Тестовые вопросы с несколькими вариантами ответа

1. Базовые концепции программной инженерии

1. Какой принцип программной инженерии подразумевает разделение системы на независимые компоненты?

a) Абстракция

b) Модульность

c) Инкапсуляция

d) Повторное использование

Ответ: b) Модульность

2. Чем отличается программная инженерия от программирования?

a) Программирование включает управление проектами, а ПИ — только написание кода.

b) ПИ охватывает полный жизненный цикл ПО, а программирование — только реализацию.

c) Программирование требует знания алгоритмов, а ПИ — нет.

d) Это синонимы.

Ответ: b) ПИ охватывает полный жизненный цикл ПО, а программирование — только реализацию.

2. Цели и задачи программной инженерии

3. Какая из перечисленных целей НЕ относится к программной инженерии в бизнес-контексте?

a) Сокращение времени выхода на рынок

b) Увеличение затрат на поддержку

c) Обеспечение масштабируемости системы

d) Минимизация рисков

Ответ: b) Увеличение затрат на поддержку

4. Что включает в себя качество ПО?

a) Только функциональность

b) Надежность, производительность и удобство использования

c) Стоимость разработки

d) Все перечисленное, кроме c)

Ответ: d) Все перечисленное, кроме c)

3. Структура дисциплины

5. Какой компонент НЕ входит в структуру программной инженерии?

a) Управление проектами

b) Маркетинговая стратегия

c) Архитектура и дизайн

d) Тестирование

Ответ: b) Маркетинговая стратегия

6. Какую роль играют стандарты (например, ISO/IEC 12207) в ПИ?

a) Увеличивают гибкость разработки

b) Обеспечивают единый подход к созданию ПО

c) Заменяют необходимость тестирования

d) Снижают требования к документации

Ответ: b) Обеспечивают единый подход к созданию ПО

4. История развития программной инженерии

7. В каком десятилетии программная инженерия стала отдельной дисциплиной?

a) 1950-е

b) 1970-е

c) 1990-е

d) 2000-е

Ответ: b) 1970-е

8. Какое событие произошло в 2001 году?

a) Создание языка Fortran

b) Разработка Agile-манифеста

c) Появление DevOps

d) Внедрение облачных технологий

Ответ: b) Разработка Agile-манифеста

5. Методологии и процессы

9. Какая методология характеризуется низкой гибкостью и медленной доставкой?

a) Agile

b) Waterfall

c) DevOps

d) Spiral

Ответ: b) Waterfall

10. Что является ключевым принципом гибкой разработки (Agile)?

a) Строгое следование плану

b) Приоритет документации над рабочим ПО

c) Готовность к изменениям требований

d) Отсутствие тестирования на ранних этапах

Ответ: c) Готовность к изменениям требований

6. Инструменты и технологии

11. Какой инструмент относится к CI/CD?

a) Git

b) Jenkins

c) Visual Studio

d) Jira

Ответ: b) Jenkins

12. Что НЕ является преимуществом автоматизации в ПИ?

a) Сокращение времени разработки

b) Увеличение количества ошибок

c) Упрощение управления инфраструктурой

d) Повышение качества кода

Ответ: b) Увеличение количества ошибок

7. Командная работа

13. Для чего нужно код-ревью?

a) Для увеличения технического долга

b) Для выявления ошибок и обмена опытом

c) Для сокращения времени на тестирование

d) Для замены документации

Ответ: b) Для выявления ошибок и обмена опытом

14. Что НЕ относится к принципам управления конфликтами?

a) Открытое обсуждение

b) Игнорирование проблемы

c) Поиск компромиссов

d) Анализ причин конфликта

Ответ: b) Игнорирование проблемы

8. Качество и тестирование

15. Какой тип тестирования оценивает удобство интерфейса?

a) Функциональное

b) Регрессионное

c) Юзабилити

d) Нагрузочное

Ответ: c) Юзабилити

16. Что включает непрерывная интеграция?

a) Ручное тестирование раз в месяц

b) Автоматизацию сборки и тестирования

c) Отказ от использования контроля версий

d) Увеличение времени релизов

Ответ: b) Автоматизацию сборки и тестирования

9. Безопасность и надежность

17. Что НЕ относится к методам защиты данных?

a) Использование HTTPS

b) Хранение паролей в открытом виде

c) Регулярное обновление ПО

d) Шифрование данных

Ответ: b) Хранение паролей в открытом виде

18. Что такое отказоустойчивость системы?

a) Полное отсутствие сбоев

b) Возможность работы при частичных сбоях

c) Отказ от резервирования

d) Увеличение времени восстановления

Ответ: b) Возможность работы при частичных сбоях

10. Современные тенденции

19. Какое преимущество дает микросервисная архитектура?

a) Зависимость компонентов друг от друга

b) Возможность параллельной разработки

c) Увеличение времени развертывания

d) Централизованное управление

Ответ: b) Возможность параллельной разработки

20. Что позволяет Serverless-архитектура?

a) Управление серверами вручную

b) Сосредоточение на коде без управления инфраструктурой

c) Увеличение затрат на инфраструктуру

d) Только локальное развертывание

Ответ: b) Сосредоточение на коде без управления инфраструктурой

11. Практические аспекты

21. Что такое технический долг?

a) Финансовые затраты на разработку

b) Последствия решений, упрощающих текущую разработку, но усложняющих будущую

c) Долгосрочные инвестиции в инфраструктуру

d) Отсутствие документации

Ответ: b) Последствия решений, упрощающих текущую разработку, но усложняющих будущую

22. Для чего проводится рефакторинг кода?

a) Увеличение технического долга

b) Изменение функциональности системы

c) Улучшение читаемости без изменения функциональности

d) Сокращение времени тестирования

Ответ: c) Улучшение читаемости без изменения функциональности