Лекция 6.

Процессы разработки

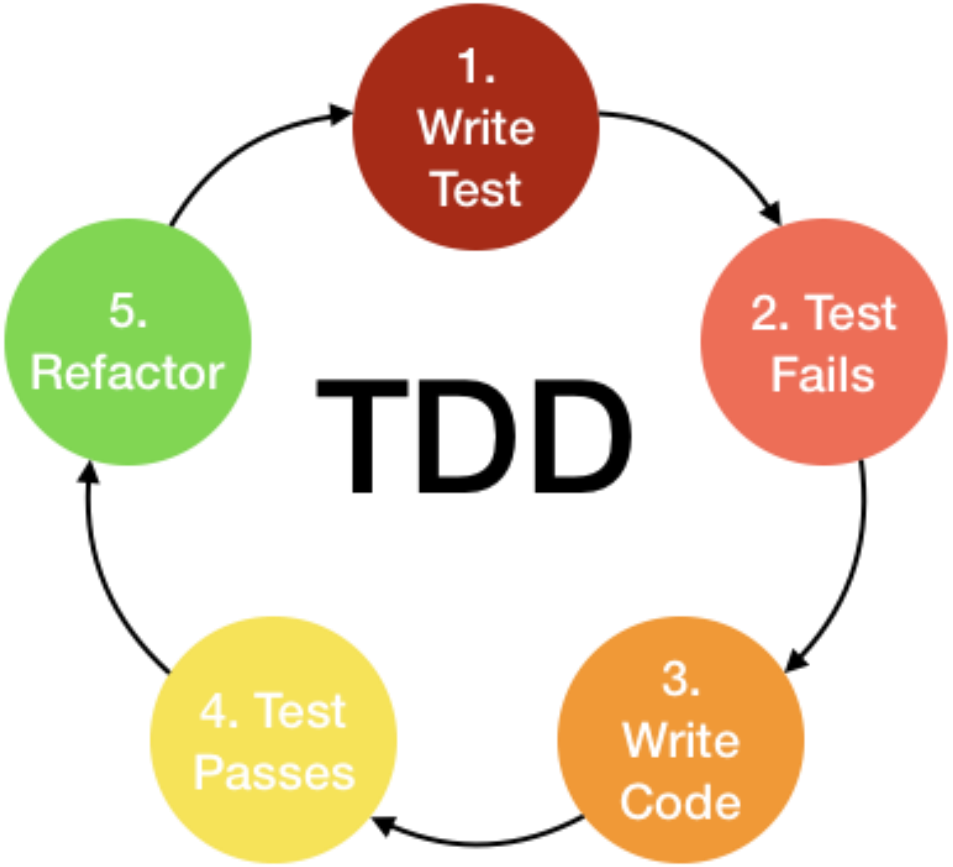
Процессы развертывания, поддержки, развития

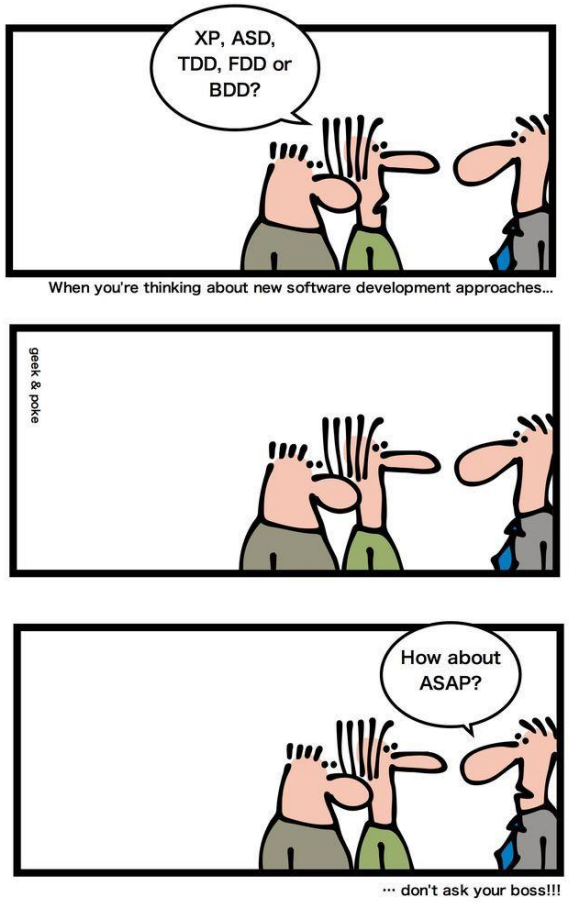
Заключение

Подходы к процессу разработки (* Driven Development)

