High Maturity, But Yet You're Late?

Emanuel R. Baker, Ph.D.
Software Eng'g Consultants, Inc., Los Angeles, CA, 90077
erbaker@swengcon.com

Elizabeth Clark, Ph.D.
Software Metrics, Inc., Haymarket, VA 20169,
betsy@software-metrics.com

Adrian Pitman
Capability Acquisition and Sustainment Group (CASG), Australian Department of Defence, adrian.pitman@defence.gov.au

Angela Tuffley
RedBay Consulting Pty Ltd,
Redland Bay, QLD, Australia, a.tuffley@redbay.com.au

Abstract

Many organizations have adopted process improvement models (e.g., *Capability Maturity Model Integration (*CMMI)) and have seen marked improvements in quality and performance. Yet, a number of these organizations, including some who have achieved Capability Maturity Level 5, still fail to achieve their product or service delivery milestones. High maturity organizations create process performance models (PPMs) which theoretically are more accurate in predicting realistic delivery dates, but don't always succeed. PPMs rely on knowing the variability of constituent subprocesses and typically include in a Monte Carlo simulation model those that have a major impact on a given process performance goal. PPMs tend to focus more on components that can be characterized by subprocesses having variability. Risk factors don't vary over time and are typically not included. Yet, risk realization often becomes the primary cause of program slippage. The failure to include risk in PPMs is a major contributing factor to a project's lateness.

To combat this shortcoming, the Australian DoD Capability Acquisition Sustainment Group (CASG) has created a review methodology called Schedule Compliance Risk Assessment Methodology (SCRAM). It addresses 11 major categories of risk and enables an assessment review team to evaluate how both customer and development organization have addressed these categories to mitigate their impact on critical milestones. SCRAM establishes the probability of meeting critical milestone dates, and is a proven methodology that has been used very successfully on a large number of projects in a variety of technology and environmental domains (Air, Land, Sea). This paper describes what SCRAM is, how it can be used, and notable successes to date.

Biographies

Emanuel R. Baker, Ph.D., a principal in two consulting firms (Software Engineering Consultants, Inc., and Process Strategies, Inc.) is certified by the CMMI Institute as a high maturity lead appraiser and as an instructor in the CMMI. He has over 45 years of experience in software engineering.

Previously, he was Product Assurance Manager at Logicon and was the author of the original draft of DoD-STD-2168 on software quality assurance.

Elizabeth (Betsy) Clark is President of Software Metrics Inc., a Virginia-based consulting company she founded in 1983. Dr. Clark was a primary contributor to Practical Software Measurement (PSM) and to the SEI's core measures. She is also a principal contributor to Australian Defence's SCRAM development. She collaborated with Drs. Barry Boehm and Chris Abts to develop the COCOTS cost estimation model. She is a long-time consultant to the Institute for Defense Analyses and, more recently, to the Software Engineering Institute. Dr. Clark earned her bachelor's degree from Stanford University and Ph.D. in Cognitive Psychology from the University of California, Berkeley.

Adrian Pitman is a Director in the Capability Acquisition and Sustainment Group (CASG) in the Australian DoD. He has 52 years of Defence experience – 20 years as a member of the Royal Australian Air Force and 32 years in defence capability acquisition. Adrian initiated and is a co-developer of SCRAM. He has participated as an acquisition project team member in Engineering, Quality Assurance and Project Management roles on multiple projects. He is a SCRAM Principal, a ISO 9001 Lead Auditor and a former Defence Materiel Organisation CMMI Lead Assessor.

*Angela Tuffley is the Director and Principal Consultant of RedBay Consulting. She has over 35 years of industry experience, both in Australia and overseas, providing expert professional services in training, assessment and advice for the acquisition, engineering and support of software intensive systems. She is a co-developer of SCRAM and provides consultation on SCRAM, the adoption of the CMMI and ISO/IEC 15504 Information Technology Process Assessment (*Software Process Improvement and Capability Determination (*SPICE)).*

1 Introduction

Many organizations have adopted process improvement models (CMMI, for example) and have seen marked improvements in quality and performance. High maturity organizations, rated at maturity level (ML) 4 or 5, have successfully implemented PPMs that predict the ability to achieve quality and performance goals. Some models have been constructed to predict the ability to achieve critical milestones, such as delivery date. In spite of that, a number of these organizations still miss the mark when it comes to delivery of the product or service on the original promised date. During development, delivery often gets renegotiated with the result that the contractor delivers on the final renegotiated date and claims victory. Yet delivery didn't really occur when the customer originally wanted it.

