Introducing Static Analysis in a Mature Codebase

PNSQC 2012

John Ruberto
Thank You!

- Reviewers
  - Robert Cohn
  - Deian Tabakov
Quick Poll

- How many have used Static Analysis?
- Opinion: was it successful?
Born out of failure

- **Telcom EMS/NMS system: 6M LOC C++**
  - Very successful in the market
  - 400K issues found through static analysis
  - Many false positives, each issue became a debate

- **Financial Web App: 1M LOC Java and javascript**
  - Very successful in the market
  - 20K issues found through static analysis
  - Each one to be reviewed, owner identified, communicated
  - Effort to manage the pile became overwhelming
What is Static Analysis?

According to Black & Mitchell*

Analysis of software artifacts, e.g., requirements or code, carried out **without execution** of these software development artifacts. Static analysis is usually carried out by means of a supporting tool.

*Black, Rex; Mitchell, Jamie; *Advanced Software Testing – Volume 3*, Rocky Nook, page 264
Example

- Null Pointer Exception:

```java
boolean sparse = subset.isFullResponse();

if (subset != null) {
    // do something here, but too late
```
if (((Boolean) mReportOptions.getBinaryVal(TOTAL, Boolean.TRUE)).booleanValue()) {
  if (dc.getCredit() != null || dc.getDebit() != null) {
    gt = new GrandTotalSection("TOTAL", this);
    dc.addToSectionSubtotal(gt);
  }
}

public Object getBinaryVal(String optionName, Object defaultVal) {
  ReportOption opt = mReportValues.get(optionName);
  if (opt == null) {
    return null;
  }
  return opt;
}

public static DateRange getDateRangeVal(String macro) {
  DateRangeMacro dateRangeMacro = DATERANGEMACROMAP.get(macro);
  if (dateRangeMacro != null) {
    return dateRangeMacro.getDateRange();
  }
  if (macro.equals(MACRO_CUSTOM)) {
    return null;
  } else if (macro.equals(MACRO_LASTPAYCHECKDATE)) {
    QBDate last = null;
    try {
      last = ROPrefs.getPrefs().getLastPaycheckDate();
    } catch (Exception e) {
    }
    return new DateRange(last, last, DateAdder.ADD_DAYS);
  }
  return DateRange.current(DateAdder.ADD_MONTHS);
Why use Static Analysis?

- **Root Cause Analysis:** defects found in system test

  - Coding errors: 39%
  - Coding: Exception Handling: 19%
  - Coding: Missing: 9%
  - Design: Functional: 4%
  - Design: Usability: 6%
  - Content Localization: 1%
  - Requirements: Missing: 9%
  - Requirements: Wrong: 3%
How to improve our coding practices

- Brain storm with developers:
  - Improve our unit test & code review practices
  - Introduce Static Analysis

- According to Jones*, 97% of defects can be removed by using these practices:
  - Code Review, Static Analysis, Unit Test

Introducing Static Analysis

- Process Design
- Tool Selection
- Deployment
- Change Management
- Backlog Reduction
Process Design

- Minimize feedback loop
  - Creation -> Detection
- Focus on new issues
- Alert developers privately
  - Focus on developer productivity
- Track issue status
- Efficient management of issues
Process Design

Static Analysis

Analysis happens before code check-in

Clean code is delivered to test

Static Analysis Reports

Build Deploy

SCM
Process Design, iteration 2

Analysis happens in parallel with build
Tool Selection

- **Tool Requirements**
  - Effectiveness with our development language
  - False positive rate with our code base
  - Compatibility with our development environment
  - Ability to uniquely identify defects
  - Either a built-in defect management system, or the ability to integrate with our system through an API
Tool Evaluation - Criteria

- **Efficacy of the checkers**
  - How powerful the tool is for finding interesting defects
  - *Interesting = defects that would result in runtime error*

- **False positive rate**
  - Tool indicates a defect, where there is no defect

- **False negative rate**
  - Tool doesn’t find a defect, which indeed exists

- **List of tools:** blog.ruberto.com
Resulted in two tools with complimentary results:

