Software and Cyber Solutions Symposium 2018

How to Develop And Use a Project Dashboard

Robert Ferguson

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0351

Who Should Be Here

Acquisition Program Managers

Development Project Managers

Associated Technical Staff

How do you know?

- The health of your project?
- That you are going to deliver on time?
- That you will not have to request more money?
- What alligators are after you?

What can you do about it?

Assumptions Strike Again

An acquisition or product development plan makes many assumptions.

- Requirements were satisfactorily determined by the start of the project
- Our ability to mature a new technology
- Correctness of the estimate
- Ability of the development team to execute at the planned rate
- Availability of personnel and other technical resources
- Effectiveness of subcontractors
- Availability of GFE and other external government resources
- … and many more

What happens if an assumption was incorrect?

How long does it take to discover the problem?

How does the problem affect the plan?

Why a Dashboard

Many interrelated items must be monitored.

The specific items to monitor changes during project.

The presentation of the information must be done in a way that facilitates comprehension.

Communication is all important in the dashboard.

- Shared understanding of progress and risk.
- Shared accountability for cost, schedule and success.

Graphs must tell the story.

Building trust (keeping our commitments and expecting others to keep theirs).

Dashboard as a Communications Tool

Lots of different people have a stake in your project.

It is important to tell them how you are doing so they will keep their commitments to you.

Purpose:

- Show the commitments you will keep
- Predict the future of the project
- Engage others to keep their commitments.
- Highlight the near-term and critical risks.
- Engage others in planning to mitigate problems.

Customers
Project Team
Executives
Other external participants

Agenda

Building a Project Dashboard
- Structure by decision type
- Describe specific examples of measures and charts

Schedule, Cost and Resource

Progress and Change Management

Product Quality and Quality Management

Effective Process

Risk

Wrap up

Extra: Developing New Measures and Indicators

Extra: Prioritizing Defects for Sustainment

Project Management Needs Decision Support

Project Management involves frequent decisions in several categories.

- Schedule, Resources, Cost
- Scope, Progress, Change Management
- Product Quality & Quality Management
- Risk and Processes

- These categories are interrelated: changes in one can often affect performance in another.

Measure to Make Better Decisions

Kahneman and Tversky (Nobel Prize Winners) ran experiments to prove:

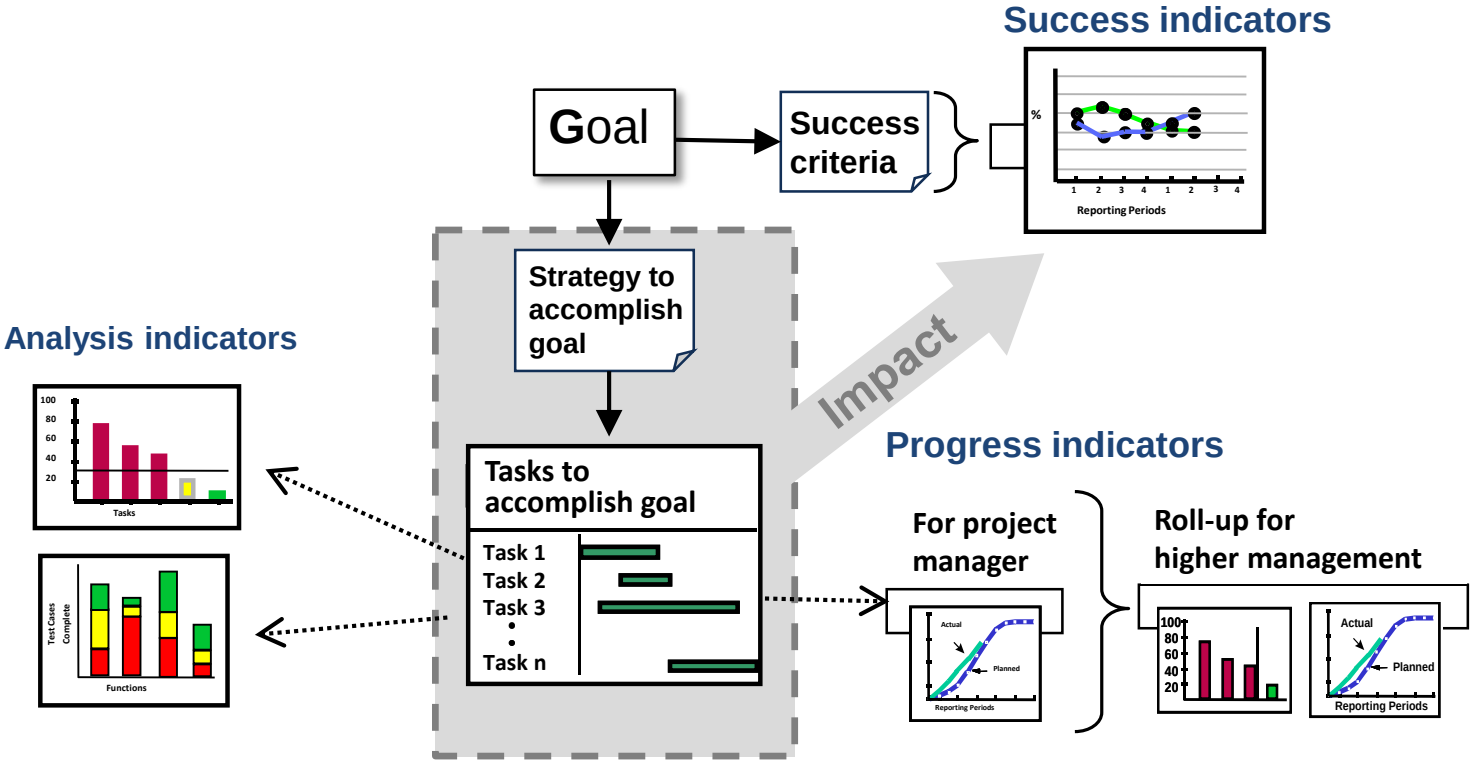
- “ From a few simple measures and a simple algorithm, we can consistently make better decisions than experts”.
- “Even when the experts told us what to measure, if they are not using the algorithm, they are not as good as we can be.” (Thinking Fast and Slow, Kahneman)

Getting data for measures does not have to be difficult.

- “You know more than you think you do.
- You need less information than you think you need.
- The information you need is easier to obtain than you think it is.”
  - “How to Measure Anything”, Douglas Hubbard

What Kinds of Data Will We Need?

The SEI's Goal-Question-Indicator-Method for Projects



Success, Progress and Analysis Indicators

Progress & Efficiency Indicators Directly Affect Plans:

- Compare planned values to actual results
  - Cost, schedule and deliverables completed.
- Provide estimates for completion or performance

Warning & Analysis Indicators Highlight Potential Problem Areas:

- Cumulative Risk
- Quality
- Process Adherence
- Change Responsiveness
- Staffing Level, Turnover
- Relationship & Contract Issues

Success Indicators Reflect Product Fitness for Use and Adoption

Dashboard Quadrant Definition and Content

Goals and Objectives:

- Program goals and objectives for the quadrant

Questions

- Things to check during a project review meeting

Indicators and Measures

- Some suggested charts and measures to collect

Decisions Informed

What types of decisions do the indicators on the dashboard inform

Dashboard construction is developed using SEI GQIM method

- *Goal-Question-Indicator-Measure*
- *This methodology supports ISO-9000, CMMI and other quality standards*

Dashboard Stoplights and Detail

Stoplights:

- Green: operating according to plan and expectations
- Yellow: operating within contingency or significant risk of change
- Red: not operating according to plan. Change requested.
- Whose decision is required for change?

Detailed Charts

- Supports plan. Demonstrates progress toward goals.
- Supports request for change. Demonstrates (potential gap).

Ready for the dashboard elements?

Dashboard Top-Level Example

Schedule, Cost & Resources

- G EVMS (program)
- Tasks Completion to Schedule
  - R Late Starts/Completions
  - G Dependencies
  - G Deferred / Accelerated tasks
- G Development Status (HW, SW design, implementation, unit//Integ test)
- N Test Status (SIL, GND, FLT)
- Staffing
  - Y Contractor
  - N Gov't

Progress & Change Management

- Requirements Trends
  - R Sizing, Trend
  - G Risk
- Development Trends
  - R Hardware
  - Y Software
  - N SLOC
    - TO Artifacts
- Test Trends
  - G TOIs
  - R Trends – SIL, Ground, Flight
- R CDRL Completion
- G Engineering Changes (# by source)

Process & Risk

- G CDRL Comments
- R Action Item Aging
- G Req Traceability
- N *Req Volatility (future)*
- N *SLOC (future)*
- N *TO Comments (future)*
- N *HW TOIs (future)*
- N *QA Reports(future)*
- R Action Items (Overdue)
- Y Problem/Defect Reports

Quality

- G Requirements Approved
- G TRL's
- G *Problem/Defect Reports: (future)*
  - *SW Defects by Phase & Severity*
  - *SW Defects – by CSCI & Severity*
- N TPMs*
  - N report all exceptions
- G Verification Dependencies (by req/test point)
- G Verification Distribution (by CSCI/CSU, HWCI)

- G Within plan
- Y Within contingency
- R Change required
- N Not used this period

Details by Quadrant and Stoplight

Quadrants High Level Goals and Objectives

Schedule, Cost, & Resources

- Plan predictability
- Corrective action and Replanning

Progress and Change Management

- Deliver the product including all elements of scope
- Verify completion forecasts based on productivity
- Prepare estimates and plans for change management

Product Quality and Quality Management

- Determine fitness for use
- Check effectiveness of quality activities

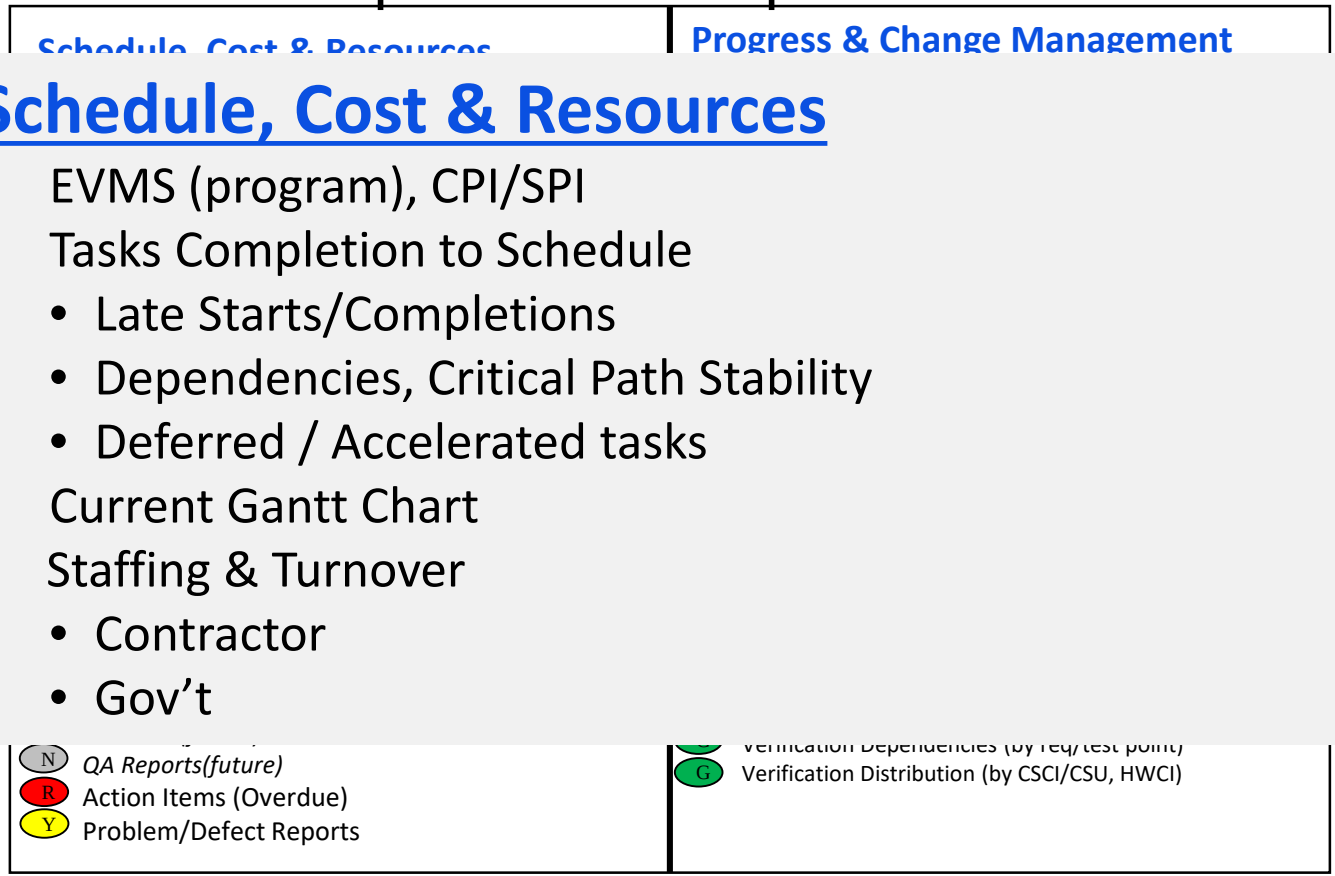
Effective Process and Risk

- Watch for process breakdown
- Assess project assumptions

SSD- Dashboard

Schedule, Resources and Cost

Dashboard Top-Level Example



G Within plan

Y Within contingency

R Change required

N Not used this period

Details by Quadrant and Stoplight

--

Schedule, Resources and Cost

Goals

- Project delivers on its forecasted date at the forecasted cost.
- Forecasts based on available data are recalculated on a regular basis.

Questions

- Has the project been predictable in recent milestone performance?
- Are forecasts regularly updated (every program review)?
- Does the current forecast predict a problem in project execution?
- Have resources been available and used as planned?

Indicators and Measures

- Percentage of predicted tasks completed within the week (or month)
- Differences between planned resources and resources used
- Earned Value Management (EVM) data used to estimate completion

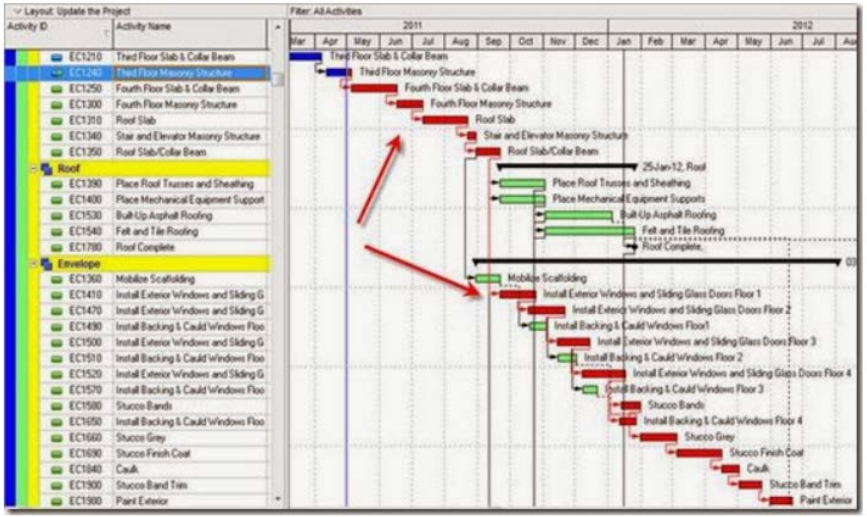
Decisions Informed

- Change schedule and task sequence
- Reassign and reallocate resources.
- Changes greater than a set contingency must be escalated.

Sample Charts for Schedule

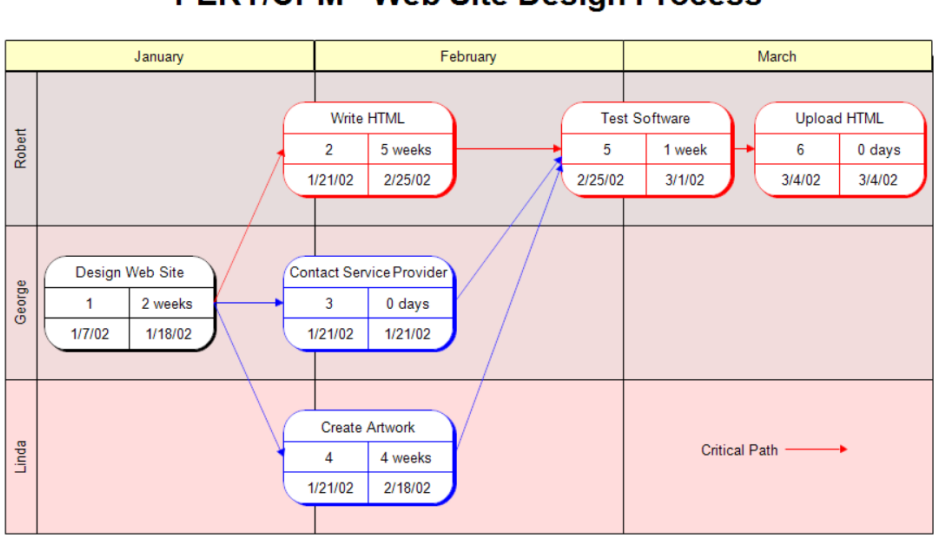
Gantt Questions

- Does Plan vs. Actual show slip?
- Is schedule of future tasks at risk?



PERT Chart Questions

- Are late tasks pushing Critical Path?
- Can we synchronize teams to milestones?
- Are resources ready for next tasks?



Comments about Critical Path

Critical Path addresses concerns for multiple teams and synchronization.

Decisions focus attention on the nearby horizon.

- PERT chart for the next 3 months highlights the next several tasks.
- Are correct resources available and prepared.
- Are inputs ready for tasks?
- Manage the tasks feeding the critical path.
  - Compressing tasks on the critical path is too difficult – 20% maximum at increased cost.
  - We must prevent tasks that feed the critical path from taking too long or taking a late start.

Changes in the Critical Path

- Critical Path should not “move around”. We lose control of schedule quickly because we must continually shift focus of various resources.
- Stoplight could be **Red** if critical path has changed for 3 months in a row or more.

