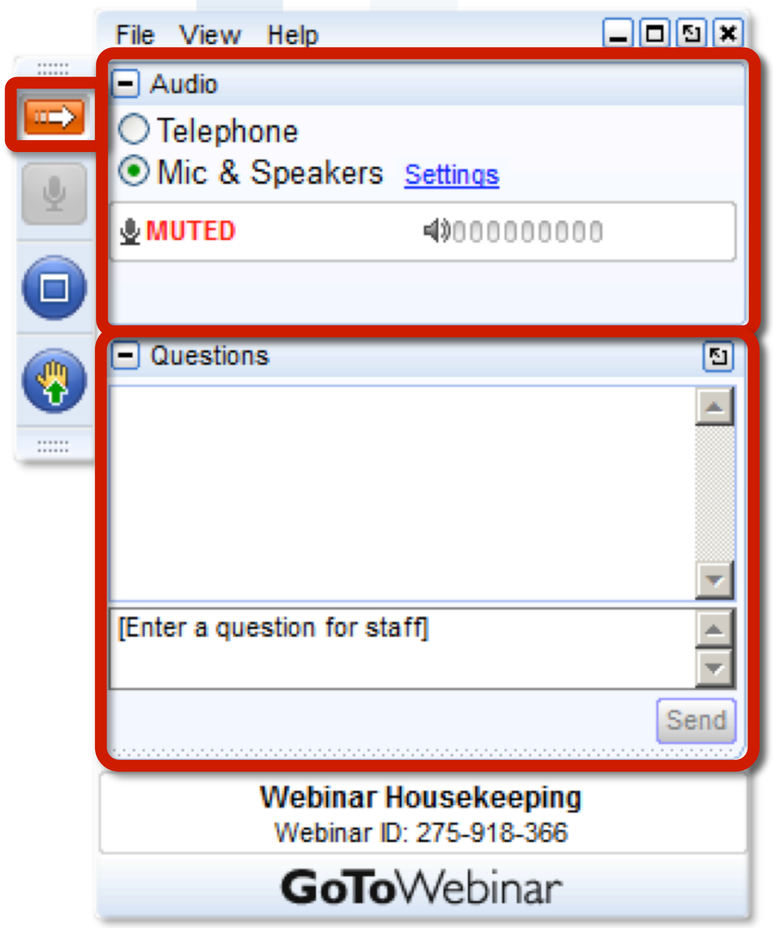
Keeping Bugs Out of Your Code: Why You Need a Development Testing Platform

Arthur Hicken - Evangelist

March 2013

GoToWebinar Housekeeping



Webinar Housekeeping
Webinar ID: 275-918-366

Your Participation

Open and hide your control panel

Join audio:

- Choose "Mic & Speakers" to use VoIP
- Choose "Telephone" and dial using the information provided

Submit questions and comments via the Questions panel

Note: Today's presentation is being recorded and will be provided within a week.

Agenda

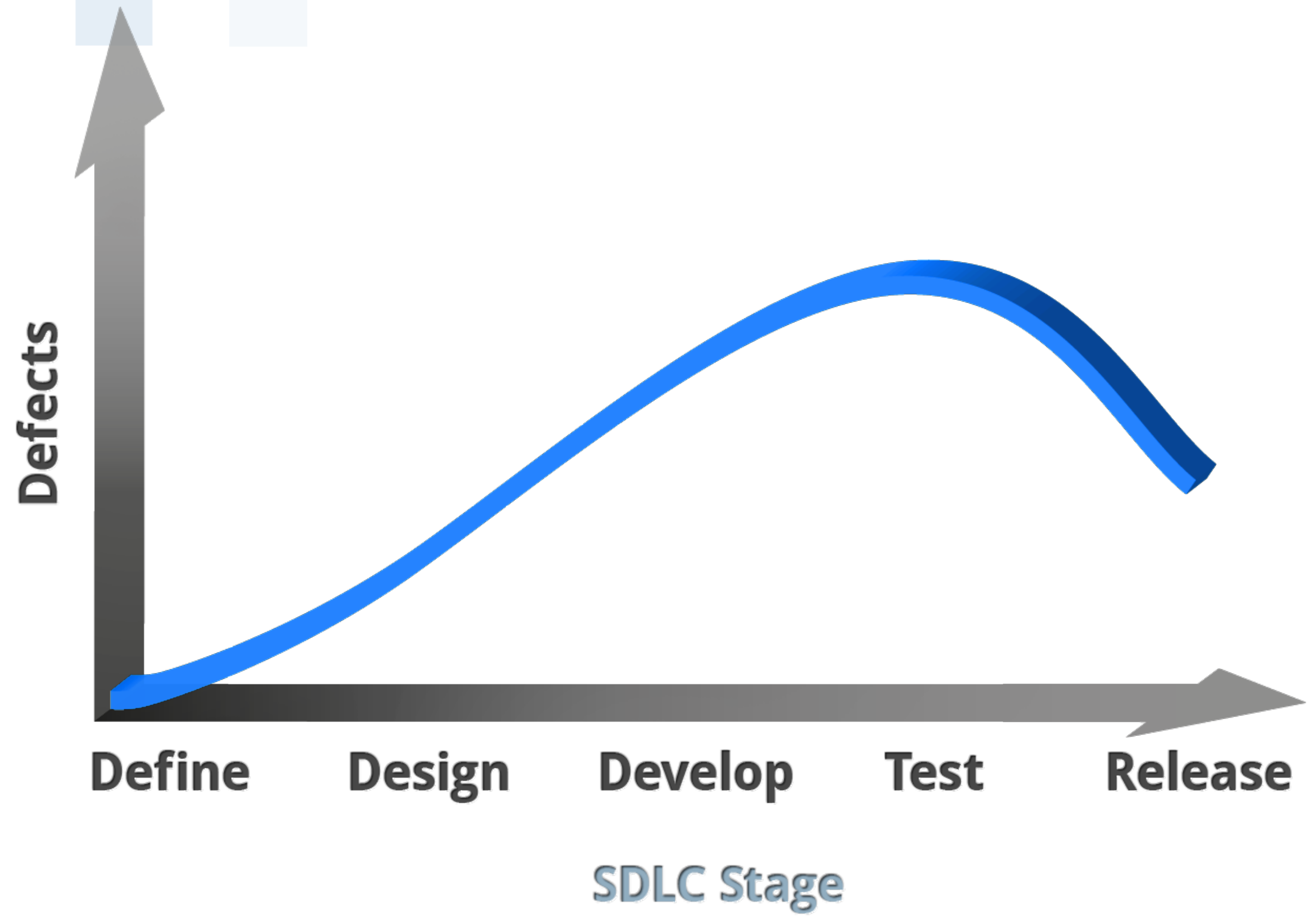
- Early detection & prevention
- The importance of policy
- A Development Testing Platform

