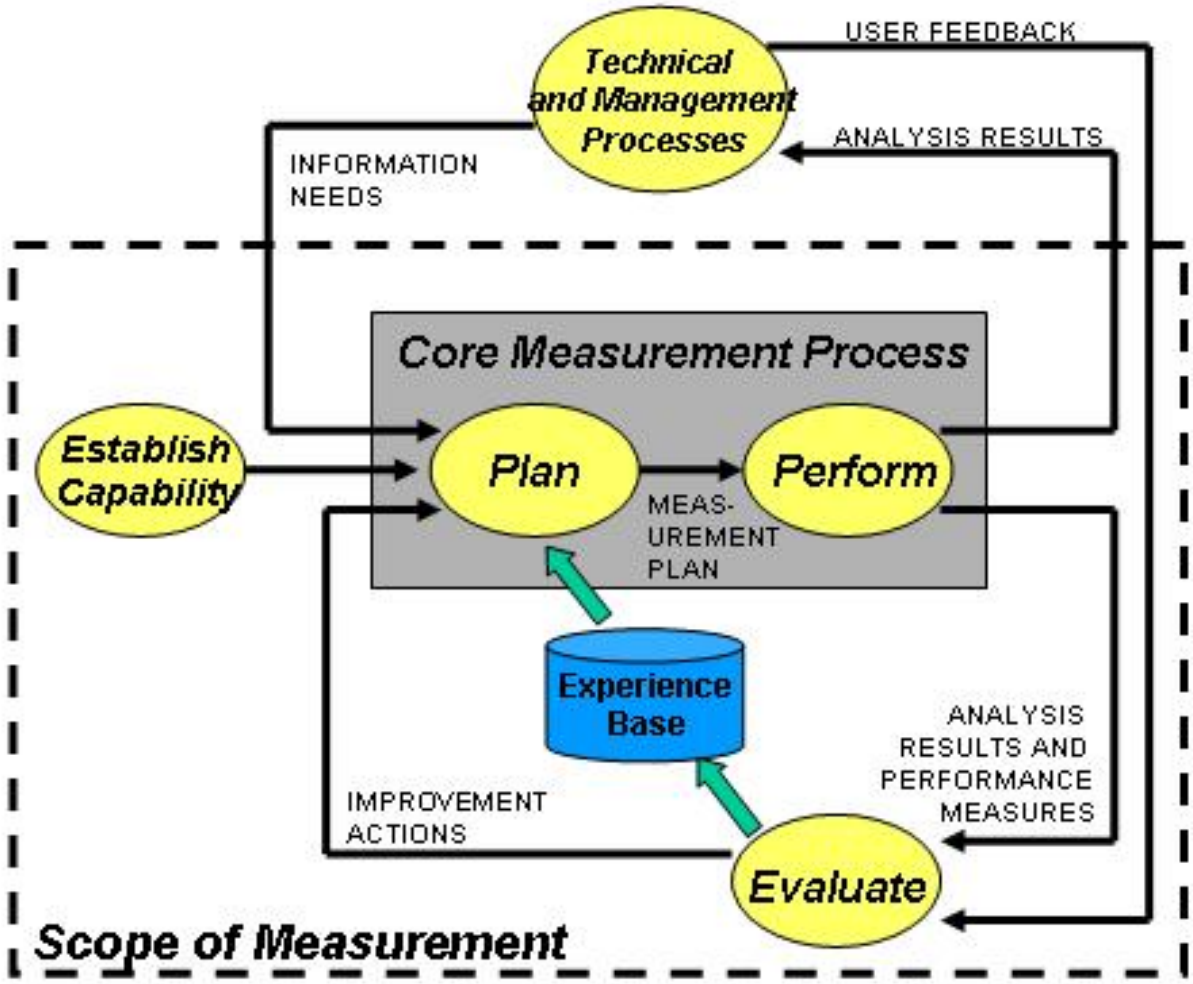
Putting "Management" into Requirements Management

