**Методологии разработки ПО. Унифицированный процесс.**

Для создания большого по появился системный подход (декомпозиция) – способ организации действий, который выявляет закономерности и взаимосвязи элементов системы с целью её более эффективного использования.

Основные принципы системного подхода

* Целостность
* Иерархичность состояния
* Структуризация
* Множественность
* Системность

Жизненный цикл – время от идеи до вывода по из эксплуатации.

Основные этапы

* Разработка требований
* Проектирование
* Тестирование
* Эксплуатация
* Анализ
* Разработка
* Внедрение
* Вывод из эксплуатации

Методологии разработки

Модель ЖЦ ПО – структура, определяющая последовательность выполнения и взаимодействия процессов действий и задач на протяжении всего ЖЦ.

Основные группы моделей ЖЦ.

* Последовательная (однократная) – определены все требования, мы проходимся по всем этапам ЖЦ 1 раз.
  + Водопадная (каскадная) модель. Основная особенность – мы можем откатиться по процессу построения ПО на двух этапах – С тестирования к архитектурному дизайна программы, с архитектурного дизайна – к определению требований.
* Инкрементная – все требования также определены, однако есть возможность разбить процесс разработки на несколько этапов.
* Эволюционная – все требования не определены, доопределяются со временем. Деградирует со временем, так как накапливает дефекты на этапах, поэтому требуется рефакторинг.
* Методология Ройса
  + В начале делаем предварительный дизайн – спроектировать, определить и создать модели обработки данных и разработать документ – образ будущей системы.
  + Документирование – наше всё.
  + Do it twice – делаем тестовую разработку и используем её в качестве пилота с сокращенным временем разработки – для оценки основного ПО.
  + Тестирование – при тестировании мы исключаем дизайнера системы из процесса тестирования. Проверяем все логический пункты программы, исправляем ошибки.
  + Подключить пользователя перед финальной поставкой продукта.
* V образная модель – в конце каждой стадии происходит её тестирование. Для кода – юнит тесты, для дизайна системы – интеграция блоков системы и её полное тестирование.
* Многопроходная модель – похожа на V образную, но разработка и интеграция происходит в несколько этапов.
* Модель прототипирования. Основной принцип – сделать быстрый прототип и показать его пользователю, так выбирается нужный прототип(пока все не устроит пользователя).
* Спиральная модель – в данной модели главным нововведение стало учет и подсчет рисков при разработке ПО.
* Быстрая разработка ПО – case системы. ПО создается из “кубиков” пользователями. Кубики – основные части системы написаны заранее.

RUP

Унифицированный процесс разработки ПО. Разбит на фазы

* **Начало**  
  Возможно ли в прицепе сделать проект за наши сроки и деньги, архитектурой итд. Построение графика работ.

Веха – Прийти к соглашению с заказчиками, оценить риски.

* **Проектирование**  
  Выбрать исполняемую структуру. Проверить архитектуру проекта. Т.е. мы уверенны что мы точно сможем все реализовать. Желательно после этого не вносить после этого этапа никаких доработок в архитектуру.

Веха – Согласовать структуру тестирование итд с заказчиком.

* **Построение**   
  Экономически эффективно и максимально быстро разработать продукт, провести тесты, подготовить продукт к установке пользователям.

Веха – Добиться стабильности продукта, все встороны готовы к передаче продукта пользователям.

* **Внедрение**Запуск продаж, реклама, договоры итд

Веха – удовлетворены ли пользователи?

**Требования и их категоризация. Атрибуты требований.**

**Язык UML.**

**Прецеденты использования. UseCase-диаграммы - состав, виды связей.**