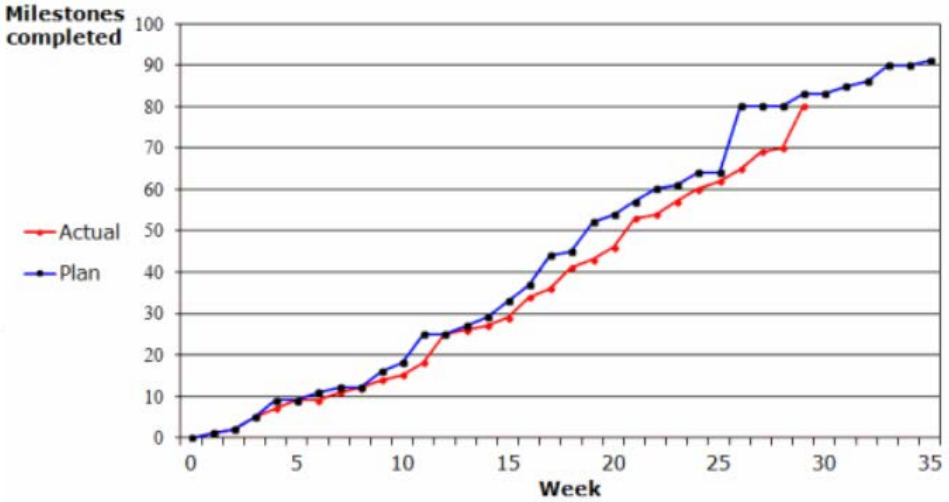
Team Level Schedule Charts

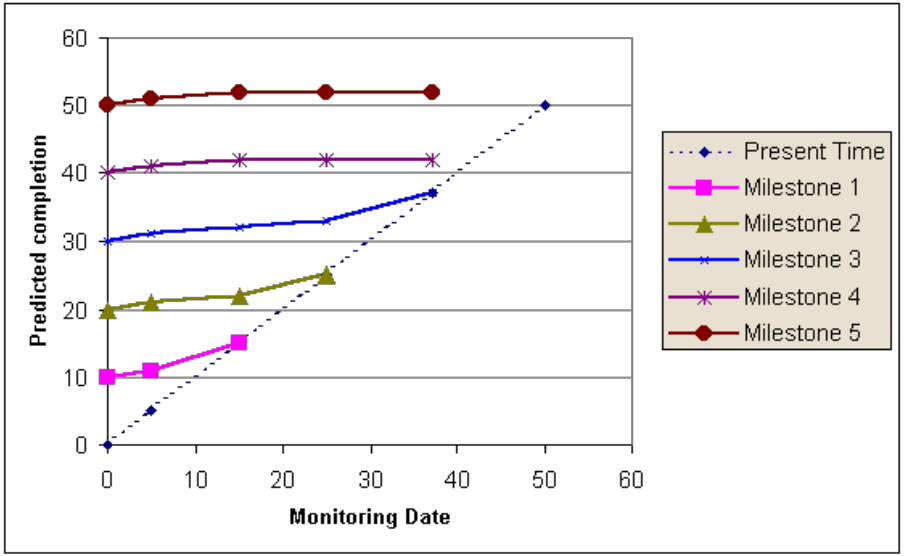
Milestone Hit Rate

- Typically used by smaller teams
- Also “%promises complete by week”

“Slip Chart”

- Are forecasts regularly updated?
- Shows how slip affects later milestones.
- Diagonal is when milestone is achieved.





Resource Allocation and Availability

In preparing the plan, resources are assigned tasks.

- If resources are not available at task execution time, task is delayed.
- If need input is not available for task, then resource is idle and will be pulled elsewhere.

Look back (have resources been utilized as planned?), and

- Have resources been consumed on project tasks for the last month?
- Resource consumption (list of task-hours by task planned), list of task hours worked)

Look ahead (are resources available for upcoming tasks?)

- Plan (list of task hours, resources assigned, hours assigned)
- Do all upcoming tasks have resources available?

Mis-allocation of resources is a common cause of project slip.

- (over-allocation, lack of availability)

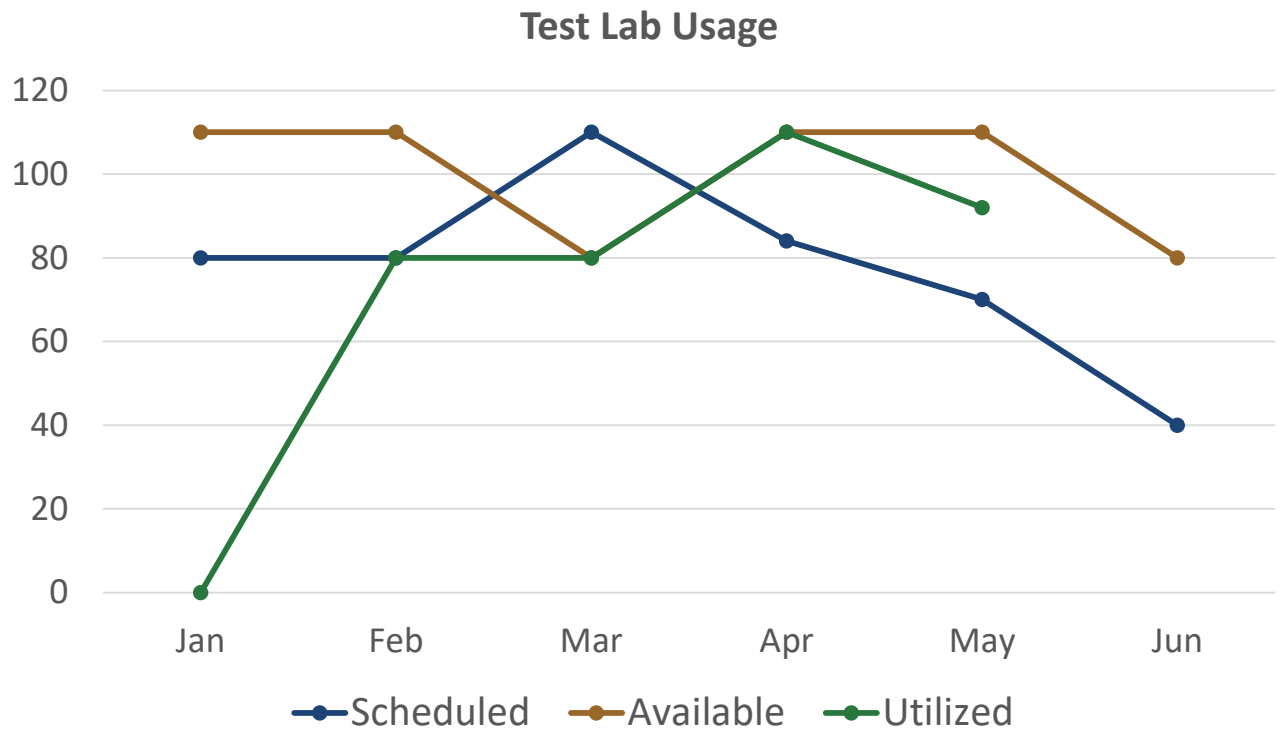
Resource Allocation Report

Detailed chart is often needed

Reporting Period		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Mgt	Planned	1	2	2	2	2	3	4	6	6	6	5	5	2	2
	Actual	2	2	2	2	2	4	4							
Analyst	Planned	1	3	4	4	4	4	4	4	4	5	4	3	1	1
	Actual	2	2	2	4	4	4	4							
Designer	Planned		3	10	11	12	11	13	12	8	3	1	1	0	0
	Actual	4	4	4	7	12	12	12							
Programmer	Planned		1	1	1	2	6	18	33	35	35	19	14	3	3
	Actual	7	7	7	11	20	20	20							
Tester	Planned			1	1	2	2	4	8	14	16	14	10	6	2
	Actual														
Other (QA,CM. etc.)	Planned		1	1	1	2	2	2	2	2	2	2	2	2	2
	Actual			0	1	2									
Total	Planned	2	10	19	20	24	28	45	65	69	67	45	35	14	10
	Actual	15	15	15	25	40	40	40	0	0	0	0	0	0	0

Is there a problem?

Test Facility Availability and Utilization



Earned Value Charts: Combines cost and schedule

Goal

- Identify and communicate schedule and cost gaps.
- Predict completion dates and costs.

Questions

- Are we expending resources at the expected rates?
- Are we completing work at the expected rates?
- What are cost and schedule estimates for completion?
- Do differences between plan and current status suggest a need for a change order?

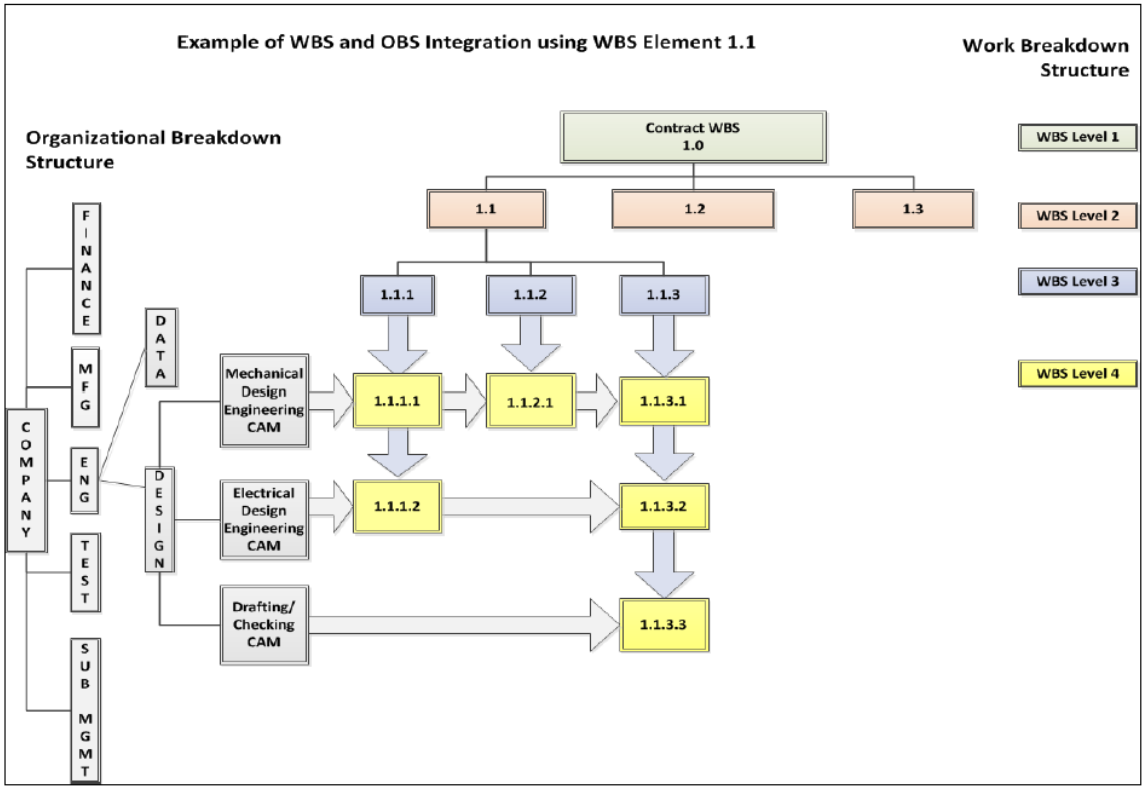
Control Accounts Integrate WBS and OBS

WBS=product lifecycle
OBS=development responsibility

Access to specific work packages (control accounts) can identify specific problems when EVM data is showing a general problem.

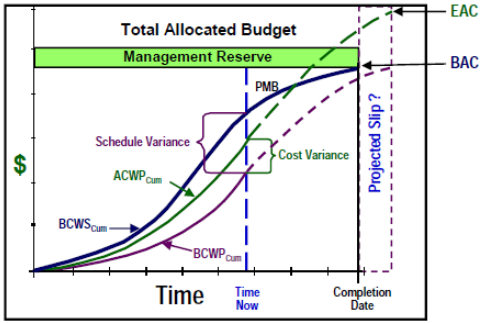
Control accounts are the intersection of elements of product lifecycle and organizational structure – very granular.

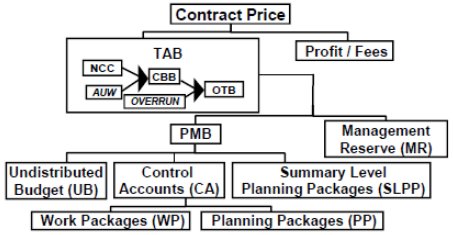
Specific task orders provide the necessary insight to the structure.



DAU: https://www.acq.osd.mil/evm/docs/DoD%20EVMSIG.pdf

Earned Value Management 'Gold Card'





VARIANCES
Positive is Favorable, Negative is Unfavorable

Cost Variance	CV	= BCWP – ACWP
	CV %	= (CV / BCWP) • 100
Schedule Variance	SV	= BCWP – BCWS
	SV %	= (SV / BCWS) • 100
Variance at Completion	VAC	= BAC – EAC
	VAC %	= (VAC / BAC) • 100

OVERALL STATUS

% Schedule	= ($BCWS_{CUM}$ / BAC) • 100
% Complete	= ($BCWP_{CUM}$ / BAC) • 100
% Spent	= ($ACWP_{CUM}$ / BAC) • 100

EFFICIENCIES

Cost Efficiency	CPI	= BCWP / ACWP Favorable is > 1.0, Unfavorable is < 1.0
Schedule Efficiency	SPI	= BCWP / BCWS Favorable is > 1.0, Unfavorable is < 1.0

BASELINE EXECUTION INDEX (BEI) & Hit Task %

BEI = Total Tasks Completed / (Total Tasks with Baseline Finish On or Prior to Current Report Period)

Hit Task % = 100 * (Tasks Completed ON or PRIOR to Baseline Finish / Tasks Baselined to Finish within Current Report Period)

ESTIMATE @ COMPLETION

= ACTUALS TO DATE + [(REMAINING WORK) / (PERFORMANCE FACTOR)]

EAC_{CPI} = $ACWP_{CUM}$ + [(BAC – $BCWP_{CUM}$) / CPI_{CUM}]

$EAC_{Composite}$ = $ACWP_{CUM}$ + [(BAC – $BCWP_{CUM}$) / (CPI_{CUM} * SPI_{CUM})]

TO COMPLETE PERFORMANCE INDEX (TCPI) §

$TCPI_{Target}$ = Work Remaining / Cost Remaining = (BAC – $BCWP_{CUM}$) / (Target – $ACWP_{CUM}$)

§ To Determine the TCPI for BAC, LRE, or EAC Substitute TARGET with BAC, LRE, or EAC

To Determine the Contract Level TCPI for EAC, You May Replace BAC with TAB

ACRONYMS

ACWP	Actual Cost of Work Performed	Cost actually incurred in accomplishing work performed = ACTUAL COST
AUW	Authorized Unpriced Work	Work contractually approved, but not yet negotiated / definitized
BAC	Budget At Completion	Total budget for total contract thru any given level
BCWP	Budgeted Cost for Work Performed	Value of completed work in terms of the work's assigned budget = EARNED VALUE
BCWS	Budgeted Cost for Work Scheduled	Time-phased Budget Plan for work currently scheduled = PLANNED VALUE
CA	Control Account	Lowest CWBS element assigned to a single focal point to plan & control scope / schedule / budget
CBB	Contract Budget Base	Sum of NCC & AUW
EAC	Estimate At Completion	Estimate of total Cost for total contract thru any given level generated by Ktr, PMO, DCMA, etc. = $EAC_{Ktr\ /\ PMO\ /\ DCMA}$
LRE	Latest Revised Estimate	Ktr's EAC or EAC_{Ktr}
MR	Management Reserve	Budget withheld by Ktr PM for unknowns / risk management
NCC	Negotiated Contract Cost	Contract Price Minus profit or fee(s)
OTB	Over Target Baseline	Sum of CBB + additional budget approved for remaining work
PAC	Price At Completion	EAC Plus Adjusted Profit or Fee(s)
PMB	Performance Measurement Baseline	Contract time-phased budget plan
PP	Planning Package	Far-term CA activities not yet defined into WPs
SLPP	Summary Level Planning Package	Far-term contract activities not yet defined into CAs
TAB	Total Allocated Budget	Sum of all budgets for work on contract = NCC, CBB, or OTB
TCPI	To Complete Performance Index	Efficiency needed from 'time now' to achieve a Cost Target = BAC, LRE, or EAC
UB	Undistributed Budget	Broadly defined activities not yet distributed to CAs or SLPPs
WP	Work Package	Near-term, detail-planned activities within a CA

EVM POLICY: DoDI 5000.02, Enclosure 1. Table 8.

EVMS in accordance with EIA-748 is required for cost or incentive contracts, subcontracts, intra-government work agreements, & other agreements valued ≥ $20M (TY $). Contracts ≥ $100M[#] (TY $) require that the EVMS be formally validated by the cognizant contracting officer. [#] Class Deviation-Earned Value Management System Threshold (9/1/15)
EVM is discouraged on Firm-Fixed Price, Time & Material Contracts, & LOE activities regardless of cost.
Refer to the IPMR Implementation Guide for IPMR Tailoring Guidance.

