DMO

Australian Government
Department of Defence
Defence Materiel Organisation

SCRAM: A METHOD FOR ASSESSING THE RISK OF SCHEDULE COMPLIANCE

San Diego INCOSE Chapter – 19 June 2013

Presented by Angela Tuffley
DMO SCRAM Principal
Director – RedBay Consulting Pty Ltd

Adrian Pitman
DMO SCRAM Principal
Director Acquisition Engineering Improvement
Defence Materiel Organisation

Dr Betsy Clark
DMO SCRAM Principal
President
Software Metrics Inc

WHAT DOES SCRAM MEAN?

- Go away!
- Secure Continuous Remote Alcohol Monitoring
  - As modeled here by Lindsay Lohan
- Schedule Compliance Risk Assessment Methodology

SCRAM

Schedule
Compliance
Risk
Assessment
Methodology

- Collaborative effort:
  - Australian Department of Defence
    - Defence Materiel Organisation
  - RedBay Consulting, Brisbane, Australia
  - Software Metrics Inc., Haymarket, VA

DMO SCRAM USAGE

- DMO equips and sustains the Australian Defence Force (ADF)
  - Manages 230+ Major Capital Equipment Projects & 100 Minor (<$20M) defence projects
- SCRAM has been sponsored by the Australian Defence Materiel Organisation (DMO)
  - To improve Project Schedule Performance in response to Government concern as identified by the Australian National Audit Office (ANAO)
    - ANAO is equivalent to the US Government Accountability Office (GAO)

DMO SCRAM - CONTEXT

- SCRAM is an assessment methodology that has evolved from reviews of DMO Projects of Interest and Concern

Schedule is almost always the primary concern of our project stakeholders

- SCRAM is a key component of DMO's initiatives to improve acquisition schedule performance:
  - SCRAM is used to Identify and remediate or mitigate root causes of schedule slippage and risk to schedule compliance

WHAT SCRAM IS (AND IS NOT)

- **SCRAM Is:**
  - A system engineering-based assessment of risk to schedule compliance
  - Used to identify all root causes of schedule slippage (not just technical) so that appropriate remediation action can be taken
- **SCRAM Is Not:**
  - An Audit. (It does not focus on non-conformance. SCRAM provides knowledge of issues and risks driving schedule so they can be eliminated)
  - A Process Assessment (like CMMI)
    - but SCRAM can identify and treat poor process performance as an issue if process is driving schedule slippage

WHAT SCRAM IS

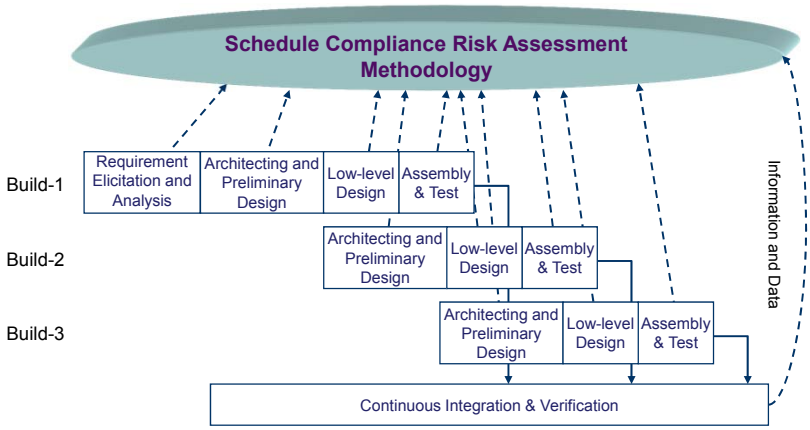
- **SCRAM:**
  - Quantifies the risk to project schedule compliance using two different scientific techniques (where possible)
    - Schedule Monte Carlo Simulation and
    - Software Parametric Analysis/Modelling
  - Embodies system engineering, schedule development and project execution management best practices
  - Recommends actions to mitigate risks and remediate issues that are driving project schedule slippage
  - Uses knowledge gained on assessments (i.e. systemic root causes / issues) to improve DMO business practices and the SCRAM process

