Upping the game

Improving your software development process

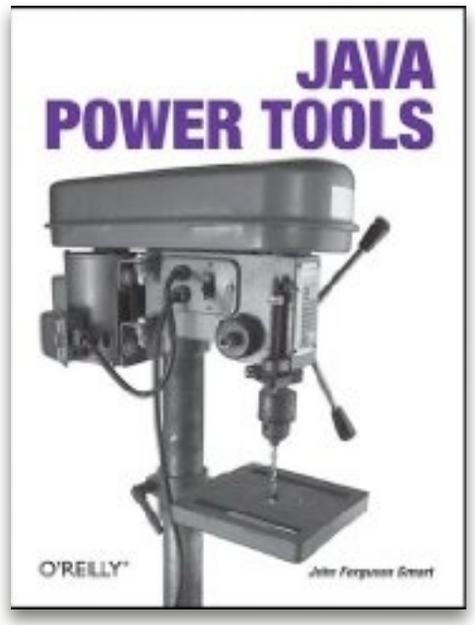
John Ferguson Smart
Principle Consultant
Wakaleo Consulting
Email: john.smart@wakaleo.com
Web: http://www.wakaleo.com
Twitter: wakaleo

Presentation Goals

Learn how to improve, harmonize and automate your development process using tools like Maven, Hudson, and many others.

Speaker's qualifications

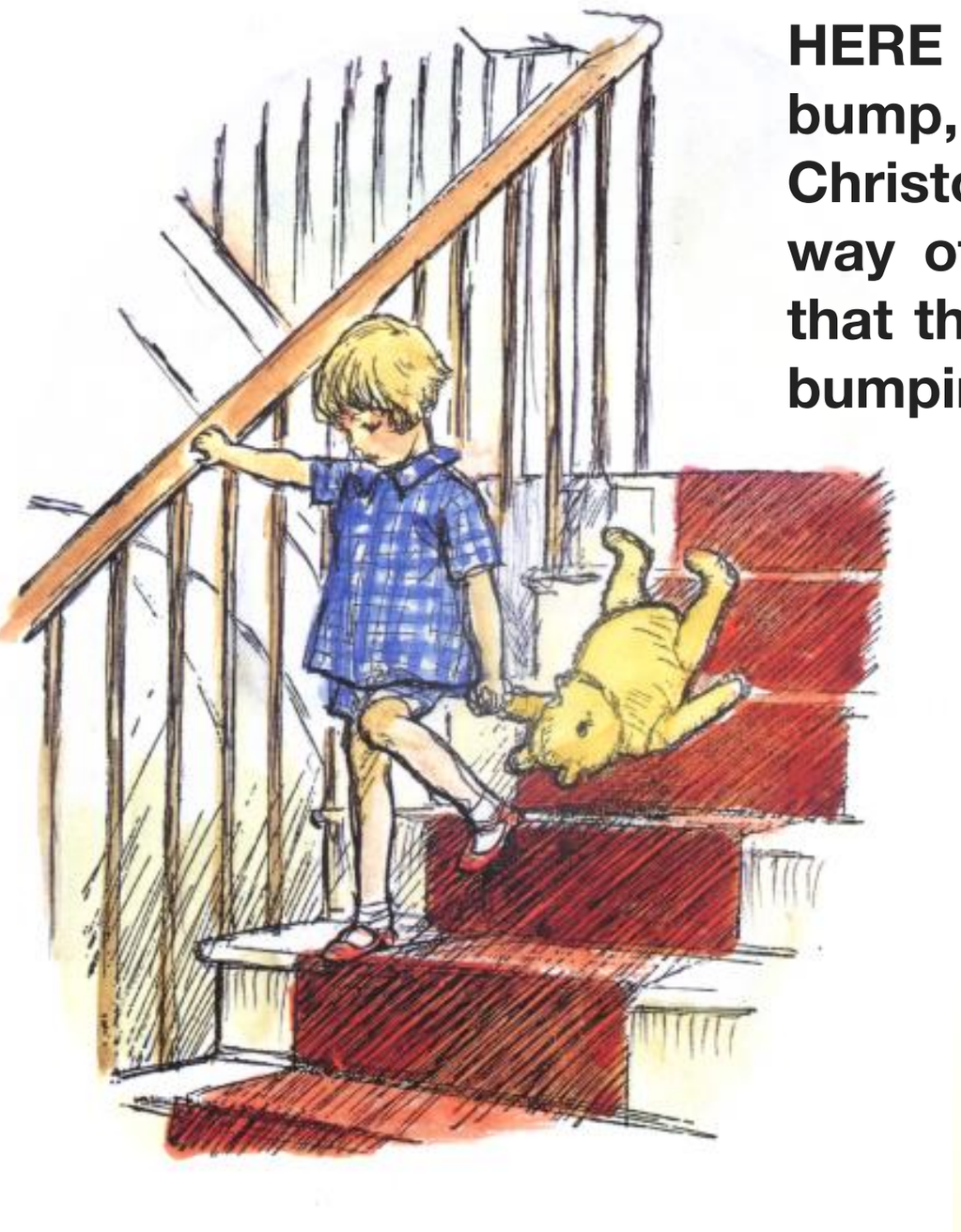
- John Ferguson Smart
  - Consultant, Trainer, Mentor, Author,...
  - Works with Enterprise Java, Web Development, and Open Source technologies
  - Author of 'Java Power Tools' (O'Reilly)
  - Writes articles for sites like JavaWorld, DevX and Java.net, and blogs on Java.net
  - Speaks at conferences, Java User Groups etc.
  - Likes to write about himself in the third person



