29th Annual INCOSE international symposium
Orlando, FL, USA
July 20 - 25, 2019

An A3AOs Method of Software Tools Integration in the Complex System Development

Marius Johanssen
University of Southeast Norway
Tangenveien 8b, 3053 Steinberg, Norway
+47 99399001
marius_j92@hotmail.com

Yang-Yang Zhao
University of Oslo
Postboks 1080 Blindern, Oslo, Norway
+47 31009699
yangyang.zhao@sfe.uio.no

Abstract. Companies commonly face cost pressures due to the increasing industry rivalry and competition. Software tools in this regard has become more and more popular in assisting the complex systems development. However, the lack of a proper system development process that integrates the usage of software tools will hinder the company efficiency. This is especially visible as each engineer may use software tools differently. In the context of complex system developments, this research investigates the optimization of the system development process by integrating software tools. We propose an architecture of a new system development process using A3 Architectural Overviews. Based on a real-life case study and quick-prototyping techniques, the architecture evolves into an optimized system development process that can increase efficiency, quality and consistency for the case company.

Introduction

Company. The research is conducted within the context of a maritime technology company in Norway. The case company has thousands of employees around the world that produces the complex systems, i.e. vessels.

Background. The major customers of this company are shipyards around the world and particularly in Asia. These shipyards have lately experienced a decrease in activity, due to the recession of the oil industry. According to Liang (2017), the workforce in the South Korean shipyards in 2016 and first half of 2017 was reduced by a total of approximately 72.000 workers. This results in a higher cost pressure for their suppliers. To be competitive in the market, the company has a need to reduce costs, while improving quality and consistency.

Challenge. The engineers at a department have an almost ad hoc way of working due to the lack of a proper system development process in the case company. This often leads to increased man-hours, reduced quality and inconsistency of the complex system development projects. On the other hand, the case company invests continuously in software tools that can automate large parts of the software production in the system development. However, these tools are not sufficiently used at the department of interest. The current use of software tools is largely based on the engineer's knowledge, experience, and know-hows from previous projects. Engineers use these tools either insufficiently or wrong in several projects. Because of that, engineers at this department differentiate in their usage of these tools. This is an opportunity to integrate the proper usage of these software tools in the system development process to improve the project performances. Hereby, the software tools refer to the ones that automates software production and configuration in the projects.

Problem Statement. To address the challenge, we focus on answering the following question:

- How to develop a system development process that integrates software tools for optimized performance?

To resolve the problem, this research aims to develop a method to design a system development process while integrating software tools. The method applied in this research could be applicable to similar issues in other industry cases.

In the context of the case company projects, the average use of last-year's man-hours is considered as the baseline for this research. In the baseline projects, approximately 2000 man-hours are used for software engineering. This research aims to reduce that to 1800 man-hours. Therefore, optimizing performance in this research means to reduce 200 man-hours while improving the quality and consistency in the projects.

Literature Review

There exists extant research regarding system development processes such as the Scrum development process (Schwaber, 1997), the Rational Unified Process (Kruchten, 2000) and the system life cycle models (SEBoK, 2017), etc. However, there is lack of a method to architect and communicate a system development process that integrates software tools for a specific context. We aim to contribute to the literature by applying the systems engineering and system architecture approach to outline a method that can remedy such a challenge.

Systems engineering (SE) approach. According to Stevens, et al. (1998), "Systems engineering is about creating effective solutions to problems, and manage the technical complexity of the resulting developments". We adopt the SE problem solving approach in this research from the identification of the problem to the implementation of a solution.

The core of SE is to enable the realization of successful project of system developments that satisfy the needs of its stakeholders (SEBoK, 2017). A stakeholder is defined as "an individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations" (ISO/IEC/IEEE, 2015). The stakeholder analysis is commonly used to identify a project's key stakeholders and to assess their interests and in which way these interests affect the project performances (Leviant, 2014). Leviant (2014) suggests mapping stakeholders in an influence/interest grid and identify the stakeholder's allegiance as an important part of the analysis. In addition to the stakeholder analysis, it is important to properly understand the problem and the origins of the problem in the department. The gap and root cause analysis can be used to serve such a purpose. Harvard school of public health (n.d.) describes that a gap analysis usually consists of listing attributes of the present situation, cross list the attributes of the desired future situation and highlight the gaps that needs to be filled. When identifying the root causes, we can use the principles of the 5 whys' method. In the 5 whys' method, one writes down the problem and ask why the problem happens; if the answer provided does not identify the root cause, one asks why again until the root cause is identified; it may take fewer or more times than five whys (iSixSigma, 2018).

The internal process development is usually associated with the system architecting. According to INCOSE (2018), "Architecture is an integral part of the concepts and practices of Systems Engineering…". In particular, Borches (2010) developed a tool for effective communication of architectural knowledge and named it A3 Architectural Overviews (A3AO). Since the objective of this research is to architect and communicate an optimized system development process that integrates software tools, A3AOs can be considered as a means to achieve that. Hence this literature review will further explore what A3AOs are and previous applications of A3AOs.

A3 Architectural Overviews (A3AOs). The name of A3 indicates the report is captured on an A3 sheet, which is a European standard with the size of 297*420mm. The A3 report was first used in Toyota Motor Corporation. Toyota introduced the A3 reports to mainly record their Plan-Do-Check-Act (PDCA) cycle. Specifically, they used A3 reports to provide status reports on ongoing projects, problem solving and to propose changes to practices and process (Sobek & Smalley, 2008). Moreover, Shook (2009) describes A3 reports as standardized storytelling:

> "One way to describe the A3 is as standardized storytelling, which refers to the ability of A3s to communicate both facts and meaning in a commonly understood format… A3s employ visual methods to share information and thinking. This helps condense key facts into meaningful visual shorthand – storytelling tools that help pack a great deal of data into an elegant presentation".

