How to accelerate the development and experience acquisition of new Systems Architects?

White Paper Resulting from Architecture Forum Meeting

March 27, 28, 2012, Stevens Institute of Technology, Hoboken, NJ - USA

Edited by:

Dr. Gerrit Muller, Buskerud University College, Embedded Systems Institute

Mr. Peter Korfiatis, Stevens Institute of Technology

Input was provided by the following participants in the Architecture Forum:

Name	Organization
Rob Cloutier	Stevens Institute of Technology
Else Dalby	Kongsberg Group
Eirik Hole	Stevens Institute of Technology
Lars Ivansen	Micronic Mydata AB
Kees Kooijman	FEI Company
Peter Korfiatis	Stevens Institute of Technology
Hugo van Leeuwen	FEI Company
Tim Majeski	Lutron Electronics

Name	Organization
Gerrit Muller	Buskerud University College/ESI
Daniel Opstad	GKN Aerospace Norway[1]
Mike Pennotti	Stevens Institute of Technology
Rolf Siegers	Raytheon Company
Martin Simons	Daimler
Eldar Tranøy	Aker Solutions
Lubos Tuma	FEI Company
Jon Wade	Stevens Institute of Technology

Published on March 31, 2014

[1] Currently at Kongsberg Group

1. Introduction

Globally, organizations suffer from a shortage of systems engineers and architects [NDIA 2010]. In this paper, we use the term systems architect; however, most of the discussions are valid for systems engineers as well. On top of the shortage, many “older” organizations foresee a retirement wave of their current systems architects or people with system overview. Typical systems architects have developed themselves over a long period of time (10-15 years [Dubey 2006]), and hence typically are middle age or older.

A competent architect needs *knowledge*, *skills*, and *experience*. Potential architects can acquire *knowledge* in multiple ways, e.g. reading books, papers, participating in network activities, e.g. visiting conferences, or following courses. They need training to acquire *skills*, or they can develop *skills* over time by *experience*. Effectiveness and productivity of architects comes from the proper use of *knowledge* and *skills*; that is what we call competence. *Experience* is a crucial part of developing a *competence*; see for example [Squires 2011a].

In this meeting, we discussed the question: “How can we accelerate the development and experience acquisition of new Systems Architects?” We discuss here the needs of future architects. The meeting took place at Stevens Institute of technology. A cohort of young systems engineering students from Buskerud University College (BUC) in Norway were following their international semester at Stevens when the meeting took place; we invited some of these students for the meeting. These students work part-time at a Norwegian company and study systems engineering simultaneously. These BUC students form a good example of the target group of experience acceleration. This meeting combined the usual mix of experienced architects and academics with a strong industrial background and a small group of engineers on their way to broader systems roles.

The academics working in research and education presented various approaches to accelerate acquisition of experience:

- The Experience Accelerator (A Systems Engineering Research Center (SERC) research project)
- Reflective Practice (a mandatory course at BUC as part of the master in systems engineering)
- project-based learning (A Leadership program for early-career, mid-career acquisition professionals, and aspiring technical leaders engineers at Stevens Institute of Technology)

We will briefly capture these presentations and the discussions from the forum regarding experience acceleration.

2. The Experience Accelerator

The Systems Engineering Research Center (SERC) studies the use of serious gaming to help potential systems engineers to accelerate the acquisition of experience. The research project formulated the following hypothesis and goals:

Hypothesis: *By using technology, we can create a simulation that will put the learner in an experiential, emotional state and effectively compress time and greatly accelerate the learning of a systems engineer faster than would occur naturally on the job.*

Goals: To build insights and “wisdom” and hone decision-making skills by:

- Creating a “safe”, but realistic environment for decision making where decisions have programmatic and technical consequences
- Exposing the participants to job-relevant scenarios and problems
- Providing rapid feedback by accelerating time and experiencing the downstream consequences of the decisions made

The initial focus of this program will be on the Systems Engineering Executive Level skills of a DoD Lead Program Systems Engineer necessary to effectively manage complex systems throughout their lifecycle from an acquisition/acquirer viewpoint in a typical Project

Management Office (PMO). This translates in a targeted competency of problem solving and recovery approach:

- Identifying the actual/root cause problems amid often-conflicting information.
- Marshaling the resources needed to solve problems.
- Recognizing the problems that have the most impact to the overall system and appropriately prioritizing plans for solving them.
- Making recommendations, using technical knowledge and experience, by developing a clear understanding of the system.
- Identifying and analyzing problems using a systems approach, weighing the relevance and accuracy of information, accounting for interdependencies, and evaluating alternative solutions.