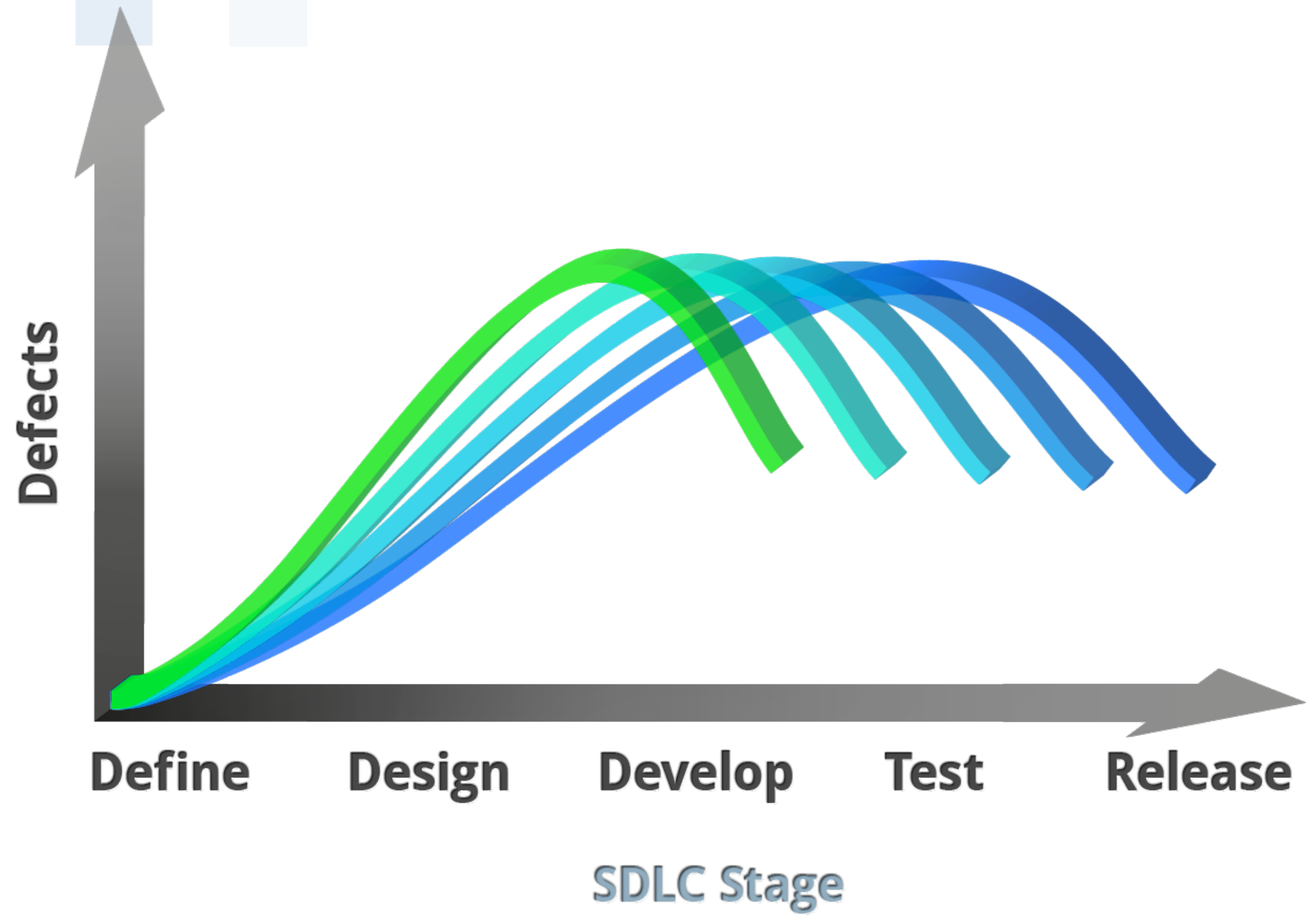
About Parasoft

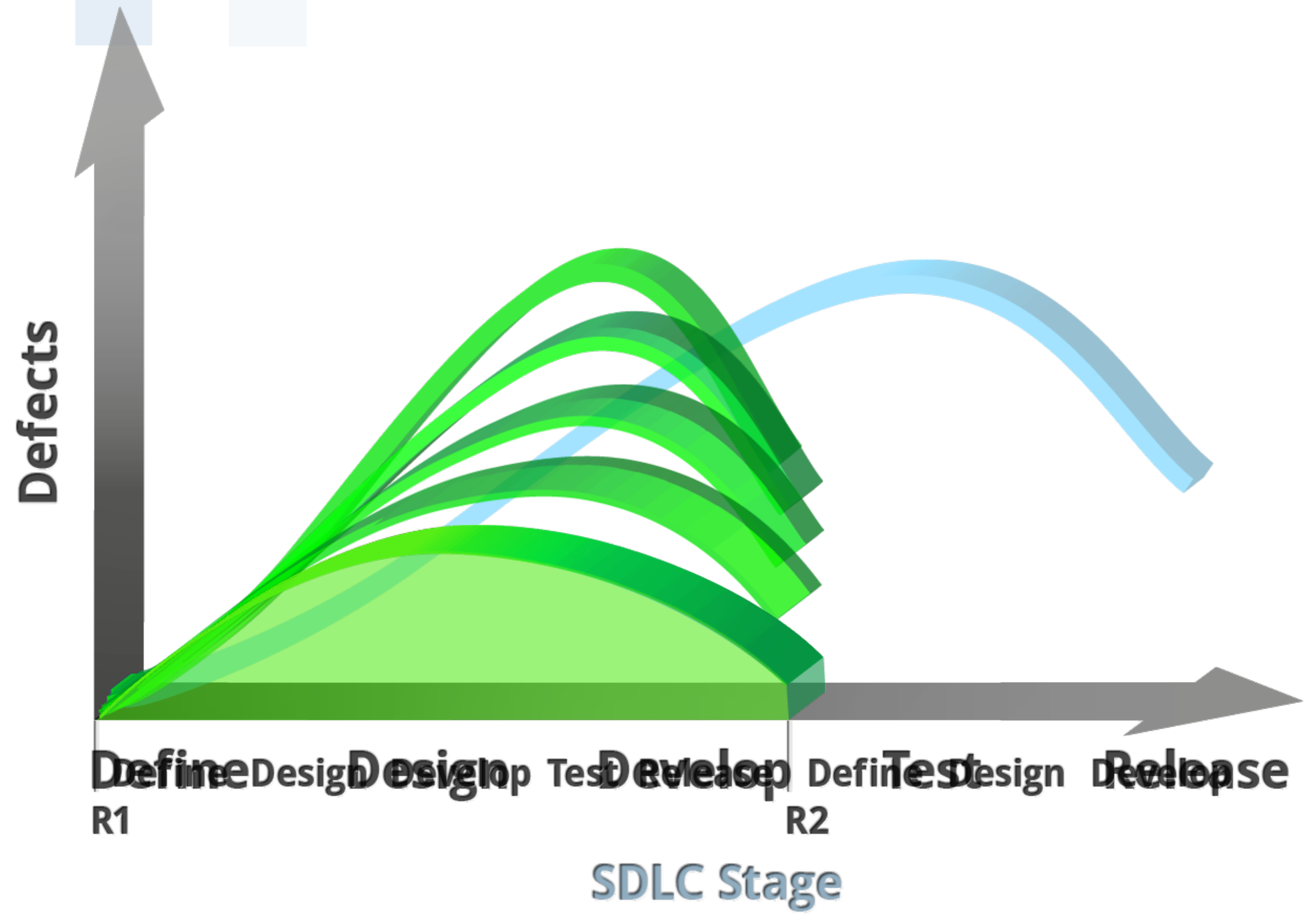
PARASOFT®

World Renowned for Automated Defect Prevention

26 Yrs	Founded in 1987
Highly Focused	Privately held No debt, No VCs
>7,000	Customers worldwide
26	Years of profitable growth Years of innovation and customer value
28	Patents associated with software quality


Defects
Define
Design
Develop
Test
Release
SDLC Stage


Defects
Define
Design
Develop
Test
Release
SDLC Stage


Defects
Define
Design
Develop
Test
Release
Define Design Develop Test Release
R1
Define Design Develop
R2
SDLC Stage

Key Proposition

Policy-driven approach builds quality into the application development lifecycle

-Accurate measurement of quality

-Accurate measurement of productivity

-Prevents errors

-Eliminates waste

ROI for implementing a Static Analysis policy is significant. Especially for embedded systems.

What's in a policy?

- What teams need to perform static analysis
- What projects require static analysis
- What rules are required
- What degree of compliance is required
- When suppressions are allowed
- When violations in legacy code need to be fixed
- Whether you ship code with static analysis violations

Policy - Is It Working?

- Setup policies
  - Test Coverage
  - Unit Tests Executed
  - SA Compliance
  - Peer review expectations
  - Measure – Monitor - Improve
- Base threshold on current results
- Incremental improvement

Prevention

Your software is secure – Have a great day!

Early Detection

You have security problems. Fixing them will delay your release. You don't have time to address the root cause, so you'll have to triage which things you can fix and just patch some of them.

Which would you prefer?

