**На 3:**

"1) Понятие архитектуры ПС.

2) Унифицированный язык моделирования ПС. Достоинства и возможности.

3) ООП: общая характеристика.

4) Метод диаграмм потоков данных.

5) Характеристики программных модулей.

6) Структурное программирование: общая характеристика.

7) Принципы структурирования программных продуктов.

8) Задачи руководства программным проектом.

9) Характеристики этапов (содержание) разработки ПО.

10) Компонентно-ориентированную модель.

11) Спиральную. модель ЖЦ.

12) Классический жизненный цикл ПО (ЖЦ).

13) Отношения в UML.

14) Предметы в UML.

15) Строительные блоки языка UML.

16) Классы эквивалентности и граничные условия.

17) Информационные потоки процесса тестирования.

18) Основные понятия и принципы тестирования ПО.

19) Диаграммы в UML: общая характеристика.

20) Классификаторы в UML.

21) Способы тестирования условий.

22) Особенности тестирование объектно-ориентированного программного обеспечения.

23) Виды тестирования объектно-ориентированного программного обеспечения.

24) Проверки, ревизии и контрольные прогоны.

25) Виды тестовой деятельности.

**На 4:**

1) Применять метод Джексона.

2) Использовать диаграммы потоков данных ПДД.

3) Использовать структурное проектирование.

4) Использовать модель СОСОМО II.

5) Использовать модель зрелости процессов разработки ПО (CMM).

6) Оценивать надежность ПО.

7) Определять характеристики качества программного обеспечения.

8) Использовать модель раннего этапа проектирования. Модель COCOMO.

9) Определять элементы данных при расчете FP-метрики.

10) Использовать функционально ориентированные метрики.

11) Оценивать временные затраты на реализацию этапов разработки ПО.

12) Использовать агрегацию при построении моделей в UML.

13) Использовать наследование при построении моделей в UML.

14) Использовать инкапсуляцию при построении моделей в UML.

15) Использовать абстрагирование при построении моделей в UML.

16) Использовать регрессионное тестирование.

17) Планировать разработку тестов.

18) Моделировать требования к ПО.

19) Разрабатывать прецеденты и потоки событий.

20) Использовать прямое и обратное проектирование.

21) Разрабатывать диаграммы деятельности.

22) Разрабатывать диаграммы сотрудничества (кооперации).

23) Разрабатывать диаграммы компонентов.

24) Разрабатывать диаграммы схем состояний. События и действия.

25) Разрабатывать диаграммы последовательности.

26) Использовать отношения в диаграммах классов.

27) Разрабатывать диаграммы классов. Свойства и операции класса и их видимость.

28) Разрабатывать диаграммы вариантов использования. (Use Case) (Привести пример).

29) Выполнять тестирование программных путей.

30) Использовать шаблон стандартного плана тестирования IEEE829.

**На 5:**

1) Методом пошаговой детализации.

2) Структурной разработкой программ.

3) Выполнением оценки проекта на основе FP-метрик.

4) Выполнением оценки проекта на основе LOC -метрик.

5) Выполнением оценки в ходе руководства проектом.

6) Документированием программного продукта.(ЕСПД).

7) Тестированием переходов между состояниями.

8) Стандартной процедурой тестирования ""черного ящика"".

9) Структурным тестированием (тестирование «белого ящика»).

10) Функциональным тестированием (тестирование «черного ящика»).

11) Оценкой цикломатической сложности ПО.

12) Использованием модели раннего этапа проектирования COCOMO2.

13) Использованием модели композиции приложенияCOCOMO2.

14) Использованием модели постархитектуры COCOMO2.

15) Разработкой технического задания.

16) Оценкой тестов на основе критерия охвата строк.

17) Разработкой плана классов эквивалентности при тестировании.

18) Способом развернутого описания прецедента.

19) Методами построения модели предметной области при ООП.

20) Методами выделения концептуальных классов при ООП.

21) Методами добавления ассоциаций и атрибутов в модель предметной области при

ООП.

22) Оценкой тестов на основе критерия охвата ветвлений.

23) Детализацией модели прецедентов с помощью описаний операций.

24) Способами проектирование на основе описаний операций.

25) Способами использования шаблонов проектирования GRASP.

26) Использованием шаблон Information Expert.

27) Оценкой тестов на основе критерия охвата условий.

28) Методами оценки сопровождения программных продуктов.

29) Методами оценки характеристик хорошего теста.

30) Способами документирования результатов функционального тестирования.