

# Experiences of Teaching Systems Architecting

Gerrit Muller  
Embedded Systems Institute  
[Gerrit.muller@embeddedsystems.nl](mailto:Gerrit.muller@embeddedsystems.nl)  
P.O. Box 513  
5600 MB Eindhoven  
The Netherlands  
<http://www.extra.research.philips.com/natlab/sysarch/>

**Abstract.** The experiences of four years teaching systems architecting are described. The duration of the course systems architecting is 5 days. The target audience consists of (potential) architects and stakeholders that cooperate intensely with the architect, such as project leaders, product managers, and group leaders. The course has been given 23 times in the period November 1999 to January 2004. The maximum number of participants is 16.

This paper discusses the course content and the course objectives, course materials, the course format, some course statistics, the expectations of the students up front and the evaluation at the end, the follow-up and the longer term results, the derived course for managers, the lack of visibility of system architects, and the broader education context that is required for a systems architecting curriculum.

## Course Objectives and Course Description

The course is based on a specific vision on the role of a system architect. According to that vision a system architect is responsible that a system fits in the context and that a system has a sound design. The architect works together with many other people in the product creation, such as project leader, product manager, and more specialized engineers. The architect needs a significant amount of technical and non-technical know-how and a lot of skills to perform this job.

In the area of System Design, Systems Engineering and Systems Architecting most of the courses and literature is based on the Systems Engineering work, for instance on the basis of (Martin1996) or (Blanchard 1998). Systems Engineering builds on the experience mostly from the aerospace and military industry. In this domain the development lifecycles are many years and safety and reliability requirements are high. Companies in with very dynamic markets, such as computing and consumer electronics, struggle with methods and processes to do Systems Architecting. The balance between time to market, cost and dependability is different from the conventional Systems Engineering field. The systems architecting course (SARCH) is positioned for systems development in the more dynamic markets. The course content extends the conventional Systems Engineering methods with means to cope with the more dynamic markets.

The SARCH course has the following objectives:

- Make participants aware of the non-technical context that plays a role in architecting, such as organization, process, people, market and business.
- Provide insight in the role of the system architect.
- Stimulate participants to be more outward focused and to be more constructive critical.

- Provide practical approaches that can be deployed by the participants in their daily work.
- Stimulate architects to be more visible in the organization.

The goal of the course is **not** to transform all the course participants in full-blown architects. The course provides a lot of value if the objectives are achieved. It is a bonus if from the many participants a few really grow into a system architect operating at the business level.

The course follows the principle of viewpoint hopping. Time boxing is applied: a half-day is reserved per viewpoint. Nine viewpoints are used:

1. Organizational position of the System Architecture Process in the Product Creation Process.
2. Role and Task of the System Architect
3. Requirements Capturing
4. System Architect Toolkit
5. Roadmapping
6. Product Families, generic developments
7. Documentation, reviewing and other supportive processes
8. The role of Software in complex products
9. Psycho Social side

The tenth half-day is used for the wrap up, to look back at the initial expectations, to look forward how to deploy the subject matter, and to evaluate the entire course.

## Course material

Every viewpoint is one module of the course. For every module a few presentations and an exercise are bundled into a single file per module. An article version, where a lot of text is added to the presentation figures, is available for each presentation. All bundled presentations and bundled articles are given to the participants on paper at the start of the course.

The course material is a deliverable of the Gaudí project, which can be found at: <http://www.extra.research.philips.com/natlab/sysarch/>. The ambition of the Gaudí project is to make the art and emerging methodology of systems architecting more accessible and to transfer this know-how and these skills to a new generation of system architects. The course material itself can be found at: <http://www.extra.research.philips.com/natlab/sysarch/SARCH.html>.

All students get a copy of “The Art of Systems Architecting” (Rechtin 1997), as recommended background material. This book is not directly used during the course.