Detection vs. Prevention

Detection / Auditing

- Identifies symptoms
- Easy to perform but …
  - Doesn’t find all vulnerabilities
  - To fix problems at the end of the process is costly
- Doesn’t prevent future vulnerabilities
- Penetration testing only finds prescribed problems
- You can't test quality or security into a product

Prevention

- Identifies the root causes of vulnerabilities
- Quality must be built into the code
- Immediate validation prevents bug chasing
- Easy enforcement of policies

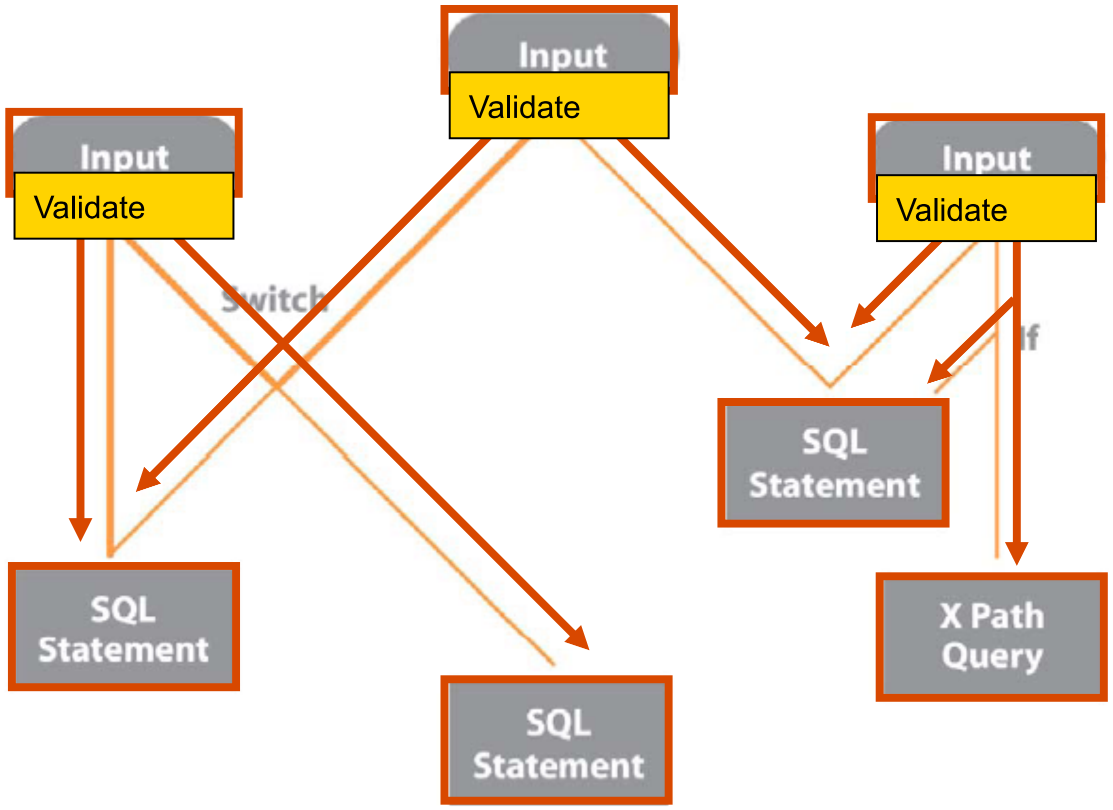
Coding Defensively: Validating inputs

The Solution = validate inputs upon entry

3 entry points

Many paths through the code

4 potential vulnerabilities



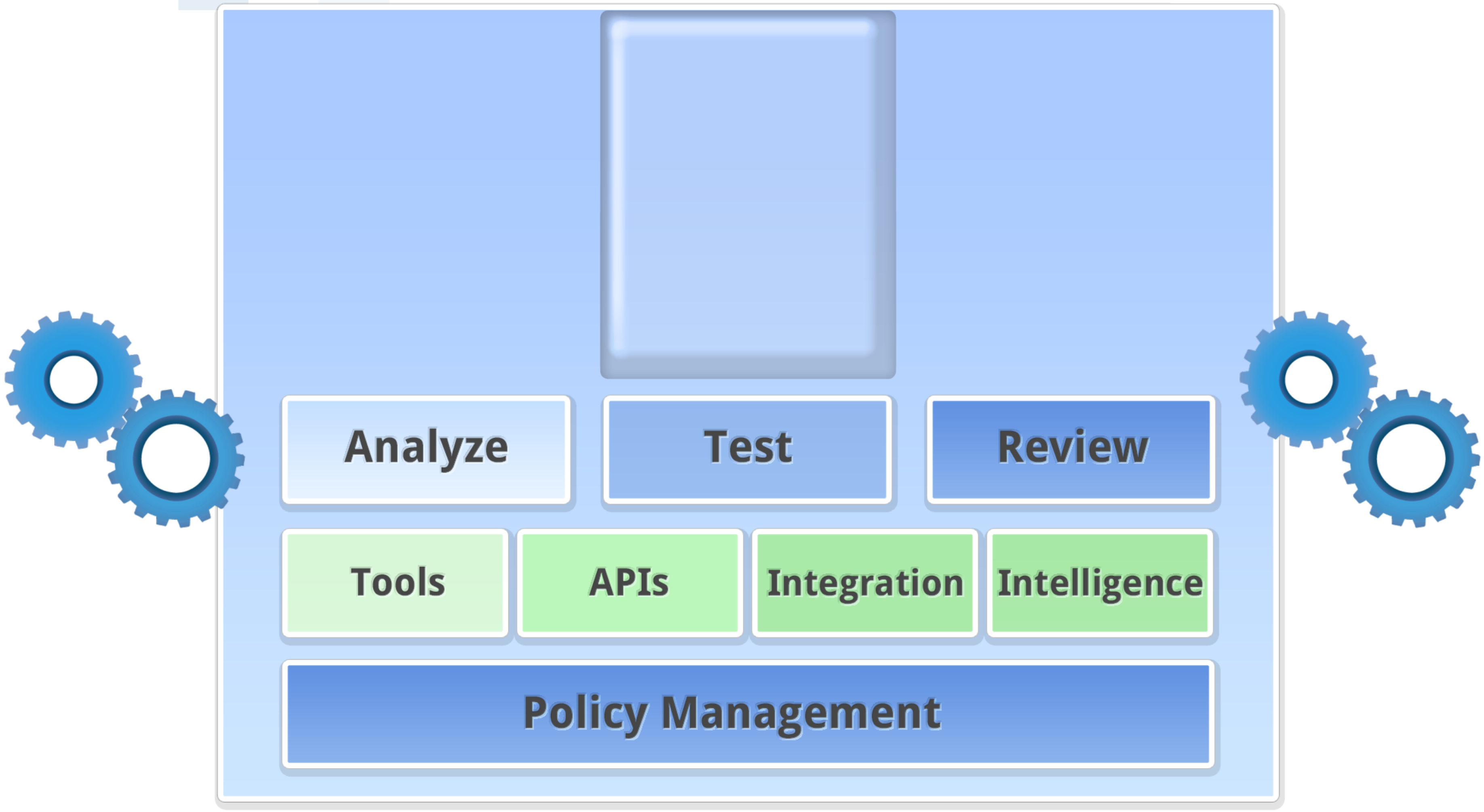
Preventing SQL Injection

```
String username = request.getParameter("USER");
String password = request.getParameter("PASSWORD");
String query = "SELECT * FROM Users WHERE username='" +
                  username + "' AND password='" + password + "'";
Statement.execute(query);

An attacker passes "' or 1=1" for username creating:
SELECT * FROM Users WHERE username='' or 1=1 AND password='foo'


Prevention: wrap input in validation:

String username = validate(request.getParameter("USER"));
String password = validate(request.getParameter("PASSWORD"));
```