The program formulated the following desired capabilities and features for the experience accelerator:

- Relevant, Authentic Experiences
  - Experiential focused…incorporates experience base of DoD Chief Engineers
  - High-fidelity simulations of complex system development
  - "soft skills" tailored to a technical perspective
  - Skill level adjustment, initial focus on expert level
- Cost Effective, Available and Open
  - Approximately 1 hour time limit for each session
  - Low Server utilization per client user…highly scalable
  - No special client hardware or administrative needs
  - Open architecture + Open Source Software with no-cost licensing

- User-friendly tool-set in parallel development

The research project had been working on a prototype of the accelerator for one year when the forum meeting took place in March 2012. At that moment, there was a working implementation based on an open architecture, described in [Wade 2012]. Figure 1 shows a screenshot of a conversation between a player and a game-based official.

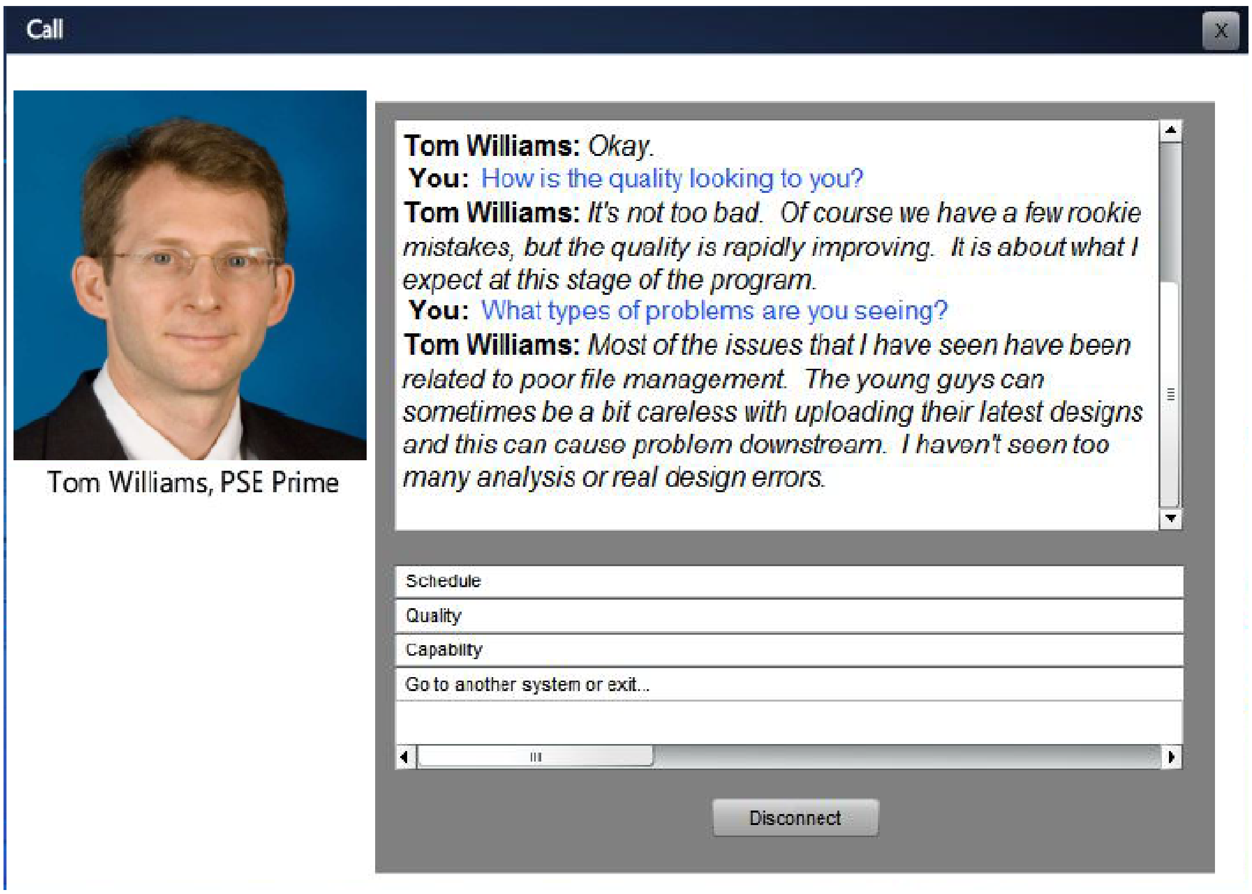

Figure 1. Screenshot of the Experience Accelerator showing a dialog between player and a game-based (an agent as part of the simulator, not a human) official.