## Course format

Theory, example and practical work are interleaved. The theory is an abstract, but powerful description of the system architecting know-how. Practical examples are very important to appreciate the theory. A wide variety of examples is used to illustrate the know-how, including its limitations. The students are stimulated to participate actively. Every time-box starts with a question or provocation to the audience that triggers the participants to reflect on a viewpoint. Halfway the time-box an exercise is done in teams of three to four participants. Most exercises address the viewpoint under discussion in the domain of the participants themselves. For example: “Describe **your** organization”, and “Present **your** roadmap”. Only a short amount of time is available for the exercise: 40 minutes. The results of the exercise have to be presented plenary in 5 minutes and using only 1 or 2 flips. Every presentation is followed by a discussion and feedback from participants and the teacher.

The core idea behind this course format is that effective transfer of know-how requires an active attitude from the audience. The short time boxes provide fast feedback. Such an incremental approach is also advocated as a preferred product creation approach in fast changing dynamic environments.

### **Some Course Statistics**

The course has been given 23 times in a period of 50 months. 16 Courses were general courses within Philips, with participants from all over Philips, from Philips partners and also from Philips spin-offs. 7 Courses were *special* courses, given on location for a single product division, such as Medical Systems, Semiconductors and Consumer Electronics. The average number of participants was 15.

The majority of the participants were experienced designers, subsystem architects, who have a more limited but multi-disciplinary scope, or mono-disciplinary architects, such as software architects. Also quite some researchers participated in the course. A few project leaders and group leaders participated. A reasonable number of the participants had already the function system architect, although carrying the title and actually being a system architect is certainly not the same.

Some other courses are derived from the SARCH course: a two-day version for management teams (given once) and a stakeholder module in the Embedded Systems Architecting (ESA) course that has been given about 14 times.

### **The Expectations Up Front and the Evaluation at the End**

Immediately at the beginning of the course the students articulate their personal expectations of the course, by writing them on a yellow sticker. The expectations are collected on a flip over on the wall classroom. At the end of the course an evaluation is done of the entire course. This evaluation consists of a plenary round and a standardized evaluation form. During the plenary round every participant has to tell:

- The expectations at the beginning of the course.
- How far have the expectations been met?
- What are the benefits?
- What are the concerns?

The evaluation of the expectations is valuable for the student (did I have realistic expectations, did I select the course carefully?) and for the teacher and the organizers (do we address the needs of the participants, do we provide clear course information?).

**Expectation evaluated as highlights.** Very common expectations are insight in the architecting process, insight in the architect role, insight in the architect prerequisites, means to capture requirements, and means to make roadmaps. In general these expectations are met fully.

**Expectations that have not been met.** Common expectations that participants have, but that are not fulfilled at the end are: to learn system design methods, to learn architecture styles or patterns, and to get advise on specific tools. Only one module of the course addresses all these subjects, where tens of modules are required to address the existing body of knowledge. Most participants who started with such expectations conclude at the end that they have learned even more useful things than they expected.

**Benefits.** Most benefits that are mentioned are directly related to the expectations that have been

met. Additional benefits that are frequently mentioned are: meeting other people that struggle with the same kind of problems, confirmation of working methods, seeing other products and architectures, obtaining a broader perspective, and trigger for reflection. The format of the course, very interactive, and with many practical examples is a common benefit.

**Concerns.** The absolute top hits in the concerns are “How to apply the lessons in my environment”, “Distance between theory and practice” and “Do we have the right architecture”.

The first two concerns are related: purely academic methods don’t fit in the daily hectic life of product development. The key message behind the more theoretical models is that these models can be used as a source of inspiration. If you have a vision of an ideal method and you can identify the tension of reaching the ideal, then the solution is often close by, see TRIZ (Altshuller 2000). The second problem in applying the lessons learned is that most stakeholders in the immediate environment of the participant do not share these same lessons. The implementation of system architecting methods requires a change management process. That takes time and requires sufficient supporters.

The concern “Do we have the right architecture” is often triggered by the many external viewpoints provided during the course. The participants discover during the course that many architectural decisions are based on limited internal considerations, varying from cost or effort reduction to individual hobbyhorses of managers or engineers. Numerous times the articulation of the customer key drivers is an eye-opener that triggers the doubts about the current architecture.