A3s are proven to be effective communication tools, which is a necessity when communicating the system development process. Borches (2010) further developed the ideas of an effective communication tool, and designed the A3AO. This tool is designed for knowledge sharing and effective communication of architectural knowledge. Borches (2010) made a cookbook on how to make A3AOs. The cookbook suggested using two-sided A3s that one side contains a summary and the other side is with the A3 model. He proposed to use a functional flow to display the systems functionality. It is found that 80% of his practitioners would rather read an A3AO than a document (Borches, 2010). While 73% stated that an A3AO is better than a document when learning a new topic. When using A3AOs, Borches (2010) and Van de Laar & Punter (2010) suggested using multiple levels of abstraction in the A3 model. Muller, Wee & Moberg (2015) also found that creators of A3s tended to realize that they need multiple levels of abstractions.

Applications of A3AOs. The existing research has applied the A3AO in several industries and domains, including reverse architecting of diesel control system (Wiulsrød, Muller & Pennoti, 2012) and lube oil system of a gas turbine (Singh & Muller, 2013). Singh & Muller (2013) combined the A3AOs with Model Based Systems Engineering (MBSE) to develop what they called the Dynamic A3 Architecture. The Dynamic A3 Architecture uses active links that allow the users to navigate through different super- and sub- systems. In addition, Brussel & Bonnema (2015) researched the possibilities of using interactive A3AOs. The Interactive A3AO embraces intuitive functionalities to convey more information. Another related application by Viken & Muller (2018), is conducted within the same department as in this research. Therefore, Viken & Muller (2018) has a similar group of stakeholders as in this research. Their stakeholders pointed the visual nature of A3AOs as more digestible and easier to understand than traditional text documents. Hence, the application of A3AOs could help the visibility and understanding of the system development process. However, the use of A3AOs for process architecting has not previously been studied. This research therefore extends the usability of the A3AO and contribute to the literature on using A3AOs to architect an optimized process.

Conceptual Solution

Based on the literature review, we plot a conceptual solution to explore the possibilities of using A3AOs to architect a system development process that integrates the software tools. The main basis for the conceptual solution is Borches (2010)' cookbook, however certain adaptations to fit the purpose of process architecting and software tools integration are made. The conceptual solution represents the core ideas of the A3AO, such as providing overview, visualization, diagrams and a platform for communication. Besides, the ideas of active links and interactive functionalities, suggested by Sing & Muller (2013) and Brussel & Bonnema (2015) are jointly applied.

Despite Borches (2010) suggested using two-sided A3AOs, Viken & Muller (2018) designed their A3AOs using a one-sided A3AO in the earlier study of the same department to our case. Thus, we adopt the idea of one-sided A3AOs and present the four main views in the conceptual solution. The model information, operational flow and the visual aids views are adapted views from Borches' (2010) cookbook. These views are used to elaborate what software tools to use, how to use them and where to find them by using hyperlinks. In addition, the system development process view which is not used in previous A3AO literature, is uniquely designed to architect the system development process and highlight when the software tools will be applied. By providing those views that communicates the system development process and software tools, the conceptual solution demonstrates that A3AOs are a valid option to meet the research goal. Figure 1 is an example of the conceptual solution with the four views presenting a software tool and the system development process.

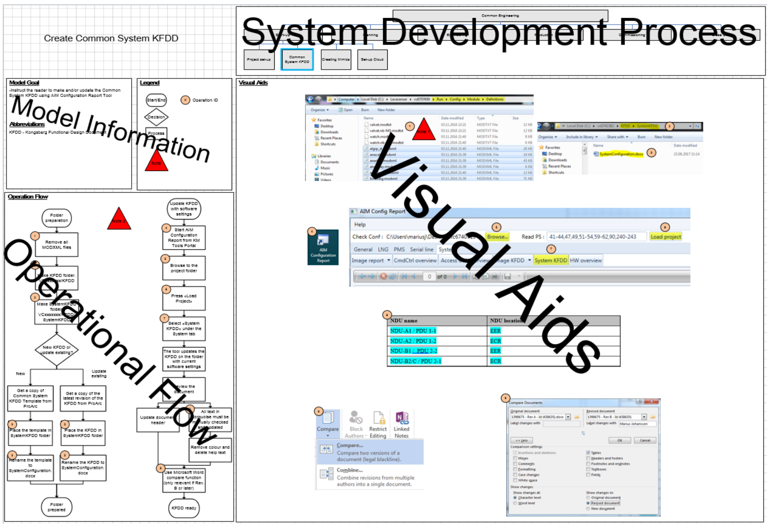

Figure 1. Conceptual Solution – An Early A3AO Example

Data and Methods

We use case study and rapid prototyping as the research methodology in this research. According to Robson (2002), "case study is a strategy for doing research that involves an empirical investigation of a particular contemporary phenomenon within its context using multiple sources of evidence". In the field of software engineering, Runeson, et al. (2012) customized a guideline for case studies, including a theoretical framework, methods for data collection and methods for data analysis. Based on these guidelines, the case-of-interest in this research is the architecture of a system development process that integrates software tools.

The proposed conceptual solution and the findings from the case study will serve as the foundation to create the solution - prototypes. Rapid prototyping is known as an iterative approach where you quickly create a future state of the system and validate it with a broader team of stakeholders (Cerejo, 2010). Creating prototypes rapidly and iteratively by generating feedback early and often in the process will reduce the need for late changes and improves the final design (Cerejo, 2010). In the rapid prototyping phase, operational prototypes of the system development architecture can be released to the stakeholder early in the development process. Thus, this rapid prototyping approach promotes early stakeholder feedbacks, verification and validation. Only after the prototype is verified and validated to meet the goal of optimizing performance, the prototype is released as the final solution. Otherwise the prototypes are modified and updated until goal realization.

