33rd Annual INCOSE international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

Coping with Verification in Complex Engineered Product Development

Ola Skreddernes
University of South-Eastern Norway
ola.skreddernes@kongsberg.com

Rune André Haugen
University of South-Eastern Norway
rune.a.haugen@usn.no

Cecilia Haskins
University of South-Eastern Norway
cha@usn.no

Abstract. Coping with system verification in complex systems is a challenging task. This paper reports on how the high complexity of closely integrated systems and long timelines in development affects the verification process in an engineering company in Norway. The literature on verification in development and a survey within the industry-as-laboratory method were used to research the practice of verification in the case company to identify where the company's practice deviates from the recommended practice according to the literature. The research found that the company is taking shortcuts in its documented processes resulting in integration testing late in the development. The authors propose the use of metrics to monitor the progress of the system of interest and the project. In conclusion, continuously tracking and improving the work process is recommended to cope with the complexity of developing the company products.

Introduction

Integration and Verification. The authors' previously experienced challenges inspired this research. We perceived that both the pace and the complexity of the projects were increasing. Kjeldaas et al. (2021) researched the state of practice for finding anomalies in a company's system of interest (SOI). Their findings suggested that the company's acceptance criteria for a given project were insufficient to uncover emergent behaviors. Verification is the process that shows the work has been completed as intended (Larsen and Buede 2002). The verification process should start as early as possible to reduce costs and delays (Callister and Andersson 2016). Any errors found late in the development may lead to budget overruns and delays. Verification is the step where one establishes that the built system is what was ordered. Misunderstandings and ambiguities often emerge in this stage, which can be the result of changes requested through the development phase, or a requirement or function may have been misinterpreted and implemented differently than stated in the requirements specification. This can lead to an error that is first discovered in the verification step. Therefore, it is important to perform verification activities throughout the whole development process. This research investigates the difficulties with integration and verification for complex systems.

Case Company. The case company is one of the major technology companies in Norway and is hereafter referred to as the *Company*. The Company develops high technological systems for both national and international markets within multiple domains. This study has been conducted for the Company as they face the many challenges of continuous improvement to remain competitive. The trends in the market mean that the Company works on multiple projects concurrently while they experience shorter delivery times than previously. Today, developing a new product can take as much as 8-16 years, and product improvements can take 2-3 years. The life cycle of a product can have a duration of many decades as products are upgraded with new technology to increase their capabilities during their useful life. New products are becoming increasingly complex driven by capabilities such as remote communication and autonomy.

The Company has a reliable and stable work force, and the average employment time is 11 years. In anticipation that many of the current employees will retire in the coming years, the Company has begun to recruit new employees to cope with the high demand for engineers. Training new employees at the Company can take up to 6 months before engineers can work without continuous supervision. It takes up to two years before an engineer becomes productive, and it can take as many as five years before the engineer is efficient.

Problem Statement. The Company projects an increase in demand for its products and needs more resources to deliver on its contracts. The current verification process requires much of the Company's resources such that they must hire more engineers and become more efficient in the development and delivery processes to handle this stage. Especially product verification lacks sufficient coverage due to the cost of testing in the real operational environment. Real-world tests are performed last, and systems verification cannot rely on these tests for verifying the system. The Company has developed Hardware-In-the-Loop (HWIL) real-time simulation test arenas to cover the operational environment verification (Bacic 2005, Mobley and Cole 1998). However, these test arenas only allow the company to test in a limited way due to limited accessibility and high cost. The Company recognizes the need for more system-level testing to increase the test coverage and uncover latent errors earlier. Too many mistakes are discovered during the late verification stages in the HWIL test arenas.

Research. This research investigates how the Company can improve its system verification supported by a literature review to find recommendations for verification activities, followed by an in-depth analysis of its current verification practices. The findings from the study report on “best practices” according to the literature and the state of practice in the company. A comparison of these findings suggests practices/changes that the Company can introduce to identify product errors during the verification process. Finally, the authors discuss the results and conclude with recommended actions.

Background

The time expended for defense product development is lengthened by the complexity and the high cost of testing in an actual and realistic environment. The Company uses HWIL test arenas to verify its systems, however, testing in this test arena is time-consuming, costly, and occasionally limited. This creates a need for more robust testing and verification to increase the maturity level of the products before the acceptance test phase. Additional challenges are introduced by ever-changing requirements and specifications from internal and external sources, which must be tracked and communicated throughout the Company's projects.

Observation indicates that the Company has evolved into a silo structure work style that restricts the teams' communication. This limits the information and knowledge flow between the different domains (Melnik and Maurer 2004), which has led to different understandings of how the developing product should function and is exacerbated by insufficient documentation. Domain knowledge is kept inside the silo and is not shared well enough across domains. The engineers in a given domain are so familiar with their own domain knowledge that much of the information is perceived as self-evident (i.e., tacit knowledge). This leads to an attitude among engineers that details that they understand very well do not need to be documented, even though others may not have the same understanding, and the resulting misunderstandings often remain undetected until systems testing and verification.

The company’s verification and acceptance testing personnel experience difficulties completing these activities. Testing often is started later than scheduled, the test capacity is low, and the analysis resources are scattered and often unavailable. The HWIL test arenas are complex and resource-demanding; the integration into the test arenas is time-consuming. Since most of the integration testing is dependent on the HWIL test arenas and the company has only a few of these HWIL test arenas, this reduces the time available for the testers to complete the integration and verification activities. After a test is run, the results must be analyzed. The testers rely on domain experts to manually analyze the test, but these domain experts are not always available. The consequence is that many of the test logs are not checked and analyzed sufficiently and the Company struggles constantly to maintain the verification schedule. All of these restrictions leave little time to improve the verification activities. The Company believes that automating the testing and analysis is one way to cope with this lack of time and resources.

Research Questions. This research investigates how verification is performed in the Company and how current practices compare with what the literature recommends. Ideally the Company wants to identify product errors early in the system verification cycle. This study aims to discover how the Company should work to mature their development processes to deliver robust and reliable products.

The research questions for this study are as follows:

- **RQ1:** Which “best practices” for verification are reported in the literature?
- **RQ2:** What does the literature suggest about changes to the current testing practice in the Company?
- **RQ3:** Which changes can help the Company improve the identification of errors before acceptance testing?

Research Methods

The first research question is answered with a literature review. The second question used a deductive approach to map the state of practice. The third research question investigated the use of internal project metrics. A survey was conducted to gather the engineers’ perceptions. In the end, the research applied these findings to suggest practical improvements and recommendations for identifying product errors before acceptance testing.

The Industry-as-Laboratory. The research was conducted with the Company as the research laboratory (Potts 1993) with the expectations that problems and challenges can be detected better from inside the Company. A combination of theories found in the literature review and observation were used to map the state of practice in the Company. The research findings are essential for acting on the last research question to recommend further actions to aid the verification process in the company. The authors reviewed the Company’s process documentation, reviewed the project Systems Engineering Management Plans (SEMPs), and conducted informal interviews with the engineers to confirm the findings.

Systems Engineering. Systems thinking was used during the research to capture the underlying complexity of the research (Arnold and Wade 2015) and to understand how the Company's work process is applied to the verification work. The research documented the work process using swim lane diagrams (Rumbaugh, Jacobson, and Booch 2005). The organization of the swim lane provides a structure to capture processes and organizational responsibility for each activity. The Company also follows a tailored version of the Vee model. The dual top-down and bottom-up approaches of the Vee model fit the Company's requirement development process (Sols 2014).

Literature Review. The literature review used Google scholar and the university’s online library to locate the relevant literature and gain access to the papers. The relevant articles were found in the Wiley, Elsevier, and IEEE Xplore databases.

The literature review used these keywords singly and in combination to search for relevant literature (Schlosser et al. 2006):

verification, requirement, validation, agile, integration, knowledge, project, measurements, metrics, expensive, systems engineering, industry-as-laboratory.