**Formalized evaluation.** The training department uses a standardized evaluation form for all courses. The participants score all aspects on a scale from 1 to 5, where 1 corresponds with the most negative meaning and 5 with the most positive. The only exception is the question about foreknowledge, where 1 means too difficult, 3 is a perfect match, and 5 means too easy. These numbers can be compared with the evaluation of other courses, where the range of 3.5 to 4 is characterized as *above average*, from 4 to 4.5 as *very good*.

	special		regular	
	Mean	Sd	Mean	Sd
1 In my case, the objectives of the course have been:	4,0	0,7	4,1	0,8
2 All the important subjects are offered in the course.	4,0	0,8	4,0	0,8
3 The contents of the course had sufficient depth.	3,9	0,9	3,8	0,9
4 The level of the course combined well with my foreknowledge of this subject.	3,4	0,7	3,3	0,7
5 I have learnt many new things during the course.	3,7	1,0	3,9	1,0
6 The way of learning and working in the course was:	4,1	0,9	4,1	0,8
7 The various subjects in the course linked-up well.	4,1	0,8	4,1	0,8
8 The contents of the course were well-supported by the written material.	3,9	1,0	4,0	0,9
9 The exercises during the meetings combined well with the subject-matter.	4,3	0,7	4,2	0,8
10 The feedback on the exercises was:			4,1	0,7
11 The practical exercises were considered as useful:	4,1	0,8	4,0	0,8
12 What I have learnt in the course, I can use in actual practice.	4,2	0,7	4,1	0,8

**Table 1. Outcome of the formalized evaluation.**

Table 1 shows the outcome of this formal evaluation. The results are shown separate for the special courses, based on 91 participants, and for the regular courses, based on 241 participants. This course from training department point of view is highly successful. This outcome shows that the participants do appreciate the course a lot:

- The participant objectives are met (1).

- Important subjects are offered (2), and well linked-up (7).
- More depth can be offered (3) and more foreknowledge may be assumed (4).
- The course does not provide many new things (5). The added value of the course is in the overview and the integration of many existing tools and methods. A very high score on this question would indicate too much distance to today's practice.
- The teaching approach and exercises are highly appreciated (6, 9, 10, 11).
- The material is appreciated, but can be improved (8).

A very important conclusion of this evaluation is that the course content can be used in actual practice (12). In the next section the results of a questionnaire will be discussed, addressing in retrospect the question what has been applied.

## The Follow-up and the Longer Term Results

The last afternoon of the course is used to discuss the deployment of the architecting methods in the dynamic work environment. The students are encouraged to look back at all viewpoints that passed during the week. Then they are asked to make a personal short term plan: "What are you going to do next Monday?" and a long-term roadmap, with a time horizon of years. For both the short-term plan as well as for the roadmap it is recommended to link the personal vision and ambition to the market and business needs. After one hour of individual work a plenary discussion follows, where the participants exchange ideas and provide mutual feedback.

The purpose of the short-term plan is to stimulate the participants to introduce the architecting methods in an incremental way. It is frequently emphasized in this discussion that convincing other people is done most effectively by *working by example*. Starting a near religious discussion of how to work mostly results in a lot of frustration, while practical examples, do it, can do wonders. Examples of short-term intentions are: modularize and reduce the documentation, create a zero-order roadmap, apply the budget method and create the key driver graph.

The long-term vision stimulates participants to think about the longer-term business prospects and the personal ambition in that context. A roadmap can be a practical means to put the constrained situation of today in a broader perspective. By identifying key success factors and bottlenecks next steps for the individual can be identified.

Once per year a system architecting event is organized for all previous participants of the course. Besides the course participants also experienced architects and involved managers are invited for this event. In 2001 this event was visited by 50 participants, the 2002 event attracted 80 people and in 2003 the event has grown to 100 visitors. The first event was organized with a full program. The later two events were organized in the open space format, see: <http://www.openspaceworld.org/>.