Figure 2 is a visual representation of the research process. The activities in the research process is marked with numbers and are illustrated below.

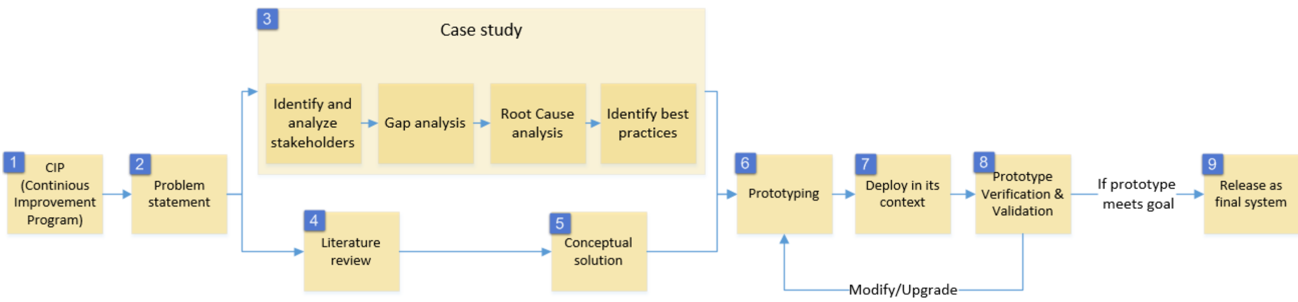

Figure 2. Research Process

1. The concern of inadequate use of software tools in the system development process is raised through a continuous improvement program (CIP) in the case company.
2. Based on the real-life challenge, this step focuses on gaining a proper understanding of the problem and defining the problem statement.
3. The case study creates the necessary case-specific knowledge to enable the development of the first prototype. The data collection methods at this stage is of an exploratory nature.
4. The literature review covers the relevant state of art in SE problem solving, system architecting and A3AOs. The literature review also identifies the specific knowledge areas that this research contributes.
5. Based on the literature review, the conceptual solution is proposed in regards to the use of A3AOs to architect the system development process.
6. Based on case study findings and the conceptual solution, the first prototype can be developed. The prototyping creates an arena for continuous knowledge creation, verification and validation.
7. Once a prototype is operational, it is released to the stakeholders. So that they can gain hands on experience to provide feedbacks and improvement suggestions.
8. For each iteration of the prototype development, it undergoes a verification and validation session.
9. The version of the prototype that is verified and validated to meet the research goal is released as the final solution.

There are three rounds of data collection in the case study. The objectives are to perform a stakeholder analysis, gap analysis, root cause analysis and to identify the best practices for the case-of-interest. The stakeholder analysis provides a means to make qualified selection of participants for data collection sessions and recognize which stakeholders to keep informed during the research. The gap analysis is done to create knowledge of the gaps between “as-is” and “to-be” situations that serves the base for the root cause analysis. The root cause analysis is an analysis of the causes to the problem at a deeper level than the gap analysis. There is a need to investigate what limits or eliminates root causes especially when certain past practices are performance. The identified best practices are implemented in the solution in such a way that root causes are limited or eliminated. Therefore, the data collection sessions are completed in the sequence below.

- Unstructured interview#1 – stakeholder analysis
- Focus group workshop#1 – gap analysis
- Unstructured interview#2 – root cause analysis and identification of best practices

The participants for all data collection sessions are carefully selected for the case of interest in this research. See Table 1 for more details on data collection methods, participants, and objectives in the case study phase.

Table 1. Data Collection Methods, Participants, Experience and Objective in the Case Study

	Unstructured interview#1	Focus group workshop #1	Unstructured interview#2
Participants	1 lead engineer	4 lead engineers, 2 senior engineers, 1 project engineer and 1 department manager	2 lead engineers, 1 senior engineer and 1 project engineer
Experience	10 years’ experience in the maritime industry.	Combined experience of over 60 years in the maritime industry.	Combined experience of over 25 years in the maritime industry.
Role	Key role in a software tool expert group and	Key roles in software engineering at the	Key roles in software engineering at the

	software engineering at the department. The participant's extensive experience and knowledge in the case of interest are sufficient to help identify the stakeholders.	department, software tool expert group and manager. Chosen based on stakeholder analysis.	department and software tool expert group. Chosen based on stakeholder analysis.
Objective	Identify and analyze stakeholders.	Gap analysis.	Root cause analysis and identify the best practices.

In addition to the data collection for the case study, there are data collection sessions for verification and validation of the prototypes. The data collection sessions are also used to collect feedback that can improve the prototype. The rapid prototypes and iterations are continued until a prototype meeting the research goal of reducing 200 man-hours, improving quality and consistency in the company projects. In total, three prototypes were made. These details are presented in Table 2.

Table 2. Data Collection Methods, Participants, Experience and Objectives in the Prototyping Phase

	Prototype 1	Prototype 2	Prototype 3
Data collection methods	Prototype demonstration + Focus group workshop#2	Prototype release + Survey#1	Prototype release + Semi-structured Interview#1
Participants	3 lead engineers, 2 senior engineers, 2 project engineers and department manager	23 participants. These participants include lead, senior, project engineers, project managers and department manager.	2 lead engineers, 1 senior engineer, 2 project engineers, 2 project managers and the department manager
Experience	Combined experience of over 50 years in the maritime industry.	Not applicable	Combined experience of over 70 years in the maritime industry.
Role	Key roles in software engineering at the department, software tool expert group and manager. Chosen based on stakeholder analysis.	Key roles in software engineering at the department, software tool expert group and managers. Chosen based on stakeholder analysis.	Key roles in software engineering at the department, software tool expert group, and managers. Chosen based on stakeholder analysis.
Objective	Explanatory qualitative data collection. Feedback on the A3AO format and functionality. Verification and validation.	Quantitative data collection. Collect suggestions of change (improvements). Gather an overview of the net promotors for this solution. Verification and validation.	Qualitative data collection. Verification and validation.

