SCRAM: A Method for Assessing the Risk of Schedule Compliance

Adrian Pitman
Defence Material Organisation
Australia

Angela Tuffley
Red Bay Consulting Pty Ltd
Australia

Betsy Clark
Software Metrics Inc.
USA

Abstract

Schedule slippage is an unfortunate reality for many large development software intensive systems engineering programs particularly when the project enters the critical systems integration and testing phase.

The Schedule Compliance Risk Assessment Method (SCRAM) was developed as a collaborative effort between the Australian Defence Material Organisation, Software Metrics Inc and RedBay Consulting. It utilises a framework for organizing the information gathered during an assessment, to identify the root causes of slippage, to communicate the reasons behind the schedule slippage, and for structuring recommendations to address the root causes.

This framework, referred to as the Root Cause Analysis of Schedule Slippage model (or RCASS) depicts the major areas impacting schedule and the relationships between these areas. SCRAM includes schedule health checks to evaluate the construction and logic of the program schedule, schedule risk analysis (SRA) to show the probability distribution of completion dates and software metric analysis (SMA) to estimate and benchmark software development.

To date, SCRAM has been applied to a number of defence programs in Australia and the US. These assessments have typically been applied when schedule problems occur during the Integration and Testing phase. However prior to this critical phase, SCRAM can be used at the commencement of a program to identify and mitigate potential risks or during program execution to monitor schedule.

In 2010, the SCRAM Process Reference / Assessment Models were developed as ISO/IEC 15504 Information technology Process assessment conformant models in an effort to maximize consistency and provide an objective framework for assessment. This effort, funded by the Australian Defence Materiel Organisation, was undertaken to support SCRAM as it is rolled out to a wider audience of users.

This presentation will discuss the SCRAM Methodology, the RCASS Model that underpins the root causes of schedule slippage, use of schedule risk analysis (SRA) and software metric analysis (SMA) including real-world examples of successful assessments using SCRAM to identify schedule risks both prior and during the systems integration and testing phase.

Keywords

Schedule risk assessment, Schedule risk analysis, Software metric analysis
SCRAM: A Method for Assessing the Risk of Schedule Compliance

ASWEC 2013

Adrian Pitman
Director Acquisition Engineering Improvement
Defence Materiel Organisation

Angela Tuffley
Director
RedBay Consulting Pty Ltd

Dr Betsy Clark
President
Software Metrics Inc. USA

Topics

• The SCRAM Methodology
  – Background
  – The Root Cause Analysis of Schedule Slippage (RCASS) Model

• Technical Issues that drive Schedule
  – Rework & Technical Debt
  – System Integration
  – Technical Progression

• Quantifying and Independently Forecasting Schedule Risk
  – Schedule Risk Analysis (SRA)
  – Software Metric Analysis (SMA)

• SCRAM Review Experiences
**Topics**

- The SCRAM Methodology
  - Background
  - The Root Cause Analysis of Schedule Slippage (RCASS) Model

- Technical Issues that drive Schedule
  - Rework & Technical Debt
  - System Integration
  - Technical Progression

- Quantifying and Independently Forecasting Schedule Risk
  - Schedule Risk Analysis (SRA)
  - Software Metric Analysis (SMA)

- SCRAM Review Experiences

---

**SCRAM: History and Context**

- Schedule Compliance Risk Assessment Methodology (SCRAM) has evolved from reviews of Projects of Interest and Projects of Concern

  *Schedule is almost always the primary concern of project stakeholders (delays to war fighter capability unacceptable)*

- SCRAM is a key component of the DMO’s initiative to identify, remediate and eliminate root causes of schedule slippage
What SCRAM Is (and is NOT)

• **SCRAM Is**
  – a technical, schedule-based risk analysis methodology
  – identifies and quantifies the risk/s to schedule compliance
    • quantifying risk requires a valid project schedule
  – used to identify root causes of schedule slippage so that appropriate actions can be taken
  – embodies technical and scheduling best practices

• **SCRAM is Not**
  – an assessment of organisational maturity / process capability
    • however, may be identified and treated as an issue if process performance is identified as contributing to slippage
  – focusses on project outcomes