High maturity organizations create PPMs which theoretically are more accurate in predicting realistic delivery dates. The problem is the basic structure of a PPM. They rely on knowing the process variability of constituent subprocesses and including in the model those that have a major impact on a given process performance goal. This data and the current performance of a constituent subprocess are typically inputs to a PPM to assess the probability of achieving that goal on the project of interest. Consequently, PPMs tend to focus more on issues of quality since aspects of quality can often be characterized by variable subprocesses. Risk factors are not easy to include since they don't vary over time and are typically characterized as point values of impact and probability of occurrence. Yet, risk realized, technical debt, and poor management decisions often become the primary causes of program slip. The inability to include these factors in PPMs is a major factor in a project's failure to meet the predicted delivery date.

To combat these shortcomings, CASG of the Australian Department of Defence created an

assessment methodology called SCRAM (Schedule Compliance Risk Assessment Methodology). SCRAM addresses 11 major categories of programmatic problems. It enables a review team to evaluate how the development organization of interest addressed these factors to mitigate their impact on meeting critical milestones. For any critical milestone, SCRAM quantifies the potential risk (slippage) and determines the numerical probability of meeting that date. It is a proven methodology that has been used very successfully on a large number of projects in a variety of capability domains. This paper describes what SCRAM is, how it can be used to address these issues, and notable successes to date.

2 Statement of the Problem and Its Solution

The media abounds with stories about well-known programs that were late, over budget, and in some cases had quality problems as well. Some notable recent examples include the California High Speed Rail (HSR) project between San Francisco and Los Angeles, the Joint Strike Fighter (F-35), the rollout of the Affordable Care Act web-based user interface, and the Los Angeles Department of Water and Power billing program. Some of these have involved organizations that have been rated at ML 5. Nonetheless, these organizations were not able to create a schedule that guaranteed delivery of the product on the date that the customer originally required. Phase 1 of the HSR project is already several years late, for example.

As an organization matures, its ability to control development processes improves, thus improving product and service quality as well as creating more accurate schedules. An organization at ML 3 has significantly-improved capability to accomplish this over an organization at ML 2. The hallmark of a high maturity organization is its use of quantitative methods to quantitatively control project processes; use of this data to make predictions concerning the achievement of quality and process performance goals; and implement corrective action on the offending process. They create control charts to quantitatively control critical subprocesses and use the current data on the performance of these controlled subprocesses in PPMs to make predictions about, for example, residual defects in the delivered product and to predict if they will meet their schedule constraints. Yet, as noted above, even high maturity organizations miss the mark.

The problem is that most PPMs do not address risk directly. Anecdotally, several lead appraisers have indicated that none of them had ever seen a PPM that included risk or other issues that create schedule risk, for example, technical debt. As noted previously, risk is typically characterized by citing a fixed probability of occurrence and numerical estimate of impact (for example, on a scale from 0 to 1). Since most PPMs are built on parameters having variability over time or frequency of occurrence, risk does incorporate well into such models because it has fixed values. Yet, there are PPMs to predict the ability to meet schedule, but because risk is rarely considered, they often fail to make accurate predictions.

SCRAM, because of its explicit consideration of risk and factors related to risk, does a significantly better job of calculating the probability of meeting critical milestones. Moreover, the process of conducting a SCRAM reveals what was not considered in creating the schedule, an example of which is technical debt. Technical debt is the price a development team pays for omitting or reducing a part of a process, often for the sake of short-term expediency. When an organization, in the interest of shortening development time, reduces the number of peer reviews for example, there is a risk that some defects will go undetected. The price and interest that is paid is the amount of rework unaccounted for in the schedule that is often revealed later in development when it is more costly to correct.