Case Study Findings

Stakeholder Analysis. Based on unstructured interview#1, the followings are identified:

- The relevant stakeholders
- The stakeholder's allegiance
- Analysis of the stakeholder's interest and influence

Based on the findings from the interview, the influence/interest grid of the relevant stakeholders are plotted, shown in Figure 3. The stakeholders are ranked from 0-5 on influence and interest in the case-of-interest. The ranking system determines where the stakeholders are in the grid of figure 3. In addition, the stakeholders are evaluated based on their perceived allegiance to the case-of-interest, which are color coded. The participants in the upcoming data collection sessions are chosen from the "manage closely" box in Figure 3, hereby known as the relevant few stakeholders.

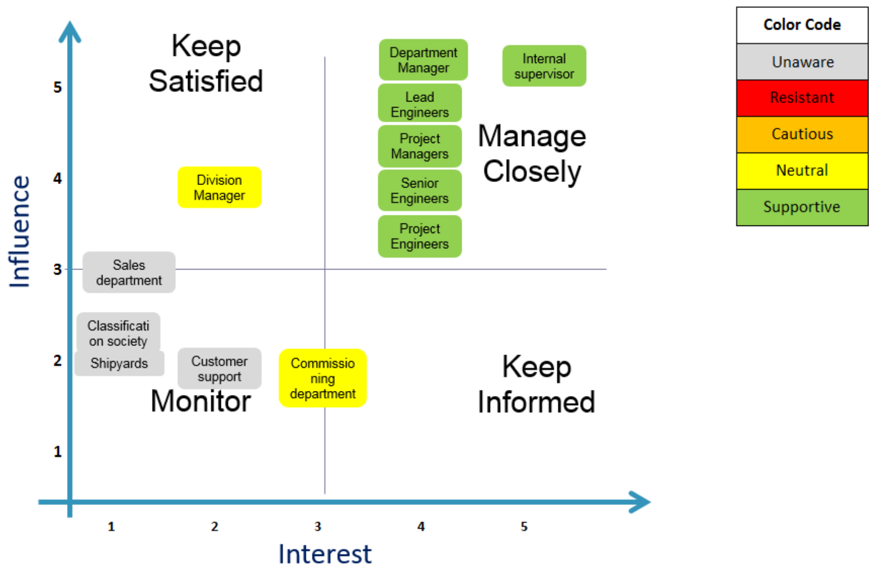

Figure 3. Stakeholders - Influence/Interest Grid for the Case-of-Interest

Gap Analysis. Based on focus group workshop#1, a gap analysis is performed with members of the relevant few stakeholders, which details can be seen in Table 1. The gap analysis includes the "As-Is" and "To-Be" situations, the identified gaps between them and the proposed actions to close these gaps.

Root Cause Analysis. The identification of root causes for the case-of-interest is through unstructured interview#2 with information from the findings of the gap analysis. To identify the root causes, the principles of the five whys method is applied. The root cause analysis is summarized in Table 3, which is organized by categories and their according causes and root causes.

Another topic in unstructured interview#2 is the identification of best practices. They are captured through discussions with the participants. During the process, the root causes are important inputs when seeking the best practices. Two categories of best practices are identified, one category is *software tool best practices*, the second category is *system development best practices*. These best practices will be implemented in the system development process in such a way that root causes are eliminated or limited. The new system development process will integrate and provide instructions on how to implement these best practices.

Table 3. Root Cause Analysis for the Case-of-Interest

Problem	Category	Cause	Root Cause
Software engineers develops software manually instead of tool based	Software tools	Hard to locate software tools	There are several different software tools available
			Located on different servers
		Not user friendly	Different user interfaces
			Poor or no user manuals
	Guidelines and processes	Differences in the use of software tools within the department	No process / guideline on what, when and how to use software tools
		No common understanding of best practices	Best practices are not captured in any process / guideline
	The department	Engineers lacks training in software tools	Lack of available time
		Engineers lacks experience in software tools	Software tools are not used on a regular basis

Prototyping

Based on the conceptual solution, the root causes and the best practices, it is possible to create the prototypes of the solution. The prototypes are A3AOs of the system development process that integrates software tools. Each prototype builds on the former prototype by implementing changes derived from data collection sessions. These changes include A3 design modifications and added functionality, in addition each succeeding prototype integrates more software tools and best practices in the system development process. Hence, the prototyping phase is of an evolutionary nature. Table 4 summarizes the most significant changes obtained through the iterative prototyping approach.

Each prototype has a verification and validation session that evaluates if the prototype can be considered as the solution to this research. The verification and validation sessions are based on expert reviews. In total, three prototypes are made in this research. All three prototypes are validated to facilitate for a reduction of man-hours and improvement of project performance. The first two prototypes do not meet the requirement of reducing 200 man-hours. However, the third prototype reduces at least 200 man-hours and improve quality and consistency in the projects, it is therefore the final solution to this research.

Prototype 1. The first introduction to the concept of architecting the integration of software tools in the system development process is through a demonstration and the succeeding focus group workshop#2. We use active hyperlinks to integrate the software tools with the system development process. The feedbacks from the stakeholders are that the A3AOs are intuitive and the visual nature of the A3AOs makes the topic interesting and easy to understand. The participants agree that the prototype will reduce man-hours and improve quality and consistency in the system development process, however at current state it would not reduce 200 man-hours. Through the focus group workshop#2 the participants came up with suggestions for improvements that can help the prototype reach the goal of reducing 200 man-hours. The suggestions result in four design modifications and three new functions, excluding minor changes (listed in Table 4).

