**ATAM**

**Введение**  
Основой любой программной системы является архитектура. Архитектура определяет структуру системы через ее базовые компоненты и их взаимосвязь, а также свойства и поведение, которые подвергаются воздействию внешнего мира. Архитектура программного обеспечения в основном определяется драйверами архитектуры: функциональными требованиями, нефункциональными требованиями и бизнес-ограничениями.  
  
Поскольку очень важно определить правильную архитектуру (из списка конкурирующих архитектур), чтобы запустить успешный проект с надежным программным продуктом, таким образом, мы (как архитектор) должны пройти предварительную проверку, чтобы смягчить риски. Чем раньше мы обнаружим проблему на этапах проектирования, тем лучше для нас (тем меньше затрат на исправление ошибки). Тем не менее, оценка архитектуры может проводиться во многих местах в процессе разработки. Мы оцениваем архитектуру программного обеспечения (архитектурные решения), которая будет разработана до начала проекта. Мы также оцениваем архитектуру устаревших систем перед выполнением модификаций, портирования, обновления и интеграции с другими системами. И, наконец, мы оцениваем приобретение программных систем, чтобы понять основную архитектуру и влияние на организацию.  
  
Существует несколько методов для оценки архитектуры программного обеспечения относительно недорогим способом:

* **ATAM**: метод анализа компромиссов архитектуры
* **SAAM**: метод анализа архитектуры программного обеспечения
* **ARID**: активные обзоры для промежуточных проектов

Общим у этих методов является то, что они представляют собой методы опроса, в которых используются сценарии и оценка атрибутов качества как способ задавать зондирующие вопросы о том, как гипотетическая архитектура реагирует на эти сценарии. Другие методы опроса включают контрольный список и вопросники.  
  
При оценке архитектуры создается отчет об оценке, который проверяет, что выбранная архитектура «подходит» для программной системы, и предоставляет список рисков, связанных с архитектурными решениями, которые необходимо смягчить с помощью дальнейшего анализа и проектирования, создания прототипов и т. Д. Теперь давайте объясним понятие «пригодности». с некоторыми примерами. Оценивая архитектуру по конкурирующим гипотетическим архитектурам, мы сначала определяем наиболее важные цели, а затем выделяем слабые и сильные стороны каждого кандидата. После принятия решения у нас может быть выбранная архитектура. Иногда выбранная архитектура «подходит» для некоторых целей и проблематична для других целей, в этом случае нам необходимо определить приоритеты бизнес-целей и включить в отчет слабые и сильные стороны архитектуры. Иногда мы выбираем подходящую архитектуру, а иногда ни одну из архитектур не выбираем, поэтому мы должны улучшить наиболее приемлемую архитектуру-кандидата или разработать новую архитектуру-кандидата. Замечательно сказать, что этот компромисс присущ процессу проектирования.  
  
Теперь давайте поговорим о методе анализа компромиссов в архитектуре (ATAM), одном из наиболее часто используемых архитектурных решений. ATAM был разработан Институтом разработки программного обеспечения (SEI) в Университете Карнеги-Меллона. Согласно SEI, целью ATAM является оценка последствий архитектурных решений в свете требований к качественным атрибутам и бизнес-целей, на общем английском языке, это означает обнаружение рисков, когда на интересующий качественный атрибут влияют архитектурные решения (компромисс между атрибутами качества), и поэтому мы можем рассуждать о структуре системы и лежащем в основе обосновании.  
  
Одним из важных понятий, связанных с любым методом оценки архитектуры, является атрибут качества. Вкратце, функциональные требования определяют, что должно делать программное обеспечение, а нефункциональные требования (атрибуты качества) определяют, насколько хорошо это должно быть сделано. Функциональные и качественные признаки являются ортогональными. В сложных системах качественные атрибуты никогда не могут быть достигнуты изолированно. Чтобы достичь одного качества, нужно повлиять на другое качество (иногда отрицательное, а иногда положительное), поэтому архитектурные решения являются компромиссом между атрибутами качества для поддержки бизнес-целей.

