

Технологии разработки программного обеспечения

Составитель: Эверстов В.В.

Дата составления: 15/03/2016

Дата модификации: 15/03/2016

Архитектура ПО

- IEEE 1472000 "Порядок описания архитектурных решений преимущественно-программных систем рекомендуемый IEEE"
- **Архитектура** - это фундаментальное устройство системы, воплощенное в ее компонентах, связях между ними, окружения и, руководящих принципах их дизайна и эволюции

Архитектура ПО

- **Архитектура программного обеспечения** — это структура программы или вычислительной системы, которая включает программные компоненты, видимые снаружи свойства этих компонентов, а также отношения между ними.

Архитектура ПО

- **Архитектура** - это организованная структура и связанные с ней функциональные возможности системы. Над архитектурой можно рекурсивно выполнять декомпозицию, которые взаимодействуют с друг с другом через интерфейсы, взаимосвязи и зависимости. Модули, которые взаимодействуют через интерфейсы включают в себя классы, компоненты и подсистемы [UML 1.5]

Архитектура ПО

- Для повышения эффективности в общем случае выгоднее использовать монолитные архитектуры, в которых выделено небольшое число компонентов (в пределе — единственный компонент).
- С другой стороны, для повышения удобства сопровождения, наоборот, лучше разбить систему на большое число отдельных маленьких компонентов, с тем чтобы каждый из них решал свою небольшую, но четко определенную часть общей задачи.
- С третьей стороны, для повышения надежности лучше использовать либо небольшой набор простых компонентов, либо дублирование функций, т.е. сделать несколько компонентов ответственными за решение одной подзадачи.

Архитектура ПО

- Вопросы организации архитектуры программного обеспечения стали складываться в самостоятельную и достаточно обширную дисциплину, уже на сегодняшний день, на фоне развития понимания архитектуры, накоплен целый комплекс подходов, созданы различные систематизированные комплексы методов, практик и инструментов, призванные в той или иной степени формализовать имеющийся в индустрии опыт.

Структуры и точки зрения

- Любая система может рассматриваться с разных точек зрения – например,
 - поведенческой (динамической),
 - структурной (статической),
 - логической (удовлетворение функциональным требованиям),
 - физической (распределенность),
 - реализации (как детали архитектуры представляются в коде) и т.п.
- В результате, мы получаем различные архитектурные представления (view).

Архитектурные стили

- Архитектурный стиль, по своей сути, шаблон проектирования макро-архитектуры - на уровне модулей, "крупноблочного" взгляда.
- Архитектурный стиль – набор ограничений, определяющих семейство архитектур, которые удовлетворяют этим ограничениям.

Стили и модели

- Blackboard
- Клиент-серверная модель (client-server)
- Архитектуры, построенные вокруг базы данных (database-centric architecture)
- Распределенные вычисления (distributed computing)
- Событийная архитектура (event-driven architecture)
- Front end and back end
- Неявные вызовы (implicit invocations)
- Монолитное приложение (monolithic application)
- Peer-to-peer
- Пайпы и фильтры (pipes and filters)
- Plugin
- Representational State Transfer
- Rule evaluation
- Поиск-ориентированная архитектуры
- Сервис-ориентированная архитектура
- Shared nothing architecture
- Software componentry
- Space based architecture
- Структурированная
- Трех-уровневая

Процесс проектирования архитектуры

- Выделение компонентов,
- Определение интерфейсов компонентов,
- Уточнение набора компонентов,
- Достижение нужных свойств.

Выделение компонентов

- Выбирается набор "основных" сценариев использования — наиболее существенных и выполняемых чаще других.
- Исходя из опыта проектировщиков, выбранного архитектурного стиля и требований к переносимости и удобству сопровождения системы определяются компоненты, отвечающие за определенные действия в рамках этих сценариев, т.е. за решение определенных подзадач.
- Каждый сценарий использования системы представляется в виде последовательности обмена сообщениями между полученными компонентами.
- При возникновении дополнительных хорошо выделенных подзадач добавляются новые компоненты, и сценарии уточняются.

Определение интерфейсов КОМПОНЕНТОВ

- Для каждого компонента в результате выделяется его интерфейс — набор сообщений, которые он принимает от других компонентов и посылает им.
- Рассматриваются "неосновные" сценарии, которые так же разбиваются на последовательности обмена сообщениями с использованием, по возможности, уже определенных интерфейсов.
- Если интерфейсы недостаточны, они расширяются.
- Если интерфейс компонента слишком велик, или компонент отвечает за слишком многое, он разбивается на более мелкие.

Уточнение набора компонентов

- Там, где это необходимо в силу требований эффективности или удобства сопровождения, несколько компонентов могут быть объединены в один.
- Там, где это необходимо для удобства сопровождения или надежности, один компонент может быть разделен на несколько.

Анализ архитектуры

- На основе возможных сценариев использования или модификации системы возможен также анализ характеристик архитектуры и оценка ее пригодности для поставленных задач или сравнительный анализ нескольких архитектур.

Атрибуты качества

- Масштабируемость
- Надежность
- Безопасность
- Usability/ Практичность

Анализ архитектуры

- Определить набор сценариев действий пользователей или внешних систем, использующих некоторые возможности, которые могут уже планироваться для реализации в системе или быть новыми. Сценарии должны быть значимы для конкретных заинтересованных лиц, будь то пользователь, разработчик, ответственный за сопровождение, представитель контролирующей организации и пр. Чем полнее набор сценариев, тем выше будет качество анализа. Можно также оценить частоту появления и важность сценариев, возможный ущерб от невозможности их выполнить.

Анализ архитектуры

- Определить архитектуру (или несколько сравниваемых архитектур). Это должно быть сделано в форме, понятной всем участникам оценки.
- Классифицировать сценарии. Для каждого сценария из набора должно быть определено, поддерживается ли он уже данной архитектурой или для его поддержки нужно вносить в нее изменения.

Анализ архитектуры

- Оценить сценарии. Определить, какие из сценариев полностью поддерживаются рассматриваемыми архитектурами.

Анализ архитектуры

- Выявить взаимодействие сценариев. Определить какие компоненты требуется изменять для неподдерживаемых сценариев; если требуется изменять один компонент для поддержки нескольких сценариев — такие сценарии называют взаимодействующими. Нужно оценить смысловые связи между взаимодействующими сценариями.

Анализ архитектуры

- Оценить архитектуру в целом (или сравнить несколько заданных архитектур). Для этого надо использовать оценки важности сценариев и степень их поддержки архитектурой.

Методики описания архитектуры

- Методики, опубликованные аналитическими компаниями, такими как Gartner, Giga Group, META Group и другими;
- Модель Захмана;
- Методика TOGAF;
- Методика POSIX 1003.23, которая основывается на разработках компании.