Practical Software and Systems Measurement

Project Managers Guide to Systems Engineering Measurement for Project Success

June 16, 2017

Greg Niemann
gregory.niemann@lmco.com

25th anniversary annual INCOSE international symposium
Seattle, WA
July 13 - 16, 2015

Project Managers Guide to Systems Engineering Measurement for Project Success

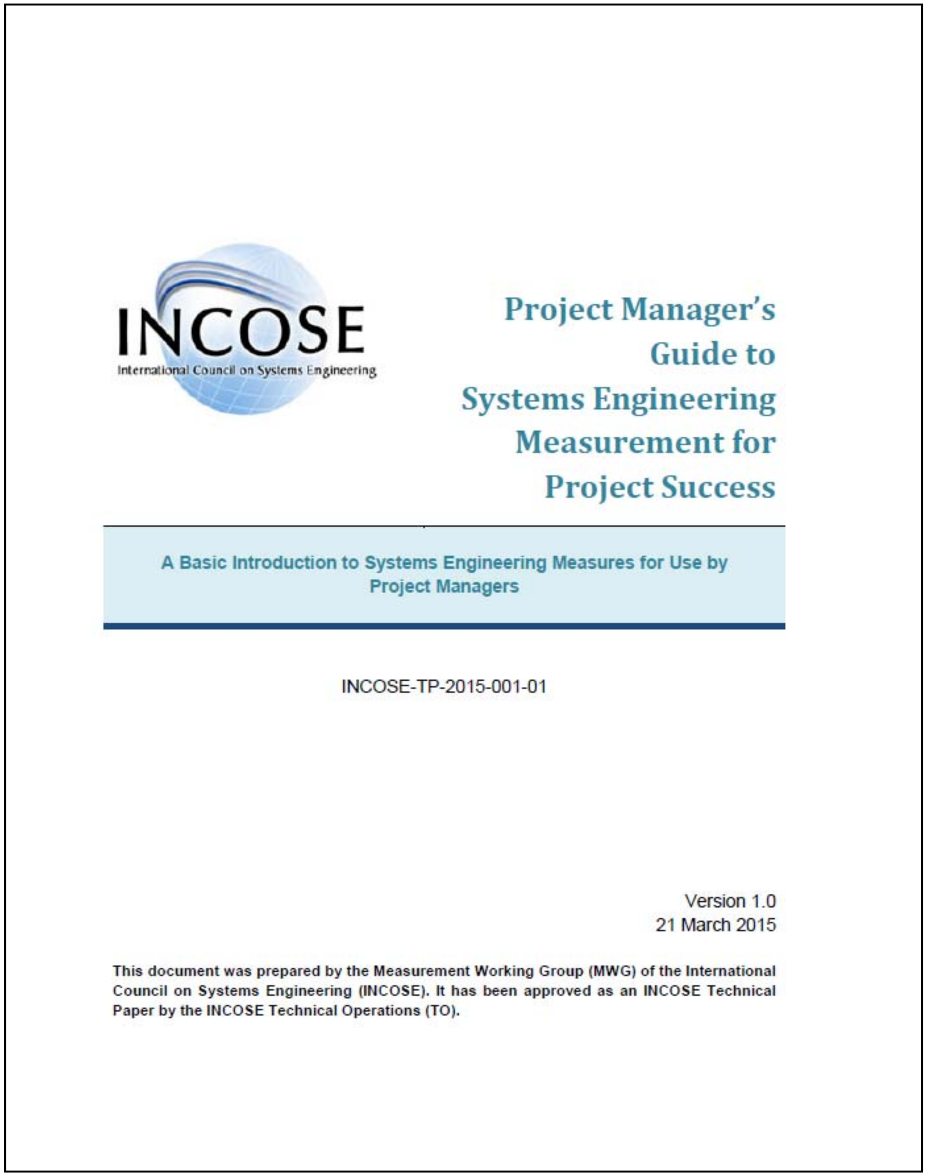
Dr. Ron Carson, Fellow, ESEP

Paul Frenz, MWG Chair, CSEP

INCOSE Measurement Working Group

INCOSE IS2015

Objective



INCOSE
International Council on Systems Engineering

Project Manager's
Guide to
Systems Engineering
Measurement for
Project Success

A Basic Introduction to Systems Engineering Measures for Use by Project Managers

INCOSE-TP-2015-001-01

Version 1.0
21 March 2015

This document was prepared by the Measurement Working Group (MWG) of the International Council on Systems Engineering (INCOSE). It has been approved as an INCOSE Technical Paper by the INCOSE Technical Operations (TO).

- Provide a “beginners” guide to project managers regarding how SE measurement can help them manage projects
  - Target audiences
    - PMs of smaller projects that may have no SEs
    - SE leaders on larger projects looking for high-leverage measures
- Publish as an INCOSE MWG product

Approach

- Keep it short
- Use PM language whenever possible
- Keep SE measurement in the context of concerns of project managers
  - Technical risk as it affects cost/schedule
  - Technical risk as it affects the ability to finish
- Enable PMs to find what they need
- Identify and explain the important few measures
- Include references to other INCOSE and industry work

Outline

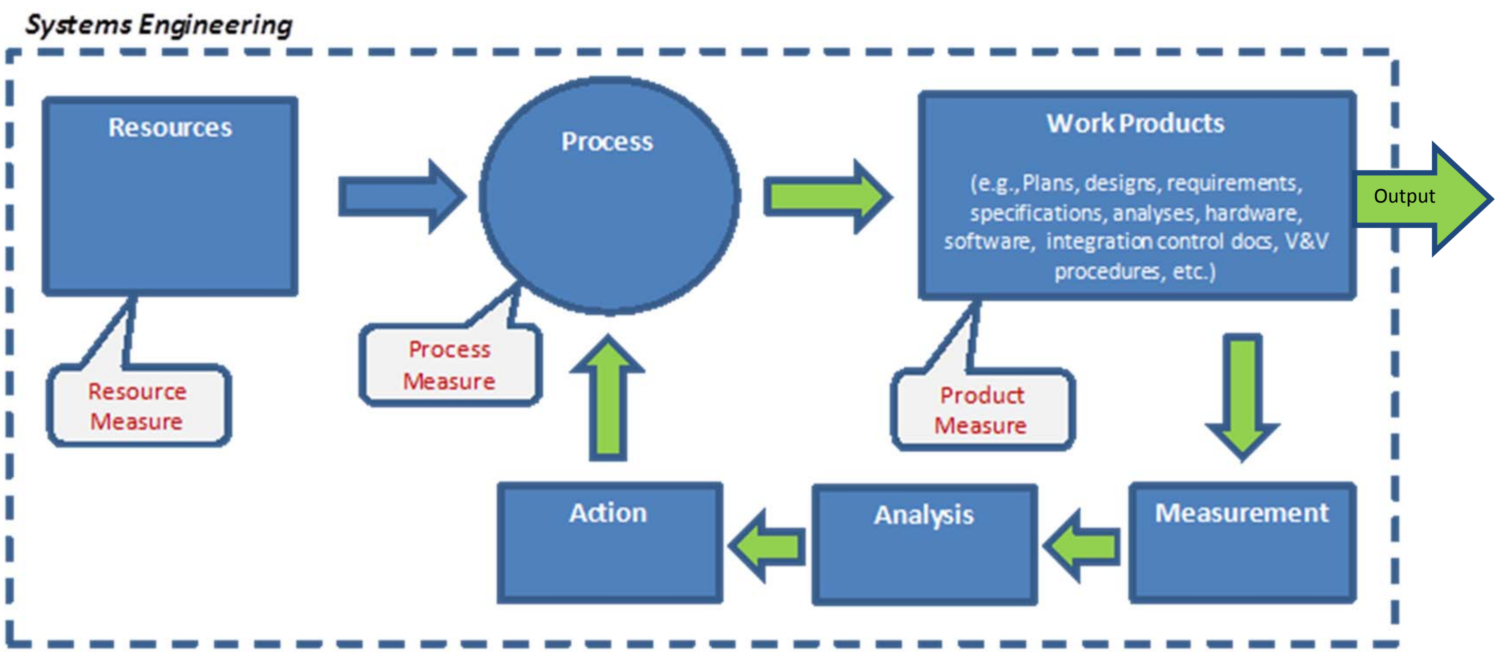
- Chapter 1: Introduction
- Chapter 2: Measurement in Systems Engineering
- Chapter 3: Quick Start Guide
- Chapter 4: A Look at Technical Debt
- Chapter 5: Project Technical Measures Throughout the Lifecycle
- Chapter 6: Case Study

