Establishing a Framework and Testbed for Evaluating and Infusing Software Assurance Tools

NASA OSMA Software Assurance Symposium
August, 2012

PI: Allen Nikora: Jet Propulsion Laboratory, California Institute of Technology
Co-Is: Prof. Dan Port: University of Hawaii
Joel Wilf: Jet Propulsion Laboratory, California Institute of Technology
Intern: Denise Shigeta: Jet Propulsion Laboratory, California Institute of Technology

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The work was sponsored by the NASA Office of Safety and Mission Assurance under the Software Assurance Research Program. This activity is managed locally at JPL through the Assurance and Technology Program Office.

Topics

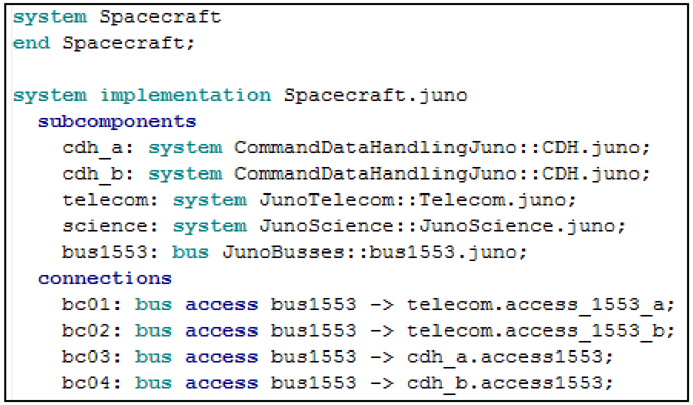
- Problem
- Approach
- Status and Results
- Future Work
- Community Involvement

Problem

- **There are a large number of tools both research and commercial that may be of useful for software assurance**
  - Investigation at JPL and NASA SAWG revealed most assurance is preformed "manually" and is perceived to be inefficient and ineffective for some tasks
  - Investigation at JPL revealed that there are impediments to tool use
- **There is significant research interest in assurance tool development and evaluation research**
  - Many SARP projects are tool related
- **There is a gap between research in tools and their use on projects**
  - Both commercial (COTS) and research-developed tools ("ROTS")

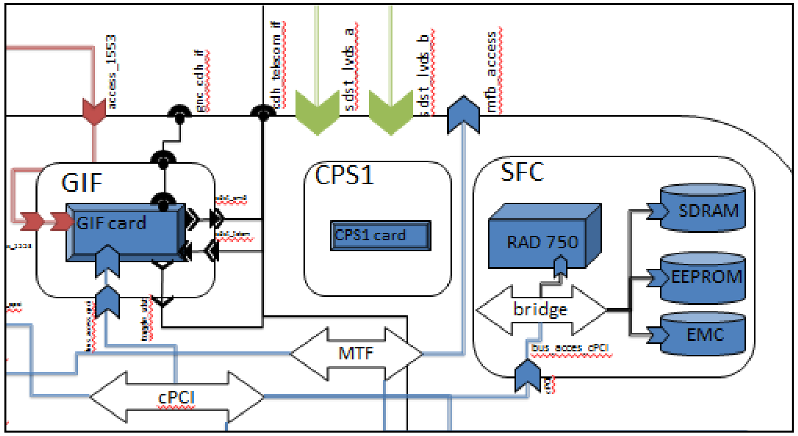
Example: OSATE (AADL IDE)

- **AADL = Architectural Analysis and Design Language**
  - Text and graphical based
  - Models run-time systems
  - Similar to UML/SysML
- **OSATE can perform analyses on AADL model using properties specified**
  - Latency
  - Schedulability
  - Processor capability
- **OSATE is open source**



