понятность

=====

Гибкость

=====

Надежность

=====

Тестируемость

+++++

12.Что из перечисленного не входит в этап создания программного обеспечения?

=====

#

Сопровождение

=====

Кодирование

=====

Тестирование

=====

Документирование

+++++

13. Что определяется способностью ПО решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя, при заданных условиях использования ПО?

=====

#

Функциональность

=====

Надежность

=====

Эффективность

=====

Удобство использования

+++++

14. Что обеспечивает способность ПО выполнять требуемые задачи в обозначенных условиях на протяжении заданного промежутка времени или указанное количество операций?

=====

#

Надежность

=====

Эффективность

=====

Удобство использования

=====

Функциональность

+++++

15. Возможность легкого понимания, изучения, использования и привлекательности ПО для пользователя - это

=====

#

Удобство использования

=====

Надежность

=====

Эффективность

=====

Функциональность

+++++

16.Способность ПО обеспечивать требуемый уровень производительности в соответствии с выделенными ресурсами, временем и другими обозначенными условиями - это

=====

#

Эффективность

=====

Удобство использования

=====

Надежность

=====

Функциональность

+++++

17.Что означает термин "Reliability"?

=====

#

Надежность

=====

Эффективность

=====

Удобство использования

=====

Функциональность

+++++

18.Что означает термин "Usability"?

=====

#

Удобство использования

=====

Надежность

=====

Эффективность

=====

Функциональность

+++++

19. Что означает термин "Maintainability"?

=====

#

Удобство сопровождения

=====

Надежность

=====

Эффективность

=====

Функциональность

+++++

20. Как называется процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа?

=====

#

Верификация

=====

Удобство сопровождения

=====

Надежность

=====

Валидация

+++++

21.Определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе - это

=====

#

Валидация

=====

Верификация

=====

Удобство сопровождения

=====

Надежность

+++++

22.Какой показатель качества программного обеспечения отражает соответствие программы спецификациям и требованиям пользователя?

=====

#

Правильность

=====

Эффективность

=====

Удобство использования

=====

Надежность

+++++

23.Какой из перечисленных факторов обычно ведет к ухудшению качества программного обеспечения?

=====

#

Отсутствие документации

=====

Тщательное планирование разработки

=====

Профессиональное обучение разработчиков

=====

Использование современных инструментов разработки

++++

24. Какие свойства качества программного обеспечения относятся к внутренним характеристикам?

=====

#

Тестируемость, модульность

=====

Надежность, эффективность

=====

Понятность, удобство использования

=====

Гибкость, масштабируемость

++++

25. Какие из перечисленных характеристик относятся к внешним свойствам качества программного обеспечения?

=====

#

Понятность, гибкость

=====

Надежность, модульность

=====

Эффективность, масштабируемость

=====

Тестируемость, эффективность использования

++++

26. Что представляет собой процесс измерения качества программного обеспечения?

=====

#

Количественное определение характеристик и параметров программы

=====

Оценка субъективного восприятия пользователей

=====

Предоставление оценки экспертов в области программирования

=====

Анализ структуры исходного кода программы

++++

27.Найдите неверный ответ: Характеристики и атрибуты качества ПО с позиции возможности и точности их измерения можно разделить на

=====

#

Табличные

=====

Категорийные-описательные

=====

Количественные

=====

Качественные

++++

28.Укажите типы мер и шкал оценивания:

=====

#

Все ответы верны

=====

Номинальная шкала

=====

Порядковая шкала

=====

Интервальная шкала

++++

29.Какой термин определяется как: «Совокупность свойств продукции, обуславливающие ее пригодность удовлетворять определенные потребности в соответствии с назначением»?

=====

#

Качество

=====

Категория

=====

Свойство

=====

Функциональность

+++++

30.Какой термин определяется как: «Свойство, определяемое эффектом, который получает потребитель от использования, эксплуатации или потребления продукции»?

=====

#

Полезность

=====

Категория

=====

Конкурентоспособность

=====

Удовлетворенность

+++++

31.Какая оценка обыкновенно осуществляется в строках кода?

=====

#

Размер программы

=====

Сложность программы

=====

Надежность программы

=====

Отказоустойчивость программы

+++++

32.Какая характеристика включает в себя анализ частоты и характера возникающих ошибок, а также время между сбоями?

=====

#

Надежность и отказоустойчивость

=====

Размер программы

=====

Производительность и эффективность

=====

Сложность программы

+++++

33.Какая характеристика может проводиться с использованием таких метрик, как время выполнения, использование ресурсов (например, CPU, память), пропускная способность и т. д.?

=====

#

Производительность и эффективность

=====

Надежность и отказоустойчивость

=====

Размер программы

=====

Сложность программы

+++++

34.Программометрия - это ...

=====

#

Измерение различных характеристик программного обеспечения с целью оценки их качества, производительности и сложности

=====

Методика изучения атмосферных явлений с помощью компьютерных программ

=====

Это научная дисциплина, изучающая метеорологические показатели с помощью специальных программных алгоритмов

=====

Программа для анализа данных, обеспечивающая комплексный подход к оценке надежности и эффективности программного обеспечения

+++++

35.Для чего не используются метрики в программометрии?

=====

#

Определения языка программирования

=====

Измерения производительности программ

=====

Измерения надежности программ

=====

Измерения эффективности программ

+++++

36.Number of Defects - этот показатель ...

=====

#

Показатель указывает на количество обнаруженных ошибок

=====

Измеряет процент кода программы, который покрывают тесты

=====

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

=====

Показывает скорость, с которой новые дефекты обнаруживаются в программном обеспечении после его выпуска

+++++

37.Test Coverage - этот показатель

=====

#

Измеряет процент кода программы, который покрывают тесты

=====

Показатель указывает на количество обнаруженных ошибок

=====

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

=====

Показывает скорость, с которой новые дефекты обнаруживаются в программном обеспечении после его выпуска

+++++

38.Defect Resolution Time - этот показатель

=====

#

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

=====

Измеряет процент кода программы, который покрывают тесты

=====

Показатель указывает на количество обнаруженных ошибок

=====

Показывает скорость, с которой новые дефекты обнаруживаются в программном обеспечении после его выпуска

+++++

39.Defect Arrival Rate - этот показатель

=====

#

Показывает скорость, с которой новые дефекты обнаруживаются в программном обеспечении после его выпуска

=====

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

=====

Измеряет процент кода программы, который покрывают тесты

=====

Показатель указывает на количество обнаруженных ошибок

+++++

40.Pass Rate - этот показатель

=====

#

Определяет процент успешно завершенных тестов от общего числа проведенных тестов

=====

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

=====

Отражает процент проваленных тестов от общего числа проведенных тестов

=====

Указывает на среднее время, затраченное на исправление дефектов после их обнаружения

+++++

41.Failure Rate - этот показатель

=====

#

Отражает процент проваленных тестов от общего числа проведенных тестов

=====

Определяет процент успешно завершенных тестов от общего числа проведенных тестов

=====

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

=====

Указывает на среднее время, затраченное на исправление дефектов после их обнаружения

+++++

42.Defect Fix Rate - этот показатель

=====

#

Указывает на среднее время, затраченное на исправление дефектов после их обнаружения

=====

Отражает процент проваленных тестов от общего числа проведенных тестов

=====

Определяет процент успешно завершенных тестов от общего числа проведенных тестов

=====

Оценивает среднее время, требуемое для исправления обнаруженных дефектов от момента их выявления до момента исправления

+++++

43.Что такое качество программного обеспечения?

=====

#

Характеристика программного обеспечения, определяемая его способностью удовлетворять потребности пользователей

=====

Набор правил и норм для программного обеспечения

=====

Серия действий для обеспечения соответствия стандартам качества

=====

Процесс проверки программного обеспечения на ошибки

+++++

44. Что включают в себя стандарты качества?