DoD's EVM CONTRACTING REQUIREMENTS:

DFARS CLAUSES	252.234-7001 "NOTICE OF EVMS" FOR SOLICITATIONS
	252.234-7002 "EVMS" FOR SOLICITATIONS & CONTRACTS
	252.242-7005 "CONTRACTOR BUSINESS SYSTEMS" FOR SOLICITATIONS & CONTRACTS
CONTRACT PERFORMANCE REPORT	DI-MGMT-81466A 5 FORMATS = WBS, ORGANIZATION, BASELINE, STAFFING, EXPLANATIONS & PROB ANALYSES
INTEGRATED MASTER SCHEDULE	DI-MGMT-81650 MANDATORY FOR DoD EVMS CONTRACTS
Integrated Program Mngt Report	DI-MGMT-81861 * 7 FORMATS = WBS, OBS / IPT, BASELINE, STAFFING, EXPLANATIONS & PROB ANALYSES, IMS, HISTORY / FORECAST COST
INTEGRATED BASELINE REVIEW	MANDATORY FOR ALL EVMS CONTRACTS
WBS For Defense Materiel Items	MIL-STD-881-C

* Combines & Supersedes DI-MGMT-81466A & 81650; Effective July 1, 2012

EVM CoP: https://acc.dau.mil/evm
eMail Address: EVM.dau@dau.mil
Revised SEP 2015

Earned Value Management – Forecasting BAC EAC

Performance Review

Tasks	BAC	EAC	VAC	2004	AC	EV	TCPI	CPI
				March / April / May				
Project A	$2,975.0	2,355.70	-619.30	$3,000.00	$1,615.00	$2,004.73	0.71	1.24
Task 1	$300.0	350.00	50.00	$2,500.00	$350.00	$300.00	0.00	0.86
Task 2	$400.0	440.00	40.00	$2,000.00	$440.00	$400.00	0.00	0.91
Task 3	$250.0	313.09	63.09		$175.00	$139.53	1.47	0.80
Task 4	$725.0	257.13	-467.87	$1,500.00	$200.00	$563.89	0.31	2.82
Task 5	$400.0	189.80	-210.20	$1,000.00	$100.00	$210.53	0.63	2.11
Task 6	$350.0	314.66	-35.34	$500.00	$200.00	$222.73	0.85	1.11
Task 7	$550.0	491.02	-58.98	$0.00	$150.00	$168.06	0.95	1.12

Planned Value
Earned Value
Budget
Actual Costs

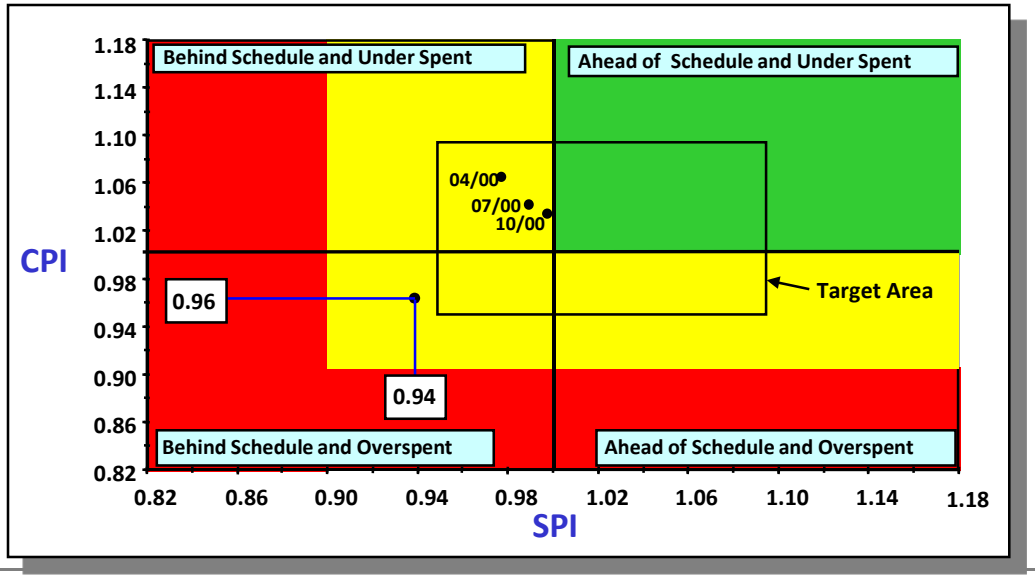
Created using Milestones software
www.kidasa.com

From www.earnedvaluemanagement.com

Efficiency Status Indicator

Change thresholds for CPI and SPI set here at 10% below nominal.

- Green area is ok – within budget and schedule plan
- Yellow is a warning.
- Red signals that a change in plan is required.



CPI=Cost Performance Indicator
SPI=Schedule Performance Indicator

Comments about EVMS

EVMS aggregates a lot of information into 3 numbers.

- Earned Value data is obtained from the accounting department.
  - It is often a month later than the date it was captured.
  - Problems that do not immediately contribute to those numbers may hide for a long time.

- We need many accounts for milestones/deliverables.
  - We must have clear completeness criteria. What do we meant “80% complete?”

- Accurate estimation is valuable and critical.
  - EVMS can tell you there is a problem and how big it is.
    It does not tell you where to find the cause or how to fix it.
  - Use EVM CPI and SPI to re-estimate and re-plan.

Schedule/Resources/Cost Dashboard

Use “stoplights” for the top level.

- Green=ok, Yellow=concern, Red=**Current plan is not working**

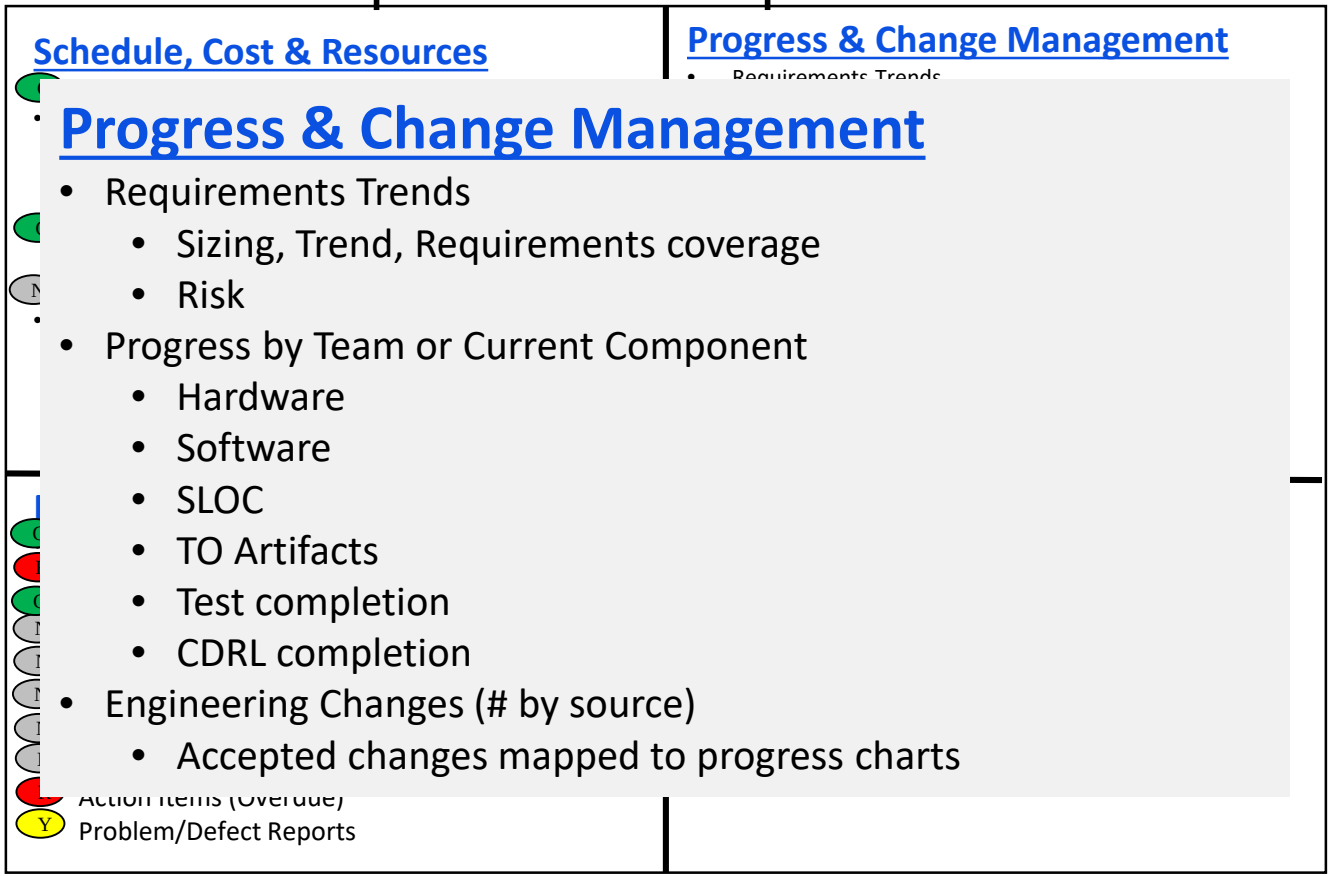
Typical backup charts

- Gantt chart to identify late items and collection of starting tasks
- Schedule slip by team
- CPI/SPI used to prepare estimate at Completion (cost and schedule)
- Selective resource allocation/utilization
- Resource turnover
- Critical path stability and near term dependencies

SSD- Dashboard

Progress and Change Management

Dashboard Top-Level Example



G Within plan

Y Within contingency

R Change required

N Not used this period

Details by Quadrant and Stoplight

It All Begins with “Scope”

We need a definition of scope agreed upon by stakeholders.

Scope: Include every work product to be delivered outside the project team.

- The components of the system
- Training materials
- External demonstrations and certain test events

Every element of scope involves multiple steps, people and significant effort

- Customer(s) identification
- Defining some requirements
- Quality goals, and quality check procedure
- Work and packaging

You can will measure progress in various ways during project lifecycle.

Progress and Change Management

Goals

- Track scope-element completion against plan.
- Recognize changes in scope and prepare the case for change.

Questions

- What is the estimated completion date?
- Is an increase is schedule or cost warranted?

Indicators and Measures

- Measures of progress by size or components or requirements
- Measures of productivity and allowable contingency

Decisions Informed

- Add or decrease scope based on productivity and contingency measures
- Create/Approve change requests

Progress and Change Questions

Chart should have three types of data:

- Estimated target value (range) that should be adjusted with changes
- Planned trend line
- Actual observed data and frequency of data collection

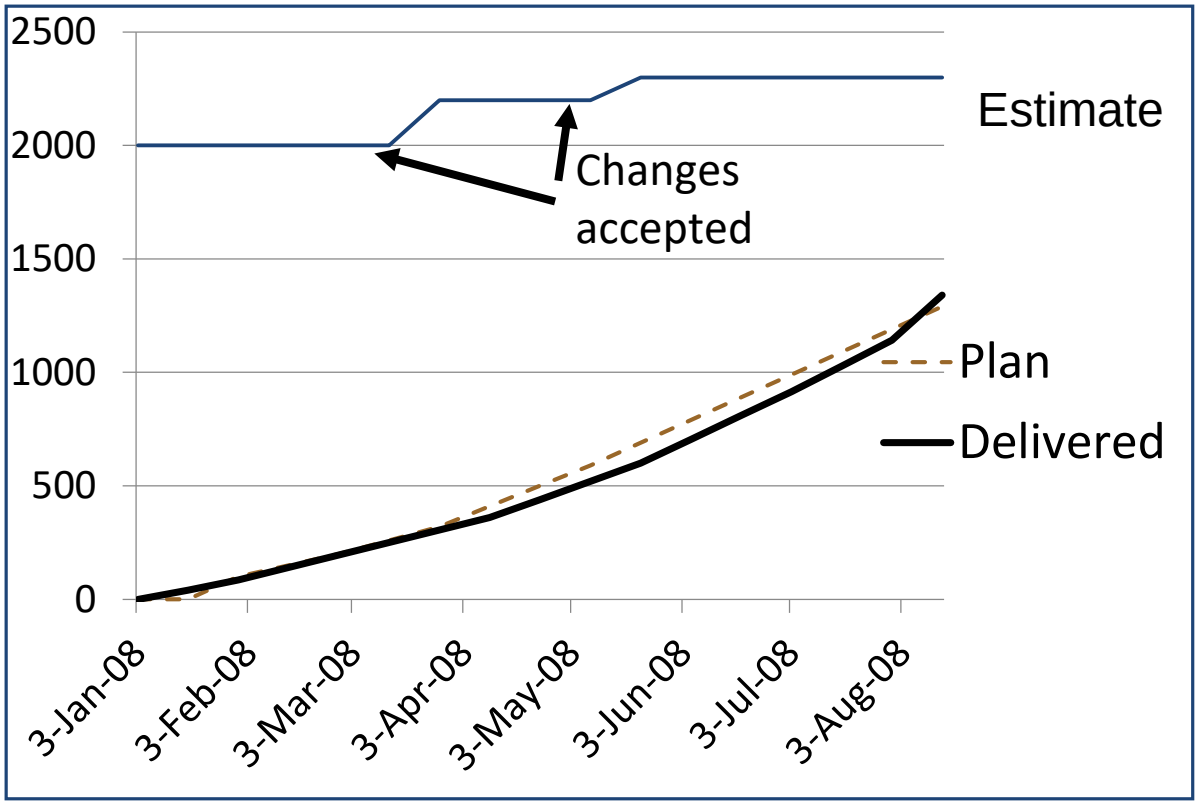
Change Threshold

- Maximum allowable change in target before change is mandated.

Record of Changes

- Show change by adjusting threshold or target line.

Progress Chart for Code Size



What do we mean by "Delivered"?
How often can we measure progress?

Track Progress by Team

A single team usually works a single deliverable at a time.

Operationalize the definition of measuring progress:

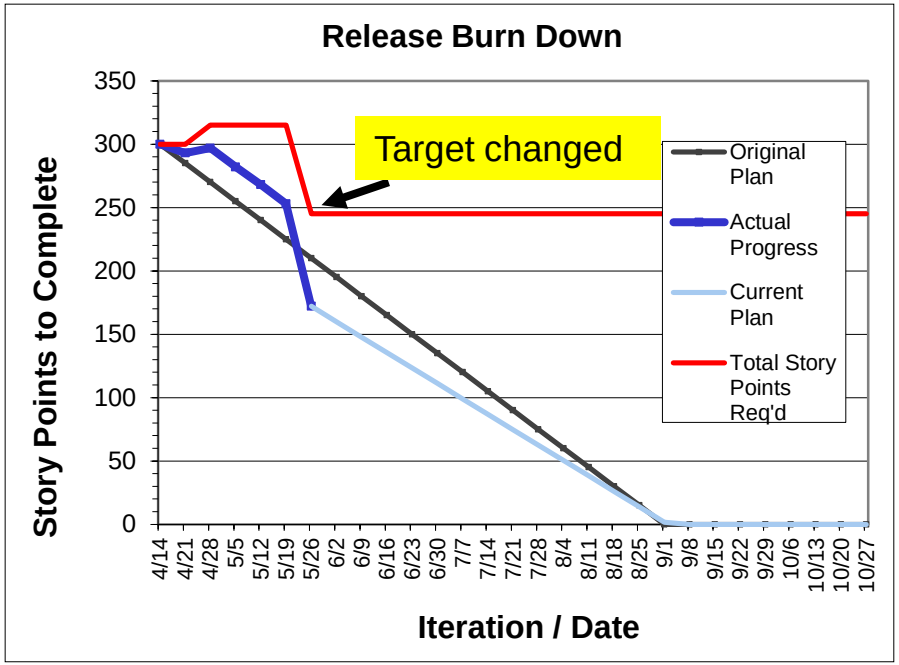
- Count completion of something: LOC, stories completed, requirements, specifications, modules or tests completed. “Completion” requires a quality check of some type.
- Anything that can be observed at a handoff as “small, medium, large” is an opportunity to identify an accumulation of “size”.

Tracking opportunities:

- Inspection or static analysis tool or code counting tool
- CM System represents a handoff by “promotion”
- Use defect tracking system to monitor defect closure
- Track test cases passed for progress

Burn Down

Agile teams usually use a burn down chart for progress



A "Burn Down" chart is ok for a single team, but progress toward a target value works better for consolidating teams and preparing for change.

I can't make a "productivity estimate" here.

Tracking Progress Using a List of Requirements

Identify each requirement

Each requirement propagates to:

- Software specification for components
- Design documents (or use cases)
- Coded components
- Test cases

Therefore **“Requirements covered”** for each step provides some interesting insight.

- Percentage of requirements with design definition can show progress.
- Test case definition can also be monitored this way.
- When requirements change, the trail is then easy to follow.

Design Coverage Analysis Using Requirements List

Requirement Number	Module Name
101	A1, B2
102	A3
103	
104	C2

Which requirements are not yet covered?
Calculate % = Rows with modules assigned / #requirements
At what percentage of non-coverage are we at risk?

Ways to Measure Testing Progress

Test case coverage
- %requirements, %specifications or %use cases
- %lines of code

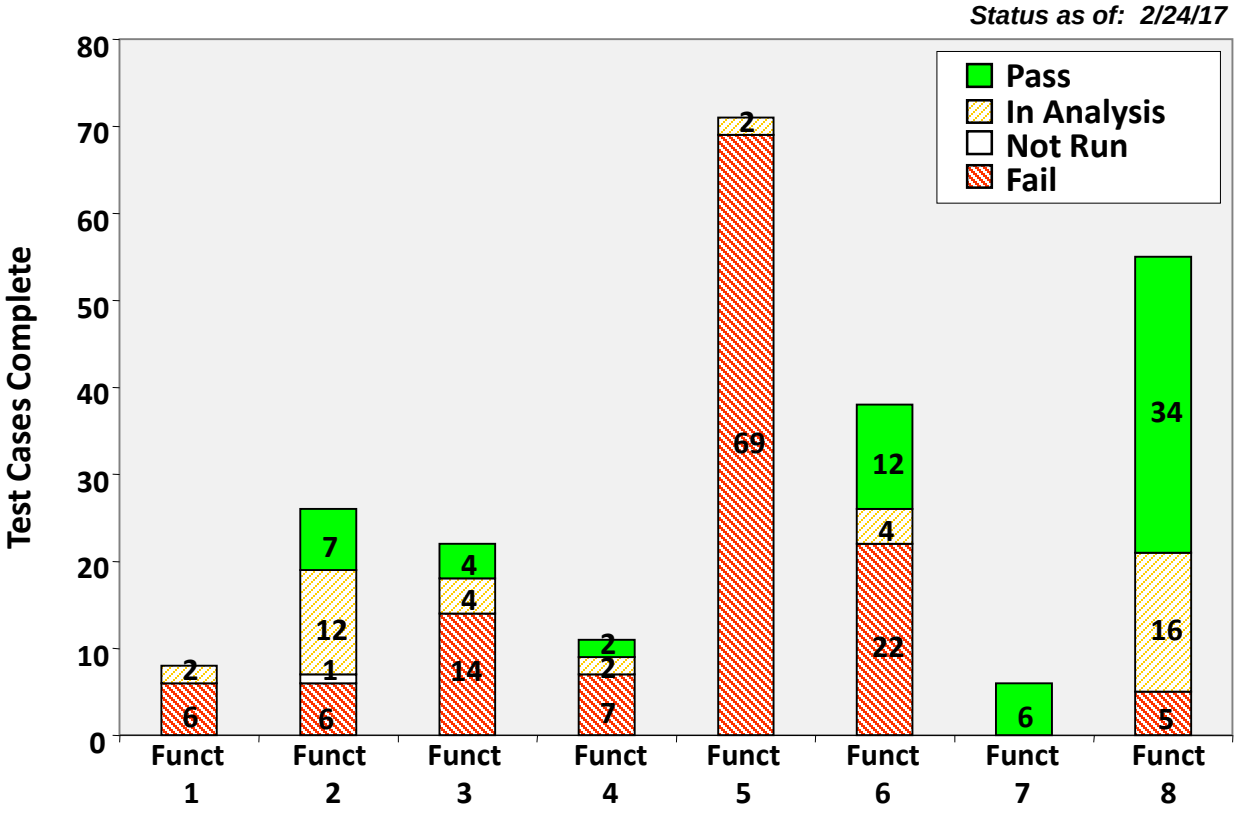
Tests executed
- %requirements, %specifications or %use cases
- %lines of code
  - Typically counted with automated tool

%Tests passed (not #defects)

Remember that you need to define the satisfactory % to count for completion.