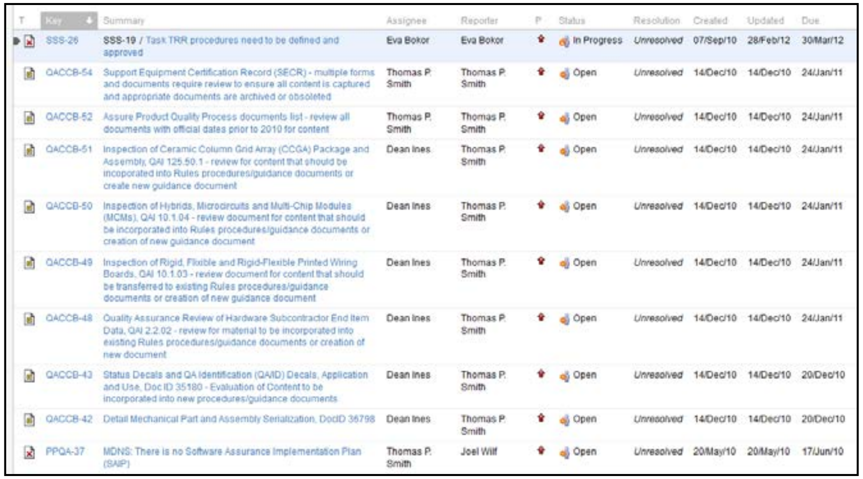
```
system Spacecraft
end Spacecraft;

system implementation Spacecraft.juno
  subcomponents
    cdh_a: system CommandDataHandlingJuno::CDH.juno;
    cdh_b: system CommandDataHandlingJuno::CDH.juno;
    telecom: system JunoTelecom::Telecom.juno;
    science: system JunoScience::JunoScience.juno;
    bus1553: bus JunoBusses::bus1553.juno;
  connections
    bc01: bus access bus1553 -> telecom.access_1553_a;
    bc02: bus access bus1553 -> telecom.access_1553_b;
    bc03: bus access bus1553 -> cdh_a.access1553;
    bc04: bus access bus1553 -> cdh_b.access1553;
```

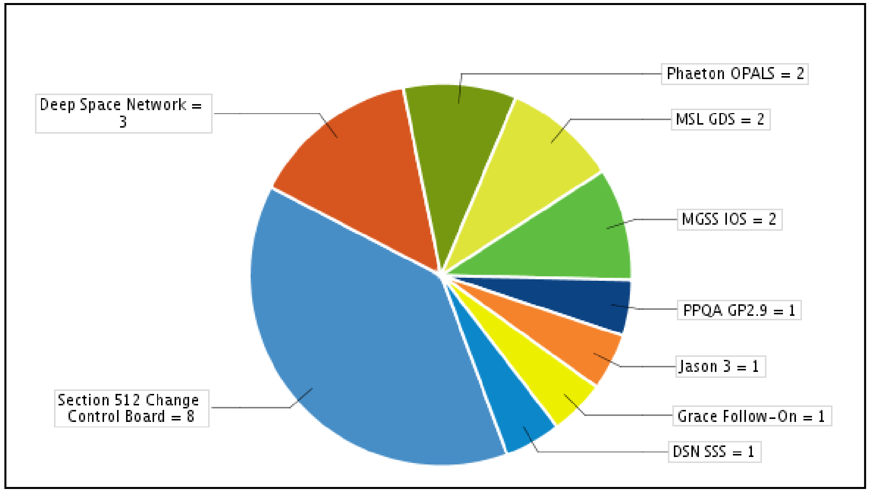


Example: JIRA

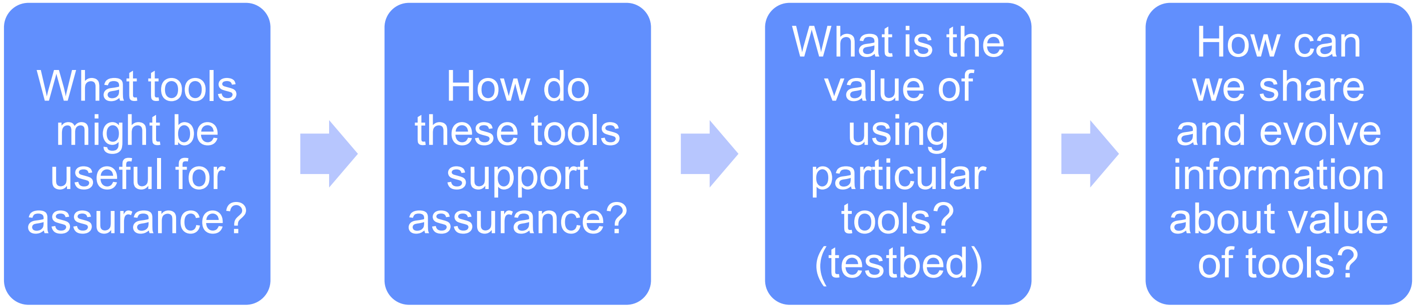
- **Used by SQEs to track SQA issues and tasks for projects**
  - Types of issues
  - Status of issues
  - Priority level of issues
- **Features:**
  - Produce charts and graphs of progress or work still left to do
  - Create filters
  - Create your own dashboard
- **Also used by software developers to track tasks**