SCRAM IS DESIGNED TO INFORM SENIOR LEADERSHIP

- **SCRAM Out-Brief and Reporting**
  - The most significant issues and risks are identified and incorporated into a high level 'Summary Finding'
  - The Summary Finding is presented as the Bottom Line Up Front (BLUF) in briefing and reports
  - Traffic light indicators support understanding of the detailed findings
  - Impact of findings are explained, along with the observations that lead to a finding
  - Executive summaries, along with detailed findings and observations are included in the final report

APPLICATION ACROSS THE DEVELOPMENT LIFECYCLE

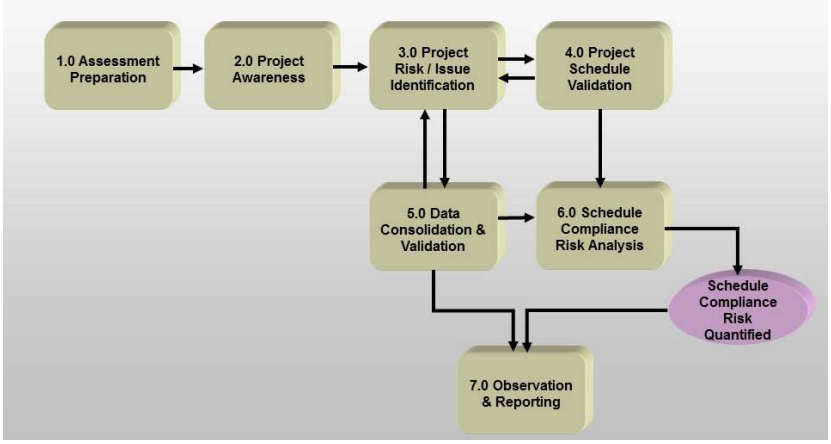
There is a wealth of information to feed the model



DIVERSITY OF SCRAM ASSESSMENTS

- The DMO SCRAM Team has completed 10 assessments on DMO major projects in the following domains:
  - Aerospace; Maritime; Logistics Enterprise Resource Program; Training; Command and Control; Communications / Telecommunications
- To date, assessments have been conducted relatively late in the system development life cycle, predominantly during System Integration and Test, when problems can no longer be deferred or hidden
- SCRAM is gaining some recognition through its recent use in assessing the F-35 JSF program

THE SCRAM HIGH LEVEL ASSESSMENT PROCESS



SCRAM ASSESSMENT KEY PRINCIPLES

- **Minimal Disruption**
  - Information is collected one person at a time
  - Interviews typically last an hour
- **Rapid turn-around**
  - One week spent on-site, second week data consolidation and corroboration nearby
  - Executive out-briefing presented at end of second week
  - Written report two weeks later
- **Independent**
  - Review team members are organisationally independent of the program under review
    - (Joint Contractor/Defence teams composition has been used successfully. Members must be independent of the specific Program organisations)

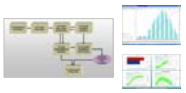
SCRAM ASSESSMENT KEY PRINCIPLES

- **Non-advocate**
  - All significant issues and concerns are considered and reported regardless of origin or source
    - Customer, Stakeholder or Contractor
  - Some SCRAM reviews have been joint contractor/customer team
    - facilitates joint commitment to resolve review outcomes
- **Non-attribution**
  - Information obtained is not attributed to any individual
  - Focus is on identifying and mitigating the issues/risk
- **Corroboration of Evidence**
  - Significant Findings and Observations are based on at least two independent sources of corroboration

SCRAM ASSESSMENT KEY PRINCIPLES

- **Openness and Transparency**
  - For the software parametric analysis component of a SCRAM assessment, the contractor is invited to
    - assist in resolving data anomalies
    - witness the data analysis process and
    - challenge model results
  - Sharing of results (no surprises)
    - Builds cooperation and trust
  - Builds confidence in the schedule forecast
  - SCRAM Team is the final arbiter

ELEMENTS OF THE SCRAM PRODUCT SUITE

Root Cause Analysis of Schedule Slippage (RCASS) Model	
SCRAM Process Reference/Assessment Model (PR/AM)	
Assessment Process and Techniques • **Schedule Monte Carlo Simulation** • **Software Parametric Modelling (Forecasting)**	
SCRAM PRAM Model and Assessor Training Courses	
SCRAM Assessor Guidebook	