After initially finding some relevant papers, the authors used the snowballing (Wohlin et al. 2020) and citation search (Papaioannou et al. 2009) methods to find additional relevant articles. For RQ1, the literature search was based on the authors' initial understanding of verification. Then RQ2 motivated additional literature review to fill the newly identified gaps found during the research.

Survey. To uncover the extent that engineers are involved with verification, engineers in a current development project participated in a survey. Answers were collected at two intervals over a 10-week period. The survey aimed to discover how the engineers work with verification and how the Company's processes contribute to the performance of verification tasks. The 28 survey questions were based on the authors' prior experience with verification activities. Questions fell into categories as follows: time, process, communication, workflow verification and validation, project metrics, documentation, milestones, preparations, and lessons learned. Appendix A lists the questions from the survey.

The survey included engineers from the top-level systems design, systems engineers, software engineers, hardware engineers, and integration engineers, with experience levels from 2 to 20+ years. Twenty-two engineers were asked to participate, and fourteen answered the survey. There were eight engineers from design and implementation and six from test and verification. Five engineers had more than ten years of experience, and nine had 2-10 years of experience in the Company. It was challenging to analyze the responses as some of them did not yield any clear patterns. Two top-level engineers also participated in interviews used to validate the survey results.

Likert-Scale. The survey used a Likert-scale response (Joshi et al. 2015). This format consists of a series of statements where the engineer responds along a scale of provided options. The responders were asked to choose between agree, somewhat agree, somewhat disagree, disagree, and N/A. Each statement can also be supplemented with a comment if the engineers choose to clarify their answer. The Likert-scale was used for its ease of use; i.e., it is easy to understand, it does not force an answer, nor does it give a binary answer. It also allows the results to be managed with a small Python program that can perform all the needed analysis and generate graphs of the survey answers. Appendix B shows a sample of the actual analysis for one question from the survey. To analyze the survey, the results were characterized further as belonging to the Vee model's left vs. right side, new vs. experienced engineers, and the survey's total number of responses. The goal is to see the differences between the early and later stages of the project and how well the processes and communication are understood in the company.

Results

This section reports on results of the literature review, survey, interviews, and observations. The confidential nature of some of the results reduces that amount of data that can be shared from the surveys and interviews.

Results From the Literature Review

According to the literature, verification is the process of checking the correctness of an element. The verification can be done with different methods: inspection, analysis, similarity, demonstration, test, and sampling. The ISO 15288:2015 standard recommends implementing a verification process to systematically identify errors and verify a system under development (ISO 2015). According to the ISO standard, properly executed verification activities reduce costs, schedule overruns, and risks. A critical part of verification activities is the preparation stage during which the project's scope, feasibility and limits are identified, and the project's verification strategy is defined. Table 1 summarizes the important factors for performing integration and verification.

Table 1: Literature Review Findings

Topics	**Summary**	**Supporting Literature**
Early verification	Early verification can remove defects early and thus reduce the risk and cost of development.	(INCOSE 2015), (Callister and Andersson 2016), (Tranøy and Muller 2014), (Muller 2007), (Bjarnason et al. 2014), (Gilb and Brodie 2005)
Requirements	The realization of systems requires feasible requirements. Verification becomes unreliable without correct, complete and consistent requirements.	(Bahill and Henderson 2005), (Tranøy and Muller 2014), (Bjarnason et al. 2014), (Cao and Ramesh 2008), (Ferguson and Lami 2006)
Agile approach	Agile embraces continuous change and focuses on delivery of value to the customer while coping with changes. Test-driven development (TDD) is one of the approaches suggested.	(Aggarwal and Singhal 2019), (Dove, Schindel, and Garlington 2018), (Dove and Labarge 2014), (Asan and Bilgen 2013), (Jain, Sharma, and Ahuja 2018)
Configuration management	Complex systems with long life cycles will experience changes. Managing the changes requires adequate management of all configurations of a system during its lifecycle.	(Whyte, Stasis, and Lindkvist 2016), (Efe and Demirors 2019), (Meyer et al. 2013), (Ali and Kidd 2014) , (Ali and Kidd 2013)
Traceability	Traceability is required to trace root cause errors, performance deficiencies, sufficient functionality, and the source of changes.	(Mohan et al. 2008), (Königs et al. 2012), (Escalona, et al. 2022), (Tufail et al. 2017), (Mäder and Egyed 2015)
Testing and Integration	Cooperation between design and integration engineers early in the development process is important to effectively remove errors and achieve successful integration.	(Whittaker 2000), (Balci 1994), (Barmi, Ebrahimi, and Feldt 2011), (Hirshorn, Voss, and Bromley 2017)

Early Verification. The ISO 15288:2015 standard (ISO 2015) recommends that verification begins early to reduce the cost, risk, and schedule deviations. The preparation for integration, verification, and validation can identify errors in the system even before it is built. Errors in the system's design should be removed as early as possible as errors introduced in the design will not necessarily be found during integration and testing (Gilb and Brodie 2005). Using quality control and inspections methods to remove errors are an efficient approach to remove the errors early in the development (Ackerman, Buchwald, and Lewski 1989).

MoE, MoP, and TPM. According to the INCOSE Systems Engineering Handbook, measurements should provide helpful information for a project (INCOSE 2015). Measures are intended to provide objective information throughout the project useful to identify and correct errors. The use of metrics can aid in verifying and assessing the effectiveness of a project (Tetlay and John 2010). The measurements should contribute to sound project decisions while tracking the project's objectives (Griffin and Page 1996).

INCOSE recommends the use of three metrics as follows: measures of effectiveness (MoE), measures of performance (MoP), and technical performance measures (TPM). The MoE shall measure the overall operational criteria for success. The MoP shall derive from the MoE and measure the capabilities needed to achieve the operational objectives. TPM shall be representative measures that show if an element/system satisfies its requirements. The measures should provide correct feedback to project management as they manage resources to achieve success (Griffin and Page 1996).

According to Raman and Jeppu (2019), the use of MoE and MoP can help detect and predict emergent behavior in the system under development. Emergent behavior is a significant issue in complex systems. The Company recognizes the need for effective practices to identify unintended emergent behavior in its engineered complex systems (Kjeldaas, Haugen, and Syverud 2021). MoE and MoP have been proven successful to detect emergent behavior elsewhere (Raman and Jeppu 2019).

Requirements. To conduct a satisfying systems verification, one needs a proper set of requirements. Incomplete or incorrect requirements will affect the entire development and its verification. Checking that the erroneous requirements or/and designs have been correctly implemented does not identify the errors (Gilb and Brodie 2005). Without an effective requirements development procedure, the system cannot be reliably verified or validated (Bahill and Henderson 2005). Changes to the specifications can lead to delays and cost overruns during the development, especially if the change is introduced late. A proper analysis early in the development often can reduce the cost and delays (Tranøy and Muller 2014). At the same time, changes during large and complex development are inevitable.

Configuration Management. For complex systems, changes are inevitable during their life cycle. Managing changes is especially important and more complicated as multiple variants and versions of a given system are created. Keeping control of changes and discovering the effect of changes are the goal of configuration management. A proper configuration management solution is essential for managing the development (Efe and Demirors 2019) and ensuring that the correct system is verified.

Testing and Integration. Utilizing systems integration resources early in the development can improve the system's design, reduce the risks and result in a more robust integration. By starting the integration early, the requirements and concept of operations (CONOPS) can be validated, the appropriate measures of mission and systems performance can be established, and the interfaces can be identified for the necessary functionality (Montgomery 2013). The goal of the systems integrator is to enhance the communications between integrated elements and find errors in the negotiated solutions. Testers cannot expect to see every error in complex systems where the probability spaces can be too large to anticipate and evaluate every outcome. The systems integrator must apply suitable test methods and techniques to achieve the best possible test coverage (Ehmer and Khan 2012). As described by Gilb and Brodie (2005), specification quality control cannot remove all errors but has the potential to reduce them to an acceptable level, thereby reducing the error handling needed later in the development cycle and reducing the development cost.