=====

#

Набор правил и норм для программного обеспечения

=====

Серию действий для обеспечения соответствия стандартам качества

=====

Процесс проверки программного обеспечения на ошибки

=====

Характеристика программного обеспечения, определяемая его способностью удовлетворять потребности пользователей

+++++

45. Что представляет собой тестирование программного обеспечения?

=====

#

Процесс проверки программного обеспечения на ошибки

=====

Набор правил и норм для программного обеспечения

=====

Серия действий для обеспечения соответствия стандартам качества

=====

Характеристика программного обеспечения, определяемая его способностью удовлетворять потребности пользователей

+++++

46. Что такое метрики качества?

=====

#

Количественные показатели для оценки различных аспектов качества ПО

=====

Процесс оценки и управления рисками, влияющими на качество ПО

=====

Серия действий для обеспечения соответствия стандартам качества

=====

Характеристика программного обеспечения, определяемая его способностью удовлетворять потребности пользователей

+++++

47.Что включает в себя подготовка в процессе создания программного обеспечения?

=====

#

Сбор и обработку требований

=====

Получение технических заданий

=====

Написание исходного кода

=====

Проверку программы на соответствие требованиям

+++++

48.Что включает в себя проектирование в процессе создания программного обеспечения?

=====

#

Получение технических заданий

=====

Все ответы верны

=====

Написание исходного кода

=====

Проверку программы на соответствие требованиям

+++++

49.Что включает в себя поддержка в процессе создания программного обеспечения?

=====

#

Исправление выявленных ошибок, поддержка пользователей

=====

Получение технических заданий

=====

Написание исходного кода

=====

Проверку программы на соответствие требованиям

+++++

50.Что представляет собой функциональность программного обеспечения?

=====

#

Способность ПО решать задачи пользователя

=====

Способность ПО выполнять задачи в определенных условиях

=====

Способность ПО быть легко понятным и привлекательным для пользователя

=====

Способность ПО обеспечивать требуемую производительность

+++++

51.Что представляет собой надежность программного обеспечения?

=====

#

Способность ПО выполнять задачи в определенных условиях

=====

Способность ПО решать задачи пользователя

=====

Способность ПО быть легко понятным и привлекательным для пользователя

=====

Способность ПО обеспечивать требуемую производительность

+++++

52.Что представляет собой удобство использования программного обеспечения?

=====

#

Способность ПО быть легко понятным и привлекательным для пользователя

=====

Способность ПО решать задачи пользователя

=====

Способность ПО выполнять задачи в определенных условиях

=====

Способность ПО обеспечивать требуемую производительность

+++++

53.Что представляет собой эффективность программного обеспечения?

=====

#

Способность ПО обеспечивать требуемую производительность

=====

Способность ПО решать задачи пользователя

=====

Способность ПО выполнять задачи в определенных условиях

=====

Способность ПО быть легко понятным и привлекательным для пользователя

+++++

54.Что представляет собой удобство сопровождения программного обеспечения?

=====

#

Легкость анализа, тестирования и изменения ПО

=====

Способность ПО выполнять задачи в определенных условиях

=====

Способность ПО быть легко понятным и привлекательным для пользователя

=====

Способность ПО обеспечивать требуемую производительность

+++++

55. Что представляет собой портативность программного обеспечения?

=====

#

Легкость переноса ПО из одного окружения в другое

=====

Способность ПО выполнять задачи в определенных условиях

=====

Способность ПО быть легко понятным и привлекательным для пользователя

=====

Способность ПО обеспечивать требуемую производительность

++++

56. Что представляет собой верификация?

=====

#

Процесс проверки правильности выполнения заданных функций ПО

=====

Оценка системы на соответствие условиям разработки

=====

Определение соответствия ПО требованиям заказчика

=====

Процесс анализа правильности функционирования программного кода

++++

57. Что представляет собой валидация?

=====

#

Определение соответствия ПО требованиям заказчика

=====

Оценка системы на соответствие условиям разработки

=====

Процесс проверки правильности выполнения заданных функций ПО

=====

Процесс анализа правильности функционирования программного кода

++++

58. Временная сложность в худшем случае - это

=====

#

Функция от размера входа, равная максимальному количеству операций, выполненных в ходе работы алгоритма при решении задачи данного размера

=====

Функция от размера входа, равная максимальному количеству памяти, к которому было обращение при решении задач данного размера

=====

Функция от размера выхода, равная максимальному количеству операций, выполненных в ходе работы алгоритма при решении задачи данного размера

=====

Функция от размера выхода, равная максимальному количеству памяти, к которому было обращение при решении задач данного размера

+++++

59.Емкостная сложность в худшем случае - это

=====

#

Функция от размера входа, равная максимальному количеству памяти, к которому было обращение при решении задач данного размера

=====

Функция от размера входа, равная максимальному количеству операций, выполненных в ходе работы алгоритма при решении задачи данного размера

=====

Функция от размера выхода, равная максимальному количеству операций, выполненных в ходе работы алгоритма при решении задачи данного размера

=====

Функция от размера выхода, равная максимальному количеству памяти, к которому было обращение при решении задач данного размера

+++++

60.Программный анализ - это

=====

#

Процесс оценки программного продукта или процесса разработки программного обеспечения с целью обеспечения и улучшения их качества

=====

Это метод изучения лингвистических особенностей текстов на различных языках программирования

====

Это процесс анализа данных, полученных из различных программ, для создания статистических отчётов

====

Это метод исследования рынка программного обеспечения с помощью аналитических программных продуктов

++++

61.Лексический анализ программы - это

====

#

Процесс разбора программного кода на лексемы, такие как идентификаторы, ключевые слова, операторы, специальные символы и т. д

====

Процесс автоматического исправления опечаток и грамматических ошибок в исходном коде

====

Метод изучения структуры и семантики языка программирования с целью создания новых синтаксических конструкций

====

Процесс определения стиля кодирования, такого как форматирование отступов и использование комментариев

++++

62.Укажите неверный показатель, основанный на лексическом анализе программ:

====

#

Время выполнения программы

====

Количество строк кода (LOC)

====

Число идентификаторов

====

Число операторов

++++

63. Показатели Холстеда — это

=====

#

Набор метрик, предназначенных для оценки сложности программного кода на основе его характеристик

=====

Инструмент для анализа музыкальных произведений и определения стилистических особенностей композиций

=====

Система метрик, используемая для оценки финансовых показателей предприятия

=====

Методика оценки показателей качества воздуха в городской среде

++++

64. Какие из нижеперечисленных аспектов включаются в показатели Холстеда?

=====

#

Количество операторов и операндов в программе

=====

Время выполнения программы

=====

Степень качества графического дизайна программного интерфейса

=====

Уровень гибкости программного обеспечения

++++

65. Для чего обычно используются показатели Холстеда?

=====

#

Для оценки сложности программного кода и прогнозирования трудозатрат на его разработку

=====

Для оценки количества проданных единиц программного обеспечения

=====

Для определения времени работы компьютерной сети

=====

Для анализа структуры данных в базах данных

+++++

66. Основные компоненты вероятностной модели текста программы могут включать в себя:

=====

#

Все ответы верны

=====

Алфавит операторов и операндов

=====

Вероятностные распределения

=====

Статистические свойства

+++++

67. Назовите измеряемые свойства программ:

=====

#

Все ответы верны

=====

Время выполнения

=====

Память

=====

Энергопотребление

+++++

68. Как называется способность программы адаптироваться к увеличению объема данных или пользовательской нагрузке без потери производительности?

=====

#

Масштабируемость

=====

Устойчивость

====

Производительность

====

Все ответы верны

++++

69. Это измеряемая характеристика, которая определяет способность программы обрабатывать некорректные входные данные или неожиданные ситуации без аварийного завершения -

====

#

Устойчивость

====

Масштабируемость

====

Производительность

====

Все ответы верны

++++

70. Это общая измеряемая характеристика, которая объединяет время выполнения, использование ресурсов и другие параметры для оценки общей эффективности программы -

====

#

Производительность

====

Устойчивость

====

Масштабируемость

====

Все ответы верны

++++

71. Scalability - это

====

#

Масштабируемость программы

====

Устойчивость программы

====

Производительность программы

====

Нет правильного ответа

++++

72. Robustness - это

====

#

Устойчивость программы

====

Масштабируемость программы

====

Производительность программы

====

Нет правильного ответа

++++

73. Performance - это

====

#

Производительность программы

====

Устойчивость программы

====

Масштабируемость программы

====

Нет правильного ответа

++++

74. SOLID - это принципы разработки программного обеспечения. Что означает S - Single Responsibility Principle - принцип единственной ответственности:

====

#

Класс должен выполнять только одну задачу

====

Класс должен иметь только один метод.