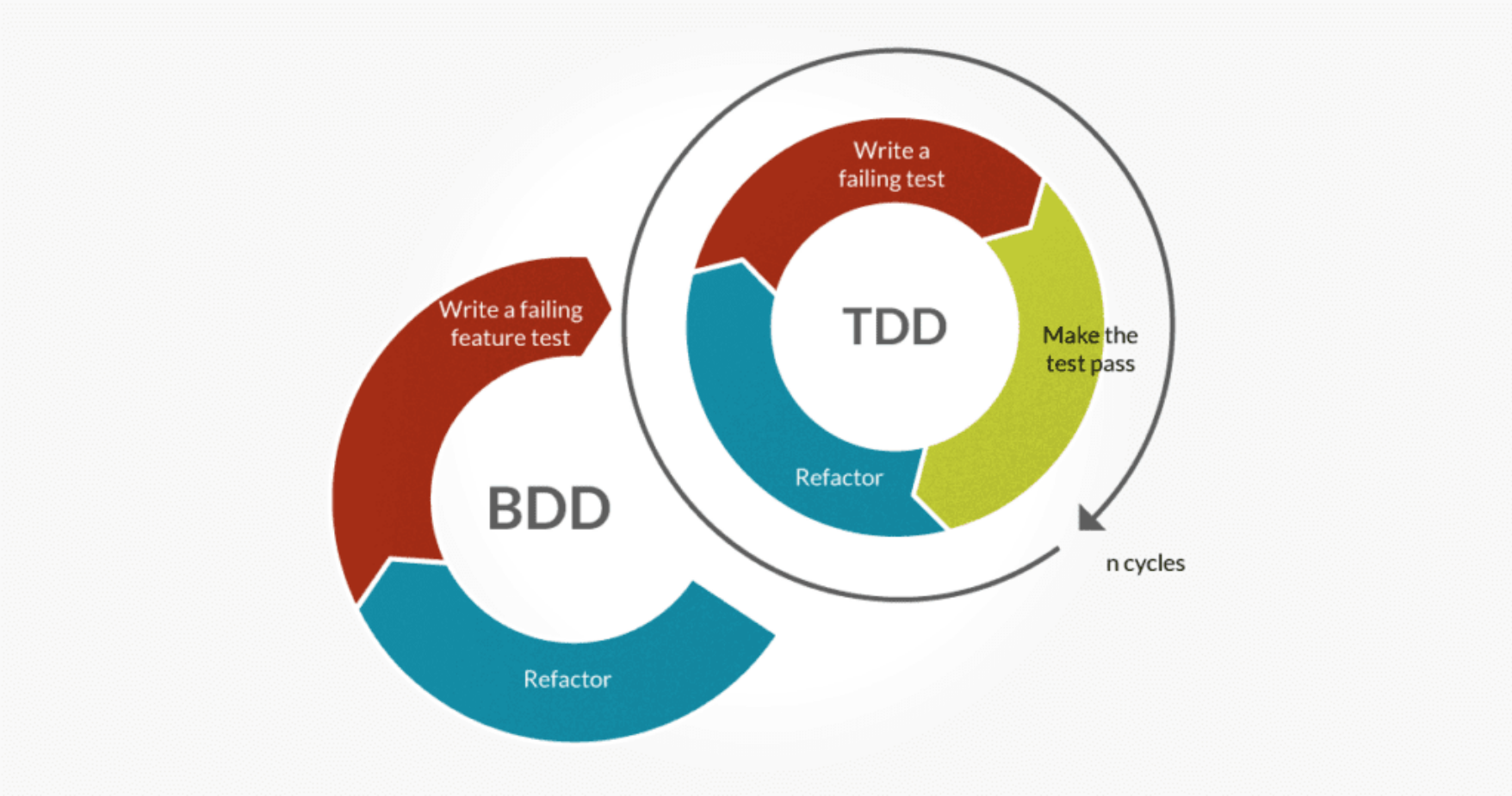
- Разработка через тестирование (TDD - Test DD)
- Разработка через пользовательские сценарии (BDD - Behavior DD)
- Разработка на основе типов (TDD — Type DD)
- Разработка на основе features (FDD — Features DD)
- Разработка на основе модели (MDD – Model DD)
- Разработка на основе паники (PDD — Panic DD)