Developing complex systems requires a considerable number of testing activities, and the need for running many tests repeatedly has led companies to automate these tasks. Automated testing is expected to cover more testing than manual tests can achieve. Still, the literature suggests that planning for what and when testing should be automated is often not given enough focus (Garousi and Mäntylä 2016).

Results From Investigating the Work Process

From investigating the Company’s work process, the authors created a swim lane diagram to capture how the Company conducts product development, as shown in Figure 1. This diagram proved to be very useful to further analyze the different steps and their interdependencies.

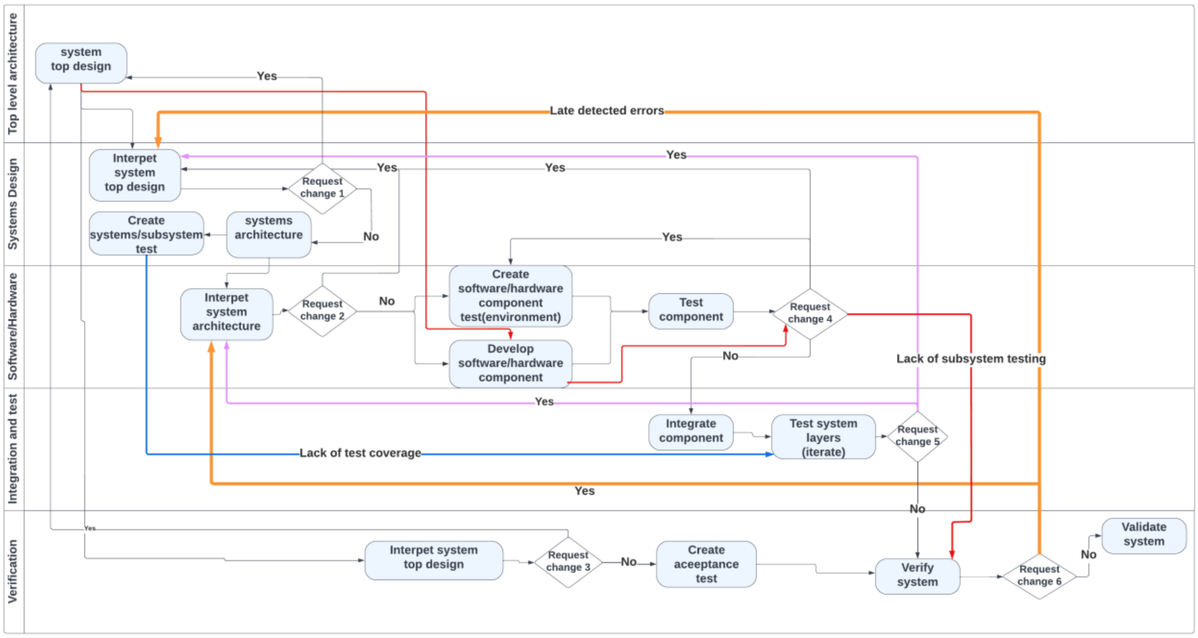

Figure 1: Workflow in the Company

The top-level design collects the customer's needs and establishes the concept and requirements, which contribute to the system acceptance verification plan. The work is delivered to the next domain, e.g., for the system design, a systems architecture is provided for the software/hardware design. Each domain interprets the delivered work to check if the work is possible to implement, or a change request is needed. Once satisfied, the domain engineers can begin their work. The goal of the interpretation is to share the knowledge between all the engineers in the project.

Engineers responsible for integration and test may need to generate change requests based on any errors found. If all the work is done perfectly from the start, there would not be a need for additional change requests. In Figure 1, the unwanted feedback is marked as lilac and orange and they represent late feedback that may lead to significant changes to the system. The lilac feedback comes from the ‘request change 5,’ which is feedback to ‘interpret system architecture.’ These misunderstandings in the design often lead to incorrect implementation, or implementation that is not capable of handling the required complexity. Acceptance testing is the final phase of testing before the system is completed in development and delivered to a customer. Errors should have been found before this stage. The orange feedback comes from ‘request change 6’ and change requests here are highly undesirable and could delay an entire project.

The red arrows represent unwanted shortcuts, e.g., when implementation is done before the system's design is completed or testing is skipped, i.e., the correct order according to the work process is not followed.

The blue line is the situation where integration testing and subsystem verification are skipped. Along this path, few tests are performed for subsystem verification, and the result is that all integration and verification testing is performed at the end in the HWIL or SWIL.

The following undesirable practices were discovered while generating the swim lane diagram.

- A verification plan focused primarily on acceptance testing, and the overall plan for integrated verification testing is not written consistently.
- In design and implementation, shortcuts are taken and steps skipped in the development process.
- The late verification of the system gives late feedback and little time to correct any errors.
- The integration and test process lacks coverage and is sometimes not performed between unit testing of the component before complete systems testing. Systems-level testing lacks automated testing and analysis to support consistent regression testing.
- Formal integration testing is not monitored consistently.

Process Shortcuts. The researchers observed in the company that there exist shortcuts that skip some steps in the development process. Due to time constraints and resource constraints, steps in the development plan are cut to fast-track the process. The plan is to complete the missing activities later in the process, e.g., either a step in the design phase or the verification phase is postponed. The most noticeable effect of missing a step in the process is that any undetected error is caught much later in the development and not found until the last step in the verification phase, viz. acceptance testing. Acceptance testing is done on a full system testing arena (HWIL), to confirm that the top-level requirements are met and satisfy the customer based on analysis that is performed manually by specialist engineers. It is not a testing arena where the system can be robustly tested in a short timeframe, nor is it suitable for regression testing. Testing that requires mass or regression testing should have been done in another test arena much earlier in the process.

The HWIL arenas are the test arenas that provide results with the highest confidence level of all the test arenas used to verify the system. But due to the manual steps involved in set-up, it is also the slowest and most costly test arena. The most worrying discovery is that occasionally software components or subsystems can be sent directly from implementation to the HWIL test arena. This introduces errors that should be caught earlier. The outcome is that acceptance testing resources are inefficiently used to uncover uncomplicated mistakes in the system leaving the acceptance testing engineers with less time to complete the verification they were intended to conduct.

Late Verification Feedback. The second observation is a consequence of the first observation. The late testing leads to late feedback on potentially significant errors. This gives the project less time to fix the errors. Design errors lead to rework and retesting throughout the whole system. This also means that the software and hardware engineers must take part in the rework, resulting in delayed work on concurrently active projects.

The lack of focus on systems integration creates a void between unit testing and acceptance testing as a result of minimum communication between system design and test engineers. The lack of focus on integration on both sides has led the Company to conduct insufficient integration testing. The integration and acceptance testing are done in the HWIL and SWIL test arenas, illustrated in Figure 2. Due to the lack of focus on integration, the company has not maintained the integration test arenas. The company has only the SWIL test arena capable of regression testing, but that is not performed system-wide. This means the company gets little feedback, and the feedback is available late.