Chesapeake Chapter INCOSE

February 2010

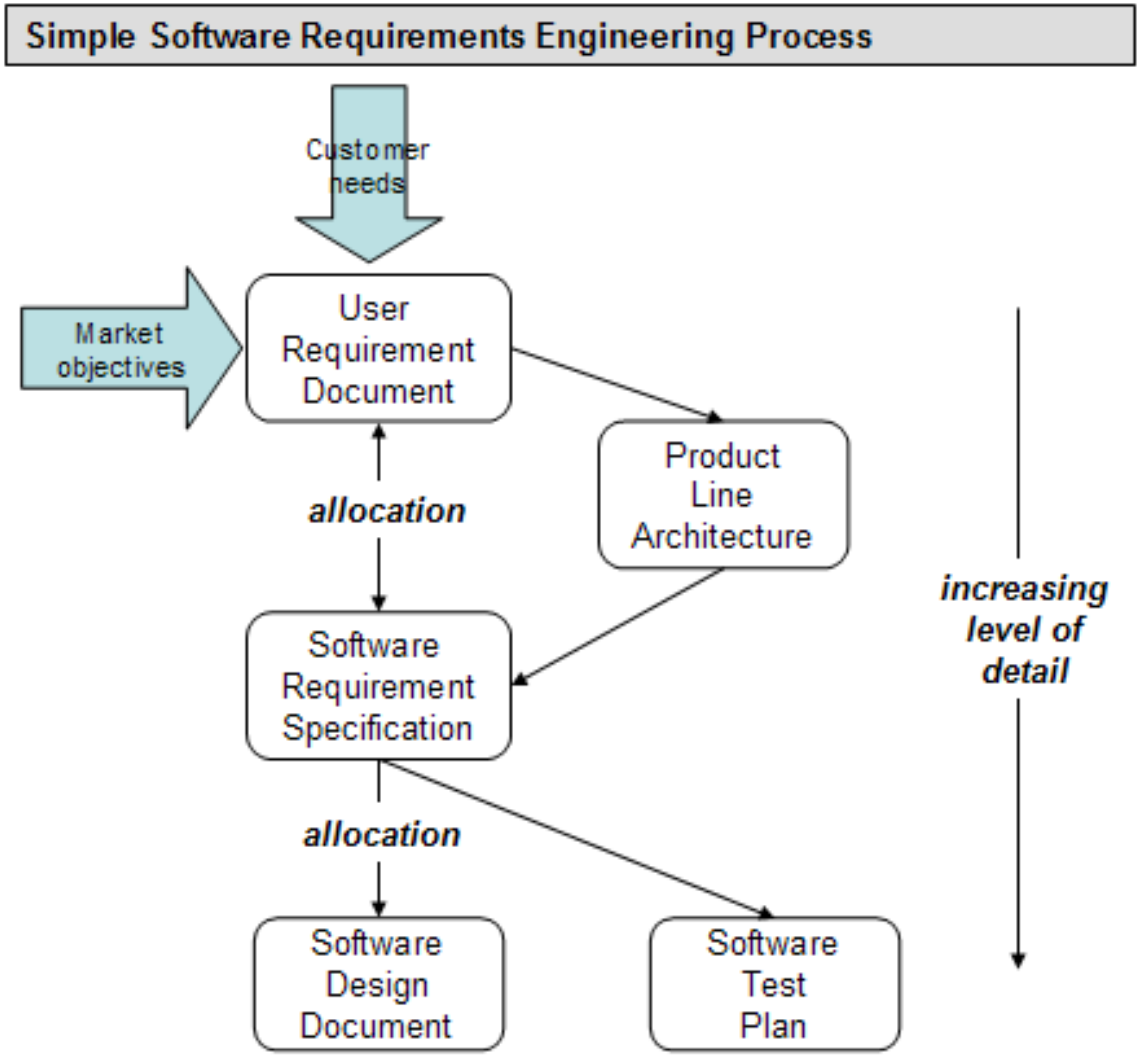
OVERVIEW OF MEASUREMENT

MEASUREMENT PROCESS

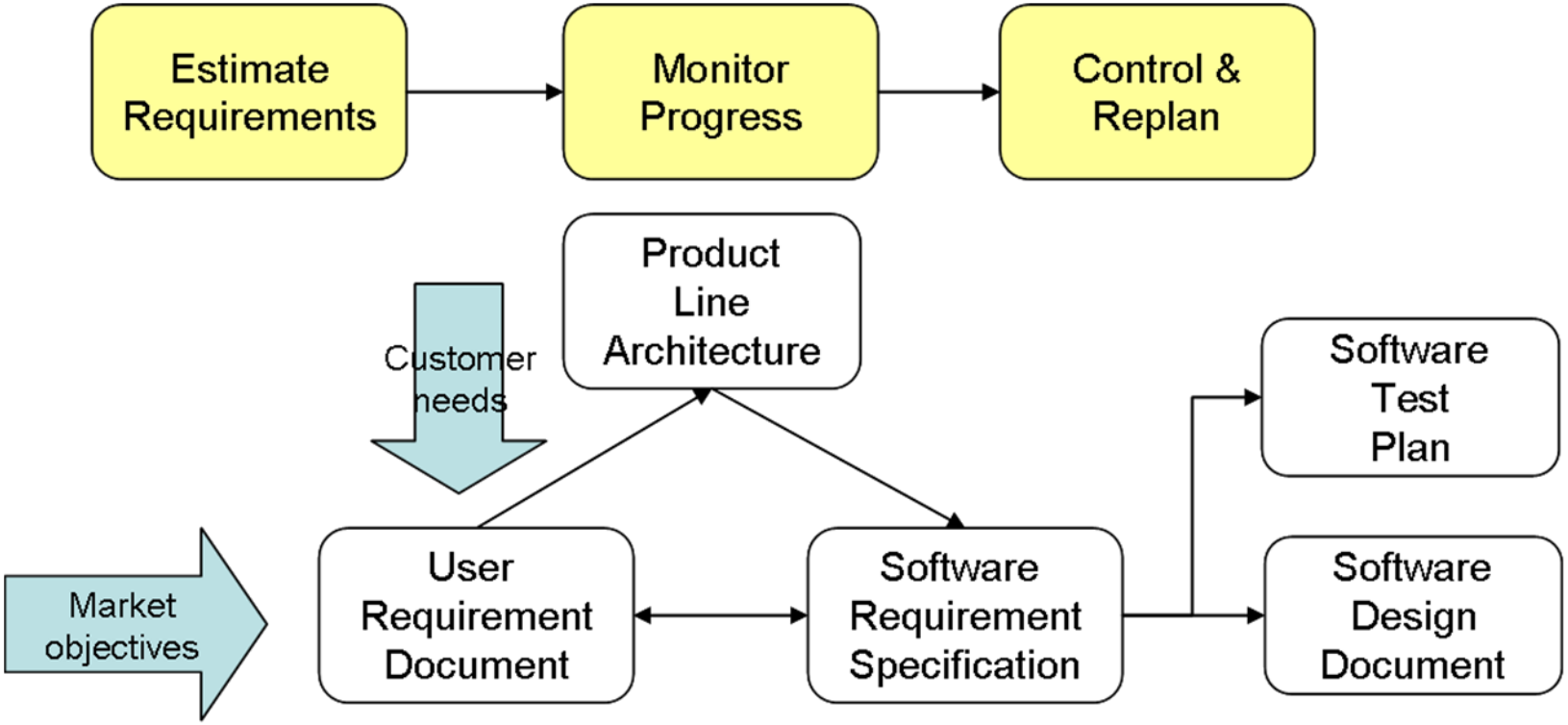