TDD – Test Driven Development

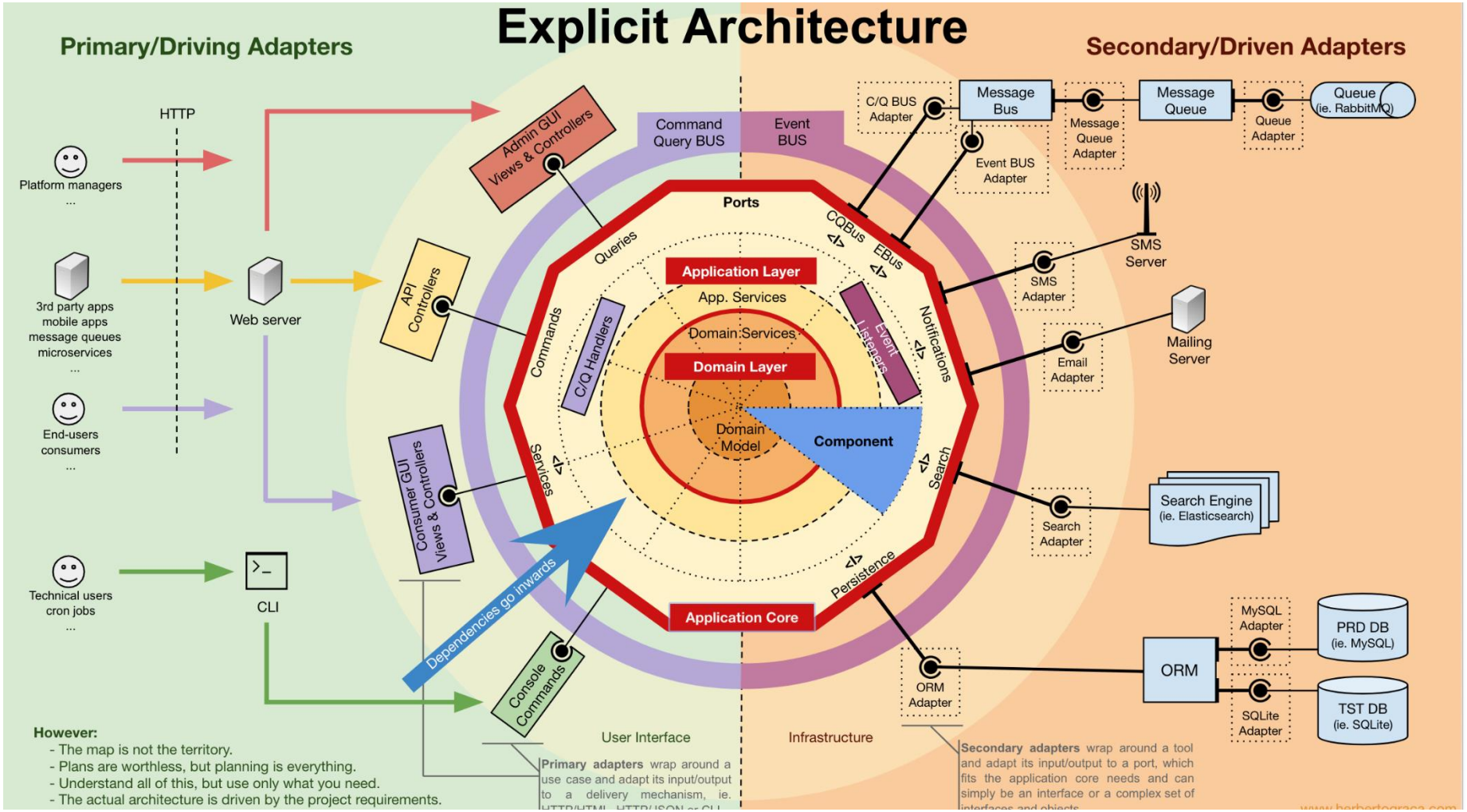




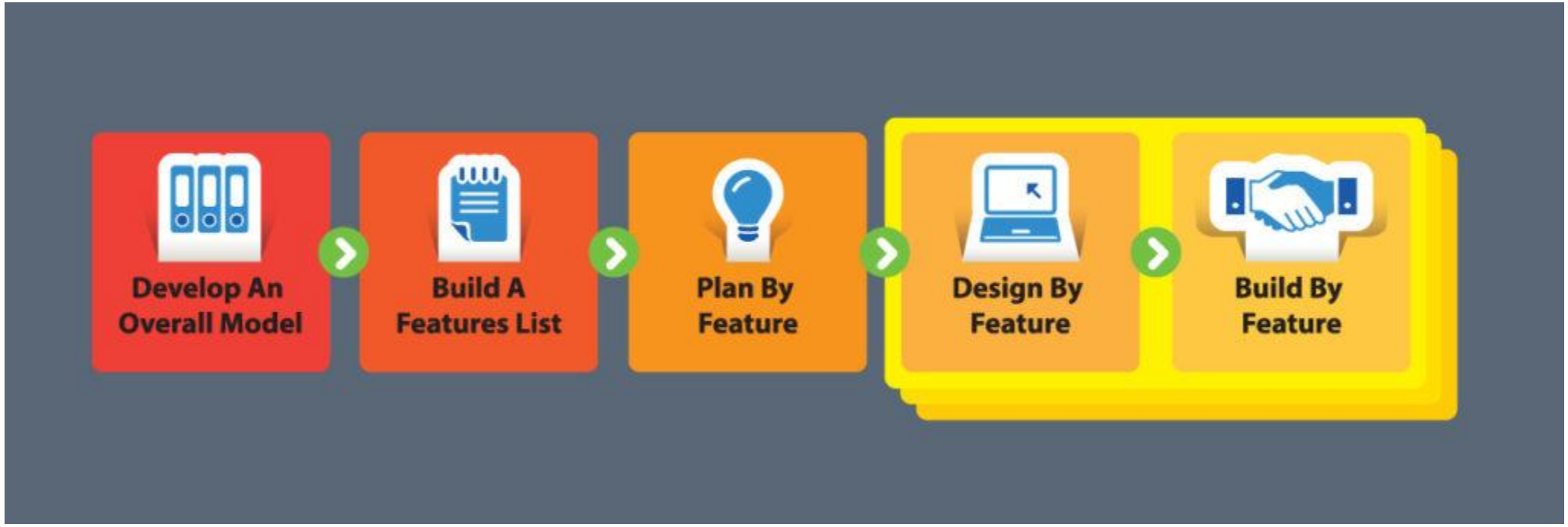
