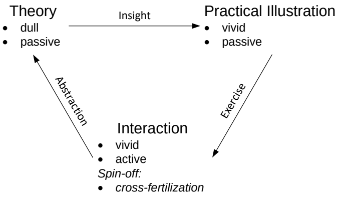
Information Masters Course The Context of Embedded System Design

-



Gerrit Muller
University of South-Eastern Norway-NISE
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
gaudisite@gmail.com

Abstract

The Masters Course The Context of Embedded System Design is a course for students following the masters "Embedded Systems". The course material is based on the SARCH course *Systems Architecting*. However, more and shorter exercises are added, and a common case is used thorughout the course.
The course addresses a wide spectrum of issues in relation with system architecture, such as: processes, business, role and task of the system architect (team), generic Developments (re-use, platforms) requirements, roadmapping, and skills.

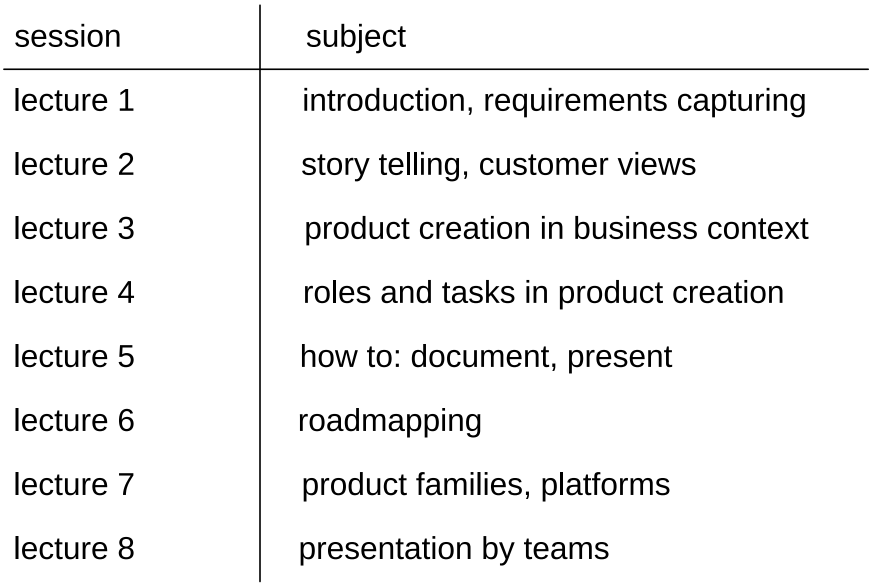
version: 0.1 status: draft September 6, 2020

1 Course Goals

The goal of this course is to create awareness of the context of embedded systems design: customer side, application, business, and creation side, processes, and organization.

2 Program

The program purposefully alternates process, business and technology views. 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	subject
lecture 1	introduction, requirements capturing
lecture 2	story telling, customer views
lecture 3	product creation in business context
lecture 4	roles and tasks in product creation
lecture 5	how to: document, present
lecture 6	roadmapping
lecture 7	product families, platforms
lecture 8	presentation by teams

Figure 1: Program of the Masters Course The Context of Embedded System Design

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.

3 Exercises

The casework is introduced in Figure 3.

The exercises for the lectures are:

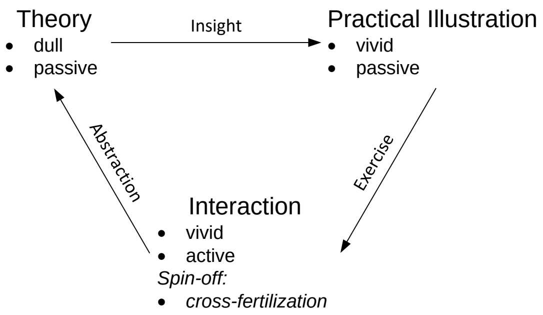
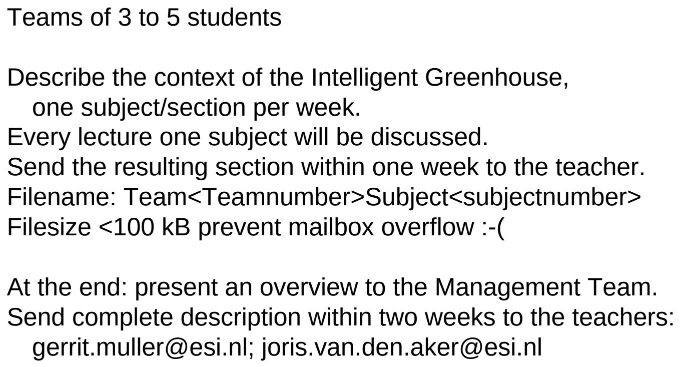


Figure 2: Alternation of theory, illustration and interaction will be used to maximize the educational effect


Teams of 3 to 5 students

Describe the context of the Intelligent Greenhouse,
one subject/section per week.
Every lecture one subject will be discussed.
Send the resulting section within one week to the teacher.
Filename: Team<Teamnumber>Subject<subjectnumber>
Filesize <100 kB prevent mailbox overflow :-(

At the end: present an overview to the Management Team.
Send complete description within two weeks to the teachers:
gerrit.muller@esi.nl; joris.van.den.aker@esi.nl

Figure 3: Introduction to the case work

3.1 Requirements

- 1 Describe a “Intelligent Greenhouse”: What does it look like, what can it do?
- 2 Identify Stakeholders and concerns
- 3 Discuss the technological opportunities and challenges
- 4 Make a key driver map

3.2 Story Telling

- 1 Create a story
- 2 Improve the story, with the criteria for stories in mind

- 3 Derive a case description from the story
- 4 Make a design to satisfy the case description

3.3 Product Creation

- 1 Identify the processes within your own company.
- 2 Make a design of the product
- 3 Make a work breakdown structure
- 4 Propose an organizational structure, quantify the size of the groups.

3.4 Roles and Tasks

- 1 Determine the most critical system functions and performance aspects
- 2 Propose an integration plan
- 3 Perform a risk assessment
- 4 Improve the organizational structure

3.5 Documentation and Presentation

- 1 Analyse the costs of the product creation, manufacturing and sales
- 2 Analyse the income
- 3 Make multi-year business forecast
- 4 Make a presentation outline for the presentation to the Board of Management

3.6 Roadmapping

- 1 Identify Market trends
- 2 Identify Technology Trends
- 3 Make a product roadmap proposal
- 4 Integrate Market, products, technology into 1 roadmap and identify Process and People issues

3.7 Product Families

- 1 Identify the members of the product family

- 2 Identify the synergy between the members of the family
- 3 Identify the member specific functionality
- 4 Propose a balanced product family approach

4 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 pluriformity.

- 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.

5 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: 0.1, date: November 16, 2005 changed by: Gerrit Muller

- changed the case from Multi-Media Mate to Intelligent Greenhouse
- added introduction to case work

Version: 0, date: November 16, 2004 changed by: Gerrit Muller

- created by modifying ShortSARCH course description