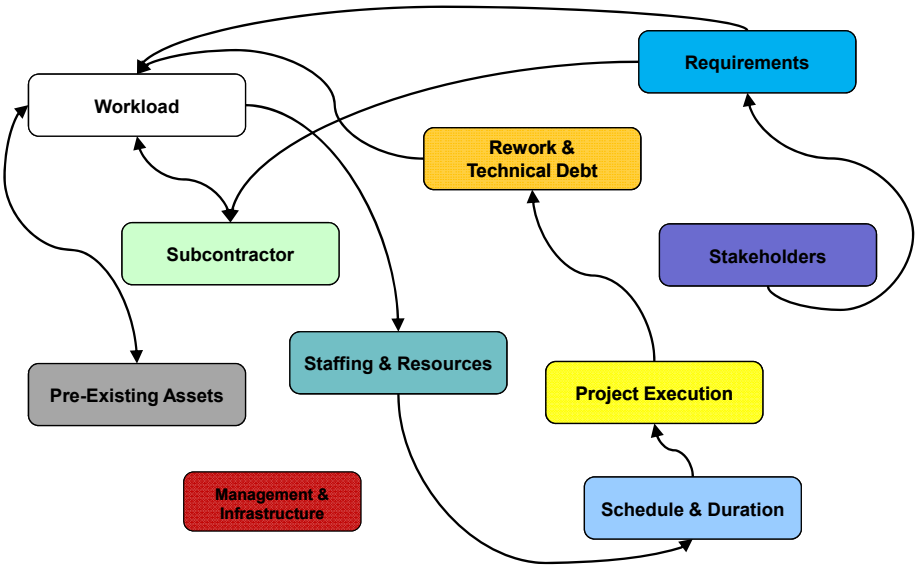
15

IDENTIFYING ROOT CAUSES OF SCHEDULE SLIPPAGE

- Program Managers are flooded with a wealth of information, data and detail
  - The challenge is to organise all of this information in a way to:
    - Identify the cause(s) of schedule slippage and risk to schedule compliance
    - Allow Project Managers to focus action to remove the root causes and mitigate risks
- SCRAM organises the available information based on the Root Cause Analysis of Schedule Slippage (RCASS) model. This approach:
  - De-clutters and organises the massive amounts of information on a project
  - Relates the information, assessment observations, risk indicators and root cause symptoms in a way that facilitates the correct identification of root causes

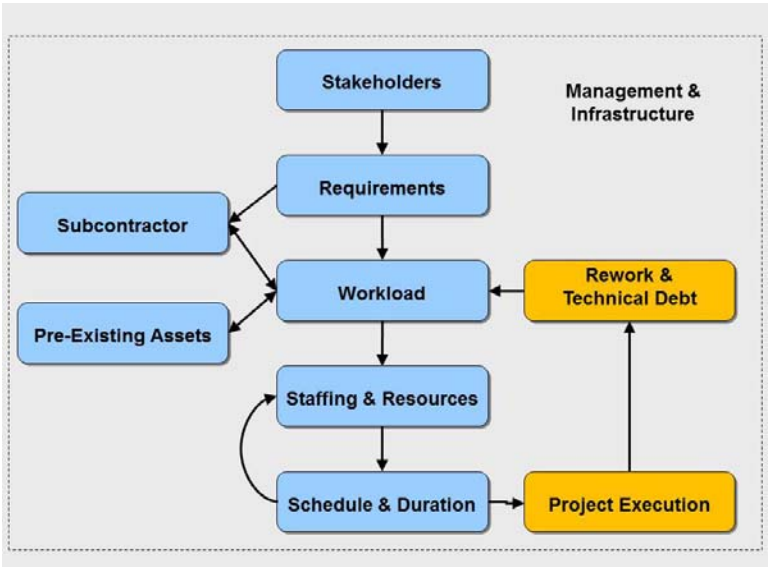
16

CATEGORIES OF PROJECT INFORMATION



17

ROOT CAUSE ANALYSIS OF SCHEDULE SLIPPAGE (RCASS) MODEL



19

ROOT CAUSE ANALYSIS OF SCHEDULE SLIPPAGE (RCASS) MODEL

- RCASS is a 'thought' model that has evolved from our experiences when conducting assessments. RCASS:
  - Shows logical dependencies and linkages between elements and phases of system development
  - Covers both the Project Preparation and Project Execution phases that contribute common root causes that drive schedule slippage

