

Лекция 1

Sunday, January 30, 2022 6:32 PM

Проектирование архитектурно-программных систем

Что такое архитектура?

Здание — искусство и наука
Строить сооружения, а также сама
совокупность зданий и сооружений.

Описание — чертежи.

Бывают разные предназначения,
разные сложности, масштабы,
Технологии, стили

Особенность — ШАБЛОНЫ.

Архитектура сооружений и ПО. СХОДСТВО

1. Разнообразие, сложность, крупно-масштабность.
2. Большие команды, используется разделение труда.
3. Есть нормативы и требования заказчика. Часто, одно из требований — документация.
4. Большое количество подсистем,

объекты рассматриваются с разных точек зрения разными специалистами.

5. [план, сроки. Есть множество этапов, результаты принимаются ответственными лицами. Процесс — прозрачный, непрерывный. Важно сокращение издержек.
6. Использование шаблонов
7. Архитектура определяет возможности использования технологий/шаблонов.

Различия:

1. ИТ — молодая область, многие решения делятся "на ходу". Ревизию шаблонов.
2. Программные системы сложны и объёмны. ПО часто обладает уникальными качествами
3. ПО виртуально. Нет физических материалов, есть идеи и концепции.
4. ПО динамично. Основное внимание на поведение.
5. В жизненном цикле ПО нет стадии производства. Обновление имеет другой характер.
6. ПО развивается постоянно и

обновляется.

Программная архитектура — целый
«город» систем, комплекс.

Что такое архитектура
программной системы?

1. Набор ключевых решений об устройстве системы.
Ключевое решение: — структура с-мы
— поведение
— характер взаимодействия
— нефункциональные свойства
— реализации и развертывание.
2. Совокупность структур с-мы
(прогр-ные эт-ты, видимые свойства этих элементов, отношения между ними).
3. Мин-ый набор свойств, определяющих, какие пр-мы будут работать в системе и какие результаты они дадут.
4. Глубинная структура программы.
5. Architecture = {elements, form, rationale}
6. Фундаментальная функция системы, заключающаяся в наборе ее компонентов, их отношений между собой и с внешней средой, а также

принципы управления и проектирования и функцией.

7. Те аспекты развернутого ПО, которые имеют всего лишь один.

8. Архитектура включает —
 — элемент с-мы
 — правила их взаимодействия
 — шаблоны их композиции
 — ограничения, накладываемые на эти паттерны.

9. Архитектура имеет дело с проектированием и реализацией высокоуровневой структуры ПО. При этом используются механизмы абстрагирования, декомпозиции, композиции. Учитываются требования стиля и эстетики.

10. Архитектура говорит, что должно произойти, а разработка — как сделать, чтобы это произошло.

11. Relatively simple internal structure.

12. The class and object structure
 No one particular architecture can really be deemed "correct".

Σ = это структурное и победительское устройство программной системы.

+ правила по развитию.

Аспекты архитектуры

- Структура системы
- Атрибуты качества (характеристики)
- Архитектурные решения
- Принципы проектирования (и разработки)
+ КОНТЕКСТ.

↓ соответствие нефункц. требованиям

Какие должны быть архитектор программной системы?

Архитектор — это тот, кто занимается «важными вещами» в рамках проекта.

Ожидание от архитектора

1. Принимает архитектурные решения и отвечает за них
2. Непрерывно анализирует арх-ту проекта
3. Знает современные технологии и предлагает пути их развития.
4. Гарантирует, что проект реализуется в соответствии с принятыми решениями.
5. Обладает опытом создания разных проектов.
6. Знает предметную область системы
7. Умеет эффективно сотрудничать

Разбирательство в политической ситуации в компании

Области знаний и умений

Проектирование; IT-DevOps, Архитектурные стили, шаблоны и антипаттерны, Механика и способы интеграции, Визуализация, Ивентуальные системы, Оценка и измерение ПО, Коммуникация, Документирование; Управление и администрирование.

↓
Широкий кругозор

↓
и тех. спец. и нет

↓
Искусство компромисса

↓
макро —, ~~микро~~ — менеджер.

Составляющие дисциплины.

1. Инструменты: Архит-ный процесс, UML, шаблоны документов.
2. Практика
3. Исторический экскурс

UML

* Унифицированный процесс - разработка ПО в целом.

- Итерационная разработка
- Управление Требованиями.
- Компонентная архитектура.
- Визуальное моделирование.
- Непрерывная проверка качества.
- Контроль изменений и конфигураций.