T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	SSS-26	SSS-19 / Task TRR procedures need to be defined and approved	Eva Bokor	Eva Bokor	↑	In Progress	*Unresolved*	07/Sep/10	28/Feb/12	30/Mar/12
	QACCB-54	Support Equipment Certification Record (SECR) - multiple forms and documents require review to ensure all content is captured and appropriate documents are archived or obsoleted	Thomas P. Smith	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	24/Jan/11
	QACCB-52	Assure Product Quality Process documents list - review all documents with official dates prior to 2010 for content	Thomas P. Smith	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	24/Jan/11
	QACCB-51	Inspection of Ceramic Column Grid Array (CCGA) Package and Assembly, QAI 125.50.1 - review for content that should be incoporated into Rules procedures/guidance documents or create new guidance document	Dean Ines	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	24/Jan/11
	QACCB-50	Inspection of Hybrids, Microcircuits and Multi-Chip Modules (MCMs), QAI 10.1.04 - review document for content that should be incorporated into Rules procedures/guidance documents or creation of new guidance document	Dean Ines	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	24/Jan/11
	QACCB-49	Inspection of Rigid, Flexible and Rigid-Flexible Printed Wiring Boards, QAI 10.1.03 - review document for content that should be transferred to existing Rules procedures/guidance documents or creation of new guidance document	Dean Ines	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	24/Jan/11
	QACCB-48	Quality Assurance Review of Hardware Subcontractor End Item Data, QAI 2.2.02 - review for material to be incorporated into existing Rules procedures/guidance documents or creation of new document	Dean Ines	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	24/Jan/11
	QACCB-43	Status Decals and QA Identification (QAID) Decals, Application and Use, Doc ID 35180 - Evaluation of Content to be incorporated into new procedures/guidance documents	Dean Ines	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	20/Dec/10
	QACCB-42	Detail Mechanical Part and Assembly Serialization, DocID 36798	Dean Ines	Thomas P. Smith	↑	Open	*Unresolved*	14/Dec/10	14/Dec/10	20/Dec/10
	PPQA-37	MDNS: There is no Software Assurance Implementation Plan (SAIP)	Thomas P. Smith	Joel Wilf	↑	Open	*Unresolved*	20/May/10	20/May/10	17/Jun/10



Approach



What tools might be useful for assurance?

How do these tools support assurance?

What is the value of using particular tools? (testbed)

How can we share and evolve information about value of tools?