Technical debt is only one aspect considered in a SCRAM Review (see Figure 2). The interview process in a SCRAM typically reveals other things, as well, that were either not considered, or were poorly considered, when the schedule was created. By accounting for these omissions or poorly

estimated effects, SCRAM is able to come up with a more accurate estimate, and identify where the root cause of schedule slippage occurred.

SCRAM can be effectively applied during the proposal phase to determine the probability that the customer-desired delivery can be met, and if not, what strategies can be applied to better align capability with the desired date. It can be used during development to assess the probability that critical milestones will be met. It can also be used as a diagnostic tool when it becomes apparent that a critical milestone will not be met. Acquiring organizations can use SCRAM to assess the probability that their contractor will meet the desired delivery date.

3 Overview of SCRAM

A SCRAM Review is an independent, non -advocate and non-attributable engineering focused approach used to evaluate a program's schedule performance and identify risks to contractual schedule compliance. SCRAM was developed to benefit defense and industry decision makers and program managers by providing a method that assists experienced engineers and program controllers to consistently identify root causes of schedule slippage and recommend corrective actions.

SCRAM[1] uses two established scientific analysis techniques: Schedule Monte Carlo simulation to model estimation uncertainty and risk impacts and a dynamic software model to assist in forecasting schedule milestone completion based on objective data characterizing performance to date. SCRAM utilizes best practices from systems and software engineering together with schedule development and program execution. In addition, SCRAM facilitates improved organizational processes and practices based on feedback and systemic issues obtained from SCRAM Reviews.

There are three SCRAM delivery or application modes. These modes are:

- *Pre-emptive SCRAM* - conducted early in the Program life-cycle to avoid systemic issues and risks (e.g. ideally prior to contract award and/or an Earned Value Management - Integrated Baseline Review (EVM-IBR));
- *Assurance SCRAM* – conducted at any point in the program lifecycle to ascertain schedule performance is on track; and
- *Diagnostic SCRAM* - when a Program is experiencing significant schedule slippage i.e. a program is of interest or concern (similar to a US Nunn-McCurdy breach situation, i.e., one that requires notification to Congress if the cost per unit goes more than 25% beyond what was originally estimated, and calls for the termination of programs with total cost growth greater than 50%.).

In addition an organization acquiring a product or a service from a prime contractor can utilize SCRAM to independently evaluate if the contractor is likely to deliver when promised.

The SCRAM Review process, shown in Figure 1, is typically conducted over a two-week period on site followed by a written detailed report two weeks later. The SCRAM process is in accordance with the guidelines described in International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 15504 – Information technology process assessment.

[1] SCRAM uses SLIM-Control, a commercially available tool from Quantitative Software Management in McLean, Virginia.

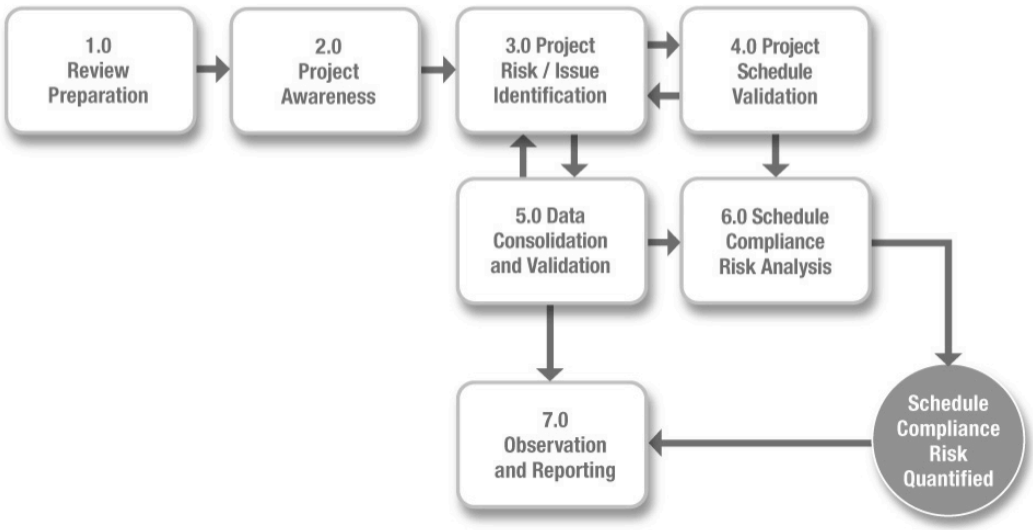

Figure 1 – High Level SCRAM Review Process

A SCRAM Review is conducted with seven Key Principles. These are:

