

# Static Code Analysis with Jenkins

Ulli Hafner

June 2011

## Agenda

**Using static code analysis in Java projects**

Live demo of static analysis tool suite in Jenkins

## How to apply static analysis in projects

- ① Static code analysis improves code quality
- ① Two options are possible when using static analysis tools:
  - Monitor the results (in your IDE or Jenkins) and use them in code reviews
  - Enforce zero warning projects (i.e., fail a build if there are new warnings)

## How to apply static analysis in projects

### ① Drawbacks when only monitoring

- Broken window effect: if there are too many warnings, the developers will ignore them at some point
- It is difficult to distinguish new warnings from existing ones in reviews

### ② Advantages of zero warning projects

- Self-healing since developers don't want to be blamed for failed builds
- Easy detection of involved change sets

## Best practices to obtain zero warnings projects

### New Projects

1. Start with complete set of warning rules (CheckStyle, FindBugs, PMD)
2. Adapt rule sets to your coding guidelines
3. Fail a build as soon as there are new warnings
4. Discuss the warnings with the team (find an agreement to either fix the code or remove the affected warning rule)

### Existing projects

- a) Spend the time to fix these warnings
- b) Use a small subset of warning rules (e.g., only severity high)
- c) Use different rule sets for existing and new projects
- d) Do not increase the number of warnings

## A strategy to obtain zero warnings in existing projects

- ① In our team we use a combination of these practices
  - Check existing projects with all warning rules
  - Select the most important ones
  - Recheck projects with the reduced set of rules
  - Spend some time to reduce the number of important warnings
  - For the remaining warnings use suppression warning filters
- ② Policy when modifying code in one of the old modules
  - Consider old projects “read-only”
  - Move code to new projects if possible (dependencies)
  - Remove suppression warnings filters
  - Fix all warnings before check-in

## Agenda

Using static code analysis in Java projects

**Live demo of static analysis tool suite in Jenkins**

## Static Analysis Tools in Jenkins

- ① Checkstyle
- ① DRY (CPD/Simian)
- ① FindBugs
- ① PMD
- ① Compiler Warnings
- ① Open Tasks
- ① Analysis Collector
- ① Violations
- ① Emma/Cobertura
- ① CppCheck
- ① Codestyle
- ① Sonar
- ① ...