- RCASS is used in a SCRAM Assessment as guidance to:
  - Focus and guide questions to identify causal chain factors
  - Categorise the wealth of data and details gathered during an assessment
  - Ensure complete coverage and highlight missing information
  - Determine the root causes of schedule slippage
  - Identify appropriate measures to serve as leading indicators
    - For visibility and tracking residual risks and risk thresholds

STAKEHOLDERS

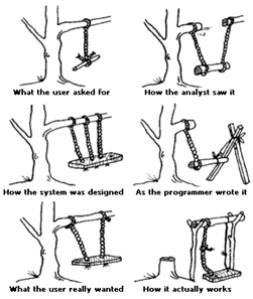
Reflects project turbulence because of difficulties in synchronising the project's stakeholders: users, customers, system engineers, developers, maintainers, others.

"Our stakeholders are like a 100-headed hydra – everyone can say 'no' and no one can say 'yes'."

SCRAM Examples

- Late in development, a critical stakeholder (customer) added a condition for acceptance that removed three months from the development schedule
- Failed organisational relationship led to breakdown in communication
  - Key stakeholders were not talking to each other even though they were in the same facility
  - Communication degraded to memos slipped under the door

REQUIREMENTS



Reflects understanding and stability of the functional requirements, performance requirements, system constraints, standards, etc. used to define and bound what is to be developed

SCRAM Examples

- Misinterpretation of a communication standard led to an additional 3,000 derived requirements to implement the standard.
- 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?

SUBCONTRACTOR

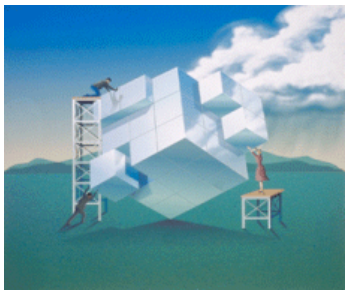
Reflects subcontractor products or services that will be delivered as a part of the overall software system.

If the subcontractor doesn't perform, additional work required by the Prime

SCRAM Examples

- Subcontractor omitting processes in order to make delivery deadlines led to integration problems with other system components
- Prime and sub-contractor schedule not linked or aligned

PRE-EXISTING ASSETS



Reflects products developed independent of the project that will be used in the final product, i.e. an asset that reduces the amount of new work that has to be done on a project.

"It doesn't do what we thought "
"There is a lot of functionality we don't need."

SCRAM Examples

- COTS products that didn't work as advertised, resulting in additional work
- COTS product required a "technology refresh" during development as the project was years late (cost the project $8M)

WORKLOAD

Reflects the quantity of work to be done and provides a basis for estimating effort/staffing and duration

"Unrealistic expectations based on inaccurate estimates are the single largest cause of software failure."
» Futrell, Schafer

SCRAM Examples

- Source lines of code is typically underestimated
- Contract data deliverables workload often underestimated by both contractor and customer
- System Integration labs underestimated resulting in additional cost to build another lab (inadequate basis of estimates)
- Re-plan based on twice the historic productivity with no basis for improvement

STAFFING & RESOURCES

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.

Bringing on people to solve a slippage problem may make it worse (especially late in the project)

SCRAM Examples

- Parent company sacking workers as soon as job is completed
  - Workers went on a "Go Slow"
- Scheduling staff for 12 hours a day (to recover schedule)
- Lack of fidelity and qualification of integration and test lab

SCHEDULE & DURATION

Reflects the task sequencing and calendar time needed to execute the workload by available staff and other resources (e.g. test labs).

SCRAM Examples

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

PROJECT EXECUTION

Focuses on monitoring and controlling the execution of the project in accordance with the project schedule

Experience from multiple SCRAM assessments has highlighted the need to focus on System Integration and Technical Progression

SCRAM Examples

- The schedule was not available to program staff or stakeholders
  - Undergoing a schedule tool transition for approx. 2 years
- Five delivery iterations not scheduled before CDRL approvals
- No System Integration Plan
- No “red” risks on a program undergoing a major contract overrun breach

REWORK AND TECHNICAL DEBT

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

Technical Debt includes suspension of peer reviews, short-cuts in unit test.

Rework is often underestimated or not planned for.

