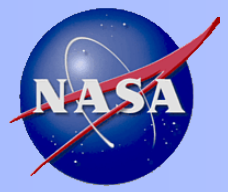


NASA Software Engineering Benchmarking Effort

PM Challenge February 21-23, 2012

OCE Detailee/GSFC/Sally Godfrey

301-286-5706

OCE Detailee/JSC/Heather Rarick
281-244-1013

Agenda

- **Background**
  - **Why did we benchmark?**
  - **Who did we benchmark?**
  - **Who was on the Benchmarking Team?**
- **What did we ask?**
- **What have we learned?**
  - **Observations**
  - **Recommendations for NASA**
- **Summary/ Unexpected Benefits of Benchmarking**

Background: Why and Who?

- **Identify best practices that will**
  - **improve the management and engineering of software intensive systems**
  - **enhance software collaboration among centers, Prog/Projects, international partners and external relationships**
  - **provide guidance or solve current NASA software issues**

- **Benchmarked 18 Organizational Groups:**
  - **Within NASA (5 of the 10 Centers were included)**
  - **NASA Industry Partners (5 Aerospace/Defense Contractors)**
  - **Government Agencies (4 groups from Army, Navy, Air Force)**
  - **NASA Academic Partners (4 Universities, University labs who do Aerospace work)**

Background: Team, When?

- **Team: At least 3 members/visit –Included Software Assurance**
  - **Heather Rarick (OCE Lead)**
  - **Sally Godfrey (Co-Lead)**
  - **John Kelly, Tim Crumbley, Kevin Carmichael (OCE reps)**
  - **SW assurance people: Cindy Naiman, Cynthia Calhoun, Joel Wilf, Martha Wetherholt, Cyrus Chow, Renee Hugger, Rosalynne Strickland, Susan Sekira**
  - **When near a Center, we included a local person: Scott Morgan, Liz Strassner, Laura Maynard-Nelson, Bill Van Dalsem, Leann Thomas, Pat Benson, Helen Housch**

- **Schedule and Status:**
  - **First interviews started in Feb. 2011; last one was in Nov. 2011**
  - **Are in process of summarizing all results/recommendations for final report**

Benchmarking: What Did We Ask?

- **Background: to understand the organization**

  Org structure, types, sizes of software, criticality, SW relation to SA, languages, life cycle, major projects

- **Training**

  Responsible parties, plans, strategy, preferred method, best classes, mentoring, mandatory or not

- **Acquisition**

  How much, how is it managed, communication of policies

- **Software Policies**

  Organization, level of detail, compliance checks, communication

- **Processes for Small Projects**

  Policies and compliance, CMMI, tailoring, infrastructure support, tools

- **Testing**

  Strategy, levels, life cycle, test team, metrics, tools, completion criteria

- **CMMI**

  Drivers, implementation strategies, benefits, obstacles

What have we learned? Training Characteristics (1 of 2)

	A	B	C	D	E	F	G	H	I	J
Mentoring/OJT	X	X	X	X	X	X		X		X
On-line training			X		X	X	X			X
Classes with exercises	X		X	X				X		X
Tiered training program							X			
Required periodic training					X	X	X			X
Lunch time seminars		X	X	X		X				
Continuous training program					X	X	X			X
Relies on hiring knowledge									X	
Role-based training			X	X	X	X	X	X		X

NASA Centers

Commercial Organizations

What have we learned? Training Characteristics (2 of 2)

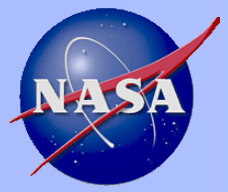
	A	B	C	D	E	F	G	H
Mentoring/OJT	X	X			X	X	X	X
On-line training			X		X	X	X	
Classes with exercises		X				X	X	X
Tiered training program	X					X		
Required periodic training	X				X	X	X	X
Lunch time seminars								
Continuous training program					X			
Relies on hiring knowledge	X			X				X
Role-based training					X	X		

Academic Organizations

Government Organizations

What have we learned? Training Observations

- **Most large organizations provided a variety of options for training**
  - **Corporate training: Often required and more general (Security, Policy)**
  - **Software-specific training and process training done by lower levels of organization**
- **Mentoring is a key component of training across all groups**
  - **New employee training on "real project" at one NASA Center–Assign lead roles to new people, use mentors to assist**
  - **One company won't allow code check-in by new employees until checked by someone more senior**
- **Flight Certification Program at one org: 3 Classes**
  - **3 Classes: Theory; Practices & Design Principles; Risk Management**
  - **Certification involves testing and is mandatory**
- **In the most impressive orgs: Training was "a matter of course"**
  - **Integrated into regular work**
  - **Available exactly when needed**