- **Minimal Disruption**: information is collected one person at a time through interviews that typically last an hour. Artifact reviews (plans, detailed schedules) are conducted offline.
- **Independent**: SCRAM Team members are organizationally independent of the program under review. Some SCRAM Reviews have been joint contractor/customer team to facilitate joint commitment to resolving the review outcomes.
- **Non-advocate**: all significant issues and concerns are considered and reported regardless of the origin of the issue (customer or contractor).
- **Non-attribution**: the information source is not attributed to any individual instead focusing on identifying and mitigating the issues/risk.
- **Corroboration of Evidence**: significant findings and observations are based on at least two independent sources of corroboration.
- **Rapid turn-around**: only one to two weeks are spent on-site with an executive out-briefing presented at end of second week and a written report two weeks later.
- **Sharing Results, Openness and Transparency**: particularly for the parametric modelling component of a SCRAM Review, the organization under review may witness the data analysis process and challenge the results, and a preliminary out brief of findings is delivered prior to departure from the review site. This principle builds cooperation, trust and confidence in the schedule forecast.

SCRAM Reviews, data gathering, analysis and reporting in SCRAM Reviews are structured around the Root Cause Analysis of Schedule Slippage (RCASS) model shown in Figure 2. RCASS organizes program information into categories related to planning and executing a program.