====

Класс должен быть унаследован только от одного другого класса

====

Класс должен иметь только один атрибут

++++

75.SOLID - это принципы разработки программного обеспечения. Как можно реализовать принцип открытости/закрытости О - Open closed Principle?

====

#

Создать классы, которые можно расширить, но не изменять

====

Изменить исходный код класса после его компиляции

====

Использовать только приватные методы в классах

====

Использовать только статические методы

++++

76.SOLID - это принципы разработки программного обеспечения. L - Liskov substitution Principle - принцип подстановки Барбары Лисков. Какие условия должны выполняться для соблюдения принципа подстановки Барбары Лисков?

====

#

Производный класс должен быть полностью совместим с базовым классом

====

Производный класс должен переопределить все методы базового класса

====

Производный класс должен быть более общим, чем базовый класс

====

Производный класс должен содержать только статические методы

++++

77.SOLID - это принципы разработки программного обеспечения. I - Interface Segregation Principle - принцип разделения интерфейсов. Как можно разделить интерфейсы, чтобы следовать принципу разделения интерфейса?

=====

#

Разбить интерфейс на несколько более мелких, специализированных интерфейсов

=====

Создать один большой интерфейс, содержащий все методы

=====

Использовать только абстрактные классы вместо интерфейсов

=====

Наследовать интерфейсы от других интерфейсов

+++++

78.SOLID - это принципы разработки программного обеспечения. D - Dependency Inversion Principle - принцип инверсии зависимостей. Какие стратегии можно применить для соблюдения принципа инверсии зависимостей?

=====

#

Модули верхнего уровня не должны зависеть от модулей нижнего уровня

=====

Жестко закодировать зависимости внутри классов

=====

Использовать конкретные классы вместо абстракций

=====

Избегать использования интерфейсов

+++++

79.Что включает в себя оценка объема работ по программированию?

=====

#

Оценка времени, затрачиваемого на написание кода, тестирование, отладку и другие этапы разработки

=====

Оценка только времени, затрачиваемого на написание кода

=====

Оценка только количества строк кода

=====

Оценка только сложности задачи

+++++

80.Какие факторы могут влиять на оценку объема работ в программировании?

=====

#

Сложность задачи, доступность ресурсов, опыт команды разработчиков

=====

Только размер команды разработчиков

=====

Степень опыта разработчиков

=====

Только сроки, указанные заказчиком

+++++

81.Какой метод часто используется для оценки объема работ в программировании?

=====

#

Метод экспертной оценки

=====

Метод случайного выбора числа

=====

Метод прямого подсчета строк кода

=====

Метод произвольного выбора

+++++

82.Что включает в себя анализ технологий программирования в разработке программного обеспечения?

=====

#

Исследование различных технологий и их применимости для решения задач проекта

=====

Изучение только одной конкретной технологии

=====

Оценка требований проекта

=====

Подбор инструментов только по их популярности

+++++

83. Какие факторы обычно учитываются при анализе технологий программирования?

=====

#

Производительность, масштабируемость, поддержка

=====

Скорость работы компьютера.

=====

Стаж работы команды разработчиков

=====

Цветовая гамма пользовательского интерфейса

+++++

84. Что такое SWOT-анализ в контексте анализа технологий программирования?

=====

#

Анализ сильных и слабых сторон, возможностей и угроз выбранных технологий

=====

Методика расчета производительности серверов

=====

Методика тестирования алгоритмов шифрования

=====

Компьютерное моделирование различных архитектур ПО

+++++

85. Языки программирования включают в себя несколько базовых элементов.

Что такое Алфавит?

=====

#

Символы, комбинации которых будут формировать различные языковые конструкции

=====

Правила записи функций, команд и прочих конструкций язык

=====

Правила интерпретации и обработки конструкций

=====

Нет правильного ответа

+++++

86. Языки программирования включают в себя несколько базовых элементов.

Что такое Синтаксис?

=====

#

Правила записи функций, команд и прочих конструкций язык

=====

Символы, комбинации которых будут формировать различные языковые конструкции

=====

Правила интерпретации и обработки конструкций

=====

Нет правильного ответа

+++++

87. Языки программирования включают в себя несколько базовых элементов.

Что такое Семантика?

=====

#

Правила интерпретации и обработки конструкций

=====

Правила записи функций, команд и прочих конструкций язык

=====

Символы, комбинации которых будут формировать различные языковые конструкции

=====

Нет правильного ответа

+++++

88. Что представляет собой метрика количества ошибок в программе?

=====

#

Количество ошибок, обнаруженных в программе после её разработки

=====

Количество ошибок, связанных с пользовательским интерфейсом

=====

Сложность алгоритмов, применяемых при обработке ошибок

=====

Время, затраченное на исправление ошибок

+++++

89. Какие типы ошибок обычно учитываются в метрике количества ошибок в программе?

=====

#

Синтаксические, семантические и логические ошибки

=====

Только ошибки, связанные с производительностью программы

=====

Только ошибки, связанные с пользовательским интерфейсом

=====

Только ошибки, обнаруженные на этапе тестирования

+++++

90. Какие действия можно предпринять на основе анализа метрики количества ошибок в программе?

=====

#

Улучшение процессов разработки и тестирования для уменьшения числа ошибок

=====

Увеличение числа разработчиков для быстрого исправления ошибок

=====

Игнорирование ошибок, если они не критичны для функциональности программы

=====

Увеличение бюджета на разработку программы для устранения ошибок

++++

91. Сколько типов метрик программного обеспечения выделяют по своему назначению?

=====

#

3

=====

5

=====

4

=====

Много

++++

92. Что представляет собой метрика Джилба?

=====

#

Метрика для измерения сложности программы

=====

Метрика для оценки производительности разработчиков

=====

Метрика для оценки производительности аппаратного обеспечения

=====

Метрика для измерения надежности программы

++++

93. Что измеряет метрика Холстеда?

=====

#

Объем и сложность программного кода

=====

Сложность алгоритмов

=====

Производительность программного обеспечения

=====

Количество строк кода

++++

94. Какие два основных показателя включает в себя метрика Холстеда?

=====

#

Количество операторов и операндов

=====

Длина и объем

=====

Количество функций и классов

=====

Сложность и абстрактность

++++

95. Что такое "N1" в метрике Холстеда?

=====

#

Общее количество уникальных операторов

=====

Общее количество уникальных операндов

=====

Общее количество операторов

=====

Общее количество операндов

++++

96. Структурная сложность программ определяется:

=====

#

Все ответы верны

=====

Числом взаимодействующих компонент

=====

Числом связей между компонентами

=====

Сложностью взаимодействия компонент

++++

97. Это одна из самых распространенных метрик структурной сложности, которая оценивает количество линейно независимых путей через код программы:

=====

#

Цикломатическая сложность

=====

Глубина вложенности

=====

Количество модулей или функций

=====

Сложность управления данными

+++++

98. Эта метрика оценивает, насколько сложно управлять данными внутри программы, включая количество и структуру используемых данных:

=====

#

Сложность управления данными

=====

Цикломатическая сложность

=====

Глубина вложенности

=====

Количество модулей или функций

+++++

99. Какой уровень сложности имеет программа, которая состоит из линейного кода без условных операторов и циклов?

=====

#

Уровень 0

=====

Уровень 1

=====

Уровень 2

=====

Уровень 3

++++

100. Какой уровень сложности имеет программа, которая имеет один уровень ветвлений или циклов?