Lack of Verification Arenas. The third observation is the lack of verification arenas, which slows the verification process and creates a situation akin to big bang integration. Software components can be integrated at the HWIL level. Hardware is often tested in more specialized arenas, not shown on the Figure 2, without the full software. The first time Hardware and Software are tested together is in one of the HWIL verification arenas. The lack of test arenas where tests can be automatically run makes the verification process slow and resource intensive. Unfortunately, the Company has reduced the number of test arenas, specifically those used to test subsystems. Today the HWIL arena is the

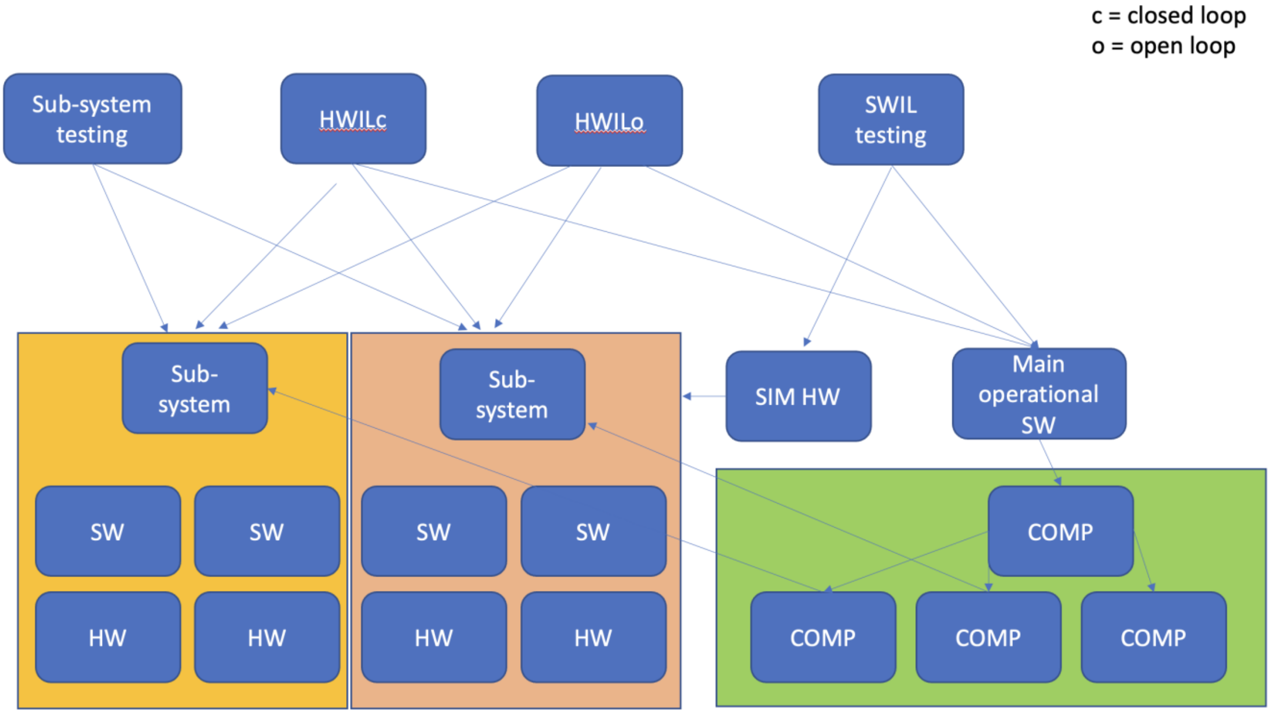

Figure 2: Test Arenas

most convenient test arena to test the system due to the extra work needed to update the SWIL test arena, e.g., the making of components (COMP) that simulate the absent hardware, ref. Figure 2. There are two HWIL test arenas, the HWILo and HWILc. HWILc is a closed-loop test arena where the sensor data is simulated. HWILo is an open-loop test arena where input data are replays of previously recorded data. The HWILo is most used in the early testing, whereas the HWILc is mostly used in the later testing.

Missing Focus on Integration Testing. As discussed, the Company performs little integration testing, with testing performed either in the domains at the lower levels or complete system testing at the top level in the HWIL and SWIL test arenas. With little focus on systems integration, there are few requests for integration testing, generating less demand for integration test arenas, and consequentially lack of early feedback for systems analysis.

Findings From the Survey

Missing Regression Testing at the System Level. Due to the workload of integration and testing, regression testing is skipped too often. Only the most necessary tests are run before delivery and formal regression work was lacking. In addition, the need for expert involvement in the analysis means much of the test data results remain unchecked.

Insufficient Analysis. In addition to the lack of test arenas and regression testing, much of the analysis is done manually, and the necessary domain experts may be unavailable. According to test engineers in the company, this is the most significant bottleneck in the Company’s verification work. Many tests can be run, but too few of the results are analyzed. Most of the data is only analyzed if the test engineers detect a noticeable anomaly and request analysis by the domain expert. Little of the analysis is done automatically, but the Company has started to create automated analyses to speed up the verification process. The effort is not coordinated or communicated well between all domains. Some domains have already introduced tools to help with automating the verification, while other domains are unable to benefit from their experience. Variations also exist between projects where some have already begun to implement new automation solutions while others have not.

The Usefulness of Documentation. From the survey, many of the engineers disagreed with the usefulness of the available documentation. The majority found the documentation helpful, i.e., ten to four. However, experienced engineers and test engineers disagree. Experienced engineers say that documentation does not contain the needed information or is too often outdated, with no apparent indication if the information is up to date. Experienced engineers identified outdated information as a major problem that leads to uncertainty regarding the validity of the information found.

The researchers observed that there is a difference between formal documentation and informal documentation. Formal documentation is stored in a storage area according to the Company documentation procedures, but informal documentation varies between teams. Important information stored informally is difficult to locate by engineers from other teams.

Schedule. The schedule between the domains is difficult to manage. Engineers agree that their input is considered when the initial schedule is created. Many engineers disagree that enough time is allocated for verification and that the allocated time too often is reduced due to earlier delays. The biggest problem with the schedule is getting the right people allocated to the right time. Internally, domain/team allocation does not seem to be a problem. It becomes problematic when people from different domains/groups need to be allocated simultaneously. The test and verification and experienced engineers feel strongly that the verification does not permit enough time to accomplish verification. At the same time, design and implementation and new employees are more divided on the topic.

Measurements and Monitoring. All except two new employees working with design and implementation say that the Company does not use modern error detection techniques at either the integration or verification level. Experts often do the analysis and prioritize requirements verification, which means functional testing and other measurements remain unchecked by the experts. The analysis is often retained by the expert and unavailable to other engineers. Monitoring is done manually by project teams. The use of monitoring is common in technical domains but lacking in the system domain in both design and verification. Without these metrics, keeping track of the system progress and the system's performance and effectiveness is difficult. The company is not utilizing modern measurement techniques system-wide, i.e., MoE, MoP, and systems maturity levels.

Process Improvement. According to the engineers, previous lessons learned are mostly ignored when new projects are started. Lessons learned are only considered by individuals in the company and are only viewed by the top level of the projects. At team levels, it is not considered except for the hardware domain.

Systems testing and verification vary between sub-systems. Safety and security systems and requirements are mandated to be evaluated rigorously. Nothing is left to chance, and everything must be evaluated. In other systems/sub-systems, the testing is left to the system/sub-system owner itself. Here, responsible engineers can use their judgment to create the needed verification plan.

Team Structure. The Company's team is structured around the domains of the company. Each team manages the domain work in their team for all products/parts. The work task starts with the top-level design in breaking down the system into sub-components/deliveries. When the work is done at the top level, the result is sent down to system design and then to software and/or hardware. This work structure does not give the engineers ownership of their work. The work tasks are just received, worked on, and sent to the next stage. The team structure serves to amplify the silo structure in the company.

It is also seen that different domains use different work methods and tools. This has led an investment in multiple tools that provide the same capabilities. Maintenance and training create unnecessary costs and redundant work for the Company.

Configuration Management. With constant iterations, changes happen often. All the components of a given system must be available in the verification arena. Insufficient awareness of change requests can result in the incorrect configuration of the test arena. Keeping control over the correct configuration and keeping track of new configurations is difficult, especially with poor communication between the domains.

Changes are a significant challenge in development, especially for system design, integration, and verification. Here configuration management consists of manual work that must track the configuration across the domains. The different tools used in the domains do not always communicate with each other, which creates extra work for configuration management. The systems design is created in MagicDraw, where each diagram must be manually checked for changes. The tool does not give the users any indication of where the change is made, only that changes have been made.

Discussion

This section discusses the findings from the research as they address the research questions.

RQ1: Which "best practices" for verification are reported in the literature?

Both ISO and the INCOSE recommend a set of processes that should be tailored to fit a company's unique environment. The literature suggests that verification should start early, defining expectations and processes to measure the expectations for the system of interest. Verification should not just be thought of as testing and checking the built system. The literature indicates that early design errors are not necessarily caught in the testing and system verification (Gilb and Brodie 2005). This shows the need for consistent specifications and writing verification tests alongside requirements as a method for finding errors early in the process. As the size and complexity increase, the need for formality increases, and the Company has not enforced this disciplined practice.