Agenda

- What we will cover today:
  - Industrializing your build process
  - Automate your builds
  - Better testing practices
  - Reducing technical debt

Why improve?



HERE is Edward Bear, coming downstairs now, bump, bump, bump, on the back of his head, behind Christopher Robin. It is, as far as he knows, the only way of coming downstairs, but sometimes he feels that there really is another way, if only he could stop bumping for a moment and think of it.

-- A. A. Milne

Why should we improve?

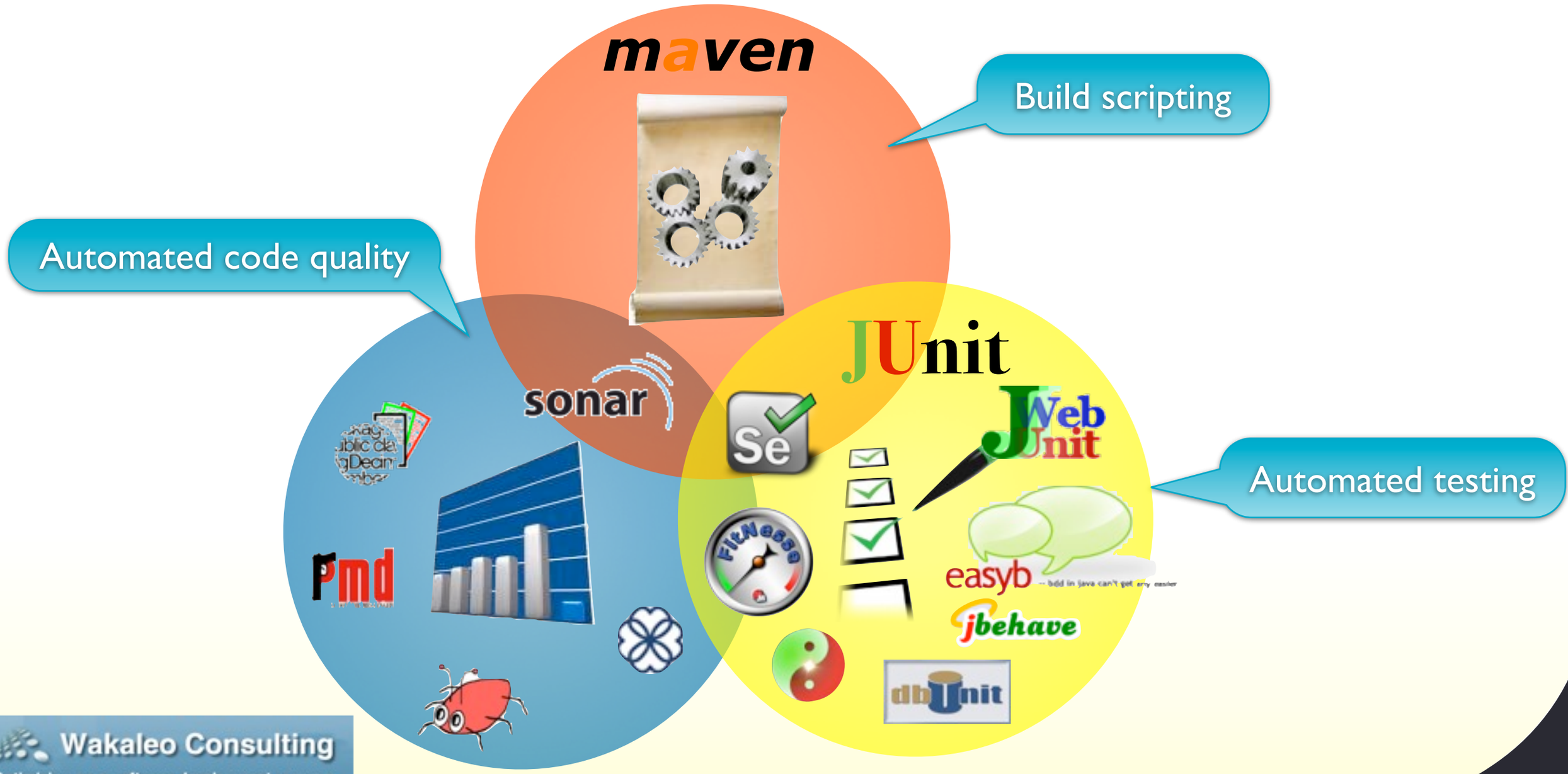
- Why should we improve our development process?
  - ☑ Lower development costs
  - ☑ Lower maintenance costs
  - ☑ Less bugs
  - ☑ Higher code quality
  - ☑ Be flexible - adapt to change more easily
  - ☑ Happier more productive users

How can we improve?

- How can we improve our development process?
  - Standardized build practices
  - Better testing practices
  - Better visibility
  - Faster feedback
  - Quality metrics
  - Automate!

Tools for the job

- There are plenty of tools available - and most are free!



Towards a better build process

- Standardize your build process with Maven
  - What is Maven, anyway?
    - A high-level open source build scripting framework
    - Extensively used in the Java world

Towards a better build process

- How does Maven help?
  - Standards
  - Conventions
  - Lower Maintenance Costs
  - Knowledge sharing
  - Dependency Management
  - Promoting good architecture

So how can Maven help me?

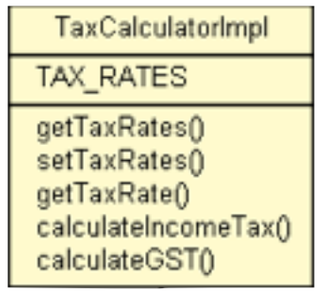
- Standards and Conventions
  - A standard directory structure
  - A standard, but extensible build lifecycle