=====

#

Уровень 1

=====

Уровень 0

=====

Уровень 2

=====

Уровень 3

++++

101. Средняя структура сложности - это уровень сложности программы

=====

#

2

=====

3

=====

0

=====

4

++++

102. Экстремальная структура сложности - это уровень сложности программы

=====

#

4

=====

2

=====

3

=====

0

++++

103. Что представляет собой модель "Waterfall" в разработке программного обеспечения?

=====

#

Это модель, основанная на последовательном выполнении этапов разработки, где каждый следующий этап начинается только после завершения предыдущего

=====

Это модель, при которой разработка программного обеспечения выполняется параллельно несколькими командами

=====

Это модель, при которой разработка программного обеспечения выполняется в несколько итераций, позволяющих быстро реагировать на изменения требований

=====

Это модель, основанная на принципе пошагового развития, где разработчики поочередно добавляют новые функции в программу

++++

104. Какие этапы включает в себя модель "Waterfall"?

=====

#

Планирование, разработка, тестирование, внедрение, поддержка

=====

Сбор требований, проектирование, реализация, тестирование, сопровождение

=====

Итерации, прототипирование, интеграция, верификация, адаптация

=====

Анализ, проектирование, реализация, тестирование, развертывание

++++

105. Какой принцип лежит в основе модели "Waterfall"?

=====

#

Принцип поэтапного развития, где каждый этап строго последовательно следует за предыдущим

====

Принцип непрерывной интеграции и развертывания

====

Принцип итеративной разработки, позволяющей быстро адаптироваться к изменяющимся требованиям

====

Принцип автоматизации всех этапов разработки

++++

106. Какой из следующих методов разработки программного обеспечения является противоположностью модели "Waterfall"?

====

#

Agile (гибкие методы разработки)

====

RAD (быстрая разработка приложений)

====

Spiral (спиральная модель)

====

V-модель

++++

107. Что представляют собой гибкие методы разработки (Agile)?

====

#

Это методы разработки, которые ставят в приоритет отношения и взаимодействие с заказчиком, а также быстрые изменения и постоянную адаптацию

====

Это методы разработки, ориентированные на предсказуемое и строго управляемое выполнение планов

====

Это методы разработки, основанные на использовании заранее определенных шаблонов и стандартов

====

Это методы разработки, ориентированные на максимальное использование автоматизации и инструментов

++++

108. Какие преимущества предлагают гибкие методы разработки по сравнению с традиционными методами, такими как каскадная разработка (Waterfall)?

=====

#

Более гибкая реакция на изменяющиеся требования заказчика, улучшенное взаимодействие с заказчиком и более быстрое получение обратной связи

=====

Большая предсказуемость в планировании и выполнении проекта

=====

Меньшее количество требуемых ресурсов и более простая организация процесса разработки

=====

Ускоренное выполнение проекта за счет использования предварительно разработанных шаблонов и компонентов

++++

109. Какой из следующих принципов характеризует методологию Scrum?

=====

#

Принцип быстрого реагирования на изменения и постоянной адаптации

=====

Превосходство документации над взаимодействием с клиентом

=====

Принцип максимальной автоматизации процесса разработки

=====

Принцип последовательного выполнения этапов разработки

++++

110. Что представляет собой методология Быстрой разработки приложений (RAD)?

=====

#

Это методология, которая ставит в приоритет быстрое создание прототипов и быструю разработку приложений с последующим итеративным улучшением

=====

Это методика, основанная на долгосрочном планировании и строгом контроле изменений

=====

Это метод, который фокусируется на детальном проектировании перед началом разработки

=====

Это метод, при котором разработка программного обеспечения выполняется без участия разработчиков

+++++

111. Какой принцип лежит в основе методологии Быстрой разработки приложений (RAD)?

=====

#

Принцип быстрого создания прототипов и непрерывной разработки на основе обратной связи

=====

Принцип поэтапного развития, где каждый этап строго последовательно следует за предыдущим

=====

Принцип создания подробного плана и жесткого его соблюдения

=====

Принцип максимального использования автоматизации и инструментов

+++++

112. Какие преимущества предлагает методология Быстрой разработки приложений (RAD) по сравнению с традиционными методами разработки?

=====

#

Более гибкая реакция на изменяющиеся требования заказчика и быстрое создание рабочих прототипов

=====

Меньшее количество требуемых ресурсов и более простая организация процесса разработки

=====

Большая предсказуемость в планировании и выполнении проекта

=====

Ускоренное выполнение проекта за счет использования предварительно разработанных шаблонов и компонентов

+++++

113. Программа по метрике Маккейба должна быть представлена в виде:

=====

#

ориентированного графа

=====

древовидной структуры

=====

иерархии

=====

сетевой структуры

+++++

114. Что из нижеперечисленного является примером процедурно-ориентированной метрики сложности программного обеспечения?

=====

#

Cyclomatic Complexity

=====

LOC (Lines of Code)

=====

Halstead Complexity Measures

=====

Fan-in and Fan-out

+++++

115. Что из нижеперечисленного определяет Cyclomatic Complexity?

=====

#

Количество возможных путей выполнения программы

=====

Общее количество строк кода

=====

Сложность каждой процедуры в программе

=====

Среднее время выполнения программы

+++++

116. Какова основная цель измерения Cyclomatic Complexity?

=====

#

Определение потенциально сложных мест в программе

=====

Оценка числа строк кода в программе

=====

Измерение времени, затраченного на отладку программы

=====

Оценка производительности программы

+++++

117. Какой результат Cyclomatic Complexity указывает на простую структуру программы?

=====

#

Значение 1

=====

Значение 0

=====

Значение 10

=====

Значение 3

+++++

118. Что означает Cyclomatic Complexity, равная 0, для программы?

=====

#

Программа не содержит никаких операторов управления потоком

=====

Программа не содержит никаких ошибок

=====

Программа содержит только один оператор управления потоком

=====

Программа имеет максимально возможную сложность

+++++

119. Cyclomatic Complexity - это ...

=====

#

Количество линейно независимых маршрутов через программный код

=====

Измерения объёма с помощью подсчёта количества строк в тексте исходного кода

=====

Количество временных затрат на выполнение кода

=====

Нет правильного ответа

+++++

120. Source Lines of Code — SLOC это ...

=====

#

Измерения объёма с помощью подсчёта количества строк в тексте исходного кода

=====

Количество линейно независимых маршрутов через программный код

=====

Количество временных затрат на выполнение кода

=====

Нет правильного ответа

+++++

121. Что означает аббревиатура SLOC?

=====

#

Source Lines of Code

=====

Software Logic

=====

Source Language of Code

=====

System Level of Complexity

+++++

122. Как определяется метрика SLOC?

=====

#

Количество строк кода, не включая пустые строки и комментарии

=====

Общее количество комментариев в исходном коде

=====

Среднее время выполнения каждой строки кода

=====

Количество функций в программе

+++++

123. Какие аспекты сложности программного обеспечения не учитывает метрика SLOC?

=====

#

Сложность алгоритмов

=====

Размер команды разработчиков

=====

Объем используемой оперативной памяти

=====

Количество комментариев в коде

+++++

124. Какой недостаток может быть связан с использованием метрики SLOC?