Example Test Case Completion

Have we passed sufficiently many tests?



Progress Indicators Across the Lifecycle

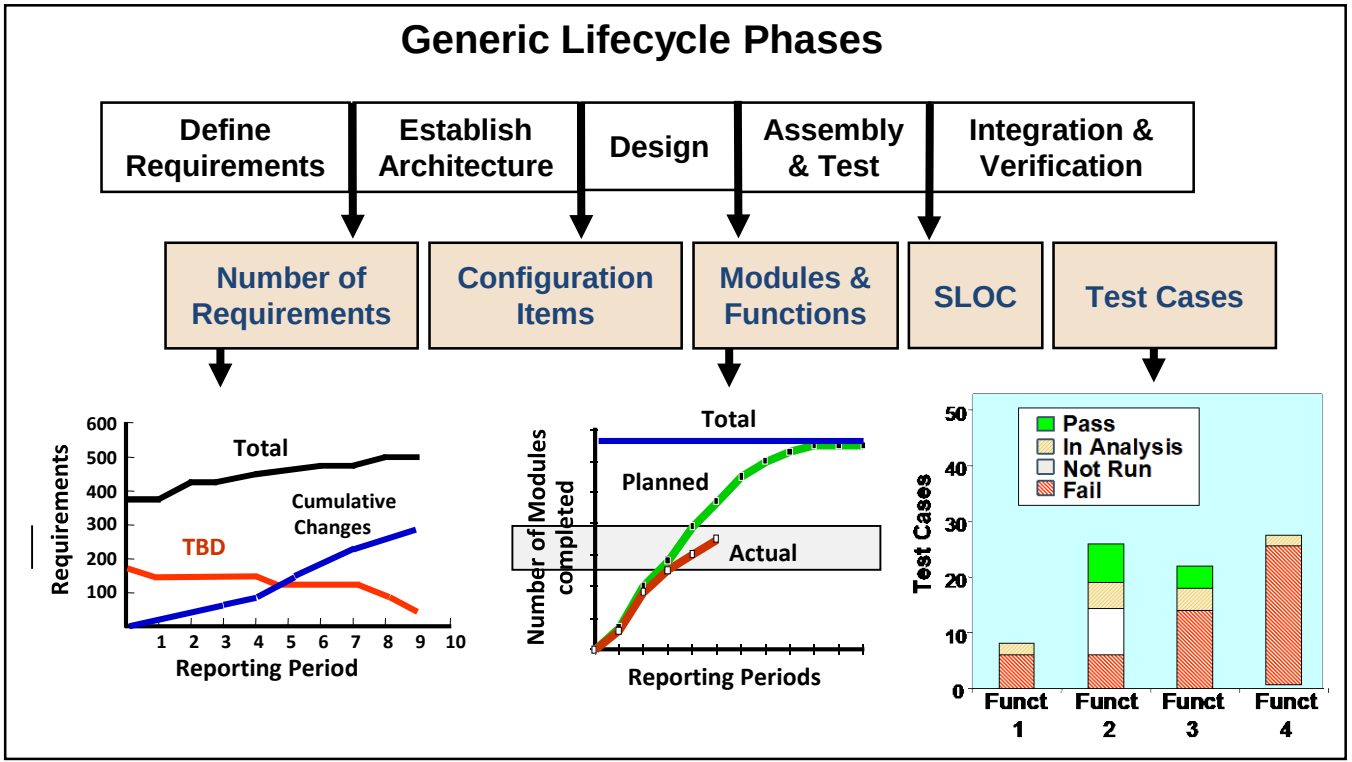
During development different teams do different types of work.

Different kinds of work product are produced as well.

Hence measures may also differ.

- A count of requirements accepted for current build
- Design may be documents or models or prototypes measured by size or some count.
- Code is stored in CM system measured by LOC or module count
- A number of test cases are developed
- A count of tests are executed and/or passed (or failed)
- A count of defects remaining open to reduce below a threshold value

Progress Indicators by Activity and Artifact Type



Mark milestones or events or handoffs to compare to progress

Opportunities to Measure at Event or Periodic Basis

Milestone events and other work **handoff** events

- Estimate: Staff estimates work and gets agreement to proceed.
- Design Completion: Reviews, artifacts checked-in to CM system and validated.
- Build, handoff to testing, completing tests.

Periodic Given a Size Measure: *Where is artifact stored?*

- #Requirements covered (Design inspection)
- #Pages of paper reviewed or delivered
- #Reviews completed
- #Lines of code written
- #Tests planned and documented
- #Tests performed
- #Tests passed
- Any estimated task value that can be measured

The CM System

Configuration Management check-in and promotion procedures may be used to track progress provided that the procedure includes some quality check.

- Typically, Check-in means work may be awaiting quality check.
- A quality check implies fitness for use. CM promotion used for “ready to build”.

The ability to monitor progress is one reason to place more intermediate deliverables under some configuration management.

- Design documents
- Test scripts
- …

Indicator should show progress compared to plan.

Informing Change Decisions

A change requires you to re-open something you thought had been completed.

- Re-planning or modification of artifact or both?
- How much of the lifecycle is affected?

New requirement:

- The software/hardware specification was derived from the requirements.
- A new requirement means those specifications must be adjusted and followed on to design and development activity.

Always identify the chain of deliverables that must be fixed when doing change management. Complete the re-estimate and re-planning.

Every Change Forces Some Rework

A change requires you to re-open something you thought had been completed.

- Replanning or modification of artifact or both?
- How will the dashboard (graphs) reflect the change?

Sample: New requirement:

- The software/hardware specification was derived from the requirements.
- A new requirement means those specifications must be adjusted and followed on to design and development activity.
- Each derived component and test case will also be modified or added.

Always identify the chain of deliverables that must be fixed when doing change management and re-estimate.

Methods for Re-Estimating Changes

Change requests come in many forms:

- Feature, requirements, or component
- Bug fix, product performance deficiency or other error
- Plan (Schedule and resources)
- Supplier or technology change

Identify which development artifacts must change.

Use performance records to get “productivity” (effort / size) by artifact (of Task Order)

- Average requirement cost to date
- Code productivity
- *Many examples*

Estimate lifecycle effects on each element of scope affected.

Update measures and the dashboard.

Sample Productivity Shortcut

If we have many requirements, then a count of the requirements is predictive.

- Phase 1 developed 33 requirements and took 24 weeks.
- We have 72 requirements for the next phase
- A quick estimate for time required is 72* (24/33) = 52.3 weeks
- If the current plan is for another 24 weeks, we probably cannot make that date.

Clearly, this is a crude estimate but such estimates are often useful.

Other opportunities for crude estimates

- Story points
- LOC
- Test cases

It is better to use current performance data rather than benchmarks to re-estimate.

Decision Potential for Progress

Monitoring progress is key to a good change management process.

Decisions include

- Whether to accept a scope (requirements) change.
- Whether to re-allocate scope to a different team (productivity is slow)
  - Can include subcontracting
- Identify which previously completed work products must be modified.
- Making changes: start with the furthest upstream part of the lifecycle to estimate

It is not necessary to be super-precise in our size measures if we are counting many things, statistics works in our favor.

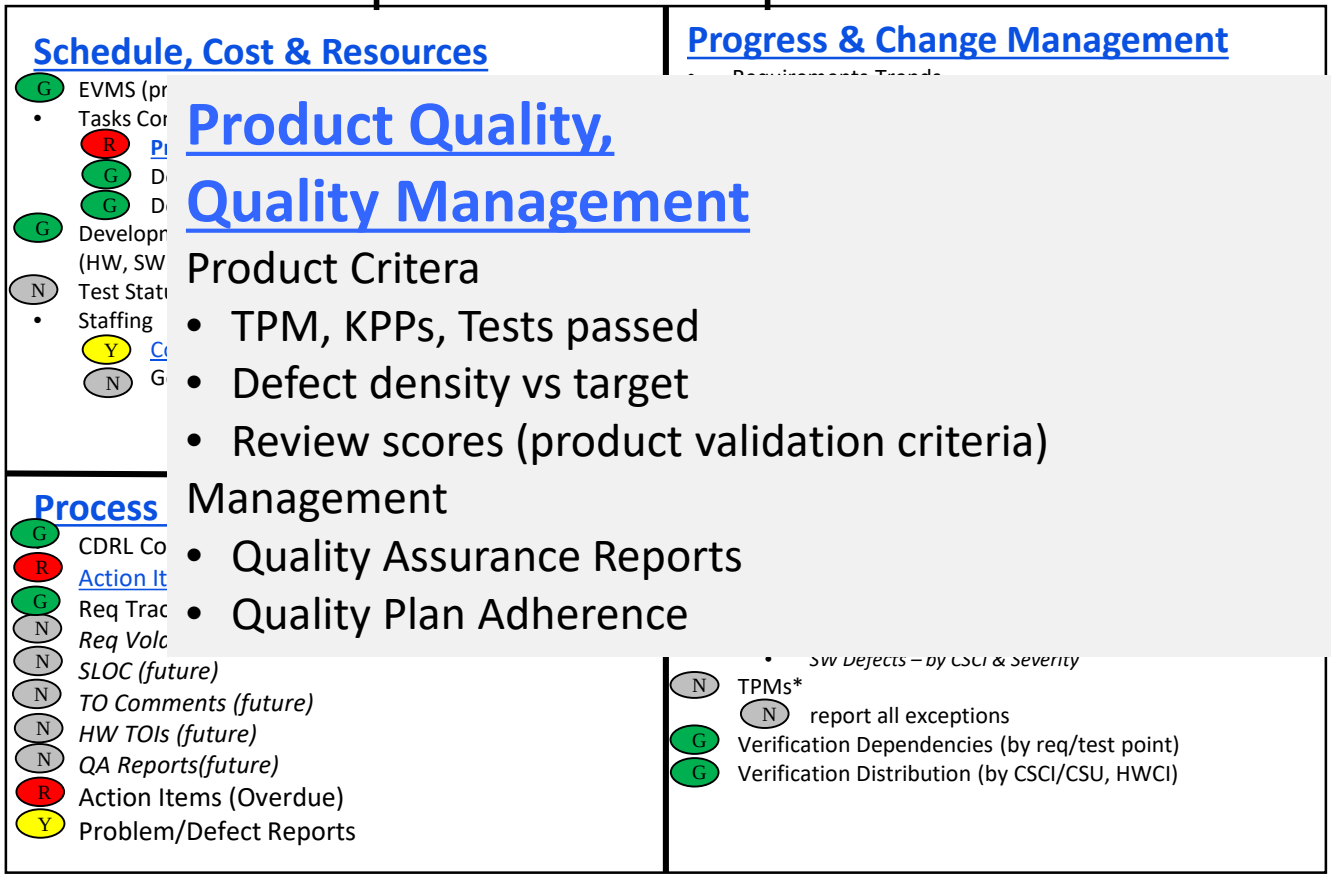
Remember: We count size in order to develop a productivity ratio for change estimation.

Since there are many small components, an average for productivity will be ok.

SSD- Dashboard

Quality Management & Product Quality

Dashboard Top-Level Example



G Within plan

Y Within contingency

R Change required

N Not used this period

Details by Quadrant and Stoplight

Quality Management & Product Quality

Goals

- Validate product fitness for use across all customers
- Reduce Deficiencies

Questions

- What scenarios of use are currently important and what is our current assessment?
- Are planned quality activities performed and are we making a good product?

Indicators and Measures

- Validation activities and scores of exercises.
- Product and process assurance reports.

Decisions Informed

- Changes to product performance criteria or additional testing/review
- Change requests to improve development quality performance

Basis for Measurement: Quality Goals/Targets

Quality targets can be set for several different types of deliverable as well as the product.

- Test cases passed
- Technology Readiness Level (TRL)
- Technical Performance Measures (TPM)
  - Typical: transaction speed, throughput, real-time performance and other performance goals
- Set a target for defect escapes – every design artifact, drawing, code module
- Safety
- Cyber security
- *Scenarios that reflect product use and support*

Quality Management defines types of quality check and allocates resources to checks

The best defined quality targets contribute to product validation work.

Discuss Quality Targets

Defects:

- Estimate defects across components
- Set targets for defect removal

Test Cases Passed

- By functional area is closer to product validation goals than by component

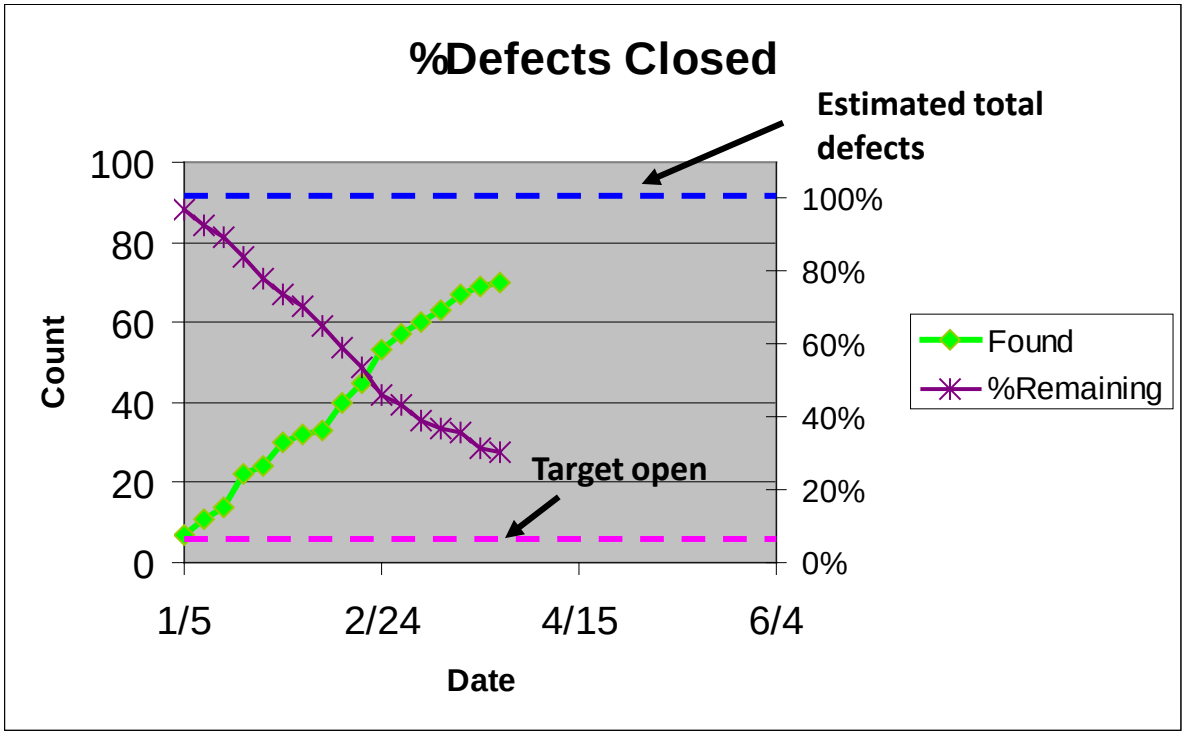
Performance factors are directly tied to product validation – fitness for use.

- Sometimes called “Non-Functional Quality Attributes”
- How would you check a TPM or KPP during design review?
- How can you check for safety?
- What other key performance criteria are important?

Defects as a Quality Target Indicator For Completion

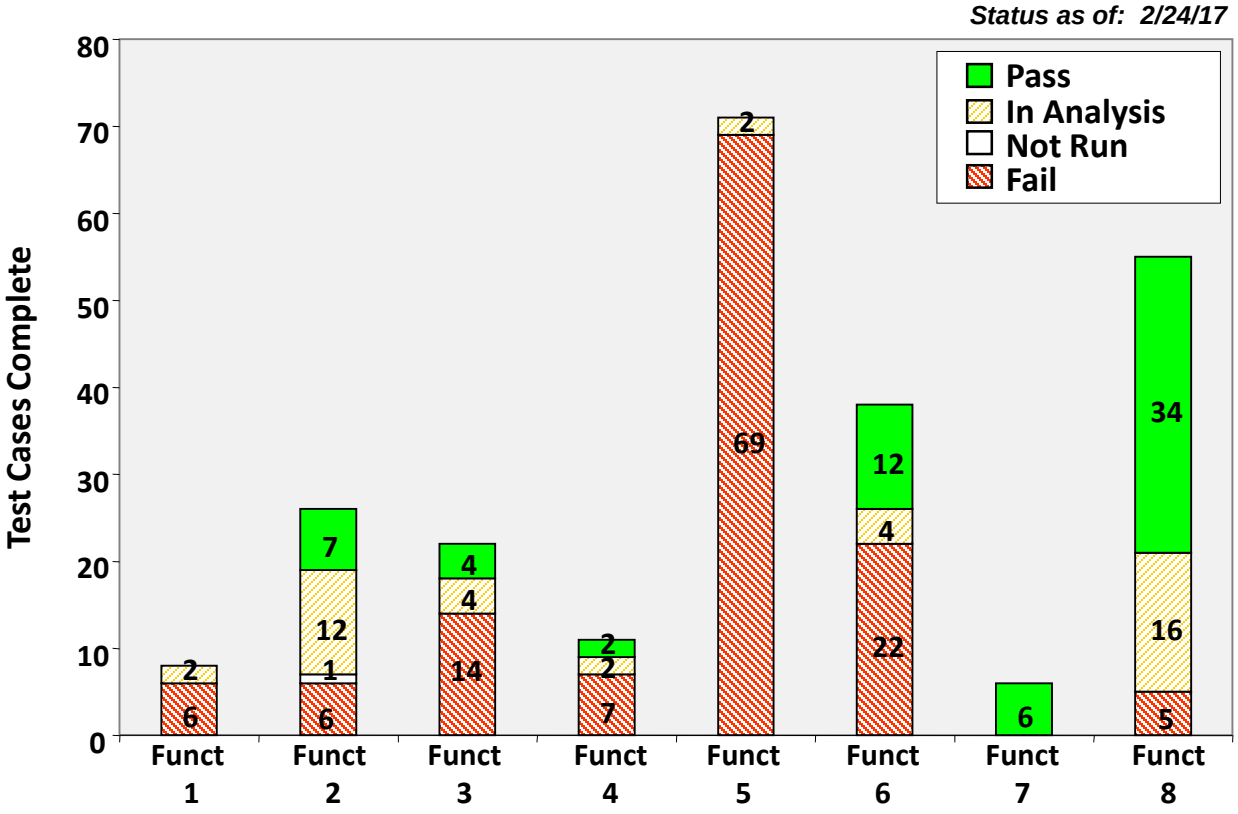
Either measure
- Found vs Estimate, or
- Open vs Target

"Found" measures defect discovery
"Remaining" measures non-quality



Example Test Case Completion

Have we passed sufficiently many tests?



