|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА - Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №3** | |
| **по дисциплине** | |
| «Архитектура программных продуктов и систем»  Методы анализа архитектуры программных систем. | |
|  | |
| Выполнил студент группы ИКБО-13-17 | Фурсов Ф.О. |
| Принял доцент, ученая степень к.т.н. | Куликов А.А. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2020 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. | *(подпись руководителя)* |

Москва 2021

**Оглавление:**

[Цель работы 3](#_Toc65880688)

[Теоретические сведения 3](#_Toc65880689)

[Задание 5](#_Toc65880690)

[Выполнение работы 5](#_Toc65880691)

[Метод SAAM 5](#_Toc65880692)

[Метод ATAM 5](#_Toc65880693)

[Сравнение подходов 8](#_Toc65880694)

[Вывод 8](#_Toc65880695)

[Список литературы 15](#_Toc65880696)

# Цель работы

Целью работы является получение навыков оценки и анализа архитектур программных систем.

# Теоретические сведения

Анализ архитектуры приложения – критически важная задача, поскольку позволяет сократить затраты на исправление ошибок, как можно раньше выявить и исправить возможные проблемы. Анализ архитектуры следует выполнять часто: по завершении основных этапов проекта и в ответ на существенные изменения в архитектуре. Создавайте архитектуру, помня об основных вопросах, задаваемых при таком анализе, это позволит как улучшить архитектуру, так и сократить время, затрачиваемое на каждый анализ.

Основная цель анализа архитектуры – подтверждение применимости базовой архитектуры и ее возможных вариантов, и также проверка соответствия предлагаемых технических решений функциональным требованиям и параметрам качества. Кроме того, анализ помогает обнаружить проблемы и выявить области, требующие доработки.

На практике для анализа дизайна архитектуры применяют следующие типовые методики:

**Метод анализа архитектуры ПО (Software Architecture Analysis Method, SAAM)**. Изначально SAAM создавался для оценки модифицируемости, но позже был расширен для анализа архитектуры относительно показателей качества, таких как модифицируемость, портируемость, расширяемость, интегрируемость и функциональный охват.

**Метод анализа архитектурных компромиссов (Architecture Tradeoff Analysis Method, ATAM).** 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.

**Активный анализ конструкции (Active Design Review, ADR)**. 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 или отдельных разделах, а не на проведении общего анализа.

**Активный анализ промежуточных конструкций (Active Reviews of Intermediate Designs, ARID)**. 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.

**Метод анализа рентабельности (Cost Benefit Analysis Method, CBAM)**. Метод CBAM основное внимание уделяет анализу затрат, выгод и планированию последствий архитектурных решений.

**Анализ модифицируемости на уровне архитектуры (Architecture Level Modifiability Analysis, ALMA)**. ALMA оценивает модифицируемость архитектуры для систем бизнес-аналитики (business information systems, BIS).

**Метод оценки семейства архитектур (Family Architecture Assessment Method, FAAM)**. FAAM оценивает архитектуры семейства информационных систем с точки зрения возможности взаимодействия и расширяемости.

# Задание

Из приведённого выше списка методов анализа дизайна архитектуры необходимо выбрать 2 метода, дать их подробное описание и провести сравнительный анализ выбранных методов.

# Выполнение работы

**Метод SAAM**

Метод анализа архитектуры программного обеспечения (SAAM) – это метод, используемый в архитектуре программного обеспечения для оценки архитектуры системы. Был документирован в середине 1990-х годов для анализа системы на предмет её модифицируемости.

SAAM был предшественником метода анализа компромиссов архитектуры (ATAM).

**Шаги в процессе SAAM**:

1. Презентация ATAM.
2. Презентация бизнес-драйверов.
3. Презентация архитектуры.
4. Выявление архитектурных решений.
5. Генерация дерева полезности и сценариев качества.
6. Анализ архитектурных решений.
7. Мозговой штурм и приоритизация сценариев.
8. Анализ архитектурных решений.
9. Представление результата.

**Метод ATAM**

В программировании аналитический метод компромисса архитектуры (ATAM) – процесс смягчения риска, используемый рано в жизненном цикле разработки программного обеспечения.

ATAM был развит Институтом Программирования в Университете Карнеги-Меллон. Его цель состоит в том, чтобы помочь выбрать подходящую архитектуру для системы программного обеспечения, обнаружив пункты чувствительности и компромиссы.

ATAM является самым выгодным, когда стоимость изменяющейся архитектуры минимальна.

