

Лекция 8. Анализ архитектуры информационных систем

Основные понятия анализа архитектур ИС

1. Что и зачем анализируется в архитектуре?
2. Атрибуты качества
3. Классификация методов анализа
4. Облегченный метод оценки архитектур-альтернатив

Что анализируется в архитектуре?

- Архитектура
 - фундаментальные концепции и свойства системы в ее окружении, выраженные в ее
 - элементах,
 - отношениях и
 - принципах их проектирования и развития

ISO/IEC/ IEEE 42010 - 2011, "Systems and software engineering — Architecture description ."

Что анализируется в архитектуре?

Системные структуры

- Статические структуры определяют
 - внутренние проектные элементы и
 - компоновку (организацию) этих элементов
- Динамические структуры определяют
 - элементы времени выполнения и
 - взаимодействие этих элементов

Проектные элементы

- Программные элементы
 - Модули, классы, хранимые процедуры, любые другие согласованные программные единицы
- Элементы данных
 - Классы, сущности, таблицы, файлы
- Технические элементы
 - Компьютеры и их части, сетевые элементы (кабели, маршрутизаторы, хабы)

Компоновка элементов

- Ассоциации, отношения, связность
 - Для программных модулей
 - Иерархии, зависимости
 - Для классов или реляционных сущностей
 - Связи одних элементов с другими
- Для оборудования
 - Физические соединения

Внутренние взаимодействия

- Потоки информации между элементами
- Параллельное или последовательное исполнение внутренних задач

Внешне проявляемые системные свойства

- Внешне проявляемое поведение
 - Функциональное взаимодействие между системой и ее окружением
 - Модель «черный ящик»
- Качественные характеристики
 - Внешне проявляемые нефункциональные свойства
 - Производительность, безопасность, масштабируемость

Атрибуты качества

- Качество ИС означает, что система
 - успешно справляется со всеми возлагаемыми на нее задачами,
 - имеет хорошие показатели надежности и приемлемую стоимость,
 - удобна в эксплуатации и обслуживании,
 - легко сочетается с другими системами и
 - в случае необходимости может быть модифицирована
- Разные пользователи имеют разные точки зрения на характеристики качества ИС
- Качество ПО определяется стандартом ISO 9126

Модель качества по ISO 9126



Атрибуты качества по ISO 9126-4

- Эффективность
 - Способность ПО предоставлять пользователям возможность решать их задачи с необходимой точностью при использовании в данном контексте
- Продуктивность
 - Способность ПО предоставлять пользователям определенные результаты в рамках ожидаемых затрат ресурсов
- Безопасность
 - Способность ПО обеспечивать необходимый низкий уровень риска нанесения ущерба жизни и здоровью людей, бизнесу, собственности и окружающей среде
- Удовлетворение пользователей
 - Способность ПО приносить удовлетворение пользователям при использовании в заданном контексте

Функциональность

- Functionality
- Способность ПО в определенных условиях решать задачи, нужные пользователям
- Функциональная пригодность (suitability)
 - Способность решать нужный набор задач
- Точность (accuracy)
 - Способность выдавать нужный результат
- Способность к взаимодействию (interoperability)
 - Способность взаимодействовать с нужным набором других систем
- Соответствие стандартам и правилам (compliance)
 - Соответствие отраслевым стандартам, нормативным и законодательным актам, другим регулирующим нормам
- Защищенность (security)
 - Способность предотвращать неавторизованный и неразрешенный доступ к данным и программам

Надежность

- Reliability
- Способность поддерживать определенную работоспособность в заданных условиях
- Зрелость, завершенность (maturity)
 - Величина, обратная частоте отказов
- Устойчивость к отказам (fault tolerance)
 - Способность поддерживать заданный уровень работоспособности при отказах и нарушениях правил взаимодействия с окружением
- Способность к восстановлению (recoverability)
 - Способность восстанавливать определенный уровень работоспособности и целостность данных после отказа

Производительность

- Efficiency
- Способность при заданных условиях обеспечивать необходимую работоспособность по отношению к выделяемым для этого ресурсам
- Временная эффективность (time behavior)
 - Способность выдавать ожидаемые результаты, обеспечивать передачу необходимого объема данных за отведенное время
- Эффективность использования ресурсов (resource utilisation)
 - Способность решать нужные задачи с использованием определенных объемов ресурсов определенных видов

Удобство использования

- Usability
- Способность быть удобным в обучении и использовании, а также привлекательным для пользователей
- Понятность (understandability)
 - Показатель, обратный усилиям, которые затрачиваются пользователями на восприятие основных понятий и осознание их применимости для решения своих задач
- Удобство работы (operability)
 - Показатель, обратный усилиям, предпринимаемым пользователями для решения своих задач с помощью ПО
- Удобство обучения (learnability)
 - Показатель, обратный усилиям, затрачиваемым пользователями на обучение работе
- Привлекательность (attractiveness)
 - Способность быть привлекательным для пользователей

Удобство сопровождения

- Maintainability
- Удобство проведения всех видов деятельности, связанных с сопровождением программа
- Анализируемость (analyzability)
 - Удобство проведения анализа ошибок, дефектов, недостатков, необходимых изменения и их последствий
- Удобство внесение изменений (changeability)
 - Показатель, обратный трудозатратам на выполнение необходимых изменений
- Стабильность (stability)
 - Показатель, обратный риску возникновения неожиданных эффектов при внесении необходимых изменений
- Удобство проверки (testability)
 - Показатель, обратный трудозатратам на проведение тестирования и других видов проверки того, что внесенные изменения привели к нужным результатам

Переносимость

- Portability
- Способность ПО сохранять работоспособность при переносе из одного окружения в другое, включая организационные, аппаратные и программные аспекты окружения
- Адаптируемость (adaptability)
 - Способность приспосабливаться к различным окружениями без проведения для этого действия помимо заранее предусмотренных
- Удобство установки (installability)
 - Способность ПО быть установленным или развернутым в определенном окружении
- Способность к сосуществованию (coexistence)
 - Способность сосуществовать с другими программами в общем окружении, деля с ними ресурсы
- Удобство замены (replaceability)
 - Возможность применения данного ПО вместо другого для решения тех же задач в определенном окружении

Атрибуты качества системы

- Availability (Доступность)
- Modifiability (Модифицируемость)
- Performance (Производительность)
- Security (Безопасность)
- Testability (Тестируемость)
- Usability (Практичность)

Коммерческие атрибуты

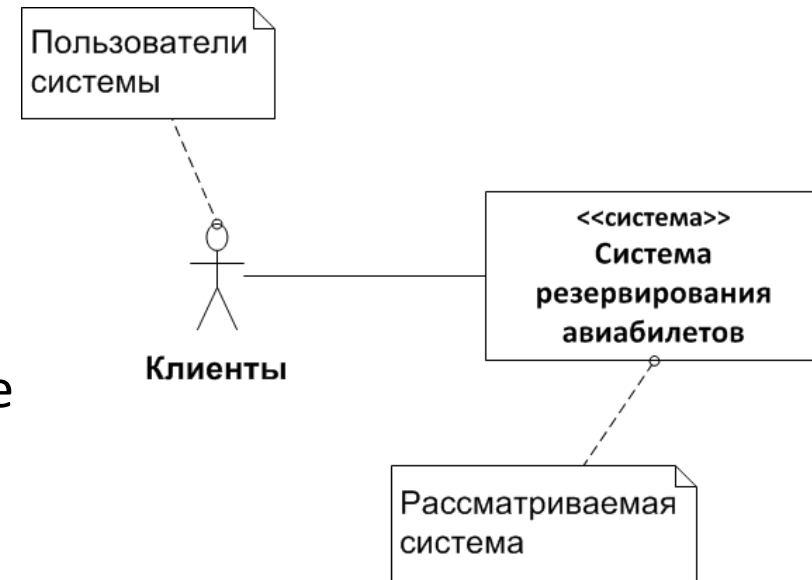
- Time (Сроки выхода на рынок)
- Cost (Стоимость и прибыль)
- Life Time (Срок службы системы)
- Target market (Целевой рынок)
- Product Schedule (График развертывания продукта)
- Interoperability (Интеграция с существующими системами)