How is this different from other Guides?

- The current work is not comprehensive from an SE perspective
  - It seeks to identify high-leverage measures of value to project managers who might not otherwise ever consider SE measures.
- New guide references existing work and standards
  - SE Measurement Primer
    - Different target audience
    - New guide uses first chapter
  - SE Leading indicators
    - Reference is made to requirements volatility and defects
  - Technical Measurement Guide
    - Reference to TPMs, MOE, MOP
  - SE Handbook - reference to TPMs only
  - ISO15939 – identified as an industry reference on measurement

Chapter 2 – SE Measurement

- Summarizes key concepts from SE measurement primer: product measures, closed-loop feedback control, process and resource measures



Chapter 3 – Quick Start Guide

Identifies questions and answers of interest to PMs

- Why should I measure?
  - What gets measured gets done. It's that simple.
- What should I measure?
  - You should measure what is critical to your program to be successful. See Chapter 5 for guidance on measurement selection.
- How do I measure with minimum budget to achieve the most?
  - You want to select the "critical few" measures that provide the insight into areas of highest risk to your specific project.
- What do I do when data are disputed by members of the project team?
  - Let's go to the expert, Dr. Edward Deming: "In God we Trust; all others bring data." In other words, trust the data *first.* Then ask, "Why are the data in question?"

Chapter 3 – Quick Start Guide

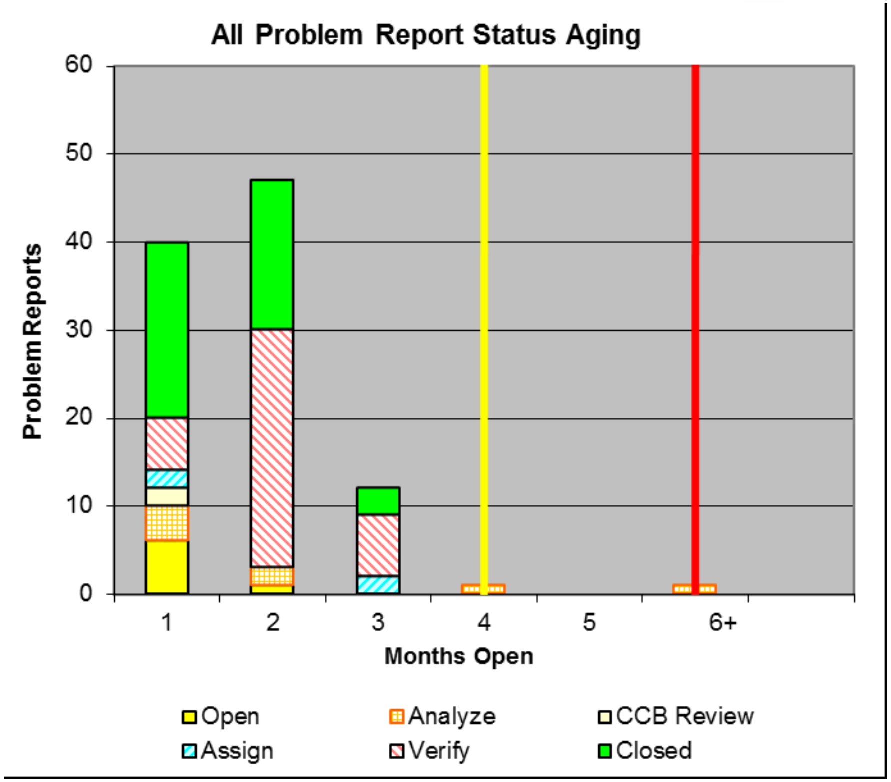
- Identifies and explains different project characteristics that may lead to selecting different measures in Chapter 5

Project Considerations	Applicable Factors					
Measurement Category	Technical Quality	Size	Complexity	Stability	Schedule	
Phase	Conceive and Define	Architect and Design	Implement and Integrate	Verify	Validate	Operate and Support
Development Strategy	Waterfall	Agile / Spiral	Increments	Acquirer-Funded	Supplier-Funded	
Tools and Databases	Manual or Spreadsheet	Requirements Management	Static Model-Based SE	Simulation-Based SE		
Product Domain	Software-Intensive	Hardware-Intensive	Complex	Regulatory Environment	Commercial	Government
Staff capability	Primarily Novice	Intermediate	Primarily Experienced	Experienced, new domain		

Chapter 4 – Technical Debt

- Guides PM in identifying technical risks and their consequences using their language

- Identifies some example measures and how they can be used to help manage technical debt



Chapter 4 – Technical Debt

- **Technical Debt** is the promise to complete a technical shortcoming in the future while declaring it "complete enough" today.

- ***What is the Technical Debt Trap?***
  - Similar to personal debt, the program is explicitly, or more commonly, implicitly **deferring a technical challenge or risk to the future** because you don't want, or cannot spend the time and/or money, to successfully solve a technical challenge before declaring the task complete.

Chapter 4 – Technical Debt

How is Technical Debt Incurred?

- Fundamentally, there are three ways in which Technical Debt is incurred.
  1. Omission: Tasks unaccounted for within schedule and/or budget;
  2. Wishful Thinking: Tasks declared completed but not really complete; and
  3. Undetected Rework: Tasks believed to be completed but done incorrectly.

Chapter 4 – Technical Debt

How to Avoid and Measure Technical Debt!

- To avoid Technical Debt, you will need to apply three methods:
  1. Account for unscheduled tasks
  2. Establish measures to provide early warning
  3. Providing cost-account manager training for properly "earning value" (schedule and cost)

Chapter 5 – SE Measures

INCOSE

- 10 measures identified with
  - Link from Quick Start "Project Considerations"
  - Explanation (what, how, why/benefit)
  - Example Excel spreadsheet or other example
  - Reference (footnote) to the literature

Technical Performance	Measures

Project Considerations	Applicable Factors					
Measurement Category	Technical Quality	Size	Complexity	Stability	Schedule	
Phase	Conceive and Define	Architect and Design	Implement and Integrate	Verify	Validate	Operate and Support
Development Strategy	Waterfall	Agile / Spiral	Increments	Acquirer-Funded	Supplier-Funded	
Tools and Databases	Manual or Spreadsheet	Requirements Management	Static Model-Based SE	Simulation-Based SE		
Product Domain	Software-Intensive	Hardware-Intensive	Complex	Regulatory Environment	Commercial	Government
Staff capability	Primarily Novice	Intermediate	Primarily Experienced	Experienced, new domain		

Technical performance measures can be applied for selected technical parameters to ensure adequate progress is being achieved. Time-based plots of estimated or demonstrated performance are compared with required values (minimum or maximum) to help manage the risk. This is a quantitative form of a risk mitigation plan. A plan line with decision bounds should be established early in the program with required progress in achieving the threshold value (e.g., "not to exceed"). Failure to achieve the required progress converts the *risk* to an *issue* and may require a design change to ensure technical compliance.

