



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

**NASA OSMA Software Assurance Symposium
August, 2011**

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

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.



Agenda

- Problem
- Approach
- Relevance to NASA
- Status and Results
- Community Involvement
- Key Applications



Problem (1)

- **There are a large number of tools that may be of useful for software assurance**
 - modeling and analysis tools
 - model checkers, theorem provers, and code analyzers
 - measurement tools
 - software reliability growth models and test coverage analyzers
 - traceability analysis tools
 - tools for assessing product and process compliance to standards
- **There is significant research interest in assurance tool development and evaluation research**
 - over 50% of the 2009 SAS technical presentations (excluding technical updates) were directly assurance tool related





Problem (2)

- **There is a gap between research in tools and their use on projects**
 - On a recent survey of JPL quality assurance personnel and assurance customers, 100% of the respondents either agreed or strongly agreed with the statement “tool use and automation for SQA is limited – manual methods dominate”
- **Investigation at JPL revealed that there are impediments to tool use**
 - high cost, lack of user training, a steep learning curve, failure to meet critical user needs, lack of institutional coordination, and high overhead in indentifying and evaluating potentially useful tools.
 - there is often little understanding of how these tools are related to each other, and how they are most effectively used together within a given assurance effort





Problem (3)

- **The current state of tool use is**
 - Underutilization of tools in supporting software assurance activities
 - Over-reliance and focus on manual methods, such as checklists and document review
 - Missed opportunities to increase the effectiveness of assurance
 - Missed opportunities to add activities that cannot be performed manually
 - Missed opportunities to improve management of the assurance organization as a whole





Approach

- **The research involves the following steps:**
 - 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





Relevance to NASA

- The goal of the research is to enable...
 - Members of the NASA-wide assurance community to find the most suitable tools for their particular assurance tasks and build their own “assurance toolbox”
 - More effective use of existing assurance tools by providing an online resource for members of the NASA-wide assurance community to share information about tools with which they are familiar
 - Members of the NASA assurance community to identify those needs which are not satisfied by existing tools, allowing NASA center assurance and OSMA to more effectively prioritize assurance tool research, development and acquisition resources





Status and Results

I. Survey available tools

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

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





Status and Results

Survey Tools: Candidate Tools

Collaborative Development Support
Code Collaborator
Cost and Schedule Creation/Analysis
COCOMO
COCOTS
COSYSMO
SCAT
Formal Specification/Analytical Verification
AADL
Alloy
Java Pathfinder (ARC)
PVS
SAL
SCR
SPIN
UPPAAL
General Purpose
MatLab/FreeMat
R
RapidMiner
RARGEN
WEKA

Integrated Development Environment
Eclipse
Rational (includes DOORS, Logiscope, etc.)
Software Developer's Assistant (Tietronix)
Issue Tracking and Reporting
AAMS - DSN's Anomaly Reporting System
Bugzilla
Clearcase/Clearquest
JIRA
JPL's Problem Reporting System (PRS)
Trac
Model-Based Engineering
Labview
Little Jil
MagicDraw
Matlab (Simulink)
Rhapsody
SpecTRM
Statemate
TEAMS
Quantitative Assessment
CASRE
COQUALMO
DDP
Dymonda (Implements CSRM)
FREstimate
SHARPE (Markov models, stochastic Petri Nets, etc.)
SMERFS^3

Requirements Specification/Analysis
ARM (NL requirements analysis tool by GSFC)
ASCE by Adelard (safety case development tool - http://www.adelard.com/web/hnav/ASCE/)
DOORS
GALILEO (Dynamic Fault Trees) Requirements Assistant
Retro (Jane Hayes' NL requirements tracing tool)
Safety Analysis Checklist (from Janie Hill)
SMART (upgraded version of ARM by GSFC)
Source Code Analysis
Code Sonar
Coverity
FindBugs
Klocwork
PMD
SCRUB
SLIC
Test Generation/Support
Ballista
gcov
JProbe
JProfiler
JUnit
T-VEC (model-based test case generator)
Other
ASL (JPL Approved Supplier List)