Мы можем сгруппировать характеристики качества на три основные категории:

* перспективы конечного пользователя: производительность, доступность, удобство и безопасность
* Технические перспективы: изменяемость, портативность, возможность многократного использования, проверяемость, совместимость
* Перспективы бизнес сообщества: время на рынок, затраты и выгода, срок службы, бюджет проекта

Чтобы оценить архитектуру программного обеспечения с использованием атрибутов качества, нам необходимо правильно их охарактеризовать с использованием сценария атрибутов качества. Сценарий - это краткое изложение, описывающее взаимодействие заинтересованных сторон с системой. Затем сценарий атрибутов качества является способом конкретизации атрибутов качества.  
Сценарий атрибута качества состоит из шести частей:

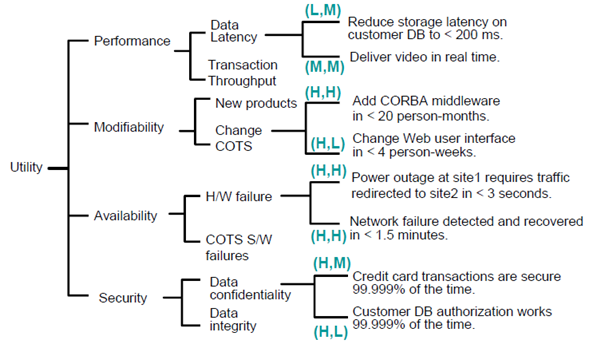
* Стимул: события, которые формируют архитектуру системы.
* Источник стимула: объект, генерирующий стимул (человек, компьютерная система и т. Д.).
* Среда: стимул возникает в условиях.
* Артефакт: стимулируется некоторый артефакт.
* Ответ: Ответ - это действие, предпринимаемое после получения стимула.
* Ответная мера: Ответ должен быть измерен для того, чтобы требования были проверены.

Давайте разберем предыдущие концепции на примере. Например, давайте проанализируем атрибут качества доступности. Доступность связана с отказом системы. Системный сбой происходит, когда система больше не предоставляет услугу. Одним из способов измерения доступности является вероятность того, что он будет работать, когда это необходимо, например, 99,9% доступности. Один из сценариев качества, описывающий это требование, состоит в следующем: в систему поступает сообщение о функциональности XXX при нормальных условиях, и ответ доступен с вероятностью 98%.  
  
Еще один пример атрибута качества производительности. Производительность касается времени отклика. Примером сценария атрибута качества является следующее: в систему поступает сообщение от внешней системы для выполнения функции YYY в нормальных условиях, и ответ предоставляется не менее чем через 8 секунд, чтобы считаться приемлемым.

Метод ATAM основан на четырех этапах:  
  
Этап 1 - Партнерство и подготовка. Команда оценки, заказчик и ключевые лица, принимающие решения по проекту, встречаются, чтобы понять метод ATAM (цели, входные данные, результаты, заинтересованные стороны и т. Д.), Бизнес-факторы и возможные подходы к архитектуре, которые необходимо оценить. Они согласны с предоставлением окончательного отчета, формальностей (техническое задание, соглашение о неразглашении), документации архитектуры системы, ожидаемых результатов оценки и т. Д. Этот этап может длиться несколько недель.  
Этап 2 - Начальная оценка и Этап 3 - Полная оценка. Эти этапы предназначены для оценки. До сих пор команда оценки изучала возможные подходы к архитектуре, и у них есть хорошее представление о бизнес-драйверах, задачах бизнеса и системы, ограничениях, наиболее важных качественных характеристиках. Клиент предоставляет информацию, связанную с бизнесом и сценарием качества. Имея эту информацию в руках, команда оценки может выбрать подходящий архитектурный подход и подготовить отчет об оценке. Два этапа состоят из девяти этапов. Эта фаза может длиться 3 или 4 дня с прошедшим временем от 2 до 3 недель.  
Этап 4 - продолжение. На этом этапе команда оценки пишет окончательный отчет. Ключевые лица, принимающие решения, вместе с главным архитектором соглашаются прекратить, изменить или переоценить другие возможные архитектурные подходы. Это также период улучшения оценки для более эффективного выполнения будущих оценок. Эта фаза может длиться одну неделю.