REQUIREMENTS MANAGEMENT

TYPICAL SPEC TREE



MANAGING REQUIREMENTS ENG.

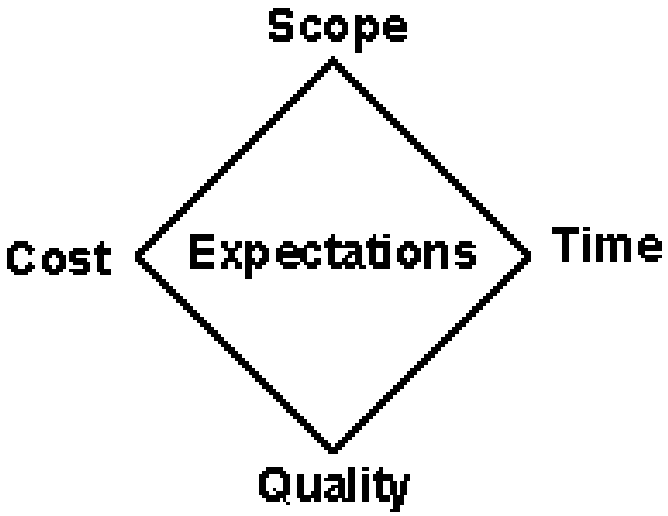


WHAT GOES WRONG?