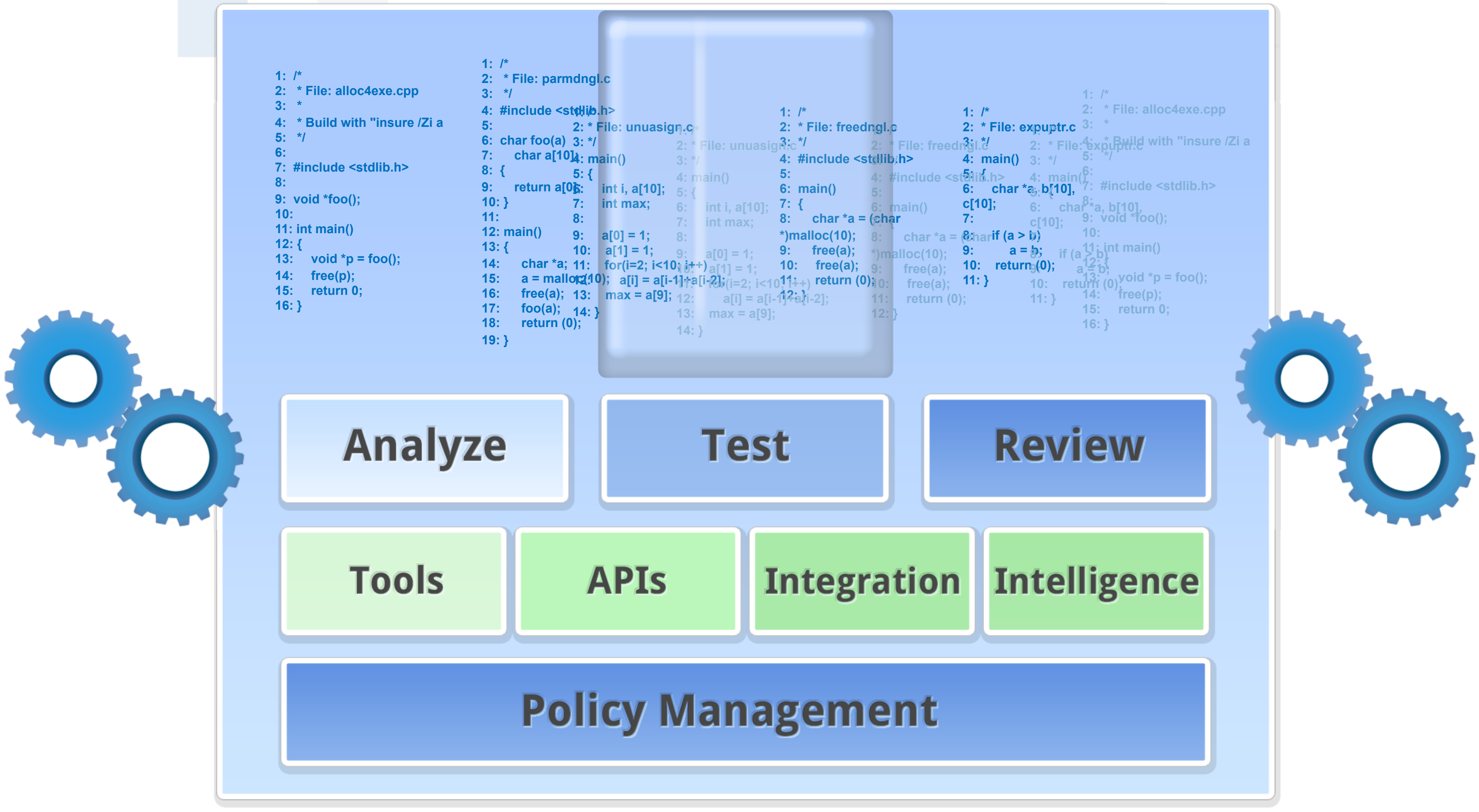
An Ounce of Prevention

- Feedback loop
  - Frequent issues from Peer Review
  - Field reports
  - QA findings
- The misunderstood relationship between flow analysis and pattern-based static analysis