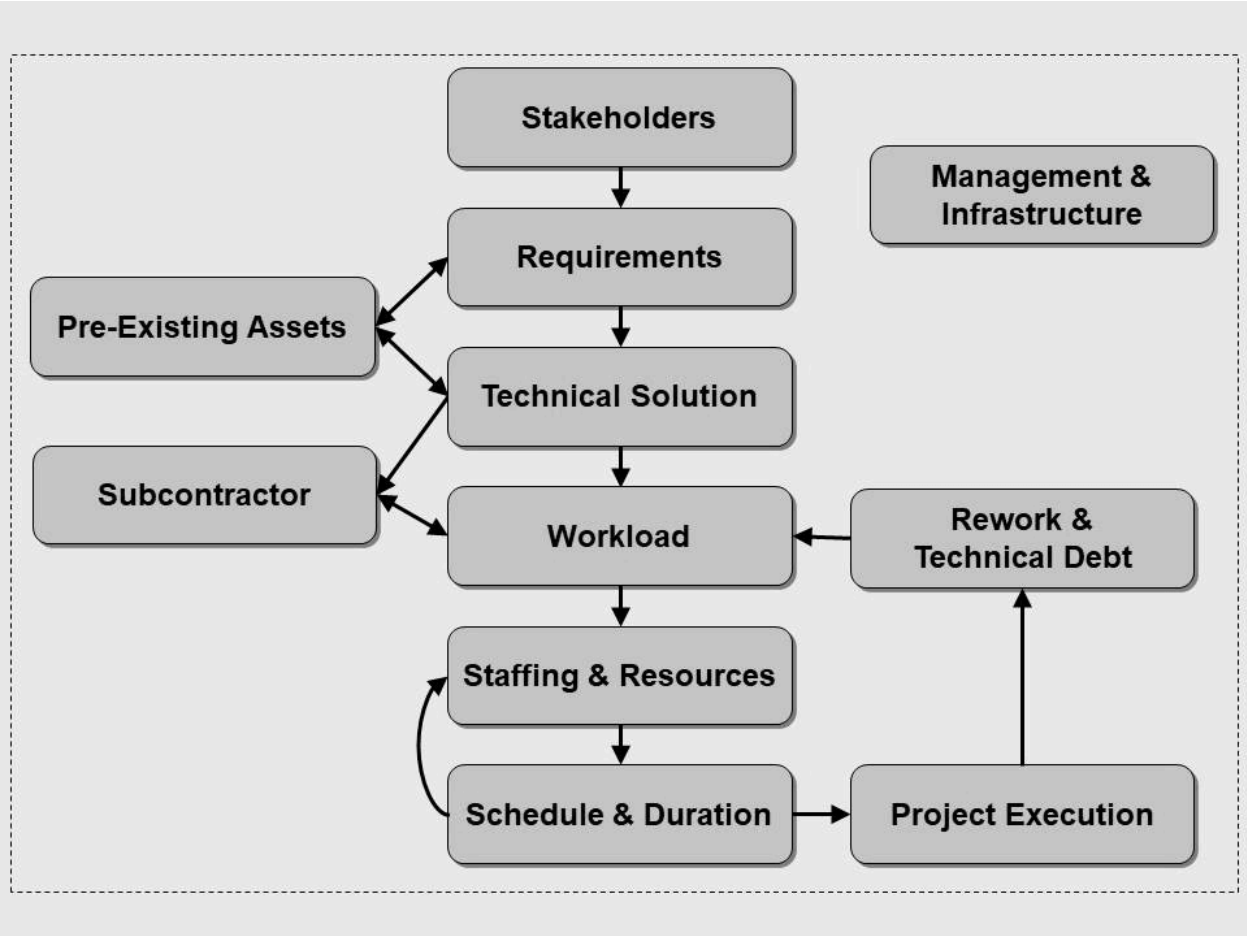

Figure 2 – Root Cause Analysis of Schedule Slippage (RCASS)

- The *Stakeholders* category reflects program turbulence and entropy because of difficulties in synchronizing the program's stakeholders: i.e., users, customers, system engineers, developers, maintainers, and others, relative to meeting commitments.
- The *Requirements* category reflects understanding and stability of the functional requirements, performance requirements, constraints, standards, for example, used to define and bound what is to be developed.
- The *Technical Solution* category reflects the design considerations and approaches needed to ensure that the chosen solution is architected, logically and physically designed to align with the business enterprise architecture, satisfy functional and non-functional requirements (quality attributes) and optimized to meet system development and sustainment life-cycle objectives.
- The *Pre-Existing Assets* category reflects products developed independently of the project that will be used in the final product, i.e. an asset that can reduce the amount of new work that has to be done on a project.
- The *Subcontractor* category reflects subcontractor products or services that will be delivered as a part of the overall deliverable system.
- The *Workload* category reflects the quantity of work that has to be done.
- The *Staffing and Resources* category reflects the availability, capability and experience of the staff necessary to do the work as well as the availability, and capacity of other resources, such as test and integration labs.

- The *Schedule and Duration* category reflects the tasks, sequencing and calendar time needed to execute the workload by available staff and other resources.
- The *Project Execution* category focuses on schedule, management and monitoring and controlling the execution of the program in accordance with the program schedule.
- The *Rework and Technical Debt* category reflects additional work caused by the discovery of defects in the product and/or associated artefacts, work that is deferred for short-term expediency (technical debt) and their resolution.
- The *Management and Infrastructure* category addresses factors that impact the efficiency and effectiveness of getting work done, e.g. work processes, use of management and technical software tools, or management practices.

A SCRAM review is conducted by a team of typically five people certified to participate in a SCRAM. It includes a certified team leader, two team members technically qualified to participate in a SCRAM for the type of project involved, a source material expert (SME), and a scheduling expert. Certification is granted by CASG after participation in a training program, demonstration that the candidate understands the SCRAM methodology, and passing an exam.