Software Projects That Failed Due to Poor Requirements Management
In a July 2005 IEEE article entitled “Why Software Fails – We Waste Billions Of Dollars Each Year on Entirely Preventable Mistakes”, Robert Charette lists “Badly Defined System Requirements” as one of the primary causes of software project failure. He estimates that software failures have cost the US economy as much as $75 billion dollars over the past five years.
In 1995, a Government Accounting Office report entitled “Radar Availability Requirements Not Being Met” document the requirement failures of a project jointly developed by the U.S. Air Force, the Federal Aviation Administration and the National Weather Service.
In 1994, the Standish Group released The Chaos Study which cited “Incomplete requirements” as the number one impairment factor in failed projects. The number six factor is “Changing Requirements and Specifications”.
In the 1993 article entitled “Analyzing Software Requirements Errors in Safety-Critical, Embedded Systems” by Robyn R. Lutz of Jet Propulsion Laboratory, the root cause of 62% of the errors in safety-critical software was identified to be poor requirements.
In 1998, Robert Glass published the book, “Software Runaways: Lessons Learned from Massive Software Project Failures”. The first reason cited reason for project failure is “Project Objectives Not Fully Specified.”

PROBLEM OF SCOPE

- Changing scope affects all other major performance drivers
- Downstream activities could easily triple due to "fan out" (aka expansion)
- Validate changes against your initial assumptions



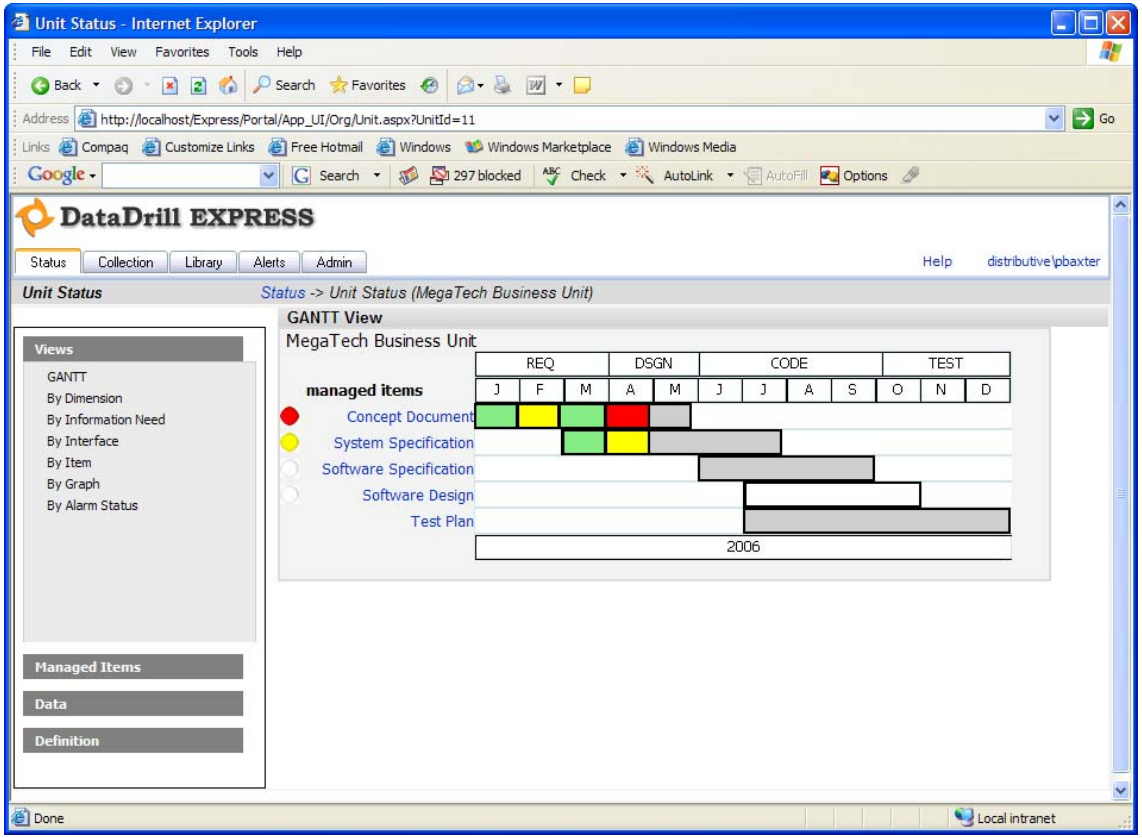
MANAGEMENT TECHNIQUES

MANAGE BY …

- Establish a method for estimation
- For each project create a target for each work product
- Monitor each work product progress
- Take action to control the process:
  - Reallocate resources
  - Modify the initial target
  - Rescope the plan / schedule
- Capture actual performance for each work product – review the estimation method