- Creating a software environment where bugs cannot survive.

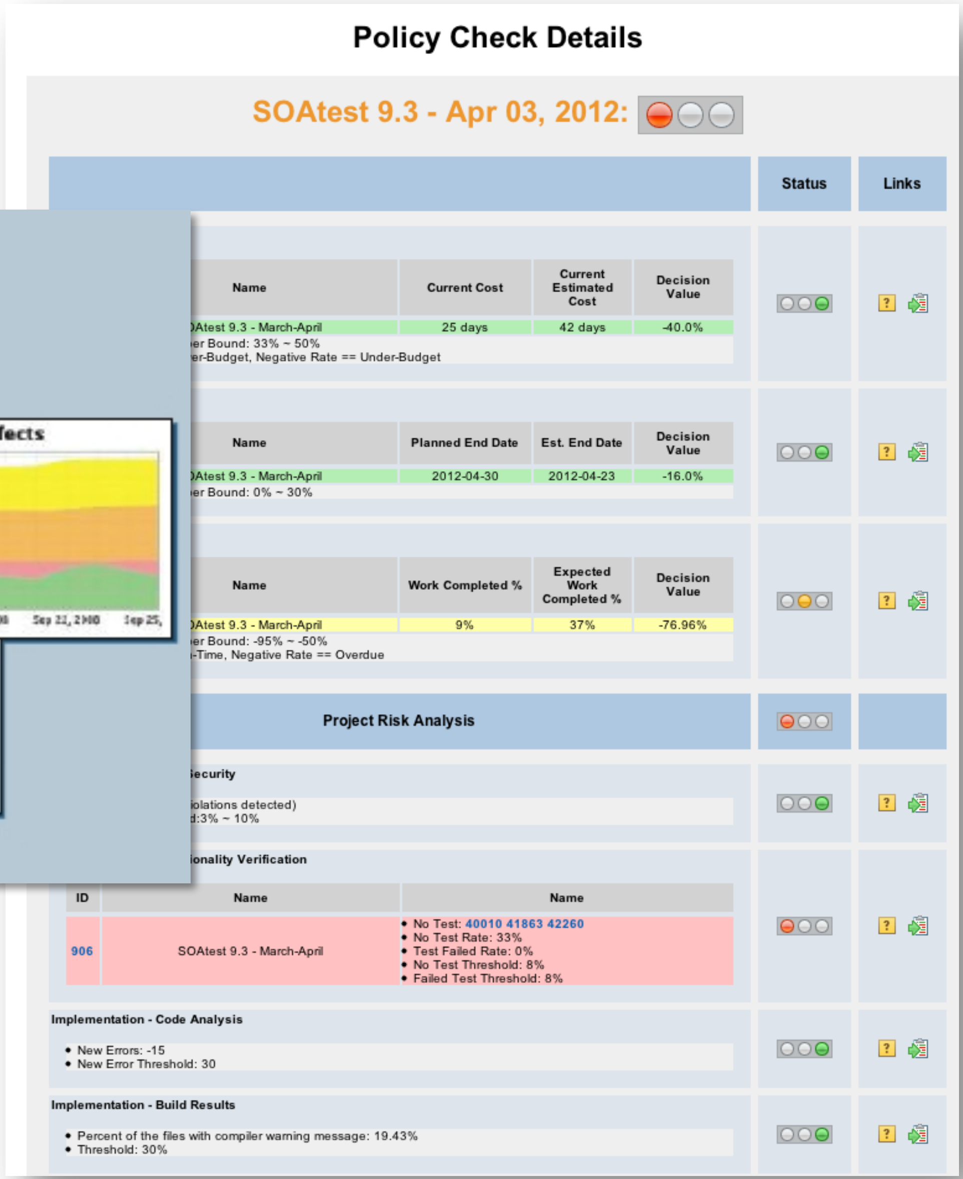
Parasoft Development Testing Platform



Parasoft Development Testing Platform



Monitor / Manage



Policy Check Details

SOAtest 9.3 - Apr 03, 2012:

