From Waterfall to Agile:
Lessons in Safety Assurance and Cyber Security from the Trenches

Angela Tuffley
RedBay Consulting Pty Ltd
5 The Rampart, Redland Bay, Queensland, Australia
a.tuffley@redbay.com.au

Elizabeth (Betsy) Clark
Software Metrics Incorporated
4345 High Ridge Road, Haymarket, VA 20169, USA
Betsy@software-metrics.com

Abstract

The safety and security assurance process depends on hazard and threat analysis of system requirements. Once identified, safety and security requirements are derived and traced through design, implementation, and verification to provide documentation evidence for safety and security certification.

In theory, traditional product development life cycles such as waterfall or the “V” model provides ample opportunity for careful consideration of system and software safety and security. In proper traditional life cycle fashion, safety hazards and/or security threats are identified in the earliest phases of development. Safety and security requirements are identified, defined and decomposed. The system is then designed to meet those (and other) requirements. Test procedures are written to verify that the requirements have been met.

As an alternative to the traditional approach, complex systems and software development are increasingly adopting an Agile approach. Agile has become more widely used because it addresses several major weaknesses associated with a traditional approach. One common weakness is associated with system requirements; in a system of any complexity, it is unlikely that all requirements can be identified and understood from the outset. Especially for systems with a major component of human-computer interaction, Agile has worked very well to deliver small increments of the system to users to obtain their immediate feedback allowing for course corrections and allowing for an evolutionary understanding of requirements. Agile also avoids a “big bang” approach to system integration and test, the point in the development of complex systems where difficulties are most often encountered.

This paper presents several real-life examples in the assurance of safety and security for complex systems. These examples are taken from a series of project reviews that were performed for the Australian Defence department between 2008 and 2016 using a review framework referred to as the Schedule Compliance Risk Assessment Methodology (SCRAM).

To date, SCRAM has been applied to over 30 major acquisition projects associated with systems of varying safety and security criticality and applying differing development life cycles. After a short introduction to SCRAM, this paper looks at whether there has been any pattern of safety or security concerns related to the life cycle approach used. The paper concludes with a brief discussion of when the two approaches are most applicable based on discussions with real-world developers.

Keywords: Safety Assurance, Security Assurance, Lessons Learned, Life Cycles

1 SCRAM Overview

The Schedule Compliance Risk Assessment Methodology (SCRAM)[1] was developed by the Australian Department of Defence’s Capability Acquisition and Sustainment Group (CASG), which is responsible for purchasing and maintaining military equipment and supplies for Defence. SCRAM Reviews entails a minimally disruptive, independent review of complex projects or programs experiencing schedule slippage in order to identify the root causes, recommend remedial actions and forecast future milestone dates. SCRAM Review teams integrate systems engineering and project management expertise. The reviews are non-advocate with risks and issues identified regardless of their source (i.e., customer, contractor or elsewhere) and a rapid turn-around (two weeks from start to presentation of the results).

To date, more than 30 different projects within Australia, the United States and the United Kingdom have undergone SCRAM reviews with several being reviewed multiple times, at their own request. Application domains include aerospace, maritime, communications, aircrew training, satellite ground control, and command and control.

The methodology draws on best practices from systems and software engineering and from schedule

[1] Additional information about SCRAM can be found at www.SCRAMsite.org.

development and project execution. SCRAM facilitates improved organisational practices based on feedback and from the identification of systemic issues obtained from SCRAM Reviews.

SCRAM can be, and has been, applied at any stage in the project life cycle. Pre-emptive SCRAMs are conducted early in the project life-cycle (e.g. prior to contract award); Assurance SCRAMs are conducted at any point in the project life cycle and Diagnostic SCRAMs when a project is experiencing significant issues.

The remainder of this paper will discuss real-world examples demonstrating safety or cyber-security/information assurance. Observations are made across various development life cycles from waterfall through to Agile.

2 Development Life Cycles

2.1 Traditional

The traditional waterfall approach to system and software development emerged during the 1970s as a framework to provide discipline and rigor at a time when software development was often described as an art rather than an engineering discipline.

The waterfall model, shown in Figure 1, is a sequential process consisting of a series of phases beginning with a concept of operations followed by requirements analysis, design, construction, integration and testing, verification and validation followed by operations and maintenance.

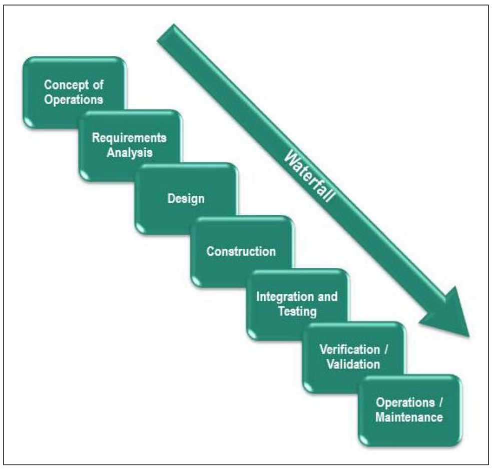

Figure 1: Waterfall Life Cycle

In theory, a waterfall approach provides ample opportunity for careful consideration of system and software safety and security. In proper waterfall fashion, safety hazards and/or security threats should be identified in the earliest phases of development. Safety and security requirements are identified, defined and decomposed. The system is then designed to meet those (and other) requirements. Test procedures are written to verify that the requirements have been met.