Атрибуты качества архитектуры

- Integrity (Целостность)
- Portability (переносимость)
- Reusability (Возможность повторного использования)
- Flexibility (Гибкость)
- Reliability (Надежность)
- Robustness (Живучесть)

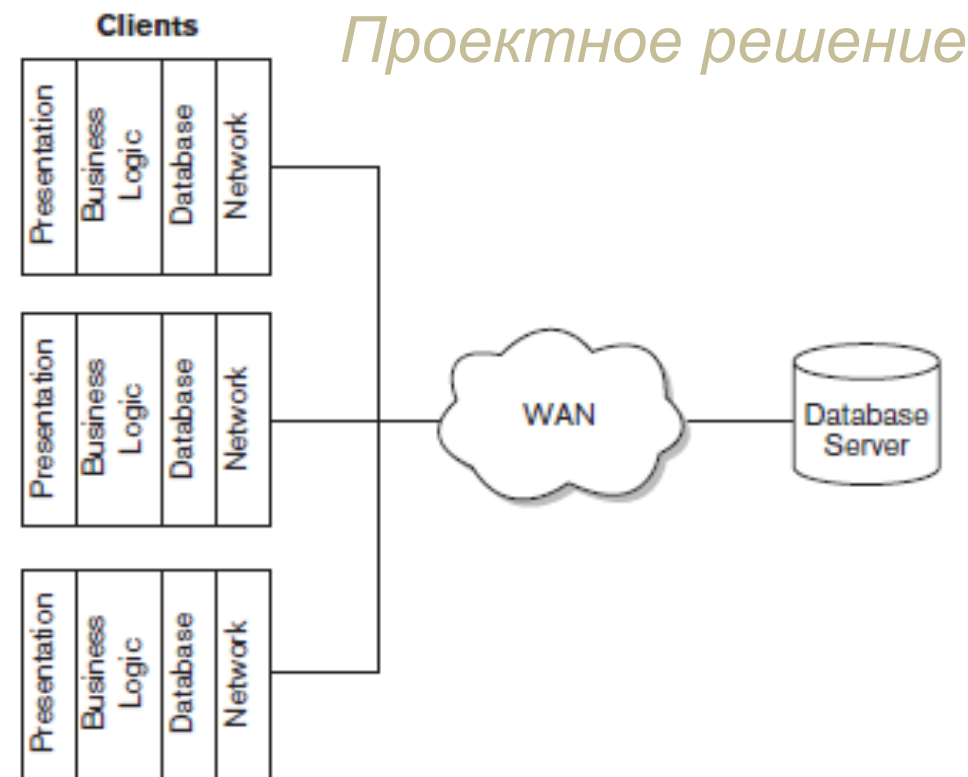
Пример. Система резервирования авиабилетов

- Назначение
 - Поддержка различных операций по заказу, бронированию авиабилетов, обновлению заказа или отмены его, оплаты авиабилета и т.п.
- Внешне проявляемое поведение
 - Реакции на операции, инициируемые клиентами
 - Бронирование места
 - Обновление резервирования
 - Отмена заказа
- Качественные характеристики
 - Среднее время отклика операции при заданной нагрузке
 - Максимальная пропускная система
 - Доступность системы
 - Время устранение дефектов (проблемы)



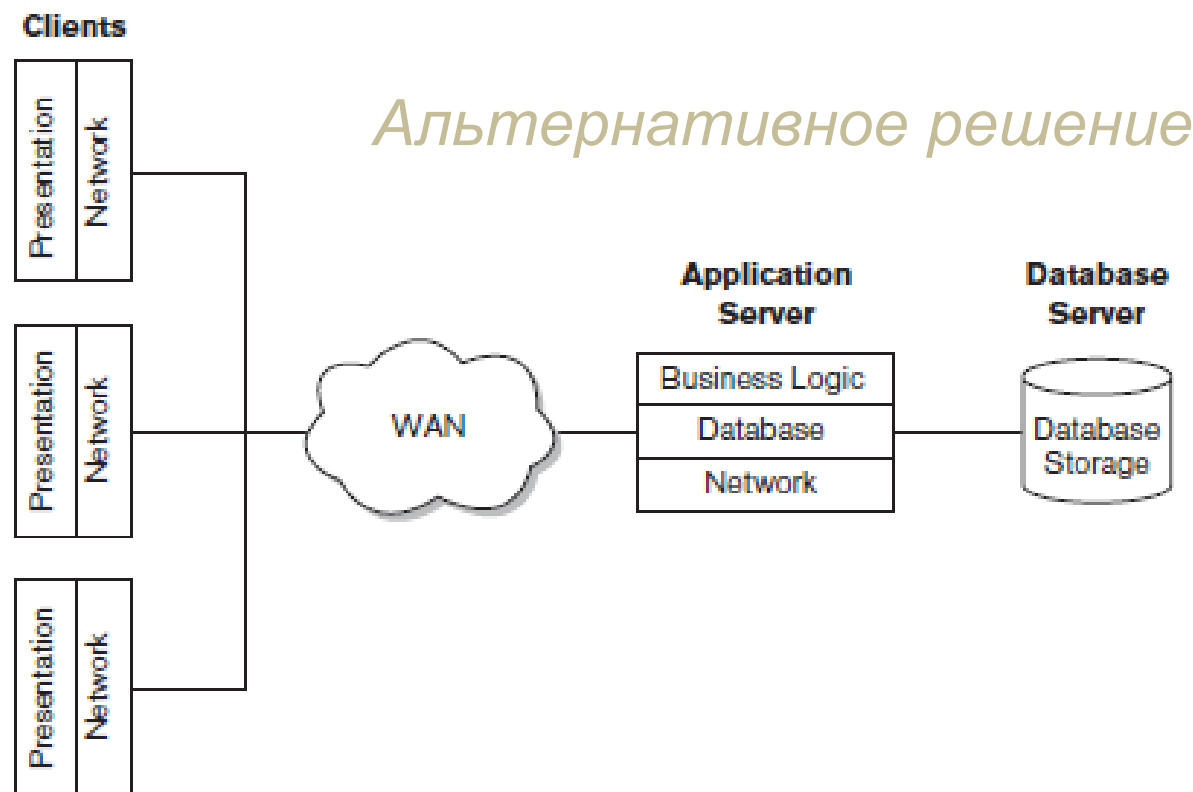
Пример. Система резервирования авиабилетов