SCRAM Key Principles

• Independent

• Non-advocate

• Non-attribution

• Corroboration of Evidence

• Sharing Results, Openness and Transparency

• Minimal Disruption / Rapid Turn-Around
The SCRAM Team

- Assessment conducted by a small team including:
  - Core of experienced system and software engineering assessors
    - to validate engineering related BoEs and work load estimates
    - Identify/understand technical project issues and risks and
    - provide inputs for schedule risk assessment
    - recommend remediation actions for technical issues
    - Supplemented by domain specific subject matter experts as necessary e.g. specialist aircraft production engineer or flight test engineer
  - Scheduler experienced in the applicable project schedule tool
    - validates schedule – conducts schedule health checks
    - performs Monte Carlo risk modelling with probabilistic inputs from engineering assessors

What Causes Projects to Slip?

- There are multiple causes of schedule slippage:
  - poor planning and schedule construction
  - issues that arise during schedule execution

- Once root causes have been identified, they can be remediated
How can we find the cause?

• Program managers are flooded with a wealth of data and details and Schedule slippage is a symptom of many factors
  – Challenge is to organize all of this information to
    • Identify causes of schedule slippage
    • Take effective action to address problems

• SCRAM is based on a Root Cause Analysis of Schedule Slippage (RCASS) model
  – Organising the information based on SCRAM should:
    • De-clutter the massive amounts of information on a project
    • Relate the different issue areas to each other
    • Highlight missing information

• In addition, risk to schedule slippage is identified through the SCRAM Risk Analysis / Monte Carlo Simulation

Root Cause Analysis of Schedule Slippage (RCASS) Model

[Diagram showing the RCASS model with stakeholders, management & infrastructure, subcontractor, requirements, workload, staffing & resources, schedule & duration, project execution, rework & technical debt, and pre-existing assets]
Root Cause Analysis of Schedule Slippage (RCASS) Model

- Model has evolved from experiences when conducting SCRAM assessments
- Used as guidance for
  - Asking questions during SCRAM Assessments
  - Sorting the wealth of data and details gathered during an assessment into categories
  - Highlighting missing information
  - Determining the root causes of schedule slippage
  - Recommending remediation activity
  - Recommending measures to serve as leading indicators
    • For visibility and tracking in those areas where there are risks and problems

Topics

- The SCRAM Methodology
  - Background
  - The Root Cause Analysis of Schedule Slippage (RCASS) Model
- Technical Issues that drive Schedule
  - Rework & Technical Debt
  - System Integration
  - Technical Progression
- Quantifying and Independently Forecasting Schedule Risk
  - Schedule Risk Analysis (SRA)
  - Software Metric Analysis (SMA)
- SCRAM Review Experiences
Technical Issues that Drive Schedule

• Rework
  – Additional work caused by changing requirements or the discovery of defects in the product and/or associated artefacts
  – Often not estimated or planned for

• Technical Debt
  – Work that is deferred for short-term expediency
  – Intentional and unintentional technical debt should be understood, identified, measured and managed
  – Implications of “when” or “if” to repay the debt
    • principal plus compound interest
  – Results in necessary rework, for example
    • shortfalls in architecture
    • suspending peer reviews
    • failure to qualify CSCIs

Technical Issues that Drive Schedule

• System Integration
  – Experience and root cause analysis of schedule slippage has shown this activity to be a major systemic contributor to project schedule slippage
  – Ensure the system integration process is properly planned, formally documented, adequately resourced and implemented in accordance with the plan
  – System Integration requirements, workload estimation, resource requirements and scheduling need to be addressed much earlier in the project life cycle
  – System Integration resource requirements (SI Labs/equipment strings/tools, integration engineers, etc) are calculated based on system size and expected system development performance metrics (e.g. expected defect density)
  – System integration activities (system debug/grooming and design verification) should be separated from and completed prior to formal system acceptance testing.
Technical Issues that Drive Schedule