During the discussion, we needed to clarify what we discussed: the vision of the experience accelerator, or the first instance shown as demonstrator. The demonstrator, due to funding reasons, had moved the prototype development in the direction of a single person experience using limited time. In particular, this prototype had been built for acquisition. There is quite a distance between this prototype and the vision.

We used the following question to discuss the experience accelerator:

- How will such an approach fit your domain
- How will this approach work? Will the game engage people?
- How does it serve the purpose of accelerating systems engineering competences, e.g. soft skills or politics?

The demonstrator seems to be more appropriate when an organization needs to push a large number of employees through a program in a rush. Some participants saw as concern that employees will see this as an impersonal approach; the company puts employees in front of a computer screen for a few hours rather than providing a mentor.

The demonstrator is mostly focused on project management, which is somewhat domain (e.g. defense, automotive, subsea) agnostic. In general, participants were hesitant in the fitness of the demonstrator for their domain. At least quite some content development is needed to make it fit. When the content includes more technology and application, then it will become more domain specific.

Most participants see the need for a version that is closer to the vision, e.g. multi-player, realistic interaction (e.g. real-time voice), and more advanced visualization to allow users to feel more immersed.

A central question is whether the benefits of the content development can be in balance with cost of content development. Is there a sweet spot in the amount of content development?

The discussion about the purpose of acceleration resulted in a broader discussion on its purpose:

- Is the purpose to get *more* or *better* systems engineers?
- Does better mean *faster*, *less costs*, or *higher quality*?

- Is the purpose to get better *individuals*, *teams,* or *organizations*?
- Who is the target audience, e.g. to develop *graduates* into *engineers*, or *engineers* into *junior systems engineers*, or *junior* into *senior systems engineers*?

Most participants had their own experiences in games used for competence development. For example, the beer game in supply chain management is well known; see http://en.wikipedia.org/wiki/Beer_distribution_game. Philips used the "electromotorenspel" for decades in their courses in business and product development. This game is played for 5 days, where every team manages a company that manufactures and sells electric motors. Human facilitators shield the game engine in this game.

These game experiences show the value of games for learning. The games mix economic and technical aspects with human aspects, e.g. psychological, social, and political behavior. The games engage participants, partially through competition, partially by the set-up with pressure and emotions. Nevertheless, most games are engaging with relatively few resources and limited amount of game rules and knowledge.

Most participants perceived the shown demonstrator as too limited, too mechanistic to fit the goal of accelerating development of systems engineers. Some of the criticism:

- Emotional content of real interaction is missing
- Learners need more visual communication; e.g. what is going on in the company?
- Appears to focus on processes, soft skills are not addressed; is the artificial intelligence technology advanced enough to create realistic interactions that support soft skill development?
- Team interaction is missing

A suggestion is to embed the current demonstrator in human facilitated environment, e.g. workshops before or after the game to complement the learning experience.

The conclusion is that forum participants appreciate the vision of the experience accelerator. The demonstrator is too limited to support the goals of the vision. Crucial capabilities that have to be added are multi-player capabilities, team interaction, and more realistic interaction, e.g. voice and advanced visualization.

At the end of the discussion, we identified the following critical issues that popped-up in all discussions:

- **Realism**: the learning experience has to be close to "reality"
- **Engagement** is what makes games attractive. A game that is not engaging fails its purpose.
- **Complexity** of the game will be limited by development (and maintenance) cost and to limit the threshold for playing. However, "reality" is complex. The complexity of reality is core to the existence of the systems engineer's role. Limiting complexity can be a threat for realism.
- **Transferable Learning:** can the experiences from the game be used in the real world? Will game participants be able to generalize the experiences so that they can be of value in different circumstances?

3. Reflection on theory and practice

The systems engineering master program in Kongsberg has the ambition to reduce the time for graduates to develop into systems engineers. A prerequisite for the study is that students must work as engineers during their study. Engineering experience is a crucial element in the development into systems engineers. The students have a 3-year part-time contract in industry and in the same 3 years follow a study with a nominal load of 120 ECTS; that is a study load of two full-time years.

