

Вопросы к экзамену «Технология разработки программного обеспечения»

Юрий Литвинов
y.litvinov@spbu.ru

1. Программная инженерия как область знания
2. Отличия разработки программного обеспечения от других инженерных областей
3. Компетенции и профстандарты в области программной инженерии
4. Понятие жизненного цикла программного обеспечения
5. Водопадная модель жизненного цикла
6. Итеративная, итеративно-инкрементальная и спиралевидная модели жизненного цикла
7. Rational Unified Process
8. Agile-подход к разработке
9. eXtreme Programming: общий подход, достоинства и недостатки
10. XP: практики «Короткий цикл разработки»
11. XP: практики «Непрерывность процесса»
12. XP: практики «Понимание, разделяемое всеми»
13. Scrum, роли в команде
14. Scrum, Backlog и спринты
15. ScrumAnd, ScrumBut, достоинства и недостатки методологии
16. Понятие и виды требований
17. Требования к требованиям
18. Работа с требованиями: выявление, анализ, проверка
19. Навыки и трудовые функции аналитика
20. Документы, связанные с требованиями
21. Моделирование требований
22. Спецификация требований к программному обеспечению (SRS)
23. Управление требованиями
24. Понятие User experience

25. Стадии проектирования пользовательского интерфейса
26. User-centered design, персонажи и сценарии
27. Activity-centred design
28. Data-driven design
29. Сторителлинг и раскадровки как инструмент проектирования UI
30. Макеты и дизайн-макеты как инструмент проектирования UI
31. Варианты исследования UX продукта
32. Usability-тестирование
33. Функции менеджера проекта
34. Матрица ответственности и план коммуникаций
35. Основные действия по управлению рисками
36. Декомпозиция проекта
37. Построение графика работ: матрица зависимостей, сетевой график
38. Построение графика работ: диаграмма Ганта
39. Типичные ошибки при оценке проектов
40. Треугольник равновесия проекта
41. Приёмы балансирования равновесия на уровне проекта
42. Приёмы балансирования равновесия на уровне бизнес-целей
43. Отслеживание прогресса проекта
44. Позитивная экосистема команды: базовые правила и сплочённость команды
45. Позитивная экосистема команды: умение слушать, умение проводить совещания
46. Совместное решение задач: анализ задач и варианты принятия решений
47. Совместное решение задач: разрешение конфликтов и непрерывное обучение
48. Особенности формирования команды
49. Особенности командной разработки программного обеспечения, ревью кода
50. Понятие и виды тестирования программного обеспечения, пирамида тестирования
51. Тестирование требований, тестирование архитектуры
52. Тестовые сценарии

53. Инструменты тестирования
54. Отслеживание ошибок, жизненный цикл ошибки
55. Законы Лемана эволюции программного обеспечения
56. Понятие и особенности унаследованных систем
57. Сопровождение программного обеспечения: факторы стоимости, прогнозирование, процесс
58. Организация технической поддержки программного обеспечения
59. Особенности работы с унаследованным кодом
60. Реинжиниринг программного обеспечения: варианты, факторы стоимости, основные этапы
61. Понятие рефакторинга
62. «Code smells»
63. Рефакторинги «Выделение метода» и «Перемещение метода»
64. Рефакторинги «Выделение класса» и «Выделение подкласса»
65. Рефакторинги «Соккрытие делегирование» и «Введение внешнего метода»
66. Рефакторинги «Самоинкапсуляция поля» и «Введение Null-объекта»
67. Рефакторинги «Замена кода типа подклассами» и «Замена условного оператора полиморфизмом»
68. Рефакторинги «Замена конструктора фабричным методом» и «Замена наследования делегированием»
69. Проблемы при проведении рефакторинга. Причины не проводить рефакторинг
70. Понятие Continuous Delivery
71. Антипаттерны управления релизами
72. Преимущества частых автоматизированных релизов
73. Принципы непрерывного развёртывания, конфигурационное управление
74. Continuous Integration
75. Оценка. Используемые метрики. Конус неопределённости.
76. Этапы процесса оценки.
77. Частые ошибки при оценке.