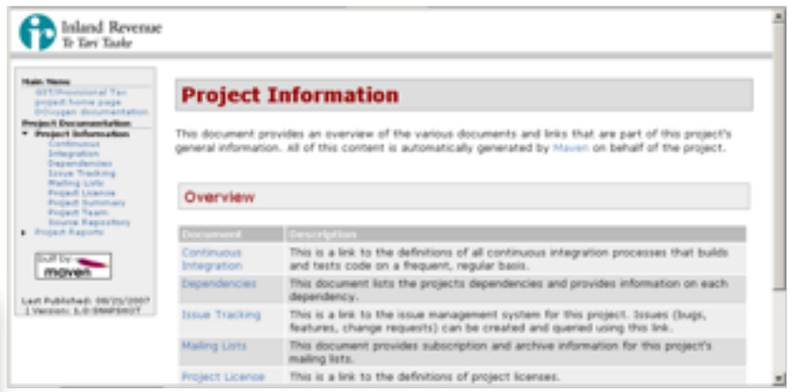
...and the build scripts are much easier to understand and maintain

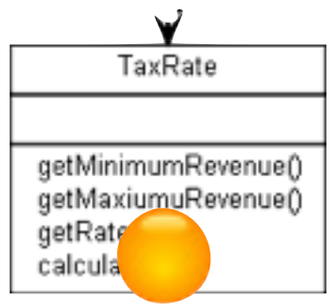
So new developers understand the project structure and build process straight away!

So how can Maven help me?

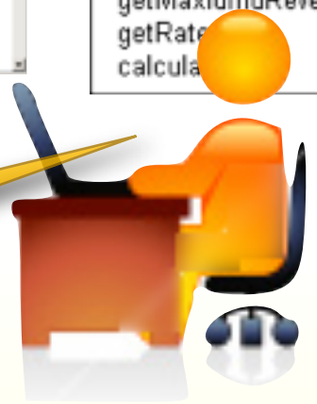
- Technical documentation
  - Generate technical project documentation
  - Easy to integrate code quality metrics







I can even generate UML diagrams in my Javadocs



And setting up code quality metrics is a breeze!

So how can Maven help me?

- Project architecture
  - Encourages developers to use modular design
  - More flexible architecture
  - Reduced complexity and maintenance costs

So how can Maven help me?

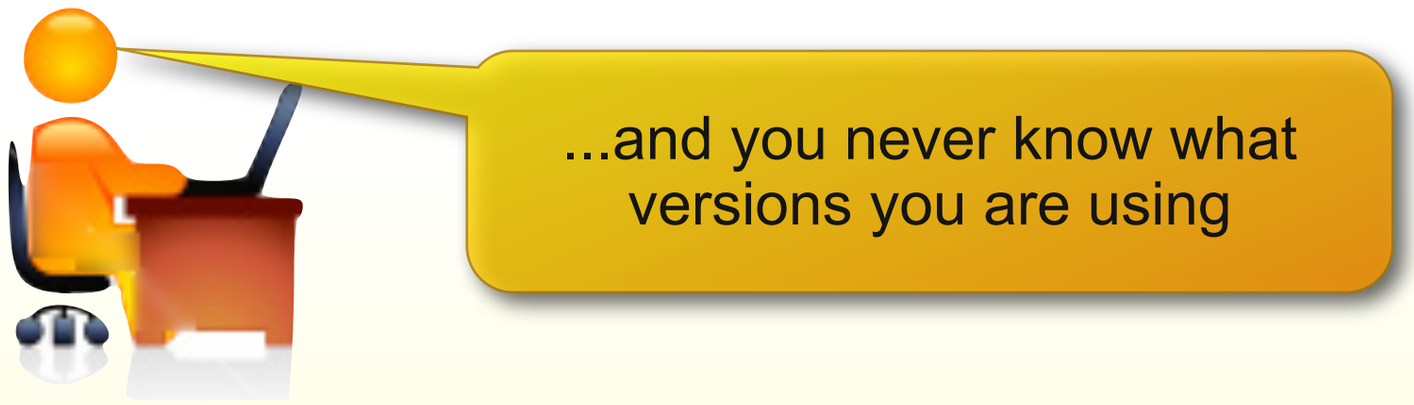
- Dependency Management
  - Understand precisely what libraries your application needs
  - Safer and more reproducible builds
  - A standard way to share internal libraries

All our libraries are shared and safely stored on a central server

...we just have to name the ones we need in our build script

So how can Maven help me?