BDD – Behavior Driven Development



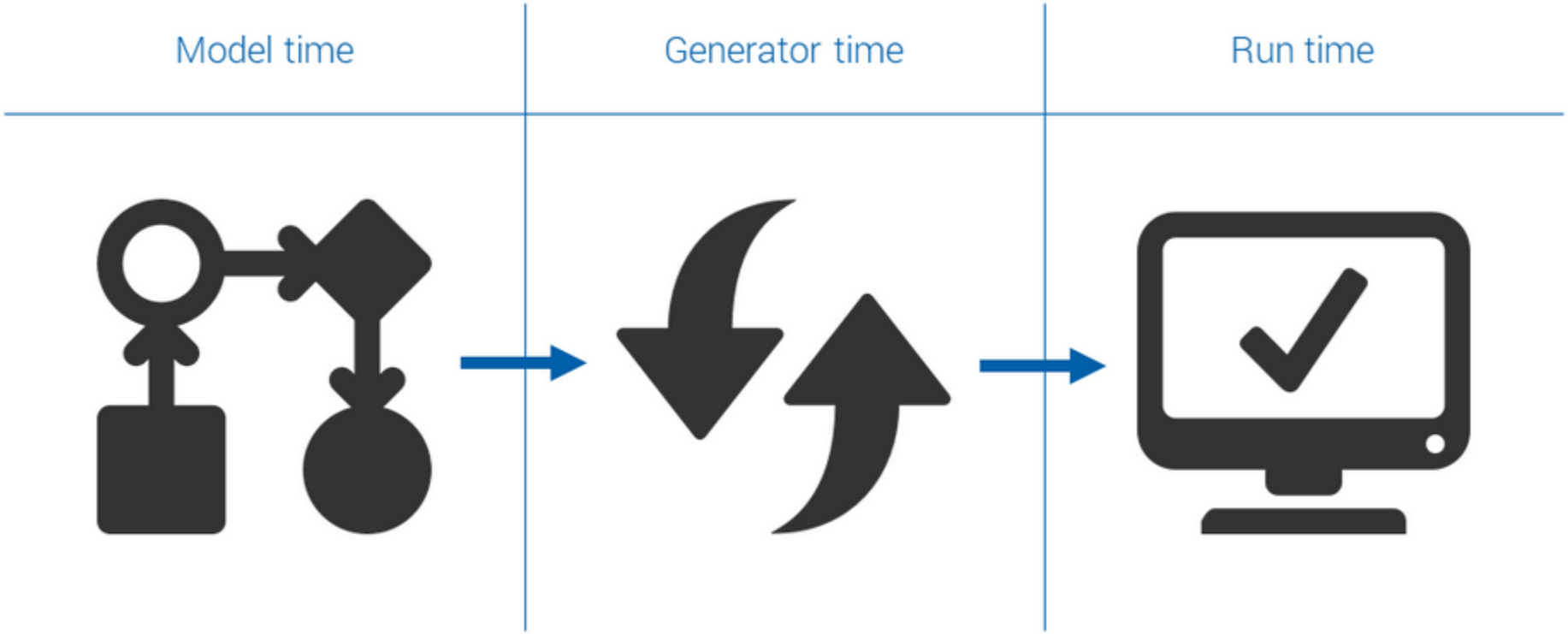
Предметно-ориентированное проектирование (Domain Driven Design, DDD)