<table>
<thead>
<tr>
<th>Resource Leaks</th>
<th>Findbugs</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS com.intuit.qbo.bl.payrollform.EditPayrollForm.getBytesFromFile(File) may fail to close stream [line 209]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>OS com.intuit.qbo.controller.util.AuthHeaderHelper.getBodyAsString(HttpMethod) may fail to close stream [line 413]</td>
<td>FP</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.monitor.LoadBalance.runCommand(String[]) may fail to close stream [line 214]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.service.provisioning.partnerconfig.PartnerConfig.initialize() may fail to close stream [line 354]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>OS com.intuit.qbo.ui.common.UiCommonFunctions.includeUIFiles(String) may fail to close stream [line 1604]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>OS com.intuit.qbo.ui.help.index.SiteIndex.run() may fail to close stream [line 252]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.salsa.ui.login.CachedAuthToken.CompanyInfo.loadFromFile(String) returns from routine while resource still open [line 69]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.xml.XmlVendor.main(String[] args) returns from routine while resource still open [line 1104]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.util.ResourceAccessGateKeeperDisableTest.touchResourceFile() returns from routine while resource still open [line 416]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.util.ResourceAccessGateKeeperDisableTest.touchResourceFile() returns from routine while resource still open [line 605]</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Possible Null Pointer Dereferences</th>
<th>Findbugs</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Possible null pointer dereference of defaultServiceItem in com.intuit.qbo.bl.comgmt.CreateCompany.addDefaultTimeItemToPrefs() [line 580]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Possible null pointer dereference of zipCode in com.intuit.qbo.bl.comgmt.QBOCompanyManager.postCreationSetup(QBOCompanyManager$CreateCompanyDataInfo boolean) [line 1953]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Null pointer dereference of aDB in com.intuit.qbo.bl.recur.EditRecurInfo main(String[]) [line 917]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Possible null pointer dereference of response in com.intuit.qbo.bl.core.Trans.TransNotification.invokeRequest(HttpClient, String, String) [line 244]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Possible null pointer dereference of checkDate in com.intuit.qbo.ui.transactions.payroll.payemployees.PayEmployeesConstants.JAVASCRIPT_SEND_PAYCHECKS(Integer, Integer, QBDate, QBDate, QBDate)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.service.util.mapping.SalesReceiptMapper.mapSalesReceiptCdnTypeToQboType(SalesReceipt) possible null pointer dereference of detail [line 124]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.ui.interviews.mini.MinInterviewHandler.handleRequest() possible null pointer dereference of first_mip [line 135]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.ui.mobi.iphone.IPhoneHandler.setLoginHelpMessage() possible null pointer dereference of locale [line 3616]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.billing.util.RenewalRenewUtil.send(int, boolean) dereference before null check [line 270]</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>com.intuit.qbo.ui.transactions.purchase.creditcard.core.CreditCardHandler.handleRequest() dereference before null check [line 144]</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Deployment

- Integrated into our build system
- Initial scans: establish baseline
- Scripting for issue assignment & notification
- Training
- Follow up, fixing glitches, etc.

- 4 weeks of effort, with 2 weeks engagement by professional services
Automated Issue Management

- Each nightly scan results in new issues & issues that no longer appear

- For each new issue:
  - Look up the developer that checked in that file
  - Assign the issue to the developer
  - Notify via email, with link to issue in the tool

- For issues no longer appearing in the scans
  - Mark them fixed in the defect system
Weekly Reports

- Regular communication is important for people to see the value, and keep the change alive
- Focus on action

<table>
<thead>
<tr>
<th>Owner</th>
<th>Major</th>
<th>Moderate</th>
<th>Minor</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amit</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Andy</td>
<td>1</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>David</td>
<td>2</td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Frank</td>
<td>2</td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Mary</td>
<td></td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Praveen</td>
<td>1</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Sandeep</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Sunil</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Vitor</td>
<td>1</td>
<td>1</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>6</strong></td>
<td><strong>7</strong></td>
<td><strong>7</strong></td>
<td><strong>20</strong></td>
</tr>
</tbody>
</table>
Monthly Reports

- Focus on Return on Investment
- Reinforce the “new code” focus
Backlog Reduction

- First, build confidence in tool & issues
- Then, let the games begin

During the Olympics:
- 10 pts each major
- 5 pts each moderate
- 1 pt each minor & each close/no fix
- Awards:
  - Gold, Silver, Bronze

Lottery:
- 1 “ticket” for each fix
- Regular drawings
- Regular posting of winners & leaderboard

Image: Creative Commons: http://www.flickr.com/photos/garryknight/
Results

- Focus on new issues first, lead to confidence in the tools and effort to eliminate the backlog

**Static Analysis Issue Status**

- Red line: Total issues
- Green line: Fixed issues

<table>
<thead>
<tr>
<th>Month</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>0</td>
</tr>
<tr>
<td>Feb</td>
<td>0</td>
</tr>
<tr>
<td>Mar</td>
<td>0</td>
</tr>
<tr>
<td>Apr</td>
<td>0</td>
</tr>
<tr>
<td>May</td>
<td>0</td>
</tr>
<tr>
<td>Jun</td>
<td>0</td>
</tr>
<tr>
<td>Jul</td>
<td>0</td>
</tr>
<tr>
<td>Aug</td>
<td>0</td>
</tr>
<tr>
<td>Sep</td>
<td>0</td>
</tr>
</tbody>
</table>
Results

- Better experience for customers
  - Fewer failures occurring in production

- Improved Code Quality
  - > 97% - 100% of issues fixed before customer release

- Reduced Technical Debt
  - Eliminated over 3000 issues
Results – Developer satisfaction

- fix coverity 17388: dereference after null check, happens after failure to undo excluded txns in DTX. This really did need to be fixed, good catch coverity.

- In my case I was referencing a variable that was null (it was set to null way above in the code - so I never realized that) and I was referencing it exception handling code. Stupid me. But this was missed in code review. I do not think Coverity can replace code review but in code review, folks tend to focus more on design and running logic than looking out for NPE (that is at least my observation)
Five Tips

- Engage Developers through the entire process
  - After all, we are doing this for developers
- Choose the right tool for your codebase
- Focus on new code, let “legacy” follow
- Focus on developer productivity, not finding fault
- Automate issue management
  - Triage, Assignment, Verification, Closure
Five Benefits

- 100% code coverage, every night
- Common errors found cheaply, supplements code review & unit test
- Feedback loop results in better coding skills
- Identifies the exact line of code with the error
- Objective closure criteria, its fixed or not fixed.
Thank You

- List of Static Analysis Tools
  - Blog.ruberto.com

- Contact me:
  - John_Ruberto@intuit.com / JohnRuberto@gmail.com
  - Blog.ruberto.com
  - @JohnRuberto