- Dependency Management before Maven
  - Each project has its own set of JAR files
  - Unnecessary duplication
  - Hard to keep track of versions
  - Errors due to incompatible JAR files
  - Overloads the source code repository



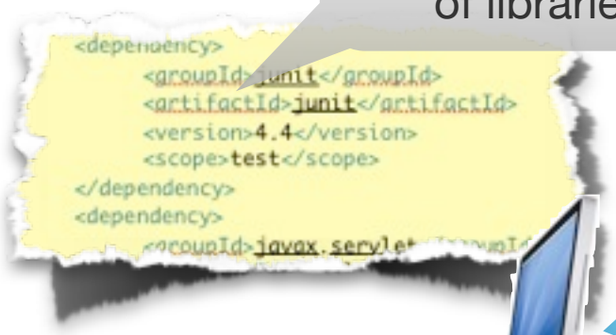
So how can Maven help me?

Dependency Management using Maven

- Library versions are stored on a central server
- Each project “declares” what libraries and versions it needs
- All the required dependencies are automatically downloaded
- The server is called a ‘Maven Enterprise Repository Manager’

Projects declare the versions of libraries they need

Libraries are stored on a central server



Libraries are downloaded as required

This way I know *exactly* what libraries my application uses

So how can Maven help me?

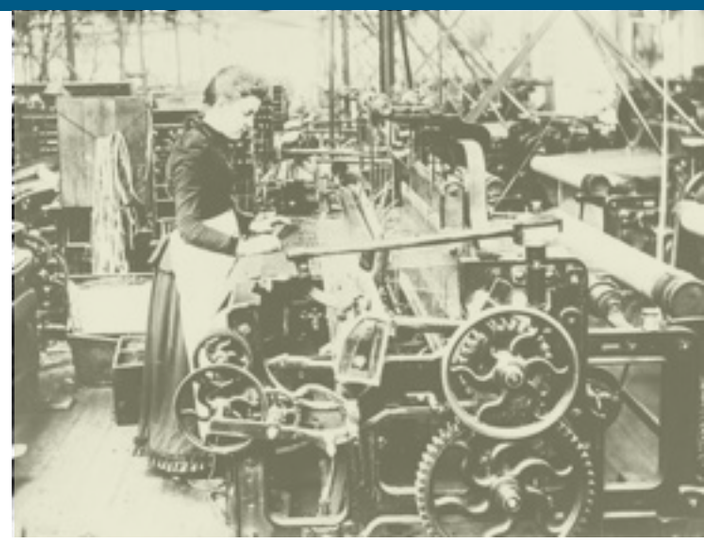
- Release Management
  - A standard way to track and release versions
  - Official versions stored on a central server
  - Can be used to automate the deployment process