FDD – Feature Driven Development



Model Driven Development



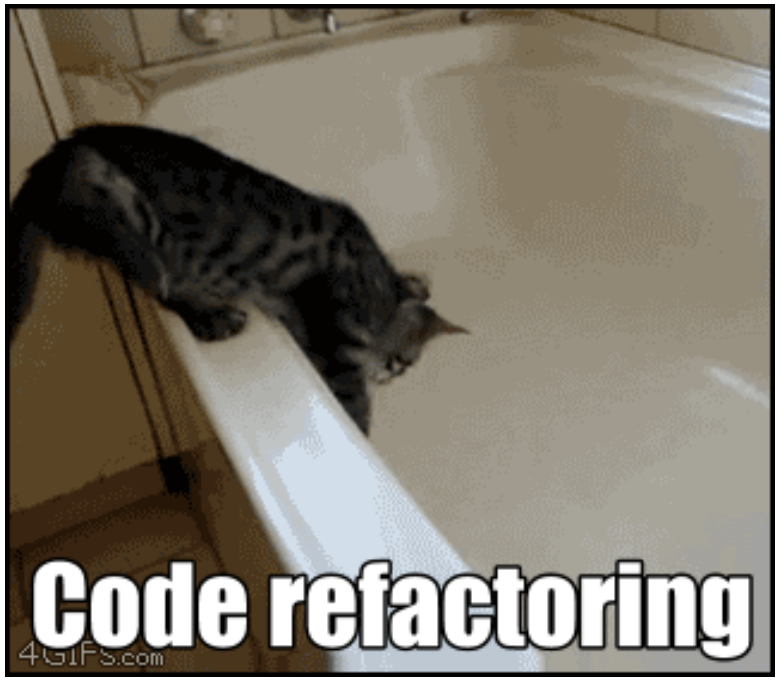
Рефакторинг

Причины:

- Модификация
- Ошибки
- Проблемы с разработкой

Подход:

- Небольшие, эквивалентные преобразования
- TDD/BDD



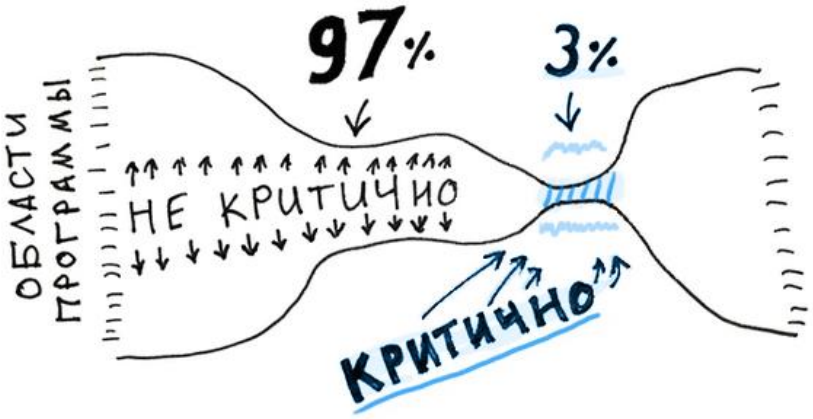
На необходимость рефакторинга указывают некоторые предупреждающие знаки, иногда называемые «запахами» (smells)

(Fowler, 1999).

Оптимизация

Программисты тратят огромное количество времени, размышляя и беспокоясь о некритичных местах кода, и пытаются оптимизировать их, что исключительно негативно сказывается на последующей отладке и поддержке. **Мы должны вообще забыть об оптимизации в, скажем, 97% случаев**; более того, поспешная оптимизация является корнем всех зол. И напротив, мы должны уделить все внимание оставшимся 3%.

Дональд Кнут



Модификация

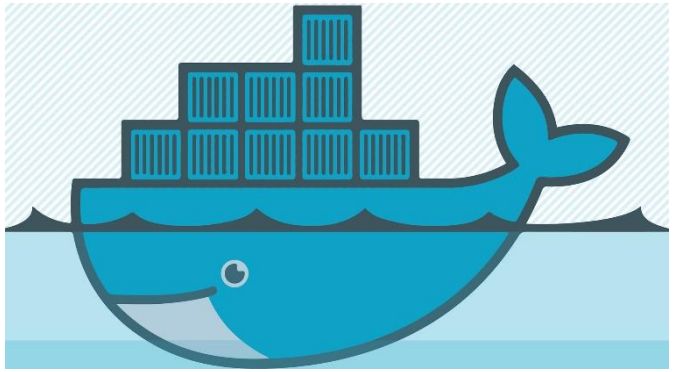
- Есть ли место в архитектуре?
- Как формализовать с тз текущей архитектуры?
- Рефакторинг архитектуры (при необходимости)
- Формализация новой функциональности