I. Survey available tools

II. Develop tool evaluation criteria

III. Evaluate tools under controlled conditions

IV. Develop a specification for the functionality, behavior, and structure of the tool evaluation framework

V. Evaluate a subset of the tools examined in Stage III on real development efforts

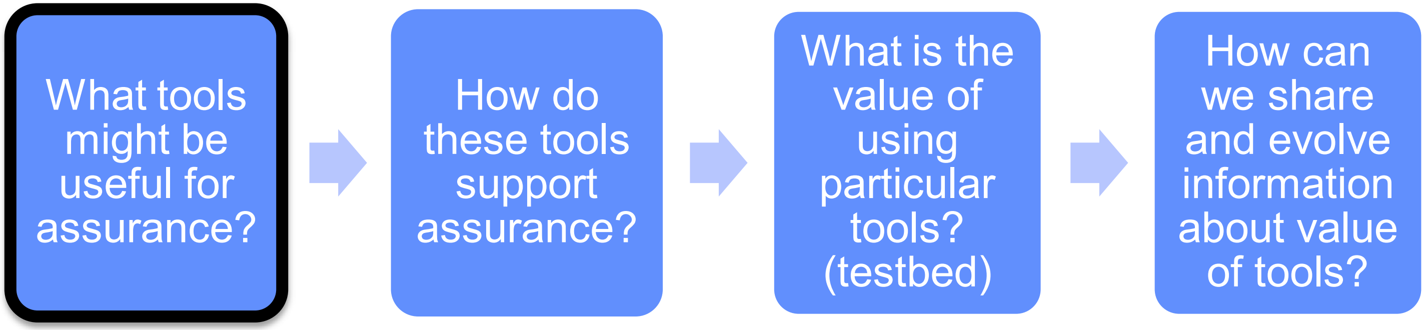
VI. Provide tool evaluations and framework to the assurance community…

Ultimate goal: encourage infusion of valuable tools

Status and Results

I. Survey available tools

- List of candidate tools DB
- Research on tool notes and summaries
- Mapping of tools to assurance activity areas
- Degree of coverage of areas by candidate tools (strengths and gaps)



What tools might be useful for assurance?

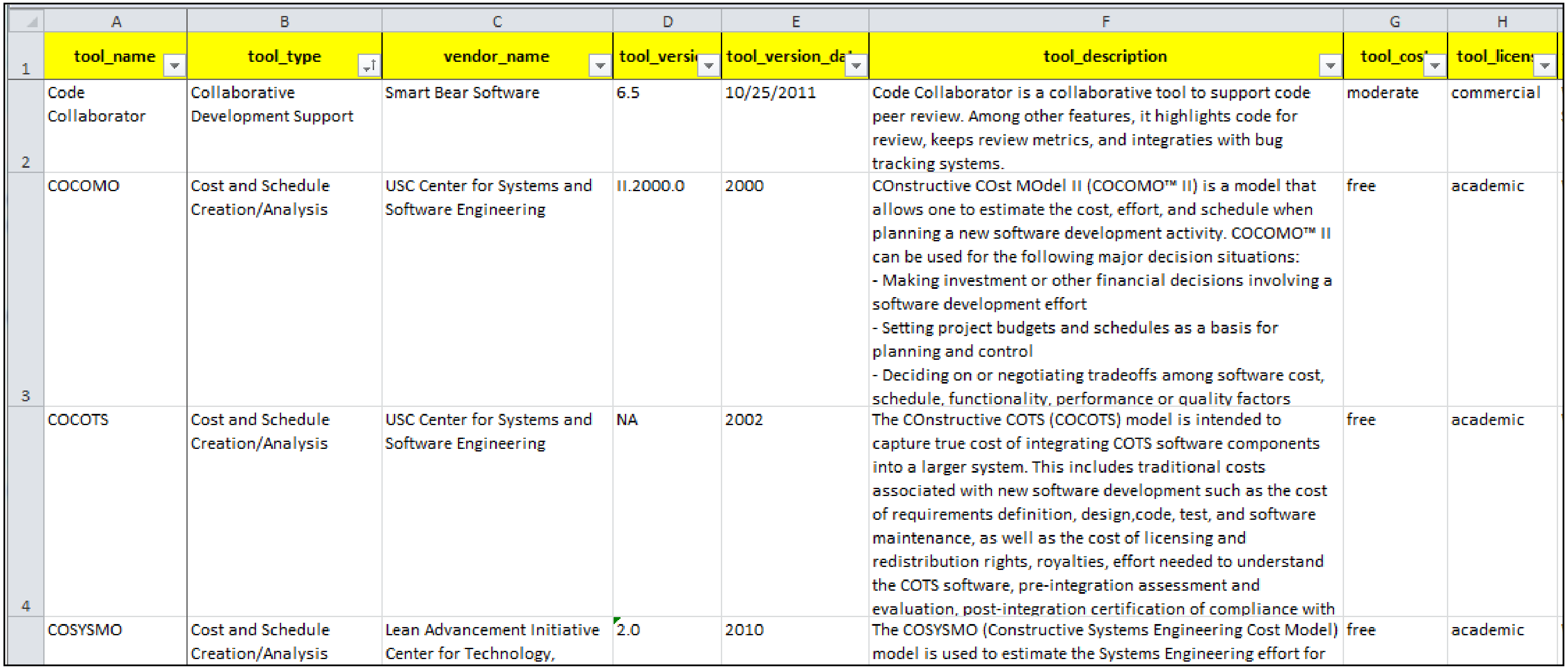
How do these tools support assurance?