INCOSE

INCOSE Systems Engineering Handbook v. 3.2.2
INCOSE-TP-2003-002-03.2.2
October 2011

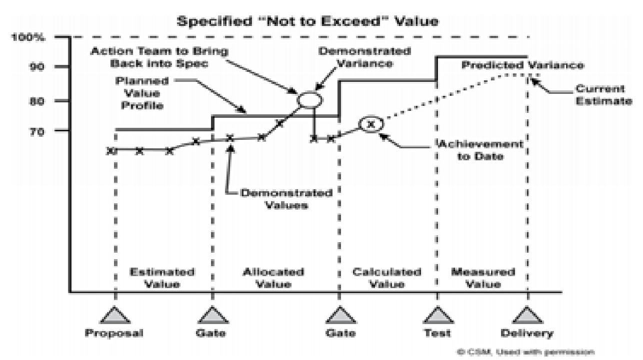


Figure 5-18 TPM Monitoring

- Track the progress of selected technical parameters compared with required values to ensure adequate progress is being achieved[1]

[1] INCOSE SE Handbook v3.2.2, http://www.incose.org/ProductsPubs/products/sehandbook.aspx (accessed June 2014) and INCOSE Technical Measure Guide (2005) http://www.incose.org/ProductsPubs/products/techmeasurementguide.aspx (accessed June 2014)

Specific Measures

1. Schedule – Late starts, late completions
2. Problem Report Aging, Peer Reviews held
3. Technical uncertainty reduction
4. Scope change (requirements volatility)
5. Technology Readiness/Maturity level
6. Solution satisfies requirements (% compliant)
7. Technical Performance
8. Counts/stability of system elements
9. Reliability, Maintainability, Availability
10. Defect Containment

1. Schedule alert – caution about starting with incomplete data
2. Delay in closing technical reviews and issues
3. Identifies decision-making threshold (like TPM)
4. Identifies requirements changes
5. Identifies technical risk in maturity of solution
6. Identifies compliance risk of solution
7. TPMs
8. Helps track changes in overall architecture
9. Helps predict O&M problems
10. Identifies effectiveness of problem identification and resolution

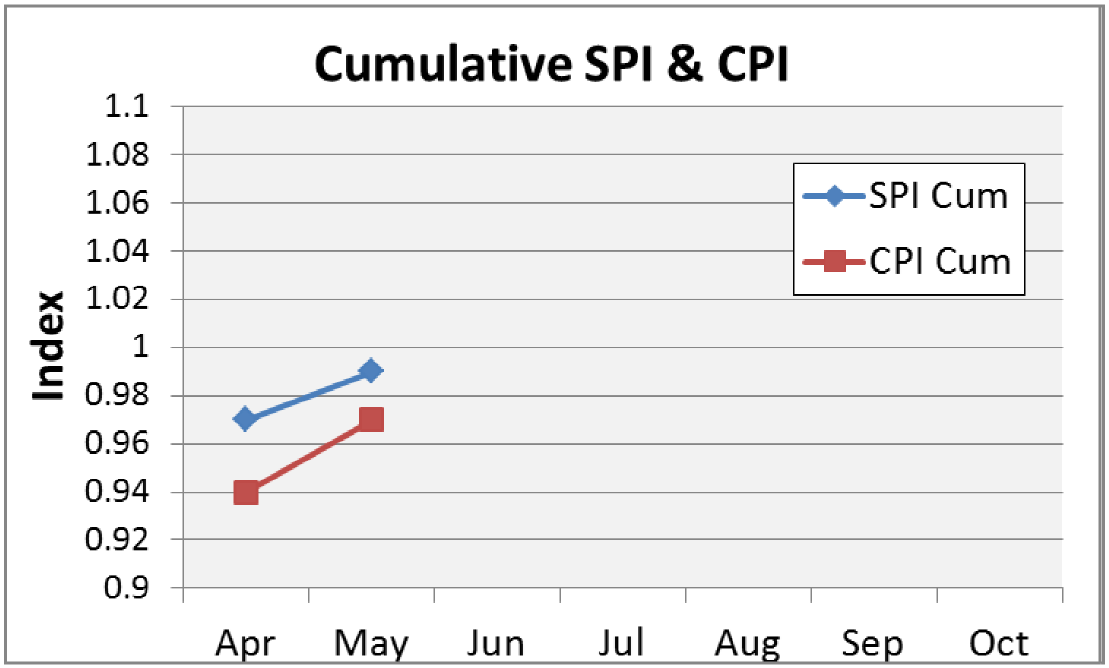
Chapter 6 – Case Study

- Example case study for a short project
  - Covers both programmatic and technical measures
  - Walks through measurement selection process
  - Demonstrates use and interpretation of various measures with corrective actions
  - Rationale and use of weekly and monthly measures
  - Employs appropriate measures at different stages of the life cycle

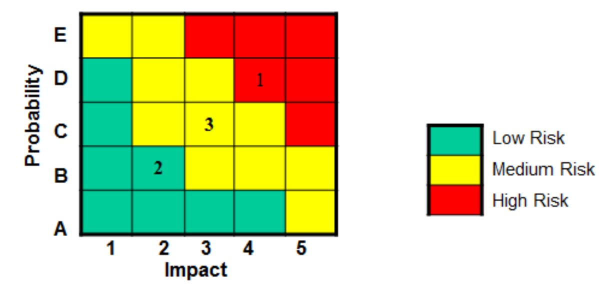
Measures Selection

- Programmatic Measures
  - Schedule Performance Index (SPI)
  - Cost Performance Index (CPI)
  - Risk matrix

Programmatic Measures



Program Risk Assessment



Risk #1 Long lead times for components will delay integration and test ▪Due to tight schedule, long lead parts could drive schedule and cost	**Risk #2 New Software drivers will be required for new components** ▪ Schedule and budget only allow for components with compatible drivers	**Risk #3 Reduced size widget will not meet power dissipation threshold** ▪ Reducing the size of widget package will create power dissipation capability
▪Mitigate through allowing component selection if commercially available	Mitigate through software review and approval before any new components can be selected	Mitigate through monitoring parameter uncertainty measure