- Статическая структура
 - Клиентская программа
 - Сервер
 - Соединения
- Динамическая структура
 - Модель «запрос/ответ»
 - Запрос идет от клиента к серверу через сеть
 - Ответ возвращается от сервера к клиенту через сеть
- Особенности
 - Относительная функциональная простота
 - Меньшее время реализации
 - Меньшая стоимость



Пример. Система резервирования авиабилетов

- Статическая структура
 - Клиентская программа
 - Сервер приложения
 - Сервер баз данных
 - Соединения
- Динамическая структура
 - Трехуровневая модель «запрос/ответ»
 - Запрос от клиента идет к серверу приложения, сервер приложения передает запрос серверу баз данных
 - Ответ от сервера баз данных получает сервер приложения, и если необходимо пересылает его клиенту
- Особенности
 - Лучшая масштабируемость при возрастании нагрузки
 - Меньшие требования к клиентской машине
 - Лучшая безопасность



Анализ архитектуры

- критически важная задача
 - позволяет сократить затраты на исправление ошибок
 - как можно раньше выявить и исправить возможные проблемы
- выполняется часто
 - в конце основных этапов проекта
 - при существенных изменениях в архитектуре

Архитектура
-кандидат

- Частный способ организации статической и динамической структур, который может потенциально отобразить внешне проявляемое поведение и обеспечить качественные характеристики системы

Цель анализа архитектуры

подтверждение применимости базовой архитектуры и ее вариантов

проверка соответствия предлагаемых технических решений функциональным требованиям и атрибутам качества

обнаружение проблем и областей, требующих доработки

Классификация методов анализа

Методы оценки архитектуры, основанной на сценариях

Методы оценки архитектуры, основанной на атрибутах качества

Другие

Оценки на основе сценариев

- Оценки на основании сценариев – это мощный метод анализа дизайна архитектуры. При такой оценке основное внимание направлено на наиболее важные с точки зрения бизнеса и имеющие наибольшее влияние на архитектуру сценарии.
- Сценарии вариантов использования
- Сценарии развития
 - Учет ожидаемых изменений в системе
- Исследовательские сценарии
 - Сценарии «стрессовых» воздействия на систему

Сценарии вариантов использования

1. Существует радикальная корректировка курса во время выпуска оружия (например, лофт), которую программное обеспечение вычисляет за 100 мс. (производительность)
2. Пользователь хочет изучить планируемые и фактические данные за разные финансовые годы без повторного ввода данных по проекту. (удобство использования)
3. Возникает исключение в данных, и система уведомляет об этом определенный список получателей по электронной почте и отображает нарушающие условия красным цветом на экранах данных. (надежность)
4. Пользователь меняет расположение графика с горизонтального на вертикальное, и график перерисовывается за одну секунду. (производительность)
5. Удаленный пользователь запрашивает отчет базы данных через Интернет в пиковый период и получает его в течение пяти секунд. (производительность)
6. Система кэширования будет переключена на другой процессор при отказе своего процессора и сделает это в течение одной секунды. (надежность)

Сценарии развития

1. Изменить основной дисплей, чтобы отслеживать несколько целей одновременно без влияния на задержку.
2. Добавить новый тип сообщения в репертуар системы менее чем за неделю работы.
3. Добавить возможность совместного планирования, при котором два специалиста по планированию, находящиеся в разных местах, совместно разрабатывают план, менее чем за один человеко-год работы.
4. Максимальное количество дорожек, обрабатываемых системой, удваивается, а максимальная задержка данных о дорожках на экране не превышает 200 мс.
5. Миграция на новую операционную систему или новый релиз существующей операционной системы менее чем за человеко-год работы.
6. Добавить новый сервер данных, чтобы снизить задержку в сценарии 5 до 2,5 секунд в течение одной человеко-недели.
7. Удвоить размер существующих таблиц базы данных при сохранении среднего времени поиска в 1 секунду

Исследовательские сценарии

1. Добавить новую функцию трехмерной карты и интерфейс виртуальной реальности для просмотра карт менее чем за пять человеко-месяцев.
2. Сменить базовую платформу Unix на Macintosh.
3. Повторно использовать программное обеспечение 25-летней давности на новом поколении самолетов.
4. Бюджет времени на отображение измененных данных трека сокращается в 10 раз.
5. Повысить доступность системы с 98% до 99,999%.
6. Половина серверов выходит из строя во время нормальной работы, не влияя на общую доступность системы.
7. Десятикратное увеличение количества заявок, обрабатываемых ежечасно, при сохранении наихудшего времени отклика менее 10 секунд.

Методы анализа архитектуры



- Изначально SAAM создавался для оценки модифицируемости, но позже был расширен для анализа архитектуры относительно показателей качества:

- Модифицируемость
- Портруемость
- Расширяемость
- Интегрируемость
- Функциональный охват

Software Architecture Analysis Method

SAAM анализ архитектуры ПО



- ATAM – это доработанная и улучшенная версия SAAM, которая позволяет пересматривать архитектурные решения относительно требований параметров качества и того, насколько хорошо эти решения отвечают конкретным целевым показателям качества.

Architecture Tradeoff Analysis Method

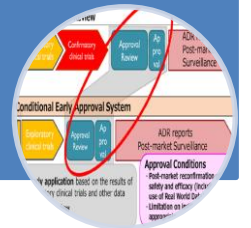
ATAM анализ архитектурных компромиссов



- ADR больше всего подходит для незавершенных архитектур или архитектур, находящихся в процессе разработки. Основное отличие этого метода в том, что анализ более сфокусирован

- На наборе проблем
- На отдельных разделах
- На архитектуру в процессе разработки

Active Design Review ADR активный анализ конструкции



- ARID сочетает в себе подход ADR анализа архитектуры, находящейся в процессе разработки, с фокусом на наборе проблем и подход методов ATAM и SAAM анализа на основании сценария с основным вниманием на параметрах качества.

Active Reviews of Intermediate Designs

ARID активный анализ промежуточных конструкций



- Метод CBAM основное внимание уделяет анализу
 - затрат
 - выгоды
- Планирование
 - последствий архитектурных решений

Cost Benefit Analysis Method

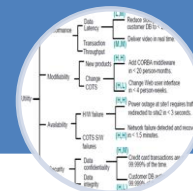
CBAM анализ рентабельности



- ADR больше всего подходит для незавершенных архитектур или архитектур, находящихся в процессе разработки. Основное отличие этого метода в том, что анализ более сфокусирован
- На наборе проблем
- На отдельных разделах
- На архитектуру в процессе разработки

Architecture Level Modifiability Analysis

ALMA анализ модифицируемости на уровне архитектуры



- Оценка семейства архитектур ИС с т.зр.
 - Взаимодействия
 - Расширяемости

Family Architecture Assessment Method FAAM оценка семейства архитектур



Облегченный метод оценки архитектур- альтернатив

- Lightweight Architecture
Alternative Assessment Method -
облегченная версия ATAM,
разработка SEI