4 SCRAM Successes

SCRAM has been used for over ten years and has proven its value in a variety of application domains including aerospace, maritime, communications, space, training, logistics, and mission planning. The RCASS model is general and is applicable to software-intensive programs and to programs with no software involvement (e.g., ship maintenance). SCRAM reviews span both acquirer and contractor organizations. Much of SCRAM's success is due to the non-advocate, collaborative and transparent approach that serves to build trust and reduce defensiveness from programs and projects being reviewed. The following comments from the government Joint Program Office for the largest defense program ever undertaken, the F-35 Joint Strike Fighter, provides testimony to SCRAM usefulness:

> ***"The SCRAM reviews were very collaborative…When we went forward with the results to our senior leadership, it was a jointly endorsed assessment. You gave us plenty of time to concur with your assessment or not. In short, we felt SCRAM was great."***
>
> - William Urschel, F-35 Software Director, Joint Program Office (2012-2015)

Nowhere is the transparency and collaborative nature of SCRAM as important as when using program data to forecast critical milestone dates. In our experience, the first question typically asked by any program when asked to provide data for forecasting is "how can this hurt us?" This is a natural and understandable concern because data is often gathered and used to make forecasts without a full understanding of the context surrounding the data. People often take data and "run with it." In contrast, part of the SCRAM forecasting methodology is to spend great care ensuring that data is understood, is summarized correctly, that any apparent anomalies are explained or corrected and only then, is data entered into the dynamic forecasting model. In addition, contractors are given the opportunity to observe the modeling and are presented with the results prior to anyone else seeing them (including the government program office). This goes a long way to dispelling concerns. Equally important, it greatly increases the likelihood of having valid, correct data.

Over the years, the SCRAM milestone forecasts have proven to be quite accurate. SCRAM Reviews included forecasts for the F-35 onboard software (approximately nine million lines of source code) and the off-board support software encompassing logistics, training, and mission planning. A testimony as

to the accuracy of those forecasts was provided by Lt. Gen. Christopher Bogdan, the former Program Executive Officer for the F-35 Program:

> ***"SCRAM gave us new techniques for measuring the progress of software development and for predicting how long the software development was going to take. In 2014, I briefed the SCRAM results to the Defense Acquisition Board. Of all the organizations that were making estimates, the SCRAM estimates, in hindsight, were the most accurate."***
>
> -- Lt. Gen. Christopher Bogdan, Program Executive Officer, F-35 Program (2012-2017)

Perhaps no other evidence is more telling of the value of SCRAM than the fact that several complex, multi-phase programs have requested multiple SCRAM Reviews over a period of years at their cost and on their own initiative. In the words of one program staff member:

> ***"We know when our program is going off the rails but we often don't know why. The SCRAM Team can bring independent eyes to quickly identify why things are starting to go off track. That's been very helpful for us"***
>
> ***Mr. Robert Jackson, Chief Engineer, ANZAC System Program Office, Australian DoD***

5 Conclusion and Summary

The private and public sectors of commerce have been plagued by projects that are delivered late, over budget, and sometimes lacking the required quality. Industry has adopted models and standards to improve quality and the ability to deliver on time and within budget (such as the CMMI and ISO 9000), but still many projects fall behind schedule even when developed by organizations rated high on the CMMI or ISO 9000. SCRAM has been demonstrated to be a very effective method to determine the ability to meet critical milestones, and can be used to great effect by development contractors and by acquiring organizations to monitor their contractor's ability to meet critical dates. It is a very effective adjunct to the CMMI, ISO 9000, and other process improvement models and standards.

We believe there is no reason to abandon the use of PPMs. SCRAM experience indicates that it is a highly valuable addition to the program management toolkit. Currently under development by the SCRAM development team is a detailed taxonomy of technical implementation risks called TIRA (**T**echnical **I**mplementation **R**isk **A**ssessment). For example, decisions that are made concerning the development approach to be taken (e.g. custom vs commercial off-the-shelf (COTS) based development) can result in a specific set of risk areas. The risks identified in TIRA will facilitate development teams in better identifying the risks that may impact their projects and its schedule and to take steps to mitigate those risks.

6 References

Nunn-McCurdy Breach, *Defense Acquisition Guidebook, Chapter 10*

Pitman, Adrian; Tuffley, Angela and Clark, Betsy. SCRAM: A method for assessing the risk of schedule compliance [online]. In: 22nd Australasian Software Engineering Conference: ASWEC 2013. Barton, A.C.T.: Engineers Australia, 2013: 45-58.

Chrissis, Mary Beth, Konrad, Mike, and Shrum, Sandy. *CMMI for Development*, Addison Wesley, Third Edition, 2011.