UML - универсальный язык моделирования (расширенный)

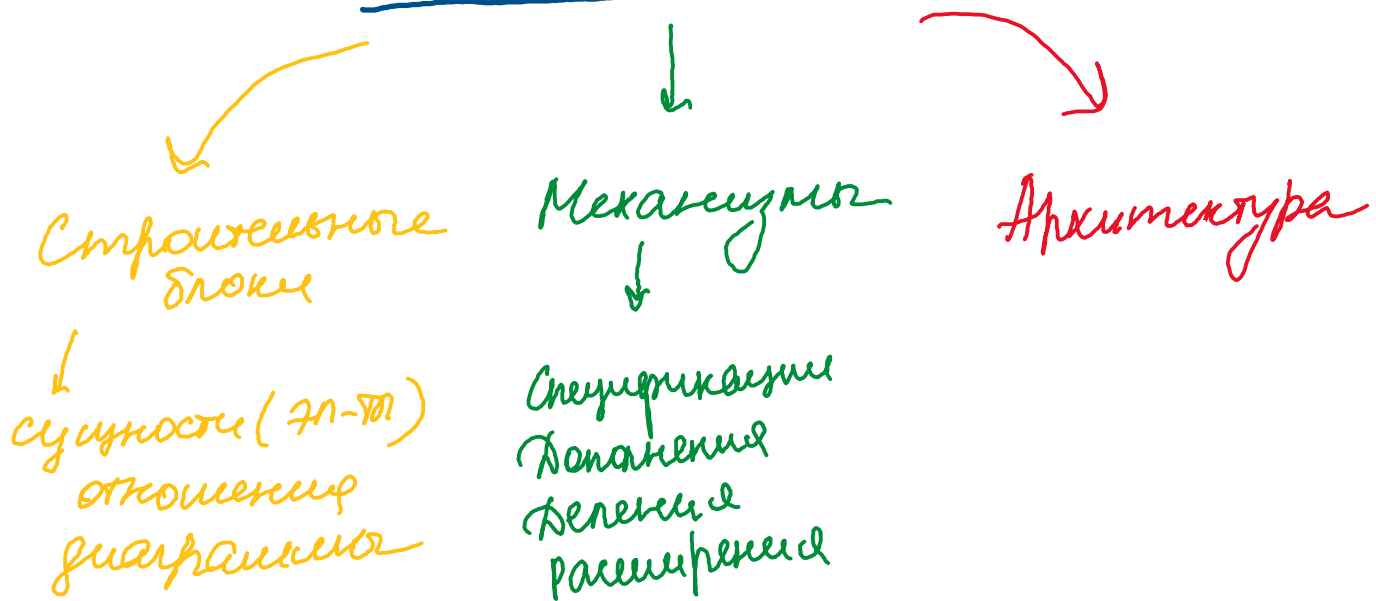
Что унифицирует UML?

0. Лучшие идеи из классических методов моделирования
1. Мотивы/этапы/виды работ жизн. цикла
2. Прикладные области
3. Языки реализации/платформы / ОС
4. Процесс разработки.
5. Свои собственные составленные и концепции.

UML - объектный язык
 - моделирование с помощью
взаимодействующих объектов, которые
 состоят из данных и поведения

Аспект модели: - структурный
 (статика)
 - поведенческий
 (динамика)

Составляющие UML



Диаграммы, например такие:
 class diagram,
 Deployment diagram,
 Use - Case diagram,
 Sequence diagram

Требования

✓
Функциональные
(поведение системы)

→ Нефункциональные
надежность, без-оп.,
удобство, производи-ть,
удобство поддержки,
особенности реализации

↓
Атрибут качества

↓
Архитектурные хар-ки

Процесс выявления требований

Цель — создать спецификацию будущей с-мы
Спецификация — договор между
автором спец-ии и разработчиком.

- Спец-ии должна быть простой
 - Предусматривает проверку соответствия себе
 - Ясна всем заинтересованным лицам и принимателем ими
- Отражает реальный мир

Артифакт уч. процесса, связанное
с требованиями

Модели прецедентов
 Спецификации
 Глоссарий
 Документы - понятия
 Бизнес-правила

Прецедент - это услуга системы,
 типовой вариант исп-тия.

Актеры/жоры - действующие лица⁺

Набор прецедентов - формат записи
 функц. требований

Фокус на что, а не как.

Актер → Первичный (инициатор)
 → Вторичный

Прецеденты

1. Определить границы системы
2. Выявить жоры
3. Опред-ть и документировать сценарии
 взаимодействия каждого жора
 с системой