Configuration management is an industry standard, but accommodating a complex system is difficult. The different domains in the Company are working with different tools with different version control solutions and this situation leads to poor coordination, which make upgrading legacy systems difficult.

Adaptability has become a focus for modern companies. The agile work methodology embraces changes and allows companies to work more flexibly and adapt to changes. Testing and verification should show evidence of the correct practices through the systematic use of methods and tools. The Company has started to transition to a more agile workflow, but these changes take time, and the lack of monitoring in the project does not provide any indication of how well the agile work method supports a given project. Without this feedback, it becomes difficult to monitor the company's progress.

The Company has tailored a process according to the ISO 15288:2015 standard that defines a set of recommended processes (INCOSE 2015). While the Company follows many of the recommended development processes, systems analysis is the most noticeable deviation; i.e., the Company focuses on the analysis of the requirements and not the SOI itself. The ISO standard focuses more on the overall system throughout the life cycle and the control process and suggests the use of measurement parameters.

The authors did not uncover papers describing the experiences from the industry during the literature review. Most papers only describe a theoretical approach, and as seen in the Company, there are too often multiple reasons for a decision, and a theoretical approach rarely accounts for the enterprise aspects.

RQ2: What does the literature suggest about changes to the current testing practice in the Company?

The Company's formal process does contain all the recommended steps in a verification process. However, as discovered in the research, deficiencies in the execution give the Company challenges during verification. Shortcuts in the development create extra work for the engineers. Removing the shortcuts would solve at least some of the problems.

To solve its issue with late feedback, the Company must facilitate integration and subsystem testing to have test arenas available earlier to support the recommended integration and subsystem verification. While the existing test arenas work well in verifying the products, the Company is now in a situation where much integration and verification are done in the HWIL test arena. The downside is the slow testing and the high need for test and analysis resources during the testing in this arena. The HWIL test arena is not an environment where regression testing can be done due to the high setup cost and test resources needed. The SWIL test arena could be used to cover much of the required regression testing, but it is not being utilized for regression testing for systems verification. Utilizing the SWIL test arena could cover much of the need for regression testing, but that would need the Company to invest in creating test cases and corresponding analysis criteria. The Company would need to develop a new test strategy to best cover the system and an analysis strategy that could verify all the tests.

Communication has been identified to be an issue for the company. Gullhav and Haskins (2020) showed that sharing the information between the system designers and hardware is a challenge, and the silo structure makes communication even more difficult. Many of the same issues have been found when sharing the information between the system designers and test engineers. The workflow does not support communication and this has led to a lack of integration testing, leading to late feedback. The Company should improve its process to encourage the engineers to share more of the information contained in their respective domains.

RQ3: Which changes can help the company improve the identification of errors before acceptance testing?

The Company can improve the execution of the development processes and is aware that the established formal processes are not always followed. Resources and tight schedules are usually the reasons given for not following a process. The lack of feedback from testing and integration has left the projects in a position where it is difficult to uncover errors, making it challenging to verify the system's design. This is exacerbated by the manual analysis of tests and feedback that is not visible to the whole development team. The analysis of tests and the product's performance should be more visible, and regression testing should be done at the integration and system levels.

The Company has started to implement metrics to monitor the project's progress and has recently changed its ‘issue and project’ tracking system, so the metrics are not yet useful for the projects. The use of project progress metrics could help the company focus on the correct work tasks in the project and improve scheduling. The Company can improve its process by adopting the use of measurements such as MoE, MoP, and TPM. Implementing these measurements correctly could be used to share knowledge and information with the whole project, thus increasing communication between the engineering domains. Using measurements would also make all the Company's data more visible to all the engineers and management. Today data is stored away in the Company's servers, and large quantities of data are collected but are not being accessed or analyzed. These datasets contain potentially helpful information that today is not relevant for most engineers due to the domain knowledge required to investigate the raw data manually. Engineers could understand the information by utilizing aggregated measurements such as MoE, MoP, and TPM. A goal for the Company should be to share information about the SOI using the stored data since communication is a significant challenge. The selection of metrics should be created with the intention of improving the communication and engaging the engineers by using measurements.

Further, using metrics can remove the communication barriers between the siloed domains in the Company’s system development process. By creating a shared set of goals, the different domains will all have a collective understanding of the system's purpose. The metrics can help engineers focus on the most important aspects of the system and prevent them from drifting in other directions and not supporting the customer objectives.

There was no evidence that product metrics are used in the integration and verification to measure effectiveness or performance. The Company should investigate how metrics can aid in the verification. Adopting modern and new tools is complicated since currently the Company uses customized tools and solutions in its project development environments. Many of the Company's tools were selected in the late 1990s as many of the mainstream tools popular today were not on the market. New tools and solutions on the market no longer interface with the legacy tools and cannot be adopted without additional expensive customized modifications. More often, investing in new tools would make it easier to use updated features that can aid the company. The current tools also make the training of new employees more complex, where using the modern mainstream tools would reduce the training time.

Utilizing Measurements in the Development. There are different ways measures could be applied to a project. The measurements could be applied to the process or the product. The solution presented in this paper focuses on the product's performance measurements. The Company could start using MoE at the top level of design which should be measured at the systems verification level. This would give an expectation and information about the system that will aid in the development and provide valuable feedback to the top level of the design. From the MoE, the MoP could be extracted and used in the systems architecture and design, which would set the expectations for the design, and the measurement would be the feedback to the designers. TPM would be the measurements for the technical domains. Figure 3 illustrates the areas of concern using a Vee model. Areas in black are the areas where the Company has relatively good control; the red area is where the company lacks coverage; and the orange areas are affected or being affected by the integration process (red area). The use of metrics could aid in bridging the gap between the system design and integration, both to set the expectation and describe the performance of the product and as feedback from the test analysis.

Conclusion

The researchers have identified four practices for verification within the Company that are in conflict with the literature recommendations:

- Insufficient verification activity is scheduled in the verification plan.
- Insufficient testing to cover the whole system, especially integration and regression testing.
- Insufficient analysis of test results and measurements of technical performance.
- Insufficient monitoring or measures for the overall product or project progress.

The lack of requested tests has meant the test engineers focus solely on the top-level verification. The integration and verification are scheduled in the same test arenas (HWIL and SWIL). This has led to late feedback and little time for sub-system testing. The Company needs to enforce its formal process and reinstate integration testing activities.

The lack of testing comes from the lack of requested tests and the lack of available test arenas, especially those suited for regression testing the system. With minimized test coverage, the system is not as well tested as it should be. The test coverage needs to increase to deliver a mature and robust system. First, more tests must be requested, then the tests must be able to be run in available test arenas and then effectively analyzed.