Table 4. Modifications and New Functionality Derived from Prototype Feedbacks

Prototype Number	Data Collection	A3 Design Modifications	New Functions
1	Prototype demonstration + Focus group workshop#2	4	3
2	Prototype release + Survey#1	5	6
3	Prototype release + Semi structured Interview#1	N/A (validated & verified as the final solution)	N/A (validated & verified as the final solution)

Prototype 2. This prototype is at a mature enough level of operation to release it to the stakeholders in the department. By releasing it, the stakeholders can use it in actual projects and acquired hands-on experience with the prototype. Survey#1 is used to collect the knowledge created by the hands-on experience and measure the net promoters for such a solution. It is found Prototype 2 would increase project performance to a higher level than prototype 1, however it is not expected to reduce as much as 200 man-hours.

Survey#1 is sent to 29 stakeholders including project engineers, senior engineers, lead engineers, project managers and the department manager. 23 participants responded to the survey. The survey consists of nine questions. The survey is used to gain confidence in the approach to the solution (using A3AOs) and to evaluate some of the functionality in the A3AOs. In addition, the participants were asked to elaborate on the following:

- *The thing that I like best about the A3 format*
- *If I could change one thing on the A3, I would change*

Those two questions facilitate suggestions of improvements. Table 4 highlights the number of changes from prototype 2. One example is that several stakeholders mentioned the zoom in, zoom out complexity in the A3AO as a possible area of improvement. This is supported by Muller (2017) that A3AOs requires zooming in/out agility. As a response to this issue, new functionality is developed to assist the users in maneuvering and zooming in/out on the A3AOs.

Survey#1 is also used to evaluate the net promotors for the A3AOs. The net promoter score (NPS) is determined by the number of promotors subtracted by the numbers of complainers. The promotors are participants that answer strongly agree and the complainers are the ones that answers neutral, disagree or strongly disagree. Therefore, a positive NPS is considered as a good result. Two examples of statements were evaluated as below:

- Do you believe the A3AO format will increase the likelihood that you use the solution actively? NPS = +6
- Do you believe the combination of diagrams and figures in the A3AO makes the content of the system development process clear? NPS = +11

Prototype 3 (The Final Solution). Through semi-structured interview#1, prototype 3 is verified and validated. Prototype 3 is the system development process that integrates software tools for optimized performance and thus is the final solution in this research.

The participants of the interviews are the two project engineers, one senior engineer, two lead engineers, two project managers and department manager. All participants believe the solution will improve quality and consistency in the projects. The participants, except for the project engineers, also evaluate that the solution either meets or exceeds the requirement of 200 man-hour reduction. Both project engineers' states that they do not have the experience to evaluate how many hours the solution will reduce, but believes it will reduce man-hours significantly. The lead engineers and the senior engineer believe the solution will reduce the projects with approximate 200 man-hours. When verifying if this solution can reduce 200 man-hours, the department manager commented "Yes, without any doubts. Will have effects on 2019 deliveries". He also mentions that the solution looks very promising, which should receive great expectations by the senior engineers. He argues that newly hired engineers as well as experienced engineers will work more efficient and reduce numbers of errors and misunderstandings with this solution. The project managers are more optimistic and believe the solution will reduce man-hours by more than 200 man-hours.

- Project manager 1: "Yes, I believe more than 200 hours in a typical project is spent on looking for the right document/user manual or tool to use for the given task. By having a good overview of where to find this, we will reduce the hours spent significantly, especially for new engineers."
- Project manager 2: "I think the potential is greater than 200, possibly 400 hours within SW engineering and production alone."

With evaluations from the two lead engineers, one senior engineer, two project engineers, two project managers and department manager, prototype 3 passes the expert reviews. Prototype 3 is the final solution to this research; it is further elaborated below.

The final solution is an architecture of a new system development process that integrates software tools for the company. It consists of 33 pages of interactable A3AOs, which is designed by using Microsoft Visio. The A3AOs have two levels of abstraction with a hyperlinked representation of the system development process that allows the operator to maneuver through tasks and phases. The A3AOs adapt interactive functionalities for manouvering and work progress control and Visual Basic programming is suitable for this purpose.

Figure 4 and 5 represent the templates developed for the two levels of abstraction. Through application of the A3AOs in this case, it is found that this setup is suitable for process architecting. The A3AO templates can be a reference point for other companies that explores the posibilites of process architecting. However, it is recommended to make adaptations of the templates to fit your need, the most important aspect is to provide a means to visualise and communicate the process.