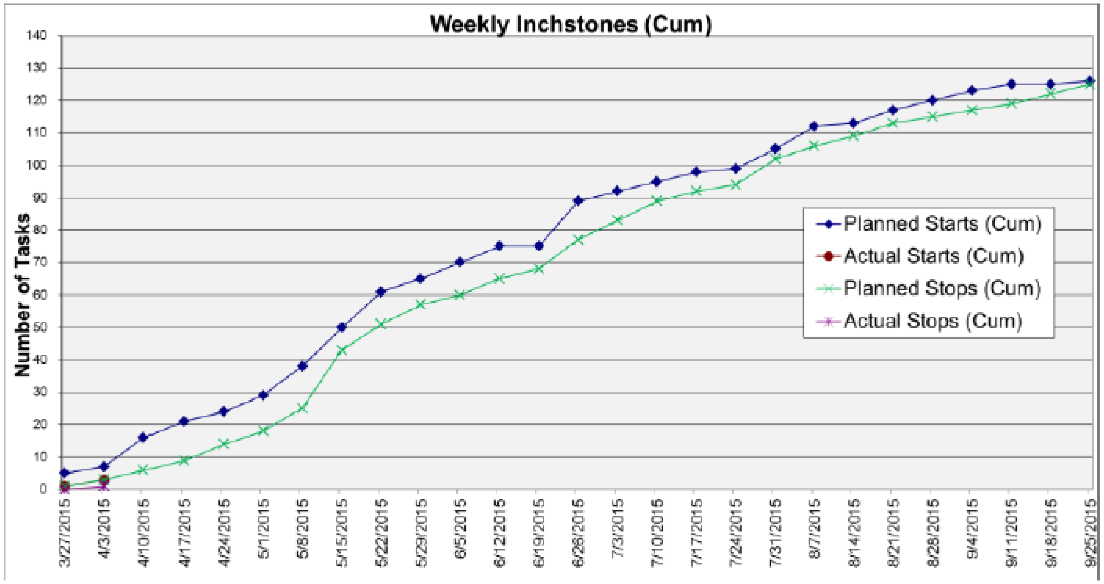
Typical programmatic measures used to manage a program used to manage schedule, cost, and risk.

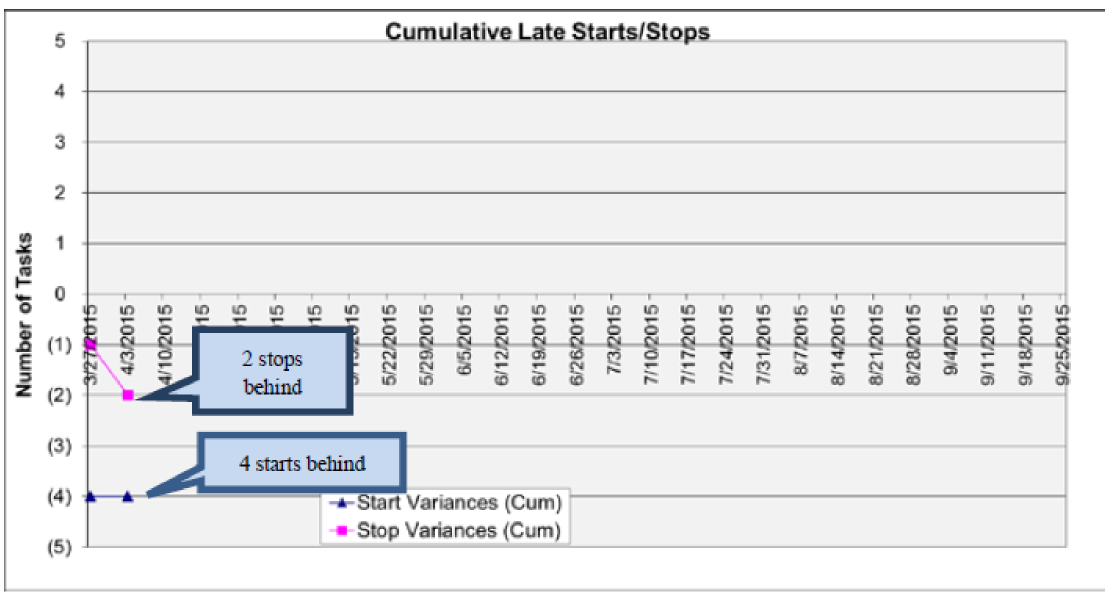
Measures Selection

- Systems Measures
  - "Inchstones" IMS Measure (Weekly)
  - Requirements Volatility (Monthly)
  - Requirements Verification Percentage (Weekly during Verification phase)
  - TPMs
    - Size (Monthly)
    - Power Dissipation Uncertainty (Monthly)

Adding finer granular schedule, volatility, verification progress, and key technical performance measures

Systems Measures

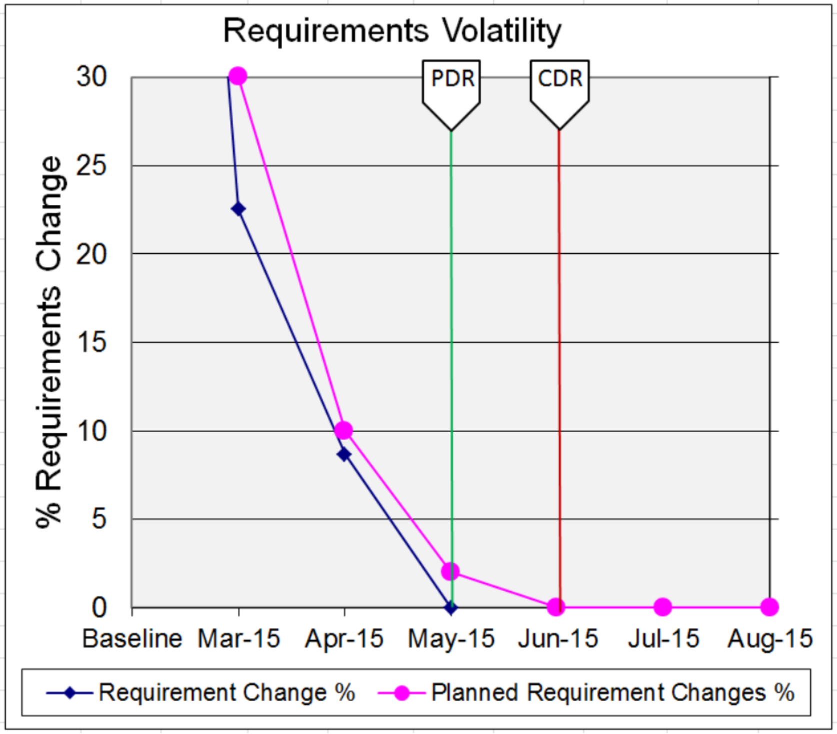


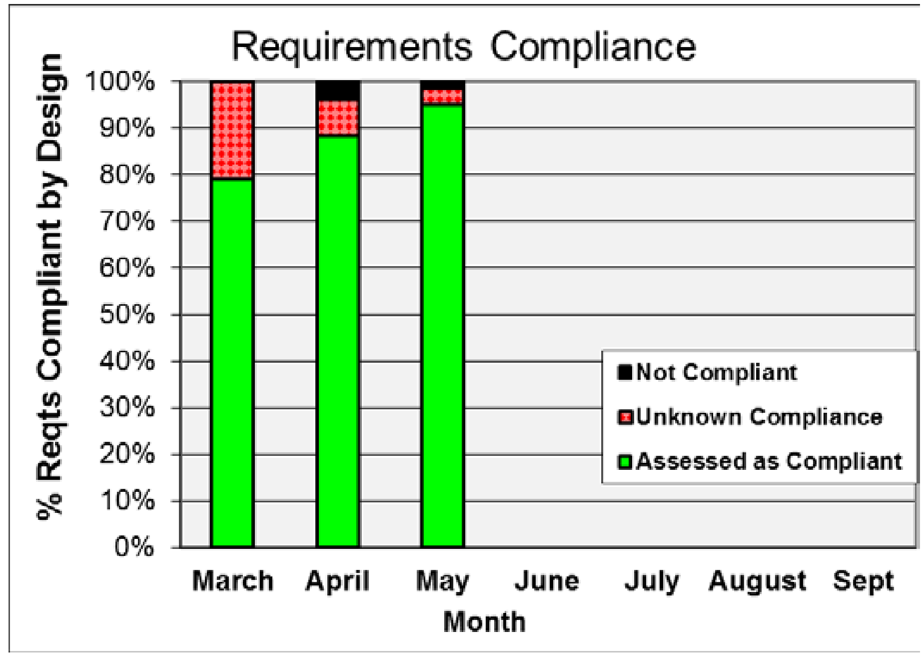


Weekly monitoring late starts and stops – Allows early intervention.