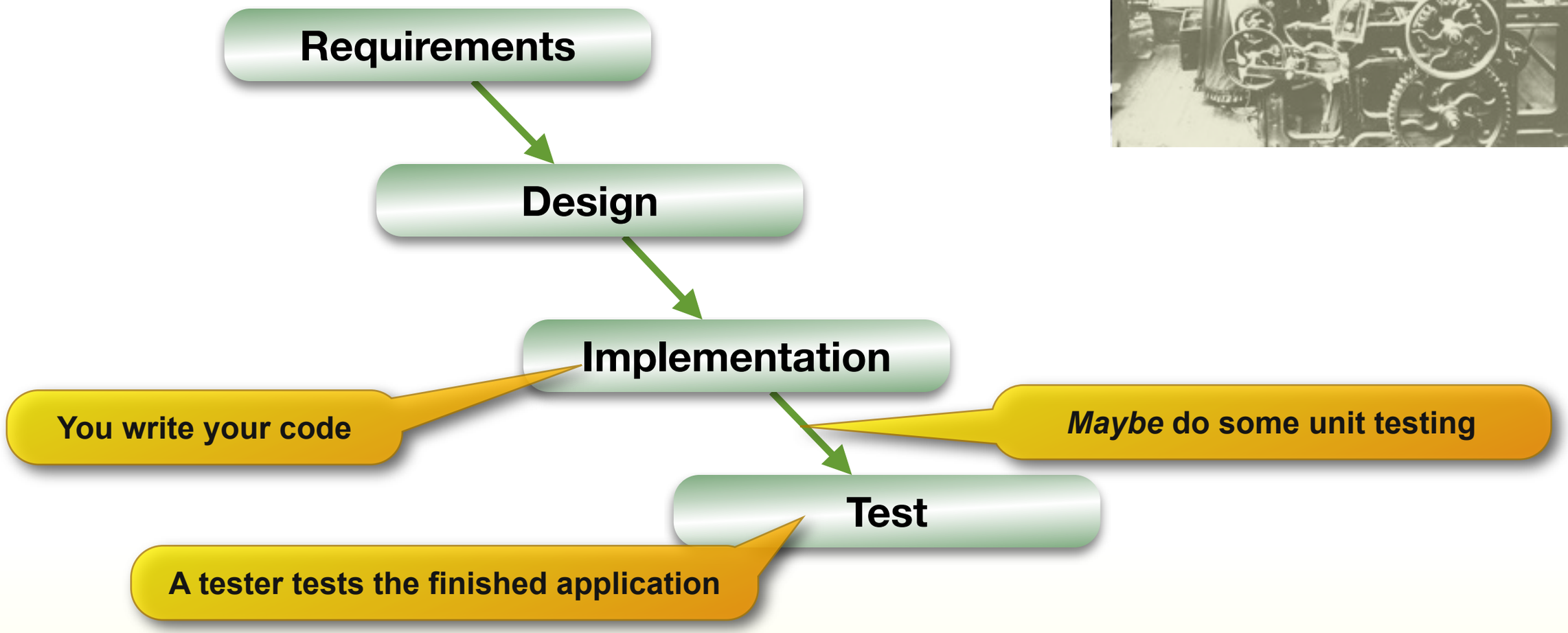
Towards better testing practices

- Why is good testing so important?
  - Development costs
  - Maintenance costs
  - Visibility
  - Flexibility
  - Documentation

Towards better testing practices

- Coding the traditional way





Towards better testing practices

- Coding the traditional way

Coding
Some unit testing?
Manual debugging?
Manual functional testing?