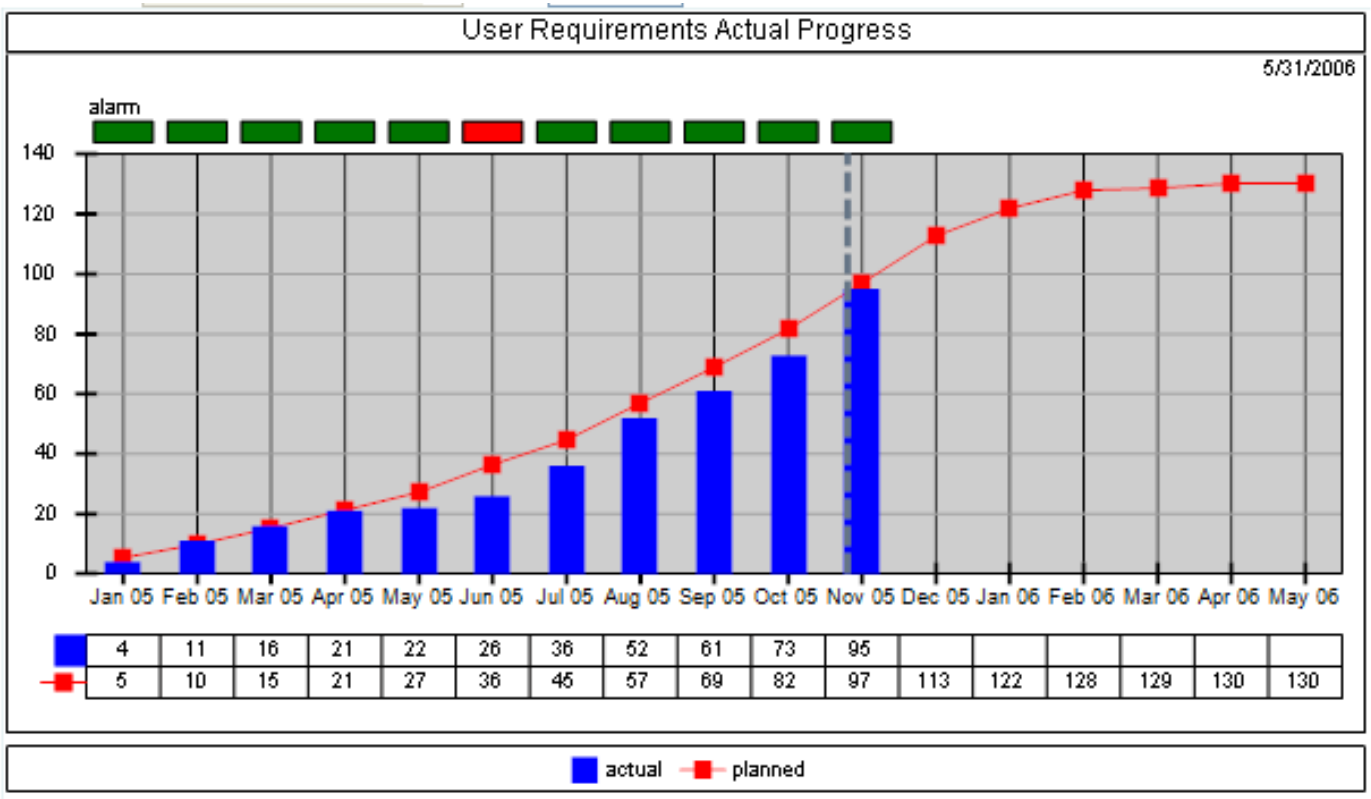
UNDERSTAND THE WORK PLAN

Requirements docs are generated throughout the life-cycle. Know when to manage them.