Best leading indicator of a late stop is a late start!

Systems Measures

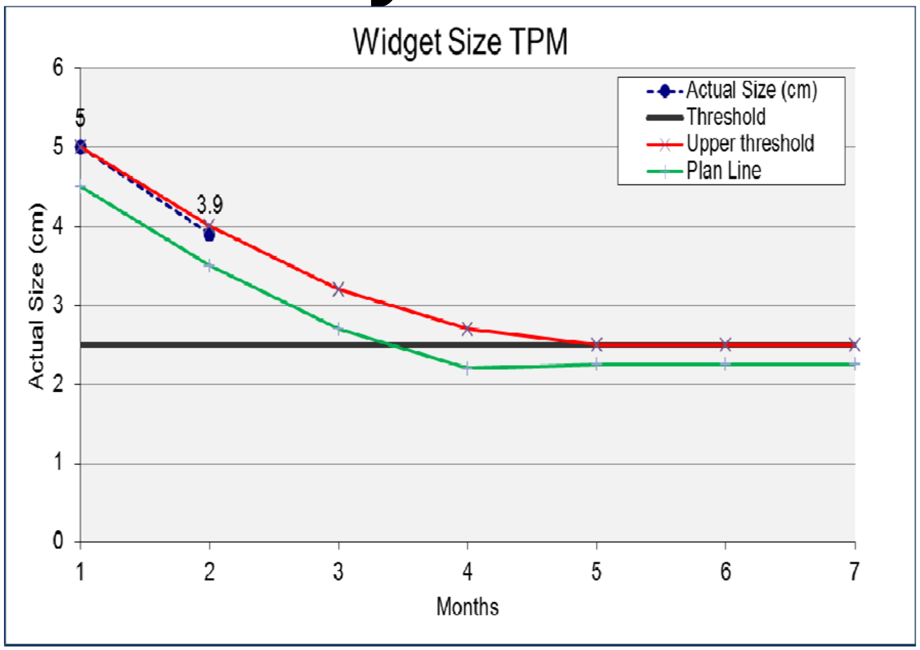


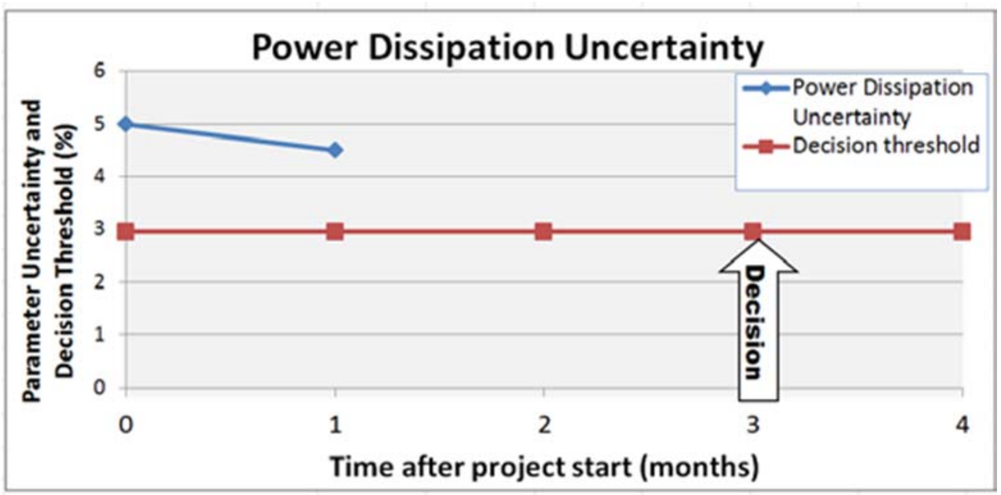


Unplanned requirement volatility can create havoc with schedule and budget.

Measuring progress with requirements compliance can be used to avoid technical debt.