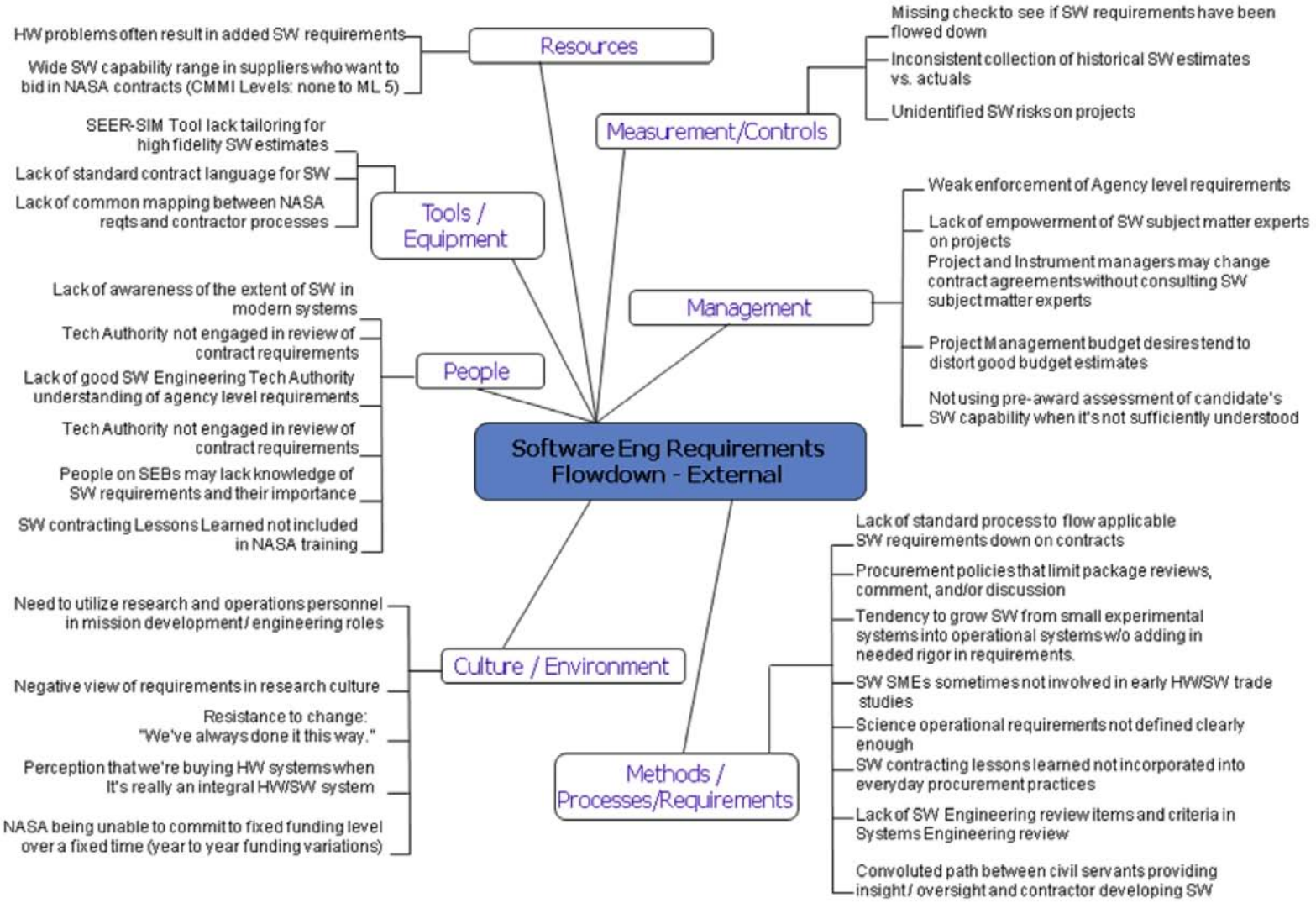


What have we learned? Testing

- **Tool usage: Not much usage of code usage or predictive tools**
  - **Tools used for defect & requirements tracking –JIRA, MKS, DOORs**
  - **Moderate use of static analysis tools- Differing opinions**
  - **Some usage of home-grown tools to run test (scripts), unit tests**
  - **Tool kits, library software, open source tested at same level as developed software**
- **Best Practices mentioned:**
  - **Use a simulator**
  - **Buy all hardware boards at once for consistency in configurations**
  - **“Test as you go”- Test as you build up functionality**
  - **Do off-nominal testing**
  - **Projections of defects across the lifecycle is used to set targeted software improvement goals**
- **Test teams vary in composition**
  - **All orgs used teams independent of developers who wrote code**
  - **Often included operations people and requirements developers**
  - **Flight software test teams sometimes not involved in final hardware/software system testing**
  - **Software Quality Assurance role in software testing not clearly defined**
- **Testing cost: Life cycle testing costs ranged from 21% to 45%**

