ПРИНЦИПЫ И КРИТЕРИИ ВЫБОРА АРХИТЕКТУРЫ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Баранов К. А.

Ролич О. Ч. – канд. техн. наук, доцент

Любая программа должна не только хорошо работать, но и быть хорошо организована. Изначально хорошо продуманная и реализованная архитектура станет хорошим помощником в будущем. Понятие хорошей архитектуры не применяется к программам с той или иной сложностью. Любая написанная программа, большая или маленькая, должна быть тщательно продумана архитектурно, чтобы избежать проблем во время реализации. Сложность, как правило, растёт гораздо быстрее размеров программы. И если не позаботиться об этом заранее, то довольно быстро наступает момент, когда ты перестаёшь её контролировать. Правильная архитектура экономит очень много физических и умственных затрат. И даже если речь идёт всего лишь о небольшом проекте все равно вначале очень полезно его спроектировать.

Не существует общепринятого термина «архитектура программного обеспечения». Тем не менее, для большинства разработчиков и так понятно какой код является хорошим, а какой плохим. Хорошая архитектура – это прежде всего выгодная архитектура, делающая процесс разработки и сопровождения программы более простым и эффективным. Программу с хорошей архитектурой легче расширять и изменять, а также тестировать, отлаживать и понимать. То есть, на самом деле можно сформулировать список вполне разумных и универсальных критериев:

1) Эффективность системы. В первую очередь программа, конечно же, должна решать поставленные задачи и хорошо выполнять свои функции, причём в различных условиях. Сюда можно отнести такие характеристики, как надёжность, безопасность, производительность, способность справляться с увеличением нагрузки (масштабируемость) и т.п.

2) Гибкость системы. Любое приложение приходится менять со временем – изменяются требования, добавляются новые. Чем быстрее и удобнее можно внести изменения в существующий функционал, чем меньше проблем и ошибок это вызовет — тем гибче и конкурентоспособнее система. Изменение одного фрагмента системы не должно влиять на её другие фрагменты.

3) Расширяемость системы. Возможность добавлять в систему новые сущности и функции, не нарушая её основной структуры. На начальном этапе в систему имеет смысл закладывать лишь основной и самый необходимый функционал, но при этом архитектура должна позволять легко наращивать дополнительный функционал по мере необходимости. Причём так, чтобы внесение наиболее вероятных изменений требовало наименьших усилии. Приложение следует проектировать так, чтобы изменение его поведения и добавление новой функциональности достигалось бы за счёт написания нового кода (расширения), и при этом не приходилось бы менять уже существующий код. В таком случае появление новых требований не повлечёт за собой модификацию существующей логики, а сможет быть реализовано прежде всего за счёт её расширения.

4) Масштабируемость процесса разработки. Возможность сократить срок разработки за счёт добавления к проекту новых людей. Архитектура должна позволять распараллелить процесс разработки, так чтобы множество людей могли работать над программой одновременно.

5) Возможность повторного использования. Систему желательно проектировать так, чтобы её фрагменты можно было повторно использовать в других системах.

6) Хорошо структурированный, читаемый и понятный код. Сопровождаемость. После написания программы сопровождать программу, как правило, приходится людям, не участвовавшем в её разработке. Поэтому хорошая архитектура должна давать возможность относительно легко и быстро разобраться в системе новым людям. Проект должен быть хорошо структурирован, не содержать дублирования, иметь хорошо оформленный код и желательно документацию. И по возможности в системе лучше применять стандартные, общепринятые решения привычные для программистов [1].

Паттерн – повторимая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Традиционным архитектурным паттерном в разработке является MVC.

Список использованных источников:

[1] Создание архитектуры программы или как проектировать табуретку [Электронный ресурс]. – Режим доступа: http://www.jvsg.com/ru/.