• Technical Progression
  – measure product technical progress towards planned completion
  – ensure it is commensurate with expended scheduled effort
    • (i.e. technical growth is aligned with schedule execution)
  – periodically evaluated throughout the system development lifecycle based on
    • KPPs, TPMs
    • evidence based technical reviews
    • system and software engineering leading indicators
      – e.g. defect error correction trends
    • MBSE, modeling and simulation, prototyping
    • Independent architectural evaluations

Topics

• The SCRAM Methodology
  – Background
  – The Root Cause Analysis of Schedule Slippage (RCASS) Model

• Technical Issues that drive Schedule
  – Rework & Technical Debt
  – System Integration
  – Technical Progression

• Quantifying and Independently Forecasting Schedule Risk
  – Schedule Risk Analysis (SRA)
  – Software Metric Analysis (SMA)

• SCRAM Review Experiences
Schedule Risk Analysis/Monte Carlo

- Rate Tasks on the Critical & Near Critical Path
  - Assign three point estimates
    - Most Likely, Optimistic and Pessimistic
  - based on the risks identified and expert judgement
- Perform Monte Carlo Simulation
  - provides a picture of the potential impact of risk on schedule
- Projects must use the results of the SRA to develop action plans to ensure that the risks don't become reality

SCRAM Schedule Risk Assessment

1. Develop a complete critical path network and prepare for SRA.
2. Identify critical milestones for risk quantification.
3. Enter risk parameters.
4. Run schedule simulation & quantify impact of risk on schedule.
5. Analyze schedule results.
7. Present position to program office. Develop risk mitigating actions.

The impact of project risk is assessed and critical risks fed back for risk reduction

Incorporate into Risk Management Processes

Courtesy NAVAIR 4.2
Software Metric Analysis (SMA)

- In addition to conducting a Schedule Risk Analysis, a SCRAM assessment uses parametric modelling to forecast software completion.
- A parametric estimation model is a statistical tool with parameters (e.g., estimated size, complexity, programmer experience) to describe the characteristics of a software development.
  - Based on the input parameters, the model estimates duration/schedule, effort/staffing, and defects.
- SCRAM uses a parametric model that uses actual performance metrics to date to forecast software completion.
- Inputs include
  - Total size in source lines of code
  - Defects discovered
  - Major milestones completed
  - Staffing

Example Forecast
Topics

- The SCRAM Methodology
  - Background
  - The Root Cause Analysis of Schedule Slippage (RCASS) Model

- Technical Issues that drive Schedule
  - Rework & Technical Debt
  - System Integration
  - Technical Progression

- Quantifying and Independently Forecasting Schedule Risk
  - Schedule Risk Analysis (SRA)
  - Software Metric Analysis (SMA)

- SCRAM Review Experiences

Diversity of SCRAM Assessments

- Completed approximately 10 assessments on projects in the following domains
  - Aerospace (25MSLOC)
  - Maritime
  - Logistics / ERP (4+MSLOC)
  - Training
  - Battlespace Command and Control
  - Communications / Telecommunications

- To date, assessments have been conducted relatively late in the development life cycle, during System Integration and Test. At this point, problems can no longer be deferred or hidden.
SCRAM Review Experiences

• Suspension of peer reviews led to a bow wave of defects found in System Test

• The master project schedule was not available to program staff or stakeholders
  – Undergoing a schedule tool transition for approx. 2 years

• Late in development, a critical stakeholder (customer) added a condition for acceptance that removed three months from the development schedule

• A large ERP project had two system specifications – one with the sponsor/customer and a different specification under contract with the developer – would this be a problem?

• Prime and sub-contractor schedule not linked or aligned

SCRAM Review Experiences (cont.)

• Over 200 instantiations of Open Source code resulted in a breach of IP and indemnity requirements

• Re-plan based on twice the historic productivity with no basis for improvement

• Scheduling staff for 12 hours a day (to recover schedule)

• No effective integrated master schedule to provide an overall understanding of the completion date of the project
  – 13 subordinate schedules
  – Critical Path went subterranean!

• Inter-organisational relationships were major drivers of project schedule slippage
Questions?

Contact Details:
Government: Adrian Pitman: adrian.pitman@defence.gov.au
Australia: Angela Tuffley: a.tuffley@redbay.com.au
USA: Betsy Clark: betsy@software-metrics.com