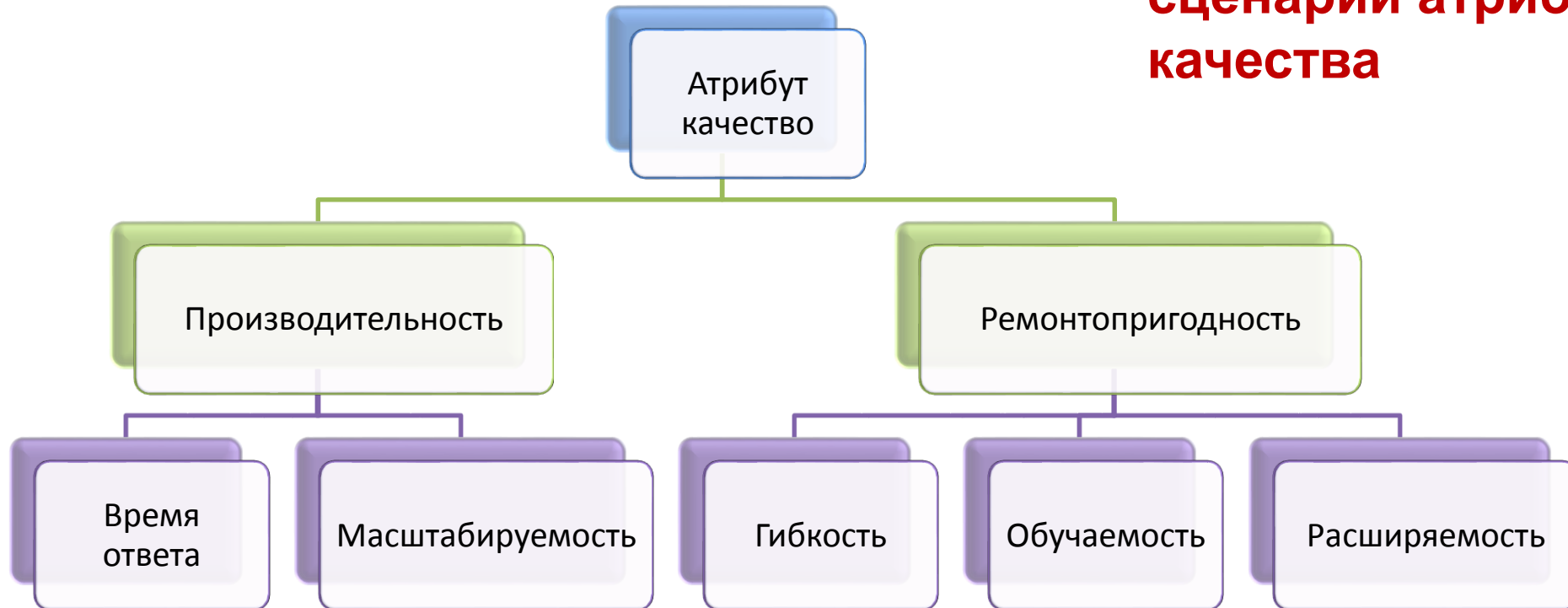
Идея метода



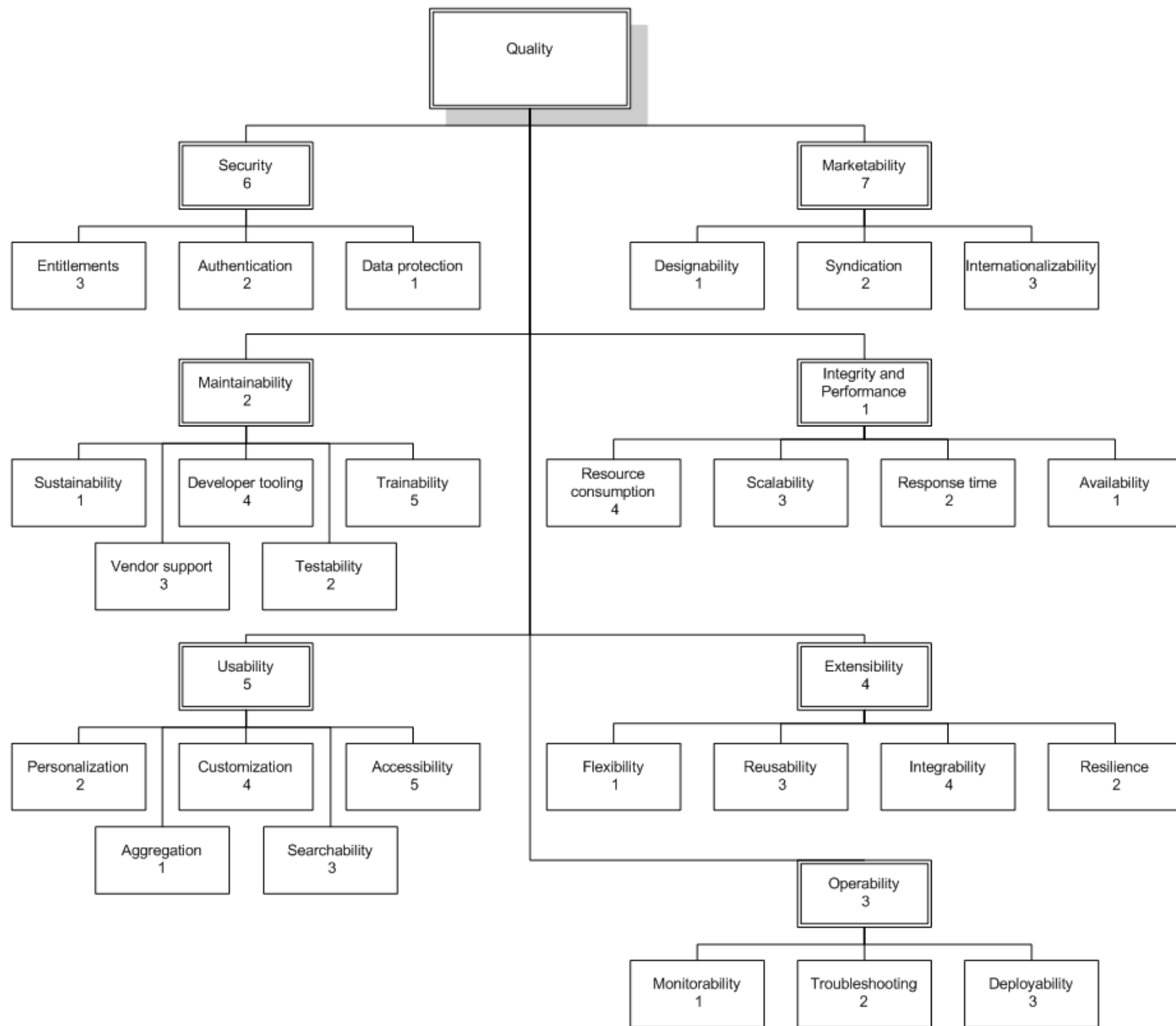
- ☐ Построить дерево атрибутов качества
- ☐ Ранжировать каждый узел
- ☐ Представить альтернативные архитектуры
- ☐ Оценить каждую альтернативу / сценарий
- ☐ Провести вычисление оценок
- ☐ Проанализировать результаты