	Status	Links

Name	Current Cost	Current Estimated Cost	Decision Value
)Atest 9.3 - March-April	25 days	42 days	-40.0%

er Bound: 33% ~ 50%
er-Budget, Negative Rate == Under-Budget

Name	Planned End Date	Est. End Date	Decision Value
)Atest 9.3 - March-April	2012-04-30	2012-04-23	-16.0%

er Bound: 0% ~ 30%

Name	Work Completed %	Expected Work Completed %	Decision Value
)Atest 9.3 - March-April	9%	37%	-76.96%

er Bound: -95% ~ -50%
-Time, Negative Rate == Overdue

Project Risk Analysis

ecurity

iolations detected)
d:3% ~ 10%

ionality Verification

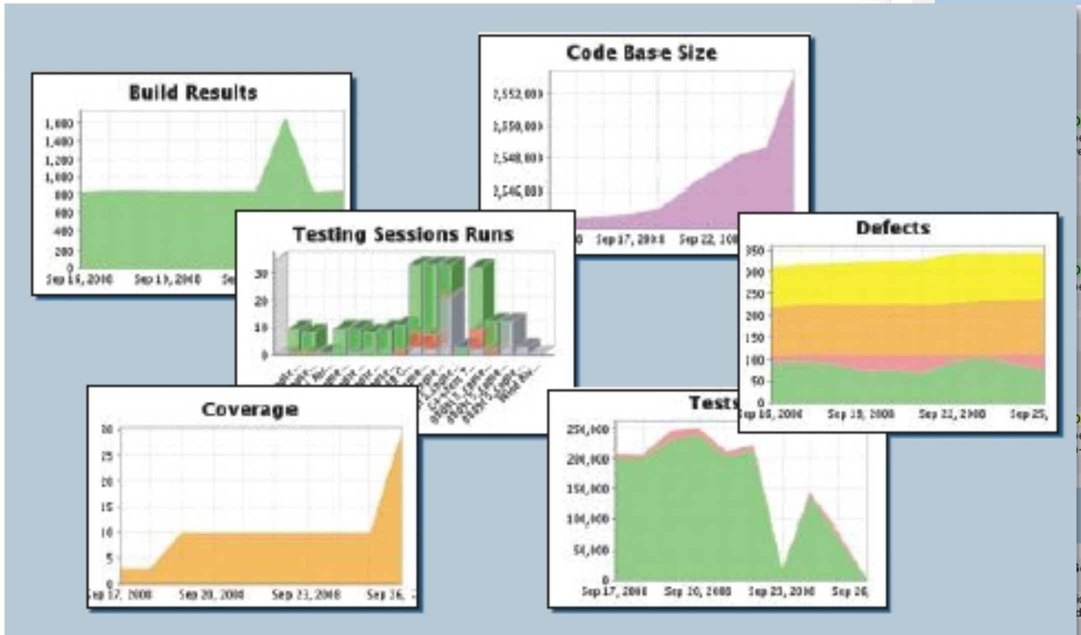
ID	Name	Name
906	SOAtest 9.3 - March-April	• No Test: 40010 41863 42260 • No Test Rate: 33% • Test Failed Rate: 0% • No Test Threshold: 8% • Failed Test Threshold: 8%

Implementation - Code Analysis

- New Errors: -15
- New Error Threshold: 30

Implementation - Build Results

- Percent of the files with compiler warning message: 19.43%
- Threshold: 30%



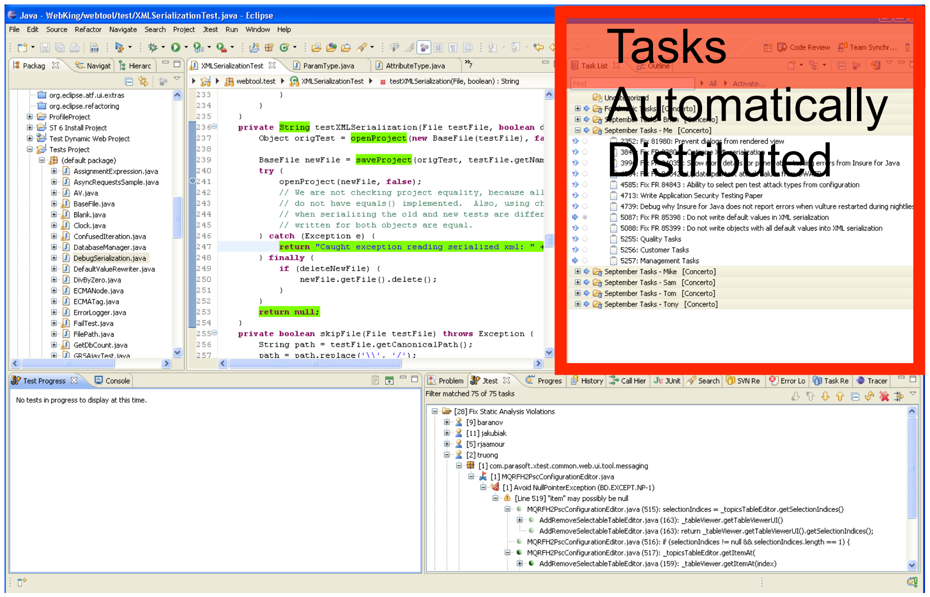
Dashboards track key development metrics

Static Analysis for Process Improvement

- Bug-finding rules have corresponding prevention rules – find all the vulnerabilities
- Prevent the potential rather than chase paths
- Analyze issues and select SA based on real problems:
  - Q/A issues
  - Bug-tracking
  - Flow analysis findings

From the platform to the desktop

- Email ≠ IDE
- SCM integration critical to efficiency
- Assign to last person to touch the code
- Send relevant tasks directly to developer UI