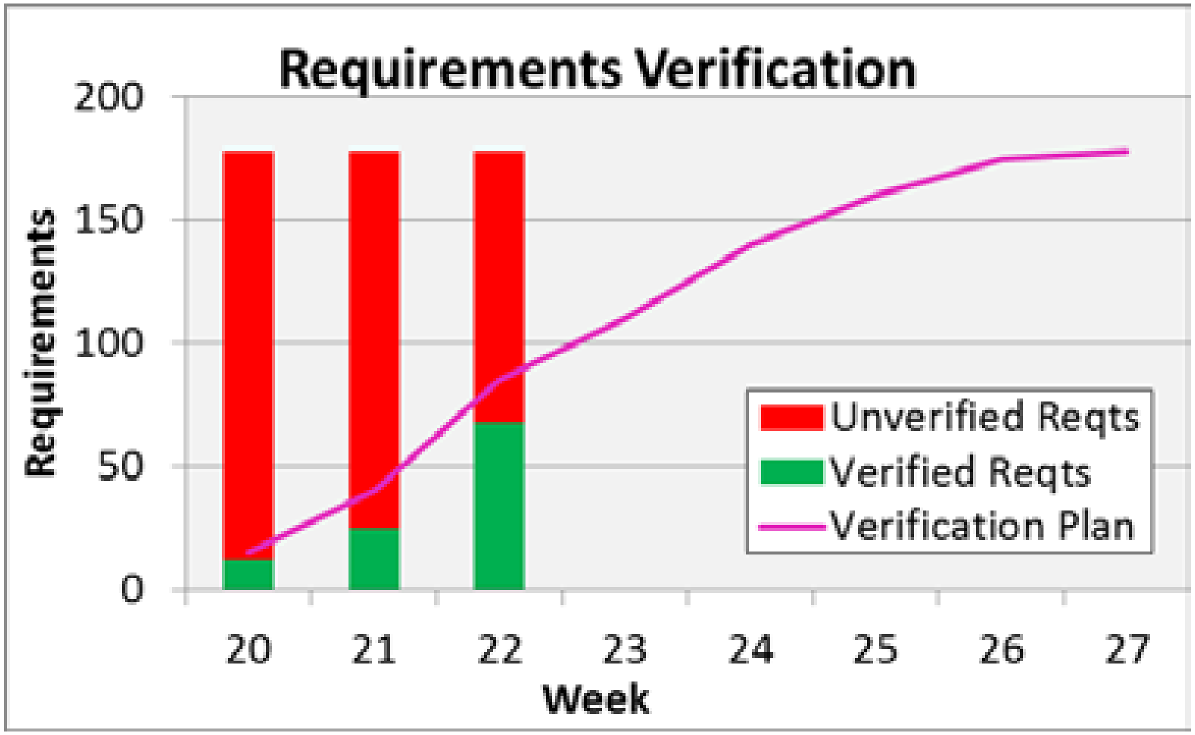
Systems Measures





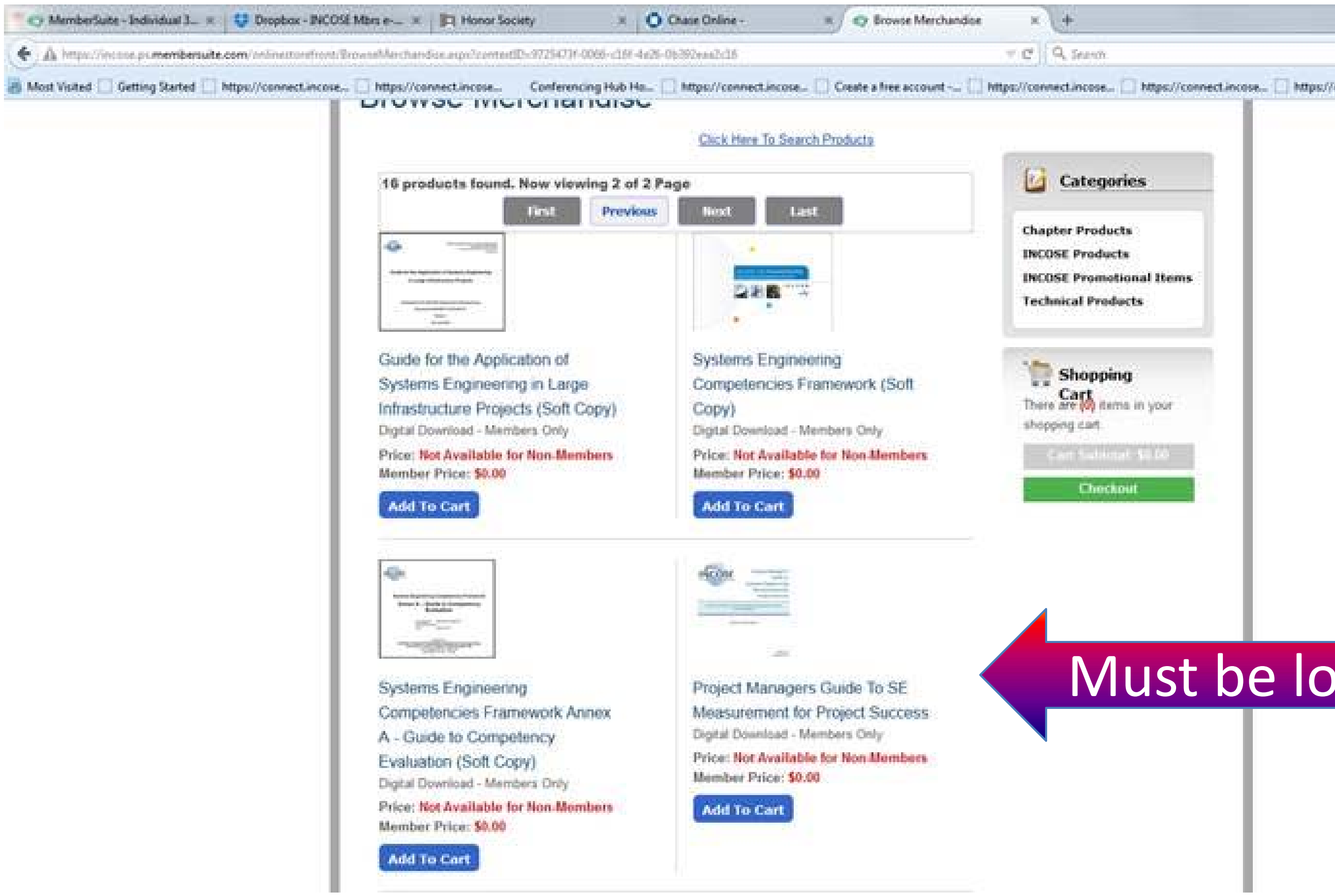
Measuring progress with technical challenges keeps focus on critical design decisions.

Systems Measures



Frequently compressed, measuring requirements verification progress can be used to manage final push to completion of development phase

Available at INCOSE.ORG within Store



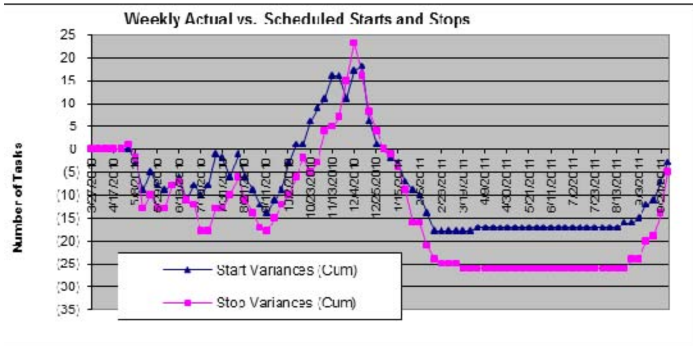
Must be logged in

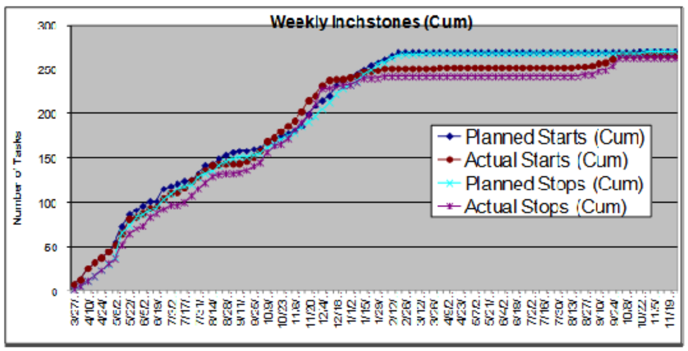
Project Managers Guide to Systems Engineering Measurement for Project Success

Questions?

1. Schedule – Late starts, late completions

- Delayed starts are leading indicators for delayed finishes. However, be wary of starting tasks when necessary data is not available, is incomplete, or is likely to change because rework of dependent input is likely; “don’t be a slave to schedule”. For effective feedback control, the measurement delay should be no greater than the measurement frequency. In this case of weekly measurement, the data should be available before the next week begins.
- For all schedule-related measures it is important to find the root cause of what is late so that the program critical path is not jeopardized and rework is not incurred by immature or incomplete work.”



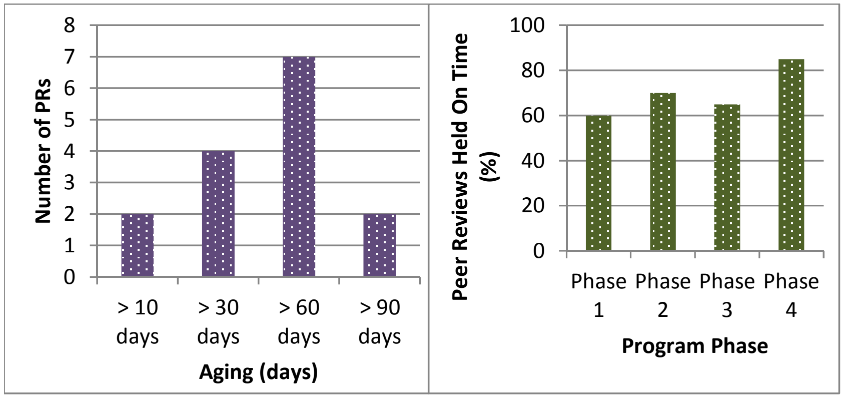


Upper Left: cumulative tracking of “planned” and “actual” starts and stops, by week, provides overall project schedule status.
Lower Left: Weekly difference of planned vs. actual starts and stops provides immediate visibility of schedule compliance.