What is the value of using particular tools? (testbed)

How can we share and evolve information about value of tools?

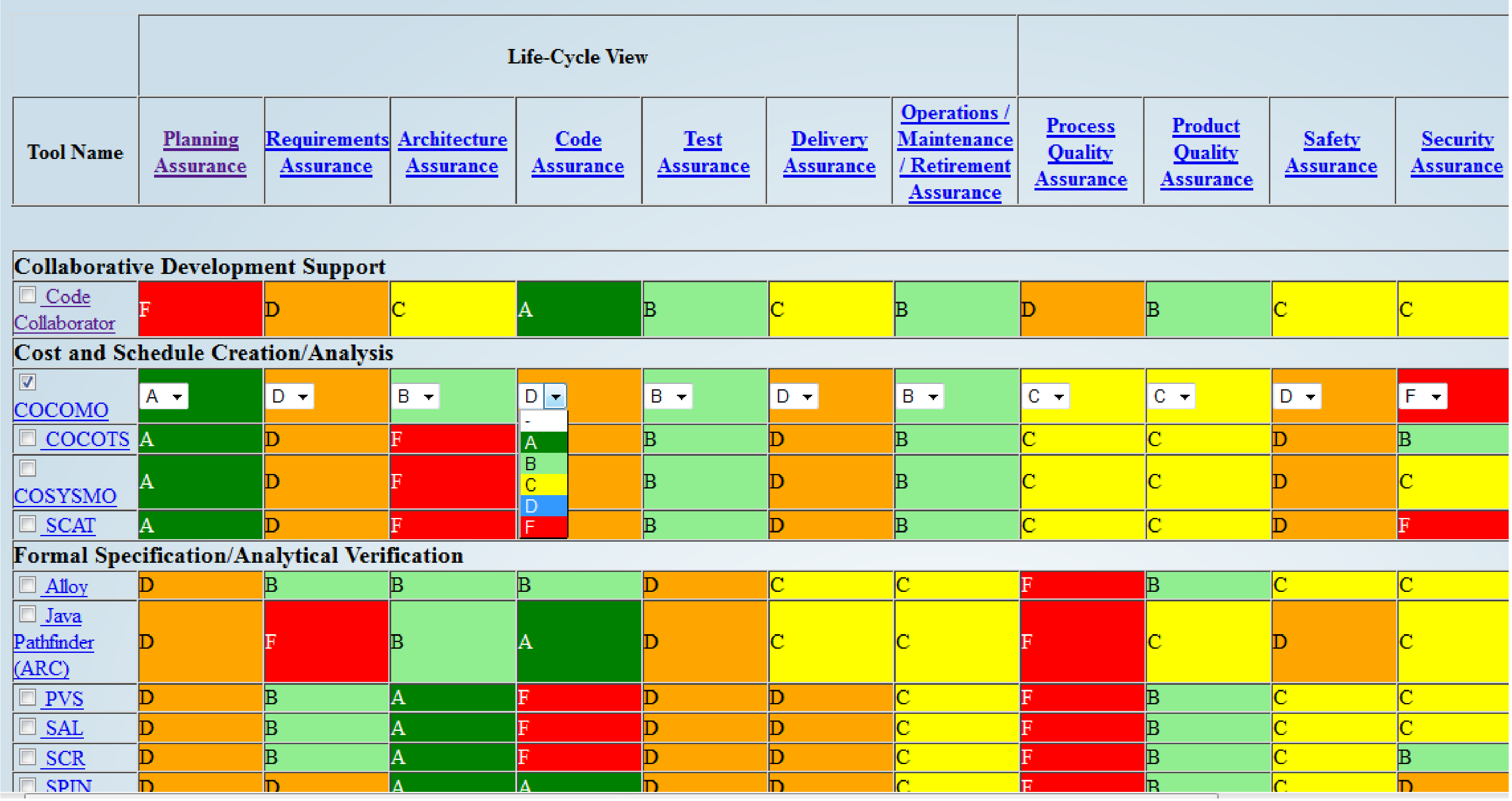
SA Tool Database

- **Information on tools include:**
  - Vendor, version, description, cost, license type, platform, dependencies, etc.
- **Will be used as the source of information for the online resource**