The program strives to connect theory at school and practice at work in multiple ways:

- Home work projects of courses can relate to practice
- A special course called Reflective Practice actively pursues such connection
- The master project at the end of the study explicitly connects both aspects

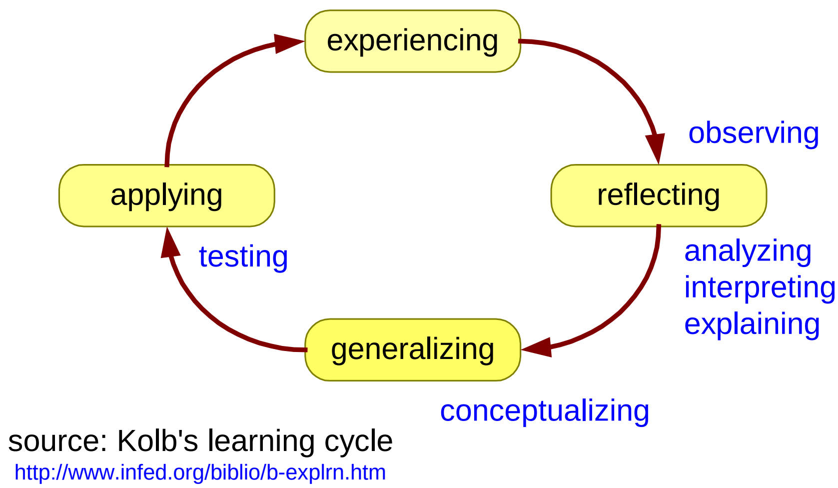

Figure 2. Kolb's learning cycle is a leading principle behind the master in Systems Engineering at Buskerud University College

Kolb's learning cycle, as shown in Figure 2, is the leading principle behind the master program and the course Reflective Practice. The way of working is inspired by Schön's book [Schön 1983].

Objectives of the course Reflective Practice are:

- to help students to develop their reflective capabilities
- to stimulate students to bring their practical experiences into the class room
- to stimulate students to apply what they learn at their company
- to stimulate students to wonder about state-of-practice
- to stimulate students to be critical to:
    - the offered education
    - the way of working in their company
    - their own position and attitude

The course Reflective Practice consists of nine workshops over 3 years: reflection, my role and style, critical thinking, domain knowledge, how to apply SE in daily work, cultural differences, communication, from student to SE, academic writing. In the fourth semester, the students have their international semester. During this semester, they run a project on cultural differences. All material for this course can be found at http://www.gaudisite.nl/BUCmasterSE.html.

The students prepare a workshop by making a pre-assignment. The workshop itself typically consists of three steps:

1. create awareness of the topic by relating to their current experiences
2. provide background for the topic, e.g. some models, methods, or techniques to understand the topic better and to cope with it
3. stimulate to think how to apply new insights in future

We discussed reflection as means to accelerate experience by discussing the following questions:

- What reflection do individuals, teams, or organization currently apply?
- How can reflection be deployed practically?
- How much would reflection help to accelerate experience?

Current practice of reflection is quite varied. Examples of reflection are retrospectives, at anomalies (very good or very bad), PFA, off-site, project evaluations, employee surveys, project/dept. meetings, lessons learned, video's, coffee machine reflections, documenting failures or options not used, post-mortem, continuous organizational restructuring. Several companies indicated that reflection is not performed routinely; it is not formalized in processes.

The forum extracted some critical aspects in the current practice of reflection. When reflection has been formalized, then it often suffers from the fact that the learning cycle is not continued. For example, often lessons learned are only written down without applying them afterwards. Question is what can work in practice. Some companies have databases full of lessons learned; is the content of these databases used afterwards? Who is the owner of the reflections process and follow-up in the form of improvements? One observation was that often the result (short-term) counts, not the improvement (long term). Doskey et al [Doskey 2012] suggest an interesting approach using Positive Deviance to analyze successful systems engineering practices.

Should the organization anchor reflection in processes, e.g. will formalization increase reflection and the value from reflection? Besides the risk that lessons learned get "write-only", formalization may trigger defensive behavior. How can we protect the whistle blower? How can we protect the critical thinker who brings the bad message? Benefits of formalization are that it will happen more structurally and that the organization allows and encourages spending time on reflection. One suggestion was to position reflection at the beginning of a project, e.g. forward looking (how can we benefit from lessons learned) rather than at the end as post-mortem, which probably will result in the write-only database. Another suggestion is to ensure that improvements are in the scope of the control of the reflecting entity; laying solutions outside the own circle of control is an escape pattern.

Participants see reflection as a no-brainer for the acceleration of experience for individuals. It definitely will be helpful early in the career. However, also later in the career it can help to stay innovative and to strive continuously for improvements. They also see reflection for organizations as a means to accelerate experience, e.g. to avoid hitting the same issues repeatedly. However, reflection at organization level is more difficult. Here it is especially relevant to find the sweet spot in the ratio reflection and acting. At a broader scope, organizations should avoid that the same issue is addressed repeatedly. Organizations need to ensure a safe environment for reflection.

