Software Architecture Template

Dr Fritz Solms

March 4, 2015

Contents

1 Software Architecture Documentation

This is template for a software architecture specification document. It is meant to provide some guidelines as an alternative to the Kruchten $4+1$ approach to documenting a software architecture and is more aligned with the approach of a software architecture providing the infrastructure within which application functionality is deployed and executed.

Throughout the document you are encouraged to use diagrams to illustrate aspects of your software architecture.

Note that you can leave out sections which are not relevant to you, add sections which are and generally modify the structure of the document to suit your specific project.

1.1 Architecture requirements

In this section extract the architectural requirments from the software requirements including

- scope of architectural responsibilities (e.g. persistence, reporting, process execution, . . .),

- quantified quality requirements,

- integration and access channel requirements, and

- any architectural constraints.

1.1.1 Architectural scope

In this section discuss architectural responsibilities which need to be addressed by the software architecture. Typical examples include those of

- providing a persistence infrastructure (e.g. database),

- providing a reporting infrastructure,

- providing an infrastructure for process execution,

- . . .

1.1.2 Quality requirements

This section should state the quality requirements in order of priority/importance. Examples of quality requirements are scalability, reliability, performance, security, auditability, integrability, ...

Each quality requirement needs to be quantified. For example, scalability could be specified in terms of number of transactions per unit time or number of concurrent users.

1.1.3 Integration and access channel requirements

In this section you need to specify the different system which your system must integrate with and the integratation channels and protocols which need to be used.

You should also specify the different access channels through which the system functionality should be made available to humans and/or other systems.

1.1.4 Architectural constraints

In this section discuss the architecturl constraints (if any) which your client has placed on the software architecture of the system you are developing. Exampes of architecturl constraints include

- that the system architecture must be based on some or other reference architecture (e.g. SOA, Java-EE, . . .).

- particular technologies (e.g. programming languages, frameworks, protocols, . . .) you should be using, and

- operating systems and/or devices across which the system must be deployable.

1.2 Architectural patterns or styles

In this section discuss any architectural patterns or styles you have chosen to use together with the rationale for using them.

For example, you might have decided to use layering in order to have lower level layers reusable across different higher level layers (e.g. have a services layer usable by a web front-end and a mobile-device client).

Discuss the elements of the architectural pattern (e.g. the different layers) and how these elements are connected (e.g. the integration channels between the layers).

1.3 Architectural tactics or strategies

In this section discuss any architectural tactics or strategies you are using to concretely address any of the quality requirements.

For example, you could be using thread pooling and/or caching to achieve a higher level of scalability or performance.

Discuss how the strategy is realized within your software architecture.

1.4 Use of reference architectures and frameworks

In this section discuss any reference architectures and/or frameworks you might be incorporating within your software architecture. For example, you could be using an object-relational mapper or a particular adapter or even an application server or an enterprise services bus.

Disucss the reasons for choosing such frameworks for your software architecture.

1.5 Access and integration channels

Discuss in this section

- the different access channels enabling human and system users to use your system (e.g. a web and mobile front-end and a web-services access channel), and

- the integration channels through which your system will integrate with external systems (e.g. via a message queue or the database).

Discuss any protocols and/or technologies you are using for those access and integration channels.

1.6   Technologies

This section can be used to specify programming languages, operating systems and other technologies which you are using for your system.