	A	B	C	D	E	F	G	H
1	tool_name	tool_type	vendor_name	tool_versi	tool_version_da	tool_description	tool_cos	tool_licen
2	Code Collaborator	Collaborative Development Support	Smart Bear Software	6.5	10/25/2011	Code Collaborator is a collaborative tool to support code peer review. Among other features, it highlights code for review, keeps review metrics, and integraties with bug tracking systems.	moderate	commercial
3	COCOMO	Cost and Schedule Creation/Analysis	USC Center for Systems and Software Engineering	II.2000.0	2000	COnstructive COst MOdel II (COCOMO™ II) is a model that allows one to estimate the cost, effort, and schedule when planning a new software development activity. COCOMO™ II can be used for the following major decision situations: - Making investment or other financial decisions involving a software development effort - Setting project budgets and schedules as a basis for planning and control - Deciding on or negotiating tradeoffs among software cost, schedule, functionality, performance or quality factors	free	academic
4	COCOTS	Cost and Schedule Creation/Analysis	USC Center for Systems and Software Engineering	NA	2002	The COnstructive COTS (COCOTS) model is intended to capture true cost of integrating COTS software components into a larger system. This includes traditional costs associated with new software development such as the cost of requirements definition, design,code, test, and software maintenance, as well as the cost of licensing and redistribution rights, royalties, effort needed to understand the COTS software, pre-integration assessment and evaluation, post-integration certification of compliance with	free	academic
	COSYSMO	Cost and Schedule Creation/Analysis	Lean Advancement Initiative Center for Technology,	2.0	2010	The COSYSMO (Constructive Systems Engineering Cost Model) model is used to estimate the Systems Engineering effort for	free	academic

Tool Application Survey



	Life-Cycle View										
Tool Name	Planning Assurance	Requirements Assurance	Architecture Assurance	Code Assurance	Test Assurance	Delivery Assurance	Operations / Maintenance / Retirement Assurance	Process Quality Assurance	Product Quality Assurance	Safety Assurance	Security Assurance
Collaborative Development Support											
☐ Code Collaborator	F	D	C	A	B	C	B	D	B	C	C
Cost and Schedule Creation/Analysis											
☑ COCOMO	A ▾	D ▾	B ▾	D ▾ - A B C D F	B ▾	D ▾	B ▾	C ▾	C ▾	D ▾	F ▾
☐ COCOTS	A	D	F		B	D	B	C	C	D	B
☐ COSYSMO	A	D	F		B	D	B	C	C	D	C
☐ SCAT	A	D	F		B	D	B	C	C	D	F
Formal Specification/Analytical Verification											
☐ Alloy	D	B	B	B	D	C	C	F	B	C	C
☐ Java Pathfinder (ARC)	D	F	B	A	D	C	C	F	C	D	C
☐ PVS	D	B	A	F	D	D	C	F	B	C	C
☐ SAL	D	B	A	F	D	D	C	F	B	C	C
☐ SCR	D	B	A	F	D	D	C	F	B	C	B
☐ SPIN	D	D	A	A	D	D	C	F	B	C	D

Example Survey Results

- Evaluations from 9 assurance practitioners – Commonly used tools
- Tools below had similar evaluations
  - Support = looked for all A's and B's
  - No support = looked for all D's and F's