Product Validation Requires Scenarios of Use

Many types of validation scenario are required.

Mission performance scenarios:

- Software startup
- Standard missions
- Missions performed when some situation is not ideal

Logistics and field support scenarios

- Deployment and field setup
- Repair

Product change scenarios

- Reload software

Scenario Construction

Context:

- What is the current context of operation. (Battle, fielding, training?)
- What is the situational need? Operations? Logistics? Maintenance?
- Operation under normal or abnormal conditions?

Stimulus:

- External request for action either by user, other system or threat, etc.

Response:

- What action does the system or system operator do in response?

Outcome:

- What is the desired outcome?
- What actually happens?
- Give the system a grade

Using Scenarios During Design Reviews

A scenario can be used to direct a portion of a design review.

“Show me how the design executes this scenario.”

- Map the stimulus-response through the various components and diagrams..
- If the reviewer cannot describe how the scenario maps onto the components, the design may be incomplete.
- If the developer has misunderstood the desired response, it can be clarified.
- Key performance parameter values can be checked. (rate-monotonic-analysis)

Scenarios can be readily modified for further checks.

- Change the context of operation.
- Introduce a new threat or different response by the operator.

Examples of Scenario Use in “Design Review”

SEI’s Architecture Tradeoff Analysis Method (ATAM)

- Identifies “quality attributes” – critical performance criteria
  - Speed, throughput, real-time criteria, maintainability, etc.
- **Scenarios** are used to check the architecture
- “Risk Themes” identify those areas where architecture is not likely to satisfy performance criteria.

SEI papers:

- Barbacci, et. al., “Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate the Software Architecture for a Product Line of Avionics Systems: A Case Study”, https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=6447
- Ferguson, et.al., “A Method for Assessing Technical Progress and Quality Throughout the System Life Cycle” https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9055

A Score for a Design Review

Goals for a design review

- Coverage of a set of requirements by components affected
- Coverage of mission for various users (warfighter, support, product maintenance)*
- Performance goals for components are satisfied
- Engineering design is intended to address performance criteria
- System workflow analysis shows how components are employed*
- Holes in the analysis can be identified (review process!)

Score design by first 5 items above

Provide action items

*Scenario-based

Quality Management Defines Quality Checks

By activity and artifact

- Meets design standards for formatting, names, identification, interface …
- Automated checks
- Simulations and prototypes
- Various types of testing
- Inspections and formal reviews

Process Checks

- Planned quality activities are staffed, executed reported and in timely fashion

Validation requires us to look at development outcomes – do we get the right products?

- **Reviews using scenarios including supply chain, logistics, maintenance**
- **User based testing**
- **Measures of coverage**

Checking Quality: The Quality Analyst Report

Content Outline

- List of quality interventions executed in period.
- Overall performance (satisfactory or not) ratio and trend.
- Specific non-conformance list.
- Risk assessment of non-conformance.

Without the risk assessment, you may be fixing unimportant problems. Make the QA analyst be a useful contributor to the project team.

An overall downward trend is symptomatic of other problems and pressures. ***Pay attention!***

Sample Quality Management Report

List of quality interventions in period.

- 4 checklist audits
- 12 inspections

Overall performance ratio and trend.

- A single audit of CM had one non-compliance
- All inspections were within expected ranges.

Specific non-conformance list.

- One team did not follow naming convention for modules and external variables.
- Inspection: Total Major Defects=3. Fixes assigned

Risk assessment of non-conformance.

- Recommend correction to source code compliance as this may cause future testing and design problems.

Question: What would stoplight(s) for this chart look like?

Anticipate Product Quality Before “Testing”

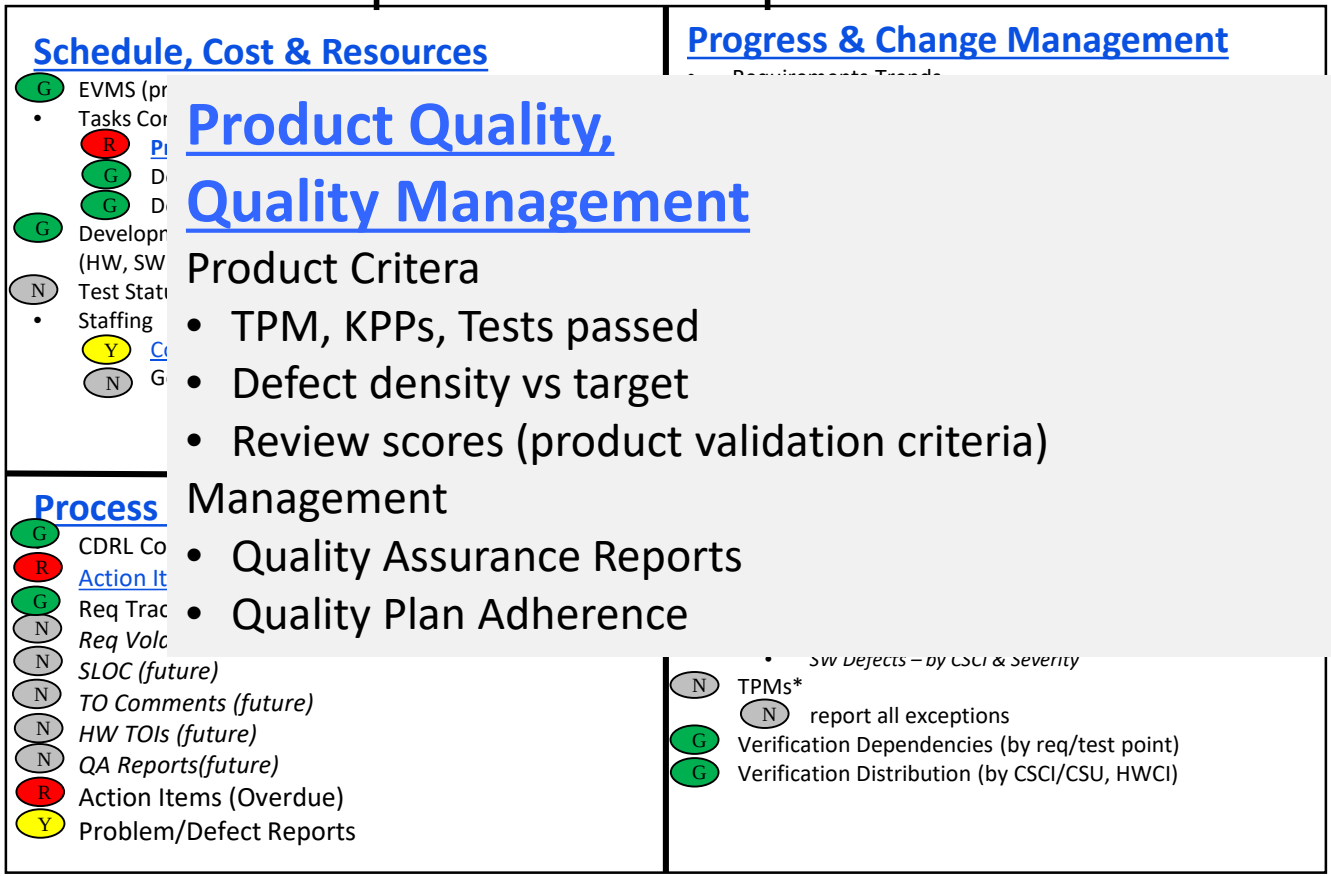
Most Testing Occurs Late in the Development Process

- The “Signal” from testing provides us only a small sample of resultant quality.

Waiting for test results is like managing a sports team by wins and losses.

- Do a test event and validation event at least 2x per year.
- *We need early signals of system performance and quality. (Think “Moneyball”)*

Dashboard Top-Level Example



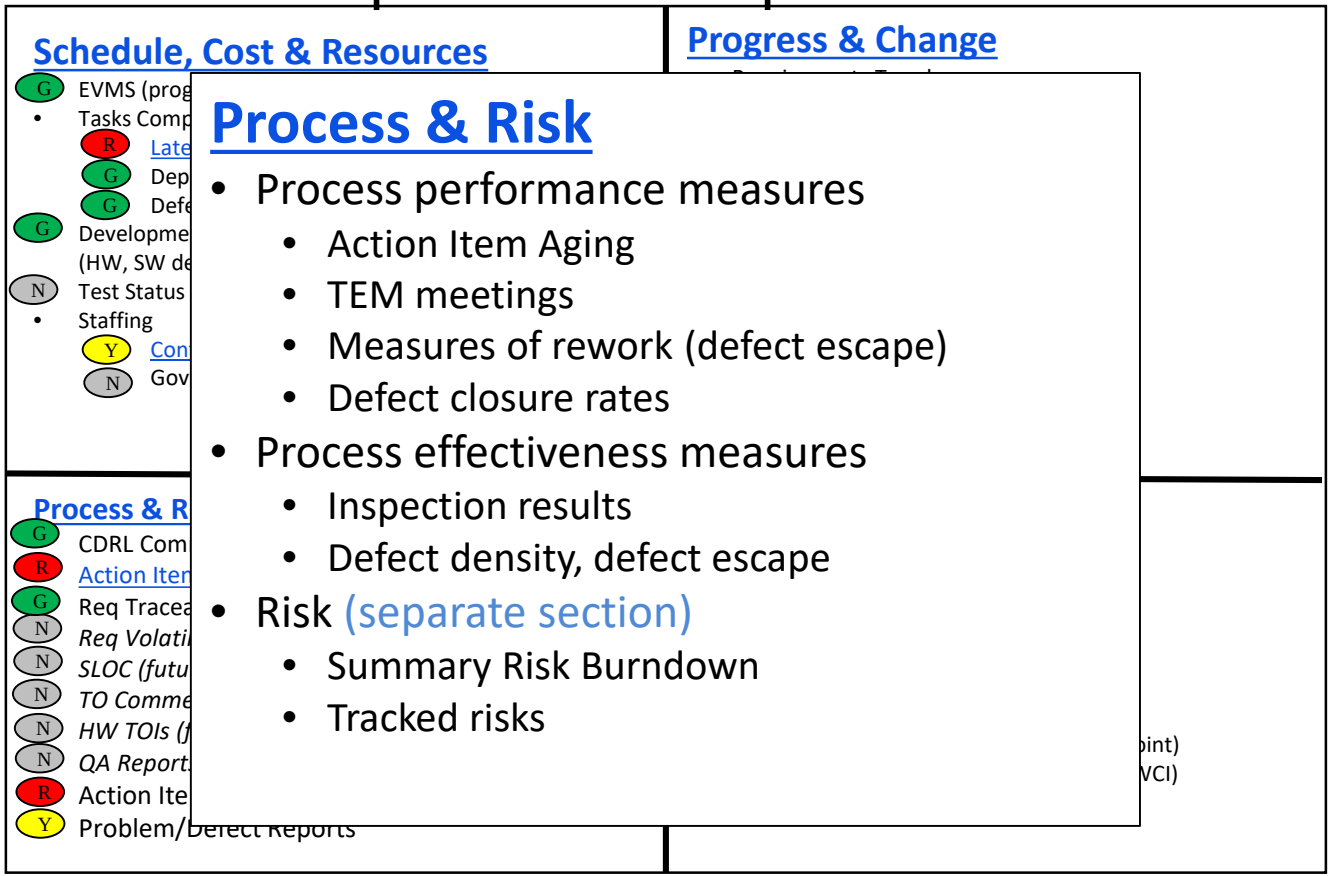
- Within plan
- Within contingency
- Change required
- Not used this period

Details by Quadrant and Stoplight

SSD- Dashboard

Effective Process

Dashboard Top-Level Example



G Within plan

Y Within contingency

R Change required

N Not used this period

Details by Quadrant and Stoplight

Process Effectiveness

Goals

- Reduce rework
- Establish and meet productivity and quality goals

Questions

- What is a “normal rate” for rework? Can we reduce it?
- Are handoffs between processes efficient or do we have delays?
- How can we catch “process breakdown”?

Indicators and Measures

- Monitor select processes for rework and throughput.
- Use select process performance metrics to catch breakdown

Decisions Informed

- Which process improvement activity will improve production?

Process Breakdown

A “process breakdown” is the failure of a process to provide a useful result on time.

- There is a normal pattern of execution for each process.
- Too much uncertainty about time can be as bad as failure.

When the pattern is not normal, we should ask whether it is a good thing or not.

- Is someone missing? Did we change tools or technology?
- Did we have to pick up something we thought was completed?
- Was input was late or missing?
- Are there new errors?
- Are management messages confusing or missing?

Pick a few processes to watch.

- Handoff from developer to integration test – “bad builds”
- Test readiness – test cases that report failure but test case was improperly designed

Analyzing a Process

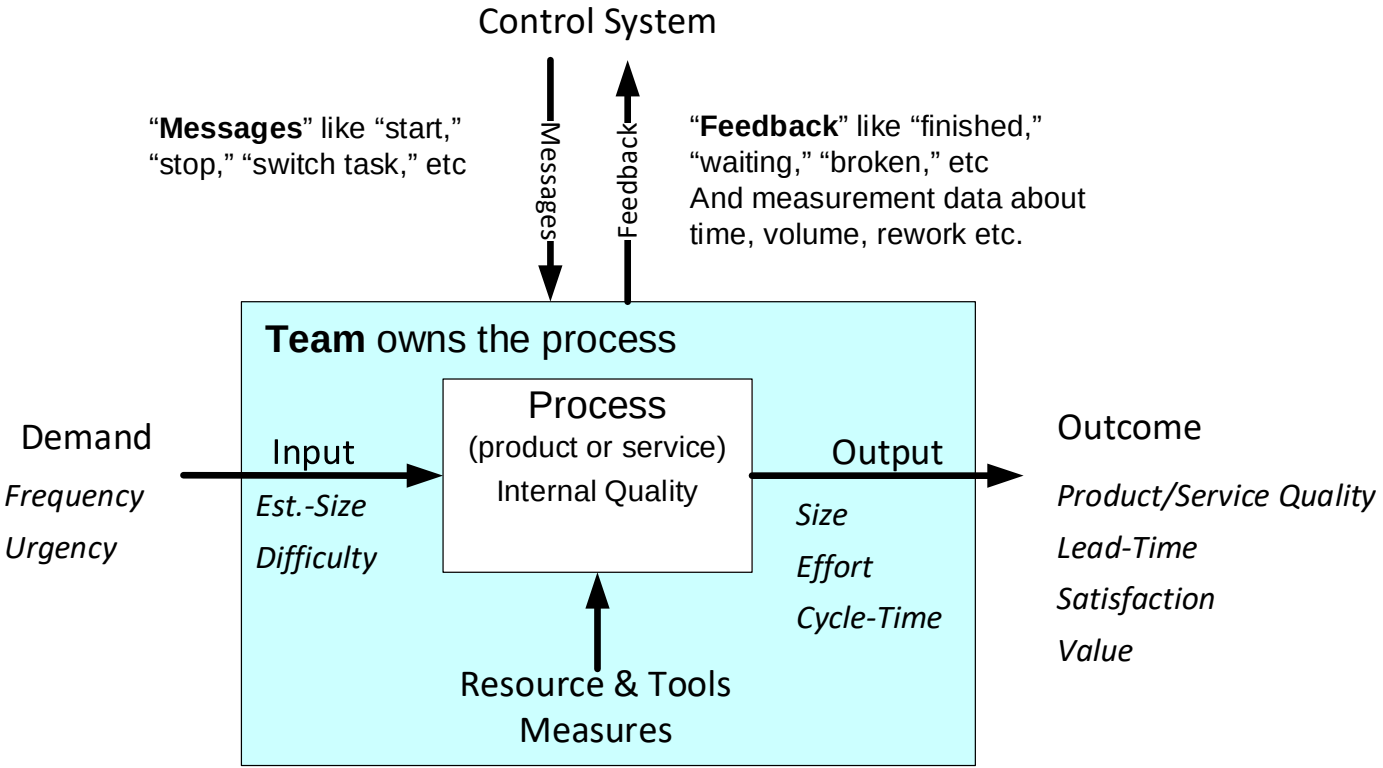
The most commonly used measures for a process are:

- **Productivity**: cost or effort per work product by size or other input parameter
- **Cycle-time**: how long resources are occupied per deliverable
- **Lead-time**: Time from request for work until work is handed off
- **Rework-rate**: Counting occurrence of rework to estimate quality of performance
- Frequency of breakdown or work stoppage
- Internal analytics to test specific measures of performance (e.g. new tools, tool wear, defect find rates) such as static analysis of code.

Define a set of “process boundaries” to show handoff and use the following picture to see the many kinds of observation that can be used.

Dozens of Opportunities to Measure

We need only choose the best of the easy ones.



Monitoring Performance of a Meeting

We have a technical exchange meeting (TEM) every week.

How would we know the meeting process was working?

- Action items are closed promptly.
- The people who are supposed to be there are present.
- Key people are replaced when necessary.
- Everyone gets an agenda before the meeting.
- Meeting starts on time.
- Meeting does not drag on.