The lack of analysis and measurements can be improved in the company. Utilizing the data collected

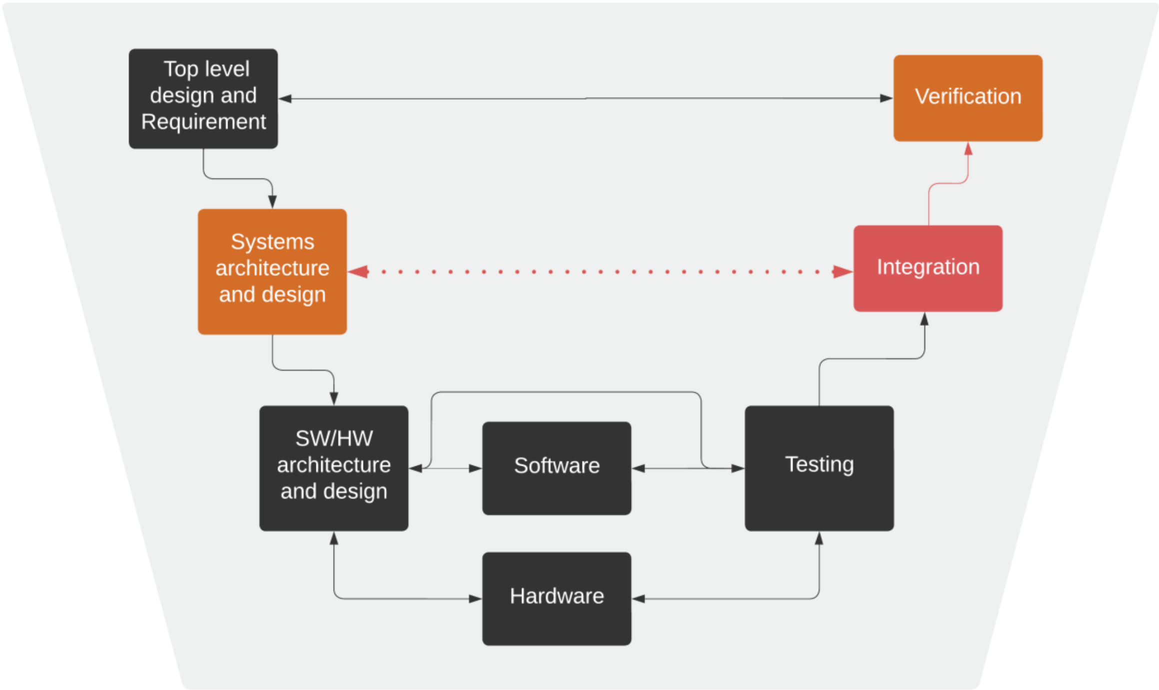

Figure 3: Problem Areas of Verification

and sharing information across the domains should become a priority for the Company. Defining the correct measurements will be a tricky task that would require multiple iterations, as the measurements must cover the objectives of the system of interest. The measures must generate information so the engineers obtain the same insight into the system's objectives. Good measurements would give a project an effective rationale to justify and defend its decisions.

Without any monitoring or measurements, it is difficult to track the progress and objective of the project. The useful information that can be gathered by monitoring and measures should be used to share knowledge, information, and objectives between the domains in the company.

The authors recognize that the survey could have been more process-focused and would have been applicable to most of the engineers which would have given a more extensive dataset and provided additional insights for this research.

Future work

For future work, the Company should focus more on how the system design engineers cooperate with other domains, especially integration. The Company should investigate how they can increase the engagement of the engineers between all the domains to see if that can improve the Company’s effectiveness and efficiency, e.g., how can MBSE be utilized to engage the whole Vee in the development? Regarding analysis, the Company should investigate how the work of writing system tests and analysis can be done consistently and effectively. The analysis should be automated for regression testing on the integration and system levels. With the system's complexity, not every test can be automatically tested and analyzed. There is a need for further research in the Company to find a solution that can satisfy the Company’s total needs.

Future continuing studies could involve creating a process to follow for any industry to measure the maturity level of verification. Further, extended research could share actual measures to help readers understand how the company fares against "best practices".

References

Ackerman, A.F., L.S. Buchwald, and F.H. Lewski. 1989. “Software Inspections: An Effective Verification Process.” *IEEE Software* 6 (3): 31–36. <https://doi.org/10.1109/52.28121>.

Aggarwal, Vikas, and Anjali Singhal. 2019. “Empirical Study of Test Driven Development with Scrum.” In *Advances in Computing and Data Sciences*, edited by Mayank Singh, P.K. Gupta, Vipin Tyagi, Jan Flusser, Tuncer Ören, and Rekha Kashyap, 1046:13–21. Communications in Computer and Information Science. Singapore: Springer Singapore. <https://doi.org/10.1007/978-981-13-9942-8_2>.

Ali, Usman, and Callum Kidd. 2013. “Critical Success Factors for Configuration Management Implementation.” *Industrial Management & Data Systems* 113 (2): 250–64. <https://doi.org/10.1108/02635571311303569>.

———. 2014. “Barriers to Effective Configuration Management Application in a Project Context: An Empirical Investigation.” *International Journal of Project Management* 32 (3): 508–18. <https://doi.org/10.1016/j.ijproman.2013.06.005>.

Arnold, Ross D., and Jon P. Wade. 2015. “A Definition of Systems Thinking: A Systems Approach.” *Procedia Computer Science* 44: 669–78. <https://doi.org/10.1016/j.procs.2015.03.050>.

Asan, Emrah, and Semih Bilgen. 2013. “Agility Problems in Traditional Systems Engineering - A Case Study.” In *Complex Systems Design & Management*, edited by Marc Aiguier, Yves Caseau, Daniel Krob, and Antoine Rauzy, 53–71. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-34404-6_4>.

Bacic, M. 2005. “On Hardware-in-the-Loop Simulation.” In *Proceedings of the 44th IEEE Conference on Decision and Control*, 3194–98. Seville, Spain: IEEE. <https://doi.org/10.1109/CDC.2005.1582653>.

Bahill, Terry A., and Steven J. Henderson. 2005. “Requirements Development, Verification, and Validation Exhibited in Famous Failures.” *Systems Engineering Wiley* 8 (1): 1–14. <https://doi.org/10.1002/sys.20017>.

Balci, Osman. 1994. “Validation, Verification, and Testing Techniques throughout the Life Cycle of a Simulation Study.” In *Proceedings of Winter Simulation Conference*, 215–20. Lake Buena Vista, FL, USA: IEEE. <https://doi.org/10.1109/WSC.1994.717129>.

Barmi, Zeinab Alizadeh, Amir Hossein Ebrahimi, and Robert Feldt. 2011. “Alignment of Requirements Specification and Testing: A Systematic Mapping Study.” In *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, 476–85. Berlin, Germany: IEEE. <https://doi.org/10.1109/ICSTW.2011.58>.

Bjarnason, Elizabeth, Per Runeson, Markus Borg, Michael Unterkalmsteiner, Emelie Engström, Björn Regnell, Giedre Sabaliauskaite, Annabella Loconsole, Tony Gorschek, and Robert Feldt. 2014. “Challenges and Practices in Aligning Requirements with Verification and Validation: A Case Study of Six Companies.” *Empirical Software Engineering* 19 (6): 1809–55. <https://doi.org/10.1007/s10664-013-9263-y>.

Callister, Peter S., and Jonas Andersson. 2016. “Evaluation of System Integration and Qualification Strategies Using the Technical Debt Metaphor; a Case Study in Subsea System Development.” *INCOSE International Symposium* 26 (1): 1016–28. <https://doi.org/10.1002/j.2334-5837.2016.00208.x>.

Cao, Lan, and Balasubramaniam Ramesh. 2008. “Agile Requirements Engineering Practices: An Empirical Study.” *IEEE Software* 25 (1): 60–67. <https://doi.org/10.1109/MS.2008.1>.

Dove, Rick, and Ralph Labarge. 2014. “Fundamentals of Agile Systems Engineering – Part 1 and Part 2.” <https://doi.org/10.13140/2.1.5150.7847>.

Dove, Rick, William Bill Schindel, and Ken Garlington. 2018. “Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group.” *INCOSE International Symposium* 28 (1): 303–20. <https://doi.org/10.1002/j.2334-5837.2018.00483.x>.

Efe, Pinar, and Onur Demirors. 2019. “A Change Management Model and Its Application in Software Development Projects.” *Computer Standards & Interfaces* 66 (October): 103353. <https://doi.org/10.1016/j.csi.2019.04.012>.

Ehmer, Mohd, and Farmeena Khan. 2012. “A Comparative Study of White Box, Black Box and Grey Box Testing Techniques.” *International Journal of Advanced Computer Science and Applications* 3 (6). <https://doi.org/10.14569/IJACSA.2012.030603>.

Escalona, María-José, Nora Koch, and Laura Garcia-Borgoñon. 2022. “Lean Requirements Traceability Automation Enabled by Model-Driven Engineering.” *PeerJ Computer Science* 8 (January): e817. <https://doi.org/10.7717/peerj-cs.817>.