Дерево атрибутов качества

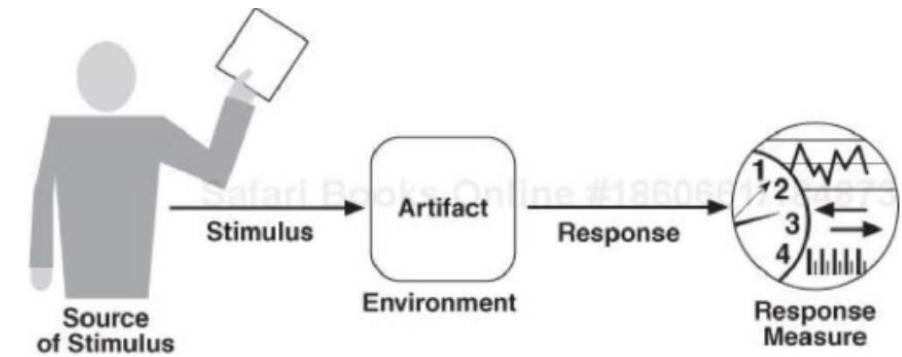
**Листья дерева:
сценарии атрибутов
качества**



Пример дерева атрибутов



Сценарий атрибута качества



- Источник воздействия
 - Субъект (человек, компьютерная система или объект, обладающий поведением), генерирующий воздействие
- Воздействие
 - Внутренний или внешний фактор, вызывающий реакцию системы
- Окружение
 - Условия , в которых происходит наблюдаемое воздействие. Система может быть в рабочем режиме, работать с перегрузкой.
- Артефакт
 - То, на что оказывается воздействие. Совокупность систем, вся система, часть системы
- Реакция
 - Ответ на пришедшее воздействие
- Мера реакции
 - Когда происходит реакция, она должна быть измерена, чтобы сравнить с исходными требованиями

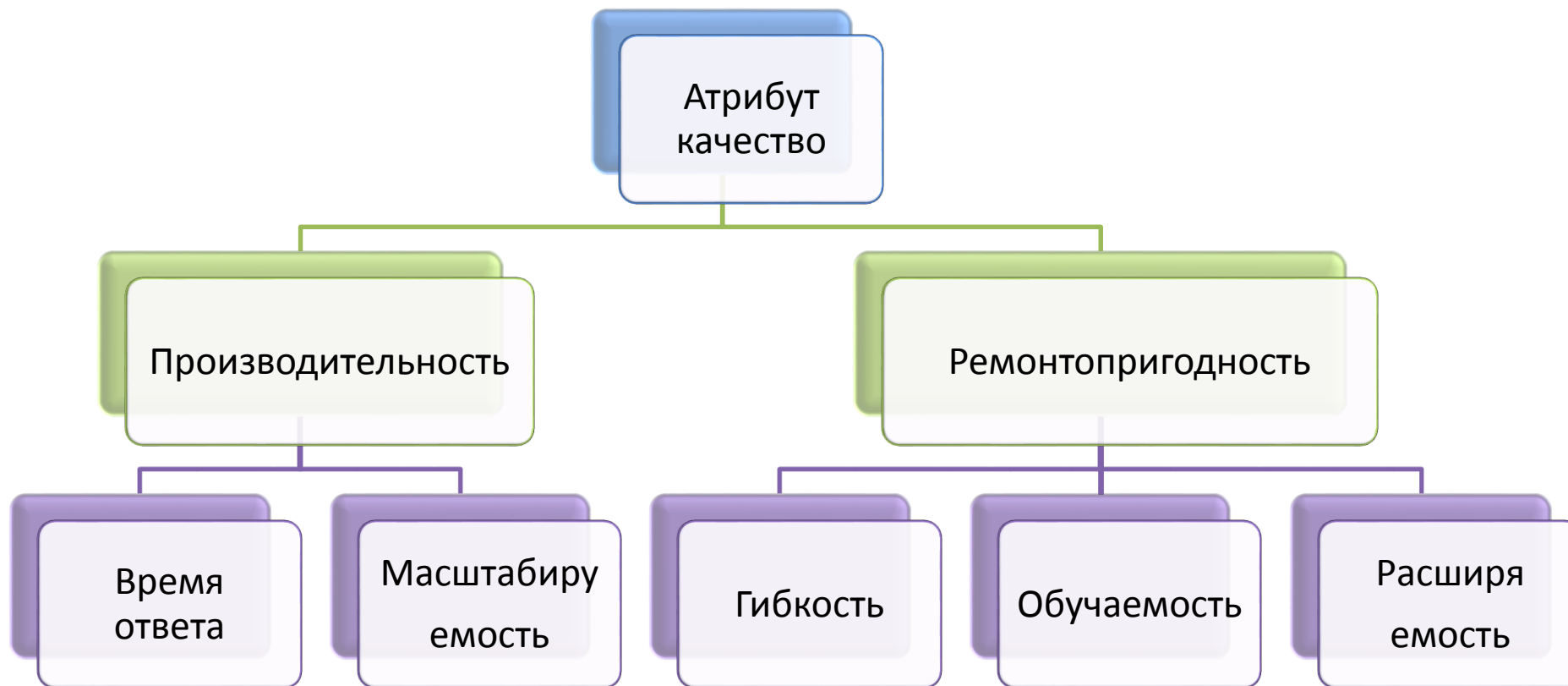
Выбор сценариев

- Формирование требований в виде сценариев использования
 - Воздействие – реакция на воздействие
- Условие: каждому требованию - как минимум один сценарий
- Сценарии выбирают так, чтобы выявить расхождения в решениях

Сценарии

- Для нового релиза, встроить реализацию нового компонента в течение 2 дней
- Сеть восстанавливается в нормальное состояние, полная синхронизация базы данных происходит в течение 30 минут
- В стационарном состоянии, обновление статуса почтового ящика занимает не более 100мс
- В пиковую нагрузку, обновление статуса почтового ящика занимает не более 500мс