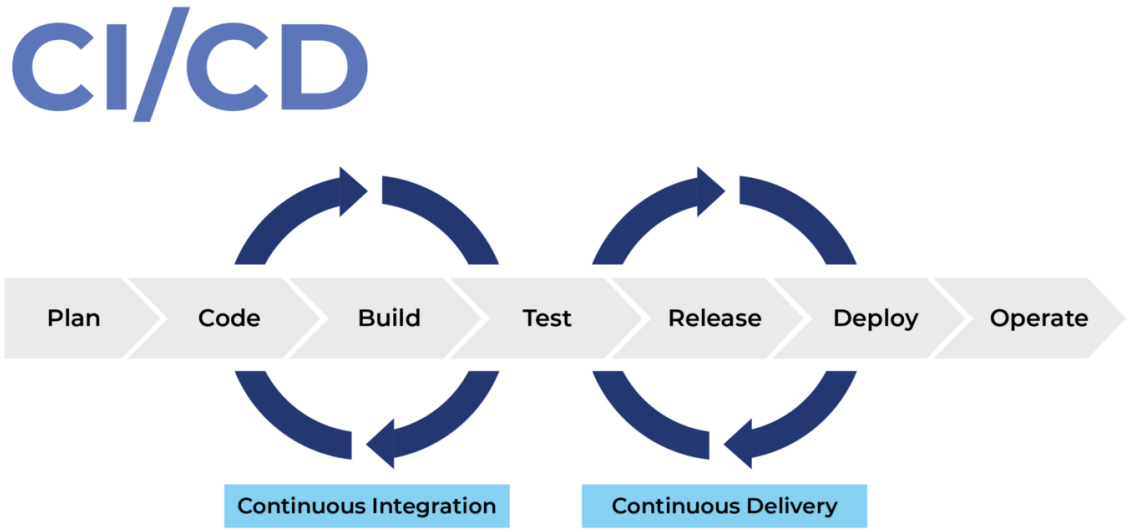
TDD и BDD – лучшие друзья

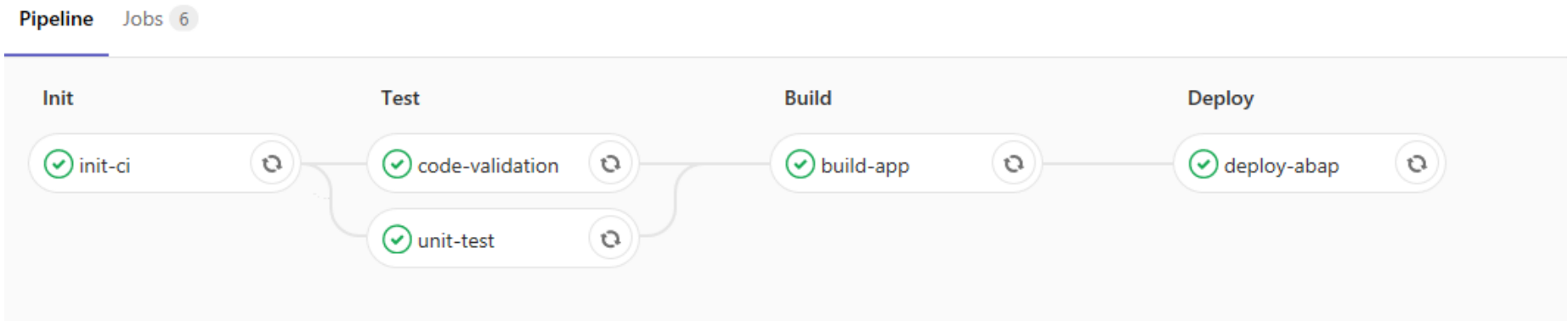
Исправление ошибок

- Написание теста на новую ошибку
- Исправление ошибки
- Надежда на светлое будущее

Развертывание







CI

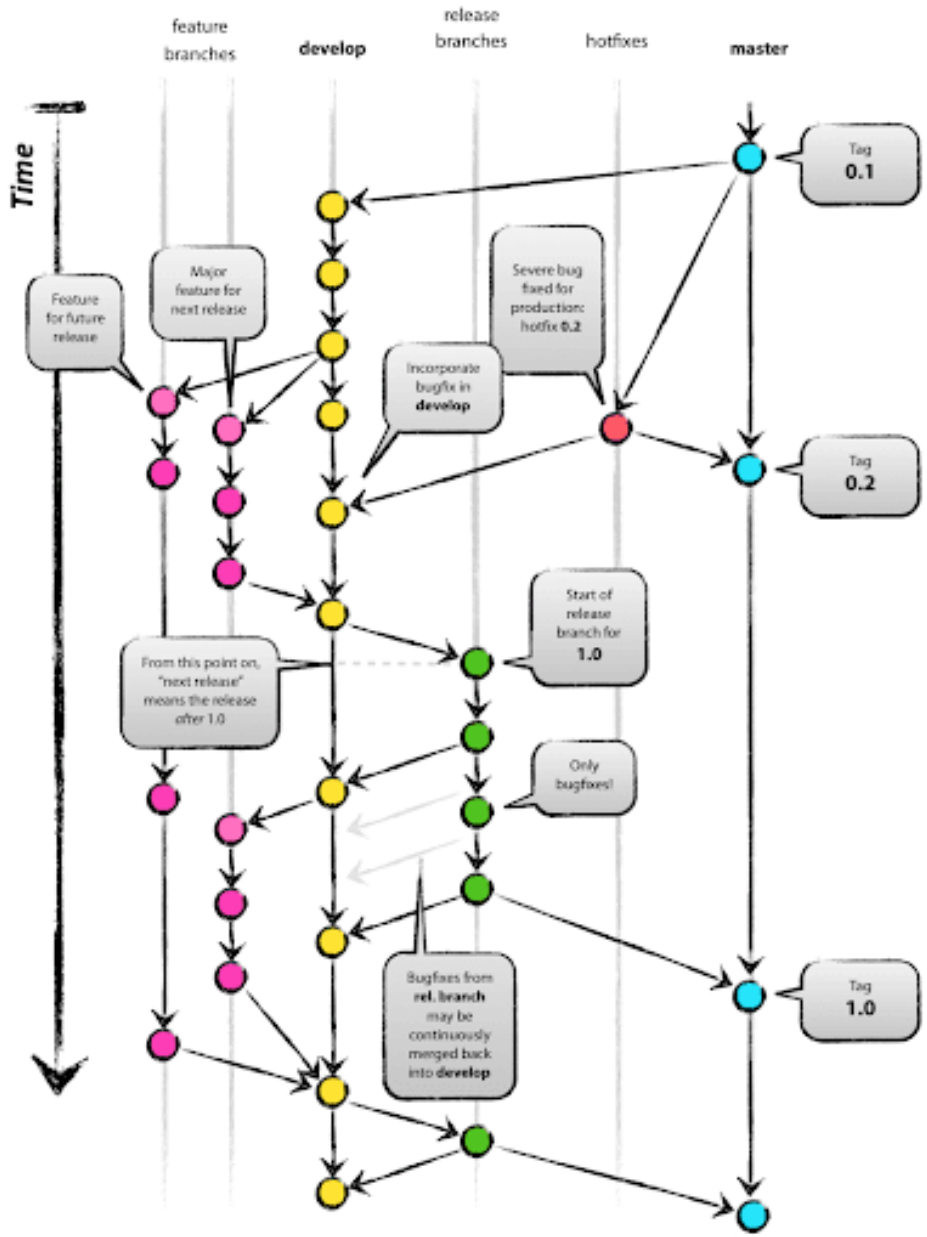
- Сборка каждой ветки по каждому коммиту
- Прогонка Unit тестов в каждой ветке по каждому коммиту
- Прогонка интеграционных и системных тестов – по запросу / автоматом в develop