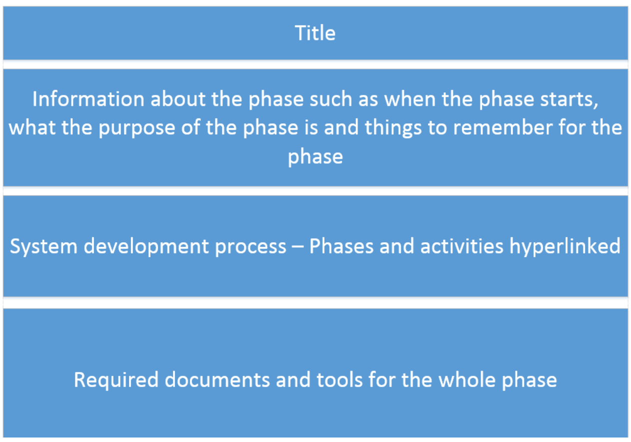

Figure 4. A3AO – Highest Level of Abstraction

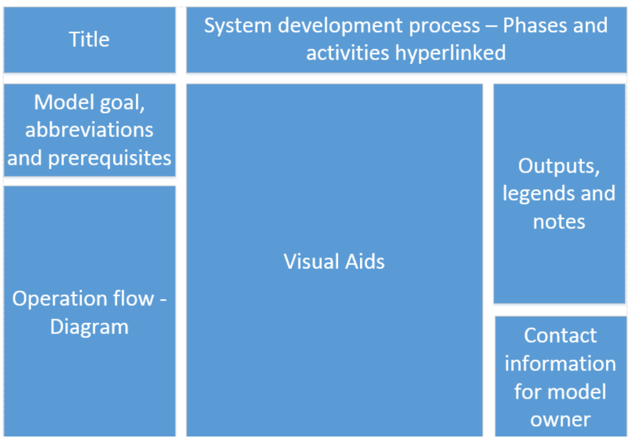

Figure 5. A3AO - Lowest Level of Abstraction

The highest level of abstraction, in Figure 4, is a sumerization page for the phase. It contains the title of the phase, information about the phase, the system development process and required documents and tools for the whole phase. The lowest level of abstraction, in Figure 5, contains detailed instructions on how to complete a task with the software tools or other best practices within the phase.

It contains the title of the task, model goals, operational flow, visual aids, system development process, outputs and other related information.

The lowest level of abstraction contains the instructions for software tools and other best practices. The A3AOs at this level of abstraction has a dropdown menu at the right side were the operator can choose the status of the task. This allows the software responsible or project manager to monitor the progress in the project. Figure 6 shows an example where the operator puts the task to completed and this affects the status of the task and the phase in the system development process.

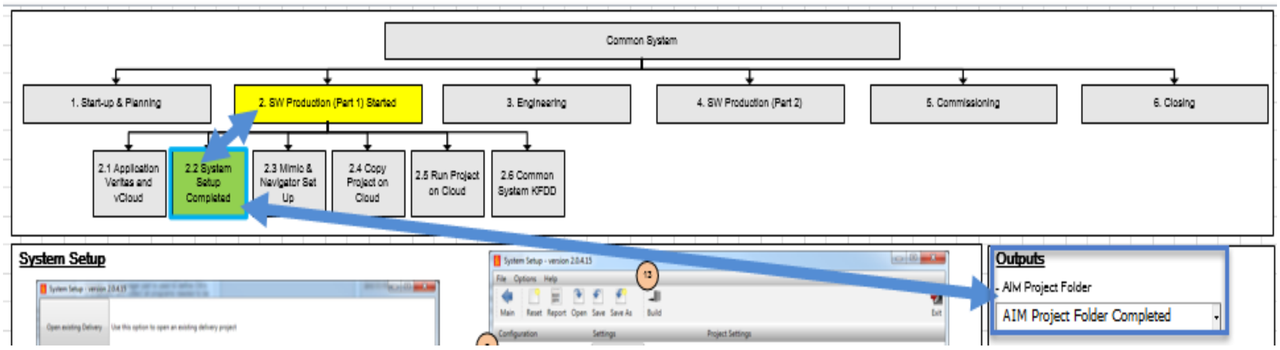

Figure 6. Dropdown Menu Controlling the Status of the Task and Phase

The main views in the lowest level of abstraction are the operator flow and the visual aids. These views give the operator instructions on how to use software tools or perform other best practices. Most steps in the operator flow has an operator ID: x. The operator ID links the operator flow with the visual aids. The operator can double klick on an operator ID to automatically zoom to the relevant visual aid. In addition, the operator can use the push button “Zoom- Operational flow” and “Zoom – Fit to Window” to ease the zooming agility of the A3AOs. The operator can also double klick on a step in the operator flow to change the color to green, which indicates a completed step as with the two first steps in Figure 7.

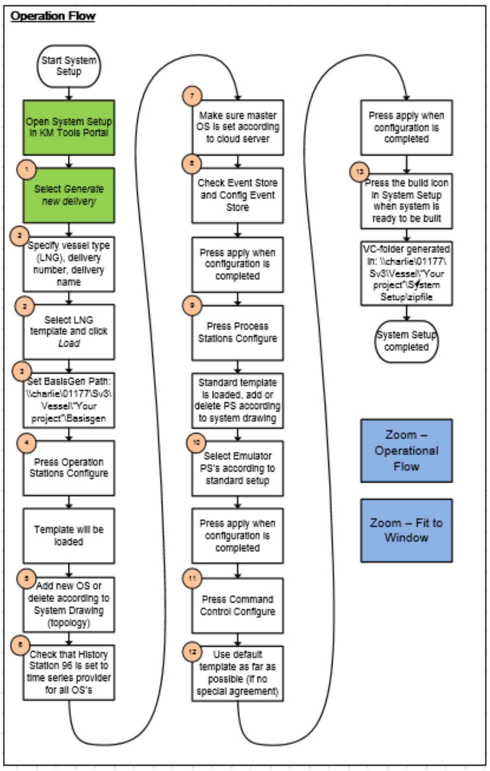

Figure 7. Operator Flow

Figure 8 is an example of a full A3AO page in the lowest level of abstraction. The operator flow is in the bottom left corner, the system development process in the top along with the title, while the visual aids are in the middle of the page.