Tasks
Automatically
Distributed
Java - WebKing/webtool/test/XMLSerializationTest.java - Eclipse
File Edit Source Refactor Navigate Search Project Jtest Run Window Help
Packag Navigat Hierarc
XMLSerializationTest ParamType.java AttributeType.java
webtool.test XMLSerializationTest testXMLSerialization(File, boolean) : String
Code Review Team Synchr...
Task List Outline
Find All Activate...
Test Progress Console
Problem Jtest Progres History Call Hier JUnit Search SVN Re Error Lo Task Re Tracer

org.eclipse.atf.ui.extras
org.eclipse.refactoring
ProfileProject
ST 6 Install Project
Test Dynamic Web Project
Tests Project
(default package)
AssignmentExpression.java
AsyncRequestsSample.java
AV.java
BaseFile.java
Blank.java
Clock.java
ConfusedIteration.java
DatabaseManager.java
DebugSerialization.java
DefaultValueRewriter.java
DivByZero.java
ECMANode.java
ECMATag.java
ErrorLogger.java
FailTest.java
FilePath.java
GetDbCount.java
GRSAjaxTest.java

```
            }
        }
    }
    private String testXMLSerialization(File testFile, boolean c
        Object origTest = openProject(new BaseFile(testFile), fa

        BaseFile newFile = saveProject(origTest, testFile.getNam
        try {
            openProject(newFile, false);
            // We are not checking project equality, because all
            // do not have equals() implemented.  Also, using ch
            // when serializing the old and new tests are differ
            // written for both objects are equal.
        } catch (Exception e) {
            return "Caught exception reading serialized xml: " +
        } finally {
            if (deleteNewFile) {
                newFile.getFile().delete();
            }
        }
        return null;
    }
    private boolean skipFile(File testFile) throws Exception {
        String path = testFile.getCanonicalPath();
        path = path.replace('\\', '/');
```

Uncategorized
Fo[illegible]ic Tasks [Concerto]
September Tasks - Brian [Concerto]
September Tasks - Me [Concerto]
2352: Fix 81980: Prevent dialogs from rendered view
384[illegible]: Fix FR [illegible]: Optimize XML serialization
399[illegible]: Fix FR 84035: Show more details for [illegible] errors from Insure for Java
[illegible]: Fix FR [illegible]: Update [illegible] OWASP
4585: Fix FR 84843 : Ability to select pen test attack types from configuration
4713: Write Application Security Testing Paper
4739: Debug why Insure for Java does not report errors when vulture restarted during nightlies
5087: Fix FR 85398 : Do not write default values in XML serialization
5088: Fix FR 85399 : Do not write objects with all default values into XML serialization
5255: Quality Tasks
5256: Customer Tasks
5257: Management Tasks
September Tasks - Mike [Concerto]
September Tasks - Sam [Concerto]
September Tasks - Tom [Concerto]
September Tasks - Tony [Concerto]

No tests in progress to display at this time.

Filter matched 75 of 75 tasks

```
[28] Fix Static Analysis Violations
    [9] baranov
    [11] jakubiak
    [5] rjaamour
    [2] truong
        [1] com.parasoft.xtest.common.web.ui.tool.messaging
            [1] MQRFH2PscConfigurationEditor.java
                [1] Avoid NullPointerException (BD.EXCEPT.NP-1)
                    [Line 519] "item" may possibly be null
                        MQRFH2PscConfigurationEditor.java (515): selectionIndices = _topicsTableEditor.getSelectionIndices()
                            AddRemoveSelectableTableEditor.java (163): _tableViewer.getTableViewerUI()
                            AddRemoveSelectableTableEditor.java (163): return _tableViewer.getTableViewerUI().getSelectionIndices();
                        MQRFH2PscConfigurationEditor.java (516): if (selectionIndices != null && selectionIndices.length == 1) {
                        MQRFH2PscConfigurationEditor.java (517): _topicsTableEditor.getItemAt(
                            AddRemoveSelectableTableEditor.java (159): _tableViewer.getItemAt(index)
```

A Continuous Improvement Infrastructure that Delivers Greater Productivity and Software Quality

Automation of critical repetitive tasks

▪ Expose structural errors ▪ Eliminate classes of errors ▪ Expose poorly-implemented requirements	▪ Reduce rework time ▪ Allow for more complete "testing" ▪ Reduce business risk

Infrastructure to test incomplete/evolving systems

▪ Effectively implement a requirement ▪ Framework for Continuous Quality ▪ Test logical "units" of work ▪ Understand Process impact to quality	▪ Quality delivered with agility ▪ Reduce the risk of change ▪ Deliver new requirements faster ▪ Predictable outcomes

Reuse of quality assets throughout SDLC

▪ Quality is a continuous process ▪ Process leverages up-stream assets	▪ Reduce manual efforts ▪ Eliminate testing re-rework

Establish and monitor quality processes

▪ Framework for continuous improvement	▪ Visibility

Q&A / Resources

- webinar@parasoft.com
- http://alm.parasoft.com
- http://www.parasoft.com/jsp/resources

- Facebook: https://www.facebook.com/parasoftcorporation
- Twitter: @Parasoft @MustRead4Dev @CodeCurmudgeon
- LinkedIn: http://www.linkedin.com/company/parasoft
- Google+ Community: *Static Analysis for Fun and Profit*