Software Architects and Testers – Collaboration Required!

Peter Zimmerer

Corporate Technology Software & Engineering (CT SE 1)
Siemens AG
Otto-Hahn-Ring 6
D-81739 Munich
peter.zimmerer@siemens.com

1 Introduction

Today there is common sense that collaboration is a key success factor in software development. Especially agile methodologies emphasize this fact (see [AM01]).

This article focuses on the collaboration between two important stakeholders: software architects who play a very essential role in the development and maintenance of products and systems and testers.

> *"Architects build software systems and testers test these systems" – so why should an architect need to know anything about testing?*

To answer this question section 2 motivates testing by providing a multidimensional view on what testing really is. Then in section 3 practices in testing regarding software architecture are presented which require active involvement and contributions by software architects. This results not only in better quality but also speeds up development by facilitating change and decreasing maintenance efforts.

2 Testing – why? What is the value of testing?

First of all software architects and testers must have a common understanding what testing is all about and appreciate the value of testing. Unfortunately there are still out-dated definitions used for testing (see [IEEE 610.12-1990] or [IEEE 829-2008]) which just limit testing to "dynamically execute a piece of software to detect bugs" but there is a growing perception (at least in the testing community) on the different dimensions of testing (see [GH88] and [IG07]):

- Demonstrate, check: show it works, gain confidence (confirmatory, constructive)
- Detect, search: find bugs as early as possible (investigative, exploratory, destructive)
- Mitigate and reduce risks: "*no risk, no test*"
- Assess, evaluate, and predict: measure quality attributes (e.g. performance)
- Prevent: Control, influence, and drive quality

These dimensions result in the overall mission and value of testing by *providing information related to quality* as stated by Cem Kaner (see for example [CK06]):

> *Testing is an empirical technical investigation of the product / system / artifact / service under test conducted to provide stakeholders with information about the quality.*

3 Practices in testing regarding software architecture

There are several testing practices which require active involvement by software architects:

- Understand the mission and the value of testing (see 2) and promote it in the development team.
- Risk-based testing strategy – actively participate in a product risk analysis workshop as one important stakeholder, perform architecture testing (see [GO08]), and map architectural risks to appropriate testing levels.
- Test-driven development – do not use it only for unit and acceptance testing but rather use the fundamental idea of preventive testing (see [GH88]).
- Design for testability – must be built into the architecture and design (by architects).
- Integration testing – is *the* important testing level to check the architecture.
- Test architectures, test automation – use architecture excellence to support and improve the test architecture and the test automation system.
- Regression testing – use the knowledge about the architecture and design of the system during change and impact analysis to select required regression test cases effectively based on priority, criticality, risks, and frequency of usage.
- Architectural quality – continuously check the internal software quality of the developed system or product and perform adequate code quality management.

4 Summary

Software architects must cooperate closely with testers to define, motivate, drive, and enforce a comprehensive understanding of and attitude to testing and quality in the whole team. Thereby, they address quality from different viewpoints:

- Software architects: build quality into the system (or rather into the architecture)
- Testers: provide information related to quality (feedback)

Software architects must know and understand their job w.r.t. testing: the different dimensions of testing (see section 2) and their required active involvement in the area of testing (see section 3).

Testers must bring to light what software architects should do w.r.t. testing (see 3). Use these practices to drive the needed collaboration and to educate software architects in better testing which definitely improves effectiveness and efficiency in software testing.

References

[AM01] Agile Manifesto: http://agilemanifesto.org/

[CK06] http://www.kaner.com/pdfs/ETatQAI.pdf

[GH88] D. Gelperin, B. Hetzel, The Growth of Software Testing, Communications of the ACM, Vol. 31, Issue 6, June 1988, pp. 687-695

[GO08] Google search for “architecture testing”, http://www.google.de/search?hl=de&q=architecture+testing&meta=

[IG07] ISTQB Standard Glossary of Terms used in Software Testing V.2.0, Dec, 2nd 2007, http://www.istqb.org/downloads/glossary-current.pdf