Теперь давайте подробно рассмотрим этапы оценки.  
  
**Шаг 1. Представьте метод ATAM**  
  
Группа оценки представляет обзор процесса ATAM, такой как ключевые шаги, методы (генерация дерева утилит, выявление и анализ архитектуры, мозговой штурм сценария) и результаты (архитектурные подходы, дерево утилит, сценарии, риски, точки чувствительности).  
 **Шаг 2. Представьте бизнес-драйверы**  
  
Заинтересованная сторона проекта и команда оценки пытаются понять контекст системы и основные бизнес-драйверы, мотивирующие ее развитие. Лицо, принимающее решения по проекту, представляет систему с точки зрения бизнеса, включая следующую информацию:

* бизнес-цели и контекст
* Основные заинтересованные стороны
* Функциональные требования высокого уровня (описанные как варианты использования или истории пользователей), которые влияют на архитектуру системы.
* Наиболее важные атрибуты качества (описаны как сценарии качества), которые влияют на архитектуру системы.
* Ограничения, такие как технические, управленческие, экономические и политические.

**Шаг 3. Представляемая архитектура**  
  
. Ведущий архитектор делает презентацию, описывающую архитектурный подход, используемый для удовлетворения требований и ограничений. Чтобы описать архитектуру, очень полезно использовать модель архитектурного представления 4 + 1, разработанную Kruchten. Эта модель представления описывает контекстное представление (отношения с людьми и другими системами), логическое представление (модуль, уровни, отношения), представление процесса (процесс, потоки, конвейер, синхронизация, поток данных, события) и представление развертывания (ЦП, хранилище , устройства, сеть). И, наконец, нам нужно описать риски, связанные с тем, чтобы соответствовать архитектурным требованиям. Как правило, архитектор должен представить наиболее важные виды при создании архитектуры.  
  
**Шаг 4. Определение архитектурных подходов.**  
  
Команда оценки пытается определить, какие ключевые архитектурные подходы используются для реализации требований и ограничений. Возможные архитектурные подходы: клиент-сервер, многоуровневое приложение, сервис-ориентированная архитектура, компонентное приложение, публикация-подписка и т. Д.  
  
На этом этапе группа оценки глубоко анализирует архитектуру, представленную на шаге 3, а затем имеет хорошее представление о том, какие шаблоны и подходы архитектор использовал при проектировании системы.  
  
**Шаг 5. Создание служебного дерева атрибутов качества**  
  
На этом шаге группа по оценке (вместе с лицом, принимающим решения по проекту) идентифицирует, устанавливает приоритеты и уточняет наиболее важные цели атрибутов качества (выраженные в сценарии качества) путем построения дерева утилит. Результатом является характеристика и определение приоритетов требований к качественным атрибутам путем составления приоритетного списка сценариев, который сообщает команде по оценке, где исследовать подходы архитектуры и обнаруживать риски. Таким образом, у нас может быть что-то осязаемое для оценки архитектуры системы, кстати, мы выбираем сценарии один за другим и оцениваем, насколько хорошо архитектура реагирует на этот сценарий.  
  
Дерево утилит - это нисходящий подход для характеристики требований к качеству, выбора наиболее важных целей качества для узлов высокого уровня (производительность, безопасность, доступность, модифицируемость, обслуживание), а листья дерева - это сценарии качества, оцениваемые важное (успех системы) и сложность (оценка архитектора).  
  