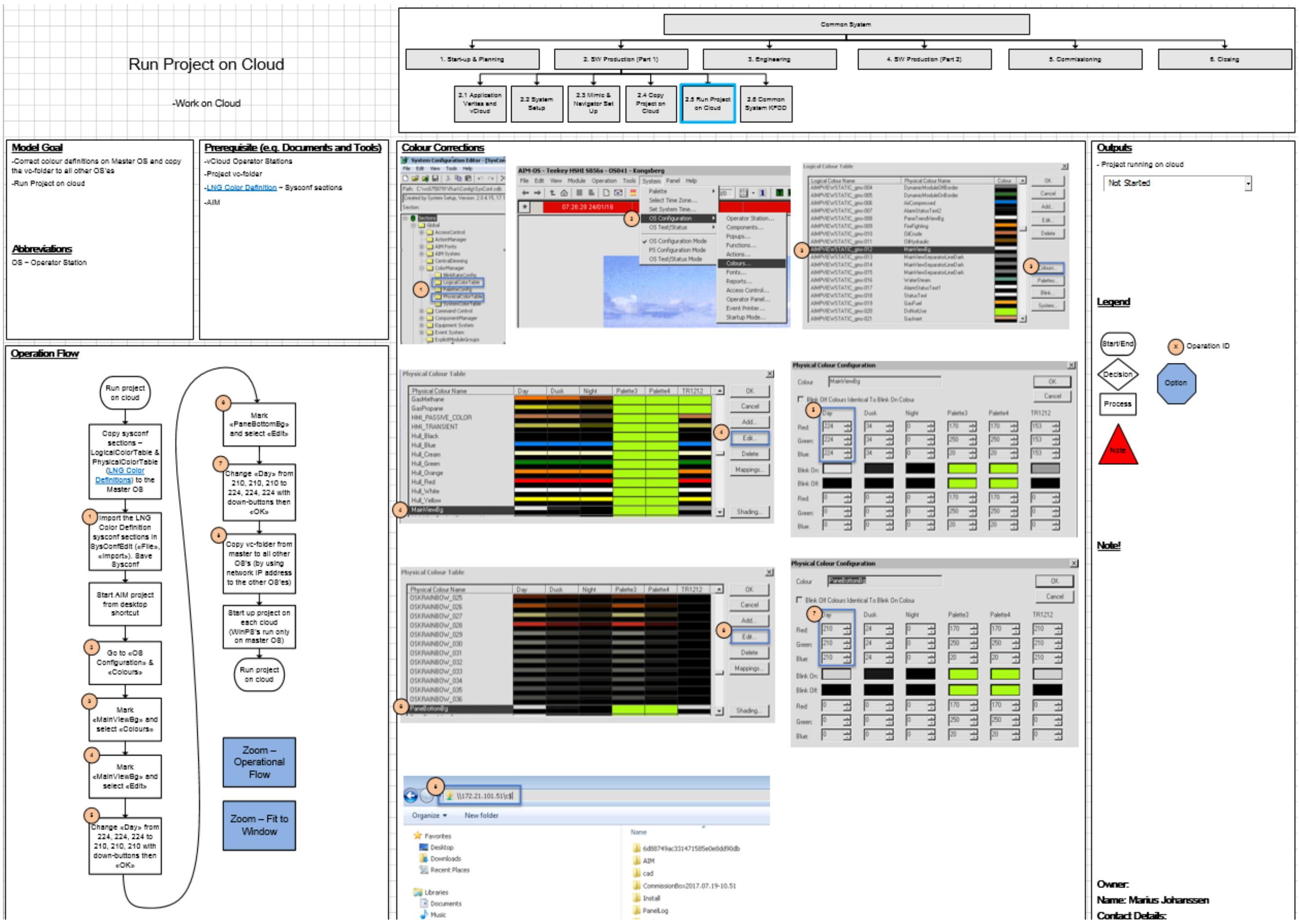

Figure 8. Example of Lowest Level of Abstraction

Reflections

The purpose of this research is to architect a system development process that integrates software tools for optimized project performance. In systems engineering and systems architecting, there are many methods to architect systems and processes. This research creatively uses A3AOs and shows promising results. The A3AOs has adapted Borches' (2010) cookbook to fit the purpose of process architecting. This research shows the possible improvement of processes trough extending the application of A3AOs. A series of research for process architecting with A3AOs in the future is believed to add more academic contributions to the literature. It will also be an interesting topic for future research to compare the A3AO method to other architectural or SE methods.

Prototype 1, 2 and 3 all reduces man-hours, but at different levels. In this research, the iteration of prototyping stops when the prototype reaches the requirements of reducing 200 man-hours and increasing quality and consistency in the projects. However, it is possible to reach further optimized solutions by engaging more iterations. Hypothetical prototype 4, 5, 6, etc. can push the company to new levels of even better performance. Therefore, it is highly recommended to continue to explore the possibility of continuing the iteration of the prototype in the company.

This research uses the measurement of man-hours for cost reduction. An additional evaluation would have been to evaluate the return on investment (ROI), which compares the return on an investment relative to the cost of the investment. This research finds an evaluation of the return (i.e. 200 man-hours each vessel), but the cost (i.e. actual time used to make the A3AOs) is unfortunately not particularly measured. The lesson learned is to measure the time spent to make the A3AOs and ROI would contribute to more qualified estimations for potential future investments in process architecting using A3AOs.

We also would like to acknowledge the limitations of the validity of this. The research evaluates the man-hour reduction by expert reviews and a possible improvement to better the validity of this evaluation is to include measurements of actual man-hour reduction in projects. However, as the department manager said in semi-structured interview#1: “we will see the effects on delivery projects in 2019”, it is not possible to obtain actual measurements from projects during this research. The other limitation to the validity of this research is that one of the authors works at the company. The participants in the data collection sessions may to some extent be influenced by the fact that the interviewer is a colleague. To limit this pitfall, we have tried to triangulate the data collection by having several rounds of data collection as well as feedback and evaluation sessions with multiple stakeholders. Consequently, a solid result clearly indicates that the performed architecture of the system development process is a success.

Conclusion

The focus of this research is: *How to develop a system development process that integrates software tools for optimized performance?* The answer can be many, but this research concludes that optimized performance is achievable through use of A3AOs to architect and communicate the system development process. The architecture is captured with dynamic-interactive A3AOs. Based on the literature review, case study and rapid prototyping, the solutions are optimized to close the gaps, limit and/or eliminate root causes and highlight best practices in this specific case at the company. Given the department in the case company has several complex system projects each year, the solution that reducing 200 man-hours each project contributes to a better chance of maintaining and increasing the company’s competitive advantage.