The feedback obtained during the architecting events shows a great appreciation for the course and the attempt to improve the system architecting methods company wide. At the other hand it is also very clear that the change towards the system architecting approach is very slow and tedious process. Most former course participants apply only a few of the methods from the course, such as the documentation or applying the CAFCR model. The CAFCR model, described in (Obbink 2000), is used as framework throughout the course. It uses 5 views (*Customer Objectives* view, *Application* view, *Functional* view, *Conceptual* view, and *Realization* view) to describe the architecture.

**Questionnaire.** In March 2004, triggered by the preparation of this article, a questionnaire was sent to 345 SARCH participants. The questionnaire contained 7 open questions:

1. In what year did you follow the course?
2. What are in retrospect your most important learning points?
3. What subjects, methods or tools from the course did you apply in practice?
4. Do you have any new suggestions for the course?
5. Do you have suggestions for the yearly follow-up system architecting event?
6. Do you have interest in a 1 week course zooming in on more specific system design methods, based on (Muller 2004).
7. Any other feedback?

The response was 40 filled in forms, 21 participants could not longer be reached because they moved and their new address was unknown. The average time between following the course and filling in the questionnaire was 2 ½ years.

**The most important learning points** were: Role of the System Architect (9 times), Roadmapping (7 times), Broad Scope of the Architect (7 times), The Use of Multiple Viewpoints (5 times), Requirements (5 times), Generic Developments (4 times) and Key Drivers (3 times). Note that Key Drivers are used as one of the specific methods to capture requirements and relate them to design decisions.

**The subjects applied in practice** were: *Roadmapping* (12 times), *Role of the System Architect* (4 times), *The Use of Multiple Viewpoints* (4 times), *Requirements* (3 times), *Stakeholders* (3 times) and *Key Drivers* (3 times).

A quote from the participants: “It is difficult to say exactly which methods or tools I'm using. The most important aspect of the course is a good and pragmatic overview of the architect in the business. I have heard people say that such overview can be considered as a collection of open doors; however, I feel the opposite. It is important to have good insight in what exactly your role is. Without the course it is very difficult to get it well defined. Being aware of your role is crucial in fulfilling it properly.”

**Suggestions for the course**, some quotes from the participants:

“Maybe about risk analysis, there is a good tool in the test course that could help. It requires estimating each component based on its relationships with others, how often it is used, size, complexity. Based on that you add complexity + size + reuse = chances of defect frequency of use + number of interfaces with other = system impact. Crossing both types of info help for risk analysis.”

“The course is quite complete and full already! Risk is that when you add a subject, an existing subject will suffer!”

“What I had expected before I entered was a more technical content. I was not disappointed with the lack thereof but I do think there is a need for some more technical sessions. Especially the module about the growing role of software could be explored further. I believe that in systems design there is often a too early acceptance of what the software content is and how complex it is going to be. Starting out with such assumptions ("SW is hard", "we need xx people", "we must have this system, OS, method, etc"), combined with department or project Management that does not understand the software is often the basis for developments that take too long, cost too much and deliver too little.”

**Suggestions for the yearly Systems Architecting Event**, some suggestion from the participants:

“On the first question, whether I would like to organize the system architecting event in another way than 'open space': No, I really like the concept. On the second question then, what

suggestions do I have for new themes: Some nice ones might be to stimulate architects to discuss how they think about "Making profit", "Market and Vision", and some more business-oriented themes. I'm not sure however if that is something that architects want to discuss."

"More structure. E.g. 3 or 4 blocks, each with a key-note speaker to stimulate the audience and initiate discussion."

"I would use such a event to create better awareness of the system architect's role outside the SARCH community. I.e. invite development and project managers, present examples where SARCH saved the day."

Most reactions show an appreciation of the open space format. However some more focus, by means of individual preparation or more keynote speeches is also being asked for.

**Interest in more specific system design methods course:** 14 people (35%) state an unconditional interest in such a course, while 4 people have no interest. The remaining 22 people have a varying degree of interest: 3 are very interested, 2 state "perhaps", 5 state an interest of other people, 7 are interested but are constrained by cost or time, and 5 are interested but do foresee an inhibiting constraint, such as cost or time.