Status and Results

Survey Tools: Assurance Areas and Representative Activities

Areas	Example Activities
Architecture A'nce	Assessment of architectural properties, formal modeling and analysis, trace verification.
A'nce Management	Estimation of scope of activities, cost estimation, issue tracking, process monitoring, reporting
Code A'nce	Static code analysis, code inspection
Contractor A'nce	Assessment of contract, assessment of supplier plan, compliance monitoring, work product acceptance
Cost A'nce	Assessment of project cost estimate, tracking plan vs. actual cost
Delivery A'nce	Software Review and Certification Record (SRCR)
Process A'nce	Process audit PPQA
Product A'nce	Product audit PPQA
Project A'nce	Tracking plan vs. actual project parameters and margins
Reliability A'nce	Tracking plan vs. actual reliability growth, Assuring Problem/Failure Report (P/FR) closure
Requirements A'nce	Assessment of requirements for completeness, consistency, and correctness
Resource A'nce	Assessment of resource adequacy to meet plan, tracking resource availability and utilization
Risk A'nce	Assessment of risk list, audit of risk management process
Safety A'nce	Preliminary hazard analysis, fault tree analysis, FMEA
Schedule A'nce	Tracking plan vs. actual schedule
Security A'nce	Assessment of threat analysis
Test A'nce	Assessment of test plan, test process audit





Status and Results

Survey Tools: Mapping Tools to Activity Areas

	Collaborative Development Support			Cost and Schedule Creation/Analysis				Formal Specification / Analytical Verifi				General Purpose				Integrated Development Environment				Issue Tracking and Repor		
	Code Collaborator	COCOMO	COCOTS	COSYSMO	SCAT	AADL, Alloy, UPPAAL	Java Pathfinder (ARC)	SPIN	MatLab/FreeMat	R	RapidMiner; WEKA	RARGEN	Eclipse (Dan&Joel: too broad?)	Rational (includes DOORS, Logiscope, etc.)	Software Developer's Assistant (Tie	AAMS, Bugzilla, Clear	JIRA	Trac				
Architecture A'nce	0	-2	1			2	1						0	1	0		-1	0	-1			
A'nce Management	-1	0		1	-2							-1		1	0		1					
Code A'nce	2	-1				1	2	-2	-2	2		0	0	-1	-1	1		0	1			
Contractor A'nce	-1		0	1	0	0	-2	-1	-1			-2		-1		1	0	1				
Cost A'nce	-2	2				-2						-2		-1		0	-2					
Delivery A'nce	0	-1				-1	0	0				-1		0	1	1						
Process A'nce	-1	0				-2						0	-1		2	0	1					
Product A'nce	1	0				1	0					1		0	0	1	1					
Project A'nce	-2	2				-2						0		-1	0	1	0	1				
Reliability A'nce	1	-1				2		0	0			0	1	1	0		-1	0	1			
Requirements A'nce	-1	-1				1		-2		-1		0	0	1	1		-1	1	-1	0		
Resource A'nce	-2	-1					2	-2	1	1	0		1	0	-1	-1		-1	0	-1		
Risk A'nce	0	1				-1						1		0				0	2	1		
Safety A'nce	0	-1				0		-1				0	1					-1	0	-1	0	
Schedule A'nce	-2	2				-2						-1							0			



Skip Details



Status and Results

Survey Tools: Tool Evaluation Values

- **2: Provides explicit support**
 - Has capabilities designed to directly support the assurance activity
- **1: Has some direct support or has been adapted or applied**
 - Not explicitly designed to support the activity, but has been adapted and used
- **0: Not Explicitly Supported but conceptually viable**
 - General capabilities exist, but no specific support for assurance activity
- **-1: Generally Inapplicable**
 - Capabilities not aligned with assurance activity
- **-2: Inconsistent with, incompatible or contraindicated**
 - Capabilities run contrary to the assurance activity





Status and Results

Survey Tools: Strengths and Gaps

	Area Support (ave)	Area Support (Med)	Contraindicated (ave)
Architecture Assurance	0.9	1	-1.3
Reliability Assurance	0.8	1	-1.0
Risk Assurance	0.8	1	-1.0
Code Assurance	0.8	1	-1.2
Schedule Assurance	0.8	0	-1.7
Cost Assurance	0.7	0	-1.9
Requirements Assurance	0.7	1	-1.4
Product Assurance	0.7	1	-1.0
Test Assurance	0.6	1	-1.1
Resource Assurance	0.6	1	-1.3
Safety Assurance	0.5	0	-1.0
Contractor Assurance	0.5	1	-1.4
Assurance Management	0.4	0	-1.5
Security Assurance	0.4	0	-1.3
Project Assurance	0.3	0	-1.5
Process Assurance	0.3	0	-1.6
Delivery Assurance	0.2	0	-1.0