Score each factor

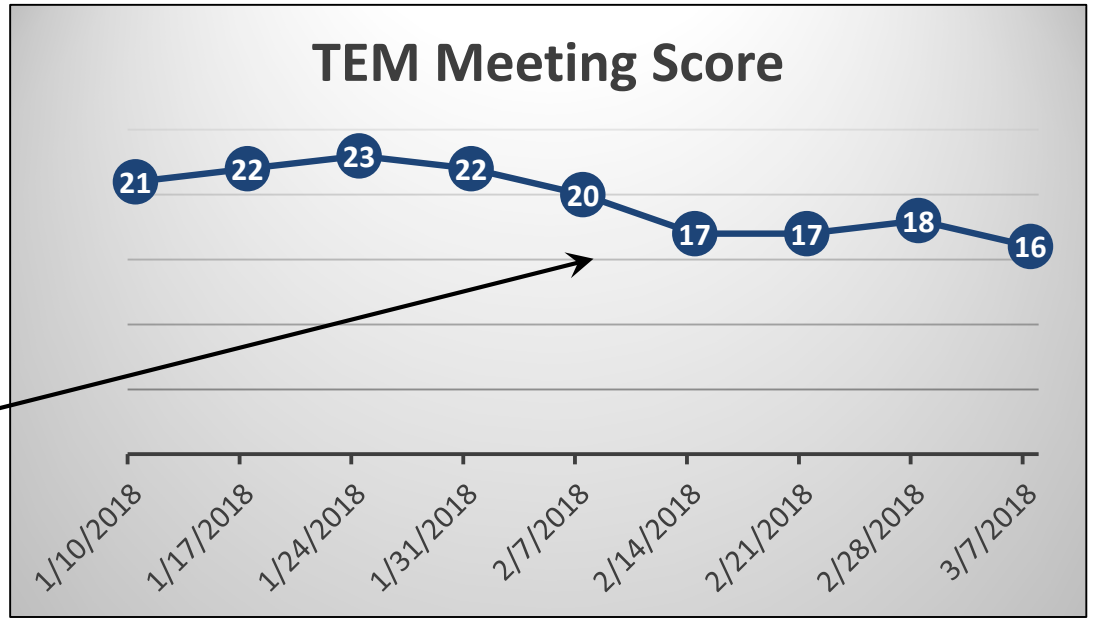
- Scale 0=Unsatisfactory, 1=Fair, 2=Satisfactory, 3=Excellent
- Plot summary by meeting date.

Chart the Meeting Score

Typical behavior will be near constant.

Watch out if behavior declines!

This one can be tracked by phone.



A Defect Process

Much can be learned by monitoring a fairly complicated process.

Defect and Defect-Process Characteristics

- Functional Area
- Affected Component
- Severity/Priority
- Dates in ClearCase
- How discovered
- How fixed

Process Indicators

- Defect Processing
- Defect Discovery (testing, inspection)

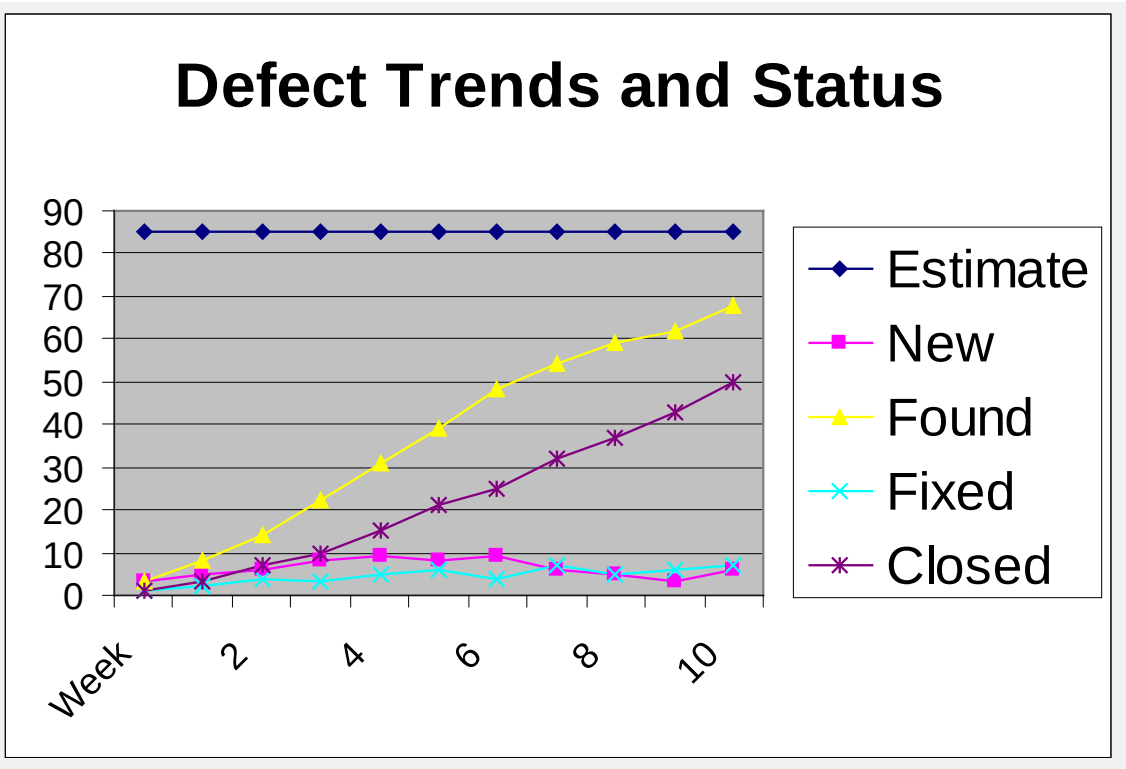
The Defect Management Process

What is the difference between “New” and “Found”?

Between “Fixed” and “Closed”?

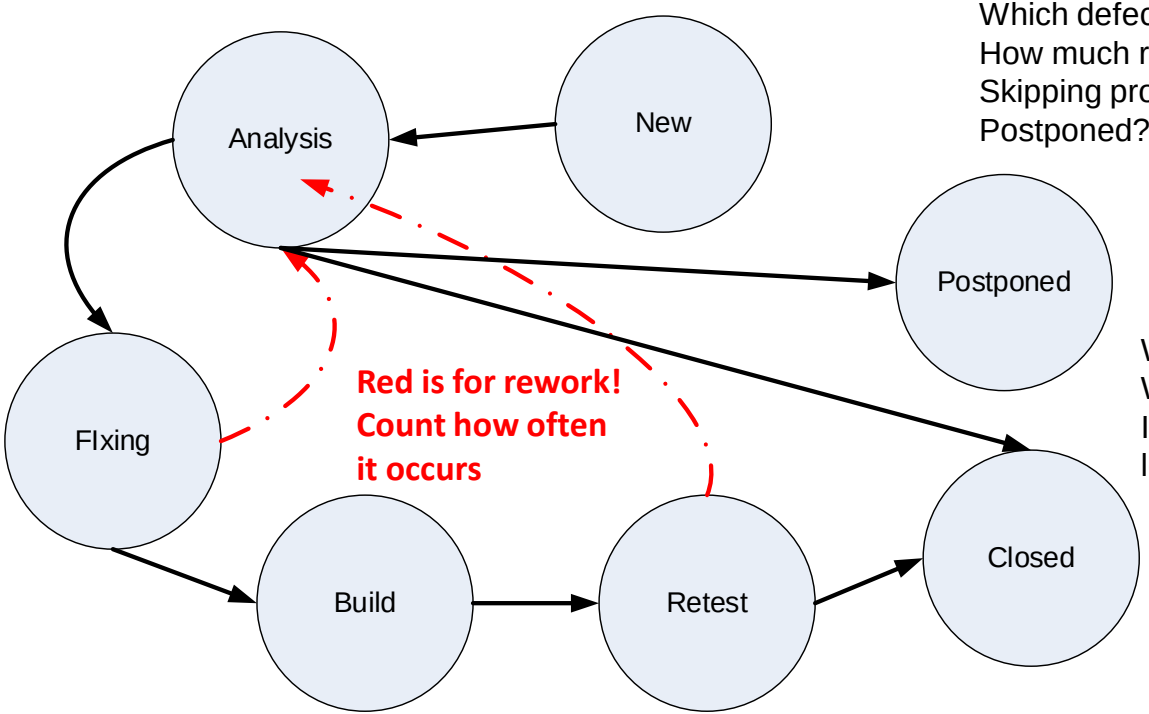
In terms of “the decision” why would you prefer one over the other?

The participants will be clear about their use of language.



Defect Management Process

Use term “State” to describe the type of work occurring now.



Which defects get “stuck”? Days in state.
How much rework? Count backtracking arrows.
Skipping process steps?
Postponed? For how long?

What is an average rate for throughput?
What is 90th percentile?
Is there a class of defect that takes a long time to analyze?

Testing Process Objectives

Project Goal: Verify the system works

Test Team Goals: Deliver the test results
Document test failures
Estimate next time

Test Team Objectives

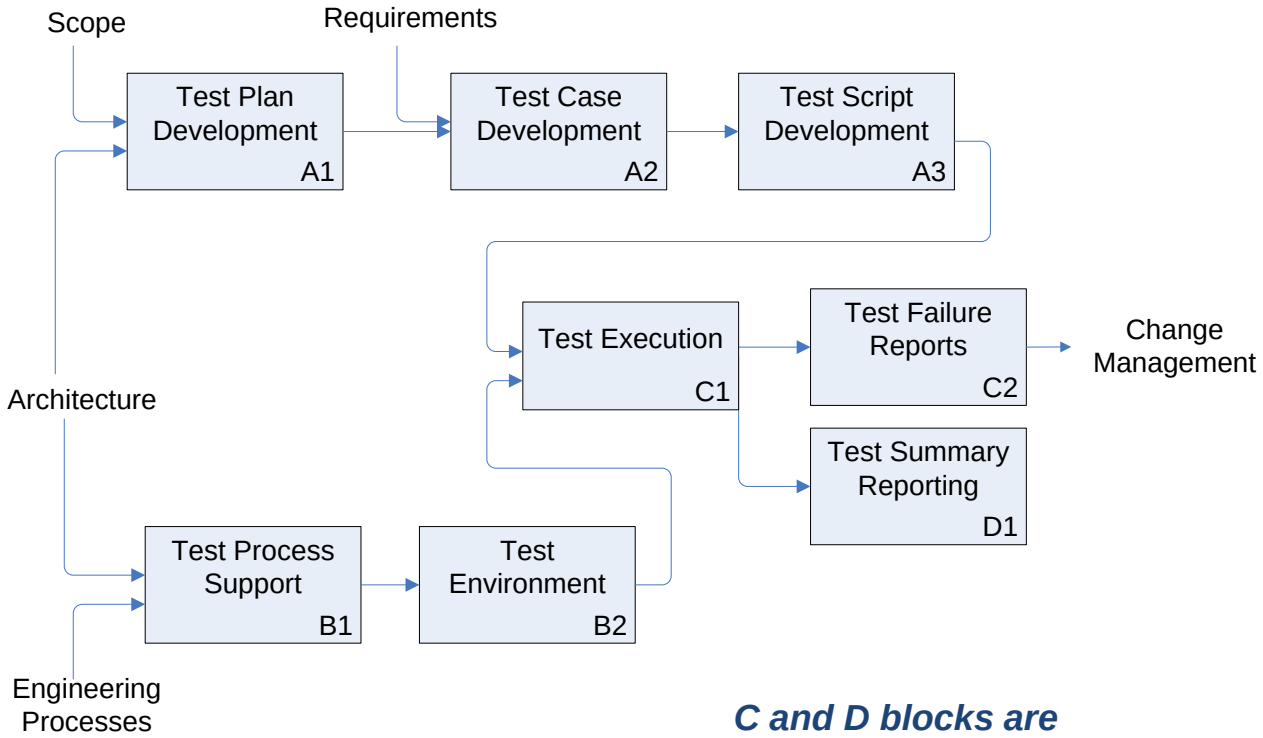
- Cover as many requirements and specifications as possible.
- Document test failures clearly and accurately
- Don’t create “bad test cases”
- Maximize testing throughput (complete the job quickly).

What would you measure?

Processes Associated to Testing

A. Development of test cases
B. Process updates to accommodate technology changes
   - “How might this change affect how we should test?”
   - For verification? For validation?
C. Test execution and failure analysis
D. Summary reporting
   - Functions passed
   - Performance goals achieved
   - Defect density vs. target or other forecast

Project Testing Processes



C and D blocks are measurement opportunity

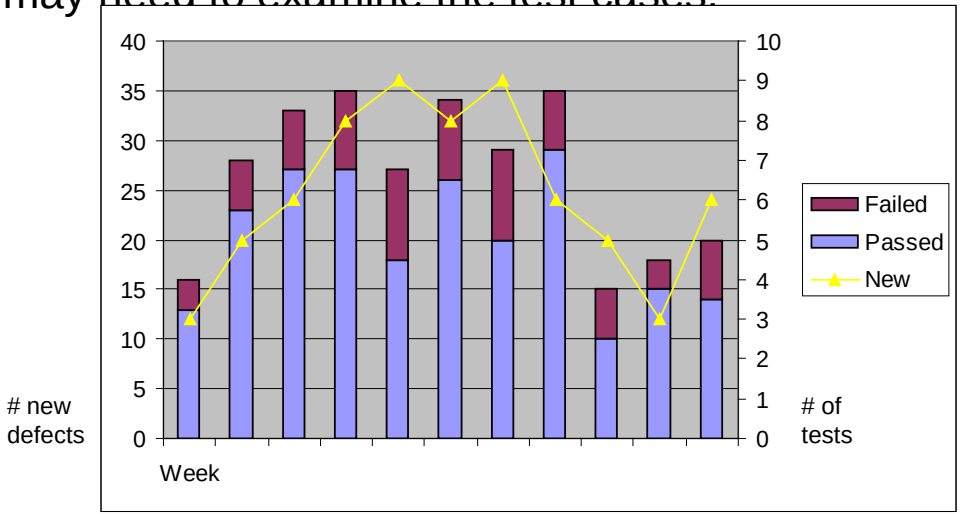
Maximize Testing Throughput

Are we executing as many tests per week as we planned?

- Since “size” of tests may differ, may measure %plan.

Are tests finding defects?

- If not, we may need to examine the test cases.



“New” – defects discovered this week; plotted with right-hand axis.

Inspection Process Example

An inspection process is a cost effective defect identification process.

- How long does it take? Pages/hour of effort
- Is it effective; are we finding the problems? Major defects/page or per hour
- Are resulting “major” defects closed?
  - “Major” means it would fail some test case.
  - “Minor” means it does not meet a documentation standard or similar.

A typical process will have “normal rates”.

- Rates outside the norm should be questioned.

Process Effectiveness Dashboard

Only a few processes can be watched closely.

Meetings are important because of action items and decisions.

Some number of quality checks is important to a good process.

Process throughput is important.

- Measures of throughput for defect removal and testing are valuable because of schedule slip at a critical point in the project.

Process “handoffs” are important.

- Delays between teams and complaints about input quality also create delays.

It is not the project manager’s job to fix the process.

It is important to know when the process has broken. There’s too much risk.

SSD- Dashboard

Risk Management

Risk and the Dashboard

Risk management crosses all boundaries.

- Make a choice if you want various risks displayed within various quadrants or consolidated in the Process and Risk Quadrant.

Critical risks can cause project cancellation.

- Project overrun of schedule or cost >200% is likely to cause cancellation
- Failure to achieve desired KPP – product is not sufficiently better than current product

If we say that project overrun >10% is "**Red**" then people will say ….
"We will monitor this risk", but never actually do anything about it.

Consequently risk management appears to be present but actually has no direct effect on project management.

Risk Management

Goals

- Identify risks and prioritize them
- Mitigate or avoid critical risks

Questions

- Which assumptions do we need to test during the project?

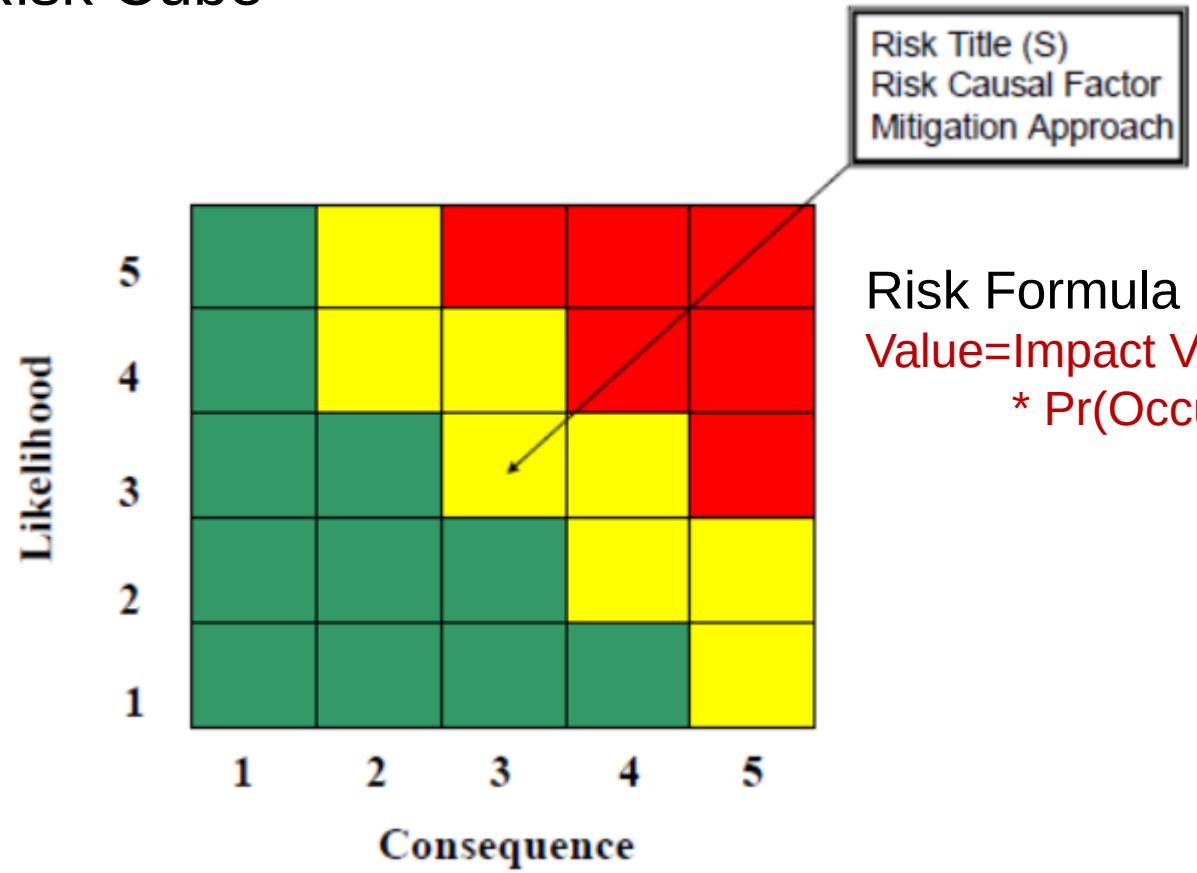
Indicators and Measures

- Risk cubes
- Risk burn-down

Decisions Informed

- Which risks are worth spending to mitigate or avoid?
- Which risks no longer are a potential problem?