=====

#

Может быть неадекватной для различных языков программирования

=====

Не учитывает структуру базы данных

=====

Не учитывает производительность программистов

=====

Не учитывает количество отладочных ошибок

++++

125. Графическое изображение метрики Мартина определяет:

=====

#

Все ответы верны

=====

Зону боли

=====

Зону бесполезности

=====

Главную последовательность

++++

126. Зона боли на графическом изображении метрики Мартина - это

=====

#

Устойчивые, конкретные компоненты

=====

Максимально абстрактные компоненты без входящих зависимостей

=====

Это та линия, вокруг которой желательно видеть распределение компонентов

=====

Нет правильного ответа

++++

127. Зона бесполезности на графическом изображении метрики Мартина -
это

=====

#

Максимально абстрактные компоненты без входящих зависимостей

=====

Устойчивые, конкретные компоненты

=====

Это та линия, вокруг которой желательно видеть распределение компонентов

=====

Нет правильного ответа

++++

128. Главная последовательность на графическом изображении метрики
Мартина - это

=====

#

Это та линия, вокруг которой желательно видеть распределение компонентов

=====

Максимально абстрактные компоненты без входящих зависимостей

=====

Устойчивые, конкретные компоненты

=====

Нет правильного ответа

++++

129. Метрики Чидамбера и Кемерера основаны на

=====

#

Анализе методов класса, дерева наследования

=====

Количестве линейно независимых маршрутов через программный код

=====

Измерении объёма с помощью подсчёта количества строк в тексте исходного
кода

=====

Количестве временных затрат на выполнение кода

++++

130. WMC (Weighted methods per class) - это

=====

#

Суммарная сложность всех методов класса

=====

Глубина дерева наследования

=====

Количество потомков (непосредственных)

=====

Сцепление между классами

++++

131. DIT (Depth of Inheritance tree) - это

=====

#

Глубина дерева наследования

=====

Количество потомков (непосредственных)

=====

Сцепление между классами

=====

Суммарная сложность всех методов класса

++++

132. NOC (Number of children) - это

=====

#

Количество потомков (непосредственных)

=====

Сцепление между классами

=====

Суммарная сложность всех методов класса

=====

Глубина дерева наследования

++++

133. В зависимости от назначения модели надежности делят на -

=====

#

Все ответы верны

=====

Прогнозные

=====

Оценочные

=====

Измерительные

++++

134. Какие модели используются для определения ожидаемого значения показателей надежности программного средства на этапе его проектирования?

=====

#

Прогнозные

=====

Оценочные

=====

Измерительные

=====

Математические

+++++

135. Какие модели используются для определения значений показателей надежности на основе анализа результатов тестирования программных средств?

=====

#

Оценочные

=====

Измерительные

=====

Математические

=====

Прогнозные

+++++

136. Какая модель позволяет оценить количество ошибок до начала тестирования по результатам тестирования программы двумя независимыми группами?

=====

#

Эвристическая

=====

Прогнозные

=====

Оценочные

=====

Измерительные

+++++

137. Какие модели используются для оценки надежности программного средства в фазе испытаний, на этапе сопровождения или эксплуатации?

=====

#

Измерительные

=====

Прогнозные

=====

Оценочные

=====

Эвристическая

+++++

138. Какая модель определяет надежность программного обеспечения через требования, спецификацию, реализацию, верификацию и обслуживание?

=====

#

Модель Касперсона

=====

Модель Лямпсона

=====

Модель Капуно

=====

Модель Бухана

+++++

139. Какая модель основана на идеях обработки ошибок и предполагает, что любая ошибка может быть детектирована, предотвращена или восстановлена?

=====

#

Модель Капуно

=====

Модель Касперсона

=====

Модель Лямпсона

=====

Модель Бухана

+++++

140. Какая модель представляет надежность программного обеспечения как соотношение между количеством обнаруженных и необнаруженных ошибок

=====

#

Модель Бухана

=====

Модель Капуρο

=====

Модель Касперсона

=====

Модель Лямпсона

+++++

141. Какие модели являются типом моделей надежности программного обеспечения, которые предназначены для предсказания вероятности сбоя или отказа программной системы в будущем?

=====

#

Прогнозные

=====

Эвристическая

=====

Оценочные

=====

Математические

+++++

142. Эта модель предсказывает среднее время между отказами -

=====

#

Модель времени наработки на отказ (МТТФ)

=====

Модель вероятности отказа в определенный период времени ($P\Phi(t)$)

=====

Модель моделирования отказов (Ф-модели)

=====

Нет правильного ответа

+++++

143. Сколько уровней зрелости процессов разработки программного обеспечения определяет модель зрелости программного обеспечения СММ/СММІ?

=====

#

5

=====

3

=====

10

=====

8

+++++

144. На каком уровне модели зрелости программного обеспечения СММ/СММІ процессы разработки программного обеспечения хаотичны, неорганизованны и нестабильны?

=====

#

Начальный уровень

=====

Повторяемый уровень

=====

Управляемый уровень

=====

Оптимизированный уровень

+++++

145. На каком уровне модели зрелости программного обеспечения СММ/СММІ процессы разработки программного обеспечения описаны в форме стандартов и руководств?

=====

#

Определенный уровень

=====

Начальный уровень

=====

Управляемый уровень

=====

Оптимизированный уровень

++++

146. В модели SWOT - Strengths (Сильные стороны):

=====

#

Фокусируется на внутренних позитивных аспектах программного обеспечения

=====

Описывает внутренние негативные аспекты, которые могут ограничивать успех программного обеспечения

=====

Фокусируется на внешних факторах и возможностях для роста, улучшений и развития

=====

Описывает внешние факторы или потенциальные угрозы, которые могут негативно повлиять на программное обеспечение

++++

147. В модели SWOT - Weaknesses (Слабые стороны):

=====

#

Описывает внутренние негативные аспекты, которые могут ограничивать успех программного обеспечения

=====

Фокусируется на внешних факторах и возможностях для роста, улучшений и развития

=====

Описывает внешние факторы или потенциальные угрозы, которые могут негативно повлиять на программное обеспечение

=====

Фокусируется на внутренних позитивных аспектах программного обеспечения

+++++

148. В модели SWOT - Opportunities (Возможности):

=====

#

Фокусируется на внешних факторах и возможностях для роста, улучшений и развития

=====

Описывает внешние факторы или потенциальные угрозы, которые могут негативно повлиять на программное обеспечение

=====

Фокусируется на внутренних позитивных аспектах программного обеспечения

=====

Описывает внутренние негативные аспекты, которые могут ограничивать успех программного обеспечения

+++++

149. В модели SWOT - Threats (Угрозы):

=====

#

Описывает внешние факторы, которые могут негативно повлиять на программное обеспечение

=====

Фокусируется на внутренних позитивных аспектах программного обеспечения

=====

Описывает внутренние негативные аспекты, которые могут ограничивать успех программного обеспечения

=====

Фокусируется на внешних факторах и возможностях для роста, улучшений и развития

+++++

150. Найдите неверный компонент модели надежности Елински-Морандо:

=====

#

Все ответы верны

=====

Исторические данные о сбоях

=====

Математическое моделирование

=====

Прогнозирование сбоев

+++++

151. Какая модель была разработана с учетом основных свойств компьютерных программ и использует методы теории вероятностей лишь в тех случаях, когда невозможно получение полной информации о том или ином факторе?

=====

#

Модель Нельсона

=====

Модель Касперсона

=====

Модель Лямпсона

=====

Модель Бухана

+++++

152. Какая модель основывается на предположении, что программные ошибки возникают из-за недостатков в разработке программного обеспечения, а не из-за непредвиденных событий или случайных факторов?

=====

#

Модель Мусона

=====

Модель Нельсона

=====

Модель Касперсона

=====

Модель Бухана

+++++

153. Какой стандарт качества определяет модель качества для оценки характеристик качества программного продукта, таких как функциональность, надежность, юзабилити, эффективность, безопасность, сопровождаемость и переносимость?

=====

#

ISO/IEC 25010 (новая версия ISO/IEC 25000)

=====

IEEE 829

=====

ISTQB

=====

CMMI

+++++

154. Какой стандарт качества описывает процесс и документацию тестирования ПО, включая планирование, проектирование, реализацию и отчетность?

=====

#

IEEE 829

=====

ISTQB

=====

CMMI

=====

ISO/IEC 25010 (новая версия ISO/IEC 25000)

+++++

155. Какой стандарт описывает принципы, практики и процессы, которые используются при тестировании ПО, и является основой для сертификации тестировщиков на разных уровнях?