**Other feedback,** some quote from the participants: "do you have a course for stakeholders to explain to them the position of the architect?" Many participants struggle with their organizational context. During the course this is also mentioned many times as a concern.

## Course for Managers

The main bottleneck in the wider introduction of systems architecting methods is the lack of support by the immediate environment of the architect. Two factors contribute to this problem of lacking support: Lack of systems architecting know-how and awareness of the managers, and lack of visibility of the existing architects. In this section the management part is addressed, in the following section the visibility of architects is addressed.

A special shorter version of the SARCH course is developed to address the managers, the so-called Management SARCH (MSARCH). This course condenses the subject matter in 1 ½ day. An additional module is added addressing the Human Resource management aspects of system architects, addressing issues such as function description, motivation, appraisal, education and selection. In the condensed form several modules are combined: role of the architect and the psychosocial side, organizational position and product family development, and roadmapping and requirements capturing. The system architect toolkit, supporting processes and the role of software in complex systems modules are skipped entirely. On purpose several exercises are also used in the MSARCH. If the exercises are left out a glossy and nice overview remains, where the participants miss the nuances of the complex hectic reality. The exercises are absolutely necessary to appreciate the system architecting methods and the tension involved in implementation of the methods.

The MSARCH has run once, with a complete management team plus a few senior architects, together about 12 people. The reaction was quite positive. However, it turns out to be difficult to motivate other management teams to follow this course. One of the main roadblocks for the managers is the duration of 2 days. Two days is perceived as a very large investment. From the teacher viewpoint the investment of 2 days is a sign of commitment. Presumably management teams that don't want to invest 2 days are not ripe for architecting methods yet.

## **Architect Visibility**

In the previous section it was stated that introduction of systems architecting methods is limited by the visibility of the existing architects. The lack of architects at management level causes an unbalance in the decision-making: the link with the product specification and design is very weak. The consequence is that many business decisions are being taken, where the specification and design risks and opportunities are not known. This situation stays the same, unless system architects build up credit at managerial level. This credit can only be build up if the system architects are knowledgeable and sufficiently visible. The system architect will have to make the translation from the technical world to the managerial interests, such as costs and benefits. By nature architects tend to have a rather introvert personality. I have often seen architects arguing: “If this manager does not understand it after my explanation, then he is incompetent. I don’t waste any more time on him.” This kind of attitude blocks the necessary communication with the management team.

An important objective of the SARCH course is to make system architects aware of the visibility problem, to make them aware of their own attitude and to provide them with practical approaches to increase the visibility. The concrete result of this objective is that the Thursday afternoon exercise has been expanded into a “presentation to higher management team”. Teams of 4 participants have to prepare a presentation of 10 minutes about an architectural issue to a management team 2 or 3 levels above them. The remainder of the participants and the teacher form the critical, and active management team. This exercise is highly appreciated by many participants, because it addresses a problem that most of them perceive. The exercise helps them to understand the problem and helps them to cope with the problem with some practical advice and experience.

## **Teachers**

The author has given the SARCH course many times. He was the only teacher for 2 ½ years. At this moment other teachers have given 3 courses. The experience of the other teachers is that giving this course requires quite some experience in architecting. The course format and the material are available and usable for other teachers. However, the illustration with practical examples must be done on the basis of the experience of the teacher. The feedback of the other teachers is that finding the right examples at the right moment is the most challenging part when teaching the SARCH course.