**Шаги процесса ATAM**

ATAM формально состоит из девяти шагов, обрисованных в общих чертах ниже:

1. Существующий ATAM – Подарок понятие ATAM заинтересованным сторонам, и отвечает на любые вопросы о процессе.
2. Существующие бизнес-факторы – все в процессе представляют и оценивают бизнес-факторы для рассматриваемой системы.
3. Представьте архитектуру – архитектор представляет архитектуру высокого уровня команде с 'соответствующим уровнем детали'
4. Определите архитектурные подходы – различные архитектурные подходы к системе представлены командой и обсуждены.
5. Произведите качественное сервисное дерево признака – определяют основной бизнес и технические требования системы, и наносят на карту их к соответствующей архитектурной собственности. Представьте сценарий для этого данного требования.
6. Проанализируйте архитектурные подходы – Анализируют каждый сценарий, оценивая их приоритетом. Архитектура тогда оценена против каждого сценария.
7. Проведите коллективное обсуждение и расположите по приоритетам сценарии – среди более многочисленной группы заинтересованной стороны, представьте текущие сценарии и расширьтесь.
8. Проанализируйте архитектурные подходы – Выполняют шаг 6 снова с добавленным знанием более многочисленного сообщества заинтересованной стороны.
9. Существующие результаты – предоставляют всю документацию заинтересованным сторонам.

Эти шаги определены в двух фазах: Фаза 1 состоит из шагов 1-6 и после этой фазы, ведущие архитектурные требования становятся известны. Фаза 2 состоит из шагов 7-9 и заканчивает оценку.

**Преимущества ATAM**:

* Способствует сбору точных требований к уровню качества ПО;
* Создает зарегистрированное обоснование для архитектурных решений;
* Способствует ранней идентификации рисков жизненном цикле ПО;
* Поощряет коммуникацию среди заинтересованных сторон;
* Имеет ясное объяснение архитектуры;
* Раскрывает возможности для повторного использования поперечного проекта;

**Процесс ATAM**

Процесс ATAM состоит из собирающихся заинтересованных сторон, чтобы проанализировать бизнес-факторы (системная функциональность, цели, ограничения, желали нефункциональных свойств.

Сгенерированные сценарии используются вместе с архитектурными подходами и архитектурными решениями, чтобы создать анализ компромиссов, пунктов чувствительности, и рисков. Этот анализ может быть преобразован, чтобы рискнуть темами и их воздействиями после чего, процесс может быть повторен.

С каждым аналитическим циклом аналитический процесс продолжается от более общего до более определенного, исследуя вопросы, которые были обнаружены в предыдущем цикле, до тех пор, пока архитектура не будет точно настроена и все риски не будут просчитаны.

**Сравнение подходов**

Метод SAAM представляет собой основоположника ряда методов анализа архитектуры программных систем. Он имеет чёткую документацию и прост в освоении.

Метод ATAM является альтернативой методу SAAM. Данный метод более предпочтителен, так как является более современным вариантом SAAM.

# Вывод

В ходе выполнения данной практической работы мной были изучены две метода анализа архитектуры программных систем: ATAM и SAAM. Мной была дана основная информация о каждом из методов, а также описаны шаги для их реализации.

# Цель работы

Целью работы является получение навыков оценки и анализа архитектур программных систем.

# Теоретические сведения

Анализ архитектуры приложения – критически важная задача, поскольку позволяет сократить затраты на исправление ошибок, как можно раньше выявить и исправить возможные проблемы. Анализ архитектуры следует выполнять часто: по завершении основных этапов проекта и в ответ на существенные изменения в архитектуре. Создавайте архитектуру, помня об основных вопросах, задаваемых при таком анализе, это позволит как улучшить архитектуру, так и сократить время, затрачиваемое на каждый анализ.

Основная цель анализа архитектуры – подтверждение применимости базовой архитектуры и ее возможных вариантов, и также проверка соответствия предлагаемых технических решений функциональным требованиям и параметрам качества. Кроме того, анализ помогает обнаружить проблемы и выявить области, требующие доработки.

На практике для анализа дизайна архитектуры применяют следующие типовые методики:

**Метод анализа архитектуры ПО (Software Architecture Analysis Method, SAAM)**. Изначально SAAM создавался для оценки модифицируемости, но позже был расширен для анализа архитектуры относительно показателей качества, таких как модифицируемость, портируемость, расширяемость, интегрируемость и функциональный охват.

**Метод анализа архитектурных компромиссов (Architecture Tradeoff Analysis Method, ATAM).** 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.

**Активный анализ конструкции (Active Design Review, ADR)**. 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 или отдельных разделах, а не на проведении общего анализа.

**Активный анализ промежуточных конструкций (Active Reviews of Intermediate Designs, ARID)**. 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.

**Метод анализа рентабельности (Cost Benefit Analysis Method, CBAM)**. Метод CBAM основное внимание уделяет анализу затрат, выгод и планированию последствий архитектурных решений.

**Анализ модифицируемости на уровне архитектуры (Architecture Level Modifiability Analysis, ALMA)**. ALMA оценивает модифицируемость архитектуры для систем бизнес-аналитики (business information systems, BIS).

**Метод оценки семейства архитектур (Family Architecture Assessment Method, FAAM)**. FAAM оценивает архитектуры семейства информационных систем с точки зрения возможности взаимодействия и расширяемости.

# Задание

Из приведённого выше списка методов анализа дизайна архитектуры необходимо выбрать 2 метода, дать их подробное описание и провести сравнительный анализ выбранных методов.