2. Problem Report Aging, Peer Reviews held

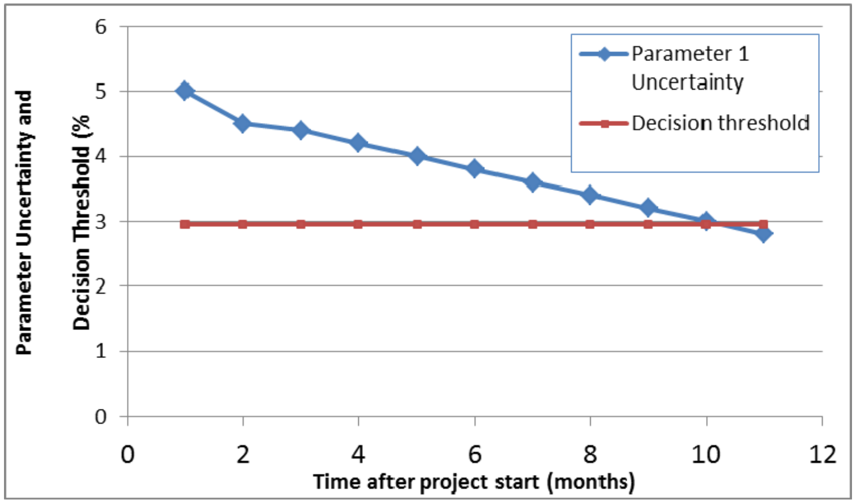
- Delayed resolution of problems or review of technical information may accumulate more technical debt and may indicate that critical decisions are being delayed, which jeopardizes the schedule.
- Histograms also work well for these types of counting measures.

- Below left, PR Aging is depicted with a histogram showing the number of Problem Reports in each category of delay. Below right, the histogram indicates % of peer reviews held on time in each program phase.



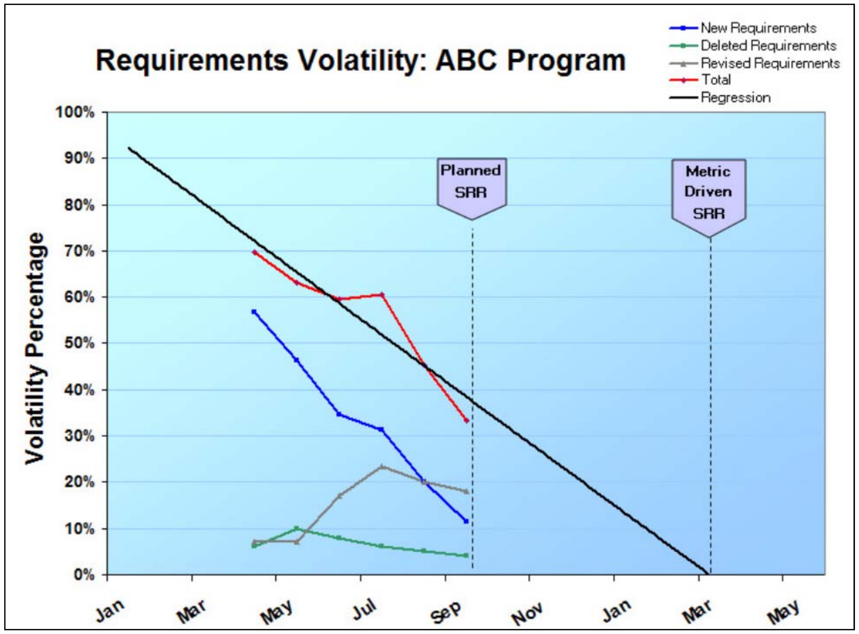
3. Technical uncertainty reduction

- Unresolved uncertainty carries technical debt into the decision-making process. The goal is not to eliminate the uncertainty, but to reduce it to a level at which a decision can be made with acceptable risk. This applies to individual technical parameters as well as to the results of technical reviews.
- Trend lines similar to technical performance measures make the uncertainty visible compared to the needed value. In the example, the uncertainty of Parameter 1 must be reduced below the decision threshold prior to making the decision.



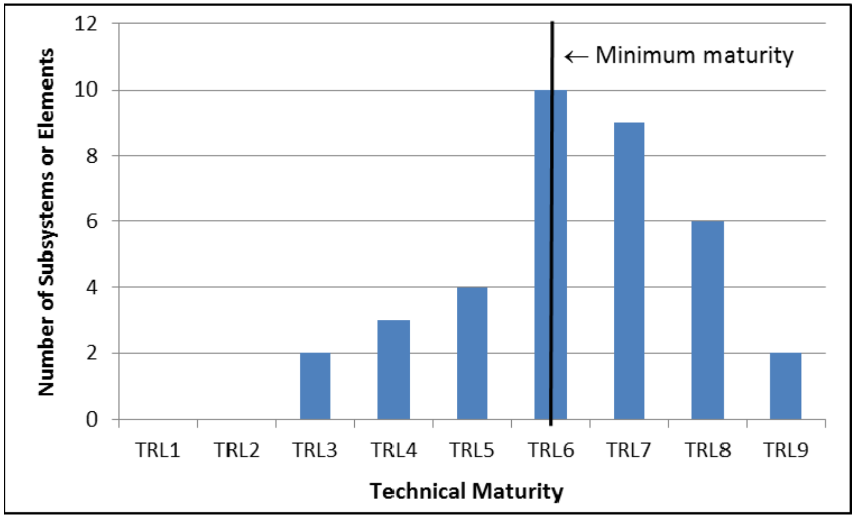
4. Scope change (requirements volatility)

- It is not uncommon to have some requirements changes during a project. Project managers need to be aware of additions or modifications to requirements that (a) affect contractual agreements or (b) change the required effort or resources necessary to meet project obligations (cost, schedule, people, laboratories).
- Trend analyses are useful for tracking scope changes. Action thresholds for change may decrease over time as the design matures and the impact of requirements changes becomes greater. Prior to a system requirements review (SRR) the volatility is expected to be high, but must settle down ahead of the SRR. Failing to move the SRR will incur technical debt and likely rework. Once the critical design review (CDR) takes place, most subsequent changes will increase project costs and lengthen schedules.



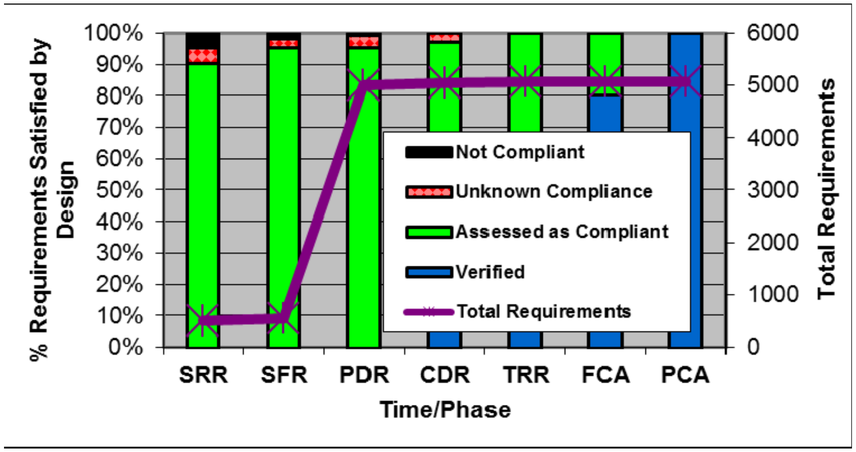
5. Technology Readiness/Maturity level

- Technical maturity (or technology readiness) level identifies the technical debt inherent in the elements of the solution based on the development status (e.g., in-production, prototype, variation on a product family). Most projects require at least TRL 6 (prototype) before incorporating an item in a development project.
- A quick way to evaluate the state of the program is to create a histogram showing how many items are in a given maturity category so that appropriate management oversight can be provided to manage the technical risk. In the example, management attention should be focused on the elements with TRL < 7 and on developing contingency plans in case any element does not achieve full maturity according to a development plan.