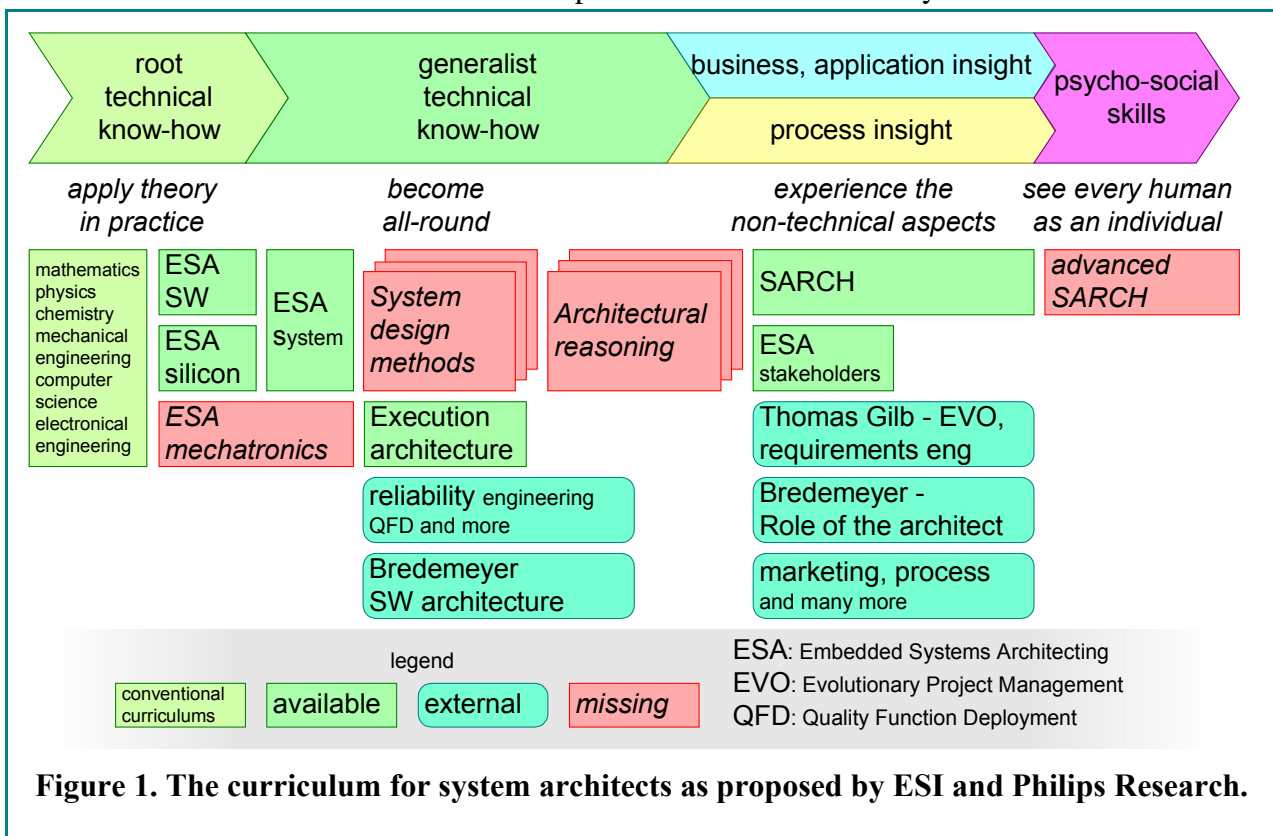
## **Towards a Systems Architecting Curriculum**

The popularity of the SARCH course shows the need for this type of course. The focus on the non-technical aspects of systems architecting is appreciated. The multi-disciplinary design problems, however, are neither addressed by this course, nor by most other courses and curricula. In fact all the steps between an academic study and the SARCH are not addressed. In the world not many educational programs are available to help designers to grow into an architect. The feedback from participants and the requests to the training departments show the need for a more complete Systems Architecting curriculum.

The Embedded Systems Institute (ESI) <http://www.embeddedsystems.nl/> together with Philips Research developed a vision for a more complete curriculum for system architects, as shown in Figure 1. At the top of this figure the growth path of a system architect is shown. This growth path is observed in many senior architects and provides a framework for the curriculum. The first phase in becoming an architect is to be a good mono-disciplinary designer: build up in



depth experience in product creation. In the next phase the technical know-how is extended, by learning more about other disciplines. In practice many system design problems are caused by the boundaries between disciplines: misunderstanding and lack of understanding between designers from different disciplines. A growing field of multi-disciplinary design methods becomes available to cope with these problems. Once the designer is sufficiently experienced and broad in technical know-how, he discovers that many product creation problems are caused by, and need to be solved in, the non-technical domain: market, business, process, people and organization. Disciplines such as requirements analysis and marketing research are examples of this other domain. Although the architect does not have to become a marketing manager, he must be able to understand and communicate clearly with the marketing manager. In the last phase the architect becomes much more interested in the human factor. Every stakeholder is in fact an individual human being, with his or her individual behaviour and characteristics. Insight and skills based on the human sciences can help the architect tremendously.