=====

#

ISTQB

=====

CMMI

=====

ISO/IEC 25010 (новая версия ISO/IEC 25000)

=====

IEEE 829

++++

156. Укажите неверный аспект стандартов процесса:

=====

#

Внутренние характеристики продукта

=====

Описание процессов

=====

Роли и ответственности

=====

Управление изменениями и улучшениями

++++

157. Какой из перечисленных стандартов указан не верно:

=====

#

Стандарты на сопровождение

=====

Стандарты на процессы

=====

Стандарты на методы контроля, испытаний, измерений, анализа

=====

Стандарты на продукцию, услуги

++++

158. Какой стандарт имеет основной целью удовлетворение потребностей клиентов и обеспечение высокого уровня сервиса?

=====

#

Стандарт услуг

=====

Стандарт терминов

=====

Стандарты на методы контроля

=====

Стандарты на процессы

+++++

159. Почему необходимо проводить сертификацию программного обеспечения?

=====

#

Все ответы верны

=====

Убедиться в соответствии программного обеспечения стандартам и требованиям

=====

Улучшение качества программного обеспечения

=====

Соответствие регуляторным требованиям (защита данных, банковское законодательство)

+++++

160. Продолжите предложение: Целью сертификации является подготовка и принятие решения о целесообразности выдачи заявителю сертификата соответствия с учетом ...

=====

#

Все ответы верны

=====

Полноты, точности и достоверности исходного технического задания и спецификаций требований

=====

Достоверности и точности измерения и обобщения результатов
сертификационных испытаний

=====

Методологии и качества интерпретации данных об объекте испытаний

+++++

161. Найдите неверный ответ: При сертификации руководствуются
следующими документами:

=====

#

Отзывами заказчика и потребителей

=====

Действующими международными, государственными и ведомственными
стандартами

=====

Утвержденным ТЗ

=====

Программой сертификационных испытаний

+++++

162. Как называется процесс проверки ПО на соответствие требованиям?

=====

#

Тестирование ПО

=====

Верификация ПО

=====

Валидация ПО

=====

Управление рисками

+++++

163. Что включает в себя этап проектирования в разработке ПО?

=====

#

Разработка спецификаций

=====

Сбор требований

=====

Написание исходного кода

=====

Обучение пользователей

+++++

164. Что описывает характеристика функциональности ПО?

=====

#

Способность решать задачи пользователя

=====

Способность восстанавливаться после сбоев

=====

Легкость понимания и использования

=====

Переносимость в другое окружение

+++++

165. Какое определение соответствует понятию "качество ПО"?

=====

#

Способность удовлетворять потребности пользователей

=====

Процесс управления изменениями в ПО

=====

Процесс создания тестовых случаев

=====

Набор правил и рекомендаций

+++++

166. Что означает термин "удобство использования"?

=====

#

Легкость понимания и изучения ПО

=====

Легкость переноса ПО в другое окружение

=====

Легкость управления изменениями в ПО

=====

Легкость выполнения тестирования ПО

+++++

167. Что включает в себя процесс верификации?

=====

#

Проверка выполнения требований

=====

Изменение конфигурации ПО

=====

Обучение пользователей

=====

Разработка графических макетов

+++++

168. Какой процесс направлен на выявление дефектов или ошибок в ПО?

=====

#

Тестирование ПО

=====

Управление рисками

=====

Верификация ПО

=====

Сопровождение ПО

+++++

169. Что включает в себя этап подготовки в создании ПО?

=====

#

Сбор и обработка требований

=====

Написание исходного кода

=====

Проверка программного кода

=====

Установка ПО и обучение

++++

170. Какое определение соответствует понятию "эффективность ПО"?

=====

#

Способность обеспечивать требуемую производительность

=====

Способность решать задачи пользователя

=====

Легкость переноса в другое окружение

=====

Способность восстанавливаться после сбоев

++++

171. Что описывает характеристика надежности ПО?

=====

#

Способность выполнять задачи без сбоев

=====

Легкость управления изменениями

=====

Способность обеспечивать производительность

=====

Легкость понимания и изучения

++++

172. Что включает в себя этап поддержки в создании ПО?

=====

#

Внедрение и сопровождение ПО

=====

Сбор и обработка требований

=====

Разработка спецификаций

=====

Написание исходного кода

++++

173. Что описывает показатель надежности ПО?

=====

#

Способность ПО работать без сбоев

=====

Насколько удобно пользоваться ПО

=====

Как быстро ПО выполняет задачи

=====

Уровень безопасности ПО от атак

+++++

174. Какой показатель оценивает быстродействие ПО?

=====

#

Производительность ПО

=====

Надежность ПО

=====

Совместимость ПО

=====

Удобство использования

+++++

175. Что измеряет показатель сопровождаемость ПО?

=====

#

Удобство модификации ПО после выпуска

=====

Совместимость ПО с другими системами

=====

Быстродействие ПО при выполнении задач

=====

Удобство интерфейса для пользователей

+++++

176. Что включает в себя показатель безопасности ПО?

=====

#

Защищенность ПО от атак

=====

Как быстро ПО выполняет задачи

=====

Способность ПО работать без сбоев

=====

Удобство интерфейса для пользователей

+++++

177. Какой показатель оценивает совместимость ПО?

=====

#

Способность работать с другими системами

=====

Как быстро ПО выполняет задачи

=====

Удобство интерфейса для пользователей

=====

Уровень безопасности ПО от атак

+++++

178. Что может привести к низкому качеству ПО?

=====

#

Недостаточно определенные требования

=====

Высокая квалификация команды

=====

Эффективное управление проектом

=====

Современные технологии разработки

+++++

179. Что включает в себя компетентное управление проектом?

=====

#

Выполнение работ в срок и в рамках бюджета

=====

Недостаточно определенные требования

=====

Низкая квалификация команды

=====

Современные технологии разработки

+++++

180. Что является целью процесса верификации ПО?

=====

#

Убедиться в выполнении требований

=====

Изменение конфигурации ПО

=====

Обучение пользователей ПО

=====

Разработка графических макетов

+++++

181. Какую шкалу можно использовать для классификации объектов без установления порядка?

=====

#

Интервальная шкала

=====

Порядковая шкала

=====

Номинальная шкала

=====

Относительная шкала

+++++

182. Какой тип шкалы можно использовать для измерения уровня удовлетворенности?

=====

#

Ликиртовская шкала

=====

Визуально-аналоговая шкала

=====

Бинарная шкала

=====

Номинальная шкала

+++++

183. Какую шкалу можно использовать для оценки присутствия или отсутствия характеристики?

=====

#

Бинарная шкала

=====

Ликиртовская шкала

=====

Визуально-аналоговая шкала

=====

Номинальная шкала

+++++

184. Какой тип шкалы используется для оценки надежности программного обеспечения?

=====

#

Numeric Rating Scale

=====

Ликиртовская шкала

=====

Визуально-аналоговая шкала

=====

Бинарная шкала

+++++

185. Что измеряет размер программы в программометрии?

=====

#

Строки кода или функциональные точки

=====

Время выполнения программы

=====

Частота обнаружения ошибок

=====

Уровень удовлетворенности пользователей

+++++

186. Какой показатель измеряет количество обнаруженных дефектов в ПО?

=====

#

Количество ошибок в программе

=====

Уровень покрытия тестами

=====

Среднее время между сбоями

=====

Процент успешных тестов

+++++

187. Как измеряется производительность программы?

=====

#

Время выполнения программы

=====

Количество строк кода

=====

Уровень покрытия тестами

=====

Процент успешных тестов

+++++

188. Как называется показатель, оценивающий скорость обнаружения новых дефектов?

=====

#

Частота появления дефектов

=====

Уровень покрытия тестами

=====

Количество строк кода

=====

Среднее время между сбоями

+++++

189. Какой показатель указывает на процент проваленных тестов?

=====

#

Процент отказов системы

=====

Уровень покрытия тестами

=====

Количество строк кода

=====

Среднее время между сбоями

+++++

190. Что оценивает временная сложность алгоритма?

=====

#

Количество операций алгоритма

=====

Объем используемой памяти

=====

Частоту возникновения ошибок

=====

Процент покрытия тестами

+++++

191. Как определяется пространственная сложность алгоритма?