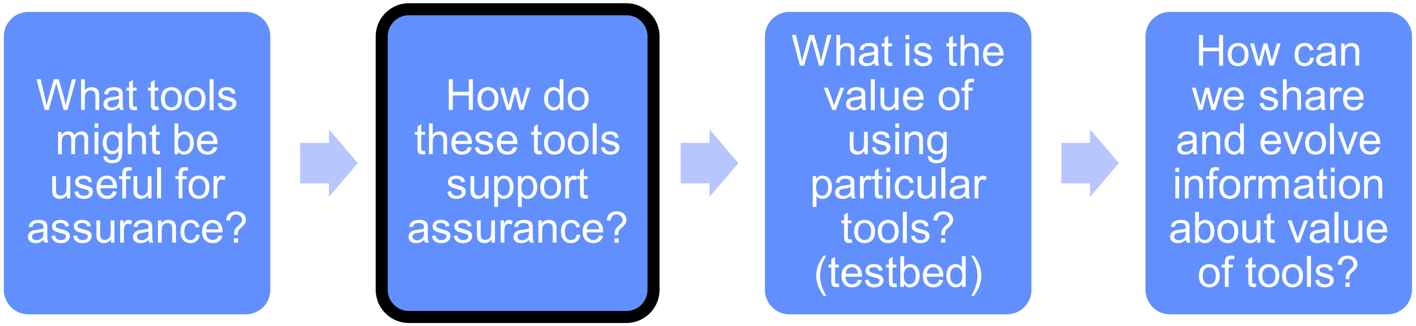
Tool	Assurance areas supported	Assurance areas not supported
Bugzilla	Code, test, dependability assurance	Cost, planning, architecture, performance, resource assurance
Coverity	Code, test, product, safety, risk assurance	Planning, requirements, architecture, process, cost, schedule, assurance management
DOORS	Requirements, test, delivery, product, risk assurance	Code, cost, schedule assurance
JIRA	Code, test, delivery, operations & maintenance, process, product, risk, contractor assurance, assurance management	Performance, cost assurance
JPLs PRS	Code, test, delivery, dependability assurance	Cost, architecture assurance
SLIC	Product, cost, schedule assurance	Planning, requirements, architecture, test, safety, security, performance, dependability, resource assurance

Status and Results (Continued)

I. Survey available tools

II. Develop tool evaluation criteria

- Candidate evaluation criteria
- Mappings of tools to decisions and evidence supplied



What tools might be useful for assurance?

How do these tools support assurance?

What is the value of using particular tools? (testbed)

How can we share and evolve information about value of tools?

Preliminary Assessment Criteria

- **Applicability**
  - From tool survey data
- **Effectiveness**
  - Scalability ratio = {max amount handleable with tool / max amount handleable manually}
  - Assurance productivity efficiency = {average amount assured per function point with tool / average amount assured per function point manual}
  - Accuracy ratio = {average number errors with tool / average number error manual}
  - Average accuracy = {average errors with tool}
  - Accuracy variance = {variance of errors with tool}
  - Coverage fraction [0-1] = {amount tool covers / total amount}
- **Tool Availability**
- **Usability**
  - As per Seffah et al consolidated usability model [Software Qual J (2006) 14: 159–178]
- **Relationship To other Tools**

Status and Results (Continued)