# Выполнение работы

**Метод SAAM**

Метод анализа архитектуры программного обеспечения (SAAM) – это метод, используемый в архитектуре программного обеспечения для оценки архитектуры системы. Был документирован в середине 1990-х годов для анализа системы на предмет её модифицируемости.

SAAM был предшественником метода анализа компромиссов архитектуры (ATAM).

**Шаги в процессе SAAM**:

1. Презентация ATAM.
2. Презентация бизнес-драйверов.
3. Презентация архитектуры.
4. Выявление архитектурных решений.
5. Генерация дерева полезности и сценариев качества.
6. Анализ архитектурных решений.
7. Мозговой штурм и приоритизация сценариев.
8. Анализ архитектурных решений.
9. Представление результата.

**Метод ATAM**

В программировании аналитический метод компромисса архитектуры (ATAM) – процесс смягчения риска, используемый рано в жизненном цикле разработки программного обеспечения.

ATAM был развит Институтом Программирования в Университете Карнеги-Меллон. Его цель состоит в том, чтобы помочь выбрать подходящую архитектуру для системы программного обеспечения, обнаружив пункты чувствительности и компромиссы.

ATAM является самым выгодным, когда стоимость изменяющейся архитектуры минимальна.

**Шаги процесса ATAM**

ATAM формально состоит из девяти шагов, обрисованных в общих чертах ниже:

1. Существующий ATAM – Подарок понятие ATAM заинтересованным сторонам, и отвечает на любые вопросы о процессе.
2. Существующие бизнес-факторы – все в процессе представляют и оценивают бизнес-факторы для рассматриваемой системы.
3. Представьте архитектуру – архитектор представляет архитектуру высокого уровня команде с 'соответствующим уровнем детали'
4. Определите архитектурные подходы – различные архитектурные подходы к системе представлены командой и обсуждены.
5. Произведите качественное сервисное дерево признака – определяют основной бизнес и технические требования системы, и наносят на карту их к соответствующей архитектурной собственности. Представьте сценарий для этого данного требования.
6. Проанализируйте архитектурные подходы – Анализируют каждый сценарий, оценивая их приоритетом. Архитектура тогда оценена против каждого сценария.
7. Проведите коллективное обсуждение и расположите по приоритетам сценарии – среди более многочисленной группы заинтересованной стороны, представьте текущие сценарии и расширьтесь.
8. Проанализируйте архитектурные подходы – Выполняют шаг 6 снова с добавленным знанием более многочисленного сообщества заинтересованной стороны.
9. Существующие результаты – предоставляют всю документацию заинтересованным сторонам.

Эти шаги определены в двух фазах: Фаза 1 состоит из шагов 1-6 и после этой фазы, ведущие архитектурные требования становятся известны. Фаза 2 состоит из шагов 7-9 и заканчивает оценку.

**Преимущества ATAM**:

* Способствует сбору точных требований к уровню качества ПО;
* Создает зарегистрированное обоснование для архитектурных решений;
* Способствует ранней идентификации рисков жизненном цикле ПО;
* Поощряет коммуникацию среди заинтересованных сторон;
* Имеет ясное объяснение архитектуры;
* Раскрывает возможности для повторного использования поперечного проекта;

**Процесс ATAM**

Процесс ATAM состоит из собирающихся заинтересованных сторон, чтобы проанализировать бизнес-факторы (системная функциональность, цели, ограничения, желали нефункциональных свойств.

Сгенерированные сценарии используются вместе с архитектурными подходами и архитектурными решениями, чтобы создать анализ компромиссов, пунктов чувствительности, и рисков. Этот анализ может быть преобразован, чтобы рискнуть темами и их воздействиями после чего, процесс может быть повторен.

С каждым аналитическим циклом аналитический процесс продолжается от более общего до более определенного, исследуя вопросы, которые были обнаружены в предыдущем цикле, до тех пор, пока архитектура не будет точно настроена и все риски не будут просчитаны.

**Сравнение подходов**

Метод SAAM представляет собой основоположника ряда методов анализа архитектуры программных систем. Он имеет чёткую документацию и прост в освоении.

Метод ATAM является альтернативой методу SAAM. Данный метод более предпочтителен, так как является более современным вариантом SAAM.

# Вывод

В ходе выполнения данной практической работы мной были изучены две метода анализа архитектуры программных систем: ATAM и SAAM. Мной была дана основная информация о каждом из методов, а также описаны шаги для их реализации.

# Список литературы

1. URL: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=5177 [электронный ресурс] (Дата обращения 03.03.2021);
2. URL: https://www.geeksforgeeks.org/architecture-tradeoff-analysis-method-atam/ [электронный ресурс] (Дата обращения 04.03.2021).
3. URL: https://medium.com/@bhagvankommadi/saam-software-architecture-analysis-method-36864cd8ea94 [электронный ресурс] (Дата обращения 04.03.2021).
4. URL: https://dl.acm.org/doi/abs/10.5555/257734.257746 [электронный ресурс] (Дата обращения 04.03.2021).