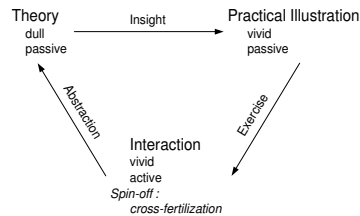


SARCH Course System Architecture

-



Gerrit Muller

Embedded Systems Institute

Den Dolech 2 (Laplace Building 0.10) P.O. Box 513, 5600 MB Eindhoven The Netherlands

gerrit.muller@embeddedsystems.nl

Abstract

This article describes the course Systems Architecting. The course is set up to make the art of system architecting more accessible. The course will address a wide spectrum of issues in relation with system architecture, such as: Processes, Business, Role and task of the system architect (team), Roadmapping, System Architect toolkit, Technical, Skills, and Psycho Social

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

All Gaudí documents are available at:
<http://www.gaudisite.nl/>

version: 2.2

status: draft

February 10, 2011

1 Course Goals

The goal of this course is to enable system architects to mature faster, by giving a wide overview. For non system architects the objective is to improve the cooperation between the system architect and other roles, such as project leader, marketing manager and technology manager.

2 Program

The program purposefully alternates process, business and technology views, see Figure 1. The timing of the program can be adapted to the amount of information and interaction that is needed per subject. However the alterations of subjects will be followed more strictly, because the change in viewpoint is essential for understanding the whole picture.

Session 1 Positioning the System Architecture Process, Product Creation Process
Session 2 Role and Task of the System Architect
Session 3 Requirements Capturing
Session 4 System Architect Toolkit
Session 5 Roadmapping
Session 6 Product Families, generic developments
Session 7 Documentation, reviewing and other supportive processes; The role of Software in complex products
Session 8 BoM presentation
Session 9 Psycho Social side
Session 10 Wrap up, Expectations, How to continue, Evaluation

Figure 1: Program of the course

The structure of the course is shown in figure 2. In other words the theory of the course is that theory, illustration and interaction will alternate.

This alternation follows the general timing as presented in figure 3.

The first step is an interactive exploration of the subject. This exploration is followed by a "broadcast" lecture in which theory and illustration are given. The amount of illustration is "experimental", due to the Philips wide target group; Examples will be based on experience of the trainer, while it is hoped that during the interaction the attendants will bring forward illustrations from their own environment

The interaction is done in 2 steps: an interactive discussion with the entire class and a work session in smaller groups. The instruction for the group work is given during the interactive discussion.

The entire subject is closed by a short collective session with conclusions and evaluation.

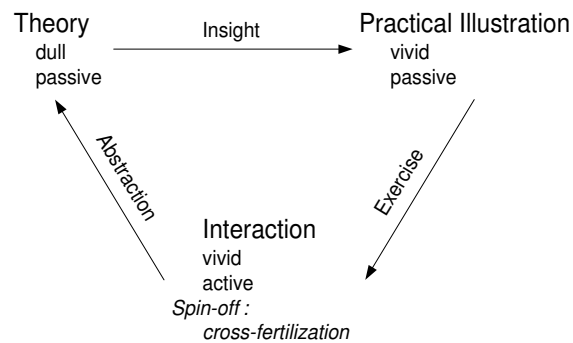


Figure 2: Alternation of theory, illustration and interaction will be used to maximize the educational effect



Figure 3: Timing per subject of the alternating theory, illustration and interaction

3 Rules during the course

The rules of the broadcast part are:

- Please write your questions/remarks/statements on yellow stickers and attach them at the end on the P-flip.
These will be used in the interactive section for discussion and to increase insight.
- Short clarification questions are welcome,
discussion will take place in the interactive part.
- Stupid questions don't exist. Learning is based on **safe** and **open** interaction.
Very individual oriented questions can be referred to a break or after the session.

The rules of the interactive and the practice part are:

- Your contribution is essential.

- Don't monopolize the time, everyone also the quiet people should have the opportunity to contribute;
The facilitator will intervene if the contribution is limited to a small group of participants.
- Respect the contribution of others;
Opinions can't be wrong, difference of opinion is normal and called pluri-formity.
- The course format is highly experimental and based on improvisation, constructive proposals are welcome;
it is your course! Regular evaluations will give the opportunity to influence the rest of the course.

4 Evaluations

Basic part of learning is the evaluation of what has been done. The course will use 3 types of evaluations:

- Personal expectations
- Benefit and Concerns on a regular base
- The CTT evaluation form

The personal expectations are recorded at the very beginning of the course. At the end we look back at these initial expectations. This has a two-way evaluation effect:

Personal Did you start with the right expectation level? Was it realistic? Did you achieve the learning goals formulated in this expectation?