4. Project-based learning

A leadership program for early-career engineers, mid-career acquisition professionals, and aspiring technical leaders at Stevens Institute of Technology uses project-based learning as didactic vehicle. All program participants follow three graduate courses to develop "T-shaped" engineers; "T-shaped" engineers combine breadth knowledge with sufficient depth knowledge. The courses offer technical breadth, technical depth, and systems engineering and architecting. Simultaneously, they participate in a design simulation project. This project is performed in three phases: design study, proposal simulation, and implementation. The participants work in teams that compete. Initially, there are about six teams of 5 to 7 members creating proposals. A steering committee with managers from the organization supervises the program. At the end of the proposal phase, this steering committee selects the best two proposals and re-groups the participants in two larger implementation teams. The program finishes with a competition between the two final solutions as implemented by the teams. Figure 3 shows the structure of the leadership program with Technical Development Core (TDC) modules. The elapsed time of the program is about one year.

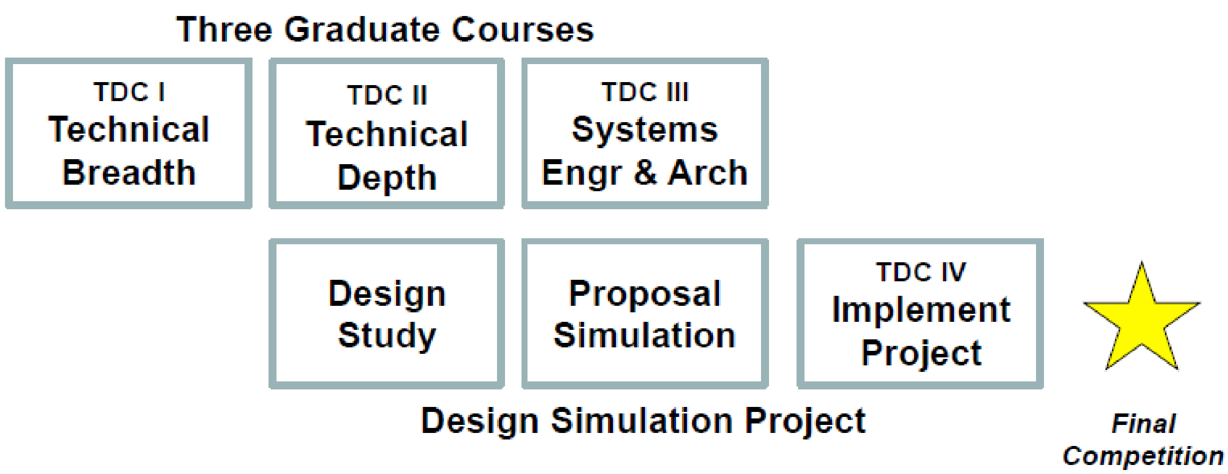

Figure 3. Structure of the leadership program at Stevens Institute of Technology.

The design simulation project concerns an autonomous command and control system. In this particular course, the goal was to specify and implement a demonstrator for an autonomous remote routing system for autonomous vehicles. These vehicles must be able to navigate in an urban environment to explore potential threats. The demonstrator at the end has to show a proof of concept on a highly simplified grid-like urban environment, see Figure 4. The teams get a small but feasible budget to build actually a demonstrator.

