THE STRATEGY OF ARCHITECTURAL STYLE CHOICE FOR NETWORK-BASED SOFTWARE IN THE SPHERE OF DISTANCE EDUCATION

Posudevskiy K.V.

LLC Playtika Bel, Belarussian State University of Informatics and Radioelectronics, Minsk, Belarus, konstantin.posudevskiy@gmail.com

Abstract. This publication describes iterative technique that you can use to think about and sketch out your potential architecture. It will help you to bring together the key decisions discussed in this guide; including quality attributes, architecture styles, application types and technologies. The technique includes a series of five main steps, each of which breaks down into individual considerations explained throughout the remainder of the guide. The iterative process will help you to produce candidate solutions that you can further refine by repeating the steps, finally creating an architecture design that best fits your application.

An architectural style is a coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style [1].

A software architecture is defined by a configuration of architectural elements – components, connectors, and data – constrained in their relationships in order to achieve a desired set of architectural properties [1].

To determine architecture constrains and properties, you should determine your inputs. The inputs to your design can help you to formalize the requirements and constraints that your architecture must accommodate. Common inputs are use cases and usage scenarios, functional requirements, non-functional requirements (including quality attributes such as performance, security, reliability, and others), technological requirements, the target deployment environment, and other constraints.

During the design process, you will create a list of the architecturally significant use cases, the architecture issues that require special attention, and the candidate architecture solutions that satisfy the requirements and constraints defined in the design process. A common technique for refining the design over time, until it satisfies all of the requirements and adheres to all of the constraints, is an iterative technique consisting of the five major stages shown in Figure 1 [2].

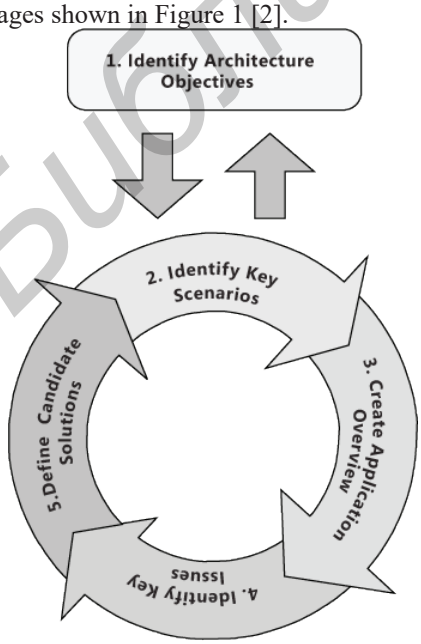

Figure 1 – The iterative steps for core architecture design activities

The steps, described in more detail in the following sections, are:

1. *Identify Architecture Objectives*. Clear objectives help you to focus on your architecture and on solving the right problems in your design. Precise objectives help you to determine when you have completed the current phase, and when you are ready to move to the next phase.

2. *Key Scenarios*. Use key scenarios to focus your design on what matters most, and to evaluate your candidate architectures when they are ready.

3. *Application Overview*. Identify your application type, deployment architecture, architecture styles, and technologies in order to connect your design to the real world in which the application will operate.

4. *Key Issues*. Identify key issues based on quality attributes and crosscutting concerns. These are the areas where mistakes are most often made when designing an application.

5. *Candidate Solutions*. Create an architecture spike or prototype that evolves and improves the solution and evaluate it against your key scenarios, issues, and deployment constraints before beginning the next iteration of your architecture.

This architectural process is meant to be an iterative and incremental approach. Your first candidate architecture will be a high-level design that you can test against key scenarios, requirements, known constraints, quality attributes, and the architecture frame. As you refine your candidate architecture, you will learn more details about the design and will be able to further expand key scenarios, your application overview, and your approach to issues [2].

It is quite difficult to build your architecture in a single iteration. Each iteration should add more detail. Focus on the major steps and build a framework on which you can base your architecture and design.

Literature

1. Fielding, R. Th. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation. University of California, Irvine, 2000. – https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

2. Microsoft Application Architecture Guide (Patterns and Practices) – ISBN: 9780735627109. – Microsoft Corporation. – 2009. – 37 p.