Risk Cube



Risk Formula

Value=Impact Value

* Pr(Occurrence)

Risk Management

Many risks are associated to basic assumptions made during planning.

- We have the right technology.
- We picked a contractor that we trust to do the job.
  - The contractor's basis of estimate was derived from real productivity numbers.
  - The contractor has the right staff and facilities to do the job.
- We can achieve the technical performance goals.
- The supply chain will be able to provide parts with the new technology
- Our customers' mission requirements are stable.

Managing risks is about mitigation (reduce effect) and avoidance.

Both strategies will increase costs.

- Managing risk is not free but it is effective.

Risk Rating on a Geometric Scale

Severity or Impact Scale

- 5 is project cancellation.
  - Cost or Schedule is 2-3 times greater than budget, **OR**
  - Product is not sufficiently better than current product (no customers)
- 4 Impact value is 33-90% of cancellation (a potential breach)
  - Either project overrun is ~50% or half the customers leave, or half the mission value can be achieved.
- 3 Impact value is 15-33%
  - Within a single breach
- 2 Impact value is 7-14%
  - Risk is readily managed within contingency
- 1 Impact value is <7%
  - Several such risks can be handled within contingency and by the team.

Frequency or Probability Rating of Risk

5 Virtually always: 90-100%

- No one escapes. For example trying to mature a new technology from TRL 4 to 7 within a year.

4 Likely: 70-90%

- More often than not. Worth generating a real mitigation plan.

3 Equally likely as not 30-70%

2 Unlikely: 10-30%

- It happens often enough to be concerned.

1 Rare: 0-10%

- It could happen, but I’ve never seen it before

Risk Display: Cubes + Summary

Construct a summary of all risks by

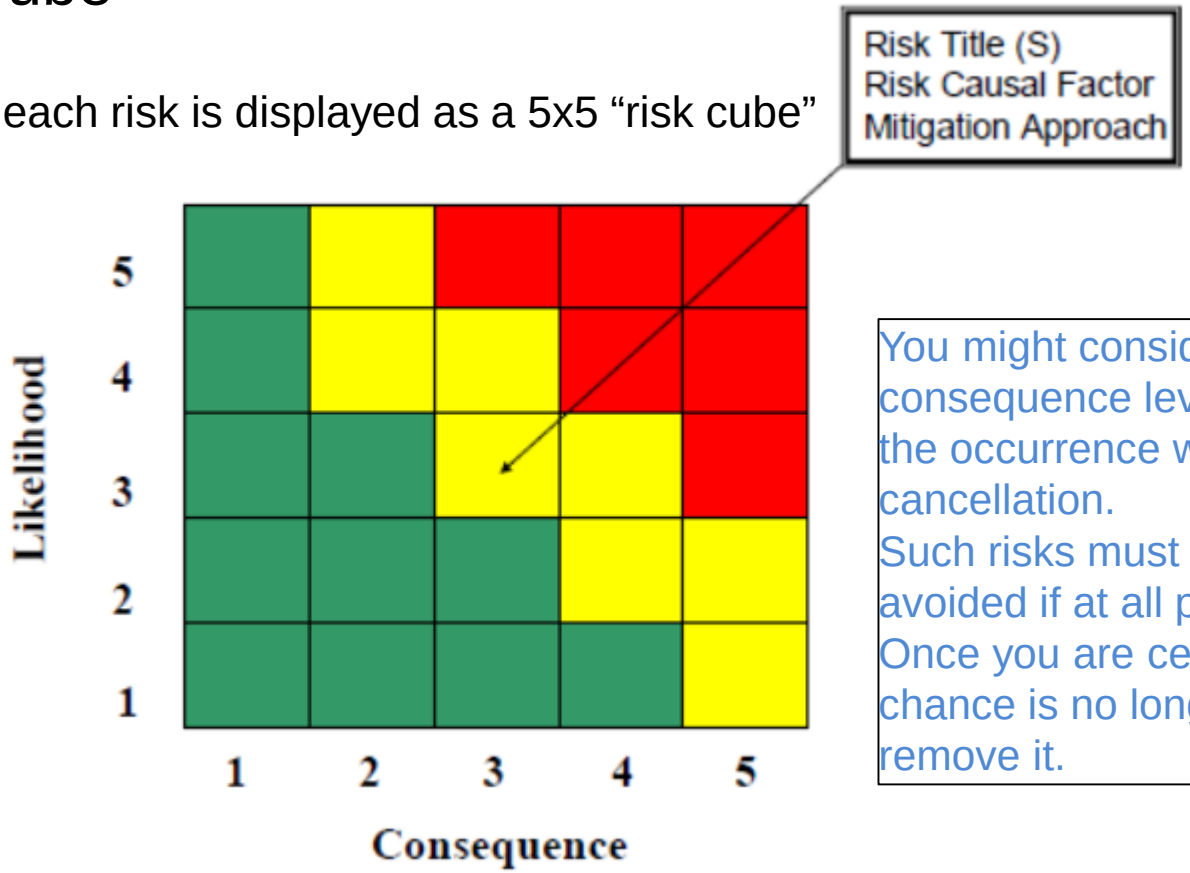
- Each risk-value = (3^Criticality-1)*Frequency *(3^4)*0.9 = is the highest risk*
- Add risk-values together
- Plot the total
  - If total risk is small, (less than 300-400), you probably have not identified all the risks.
  - Use this chart to show risk burndown over time
  - By milestone C total risk should be less than 100

New product development in DOD has very high risk

- 2011 report: Over $32B spent on abandoned projects since 1995
- None of the services is safe
- It requires a lot of energy to reduce the risk to manageable levels

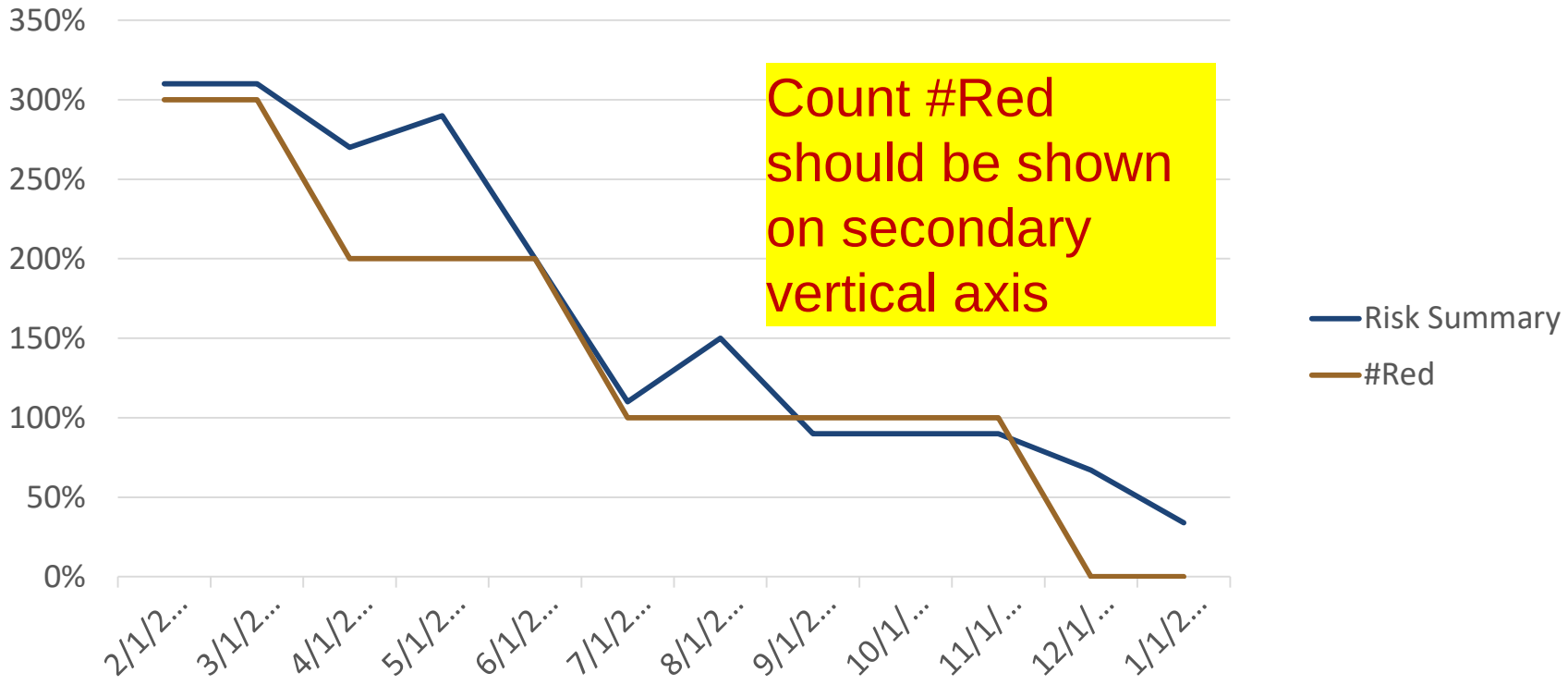
Risk Cube

Typically each risk is displayed as a 5x5 “risk cube”



You might consider that any risk of consequence level 5 is “red” since the occurrence will likely result in cancellation.
Such risks must be mitigated or avoided if at all possible.
Once you are certain that a 10% chance is no longer true, then remove it.

Risk Burndown Chart



Summary

Risk can affect any aspect of a development project.

- At the start of development, the sum of the risks may be many times greater than the project value.
- A project showing little risk has not been properly evaluated.
- Looking deeper should identify several untested assumptions.

Risk mitigation

- Find ways to validate the assumptions early.
  - Prototyping, check the supply chain
  - Check contractor's claims for productivity and quality ….

Risk avoidance

- Pick alternatives if the risk is too great and has a high likelihood value.

Watching burndown helps everybody on the program.

SSD- Dashboard

Goal-Driven Measurement

Developing New Indicators

Goal-Driven Measurement: Indicator Types

Success or Outcomes

- Show that improvement efforts or product or service has the desired impact from the stakeholder's viewpoint.
- Product achieves revenue or market share
- Improvement reduces cost, rework, cycle-time

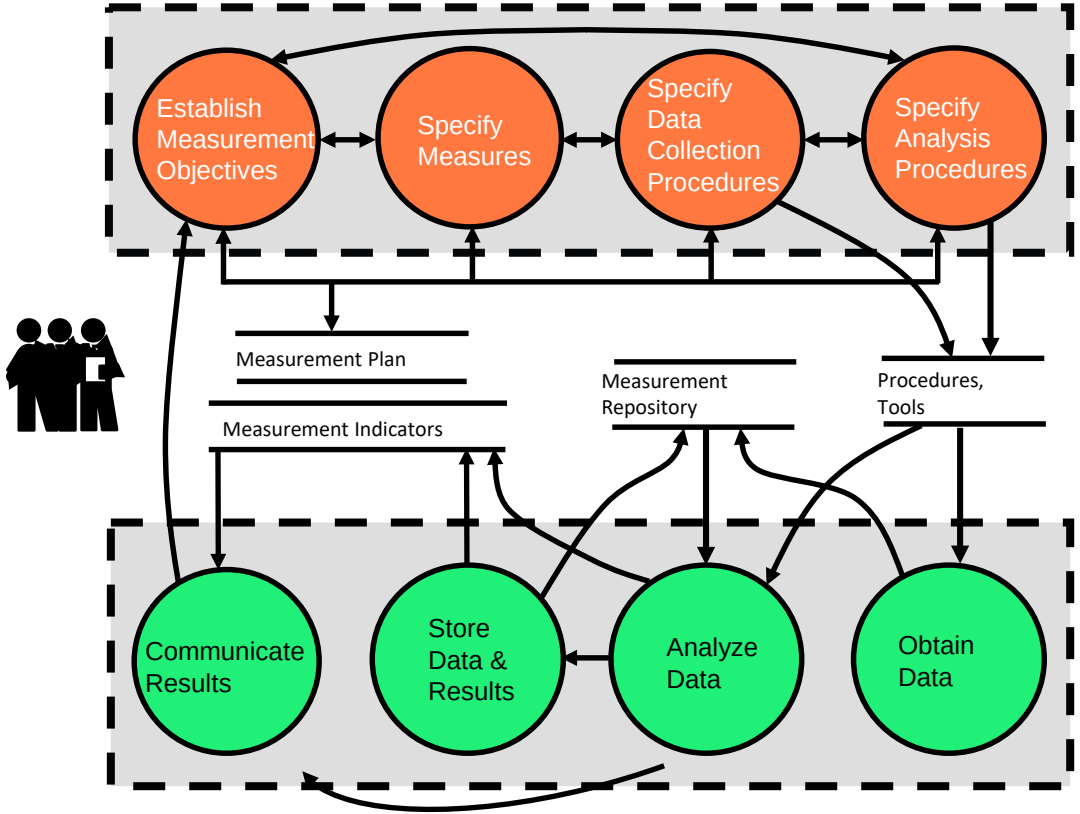
Progress

- Tasks are performed and deliverables produced
- Code growth or test cases passed
- Used to prove plan is viable and direct work efforts.

Analysis

- Resources are properly allocated
- Open vs. closed defects
- Used for problem identification and problem solving.

CMMI Measurement Activities



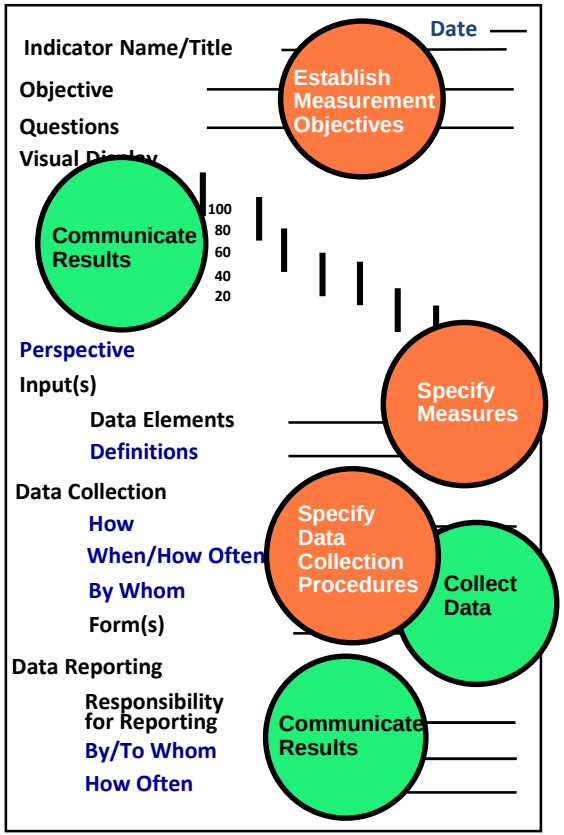
Goal 1

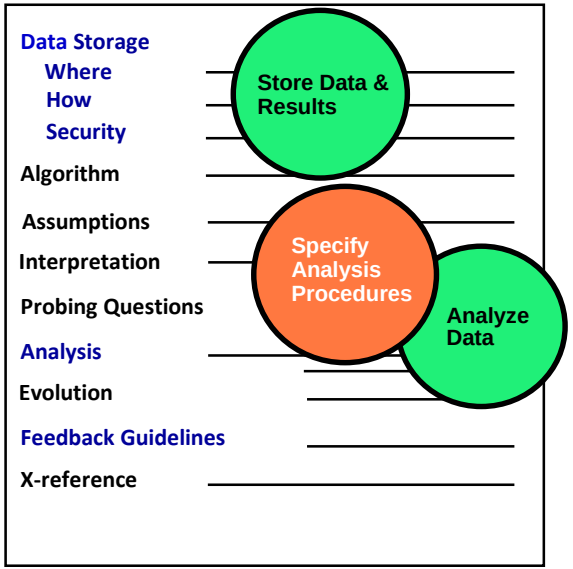
Align Measurement Activities

Goal 2

Provide Results

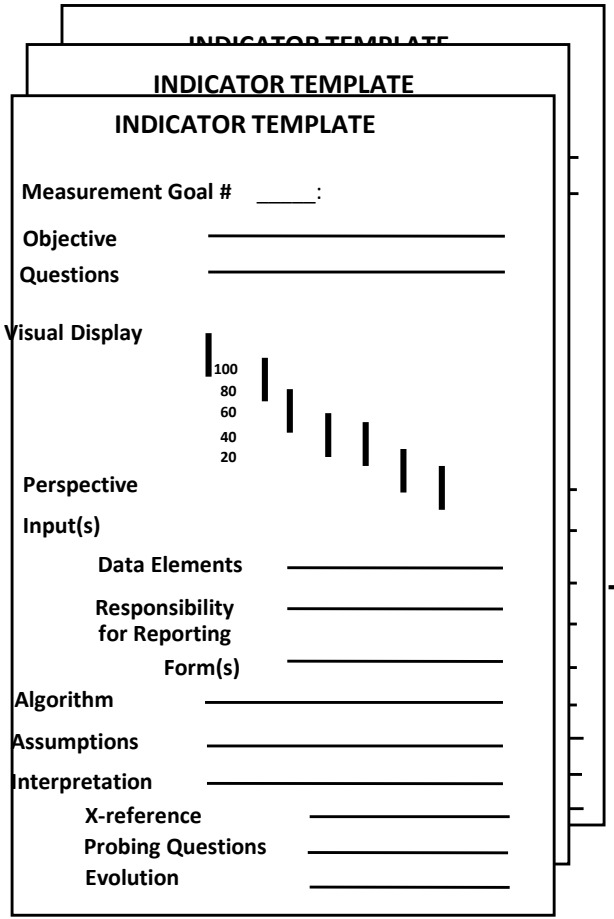
Measurement Practices Mapped to the Indicator Template



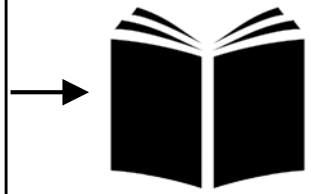


Indicator Template

Documents the who, what, when, where, why, and how.