6. Solution satisfies requirements (% compliant)

- The key technical progress measure for development programs is an evaluation of the degree to which the design is satisfying the requirements. Any non-compliance is an issue that must be corrected and indicates a need for rework. Unknown compliance is risk of a future discovery of non-compliance and is therefore a form of technical debt based on uncertainty.
- This measure can be represented as a time-dependent bar chart showing progress of technical compliance until all requirements are verified.



7. Technical Performance

- Technical performance measures can be applied for selected technical parameters to ensure adequate progress is being achieved. Time-based plots of estimated or demonstrated performance are compared with required values (minimum or maximum) to help manage the risk. This is a quantitative form of a risk mitigation plan. A plan line with decision bounds should be established early in the program with required progress in achieving the threshold value (e.g., “not to exceed”). Failure to achieve the required progress converts the *risk* to an *issue* and may require a design change to ensure technical compliance.

INCOSE Systems Engineering Handbook v. 3.2.2
INCOSE-TP-2003-002-03.2.2
October 2011

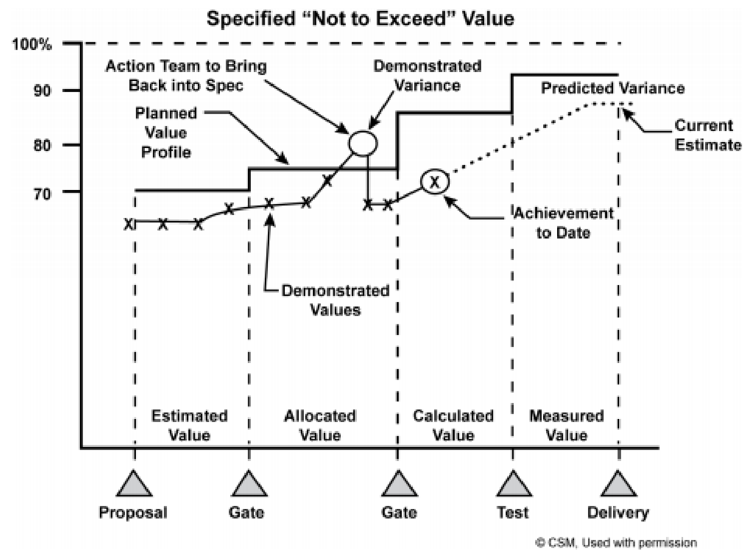
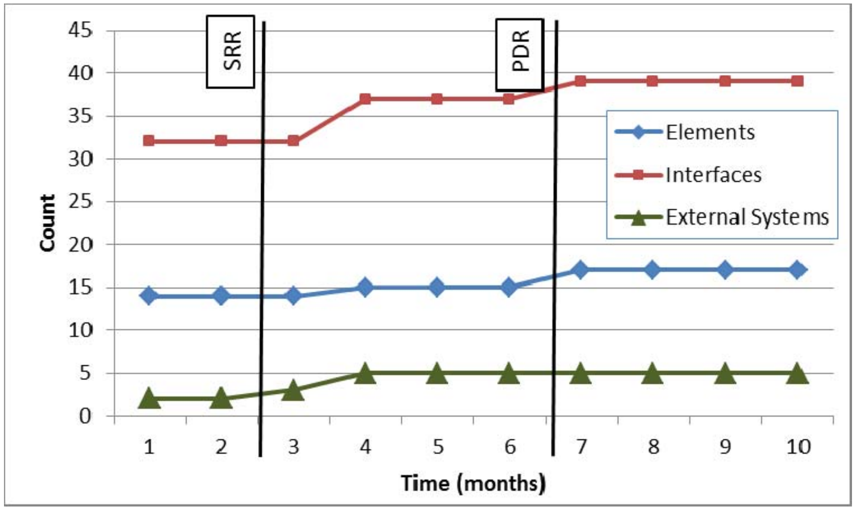


Figure 5-18 TPM Monitoring

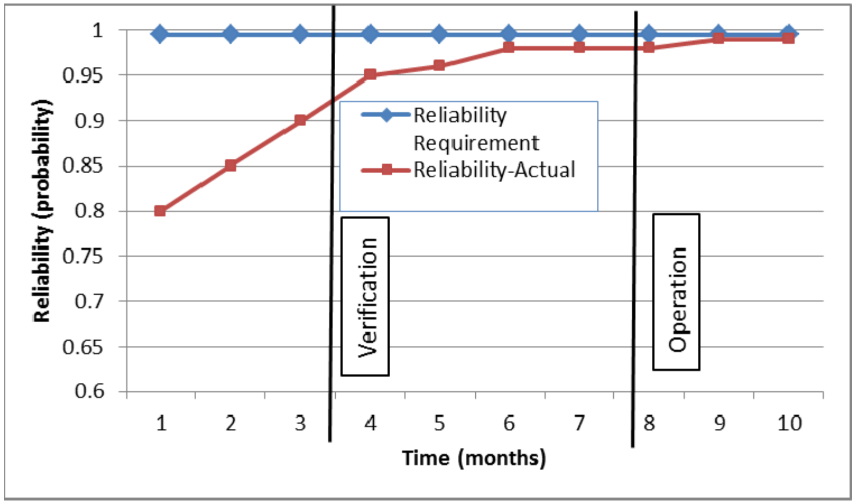
8. Counts/stability of system elements

- Database tools enable managers to more easily count elements of the solution, whether requirements, interfaces, or solution elements (subsystems, boxes, wires, etc.). While the absolute numbers may not be critical, sudden growth can indicate scope change or increased complexity and development risk.
- Visibility of these changes is provided by simple charts of counts vs. time. Project managers should monitor these measures for unexpected changes while the design should be stable. For example, “External Systems” should be stable at Systems Requirements Review, and “Elements” and “Interfaces” should be stable at Preliminary Design Review. In the graph none of the three conditions is satisfied so that the project manager should investigate root causes and take corrective action to avoid additional technical debt from the changing design. Increasing complexity based on increasing element and interface counts may also lead to more risk during the integration phase after the critical design review.



9. Reliability, Maintainability, Availability

- Once development is nearly complete the project can begin to accumulate data on operational performance for reliability and system availability. The Verification phase provides a “first look” at these system performance measures that have significant consequences during operations and support phase.
- A time-dependent line chart can be used to compare current performance vs. operational need or requirement. The need for design or other changes can become apparent if deficiencies are other than initial “growing pains”. In the example below the implemented design is failing to meet its reliability requirement even as the system moves into operation, and root cause investigation may be required to identify and correct the deficiency.



10. Defect Containment

- Technical debt in the form of rework accumulates when errors in technical data are not identified and corrected before the data is used by other groups (e.g., Requirements for Design and Verification, Design for Build and Verification, Trade-off Analyses for Design). The longer the delay in discovering the error, the larger the cost of the rework.
- Histograms of defect containment are a valid way to display this information (defects introduced by phase vs. phase in which they are discovered and corrected).
- This measure can be used within a project for additional spirals, increments, or agile scrums so that more rigor is applied in finding defects prior to propagation. The measure is also useful for organizational and system process improvement so that error propagation can be reduced on successive projects.