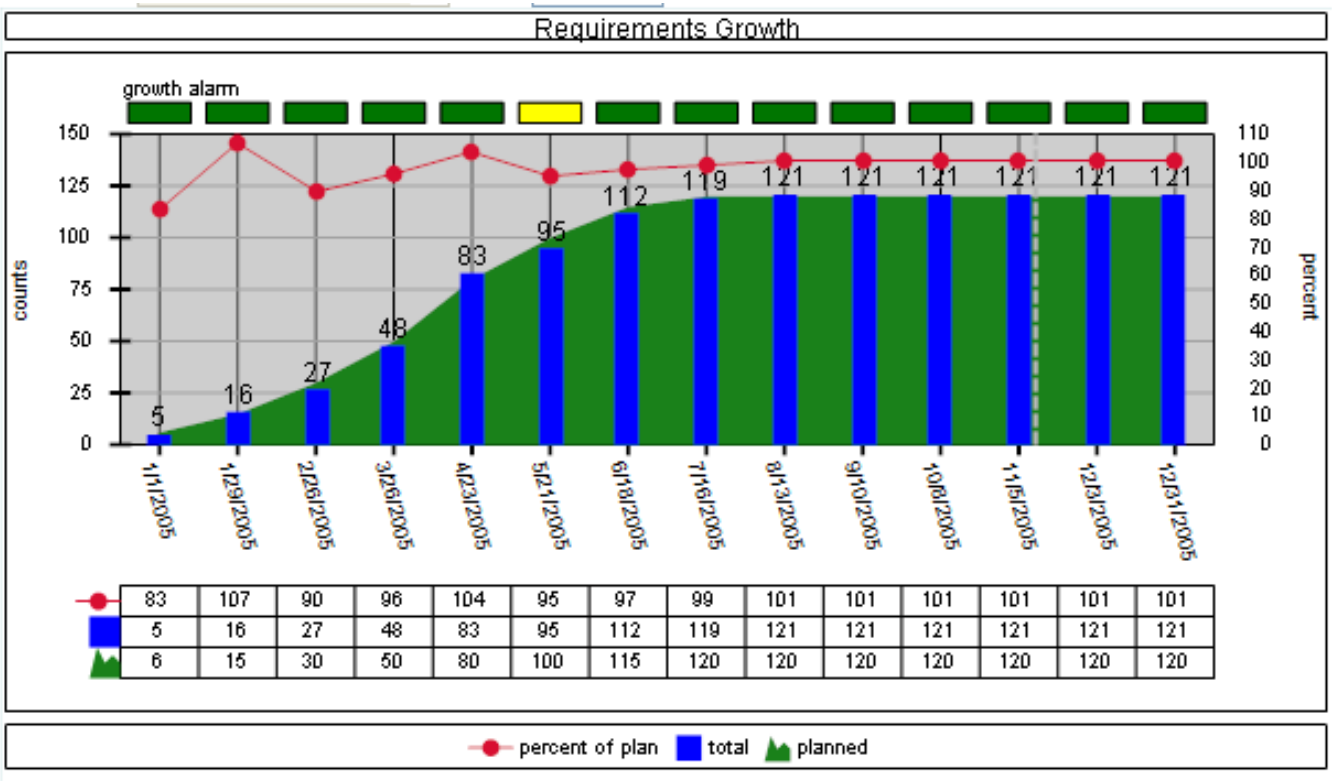
MANAGE USER EXPECTATIONS

Make sure you have visibility into what the user has asked for = no surprises.