CD

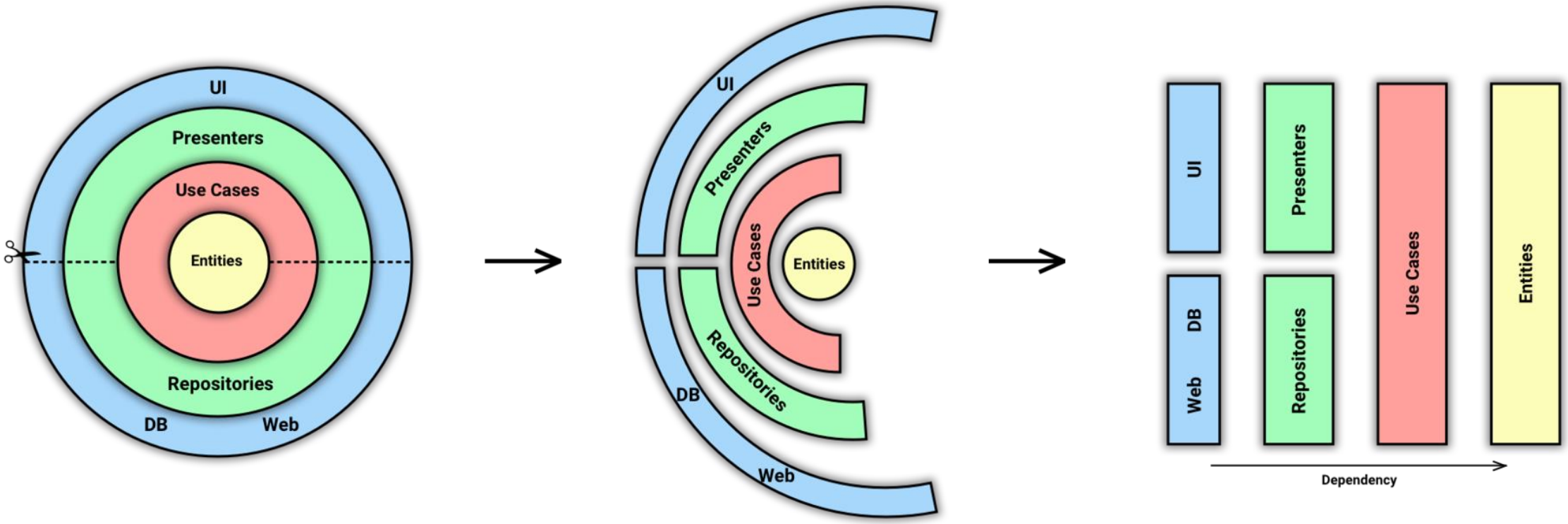
- Сборка релизов
- Доставка (развертывание релизов) на разных средах
  - Dev
    - Feature - среды
  - Staging
  - Production

Git flow

- Develop
- Feature
- Release
- Hotfix
- Master



Архитектура?

