ТСПП

23.04.2018

Лекция №10

Тема: Архитектурное проектирование.

После определения системных требований к ПП и составления спецификации требований.

Архитектурное проектирование – это первый этап проектирования, в ходе которого определяется общая структура системы, определяются основные подсистемы, порядок их управления, взаимодействия и выполняется их модульная декомпозиция.

АП, включает в себя условные этапы (\*):

1. Структурирование системы – представляется в виде совокупности относительно самостоятельных подсистем, выбор статической структуры системы.
2. Моделирование управления – выбор базовой модели управления взаимодействиями, между частями системы. Рассматривается динамическая модель системы.
3. Модульная декомпозиция – каждая подсистемы выделенная на 1 этапе разбивается на модули.

Данные этапы чаще всего ведутся параллельно.

Результатом АП является проект(документ отражающий архитектуру системы).

Основные виды моделей:

* Статическая структурная модель – представляет подсистемы или компоненты, которые будут разрабатываться независимо.
* Динамическая модель процессов – отображает структуру процессов во время работы системы.
* Интерфейсная модель – предоставляет сервисы каждой системы, через общедоступный интерфейс.
* Модели отношений – отображает отношения между подсистемами и модулями, потоки данных и потоки управления.

Для построения и демонстрации данных моделей используются средства(UML).

Выбор архитектуры решающим образом влияет на не функциональные характеристики будущего ПП. Ключевые требования:

СКОРОСТЬ РАБОТЫ – за реализацию ключевых сервисов отвечает как можно меньше подсистем, с минимальным взаимодействием между ними.

ЗАЩИЩЕННОСТЬ – рекомендуется использовать многоуровневую архитектуру (вложенную) и наиболее важные сервисы реализовать на нижних уровнях, а проверку осуществлять на верхних.

БЕЗОПАСТНОСТЬ – за операции, влияющие на безопасность системы в целом, должно отвечать как можно меньше подсистем, при этом на разработку таких подсистем уделяется больше времени.

НАДЕЖНОСТЬ (ОТКАЗОУСТОЙЧИВОСТЬ) – ключевые сервисы должны предоставляться с высокой долей вероятности, т.е. архитектура должна подразумевать дублирование и пересечение компонентов, при этом, не влияя на общую работу системы. Использование микросервисной архитектуры.

УДОБСТВО СОПРОВОЖДЕНИЯ – структурирование системы, используя большое количество мелких компонентов.

Требования к архитектуре – часто противоречат друг другу.

Проектирования является итерационным процессом.

\* СТРУКТУРИРОВАНИЕ СИСТЕМЫ.

Система разбивается на несколько подсистем.

Варианты архитектуры системы:

***Вариант репозитория*** (централизованное хранилище) – предполагает, что все данные хранятся централизованно (используется единая модель данных (структура БД)). Непосредственной передачи данных между подсистемами НЕТ.

**«+»:**

* Обеспечивается целостность данных;
* Передача и администрирование данных (копирование, изменение, удаление, резервное копирование) осуществляется централизованно, без отвлечения остальных подсистем;
* Независимость подсистем, возможность их изменения без серьезных последствий, так как отсутствуют непосредственные связи.
* Упрощается расширение системы. Новые подсистемы легко интегрируются, адаптируясб с моделью данных репозитория.

**«–»:**

* Сильная зависимость подсистем от репозитория, поэтому предъявляются повышенные требования к его надежности. При выходе репозитория из строя, возможна остановка всей системы.
* Данные разных подсистем имеют разную степень важности и безопасности, но так как администрирование производится централизованно, к ним применяются одни требования.

***Клиент-серверная*** ***архитектура*** – распределенная архитектура, предполагающая разбиение подсистем на два вида: клиент и сервер.

*Клиенты* - подсистемы могут отправлять данные.

*Серверы* - подсистемы (автономные компоненты), предоставляют сервисы для работы.

*Среда передачи данных* – обеспечивает взаимодействие между клиентами и серверами.

Легко интегрируются серверные подсистемы.

«+»:

* Распределенность системы. Легкая масштабируемость.

«–»:

* Большое внимание уделяется среде передачи данных:
* Низкое быстродействие, так как используется внешняя среда передачи данных.
* Проблемы с надежностью среды передачи данных.

*Решение проблем с ошибками среды передачи данных:*

Выставление таймаутов на отправление, обработку и получение данных.

Использование свойства идемпотентности: отправка запросов без изменения состояния сервера.

***Модель абстрактной машины*** (многоуровневая модель) – каждый из модулей реализует абстрактную машину, со своим набором команд и функций. Каждый следующий уровень строится на основе функций нижележащего уровня.

«+»:

* Переносимость. Для работы приложения на другой платформе достаточно изменить только внутренний слой.
* Надежность. Функционал, требующий наибольшей защищенности, реализуется на внутреннем уровне.
* Обеспечивается слабая связность. Каждый модуль взаимодействует с 1 модулем выше и 1 ниже.

«–»:

* Низкая скорость работы, так как требуется передавать управление на самый низкий уровень.
* Сложность структуры.
* Плохая работа, при совместной работе функций с разных уровней.