SCRAM Examples

- Suspension of peer reviews led to a bow wave of defects found in System Test
- Accrual of Technical Debt with no repayment plan

MANAGEMENT & INFRASTRUCTURE

Addresses the factors that impact the efficiency and effectiveness of getting work done, e.g. work processes, use of management and technical software tools, management practices, etc.

Includes processes for Verification and Validation, Infrastructure, Quality Assurance, Process Improvement and Configuration Management

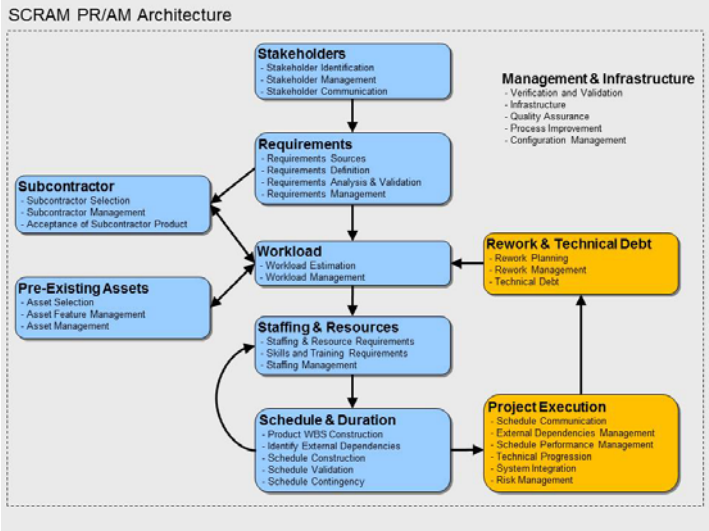
SCRAM Examples

- Change management process and tools could not keep pace with requirements for System Integration and Test
- Lack of software quality assurance

SCRAM PROCESS REFERENCE/ASSESSMENT MODEL (PR/AM)

- The **SCRAM PR/AM** Processes and Best Practices are used to:
  - Provide guidance to project staff on the preparation and construction of a project schedule and best practices to help avoid common causes of Schedule Slippage during project execution
    - The PR/AM is periodically updated to capture new lessons learned from SCRAM reviews of troubled projects
  - Guide SCRAM assessment coverage to uncover root causes of schedule slippage
- The PR/AM can be used to determine a Program's schedule performance capability maturity (but not organisational maturity)
- Currently contains the 37 process areas which are grouped under the 10 RCASS Categories

RCASS AND SCRAM PR/AM MODELS COMBINED



Available to download from www.scramsite.org

EXAMPLE PR/AM PROCESS: TECHNICAL DEBT

RTD-TD – Technical Debt

Purpose

The purpose of Technical Debt is to ensure both intentional and unintentional technical debt is understood, identified, measured and managed.

Note: Technical Debt is a concept that applies to both systems and software engineering.

Outcomes

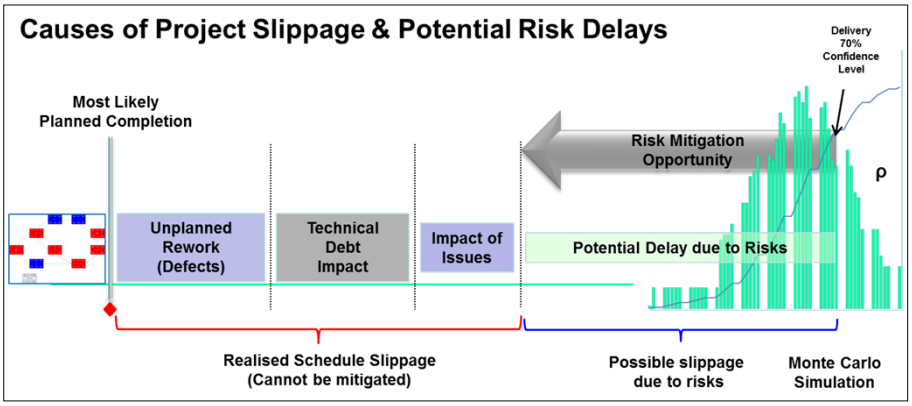
RTD-TD-1: Decisions to incur Technical Debt are made fully cognizant of implications, impacts and risks (schedule, cost and technical)