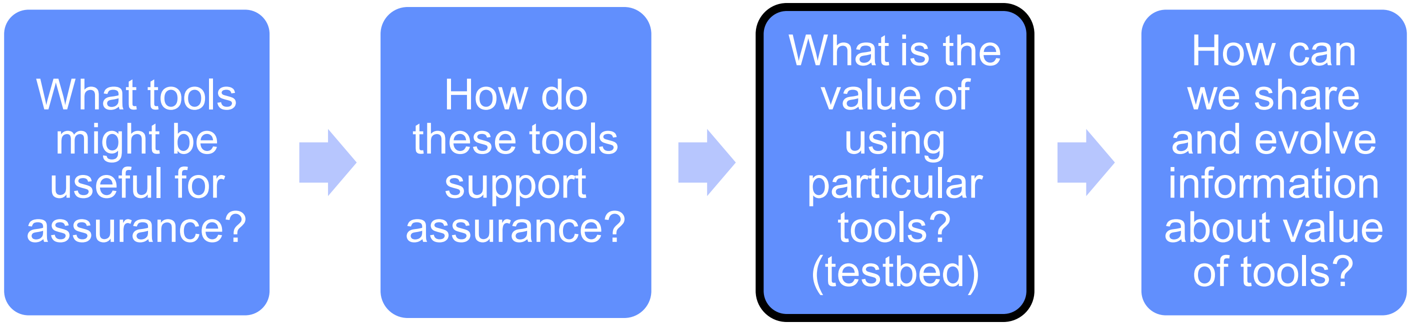
I. Survey available tools

II. Develop tool evaluation criteria

III. **Evaluate tools under controlled conditions**
   - Example: ODASA Static Code Analyzer *(in progress)*

IV. Develop a specification for the functionality, behavior, and structure of the tool evaluation framework
   - Top-down framework found to be too constricting
   - Use common criteria + user criteria + user experience

V. **Evaluate a subset of the tools examined in Stage III on real development efforts (III and V done in concert)**
   - Example: ODASA Static Code Analyzer *(in progress)*
   - Evaluating on SMAP, MGSS IOS, ICX (DoD project) and others (*in progress*)



What tools might be useful for assurance?

How do these tools support assurance?

What is the value of using particular tools? (testbed)

How can we share and evolve information about value of tools?

Status and Results (Continued)

I. Survey available tools

II. Develop tool evaluation criteria

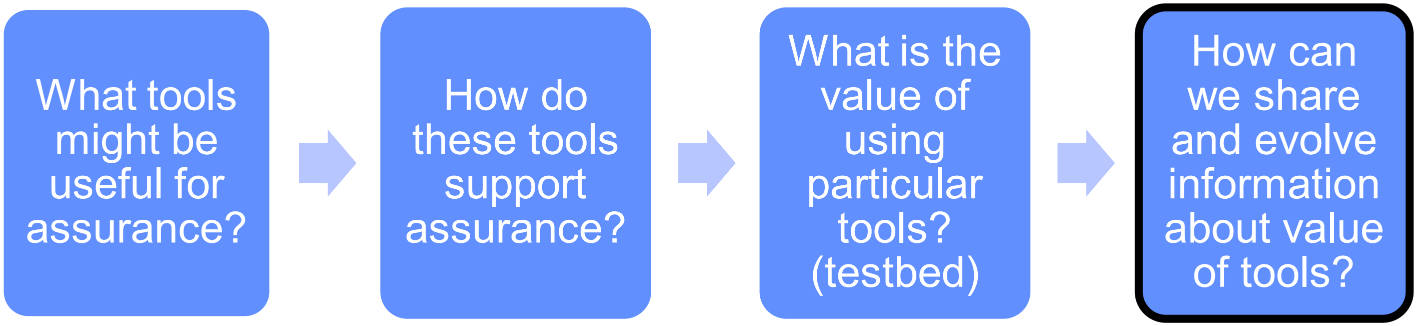
III. Evaluate tools under controlled conditions

IV. Develop a specification for the functionality, behavior, and structure of the tool evaluation framework

V. Evaluate a subset of the tools examined in Stage III on real development efforts (III and V done in concert)

VI. Provide tool evaluations and framework to the assurance community – *In Progress*

- *Setting up JPL externally-facing site by end of FY*
- *Will include Assurance Tool Survey, Database, and WIKI*



What tools might be useful for assurance?

How do these tools support assurance?

What is the value of using particular tools? (testbed)

How can we share and evolve information about value of tools?

Future Work

- **Collect more data**
  - SARP TIM next week – extract knowledge from NASA assurance researchers while they're here!
- **Analyze/interpret the data**
  - Statistical hypothesis testing
  - Distribution of "grades"
  - Determine which tools provide strong support
  - Are there any areas lacking tool support where we should develop new tools?
  - What opportunities for tool use are there?
- **Set up online resource *(before end of FY12)***
  - Survey: to collect more data and evolve tool/applicability matrix
  - Database: to communicate basic tool information
  - WIKI: to comment on and learn about tools experience

Community Involvement

- **Get interviewed!**
  - Contribute your assessment on tools applied to assurance
- **Use, evaluate, and infuse tools**
- **Contribute to the tools WIKI when it goes online**