

# OSP STAGE 3<sup>rd</sup> Cycle for : Project Fluxvator



**Static Analysis Follow-up**

**System Tests Follow-up**

**Misc. Change Notes**

**Closing Remarks**

# Static Analysis :

## The follow-up

# 1. Feedback Summary

Static Analysis : the Follow-up



# 1. Feedback Summary

QA Team's primary tool : CodePro

CodePro Audit : **660** counts of violation

CodePro Metric : 0 counts of violation

Static Analysis : the Follow-up



1. Feedback Summary  
Of the 660 violations,  
65 fell in HIGH,  
**559** in MEDIUM,  
and 36 in LOW categories  
In regards to their severity.

Static Analysis : the Follow-up



# 1. Feedback Summary

## Violated rules are:

Violations by Audit Rule		
<a href="#"><u>Always Override toString</u></a>	<a href="#"><u>File Comment</u></a>	<a href="#"><u>String Concatenation</u></a>
<a href="#"><u>Caught Exceptions</u></a>	<a href="#"><u>Import Order</u></a>	<a href="#"><u>String Literals</u></a>
<a href="#"><u>Constants in Comparison</u></a>	<a href="#"><u>Local Declarations</u></a>	<a href="#"><u>Synchronized Method</u></a>
<a href="#"><u>Dangling Else</u></a>	<a href="#"><u>Method Javadoc Conventions</u></a>	<a href="#"><u>Too Many Violations</u></a>
<a href="#"><u>Debugging Code</u></a>	<a href="#"><u>Missing Block</u></a>	<a href="#"><u>Type Javadoc Conventions</u></a>
<a href="#"><u>Declare Default Constructors</u></a>	<a href="#"><u>Missing Default in Switch</u></a>	<a href="#"><u>Unnecessary Exceptions</u></a>
<a href="#"><u>Declared Exceptions</u></a>	<a href="#"><u>No Spelling Dictionary</u></a>	<a href="#"><u>Use Compound Assignment</u></a>
<a href="#"><u>Empty Catch Clause</u></a>	<a href="#"><u>Non-terminated Case Clause</u></a>	<a href="#"><u>Use equals() Rather Than ==</u></a>
<a href="#"><u>Empty If Statement</u></a>	<a href="#"><u>Numeric Literals</u></a>	<a href="#"><u>Variable Declared Within a Loop</u></a>
<a href="#"><u>Explicit "this" Usage</u></a>	<a href="#"><u>Platform Specific Line Separator</u></a>	<a href="#"><u>Variable Should Be Final</u></a>
<a href="#"><u>Field Javadoc Conventions</u></a>	<a href="#"><u>Static Field Naming Convention</u></a>	

Static Analysis : the Follow-up



# 1. Feedback Summary

## Other opinions from the QA Team:

- Lack of Unit Test Coverage in Some Classes
  - Circular Dependency Issues
  - Repeated Use of Static Variables

Overall Clean, Albeit Ridden with Violations

Static Analysis : the Follow-up





## 2. The Response

Static Analysis : the Follow-up



## 2. The Response

Declined Feedback Suggestions

Accepted Feedback Suggestions

Static Analysis : the Follow-up



# Declined Feedback Suggestions

## Violations by Audit Rule

<a href="#">Always Override toString</a>	<a href="#">File Comment</a>	<a href="#">String Concatenation</a>
<a href="#">Caught Exceptions</a>	<a href="#">Import Order</a>	<a href="#">String Literals</a>
<a href="#">Constants in Comparison</a>	<a href="#">Local Declarations</a>	<a href="#">Synchronized Method</a>
<a href="#">Dangling Else</a>	<a href="#">Method Javadoc Conventions</a>	<a href="#">Too Many Violations</a>
<a href="#">Debugging Code</a>	<a href="#">Missing Block</a>	<a href="#">Type Javadoc Conventions</a>
<a href="#">Declare Default Constructors</a>	<a href="#">Missing Default in Switch</a>	<a href="#">Unnecessary Exceptions</a>
<a href="#">Declared Exceptions</a>	<a href="#">No Spelling Dictionary</a>	<a href="#">Use Compound Assignment</a>
<a href="#">Empty Catch Clause</a>	<a href="#">Non-terminated Case Clause</a>	<a href="#">Use equals() Rather Than ==</a>
<a href="#">Empty If Statement</a>	<a href="#">Numeric Literals</a>	<a href="#">Variable Declared Within a Loop</a>
<a href="#">Explicit "this" Usage</a>	<a href="#">Platform Specific Line Separator</a>	<a href="#">Variable Should Be Final</a>
<a href="#">Field Javadoc Conventions</a>	<a href="#">Static Field Naming Convention</a>	

Static Analysis : the Follow-up



## Declined Feedback Suggestions

Our reasons being:

- Low impact/priority: comments, import order...
- “Intended” violations : local declarations, empty catch...
- Allotted time issues: synchronized

Static Analysis : the Follow-up



## Declined Feedback Suggestions:

-Unit test Coverage-

Our code structure required referencing  
across multiple classes,  
consequently making unit tests tricky  
save for the few classes we did.