Completed indicator templates are the output of GQ(I)M work.



Measurement Handbook

Some Example Questions

Productivity is important when input size and sources are variable.

- Quality checks on input and making estimates are critical to commitment dates.

Feedback and control (management communications)

- Important when team agility is a factor. Fighting real fires (not a metaphor).

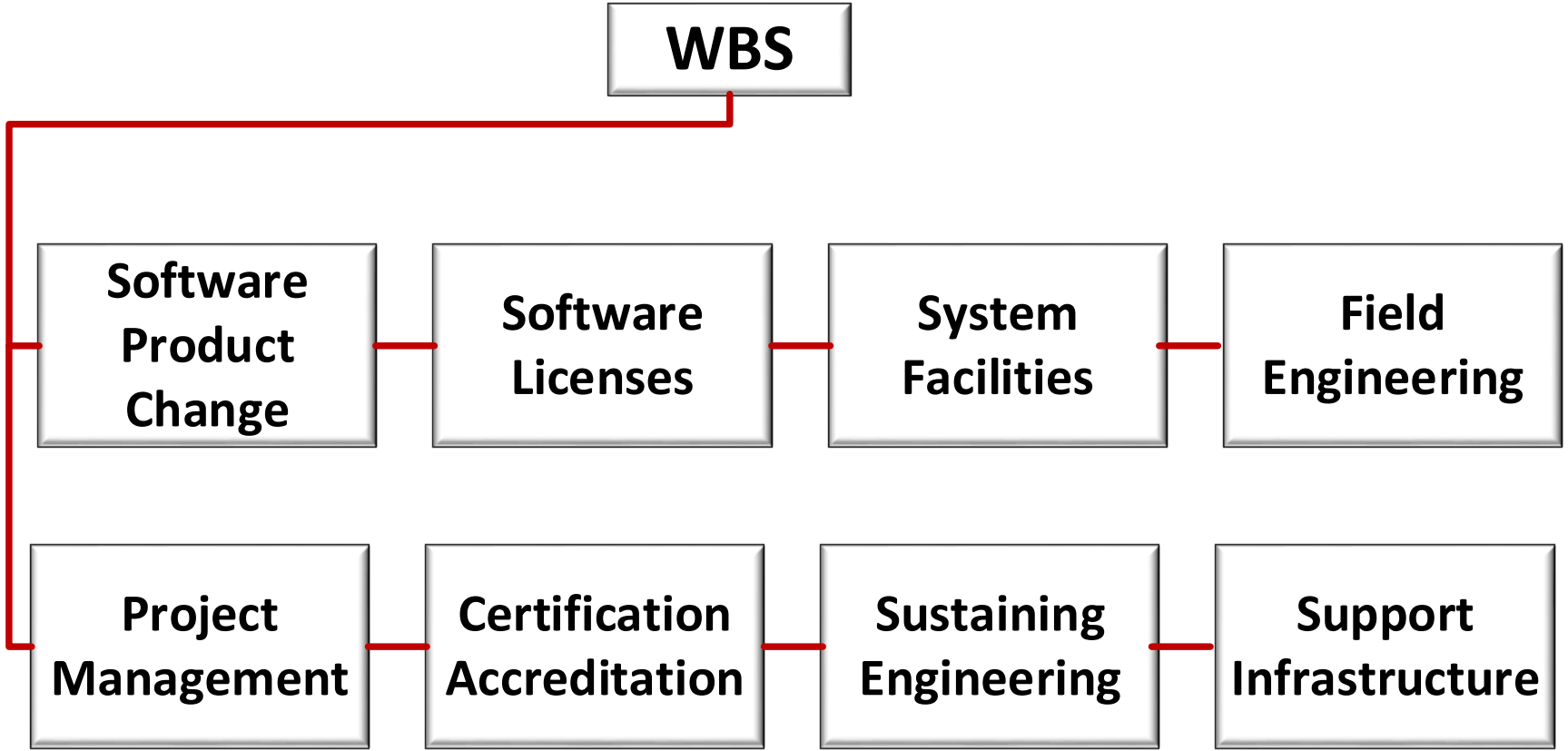
Resources and analytic tools

- Important when technology changes or complexity of problem changes.
- Results may cause estimates to change.
- Also important when redefining process such as attempts to achieve higher quality.

SSD- Dashboard

Sustainment Indicators

Software Sustainment Lifecycle



Software Maintenance WBS

1.0 Software Change Product - products and activities associated with defining, allocating, generating, integrating, and testing software changes for an operational software product or system

2.0 System Project & Technical Management - products and activities associated with system specific software maintenance project and technical management

3.0 Software Licenses - products and activities associated with the procurement and renewal of software licenses for operational software

4.0 Certification and Accreditation - products and activities associated with verifying a software system against externally defined domain performance criteria

5.0 System Facilities - products and activities associated with establishing and operating software maintenance related development, integration, and test facilities, and support equipment and tools

6.0 Sustaining Engineering - products and activities associated with system specific test, delivery, and training support

7.0 Field Software Engineering - products and activities associated with the on-site support of a deployed software product or system in its operational environment

8.0 Support Infrastructure - products and activities associated with establishing and operating the organizational infrastructure required to implement common software maintenance business and technical processes across multiple software systems

Software Product Changes

One problem in changing software is deciding how to allocate effort between:

- Fixing problems and
- Creating additional functionality and
- Improving product performance

The problem is made more difficult over time when product quality is poor.

- A 30% defect escape rate results in 90% of effort goes to fixing problems after about three successive releases.
- A high quality process is important.

Prioritizing Defect Fixes

"Defect Severity" does not provide sufficient information to prioritize fix order.

- Discussion becomes politicized (my severity is more important than yours).
- What is the desired outcome of the fixes?
- After fixes are performed is there a reasonable measure of "better than before"?

If we agree that improving **system availability** is a good goal, we can define a measure.

"Risk Priority Number"

- There is no good measure of "Software Reliability" since we have no measure of "unit usage".
- Reliability depends on monitoring the usage of unit such as "a tractor" or "an engine".

We do not need an absolute measure of system availability to demonstrate that we have "improved system availability".

Defining Risk Priority Number (RPN)

RPN can be used to prioritize defect repair since it measures product availability.

What is RPN?

- Defect is evaluated for severity: 1-5 (5 is high)
  - 5 = Product is unavailable as in system crash
  - 4 = Function doesn't work e.g. menu is inaccessible
  - 3 = Wrong answer but work around exists
  - 2 = Misleading, requires user to back up and take another path. Can be fixed in training.
  - 1 = Cosmetic, result looks like another type at first glance – wrong color, bad placement on display.
- Frequency of occurrence (every time, once a day, once a week, once a month, rare)
- Difficulty of identification 1=immediately apparent, 5=near impossible
  - Cyber events are difficult to detect but may have severe consequences.

RPN=Product of all 3 values (value range 1-125)

Using Risk Priority Number

Selecting fix order

- Calculate RPN for each defect.
- Sort by highest-lowest.

Reporting Value

- Sum up RPN for all defects
- Sum up RPN for current fixes
- Improvement = (Current fixes RPN)/(Sum of all RPN) shows %availability improvement

Method is not theoretically perfect. It is a simplification of a statistical calculation.

- Nevertheless, it does improve decision making
- Improving product availability is important to mission capable availability.

RPN Value

Consider each RPN score as representing a loss of system availability.

- We can then claim that fixing a set of defects reduces loss of availability by a known amount.
- This approach removes some of the politics of defect prioritization.
- Improving product availability is important to mission capable availability.

Method is not theoretically perfect. It is a simplification of a statistical calculation.

- Nevertheless, it does improve decision making

A small number of organizations have utilized this approach or a variant and have had moderate success.

SSD- Dashboard

Wrap Up

Dashboard Top-Level Example

Schedule, Cost & Resources

- G EVMS (program)
- Tasks Completion to Schedule
  - R Late Starts/Completions
  - G Dependencies
  - G Deferred / Accelerated tasks
- G Development Status (HW, SW design, implementation, unit//Integ test)
- N Test Status (SIL, GND, FLT)
- Staffing
  - Y Contractor
  - N Gov't

Scope, Progress & Change

- Requirements Trends
  - R Sizing, Trend
  - G Risk
- Development Trends
  - R Hardware
  - Y Software
  - N SLOC
    - TO Artifacts
- Test Trends
  - G TOIs
  - R Trends – SIL, Ground, Flight
- R CDRL Completion
- G Engineering Changes (# by source)

Process & Risk

- G CDRL Comments
- R Action Item Aging
- G Req Traceability
- N *Req Volatility (future)*
- N *SLOC (future)*
- N *TO Comments (future)*
- N *HW TOIs (future)*
- N *QA Reports(future)*
- R Action Items (Overdue)
- Y Problem/Defect Reports

Quality

- G Requirements Approved
- G TRL's
- G *Problem/Defect Reports: (future)*
  - *SW Defects by Phase & Severity*
  - *SW Defects – by CSCI & Severity*
- N TPMs*
  - N report all exceptions
- G Verification Dependencies (by req/test point)
- G Verification Distribution (by CSCI/CSU, HWCI)

The Dashboard

At the top level

- Green is within plan
- Yellow is eating into contingency or potential risk
- Red means the plan is not working.

Divide dashboard into different types of decisions

- Top level stoplights highlight areas of concern
- Each concern is addressed by one or more charts with data

Develop indicators

- Each indicator has an “owner,” someone knows how the data was obtained.

Problem Diagnosis

The diagram shows some possible causal relationships.

A program experiencing cost and schedule overrun,

- May have a problem with scope growth,
- Technical adequacy (TRL, achieving TPM),
- Or development performance

Scope growth would be reflected in requirements, or changing mission.

Development performance can be diagnosed by comparing productivity to estimated performance.
(ratio =Cost/Requirement)

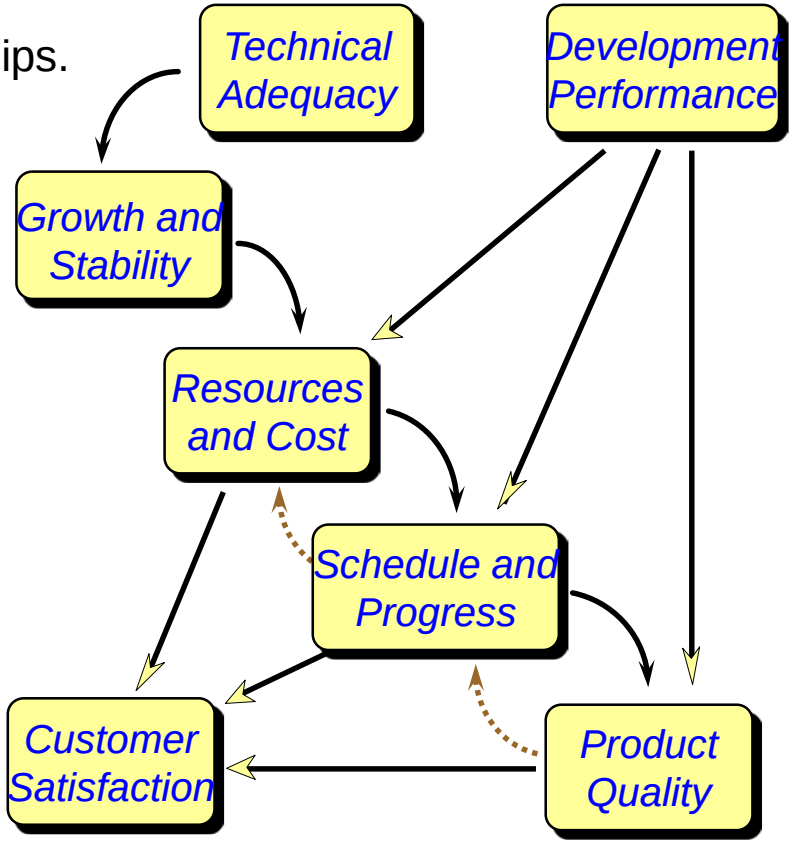
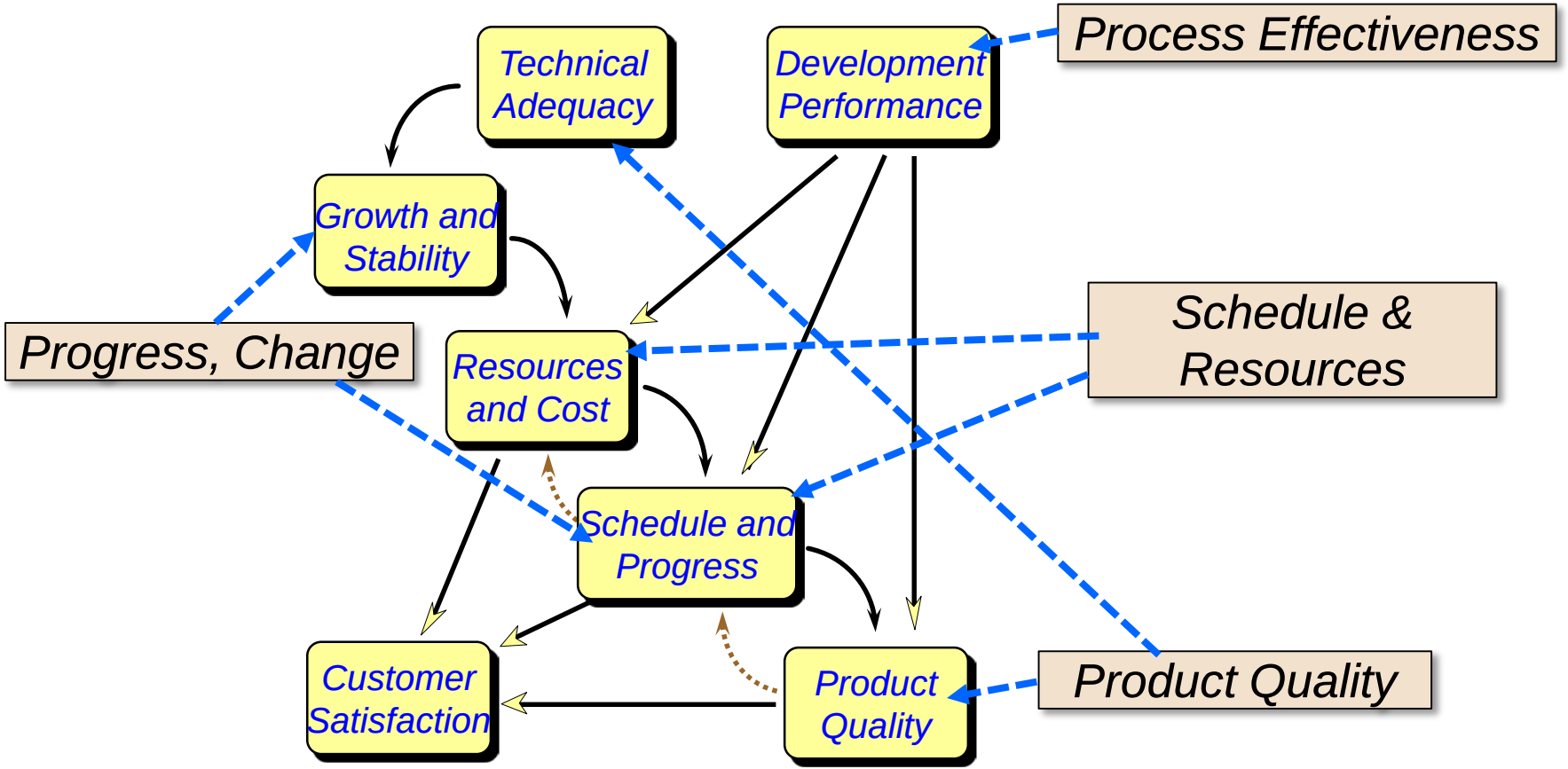


*Diagram from PSM Guide 4.0, Section 4, www.psmsc.com

Dashboard-to-Problem Diagnosis



Diagnosing Problems

Expect to make comparisons

To plans and estimates.

- The estimate should contain a great deal of information about expectations.
- Does the work progress match expectations?

Scope change

- Have requirements and mission been stable? Mission determines acceptance criteria.
- Has customer representative changed? Are there additional customers?

Development performance

- Resource availability or wrong resources
- New technology learning curve
  - Too much rework has same effect as resource shortage

Wrap Up –1

A project dashboard is a useful mechanism for communicating top level status.

Be prepared with forecasts.

If there are problems, be prepared with problem diagnosis:

- What is the evidence of the problem? How big is it (as a risk for example)?
- What do we think is a likely cause?
- What are some alternative responses?
- Test the responses and make a recommendation.

Wrap Up -- 2

Implementation can be a challenge

Contractor resistance

- Use contractor's data and charts first.
  - Don't force your own charts. Don't expect perfection at the start.
- If there is an empty quadrant, ask questions.
  - A contractor with a mature process knows these things. If he doesn't, then you should suspect future problems.

Reduce primary concerns to a stoplight.

- Charts are required to explain the concern.
- Remember: "Red" means change is required – not optional for the correct audience

PMO needs its own process measures.

- *"Eat your own dogfood"*

Contact Information

Presenter / Point of Contact match to Information Sheets

Name: Dave Zubrow
Title: Assoc. Director
Telephone: 412-268-5243
Email: dz@sei.cmu.edu

Name: Tom Merendino
Title: Sr. Engineer
Telephone: 412-268-1154
Email: tjm@sei.cmu.edu

Name: Bob Ferguson
Title: Sr. Engineer
Telephone: 412-512-9585
Email: rwf@sei.cmu.edu