**Figure 1. The curriculum for system architects as proposed by ESI and Philips Research.**

In the diagram courses are shown that fit in that specific phase of the development of the architect. Some of these courses are already available from Philips or ESI, such as the ESA and SARCH courses. Some courses are available on the market, such as the Bredemeyer courses on software architecture (Bredemeyer 2003) and Role of the architect (Bredemeyer 1999), and the Gilb courses about Evolutionary development and requirements engineering (Gilb 1999). In the figure also a large set of courses is shown that is not yet available: System design methods, methods that help to achieve a limited set of system objectives and architectural reasoning, methods that help in reasoning over many different system objectives. The Embedded Systems Institute will develop these courses in an incremental way, together with the Philips training department and other partners.

## References

- Genrich Altshuller. *The Innovation Algorithm; TRIZ, systematic innovation and technical creativity*. Technical Innovation Center, Worcester, MA, 2000. Translated, edited and annotated by Lev Shulyak and Steven Rodman.
- Benjamin S. Blanchard and Wolter. J. Fabrycky. *Systems Engineering and Analysis*. Prentice Hall, 1998.
- Dana Bredemeyer and Ruth Malan.. *Role of the software architect*. <http://www.bredemeyer.com/role.pdf>, 1999.
- Dana Bredemeyer and Ruth Malan. *The visual architecting process*. [http://www.bredemeyer.com/pdf\\_files/WhitePapers/VisualArchitectingProcess.PDF](http://www.bredemeyer.com/pdf_files/WhitePapers/VisualArchitectingProcess.PDF), 2003.
- Thomas Gilb. *Competitive engineering*. [http://www.pimsl.com/TomGilb/Competitive\\_Engineering\\_M.pdf](http://www.pimsl.com/TomGilb/Competitive_Engineering_M.pdf), 1999.
- James N. Martin. *Systems Engineering Guidebook*. CRC Press, Boca Raton, Florida, 1996.
- Gerrit Muller. *CAFCR: A Multi-view Method for Embedded Systems Architecting; Balancing Genericity and Specificity* (PhD thesis), <http://www.extra.research.philips.com/natlab/sysarch/ThesisBook.pdf>, 2004
- Henk Obbink, Jürgen Müller, Pierre America, and Rob van Ommering. *COPA; a component-oriented platform architecting method for families of software-intensive electronic products*. [http://www.extra.research.philips.com/SAE/COPA/COPA\\_Tutorial.pdf](http://www.extra.research.philips.com/SAE/COPA/COPA_Tutorial.pdf), 2000.
- Eberhardt Rechtin and MarkW. Maier. *The Art of Systems Architecting*. CRC Press, Boca Raton, Florida, 1997.

## Biography



**Gerrit Muller.**

Gerrit Muller received his Master's degree in Physics from the University of Amsterdam in 1979. He worked from 1980 until 1997 at Philips Medical Systems as system architect. From 1997 to 1999 he was manager System Engineering at ASML. From 1999 - 2002 he worked at Philips Research. Since 2003 he is working as senior research fellow at ESI (Embedded Systems Institute). The main focus of his work there is to work on System Architecture methods and to enable education of new System Architects. Special areas of interest are:

- ways to cope with the exponential growth of size and complexity of systems. Examples of methods to address the growing complexity are product lines and composable architectures.
- the human aspects of systems architecting (which in itself is a crucial factor in coping with the above mentioned growth)

At this moment he is finishing his PhD thesis (Muller 2004).

All information (System Architecture articles, course material, curriculum vitae) can be found at: <http://www.extra.research.philips.com/natlab/sysarch/>