Сценарии на дереве качества

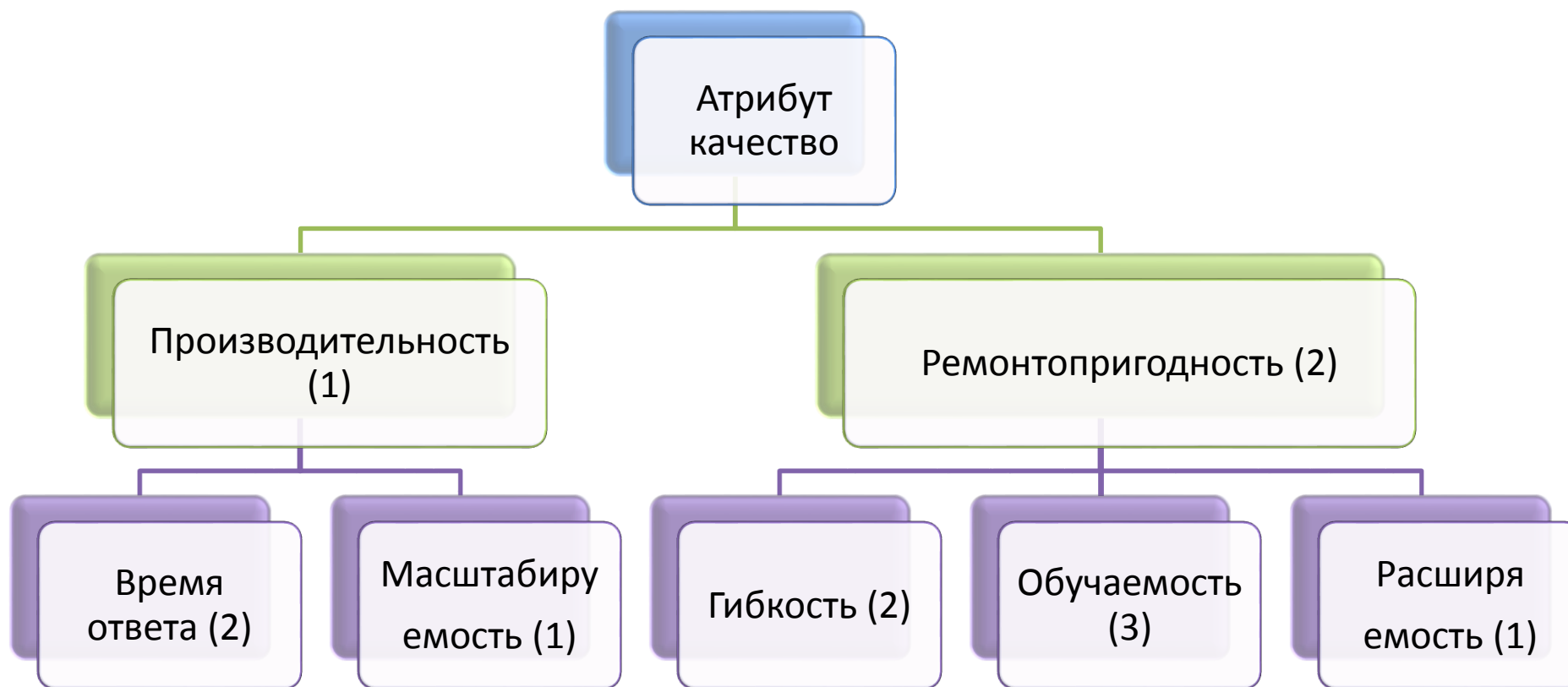


- В стационарном состоянии, обновление статуса почтового ящика занимает не более 100мс
- В пиковую нагрузку, обновление статуса почтового ящика занимает не более 500мс

Проводим приоритезацию узлов дерева

- На основе **субъективных** оценок приоритетов (рангов) и **субъективных** оценок соответствия решения заявленному требованию
- Выставляем приоритет:
 - 1 – самый приоритетный
 - 2 – менее приоритетный
 - 3 – еще менее приоритетный
 - 4 – и т.д.
- Значение ранга зависит от числа узлов на уровне

Пример ранжирование



- (1) В стационарном состоянии, обновление статуса почтового ящика занимает не более 100мс
- (2) В пиковую нагрузку, обновление статуса почтового ящика занимает не более 500мс

Пример вычисления

Формула веса:

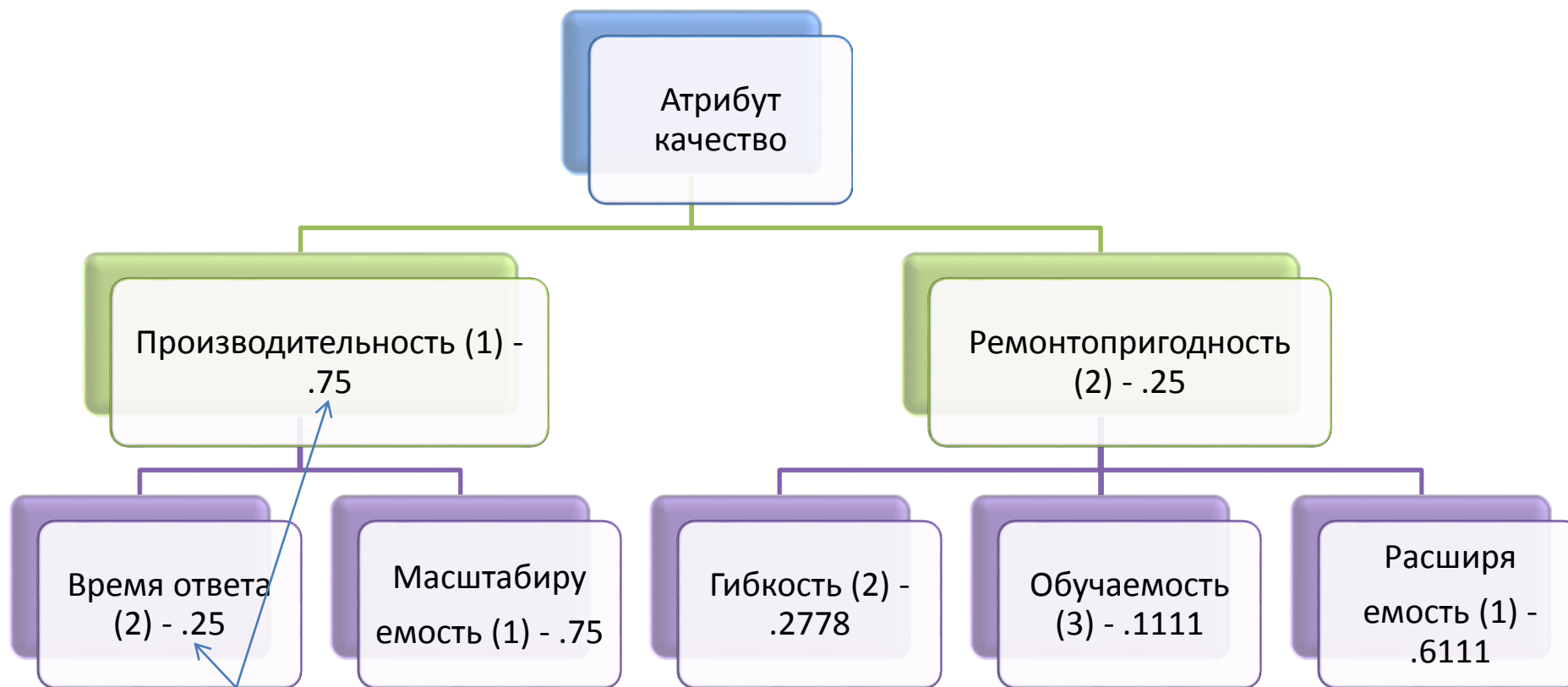
$$f(k) = \sum_{n=k}^N \left(\frac{1}{n * N} \right)$$

- Где
 - k – приоритет атрибута
 - N – количество сравниваемых атрибутов
- Сумма весов всегда равна 1

Таблица весов рангов

Значения					
Число вариантов	Приоритет (Rank)				
	1	2	3	4	5
1	1				
2	.75	.25			
3	.61111	.27778	.11111		
4	.52083	.27083	.14583	.0625	
5	.45667	.25667	.15667	.09	.04

Пример распределения весов



(1) В стационарном состоянии, обновление статуса почтового ящика занимает не более 100мс

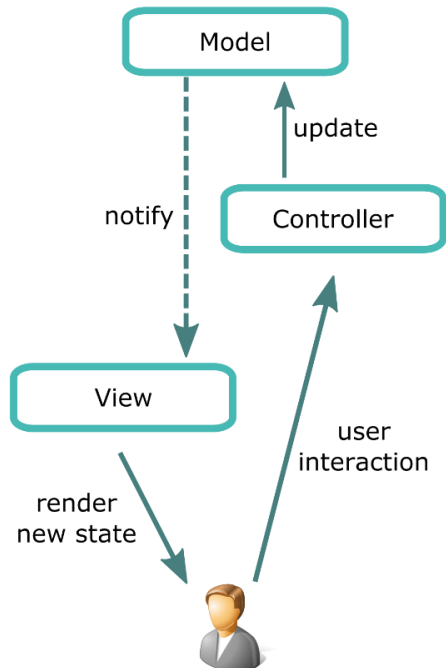
$$W = .75 \times .25 \times .75$$

(2) В пиковую нагрузку, обновление статуса почтового ящика занимает не более 500мс