What have we learned? Acquisition

Potential root causes when NASA Software Engineering Requirements fail to be included in contracts



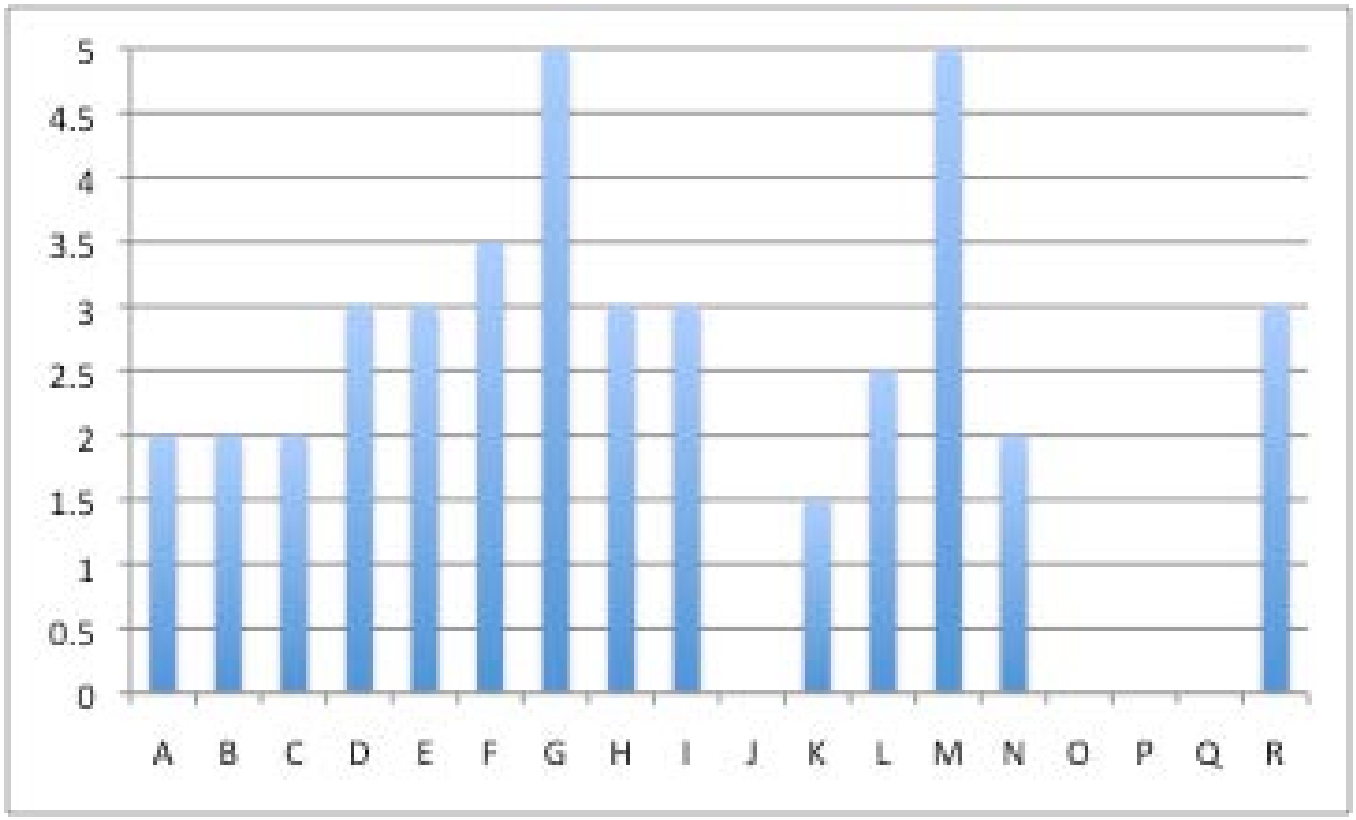
What Have We Learned ? Acquisition Observations

- **DoD groups interviewed didn't seem to have NASA's problems with flowing down requirements to contractors**
  - **DoD had a large set of specific contract requirements for software**
  - **Acquisition training is provided to procurement and technical people**
  - **Some DoD groups used CMMI appraisers to evaluate contractors**
- **Some industry orgs had specific contract requirements for software**
- **Levying NPR 7150.2 directly on contracts caused confusion for Class B and C projects**
- **CMMI provided a good communication vehicle for contract expectations**
- **Include a software representative on systems procurements**
- **NASA's cost estimation and budgeting processes caused problems for contractors**
- **General need to address acquisition process in terms of using open source software in the development activities**