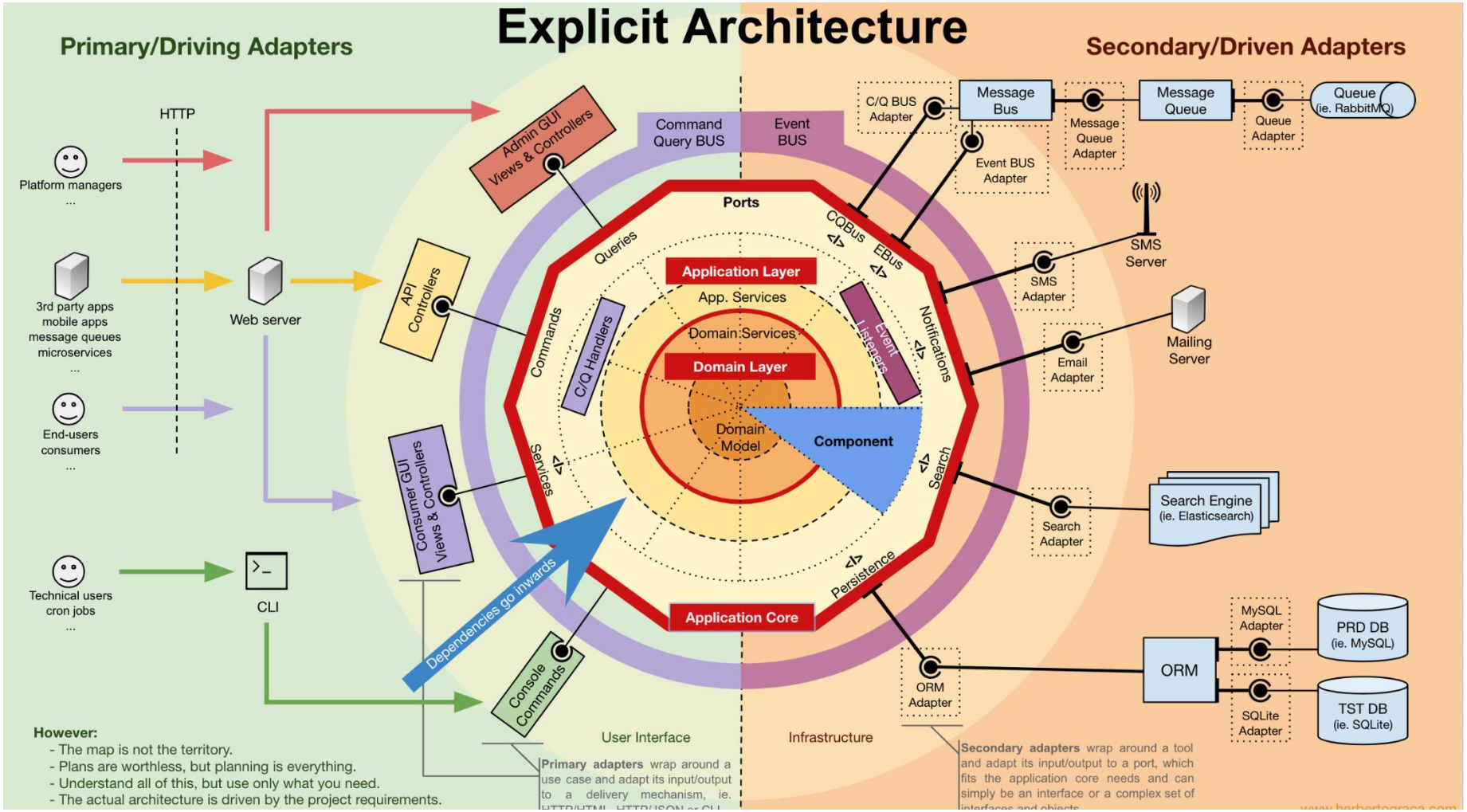


Архитектура?

- Сущности системы
- DTO для разных UI и интеграций
- Сущности БД для хранения

- Слой сущностей
- Слой бизнес-логики и Use-case
- Слой контроллеров и репозиториев
- Слой UI, баз данных, переферии, протоколов

Архитектура?



Архитектура: вчера, сегодня, завтра

- Ничего не меняется
- Новые вызовы – новые решения
- Новые возможности – новые решения

Главный вопрос проектирования:

Как сделать так, чтобы все было хорошо?

Более конкретно:

Что разделять, а что объединять? Где проводить границы?

И еще раз, архитектура

- Компонентная слоёная архитектура
- Разделение
  - Устойчивое и неустойчивое
  - Разные функции
  - Разные акторы и сценарии использования
- Всегда разделяется логика работы, хранение и представление данных
- Приоритет – декларативным описаниям
- Проектирование API
- TDD/BDD
- Документация, журналирование и мониторинг
- CI/CD

Архитектор ПО

- Анализ всех требований
- Построение четкой ментальной картины предметной области и требуемой функциональности
- Формализация предметной области
- Выбор архитектурного подхода
- Верхнеуровневая формализация архитектуры
- Постепенная детализация и декомпозиция до уровня программных компонент
- Постоянный контроль соблюдения выбранного подхода
- Постоянный поиск решений «новых вызовов» - проблем и требований

Компетенции архитектора

- Software Engineer
- System Analyst
- Team Lead/Tech Lead
- Software/System Architect
- Solution Architect
- Enterprise Architect