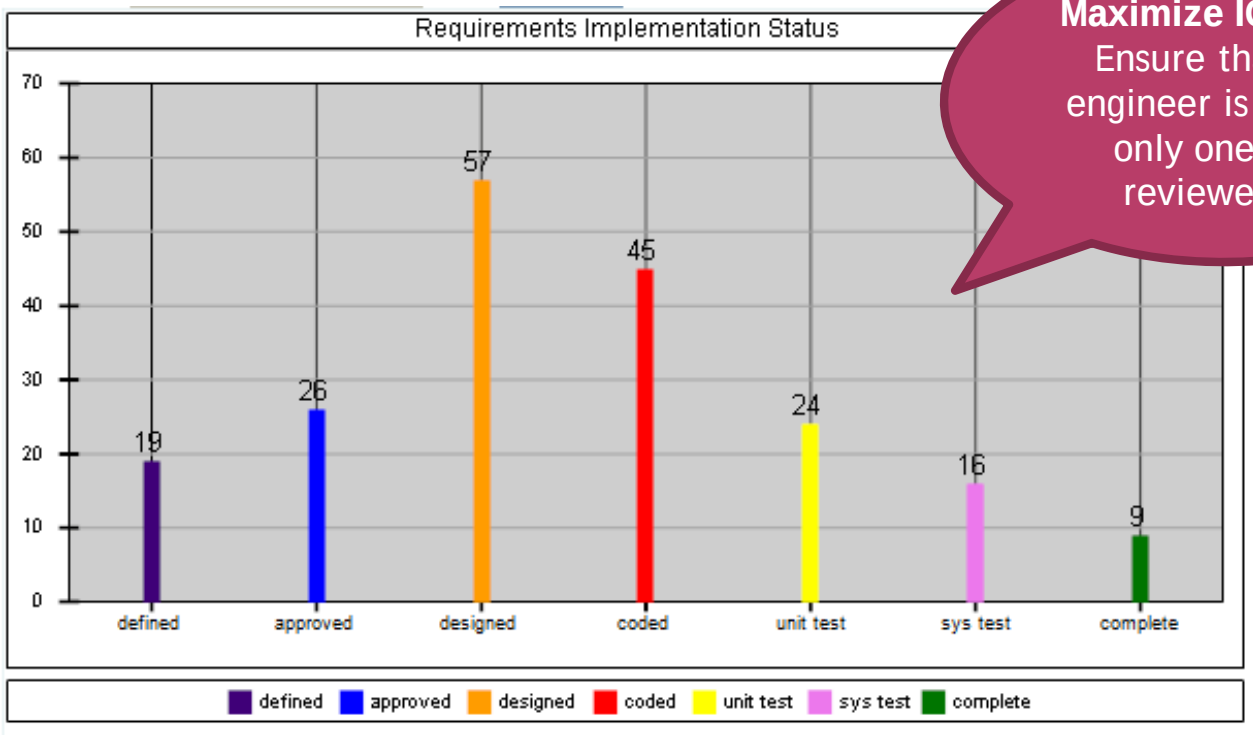
CONTROL CREEP

Keep track of how close the “actuals” match the “plan”.



MONITOR CRB/CHANGE STATUS

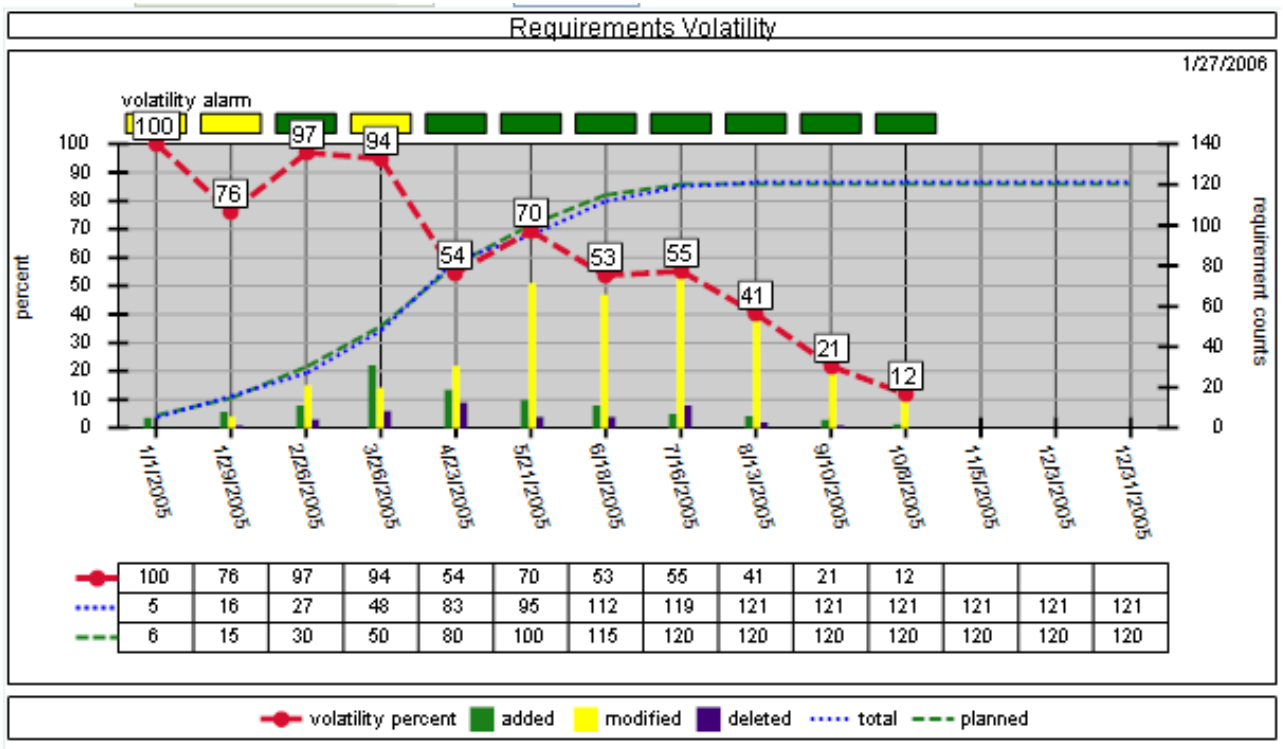
Requirements have a CM status, evidence that more than one person has looked at it. Ensure that all requirements move right ➡



Maximize IQ Points
Ensure that the engineer is not the only one who reviewed it.