What Have We Learned? Small Projects and Processes

- **Half of the large industry organizations only had small projects that were research focused**
  - **Small more critical ones were done under a larger project**
  - **One large company had critical smaller projects, with institutional support**
- **One military organization had ¾ small projects; university projects were small; and one industry organization was all small**
- **Best practices for critical small projects:**
  - **Still required to follow rigorous processes –tailoring used on formality of reviews and extent of documentation**
  - **Strong peer review culture**
  - **Institutional support for help in tailoring, tool support**
  - **Sometimes functions handled across projects-CM, test groups**
  - **One organization commented small projects are better at following process –"Gets to be a way of doing business –Just easier"**
- **The NASA review processes on small projects were very time consuming**
  - **Tailored version could be: Experts interacting in detail around the information in its original format.**
- **NASA oversight was focused mainly on the HW aspects of the project (not much on SW)**

What Have We Learned? CMMI Levels of Organizations



Organizations

A-E: NASA Centers
F-J: Industry
K-N: Government
O-R: University

13 of the 14 NASA Centers, Industry Partners and Other Government Agencies interviewed are working with the CMMI model to improve their software engineering processes and software quality

What Have We Learned? CMMI

- Of the organizations without ratings, most had investigated using it (or had mapped to it or felt they used most of it)
- Senior management support of software improvement as well as targeted goals (CMMI levels, measurable quality goals, etc.) are key to excellence in Software Engineering
- Need a good set of organizational measures
- Benefits mentioned:
  - Better cost estimation (more credibility with projects)
  - Better planning, schedule tracking (ability to tell project exactly HOW much longer – not 90% done syndrome!) , Test plan and test procedures were completed earlier in the lifecycle development
  - Formal inspections –Errors found earlier, less phase-escape problems
  - Did feel ML5 was a benefit: Manage by metrics; Being able to determine impacts of improvements; 240% improvement seen between CMMI levels 3 and 5
  - CMMI enables "managing with metrics" to become real (rather than being a buzzword)
  - Common Processes to allow moving people around, faster start up time, faster product development
  - Tracking and predictability, Monthly measurement reports show how you are doing
  - Better requirements management (control requirements changes)
  - Customer satisfaction, Cost and effort predictability
  - Reuse of procedures
  - Solidifies organizational culture

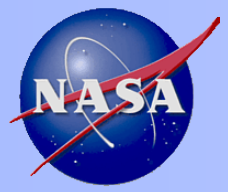
"Process isn't extra—
it's how we do business"

What Have We Learned? Common Processes

- **Effective utilization of workforce goes hand-in-hand with common processes for related organizations**
- **Economics, affordability and competition is driving organizations into the use of common organizational level processes**
- **Common organizational processes facilitated cross organizational projects**
- **Common Processes to allow moving people around, faster start up time, faster product development**
- **Advantage will be to share projects in a seamless manner across the sister organizations**
- **Strategy to share common processes across sister organizations**
  - **Telecons**
  - **Enterprise level Software Engineering Process Groups in place to support the development and management of common organizational processes**
  - **Enterprise approach can take 2 – 3 years to implement**
  - **Advantage will be to share projects in a seamless manner across the sister organizations**
  - **Will be able to compete as an organization with a workforce of 3000 people, instead of separate 1000 tech workforce organizations**

Summary

- **Benchmarking was very interesting and provided a wealth of information**
  - **We did see potential solutions to some of our "top 10" issues**
  - **We have an assessment of where NASA stands with relation to other aerospace/defense groups**
- **We formed new contacts and potential collaborations**
  - **Several organizations sent us examples of their templates, processes**
  - **Many of the organizations were interested in future collaboration: sharing of training, metrics, CMMI appraisers, instructors, etc.**
- **We received feedback from some of our contractors/ partners**
  - **Desires to participate in our training; provide feedback on procedures**
  - **Welcomed opportunity to provide feedback on working with NASA**



Backup

Top Ten Software Issues (2010)

1. *Internal NASA-wide requirements (NPD, NPR, & Standards)*
2. *Software Cost Estimation*
3. *Software Workforce level*
4. *Systems Eng. / Software Eng. Interface*
5. *Small Project Implementations*
6. *Empowerment of SW Personnel*
7. *SW Requirements*
8. *Complex Electronics*
9. *Training & Skill Development*
10. *Insufficient attention to SW on Contracts*

…

#13: SW Architectural Analysis & Review (JPL's #1)

…

#15: Model Based SW Development (MSFC's #1)