PA1308 Self Study Package – Fundamentals

Mikael Svahnberg
Blekinge Institute of Technology
SE-371 79 Karlskrona SWEDEN
Mikael.Svahnberg@bth.se

September 5, 2012

Abstract

The focus of this self study package is to provide a foundation of concepts and ideas in software architectures and software architecting.
Version: 1.1

1 Basic Concepts

What is an architecture?
There are many definitions of what an architecture is and what it should consist of. Based on your experience, what do *you* think an architecture is? Why do you think your definition defines an architecture? What are the good things with your definition? What do you think is missing when using your definition?
Hints: `http://www.sei.cmu.edu/architecture/`

Architecture vs. Design
What is the difference between architecture and design?

Why architectures?
Why do you think it is important to construct an architecture? For whom do you construct the architecture? What purposes does the architecture serve when you continue developing?

A good Architecture?
List product and process requirements for a "good" architecture? Why are these requirements relevant? How can a software architecture help organisations meet their business goals?

Quality Attributes I
Consider a software product that you like. Why do you like it? If parts of your answer includes something like "it has good quality", then describe what you think quality means to you. Do the same for a software product that you do not like. More generally, is there a commonly agreed upon list of items that together form the concept of quality?
Hints: ISO9126 Swebok

Quality Attributes II
Are the same qualities relevant when building a system as when you are executing it? What about when you are maintaining the system?
Hints: ISO9126 Swebok

2 Applied Software Architecture

Decision Trace
How may we ensure that there is a trace from requirements down to your software architecture? Why do we need this trace?
Hints: Hofmeister et al. "Applied Software Architecture", `http://www.amazon.com/exec/obidos/ISBN=0201325713/royrweilA`

Documenting an Architecture
What is important to document in an architecture? Why?
Hints: Clements et al. "Documenting Software Architectures: Views and Beyond", `http://www.amazon.com/exec/obidos/ISBN=0201703726/royrweilA`

Mixing Descriptions
How can you describe *both* how you are going to split your system up into development modules as well as how it will behave in runtime? Should you mix this in one place? Should you maintain two separate descriptions? How would you ensure that the two views are consistent?
Hints: Kruchten, P.B.; , "The 4+1 View Model of architecture," Software, IEEE , vol.12, no.6, pp.42-50, Nov 1995 doi: 10.1109/52.469759 `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=469759&isnumber=9910`

Organisational Influence
How does your own organisation influence how you develop a software system? Should you express this in the architecture? Why or why not? How would you express this in the architecture? How does the organisation of the company you are developing for influence the system you develop? Should you express this in the architecture? Why or why not? How would you express this in the architecture?

Quality Attributes
Which, do you think, is harder to achieve in an architeture? The right functionality, or that the system is able to deliver the functionality within a certain response time? Why? In which parts of the architecture documentation would you address the response time?

When to start?
When should you start developing your software architecture? Why not earlier? Why not later?

Architecture Design Methods
List at least three methods for designing software architectures. What are their differences? What are their similarities? How can they help to create and document an architecture so that it is usable for all its purposes? Are there different situations where you would prefer to use different methods? Why and when?

3 Advanced Concepts

Quality Attribute Evolution
Would you expect the same quality attributes to be relevant for a young system as for a system that has existed for a while? Why or why not? Can you design your software to be prepared for this?