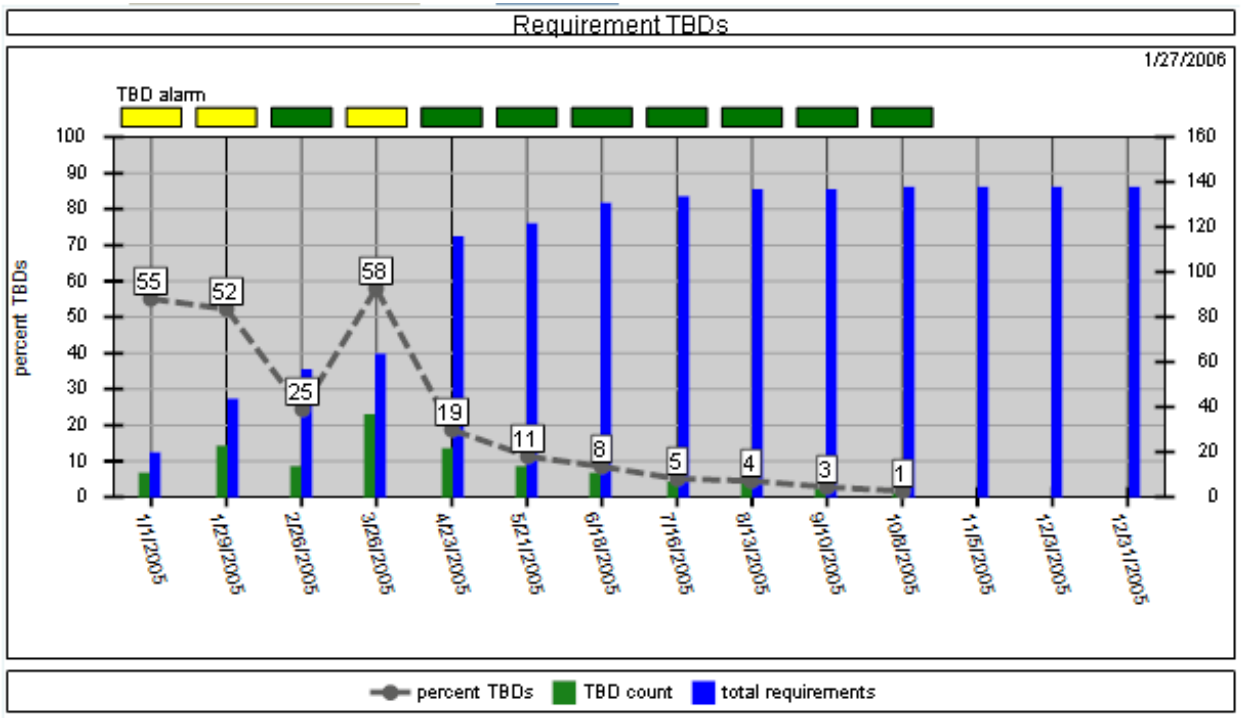
CONTROL VOLATILITY

Monitor and control churn (changes continually made). It must reach “0” before you ship.



MANAGE THE I-O-Us

TBDs are often the most difficult and complex requirements. Make sure the hard ones get finished.



SUMMARY

- Managing requirements is not a difficult technical issue
  1. Create an estimate
  2. Plan work products
  3. Monitor progress
- The techniques are simple and easy to implement
- So why do only a few companies do it?
  - If you don't ship it, it must not be worth the customer's money
  - It's not code
  - You never tried
  - Blah, blah, blah

ADDITIONAL RESOURCES

Requirements Management Guidance on the Web	
Software Engineering Institute	www.sei.cmu.edu
Distributive Management	www.distributive.com/resources
Crosstalk Magazine from STSC	www.stsc.hill.af.mil/crosstalk
Scott Ambler's Web Site	www.ambysoft.com
Karl Weiger's Web Site	www.processimpact.com

QUESTIONS?

CONTACT

Distributive Management

www.distributive.com
Peter Baxter
pbaxter@distributive.com
800.779.6306