Chapter 2
Software Architecture:
Developing Knowledge, Skills, and Experiences

Perla Velasco-Elizondo
Universidad Autónoma de Zacatecas (UAZ), México

ABSTRACT

What is software architecture? A clear and simple definition is that software architecture is about making important design decisions that you want to get right early in the development of a software system because, in the future, they are costly to change. Being a good software architect is not easy. It requires not only a deep technical competency from practicing software architecture design in industry, but also an excellent understanding of the theoretical foundations of software architecture are gained from doing software architecture research. This chapter describes some significant research, development, and education activities that the author has performed during her professional trajectory path to develop knowledge, skills, and experiences around this topic.

INTRODUCTION

What is software architecture? To say it simple: software architecture is about making the design decisions that you want to get right early in the development of a software system, because future changes are costly. Today, software architecture development is necessary as never before; no organization begins a complex software system without a suitable software architecture.

Within the context of the software life cycle (Sommerville, 2011), software architecture is an artifact produced during the design phase. A software architect, or the software architecture design team, is responsible for defining software architecture. Being a good software architect is not an easy matter (Rehman et al., 2018), (Shahbazian, Lee & Medvidovic, 2018). The author considers that, it not only requires deep technical competency which comes from practicing software architecture design in industry; but also a very good understanding of the theoretical foundations of software architecture gained from doing software architecture research.

Dr. Velasco-Elizondo finds the topic of software architecture fascinating. This chapter describes some of the significant research, education and, coaching activities she has undertaken during her professional trajectory path to develop knowledge, skills and experiences on this topic. She hopes that this material helps to encourage readers and, particularly, other women to get involved in science, technology and engineering.

This chapter will cover the following sections:

- **Getting it right: software architecture foundations**. This section describes how software architecture foundations are conceived and an example of why preserving them in practice is not always straightforward. It will discuss how to tackle this shortcoming with the proposal of exogenous connectors.
- **Practicing it right: software architecture methods**. In this section, the notion of software architecture lifecycle is introduced. Relevant methods for software architecture development are then briefly discussed within the context of this lifecycle, as well as some limitations related to the difficulty of adopting these methods in practice. Finally, an explanation of why and how technology has to be considered as a first-class design concept in order to tackle one of these limitations will be given.
- **Automating technology selection**. This section presents a software tool, recently developed, which uses information retrieval, natural language processing and sematic web techniques to address the problem of automating NoSQL database technologies search.
- **Software architecture education**. This section describes two educational projects Dr. Velasco-Elizondo has led to promote knowledge and practical experiences on software architecture design and development.
- **Hands on.** Dr. Velasco-Elizondo has had the opportunity to work, as a coach, with practicing software architects and developers helping them to deploy software architecture practices and methods. In this section some of these works will be described.

GETTING IT RIGHT: SOFTWARE ARCHITECTURE FOUNDATIONS

Software architecture has always existed as part of the discipline of Software Engineering. This section describes how software architecture foundations are conceived and gives an example of why preserving them in practice is not always straightforward. The proposed use of exogenous connectors to tackle this shortcoming is also included.

Foundations in Theory

Back when systems were relatively "less complex and small", abstract diagrams were drawn to give stakeholders a better understanding of software designs when describing them. Later, systems went beyond simple algorithms and data structures becoming more complex and larger in size. Therefore, similar in practice to other branches of engineering, more structured diagrams were essential to describe software system designs and communicate regarding aspects such as their main parts and responsibilities, their communication and data model, etc.

Related Content

Optimal Routing for Vehicles and Pedestrians in the Urban Area
Gitae Kimand Jinsik Choe (2022). *International Journal of Software Innovation (pp. 1-18).*
www.irma-international.org/article/optimal-routing-for-vehicles-and-pedestrians-in-the-urban-area/301219

Image Segmentation Using Electromagnetic Field Optimization (EFO) in E-Commerce
Applications
Pankaj Upadhyayand Jitender Kumar Chhabra (2019). *International Journal of Information System
Modeling and Design (pp. 76-91).*
www.irma-international.org/article/image-segmentation-using-electromagnetic-field-optimization-efo-in-e-commerce-
applications/234772

Automated Ranking of Relaxing Query Results Based on XML Structure and Content
Preferences
Wei Yan, Li Yanand Z. M. Ma (2013). *Mobile and Web Innovations in Systems and Service-Oriented
Engineering (pp. 44-62).*
www.irma-international.org/chapter/automated-ranking-relaxing-query-results/71990

Integrating DSLs into a Software Engineering Process: Application to Collaborative Construction
of Telecom Services
Vanea Chiprianov, Yvon Kermarrecand Siegfried Rouvrais (2013). *Formal and Practical Aspects of
Domain-Specific Languages: Recent Developments (pp. 408-434).*
www.irma-international.org/chapter/integrating-dsls-into-software-engineering/71828

VIPER: Evaluation of an Integrated Group VoiceIP Software Application for Teaching and
Learning in Higher Education
John Beaumont-Kerridge (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications
(pp. 1394-1403).*
www.irma-international.org/chapter/viper-evaluation-integrated-group-voiceip/29452