Static Analysis : the Follow-up



# Accepted Feedback Suggestions

## Violations by Audit Rule

<a href="#">Always Override toString</a>	<a href="#">File Comment</a>	<a href="#">String Concatenation</a>
<a href="#">Caught Exceptions</a>	<a href="#">Import Order</a>	<a href="#">String Literals</a>
<a href="#">Constants in Comparison</a>	<a href="#">Local Declarations</a>	<a href="#">Synchronized Method</a>
<a href="#">Dangling Else</a>	<a href="#">Method Javadoc Conventions</a>	<a href="#">Too Many Violations</a>
<a href="#">Debugging Code</a>	<a href="#">Missing Block</a>	<a href="#">Type Javadoc Conventions</a>
<a href="#">Declare Default Constructors</a>	<a href="#">Missing Default in Switch</a>	<a href="#">Unnecessary Exceptions</a>
<a href="#">Declared Exceptions</a>	<a href="#">No Spelling Dictionary</a>	<a href="#">Use Compound Assignment</a>
<a href="#">Empty Catch Clause</a>	<a href="#">Non-terminated Case Clause</a>	<a href="#">Use equals() Rather Than ==</a>
<a href="#">Empty If Statement</a>	<a href="#">Numeric Literals</a>	<a href="#">Variable Declared Within a Loop</a>
<a href="#">Explicit "this" Usage</a>	<a href="#">Platform Specific Line Separator</a>	<a href="#">Variable Should Be Final</a>
<a href="#">Field Javadoc Conventions</a>	<a href="#">Static Field Naming Convention</a>	

Static Analysis : the Follow-up



## Accepted Feedback Suggestions

Most notably, static method/variable violations had strong relevance to our only undocumented system error.

Static Analysis : the Follow-up



# Accepted Feedback Suggestions

We resolved those by:

- **Removing static properties** from QueueAlgorithm Class, reverting them into normal variables.
- Slightly **altering code structures** in addLoad(), update() method.
- **Abstaining from making more violations** in further code modifications.

Static Analysis : the Follow-up





# System Tests: The Follow-up

# 1. Feedback Summary

System Tests : the Follow-up



# 1. Feedback Summary

..Surprisingly, unlike the first, our second cycle reportedly did not yield any significant system errors.

System Tests : the Follow-up



# 1. Feedback Summary

Nevertheless, there was one undocumented error that our dev team had caught prior to 2<sup>nd</sup> testing phase:

One of the elevators had problem unloading passengers during emergency protocol.

System Tests : the Follow-up



## 1. Feedback Summary

The QA Team did not document this issue, because the **dev team had informed them** beforehand for brief discussion.

System Tests : the Follow-up



# 1. Feedback Summary

This error was solved while **following up on the static methods/variable violations** from static analysis.

System Tests : the Follow-up



# Misc. Change Notes

Other changes we've made to the code include:

Adaptation of the **Observer pattern**

(in junction with our follow-up in static analysis)

Misc. Change Notes





# Closing Remarks

# Impressions

(Gathered from all teammates)

Closing Remarks



# General Impressions from Class

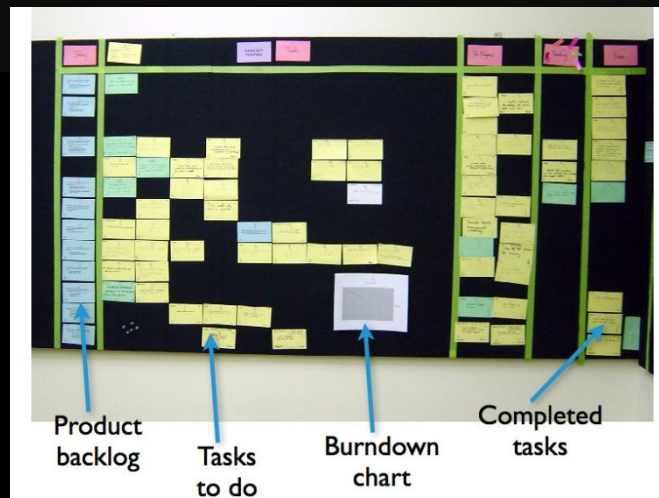
- Satisfied with what we've accomplished so far
- Learned more about concepts such as design patterns
- Hope to further apply software techniques in future projects

Closing Remarks



# Impressions : OSP

- Documents : proven worthy for data recording, not so much for keeping track of time & progress
- Found ourselves hardly looking up the documents:  
alternative methods for task management?  
(e.g. dedicated post-it boards)



# Impressions : SVN & other tools

- Centralized version control was awesome, aside from the issue of syncing library paths, development environment
  - Maybe use with project management tools?  
(e.g. Maven, Gradle)

Closing Remarks



# Impressions : Testing

- The existence of dedicated testing platforms was very new to us and amazing.
- We did some “brute force testing” on the dev side as well.
- The notion of trying hard to break our own program was entertaining.

Closing Remarks



# Impressions : Risk Management

- Individual factors(e.g. motivation) had their ups and downs.
  - We've underestimated the barrier of language in communication: conveying complex concepts to one another wasn't easy.

Closing Remarks



The End