=====

#

Количество памяти алгоритма

=====

Время выполнения алгоритма

=====

Частота возникновения ошибок

=====

Процент покрытия тестами

+++++

192. Что означает $O(n)$ в теории алгоритмов?

=====

#

Линейная сложность

=====

Квадратичная сложность

=====

Константная сложность

=====

Логарифмическая сложность

+++++

193. Как называется метод анализа для сравнения затрат времени различных алгоритмов?

=====

#

Асимптотический анализ

=====

Анализ покрытия тестами

=====

Анализ производительности

=====

Анализ кода программы

+++++

194. Какая сложность лучше всего подходит для алгоритмов реального времени?

=====

#

$O(N \cdot \log(N))$

=====

$O(N^2)$

=====

$O(C^N)$

=====

$O(N!)$

+++++

195. Какая сложность указывает на количество памяти, необходимое для алгоритма?

=====

#

Пространственная сложность

=====

Временная сложность

=====

Сложность в худшем случае

=====

Средняя сложность

+++++

196. Как называется функция, показывающая максимальное количество операций алгоритма при данном размере входных данных?

=====

#

Временная сложность в худшем случае

=====

Пространственная сложность

=====

Сложность в среднем случае

=====

Сложность в наихудшем случае

+++++

197. Какая из функций имеет наименьшую сложность?

=====

#

C – константа

=====

N^C $0 < C < 1$

=====

$N \cdot \log(N)$

=====

$C^N \ C > 1$

+++++

198. Какая из функций имеет наибольшую сложность?

=====

#

$N!$

=====

$N \cdot \log(N)$

=====

$N^C \ C > 1$

=====

$\log(N)$

+++++

199. Какую сложность имеет алгоритм с линейной зависимостью от входных данных?

=====

#

$O(n)$

=====

$O(n^2)$

=====

$O(\log(n))$

=====

$O(C^N)$

+++++

200. Что такое статический анализ кода?

=====

#

Анализ кода без его выполнения

=====

Анализ кода во время его выполнения

=====

Анализ дизайна и архитектуры ПО

====

Анализ процессов разработки ПО

++++

201. Что оценивает показатель LOC?

====

#

Количество строк в коде программы

====

Количество уникальных операторов

====

Количество уникальных операндов

====

Количество комментариев в коде

++++

202. Как называется метрика, измеряющая общее количество операторов и операндов в программе?

====

#

Объем программы

====

Длина программы

====

Размер программы

====

Сложность программы

++++

203. Что представляет собой семантический анализ?

====

#

Анализ значений и типов данных

====

Разбор кода на лексемы и токены

====

Проверка синтаксиса программы

====

Анализ выполнения кода во времени

++++

204. Что измеряет показатель L в метриках Холстеда?

=====

#

Предполагаемое количество строк кода

=====

Общее количество операторов и операндов

=====

Количество уникальных операторов

=====

Количество уникальных операндов

++++

205. Какой анализ используется для проверки соответствия кода стандартам?

=====

#

Статический анализ программы

=====

Динамический анализ программы

=====

Синтаксический анализ программы

=====

Семантический анализ программы

++++

206. Что такое вероятностная модель текста программы?

=====

#

Модель для анализа кода

=====

Метод тестирования кода

=====

Средство отладки кода

=====

Процесс компиляции кода

++++

207. Какие компоненты входят в вероятностную модель текста программы?

=====

#

Алфавит операторов

=====

Средства отладки кода

=====

Описание классов программы

=====

Функции обработки ошибок

+++++

208. Что учитывают вероятностные модели текста программы?

=====

#

Структурные особенности

=====

Типы переменных

=====

Архитектуру системы

=====

Версии операционных систем

+++++

209. Для чего используются вероятностные модели текста программы?

=====

#

Анализ и предсказание кода

=====

Сборка и компиляция кода

=====

Мониторинг выполнения кода

=====

Визуализация структуры кода

+++++

210. Какое измеряемое свойство важно для мобильных устройств?

=====

#

Энергопотребление

=====

Объем кода

=====

Количество комментариев

=====

Число пользователей

+++++

211. Какое свойство измеряет способность программы обрабатывать ошибки?

=====

#

Устойчивость

=====

Производительность

=====

Время выполнения

=====

Использование памяти

+++++

212. Какое свойство объединяет время выполнения и использование ресурсов?

=====

#

Производительность

=====

Масштабируемость

=====

Устойчивость

=====

Энергопотребление

+++++

213. Что такое оптимизация программных модулей?

=====

#

Улучшение производительности

=====

Упрощение структуры кода

=====

Создание новых модулей

=====

Разработка интерфейсов

+++++

214. Как называется принцип, где каждый модуль отвечает за одну часть функциональности?

=====

#

Принцип единственной ответственности

=====

Принцип разделения интерфейсов

=====

Принцип инверсии зависимостей

=====

Принцип открытости-закрытости

+++++

215. Как называется метод проектирования, начиная с высокоуровневой архитектуры?

=====

#

Проектирование сверху вниз

=====

Проектирование снизу вверх

=====

Параллельное проектирование

=====

Инкрементальное проектирование

+++++

216. Что учитывают при анализе технологий программирования?

=====

#

Цели проекта

=====

Лицензии ПО

=====

Количество багов

=====

Время тестирования

+++++

217. Какие элементы включают языки программирования?

=====

#

Алфавит и синтаксис

=====

Количество строк кода

=====

Время компиляции

=====

Количество комментариев

+++++

218. Что означает термин «метрика программного обеспечения»?

=====

#

Числовая характеристика программы

=====

Метод тестирования программы

=====

Способ компиляции кода

=====

Алгоритм выполнения задач

+++++

219. Какое свойство измеряется метрикой Джилба?

=====

#

Логическая сложность программы

=====

Время компиляции программы

=====

Количество строк кода

=====

Частота использования переменных

+++++

220. Что учитывает метрика Чепина?

=====

#

Сложность потока данных

=====

Время выполнения кода

=====

Объем памяти программы

=====

Количество строк кода

+++++

221. Что измеряет метрика Холстеда?

=====

#

Количество операторов и операндов

=====

Время выполнения кода

=====

Частоту использования переменных

=====

Временные затраты на разработку

+++++

222. Как определяется длина программы по метрике Холстеда?

=====

#

Сумма всех операторов и операндов

=====

Количество строк кода

=====

Время компиляции программы

=====

Частота использования переменных

+++++

223. Что учитывается при определении структурной сложности?

=====

#

Число взаимодействующих компонент

=====

Время компиляции кода

=====

Объем используемой памяти

=====

Число разработчиков проекта

+++++

224. Что определяет цикломатическая сложность?

=====

#

Количество независимых путей

=====

Число строк в коде программы

=====

Время выполнения кода

=====

Объем использованной памяти

+++++

225. Какой уровень структурной сложности имеет программа с линейным кодом?

=====

#

Уровень 0

=====

Уровень 1

=====

Уровень 2

=====

Уровень 3

+++++

226. Что характеризует уровень 1 структурной сложности?

=====

#

Один уровень ветвлений

=====

Линейный код без ветвлений

=====

Многоуровневые ветвления

=====

Глубокая вложенность структур

+++++

227. Какой уровень структурной сложности имеет программа с несколькими вложенными if-else операторами?

=====

#

Уровень 2

=====

Уровень 0

=====

Уровень 1

=====

Уровень 3

+++++

228. Что характеризует водопадную модель разработки?

=====

#

Линейный подход к разработке

=====

Гибкость и итеративность

=====

Постоянная обратная связь

=====

Использование прототипов

++++

229. Что является основным плюсом Agile-методологии?

=====

#

Гибкость и адаптивность

=====

Строгое следование плану

=====

Меньше взаимодействия команды

=====

Долгий цикл разработки

++++

230. Что включает в себя методология Scrum?

=====

#

Управление проектами на основе Agile

=====

Линейный подход к разработке

=====

Разработка без обратной связи

=====

Использование прототипов

++++

231. Что характеризует быструю разработку приложений (RAD)?