The V-model, shown in Figure 2, is an extension of the waterfall model, where the phases shown on the left hand side consist of definition and decomposition activities while the right side consists of integration and test. Each phase on the left has an associated phase on the right that verifies that the resulting product meets its description.

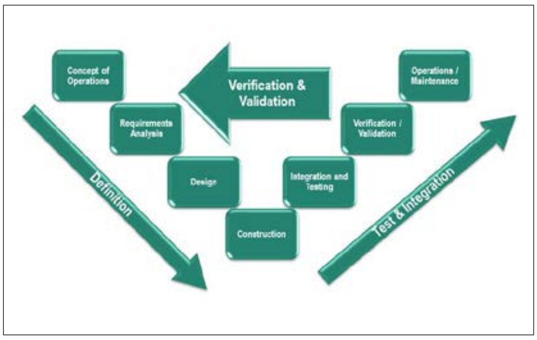

Figure 2: "V" Model Life Cycle

2.2 Agile Development

Agile development emerged in the 1990s as an alternative to what has been viewed as a major shortcoming of the waterfall approach being its rigid sequential and documentation-heavy nature. In an Agile development, requirements and solutions evolve in small increments throughout development. While Agile was first applied to software, it is now being used for hardware and systems development as well.

The Agile Manifesto [1] adopts four key values.

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Typical iterations, known as sprints, of an Agile development result in new functions being demonstrated every two weeks.

Agile development evolved from projects where the specific set of requirements were generally not fully known or understood. In the Defence domain, organisations adopting an Agile approach to development are initially applying the traditional approach of bounding the solution through rigorous requirements analysis, followed by the Agile approach of sprints for the remaining phases. In this paper we describe this approach and life cycle as Modified Agile as distinct from the evolution of requirements in a typical Agile environment.

In theory, one would expect an Agile approach to facilitate rapid responses to changing security threats. One drawback stems from software or system designs that are not architected from the beginning to handle identified hazards or threats.

3 Lessons Learned from the Trenches

The safety and security assurance of complex development projects is, in itself, a complex domain requiring technical expertise. To assure these systems for safety and security certification, it is commonly understood that the analysis of hazards and threats should be done early in the development life cycle, during the requirements analysis phase, then traced through design to implementation and verification. Adopting this process should provide sufficient documentation and assurance for certification bodies to ultimate certify the system. However, based on the authors' observations of projects that have undergone SCRAM reviews, this is not typical practice.

Of the more than 30 projects that have been reviewed, five projects, identified simply as Projects A through E in Table 1, were analysed for specific safety and security assurance observations. These projects represent two legacy systems, Projects A and B and three developmental projects, Projects C, D and E.

Table 1 shows the development approach (Traditional or Modified Agile) adopted by each project and the phase the project was in when it underwent the SCRAM Review. The two projects using a "Modified Agile" life cycle both approached requirements in a waterfall fashion and then adopted an Agile approach in design, implementation and test.

Four of these projects experienced significant schedule delays stemming from safety or security certification issues. Several were delayed by months and one for more than a year in trying to understand and conform to certification requirements.

One thing in common with the four projects that experienced delays, Projects A through D, was that they all addressed assurance concerns late in development when it is difficult if not impossible to try to re-engineer the system for assurance. The specific cause of this late understanding was different for each of the instances we saw. However, Project E was proactively addressing certification early in the life cycle.

Project	Life Cycle	Phase
A	Traditional	Verification
B	Traditional	Operations & Maintenance
C	Traditional	Verification
D	Modified Agile	Verification
E	Modified Agile	Requirements/ Design/ Implementation

Table 1: Projects

In reviewing these projects, five key areas impacting safety and security assurance were identified.

- Legacy systems
- Certification Requirements
- Assurance Processes
- Integrating Commercial-Off-the-Shelf (COTS) Products
- Outsourcing Security Assessment

3.1 Legacy Systems

Project A had implemented an enhancement to a system that was originally developed in the 1980s. The legacy code was mature with a large international customer base. An Australian-specific enhancement resulted in additional safety critical features. The legacy system lacked appropriately detailed documentation to enable adequate tracing of the safety critical enhancement from requirements through design and code to test cases. A significant effort was required to develop these artefacts to enable the system to be certified in Australia.

Project B involved the operations and maintenance of a rapidly aging legacy system with a plan to redesign and replace the system. The project found itself in the situation of not being able to progress past design acceptance as the user had changed the system configuration without appropriate approvals. To address this, the Government Project Office was conducting a Physical Configuration Audit (PCA) to document the current configuration; this would then form the baseline for the legacy design.

3.2 Certification Requirements

Depending on the domain, a number of different agencies provide the system safety and security certifications. Australia has certification requirements that extend beyond those in other countries; this has resulted is systems developed outside of Australia may meet their local certification requirements but not those for Australia.