RTD-TD-2 :Unintentional Technical Debt is identified and managed

RTD-TD-3: Technical Debt Repayment is estimated, resourced and incorporated into the Project Schedule

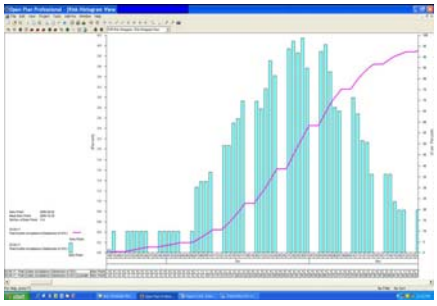
RTD-TD-4: Technical Debt is recorded and periodically reviewed

SCRAM IDENTIFIES AND QUANTIFIES SCHEDULE SLIPPAGE ROOT CAUSES AND RISK

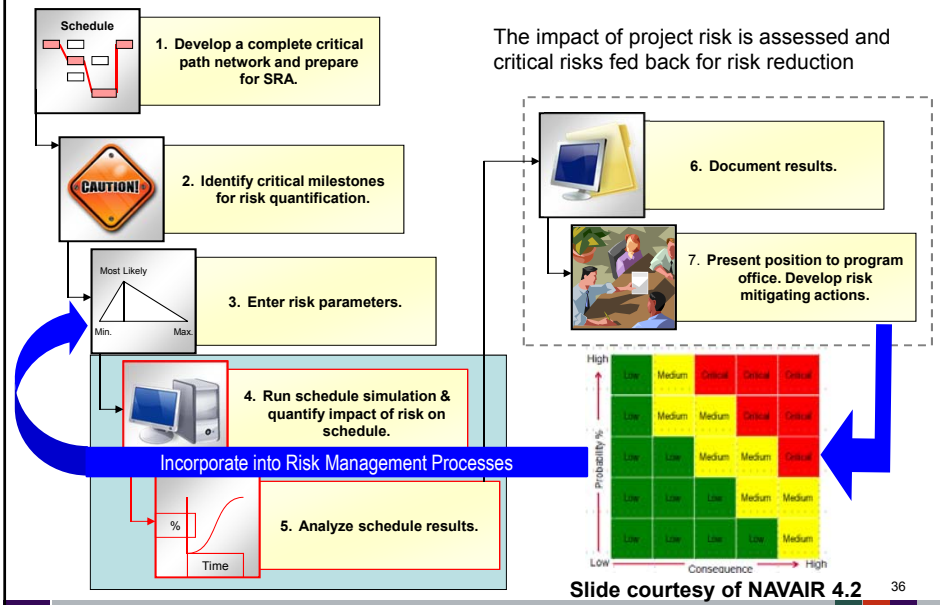


SCHEDULE RISK ASSESSMENT/MONTE CARLO

- Assign three point estimates (Most Likely, Optimistic and Pessimistic) to Tasks that are on the Critical or Near Critical Path
  - based on identified risks, issues, rework, technical debt and any other sources of delays
- 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 remediation or mitigation plans to ensure that the risks don't become reality



SCRAM - SCHEDULE RISK ASSESSMENT

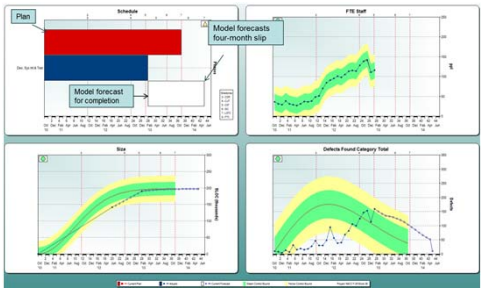


Slide courtesy of NAVAIR 4.2

PARAMETRIC MODEL FORECAST

- In addition to conducting a Schedule Risk Analysis, SCRAM uses a parametric model, QSM SLIM-Control, that uses actual performance data to forecast software completion

- Inputs include
  - Source lines of code (SLOC – estimated total and SLOC completed to date)
  - Defects discovered
  - Major milestones completed
  - Staffing Plan



TECHNICAL ISSUES THAT DRIVE SCHEDULE

- From experience and root cause analysis the following areas have emerged as the main technical issues that cause a project schedule to slip
  - Rework
  - Technical Debt
  - System Integration
  - Technical Progress
- Other technical issues on the project may be present but the above are often neglected