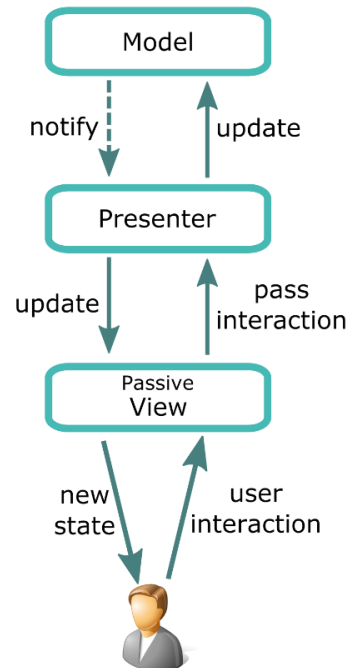
$$W = .75 \times .25 \times .25$$

Представить альтернативные архитектуры

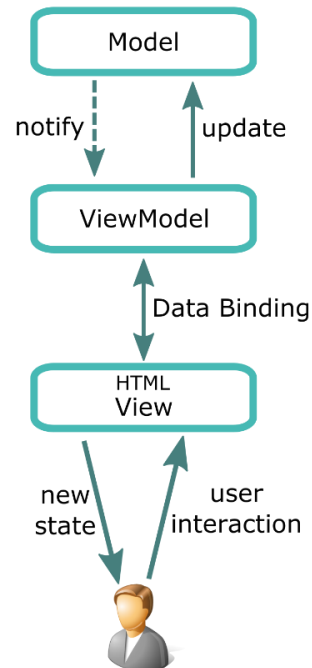
MVC



MVP

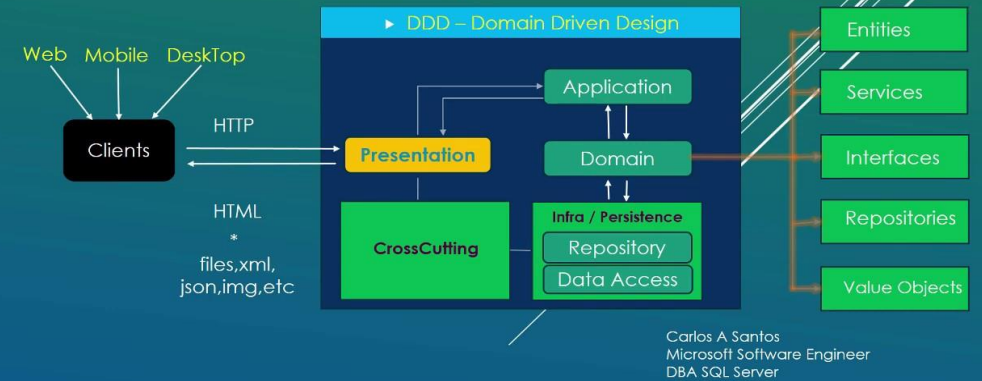


MVVM



ASP.NET CORE MVC

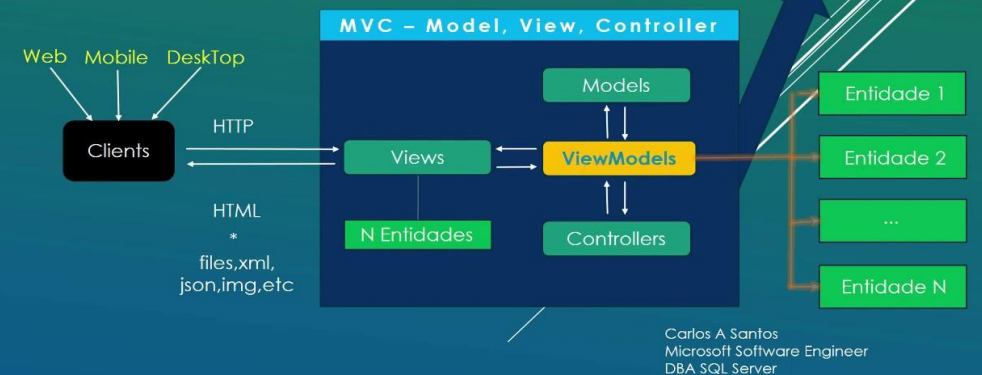
Fluxo Básico | Response e Request HTTP e HTML...