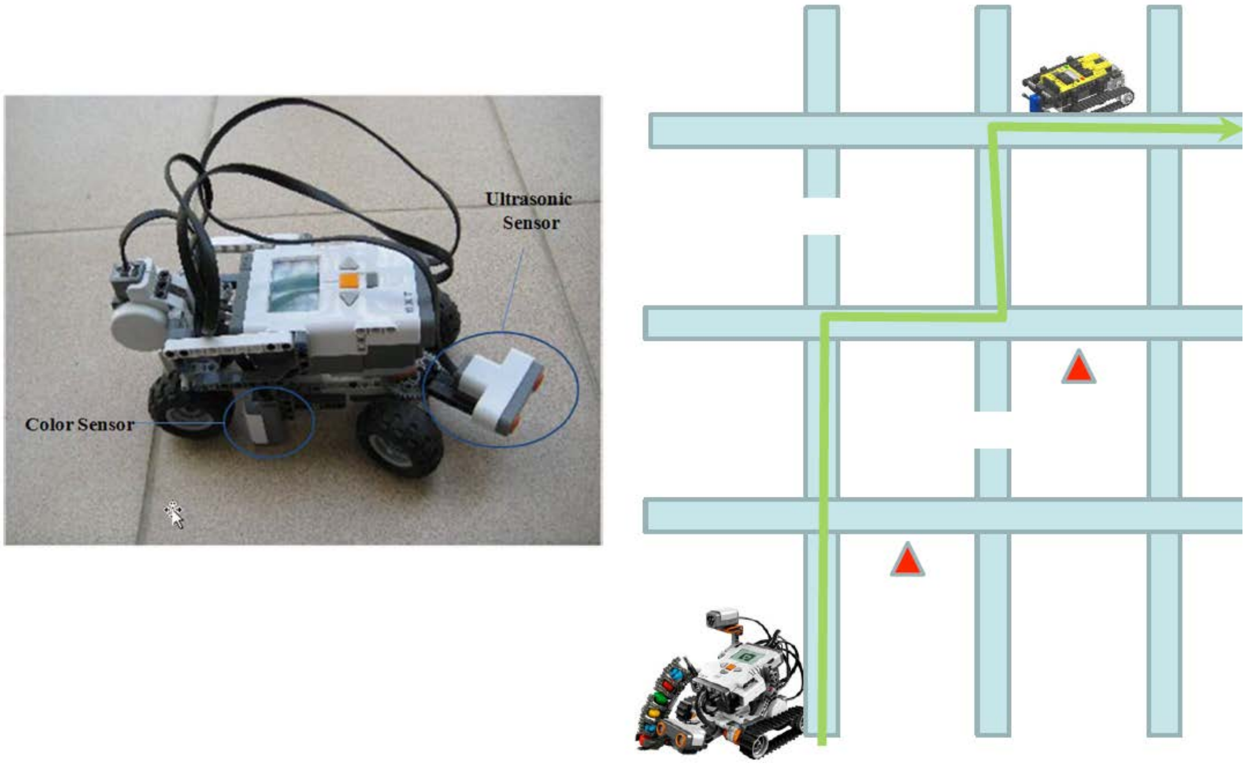

Figure 4. Proof of Concept at the end of the program.

The teams follow the prescribed documentation standards, for example, with four volumes in the request for proposal (executive summary, technical, management, and price). The volumes are elaborated more, for example, the technical volume has 7 sections.

The steering committee had expectations based on Lego competitions of driving robots, where children build and program robots and let them do all kinds of fancy operations. The demonstrators as delivered by the two teams were disappointing from systems engineering perspective. Both demonstrators failed already in the simplest operations, e.g. driving on the roads of a grid. However, from educational point of view, the project was painful but a powerful learning experience. The following feedback supported this conclusion:

- **Successful Learning Experience**
  - *"Despite all the hard work, up and down, as well as sweat and blood, the TDC process has been a **great learning experience**."*
- **Despite all the warnings by mentors and management, it took the firsthand experience for the lessons to really sink in.**
  - ***Integration***
  - ***Virtual Work and Planning***
  - ***80% solution***
- **Positive experience**
  - *I enjoyed this class more than any before it because it is a **realistic learning environment**.*
  - ***I can build on the insight and knowledge gained** through these experiences to successfully lead now and in the future.*
  - *The entire **process taught me more about the proposal and execution processes than I could have hoped to learn in several years of work**.*
  - *TDC allows future engineering leaders to be immersed in a program and **make critical decisions without any kind of risk** to the company.*

More specific are the learning points from the participants, clustered in the areas of systems integration and virtual teaming.

Learning points from integration:

- "Most major issues occur at the interface between two or more subsystems."
- "Encountered major difficulties integrating the navigation software with the motor controller software… I would make sure that test plan implementation occurs as early as possible, especially for a program with multiple layers of integration and test occurring sequentially and/or concurrently."
- "Our project schedule did not leave enough time for integration and test."

- “I would also implement earlier integration and testing of the subcomponents, which always takes orders of magnitude longer than expected or planned.”

Learning points from virtual teaming:

- “The most important aspect I learned from the execution phase was working in a virtual environment.”
- “Final system integration at one site helped the team make huge leaps towards the final product.”
- “It is especially important to efficiently manage resources. This is especially difficult on a program with multiple sites and where virtual engineering is required. “
- “It is sometimes easy to work virtually when dealing with software, but it can be very difficult when hardware is also involved. Even when not developing hardware or software, simply having a face-to-face interaction during a meeting can have a big impact.”
- “We did not communicate our vision well and at times the team presented conflicting viewpoints.”
- “Not all team members share the same motivation, which is crucial to recognize early.”
- “Lack of communication and proactive focus ultimately led the team to a disjointed solution.”