Ordered most to least

2=all tools have explicit support

0=no tools have explicit support





Status and Results

- I. Survey available tools**
- II. Develop tool evaluation criteria**
 - Candidate evaluation criteria
 - Mappings of tools to decisions and evidence supplied (in progress)
- 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**





Status and Results

Develop Evaluation Criteria: Candidate Evaluation Criteria

- I. Applicability**
- II. Effectiveness**
- III. Tool Availability**
- IV. Usability**
- V. Relationship to Other Tools**





Status and Results

Develop Evaluation Criteria: Decisions and Evidence

Activity	Decisions to be Made	Evidence Supplied
Assurance Management		
<ul style="list-style-type: none">Initial SQA Cost/Value Model estimation	<ul style="list-style-type: none">Plan according to given level of SQARaise SQA level	<ul style="list-style-type: none">Cost/Value ModelWaivers/Risk analysis
<ul style="list-style-type: none">Support MAM status meetings	<ul style="list-style-type: none">Address assurance issues and concernsNo action needed	<ul style="list-style-type: none">Report on Project healthDescription of issues and concerns
<ul style="list-style-type: none">Support OSMS reviews	<ul style="list-style-type: none">Address assurance issues and concernsNo action needed	<ul style="list-style-type: none">OSMS fever chartsDescription of issues and concerns





Status and Results

Work in Progress and Work Remaining

- 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

**** IN PROGRESS ****





Community Involvement

I. We have implemented an online center to contribute to the tools to activity areas mapping

- Add new tools to be evaluated
- Evolve relevance to assurance areas assessments
 - Add your knowledge of tool application
 - Correct or comment on current evaluations

	Collaborative Development Support	Code Collaborator	Cost and Schedule Creation/Analysis	COCOMO	COCOTS	COSYSMO	SCAT	Formal Specification/Analytical Verif	AAVL Alloy, UPPAAL	Java Pathfinder	PI's SAL	SCR	SPIN	General Purpose	MatLab/FreeMat	R	RapidMiner WEKA	RARGEN	Integrated Development Environment	Eclipse (Dan&Joel: top broad?)	Rational (includes DOORS)	Software Developer's Assistant (Trac)	Issue Tracking and Rep	AAMS - Bugzilla, Clear	JIRA	Trac
Architecture A'nce	0	-2	1				2	1					-1						0	1	0		-1	0	-1	
A'nce Management	-1	0			1	-2							-2						-1		1	0		1		
Code A'nce	2	-1					1	2	-2	-2	2				0	0	-1	-1	1		0	1				
Contractor A'nce	-1		0	1	0	0	-2	-1	-1				-2						-1		1	0		1		
Cost A'nce	-2	2					-2						-2						-1		0	-2				
Delivery A'nce	0	-1					-1	0	0				-1						-1	0	1	1				
Process A'nce	-1	0					-2						0	-1				0		-1	0	1		1		
Product A'nce	1	0					1		0				0					1		0	0	0		1		
Project A'nce	-2	2					-2						0						-1	0	1	0		1		
Reliability A'nce	1	-1					2		0	0			0	1	1	0			-1	0	-1	0	1			
Requirements A'nce	-1	-1					1	-2				-1		0	0	1	1		-1	1	-1	0				
Resource A'nce	-2	-1					2	-2	1	1	0			1	0	-1	-1		-1	0	-2		-1	0	-1	
Risk A'nce	0	1					-1						1				2	1	0			0	2	1		
Safety A'nce	0	-1					0	-1					0	1					-1	0	-1	0				
Schedule A'nce	-2	2					-2						-1						-1		0					





Key Applications

- **At the conclusion of this work we provide**
 - Web accessible DB of assurance tools and evaluations
 - Means to contribute and evolve tool evaluations
- **How can you use this?**
 - Identify potentially useful tools for given assurance tasks
 - Use as an outlet to infuse or market a new tool (e.g. SARP research deliverable)
 - Identify areas that may benefit from or need new tools
 - Use effectiveness assessments to benchmark, improve planning and management of assurance activity