Дерево утилит начинается с утилиты в качестве корня. Полезность является выражением правильности системы. Атрибуты качества - это узлы второго уровня, которые определяются на шаге 2. Наиболее распространенные атрибуты качества - это безопасность, производительность, доступность, модифицируемость, удобство обслуживания и удобство использования. Узлы третьего уровня состоят из сценариев атрибутов качества или атрибутов качества, которые являются достаточно конкретными для анализа и определения приоритетов. Сценарии - это листья дерева, сгруппированные по атрибуту качества, который они выражают. Служебное дерево может содержать несколько сценариев в своих листьях, поэтому нам нужно расставить приоритеты для листьев, назначив такие значения, как High, Medium и Low. После этого сценарии устанавливаются по приоритетам во второй раз, связывая пару порядка (наиболее важный, наиболее сложный) со значениями Высокий, Средний и Низкий.   


**Шаг 6 - Анализ архитектурных подходов**  
  
На этом этапе команда оценки рассматривает сценарии с наивысшим рейтингом по одному, чтобы понять, как предложенная архитектура поддерживает каждый из них, а также определить и задокументировать решения архитектуры и их обоснование, риски, не -риски, точки чувствительности и компромиссы.  
  
Вы можете записать эту информацию, используя следующую форму, показанную на рисунке 3, где мы фиксируем анализ архитектурного подхода для сценария.

На этом этапе фаза 1 процесса оценки заканчивается. Команда по оценке начинает документировать резюме в течение одной или двух недель. Можно анализировать больше сценариев и решать вопросы.  
  
Когда группа по оценке и лицо, принимающее решения по проекту, готовы возобновить процесс оценки, заинтересованные стороны собираются, и начинается этап 2. Целью этого этапа является выявление точек зрения нескольких заинтересованных сторон для проверки результатов этапа 1.  
  
**Шаг 7 - Мозговой штурм и определение приоритетов сценариев**  
  
На этом этапе заинтересованные стороны генерируют сценарии с использованием упрощенного процесса мозгового штурма. После того, как сценарии сгенерированы, они должны иметь приоритет. Во-первых, заинтересованным сторонам предлагается объединить сценарии, которые, по их мнению, представляют одинаковые требования к качеству. После этого они должны проголосовать за те сценарии, которые они считают наиболее важными. Каждому заинтересованному лицу дается количество голосов, равное 30% от количества сценариев, округленных в большую сторону. Например, если у нас собрано двадцать сценариев, каждому участнику предоставляется шесть голосов. Каждый участник отдает свой голос публично. После подсчета голосов руководитель оценки упорядочивает сценарии по общему количеству голосов и устанавливает ограничение для сценариев. Сценарии над линией используются для следующих шагов. Например, команда должна рассмотреть пять лучших сценариев.  
  
Приоритетный список сценариев «мозгового штурма» сравнивается со сценариями в дереве служебных программ на шаге 5. Если они согласны, это указывает на согласованность того, что хотят заинтересованные стороны, и какое решение предлагает архитектор. Если обнаружены дополнительные сценарии, это указывает на некоторые риски в предлагаемой архитектуре. Таким образом, новые сценарии добавляются в дерево утилит, и архитектура переоценивается на их основе.  
 **Шаг 8 - Анализ архитектурных подходов**  
  
На этом этапе команда оценки направляет архитектора в процессе выполнения сценариев с наивысшим рейтингом из шага 7. Архитектор определяет, как на архитектурные подходы влияют сценарии, сгенерированные на предыдущем шаге. Риски, отсутствие рисков, точки чувствительности и компромиссы по-прежнему идентифицируются и определяются архитектурные решения.  
 **Шаг 9 - Представление результатов**  
  
И, наконец, информация, полученная в процессе ATAM, должна быть представлена ​​заинтересованным сторонам. Команда оценки может написать отчет и представить идеи в виде слайдов. Ведущий архитектор должен сообщить бизнес-контекст и факторы, требования и ограничения, а также документацию по выбранной архитектуре, ряду приоритетных сценариев, дереву служебных программ и обнаруженным рискам, отсутствию рисков, точкам чувствительности и компромиссам.