Another pitfall that the teams hit was over-engineering, as shown by the following quotes:

- *“What would make the most difference is a change to the solution itself.* ***Our solution was complicated****… it contributed to our downfall during execution”*
- *“Knowing what I know now, I would stress the importance of designing the* ***simplest solution that meets the customer’s needs*** *in order to reduce the risk of not being able to implement the system within the customer’s schedule.”*

- *"My team's project was not declared the winner because the team's project was* ***over-engineered with high complexity*** *for the available allocated time and what is needed to accomplish the mission."*

Some conclusions of participants:

- *"At the demo, our team project was declared the winner. And I was only partially proud of it because the requirement got de-scoped due to our inability to fully execute. Also, we won because the other team did worse, so they picked the lesser of the two evils. On the other hand, we have learned a lot, and motor-controlling, the part I was responsible for, was fully functioning."*
- *"To be honest, I don't feel like we succeeded. With a few changes early in the execution phase we could have done a lot better."*

The presenter started with a preamble that becomes clear when we discuss the projects results. The preamble is

> *This is not a story about bad people. It is a story about bad results. We believe these results are repeatable and are the product of the context in which people were placed and the processes they employed, rather than the people themselves.*

The steering committee reflected on the results and wondered: How could it be that after hundreds of hours of effort by teams of the best and the brightest neither team accomplished the first task? The committee finds it probable that the course modules and project set-up locked the participants in "nested boxes":

- An inner technical box, characterized by young engineers getting in-depth technical courses and a technically interesting design problem. They played in multiple design teams in a competition.
- An outer programmatic box with a mix of concerns from technical to managerial, e.g. cost, work break down structures, project phasing, schedules, integrated project

teams. Lack of funds, geographic dispersion, and an expansion from 6 to 19 participants per team complicated life further.

The acquisition professionals identified a third box embracing the programmatic box; see Figure 5. This acquisition box imposes a request for information (RFI) and request for proposal (RFP) way of working with fixed evaluation criteria and procedures for protest.

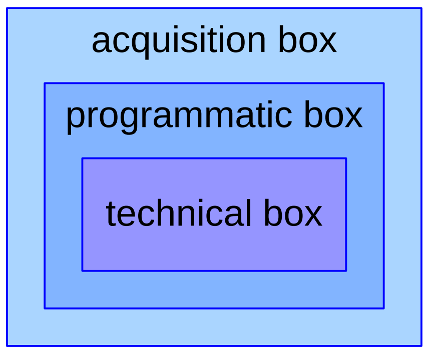

Figure 5. Participants were locked in nested boxes.

5. How to accelerate experience?

The discussions resulted in a list of approaches to accelerate experience:

- Serious gaming
- Stimulate reflection
- Mentoring, coaching
- Case project
- Job/role rotation
- Immersion in big challenge / provide challenges to junior engineers
- Mix, cross fertilize (to build network); internal to company and outside the company
- Harvest from gray-beards
- Peer discussions

- Consolidation / documentation of results of reflection

These approaches are not exclusive. The challenge is to find the right mix. The context or organization will constrain possibilities, e.g. in the way funding works. The number of people involved will have a significant impact on the approach.

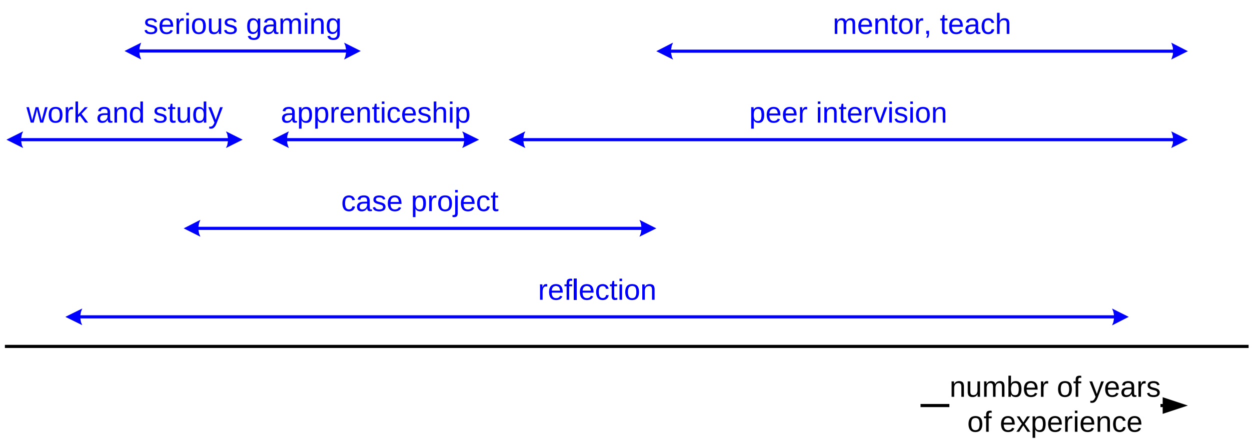

Figure 6. The acceleration approach is probably related to the experience level

During the discussion, we gained the insight that the acceleration approach relates to the experience level of the individual. Figure 6 shows some of the experience approaches on a non-calibrated experience scale. The calibration and the best fitting approach probably depend on the individual. The discussion resulted in another insight: skills like mentoring and teaching need to be part of architecting training. Vice versa, not all employees are good mentors and coaches; hence, the manager should select the mentors with care.