Ferguson, Robert, and Giuseppe Lami. 2006. “An Empirical Study on the Relationship between Defective Requirements and Test Failures.” In *2006 30th Annual IEEE/NASA Software Engineering Workshop*, 7–10. Columbia, MD, USA: IEEE. <https://doi.org/10.1109/SEW.2006.9>.

Garousi, Vahid, and Mika V. Mäntylä. 2016. “When and What to Automate in Software Testing? A Multi-Vocal Literature Review.” *Information and Software Technology* 76 (August): 92–117. <https://doi.org/10.1016/j.infsof.2016.04.015>.

Gilb, Tom, and Lindsey Brodie. 2005. *Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage*. Oxford Burlington, MA: Butterworth-Heinemann.

Griffin, Abbie, and Albert L. Page. 1996. “PDMA Success Measurement Project: Recommended Measures for Product Development Success and Failure.” *Journal of Product Innovation Management* 13 (6): 478–96. <https://doi.org/10.1111/1540-5885.1360478>.

Gullhav, Katrine, and Cecilia Haskins. 2020. “Engaging Mechanical Engineers in System Modeling.” *INCOSE International Symposium* 30 (1): 356–72. <https://doi.org/10.1002/j.2334-5837.2020.00727.x>.

Hirshorn, Steven R., Linda D. Voss, and Linda K. Bromley. 2017. “NASA Systems Engineering Handbook Rev 2.” NASA. <https://ntrs.nasa.gov/api/citations/20170001761/downloads/20170001761.pdf>.

INCOSE. 2015. *Systems Engineering A Guide for System Life Cycle Process and Activities (4th Ed.). D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and, T. M. Shortell (Eds.). San Diego, CA: International Council on Systems Engineering. Published by John Wiley & Sons, Inc.*

ISO, 15288. 2015. “ISO/IEC/IEEE 15288:2015 Systems and Software Engineering — System Life Cycle Processes.” IEEE. <https://doi.org/10.1109/IEEESTD.2015.7106435>.

Jain, Parita, Arun Sharma, and Laxmi Ahuja. 2018. “The Impact of Agile Software Development Process on the Quality of Software Product.” In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 812–15. Noida, India: IEEE. <https://doi.org/10.1109/ICRITO.2018.8748529>.

Joshi, Ankur, Saket Kale, Satish Chandel, and D. Pal. 2015. “Likert Scale: Explored and Explained.” *British Journal of Applied Science & Technology* 7 (4): 396–403. <https://doi.org/10.9734/BJAST/2015/14975>.

Kjeldaas, Kent Aleksander, Rune André Haugen, and Elisabet Syverud. 2021. “Challenges in Detecting Emergent Behavior in System Testing.” *INCOSE International Symposium* 31 (1): 1211–28. <https://doi.org/10.1002/j.2334-5837.2021.00896.x>.

Königs, Simon Frederick, Grischa Beier, Asmus Figge, and Rainer Stark. 2012. “Traceability in Systems Engineering – Review of Industrial Practices, State-of-the-Art Technologies and New Research Solutions.” *Advanced Engineering Informatics* 26 (4): 924–40. <https://doi.org/10.1016/j.aei.2012.08.002>.

Larsen, Robert F., and Dennis M. Buede. 2002. “Theoretical Framework for the Continuous Early Validation (CEaVa) Method.” *Systems Engineering* 5 (3): 223–41. <https://doi.org/10.1002/sys.10022>.

Mäder, Patrick, and Alexander Egyed. 2015. “Do Developers Benefit from Requirements Traceability When Evolving and Maintaining a Software System?” *Empirical Software Engineering* 20 (2): 413–41. <https://doi.org/10.1007/s10664-014-9314-z>.

Melnik, G., and F. Maurer. 2004. “Direct Verbal Communication as a Catalyst of Agile Knowledge Sharing.” In *Agile Development Conference*, 21–31. Salt Lake City, UT, USA: IEEE. <https://doi.org/10.1109/ADEVC.2004.12>.

Meyer, Stefan, Philip Healy, Theo Lynn, and John Morrison. 2013. “Quality Assurance for Open Source Software Configuration Management.” In *2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 454–61. Timisoara, Romania: IEEE. <https://doi.org/10.1109/SYNASC.2013.66>.

Mobley, Scott B, and John S Cole. 1998. “Use of Hardware-in-the-Loop Simulation (HWIL) in the Development, Test, and Evaluation of Multi-Spectral Missile Systems.”

Mohan, Kannan, Peng Xu, Lan Cao, and Balasubramaniam Ramesh. 2008. “Improving Change Management in Software Development: Integrating Traceability and Software Configuration Management.” *Decision Support Systems* 45 (4): 922–36. <https://doi.org/10.1016/j.dss.2008.03.003>.

Montgomery, Paul R. 2013. “Model-Based System Integration (MBSI) – Key Attributes of MBSE from the System Integrator’s Perspective.” *Procedia Computer Science* 16: 313–22. <https://doi.org/10.1016/j.procs.2013.01.033>.

Muller, Gerrit. 2007. “7.1.4 Coping With System Integration Challenges in Large Complex Environments.” *INCOSE International Symposium* 17 (1): 1026–40. <https://doi.org/10.1002/j.2334-5837.2007.tb02930.x>.

Papaioannou, Diana, Anthea Sutton, Christopher Carroll, Andrew Booth, and Ruth Wong. 2009. “Literature Searching for Social Science Systematic Reviews: Consideration of a Range of Search Techniques: Literature Searching for Social Science Systematic Reviews.” *Health*

Information & Libraries Journal 27 (2): 114–22. <https://doi.org/10.1111/j.1471-1842.2009.00863.x>.

Potts, Colin. 1993. “Software-Engineering Research Revisited.” *IEEE Software* 10 (5): 19–28. <https://doi.org/10.1109/52.232392>.

Raman, Ramakrishnan, and Yogananda Jeppu. 2019. “An Approach for Formal Verification of Machine Learning Based Complex Systems.” *INCOSE International Symposium* 29 (1): 544–59. <https://doi.org/10.1002/j.2334-5837.2019.00620.x>.

Rumbaugh, James, Ivar Jacobson, and Grady Booch. 2005. *The Unified Modeling Language Reference Manual*. 2nd ed. The Addison-Wesley Object Technology Series. Boston: Addison-Wesley.

Schlosser, Ralf W., Oliver Wendt, Suresh Bhavnani, and Barbara Nail-Chiwetalu. 2006. “Use of Information-seeking Strategies for Developing Systematic Reviews and Engaging in Evidence-based Practice: The Application of Traditional and Comprehensive Pearl Growing. A Review.” *International Journal of Language & Communication Disorders* 41 (5): 567–82. <https://doi.org/10.1080/13682820600742190>.

Sols, Alberto. 2014. *Systems Engineering: Theory and Practice*. <http://site.ebrary.com/id/11002046>.

Tetlay, Abideen, and Philip John. 2010. “Clarifying the Concepts of System Maturity, System Readiness and Capability Readiness through Case Studies.” *CSER*. <http://hdl.handle.net/1826/4325>.

Tranøy, Eldar, and Gerrit Muller. 2014. “7.1.1 Reduction of Late Design Changes Through Early Phase Need Analysis.” *INCOSE International Symposium* 24 (1): 570–82. <https://doi.org/10.1002/j.2334-5837.2014.tb03168.x>.

Tufail, Hanny, Muhammad Faisal Masood, Babar Zeb, Farooque Azam, and Muhammad Waseem Anwar. 2017. “A Systematic Review of Requirement Traceability Techniques and Tools.” In *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, 450–54. Milan: IEEE. <https://doi.org/10.1109/ICSRS.2017.8272863>.

Whittaker, James A. 2000. “What Is Software Testing? And Why Is It so Hard?” *IEEE Software* 17 (1): 70–79. <https://doi.org/10.1109/52.819971>.