REWORK

- Additional work caused by changing requirements or the discovery of defects in the product and/or associated artefacts
- 2 processes that specifically address Rework
  - RTD-PLN Rework Planning
  - RTD-MAN Rework Management
- Often NOT estimated or planned for
- Defects can occur in any system element including documentation, hardware, software, test procedures and project development processes and drive rework

REWORK: KEY FACTORS

- Has the project estimated, resourced, scheduled and planned for rework?
  - Particularly software defect removal
- Is this based on historical evidence?
  - If there is no history, industry benchmarks indicate that Rework accounts for about 40% of effort in software development
- Have they "optimistically" increased their defect removal rate with no process improvements / tools to support the increase?
- Just because it not software doesn't mean there is no rework!
  - Defects can occur in any system element

TECHNICAL DEBT

- Work that is deferred for short-term expediency
  - Intentional technical debt
    - strategic decision to address a short term problem
      - e.g. the need to meet a dead-line
  - Unintentional technical debt
    - from doing a poor job
- 1 process that specifically addresses Technical Debt
  - RTD-TD Technical Debt
- Technical Debt = Principal plus Compound Interest
- Results in necessary rework

TECHNICAL DEBT: KEY FACTORS

- Does the project understand, identify, measure and manage both intentional and unintentional technical debt?
- Does the project know the implications of
  - "when" or "if" to repay (principal plus compound interest) the debt?
  - the impact on schedule, cost and risk?
- Some example project activities that cause Technical Debt (as seen on SCRAM reviews)
  - shortfalls in architecture
  - suspending peer reviews
  - failure to qualify CSCIs prior to system integration and test

42

SYSTEM INTEGRATION

- A major systemic contributor to project schedule slippage
- 1 process and 4 Outcomes that specifically address System Integration

Process/Outcome	Process	Description
EXE-SI	System Integration	
REQ-SRC-6	Requirements Sources	System Integration and Test Requirements are identified
WRK-EST-1	Workload Estimation	Work Units are identified for each Major Activity (includes System Integration)
SAE-REQ-3	Staffing & Resource Requirements	System Integration and Test Resources are estimated based on Work Unit Estimates
SAD-CON-6	Schedule Construction	The Schedule for Planning, Construction and Qualification of System Integration and Test Resources is aligned with the Master Project Schedule

- A significant number of errors are present when the system exits development and enters system integration and test

43

SYSTEM INTEGRATION: KEY FACTORS

- Has the project planned, documented, adequately resourced and implemented a System Integration Plan?
- Has the project addressed System Integration requirements, workload estimation, resource requirements and scheduling early in the project life cycle?
- Has the project calculated System Integration resource requirements (SI Labs/equipment strings/tools, integration engineers, etc) based on
  - system size and expected system development performance metrics (e.g. expected defect density)
- Are System Integration activities (system debug/grooming and design verification) separated from and completed prior to formal system acceptance testing?

TECHNICAL PROGRESSION

- Measures product technical progress towards planned completion
- 1 process that specifically addresses Technical Progression
  - EXE-TEP Technical Progression
- 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, modelling and simulation, prototyping
  - independent architectural evaluations

TECHNICAL PROGRESSION: KEY FACTORS

- Is Technical Progression on the project commensurate with expended scheduled effort?
  - (i.e. technical growth is aligned with schedule execution)
- Are Technical Performance Measures (TPMs) derived from Key Performance Parameters (e.g. weight, time etc) and apportioned or budgeted across components/elements of the system?
- Does the project conduct Evidenced Based Design Reviews to:
  - assess the maturity of the design and identify Technical Risk
  - check the proposed design configuration against technical requirements, customer needs, system requirements and clarify design requirements
  - independently evaluated by experts based on feasibility evidence provided by the developer

SCRAM FUTURE

- Version 2 of the SCRAM Model and Methodology (released June 2013) incorporated additional technical process guidance, lessons learned and systemic issues identified from earlier SCRAM reviews
- The DMO SCRAM capability is being expanded in the DMO through the development and delivery of SCRAM Introduction and Assessor Training
- Consideration is being given to using SCRAM in a pre-emptive mode to prevent the repetition of common systemic issues rather than rely solely on the current diagnostic SCRAM approach

Thank You

Any questions?