

ВИКОРИСТАННЯ АТАМ ДЛЯ АНАЛІЗУ АРХІТЕКТУРНИХ РІШЕНЬ У РОЗРОБЦІ ПЗ

Вінницький національний технічний університет

Анотація

Дана робота описує метод АТАМ (Architecture Tradeoff Analysis Method), який використовується для оцінки архітектури програмних систем. Метою дослідження є аналіз і застосування цього методу для виявлення компромісів між різними атрибутами якості системи, такими як надійність, продуктивність, безпека, модифікованість. У роботі розглянуто етапи, що включають збір вимог та сценаріїв, оцінку архітектурних рішень, виявлення ризиків і точок компромісу, а також співвідношення архітектурних рішень із вимогами до якості.

Ключові слова: метод АТАМ, оцінка архітектури, програмні системи, атрибути якості, надійність, продуктивність, безпека, модифікованість, виявлення ризиків, точка компромісу, аналіз архітектурних рішень, системні слабкості, прийняття рішень, аналіз вимог.

Abstract

This paper describes the ATAM (Architecture Tradeoff Analysis Method), which is used for evaluating the architecture of software systems. The aim of the study is to analyze and apply this method to identify trade-offs between various quality attributes of the system, such as reliability, performance, security, and modifiability. The paper examines the steps of ATAM, including the collection of requirements and scenarios, evaluation of architectural decisions, identification of risks and trade-off points, as well as the alignment of architectural decisions with quality requirements.

Keywords: ATAM method, architecture evaluation, software systems, quality attributes, reliability, performance, security, modifiability, risk identification, tradeoff point, architectural decision analysis, system weaknesses, decision making, requirements analysis.

Метод АТАМ, або Architecture Tradeoff Analysis Method, це такий метод оцінки якості проектування архітектури ПЗ, який враховує декілька атрибутів, таких як надійність, продуктивність, доступність, модифікованість і так далі. Ці атрибути взаємодіють між собою, і бажання покращити один з них часто відбувається за ціною погіршення одного чи навіть кількох інших. Наприклад, модифікованість впливає на продуктивність; доступність впливає на безпеку; безпека впливає на продуктивність; всі атрибути впливають на вартість. Метод ідентифікує ці точки компромісів, що є головною відмінністю АТАМ від інших методів аналізу програмного забезпечення, адже він прямо враховує взаємозв'язки між атрибутами та дозволяє систематично оцінювати так звані компроміси, що неминуче виникають.

Компроміси є частиною будь-якого проектування, незалежно від галузі. Головною проблемою є як робити саме обґрунтовані та найбільш оптимальні компроміси, і структурований підхід тут може допомогти, бо якщо на ранніх етапах поставити правильні питання (наприклад, під час визначення вимог до ПЗ), то можна виявити проблеми і вирішити їх з меншими витратами [1].

Загалом метод передбачає декілька головних кроків. На ранніх етапах необхідно зібрати вимоги і фактори середовища, адже обмеження проектного простору прямо впливатиме на аналіз атрибутів, і також визначитись з обмеженнями. Іноді вимоги визначаються ще до початку архітектурного аналізу, а іноді сценарії формують вимоги. Виявлення сценаріїв допомагає зробити функціональні та якісні вимоги більш конкретними, що полегшує комунікацію між зацікавленими сторонами (stakeholders), та дозволяє створити спільне бачення ключових функцій системи.

Другий крок пов'язаний з описом архітектурних представлень та реалізацією сценарію. Вимоги, сценарії та інженерний дизайн формують архітектурну модель, яка описується через елементи архітектури та властивості, що мають значення для кожного з важливих атрибутів якості. Наприклад, механізм дублювання та алгоритми узгодженого прийняття рішень є важливими елементами для надійності. Для підвищення продуктивності слід врахувати розподіл процесів, пріоритети та оцінку

пропускної здатності. Безпека системи залежить від моделей поведінки потенційних злоумисників. Адаптивність та зручність модифікації забезпечуються завдяки використанню принципів модульності, інкапсуляції та абстракції. В АТАМ зазвичай розглядається кілька архітектурних варіантів для порівняння.

Третій крок вимагає аналіз кожного атрибуту якості окремо для кожної архітектури. Наприклад, запити обробляються в середньому за 60 мс, система захищена від відомих атак, для обслуговування програмного забезпечення знадобиться 4 людини на рік і тд.

Крок чотири можна поділити на два етапи. Спочатку визначається, як зміни в архітектурних елементах впливають на окремі атрибути, і якщо змінений елемент суттєво впливає на атрибут, це називається чутливою точкою.

Наступним етапом є критика моделей, побудованих на кроці 3, і пошук архітектурних точок компромісу. Наприклад, у клієнт-серверній архітектурі продуктивність покращується зі збільшенням кількості серверів; надійність також підвищується. Проте безпека може знижуватись, оскільки з'являється більше точок потенційної атаки. Кількість серверів у цьому випадку є точкою компромісу, і саме такі компроміси АТАМ допомагає ідентифікувати та проаналізувати, щоб ухвалити обґрунтовані архітектурні рішення.

Після виконання всіх зазначених вище кроків, можна зіставити результати аналізу з початковими вимогами. Якщо прогнозована поведінка системи відповідає встановленим критеріям, розробники можуть перейти до детальнішого проектування або безпосередньо до реалізації. Однак на практиці доцільно продовжувати відстежувати архітектуру за допомогою аналітичних моделей, що сприятиме ефективнішому розвитку. Процес проектування триває протягом усього життєвого циклу системи, тому аналітичні перевірки також не повинні припинятись.

Важливо розуміти, що ці етапи не виконуються суворо послідовно. Вони взаємопов'язані та можуть впливати один на одного. Наприклад, результати аналізу можуть призвести до повторного перегляду вимог, а створення моделі може виявити слабкі місця в архітектурі[2].

Отже, метод АТАМ дозволяє структуровано оцінити архітектуру системи, визначаючи компроміси між різними атрибутами якості. Його результати включають стислий опис архітектури, формулювання бізнес-цілей, визначення пріоритетних вимог до якості та виявлення потенційних ризиків. Також АТАМ допомагає ідентифікувати слабкі місця в архітектурі, процесах і команді, а також співвіднести архітектурні рішення з вимогами до якості. Це дозволяє вчасно виявити проблеми та прийняти обґрунтовані рішення для подальшого проектування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, J. Carriere. The Architecture Tradeoff Analysis Method [Електронний ресурс] – Режим доступу: https://insights.sei.cmu.edu/documents/1186/1998_005_001_16646.pdf
2. Geeksforgeeks. Architecture Tradeoff Analysis Method (ATAM). [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/architecture-tradeoff-analysis-method-atam/> (дата звернення 07.03.2025)

Козак Олександра Володимирівна – студентка групи ЗПІ-23б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: iam.asssya@gmail.com

Науковий керівник: **Бабюк Наталя Петрівна** – доцент кафедри ПЗ, Вінницький національний технічний університет, м. Вінниця, e-mail: babiuk@vntu.edu.ua

Olexandra V. Kozak – a student of ЗПІ-23b, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: iam.asssya@gmail.com

Supervisor: **Natalia P. Babiuk** – Associate Professor of the Department of Software, Vinnytsia National Technical University, Vinnytsia, e-mail: babiuk@vntu.edu.ua