More importantly, this research contributes to the systems architecture literature and extend the use of A3AOs for process optimization through software tools integration. The method developed can benefit other companies with similar problems. If a company has a need to architect a process or simply unify their way of working, it is recommended to use A3AOs to communicate this architectural knowledge in the complex systems development process. The templates in Figure 4 and 5 can be a good reference point.

References

Bereiter, C & Scardamalia, M 1993, *Surpassing ourselves: An inquiry into the nature and implications of expertise*, Chicago and La Salle, IL: Open Court.

Borches, DP 2010, ‘A3 Architecture Overviews: a tool for effective communication in product evolution’, PhD thesis, University of Twente.

Brussel, FF & Bonnema, GM 2015, ‘Interactive A3 Architecture Overviews. Intuitive Functionalities for Effective Communication’, *Conference on Systems Engineering Research*, pp. 204-213.

Carejo, L 2010, ‘Design Better And Faster With Rapid Prototyping’, viewed 25 May 2018, <https://www.smashingmagazine.com/2010/06/design-better-faster-with-rapid-prototyping/>

Harvard school of public health. (n.d) 2018 ‘Gap Analysis’, viewed 20 May 2018, <https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1678/2014/10/Gap_Analysis.pdf >

INCOSE 2018, ‘Architecture Working Group’, viewed 10 April 2018, <https://www.incose.org/ChaptersGroups/WorkingGroups/process/architecture>

iSixSigma 2018, ‘Determine the Root Cause: 5 Whys’, viewed 13 February 2018, <https://www.isixsigma.com/tools-templates/cause-effect/determine-root-cause-5-whys/ >

ISO/IEC/IEEE 2015, ‘Systems and Software Engineering - System Life Cycle Processes’, Geneva, Switzerland: International Organization for Standardization (ISO)/International

Electrotechnical Commission/Institute of Electrical and Electronics Engineers (IEC), ISO/IEC/IEEE 15288:2015 (E).
Kruchten, P 2000, *The Rational Unified Process - An Introduction*, Addison-Wesley, MA (US).
Leviant, B 2014, ‘Stakeholder Analysis toolkit’, viewed 16 April 2018, <https://www2.mmu.ac.uk/media/mmuacuk/content/documents/bit/Stakeholder-analysis-toolkit-v3.pdf>
Liang, HL 2017, South Korea’s shipbuilding and related businesses cut 35,400 workers in first half. viewed 3 January 2018, <http://www.seatrade-maritime.com/news/asia/south-korea-s-shipbuilding-and-related-businesses-cut-35-400-workers-in-first-half.html>
Muller, G, Wee, D & Moberg, M 2015, Creating an A3 Architecture Overview: A Case Study in SubSea Systems. *INCOSE International Symposium*, 25 (1), 448-462.
Muller, G 2017, ‘Conceptual Modeling: How to Connect Architecture Overview and Design Details?’ viewed 24 April 2018, < http://gaudisite.nl/ConceptualModelingConnectingAOtoDetailsSlides.pdf>
Robson, C. (2002). Real World Research 2nd ed. United Kingdom: Blackwell Publishing
Runeson, P, Höst, M, Rainer, A & Regnell, B 2012, *Case Study Research in Software Engineering*, John Wiley & Sons: New Jersey, NY (US).
Schwaber, K 1997, ‘SCRUM Development Process’, In Sutherland J., Casanave C., Miller J., Patel P., Hollowell G. (Ed.), *Business Object Design and Implementation* (pp. 117-134). Springer, London (UK).
SEBoK 2017, *Guide to the Systems Engineering Body of Knowledge (SEBoK),* The Trustees of the Stevens Institute of Technology Hoboken, NJ (US).
Shook, J 2009, ‘Toyota’s Secret: The A3 Report,’ *MIT Sloan Management Review*, 50 (4), 30.
Singh, V & Muller, G 2013, ‘Knowledge Capture, Cross Boundary Communication and Early Validation with Dynamic A3 Architectures’, *INCOSE International Symposium*, 23(1), 84-97.
Sobek, D & Smalley, A 2008, *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*, Taylor & Francis Group, NY (US).
Sols, A 2014, *Systems Engineering: Theory and Pratice*, Universidad Pontificia Comillas, Madrid (Spain).
Stevens, R, Brook, P, Jackson, K & Arnold, S 1998, *Systems Engineering Coping With Complexity*, Pearson Education Limited (UK).
Viken, A &Muller, G 2018, ‘Creating and Applying A3 Architecture Overviews: A Case Study in Software Development,’ (Unpublished Master Thesis), University of Southeast Norway, Norway.
Wiulsrød, B, Muller, G, & Pennoti, M 2012, ‘Architecting Diesel Engine Control System using A3 Architectue Overview’, *INCOSE International Symposium*, 22(1), 1791-1805.
Yin, RK 2003, *Case Study Research: Design and Methods*, SAGE Publications (US).

Biography

Marius Johanssen has been working in the maritime industry since 2015. While working for the maritime technology company, he gained his master’s degree in Systems Engineering in the University of Southeast Norway in June 2018. Prior to that, he received a bachelor’s degree in electrical engineering in 2015 from Buskerud and Vestfold University College (i.e. University of Southeast Norway since 2018).

Yang-Yang Zhao is currently an Associate Professor at Department of Informatics in University of Oslo. Prior to that, she has been an Associate Professor at Department of Science and Industry Systems in University College of Southeast Norway since 2013. She also worked in the Center for Design Research at the Stanford University and in the Sino-Finnish Center at the Tongji University in 2015. Besides, she has industrial experience in process management and business development and entrepreneurial experience in IT solutions in Asia Pacific. Her research interests include humancentered design, systems engineering, technological catching-up and innovation strategy especially in emerging markets.