=====

#

Итеративный подход к разработке

=====

Линейный подход к разработке

=====

Отсутствие прототипов

=====

Длительный цикл разработки

++++

232. Какой основной недостаток метода RAD?

=====

#

Не подходит для больших проектов

=====

Требует много документации

=====

Ограниченная гибкость

=====

Долгий цикл разработки

++++

233. Что включает в себя методология Lean?

=====

#

Повышение эффективности за счет устранения лишнего

=====

Линейный подход к разработке

=====

Отсутствие документации

=====

Длительный цикл разработки

++++

234. На чем основана функционально-ориентированная разработка (FDD)?

=====

#

Создание небольших функций или блоков

=====

Линейный подход к разработке

=====

Отсутствие прототипов

=====

Длительный цикл разработки

++++

235. Какой основной плюс методологии Agile?

=====

#

Способность к быстрой адаптации

=====

Высокая стоимость лицензий

=====

Отсутствие документации

=====

Долгий цикл разработки

+++++

236. Что является основным недостатком водопадной модели?

=====

#

Трудно вернуться назад для изменений

=====

Высокая стоимость лицензий

=====

Отсутствие документации

=====

Длительные сроки разработки

+++++

237. Что учитывает метрика Маккейба для оценки сложности программы?

=====

#

Логика программы

=====

Объем данных в программе

=====

Время выполнения программы

=====

Количество строк кода

+++++

238. Что представляет собой граф управления в метрике Маккейба?

=====

#

Граф вершин и дуг

=====

Дерево функций и методов

=====

Список переменных и функций

=====

Таблица значений и констант

+++++

239. Что измеряет временная сложность алгоритма?

=====

#

Количество входных данных

=====

Объем используемой памяти

=====

Количество строк кода

=====

Количество комментариев

+++++

240. Что характеризует максимальную сложность алгоритма?

=====

#

Наиболее неблагоприятный случай

=====

Наиболее благоприятный случай

=====

Среднее время выполнения

=====

Объем используемой памяти

+++++

241. Какой показатель указывает на излишнюю сложность программы по Маккейбу?

=====

#

Цикломатическое число больше 10

====

Время выполнения больше 10 секунд

====

Объем памяти больше 10 МБ

====

Количество строк больше 100

++++

242. Что помогает улучшить качество кода по метрике Маккейба?

====

#

Разбиение сложного кода

====

Увеличение объема документации

====

Сокращение числа функций

====

Увеличение количества комментариев

++++

243. Что такое процедурно-ориентированные метрики?

====

#

Оценка качества программного кода

====

Подсчет количества строк кода

====

Измерение времени выполнения

====

Оценка объема памяти

++++

244. Что включает в себя набор метрик Мартина?

====

#

Категории классов

====

Время выполнения кода

=====

Количество методов в классе

=====

Глубину наследования

+++++

245. Что такое зона боли в метриках Мартина?

=====

#

Устойчивые конкретные компоненты

=====

Классы с низким уровнем наследования

=====

Время выполнения программы

=====

Объем используемой памяти

+++++

246. Что показывает метрика NOC (Number of Children)?

=====

#

Количество потомков класса

=====

Время выполнения методов

=====

Количество строк кода

=====

Объем используемой памяти

+++++

247. Какую роль играют метрики Чидамбера и Кемерера?

=====

#

Оценка качества методов и классов

=====

Время выполнения программы

=====

Количество строк кода

=====

Объем используемой памяти

++++

248. Какой моделью является модель Холстеда?

=====

#

Прогнозной моделью

=====

Оценочной моделью

=====

Измерительной моделью

=====

Моделью управления памятью

++++

249. Какой метод используется в прогнозных моделях для анализа данных?

=====

#

Статистический анализ

=====

Логический анализ

=====

Синтаксический анализ

=====

Семантический анализ

++++

250. Что оценивает модель зрелости CMM/CMMI?

=====

#

Зрелость процессов разработки

=====

Время выполнения программ

=====

Объем используемой памяти

=====

Количество комментариев в коде

++++

251. Что включает SWOT-анализ?

=====

#

Сильные и слабые стороны

=====

Время компиляции кода

=====

Объем используемой памяти

=====

Количество строк кода

++++

252. Что учитывает модель Елински-Морандо?

=====

#

Исторические данные о сбоях

=====

Количество строк в коде

=====

Время компиляции программы

=====

Объем используемой памяти

++++

253. Что используется в модели Нельсона для описания прогонов программы?

=====

#

Теория вероятностей

=====

Логические схемы

=====

Количество строк кода

=====

Время выполнения

++++

254. Что характеризует вероятность Р в модели Нельсона?

=====

#

Обнаружение дефекта при прогоне

=====

Количество строк в коде

=====

Время компиляции программы

=====

Объем используемой памяти

+++++

255. Что показывает вероятность R в модели Нельсона?

=====

#

Приемлемый результат прогона

=====

Время выполнения кода

=====

Количество комментариев в коде

=====

Объем используемой памяти

+++++

256. Что оценивает модель Мусона?

=====

#

Надежность программных систем

=====

Время выполнения программы

=====

Количество строк в коде

=====

Объем используемой памяти

+++++

257. Что определяет стандарт IEEE 730?

=====

#

Требования к плану обеспечения качества

=====

Модель зрелости процессов

=====

Время выполнения программы

=====

Объем используемой памяти

+++++

258. Что описывает стандарт IEEE 829?

=====

#

Процесс и документацию тестирования

=====

Модель зрелости процессов

=====

Время выполнения программы

=====

Объем используемой памяти

+++++

259. Какой стандарт определяет общие процессы разработки ПО?

=====

#

ISO/IEC 12207

=====

ISO 9000 серии

=====

IEEE 730

=====

CMMI модели

+++++

260. Что включает в себя стандарт ITIL?

=====

#

Практики для управления IT-инфраструктурой

====

Время выполнения программы

====

Количество строк в коде

====

Объем используемой памяти

++++

261. Что определяет стандарт услуг?

====

#

Критерии и процедуры обеспечения качества

====

Время выполнения программы

====

Объем используемой памяти

====

Количество строк в коде

++++

262. Что определяет стандарт терминов?

====

#

Единый словарь ключевых терминов

====

Время выполнения программы

====

Объем используемой памяти

====

Количество строк в коде

++++

263. Какая модель жизненного цикла предусматривает последовательное
выполнение всех этапов проекта?

====

#

Каскадная модель

====

Спиральная модель

=====

Итеративная модель

=====

Модель с промежуточным контролем

+++++

264. Что включает в себя поэтапная модель с промежуточным контролем?

=====

#

Итерации с циклами обратной связи

=====

Полное завершение каждого этапа

=====

Время выполнения программы

=====

Количество строк в коде

+++++

265. Что представляет собой сертификация программного обеспечения?

=====

#

Оценка качества программных объектов

=====

Разработка программного обеспечения

=====

Управление проектами разработки

=====

Тестирование программного обеспечения

+++++

266. Что является основной целью сертификации программных средств?

=====

#

Контроль качества технологий

=====

Увеличение объема данных

=====

Сокращение времени разработки

=====

Уменьшение количества строк кода

+++++

267. Что такое система сертификации?

=====

#

Система правил и процедур сертификации

=====

Модель зрелости процессов

=====

План обеспечения качества

=====

Описание жизненного цикла ПО

+++++

268. Какие виды сертификации существуют?

=====

#

Обязательная и добровольная

=====

Формальная и неформальная

=====

Внутренняя и внешняя

=====

Прямая и косвенная

+++++

269. Какие документы используются при сертификации?

=====

#

Международные и государственные стандарты

=====

Модель зрелости процессов

=====

План обеспечения качества

=====

Описание жизненного цикла ПО

++++

270. Что принимается на сертификацию?

=====

#

Рабочие версии программных продуктов

=====

Демо-версии программных продуктов

=====

Бета-версии программных продуктов

=====

Альфа-версии программных продуктов

++++