Project A was delayed by a difference in expectations between what was required by the local certification authorities and what was required for Australian authorities resulting is considerable additional effort.

Project D, developing a system that will be hosted on the defence networks of the US, Australia and other countries, failed to fully understand the requirements for cyber-security certification. Additional software had to be written to support system penetration testing. Several government agencies were involved in the certification and the project had to work with them for some period of time on how to test the security features of the design; the total delay to the project was more than a year.

3.3 Assurance Processes

Two of the projects suffered from the lack of a rigorous safety and security process.

In Project A, much of the legacy system documentation had to be revised in order to produce adequate safety documentation. Whilst the product was mature, the quality of documentation was poor, lacking sufficient evidence of testing and traceability and requiring extensive corrective actions to cover this deficiency. However, to further address safety, the software developers sat with a software safety expert to conduct detailed software safety analyses followed by independent peer reviews from other development teams.

In Project C, there was a "penny drop" moment around the Commonwealth's requirements for a safety case which occurred very late when this project was in the Verification phase. There was no agreed position between the Commonwealth and the contractor on what was required for System Acceptance and that the main issue for Acceptance would be getting through Functional Configuration Audit (FCA) which requires, in part, the safety case and cyber-security assessment. At this point, the project's safety engineer began working closely with the Commonwealth's technical authority whereas prior to this communication between the safety engineer and the technical authority had only been via email.

3.4 Integrating Commercial-Off-the-Shelf (COTS) Products

Irrespective of development life cycle, many projects are integrating substantial Commercial-Off-The-Shelf (COTS) components into their systems. Whilst COTS products can substantially reduce development time, they present a different set of risks and issues particularly in the area of safety and security assurance. Apart from supply chain threats from components manufactured overseas, safety and security certification has to generally rely on evidence supplied by the COTS vendors.

One of the incremental releases for Project D involved an upgrade for two operating systems, one client and one server, requiring re-certification. This re-certification was several years after the one mentioned earlier that led to more than a years delay. Based on the previous experience e, this upgrade was much smoother because they knew what to expect and could plan for it.

Of concern for Project E was that security certification had never been granted across Defence for wireless COTS devices and would be breaking new ground. To mitigate, Project E had established working groups with the signals directorate to understand what would be required. Thus, this was the one project in the set of five that worked certification risks proactively and early in development.

3.5 Outsourcing Security Assessment

Independent assessment of security provides some assurance in this area.

Project E's prime contractor outsourced security penetration testing of the system with the subcontractor conducting a threat assessment and providing the prime contractor's team with training on defensive coding. Following the initial threat assessment and training, the subcontractor would then conduct regular "red team" attacks on the system. This provides another example of this project's proactive approach to cyber security.

4 Conclusion

In this paper, we have discussed five projects, four of which experienced significant delays as a result of safety or security assurance issues. We also discussed one project that was proactive in addressing risks to security certification by engaging with certification authorities early and hiring third-party security experts. Clearly, we have seen challenges with waterfall and with modified agile development approaches.

Five key areas were identified that impact safety and security assurance:

- Legacy systems
- Certification Requirements
- Assurance Processes
- Integrating Commercial-Off-the-Shelf (COTS) Products
- Outsourcing Security Assessment

However, as demonstrated in Project E, the life cycle used is not a key factor for ensuring safety and security assurance but instead an appreciation and identification of these requirements early in the life cycle to drive activities.

5 Addendum

In preparing this paper, we had discussions with Mr Jeff Morris, a retired Vice President from Lockheed Martin who has worked a number of years in Australia (on the Jindalee Over the Horizon Radar Network), in the United States, including being in charge of the development of the Mission Systems software for the F-35 Joint Strike Fighter) as well as a number of other projects. Jeff has spent his career being called into to turnaround troubled projects, several of which had gotten into trouble because they used Agile. He shared the following thoughts with us on the applicability of Agile.

He commented that Agile works best in systems with a high degree of user interaction where the user interface is decoupled from the rest of the application, allowing for changes to made in the user interface without impacting the underlying application. The immediacy and high frequency of product demonstrations for users is a major strength of an Agile approach.

Where it doesn't make sense to use Agile and where he's seen projects get into trouble attempting Agile is in real-time, embedded applications that have to operate within a highly constrained environment with high availability and strict timing requirements. In that case, the requirements have to be defined and traced from top to bottom and the design has to done keeping the system constraints in mind front and centre. He offered the view that once this work has done, through the stage of detailed design, then implementation, integration and test can proceed in an Agile fashion, implementing smaller pieces of the work incrementally.

The domain of cyber security is becoming more complex. Methods to protect the system from security threats and to detect intrusions have to be specified up front and be part of the system design such that when there've been implemented they can be tested and certified. A pure Agile approach, in which the requirements and design are developed piecemeal over time is antithetical to a design developed with security and/or safety front and centre.

6 References

[1] Manifesto for Agile Software Development; Beck, K et al; https://www.agilealliance.org/agile101/the-agile-manifesto/. Accessed 23 April 2017.

Schedule Compliance Risk Assessment Methodology (SCRAM) Reviews; Projects A through E, (2008-2016)