Whyte, Jennifer, Angelos Stasis, and Carmel Lindkvist. 2016. “Managing Change in the Delivery of Complex Projects: Configuration Management, Asset Information and ‘Big Data.’” *International Journal of Project Management* 34 (2): 339–51. <https://doi.org/10.1016/j.ijproman.2015.02.006>.

Wohlin, Claes, Emilia Mendes, Katia Romero Felizardo, and Marcos Kalinowski. 2020. “Guidelines for the Search Strategy to Update Systematic Literature Reviews in Software Engineering.” *Information and Software Technology* 127 (November): 106366. <https://doi.org/10.1016/j.infsof.2020.106366>.

Biography

Ola Skreddernes is a software engineer at Kongsberg Defence and Aerospace (KDA). He holds a certificate of apprenticeship as an electrician, a BSc in automation from the Arctic University of Norway (UIT), and a MSc in systems engineering from the University of South-Eastern Norway (USN).

Rune André Haugen is an industrial-PhD candidate at the University of South-Eastern Norway (USN). He was in service with the Royal Norwegian Air Force (RNoAF) from 1997 to 2003, including graduation from the RNoAF Officer Candidate School in Stavern (1999) and the RNoAF Academy in Trondheim (2001). He holds both a BSc (2006) and a MSc (2013) in Systems Engineering from USN. He has worked as a design engineer at FMC Kongsberg Subsea from 2006 to 2008 (3D modeling), and as a system engineer at Kongsberg Defence and Aerospace since 2008 (system design and system test).

Cecilia Haskins is recently retired and continues in emeritus status with the Norwegian University of Science and Technology (NTNU) and the University of South-Eastern Norway (USN). Her career included over 30 years as a practicing systems engineer and over 20 years educating the next generation of engineers on the importance of systems approaches. She joined INCOSE in 1993 where she held a variety of leadership and other volunteer positions, was recognized as an INCOSE Founder, and continues to be active as a mentor and author. Her educational background includes degrees in chemistry, business, and eventually a PhD from NTNU for application of systems engineering to sustainable development.

Appendix A

Table 2: Survey Questions

1	The project schedule allocates enough time for development work to be accomplished.
2	The project schedule allocates enough time for verification work to be accomplished.
3	The project follows the guidelines from the Systems Engineering Management Plan (SEMP)/Company process.
4	The SEMP/Company process guidelines are helpful to completing my work tasks.
5	Communication between the disciplines supporting a project is effective.
6	The development process/plan addresses interface management (internal and external).
7	The development process/plan is built on inputs from the persons responsible for the work tasks.
8	It is easy for me to report when I find issues in the project schedule.
9	Project managers ensure that resources from other disciplines are available when needed.
10	Requirements/designs/solutions are validated to ensure that they can be implemented and verified before work products are developed.
11	The existing requirement documentation is sufficient for completing my work.
12	Independent reviewers verify individual work products.
13	The verification templates/plan are completed shortly after the requirements are validated.
14	The verification template is useful in the verification/implementation work.
15	The project monitors the amount of change requested for documentation.
16	The project monitors the success (and failure) rate of testing activities.
17	The project uses modern error detection sampling methods to schedule testing.
18	I usually can find the documentation that I need to complete my work tasks.
19	I usually find what I need in the documentation.
20	I use documentation to understand how a system/component works.
21	It is easy for me to report when I find errors in documentation.
22	Project reviews are an opportunity to find hidden errors.
23	The project encourages periodic internal peer reviews of work products (e.g., code, documentation, etc.).
24	Sufficient time is allocated to prepare the verification plan.
25	Sufficient time is allocated to execute the verification plan.

26	Verification requirements are updated as changes are made in the project documentation.
27	We use prior lessons learned when planning new projects.
28	We use prior lessons learned to understand the criteria for successful projects.

Appendix B

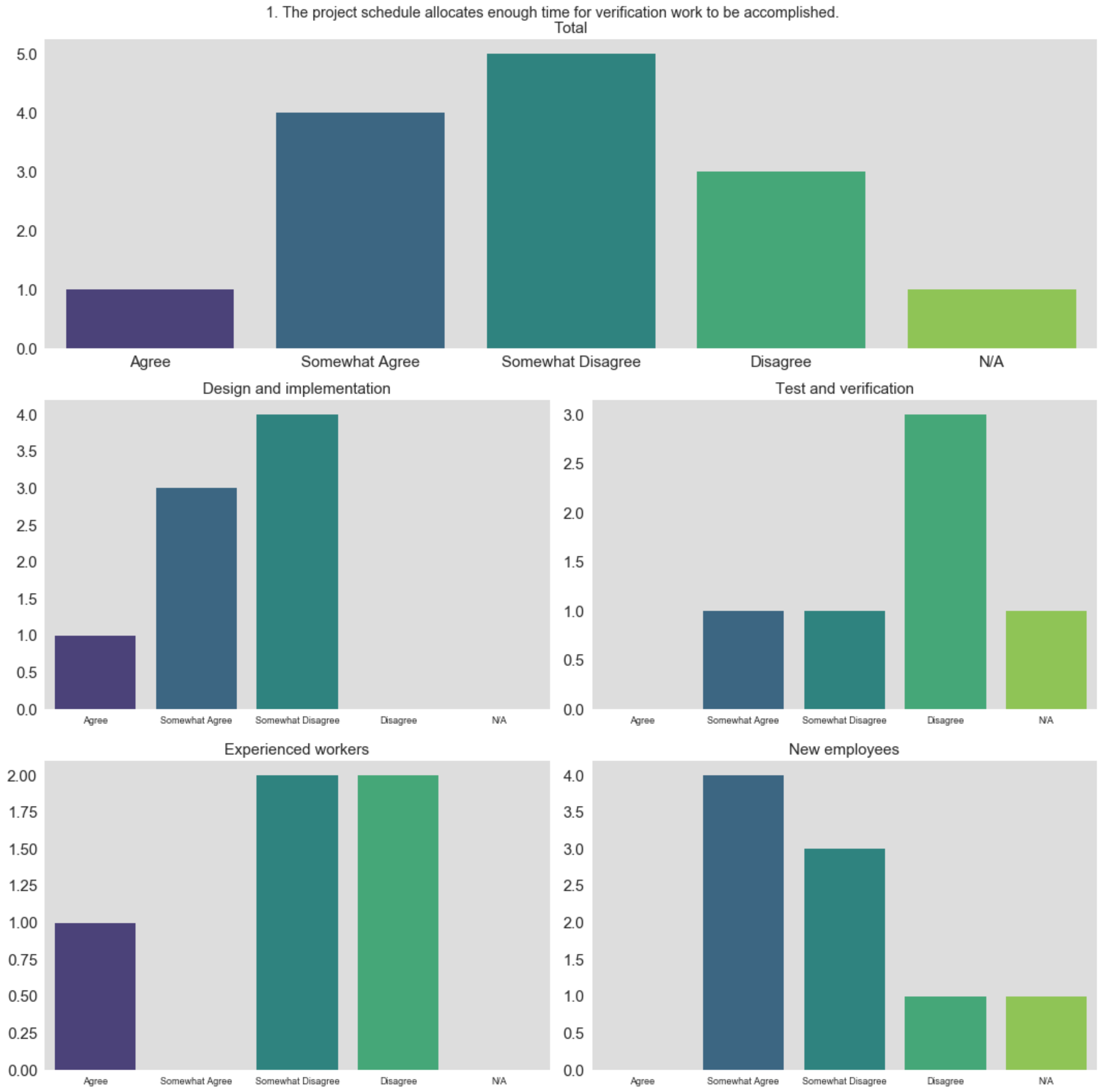

Figure 4: Sample Survey Questions with Graphical Summary of the Responses Along the Likert-Scale and Categorized by Engineer Background and Employment Experience.

Despite the perception that insufficient time is allocated for verification, the actual issue is the availability and allocation of resources, both test arenas and expert engineers to analyze the test results. The time allocated for verification is often enough.