Trainer and CTT Did we communicate the right information to enable people to select this course? Do we apply the right selection criteria?

The benefit and concern evaluation method is based on the basic feedback method, which prescribes to start with formulating the strong points, before addressing the weaker issues. The idea is that improvement is based on building on the strong points and to change with respect to the weaker issues. A side effect is that everyone is forced to think also about the positive aspects, not only about the negative.

The benefit and concern evaluation is done regular, in the beginning with a high frequency, to be able to adapt the course directly.

The benefits and concerns are collected by a brainstorm or on yellow stickers. The rule is that one should always start with a benefit before mentioning a concern.

The benefit and concern method is widely used by CAP Gemini employees, often called B&C or Beer&Chips.

The CTT evaluation form is the "standard" CTT evaluation form which evaluates the different aspects of the course.

References

- [1] Kent Beck. *Extreme Programming Explained; Embrace Change*. Addison-Wesley, Reading, MA, 2000.
- [2] B.W. Boehm. A spiral model of software development and enhancement. *IEEE Computer*, May 1988.
- [3] Frederick P. Brooks. *The Mythical Man-Month*. Addison Wesley, 1975, ca. 1995.
- [4] J. C. DeFoe (Editor). An identification of pragmatic principles. <http://www.incose.org/workgrps/practice/pragprin.html>, 1999.
- [5] Watts S. Humphrey. *Managing for Innovation; Leading Technical People*. Prentice-Hall, 1987.
- [6] INCOSE. International council on systems engineering. <http://www.incose.org/toc.html>, 1999. INCOSE publishes many interesting articles about systems engineering.
- [7] Philip Kruchten. The software architect- and the software architecture team. In *Software Architecture; TC2 First Working IFIP Conference on Software Architecture (WICSA1)*, pages 565–583. IFIP, 1999. This article describes required skills for architect and architecture team; traps and pitfalls; Personality profile based on Myers-Briggs Type Indicator.
- [8] Harold W. Lawson. Function distribution in computer system architectures. *Proceedings of the Third Annual Symposium on Computer Architecture, Clearwater, Florida*, January 1976.
- [9] James N. Martin. *Systems Engineering Guidebook*. CRC Press, Boca Raton, Florida, 1996.
- [10] Gerrit Muller. Positioning the system architecture process. <http://www.gaudisite.nl/PositioningSystemArchitectureProcessPaper.pdf>, 1999.

- [11] Gerrit Muller. The system architecture homepage. <http://www.gaudisite.nl/index.html>, 1999.
- [12] Eric S. Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, 1999.
- [13] Eberhardt Rechtin and Mark W. Maier. *The Art of Systems Architecting*. CRC Press, Boca Raton, Florida, 1997.
- [14] Carnegie Mellon Software Engineering Institute SEI. Software engineering management practices. <http://www.sei.cmu.edu/managing/managing.html>, 2000.
- [15] Roel J. Wieringa. *Requirements Engineering: Frameworks for Understanding*. Wiley&Sons, Baffins Lane, Chichester, West Sussex PO19 1UD, England, 1995. Introduction and Chapter 2 **Systems** provide a clear definitions of systems and related systems engineering terminology.
- [16] Jan Gerben Wijnstra. Quality attributes and aspects of a medical product family. submitted to the Software Track of the HICSS-34, January 2001 http://nlww.natlab.research.philips.com:8080/research/swa_group/wijnstra/ExternalPublications/hicss34/HICSSPaperJGW.pdf, 2000. This article describes the flow of non functional requirements to (software) implementation.

History

Version: 2.2, date: April 24, 2002 changed by: Gerrit Muller

- updated abstract
- changed program into figure
- changed status in draft

Version: 2.1, date: April 24, 2002 changed by: Gerrit Muller

- minor editorial changes

Version: 2.0, date: October 5, 2001 changed by: Gerrit Muller

- added section "Course goals"

Version: 1.2, date: october 2, 2000 changed by: Gerrit Muller

- program "Requirements" and "Roadmapping" exchanged in the programming
- layout change

Version: 1.1, date: august 9, 2000 changed by: Gerrit Muller

- added references to the bibliography
- made the document not course specific
- removed exercises from this document

Version: 1, date: january12 2000 changed by: Gerrit Muller

- changed the course dates to february 2000
- shifted the "Product Creation Process" to monday morning
- shifted "roadmapping" to tuesday morning
- added exercises
- removed empty subsections
- references added

Version: 0, date: november 2 1999 changed by: Gerrit Muller

- Created, no changelog yet