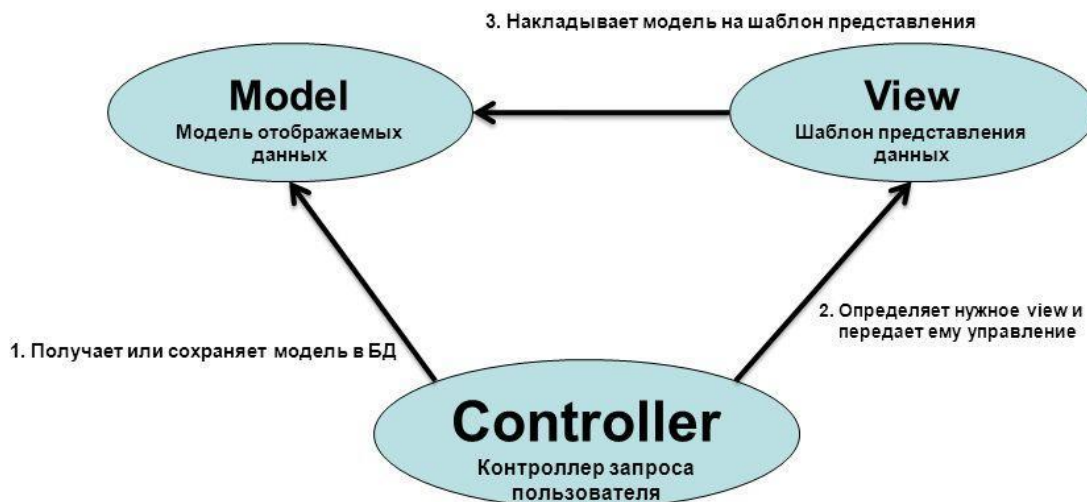
ASP.NET CORE MVC

► MVVM – Model-View-ViewModel

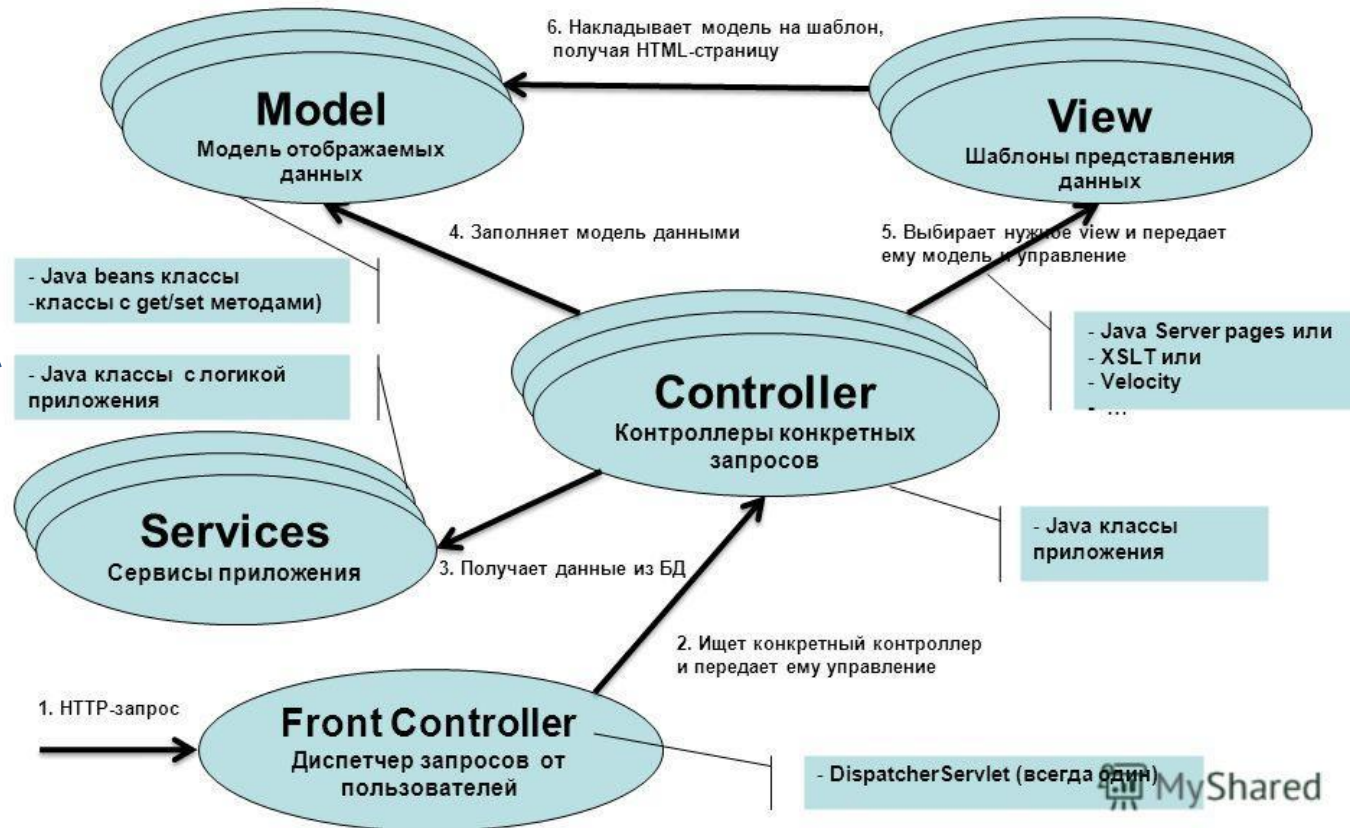
Fluxo Básico | Response e Request HTTP e HTML...



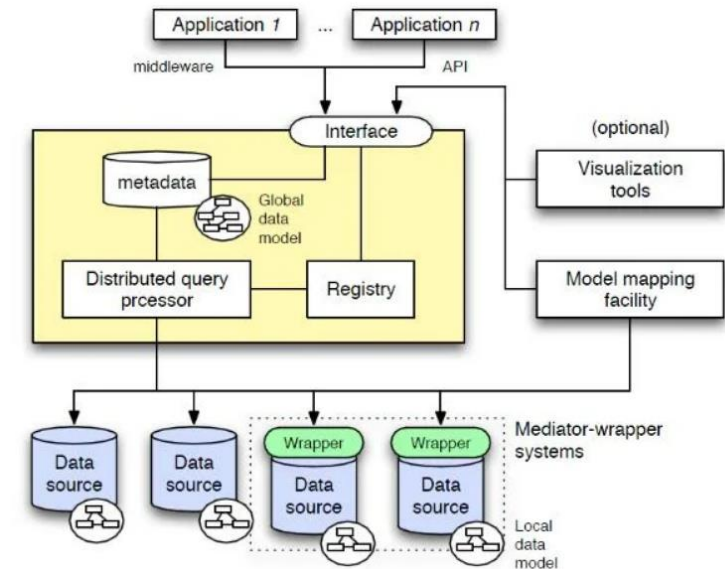
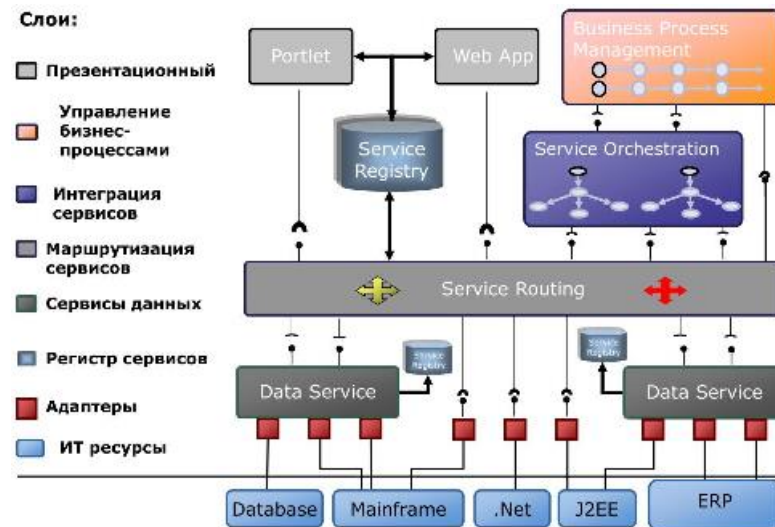
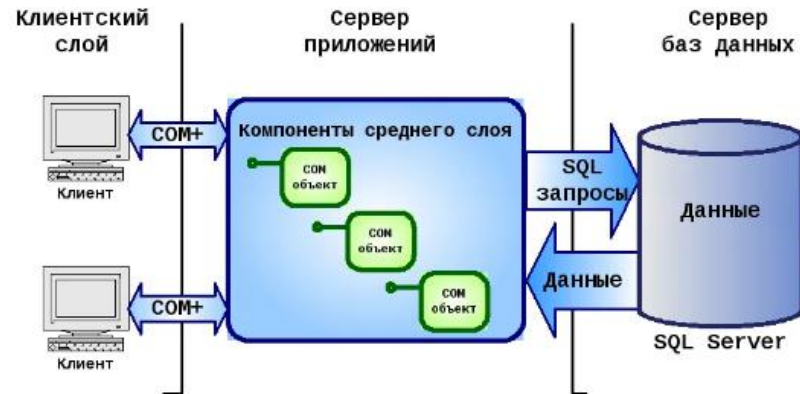
Model View Controller (MVC)



Model View Controller в Web-приложении



Представить альтернативные архитектуры



Оценить альтернативы

- Используем выбранные сценарии и рассчитанные веса
- Выставляем оценку возможности реализовать сценарий в рамках оцениваемой альтернативы
 - 0 – не реализуется
 - 1 – трудно достигим
 - 2 – реализуем, но с ограничениями
 - 3 – полностью соответствует
 - 4 – высшая оценка

Результат оценивания

Матрица оценивания

Сценарий	Weight	Архитектура 1	Архитектура 2	Архитектура 3
1	.140625	0	1	4
2	.046875	3	2	1
Итого		.140625	.234375	.609375

Победитель