The acceleration cannot be forced. Personal development depends on characteristics of individuals, for example the incubation time. Timing is a critical aspect. In education, the acronym JITT (Just In Time Training) is used to indicate that participants need to have experienced a need to appreciate teaching and the participant should be able to use the new insights and knowledge immediately.

6. Validity of the acceleration concept

The idea behind acceleration is that we facilitate potential systems architects to become fundamentally faster effective as systems architect. Faster means here faster than the current personal development processes. Fundamentally means that we need long lasting skills and a rich varied frame of reference (e.g. principles, insights, patterns, considerations). An additional question is how to make teams or organizations fundamentally more effective.

Some discussion popped-up about the value of experience. Although participants do agree on the need for experience, we have to be aware that experience can backfire too. People can develop a limiting belief. For example, after three failures of a concept, they believe that the concept is infeasible. Experience may constrain creativity or openness of mind. The goal of experience acceleration is to get a more effective workforce.

The participants mentioned some risks as well. For instance, the risk that training in an environment without real consequences may make people more careless in the real world. The training may result in a uniform workforce. A uniform workforce may work better for some problems; however, it may reduce solution capability (and creativity) too.

After the meeting, we discussed what experience acceleration may improve the master program in Kongsberg. A suggestion is that serious gaming or a simulated environment may be a useful format in the second half of the study. A value of such environment is that participants can experience roles outside their current position in the company. This idea needs further elaboration.

7. Conclusions

The maturity level of the subject of acceleration plays a role in the fitness of the acceleration approach. Serious gaming and simulation approaches probably are most suitable for employees early in their careers. Mentoring and peer intervision are applicable for more experienced employees. Reflection is a powerful technique that is beneficial for all experience levels.

Experience is mostly a positive contributor to effectiveness. However, experience may also "box" people, which can limit their effectiveness.

A significant challenge for educators is to identify what experience to capture. Next challenge is to capture this experience efficiently and effectively. The promise of serious gaming and project based learning is that participants can fail without real harm. At this moment we do not have an answer whether we can afford the costs (of capturing and canning experience) to reap the benefits (preventing big failures in the real world). However, the discussion shows that education, simulation, and gaming will never completely replace real experience building.

Acknowledgements

The presence of industry masters (graduate students working concurrently in industry) helped to shape this meeting.

Literature

[NDIA 2010] NDIA SE Division (2010, July). "Top systems engineering issues in department of defense and defense industry(Final 9a-7/15/10)."

[Dubey 2006] Dubey, R. (2006) "Study and Analysis of Best Practices for the Development of Systems Engineers at a Multi-National Organization." Thesis, MIT, Boston, MA, USA.

[Squires 2011a] A.F. Squires, J. Wade, D.A. Bodner, M.Okutsu, D. Ingold, P.G. Dominick, R.R. Reilly, W.R. Watson, D. Gelosh, Investigating an Innovative Approach for Developing Systems Engineering Curriculum: The Systems Engineering Experience Accelerator, 2011, proceedings of ASEE 2011

[Bodner 2012] Bodner, D. A., Wade, J. P., Squires, A. F., Reilly, R. R., Dominick, P. G., Kamberov, G., Watson, W. R. (2012), "Simulation-Based Decision Support for Systems Engineering Experience Acceleration", IEEE International Systems Conference, March 19 - 23, 2012, Vancouver, BC, Canada.

[Squires 2011b] Squires, A., Wade, J., Dominick, P., Gelosh, D. (2011) "Building a Competency Taxonomy to Guide Experience Acceleration of Lead Program Systems Engineers", Proceedings from the Ninth Annual Conference on Systems Engineering Research (CSER), Redondo Beach, CA, April 14-16, 2011.

[Wade 2012] Wade, J. P., Bodner, D. A., Kamberov, G., Squires, A. F. (2012). "The Architecture of the Systems Engineering Experience Accelerator", Proceedings of the 22nd Annual International Symposium, INCOSE 2012, Rome, Italy, July 9-12, 2012.

[Schön 1983] Schön, D.A., The Reflective Practitioner: How Professionals Think In Action, 1983, Basic Books

[Doskey 2012] Doskey, S., MAzzuchi, T., Sarkani, S., Positive Deviance Approach for Identifying Next-Generation System Engineering Best Practices, CSER 2012 in St Louis