Fragile untested code → Lots of bugs

Low code coverage → Changes introduce new bugs

No regression tests → Changes are costly

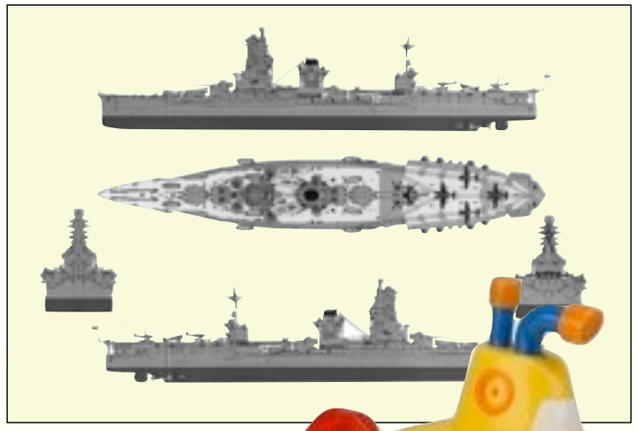
Bug detected late → Bugs hard to fix

Bug detected late → Bugs expensive to fix

Hard to trace code back to requirements → Code fails to meet requirements

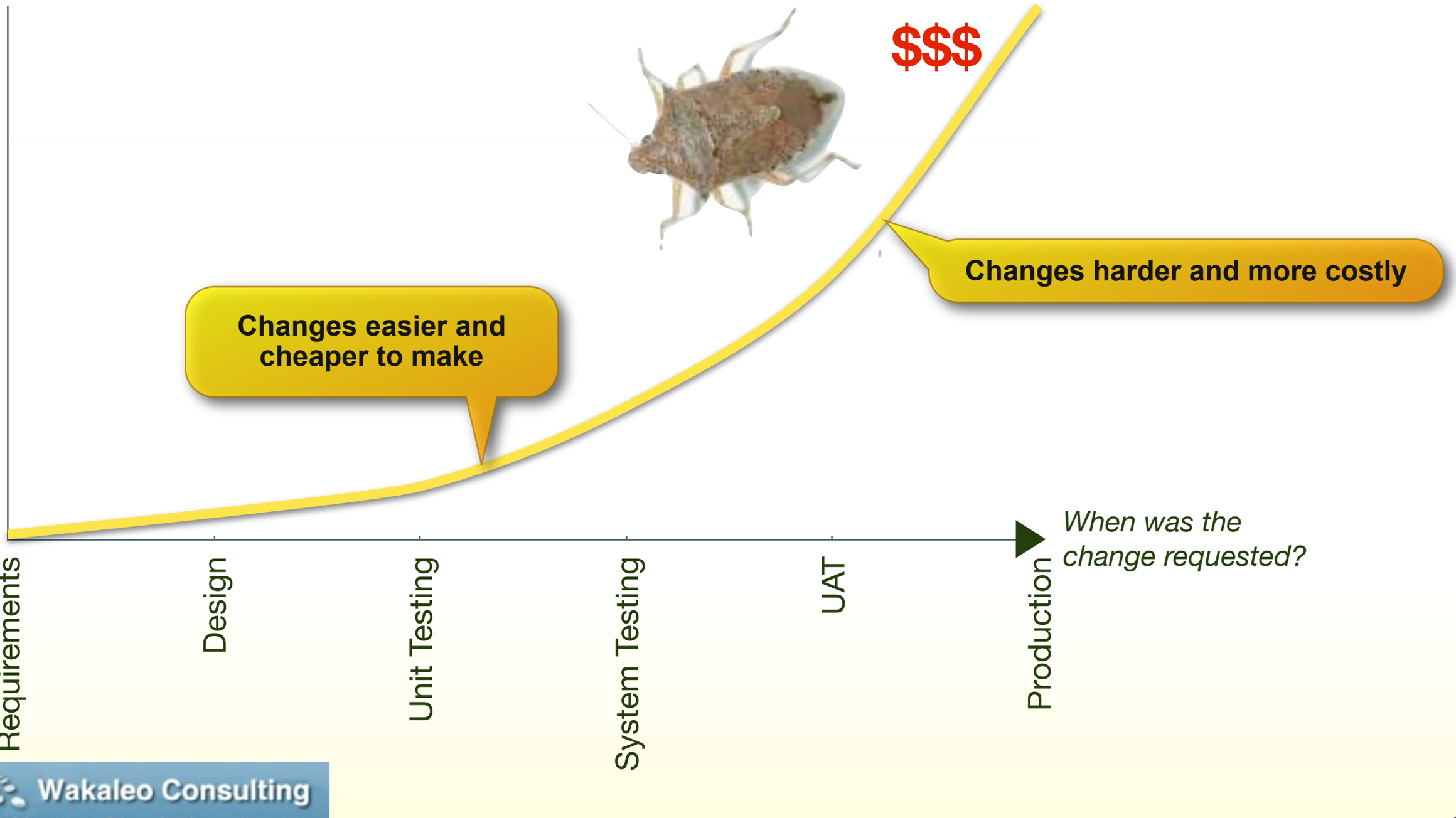
Towards better testing practices

- So what's wrong with the old way?
  - Lots of defects. Really, lots.
  - High maintenance costs
  - Hard to introduce new features
  - Doesn't meet the actual requirements
  - Delayed deliveries
  - Unhappy end-users



Towards better testing practices

- How much does a bug cost to fix, anyway?

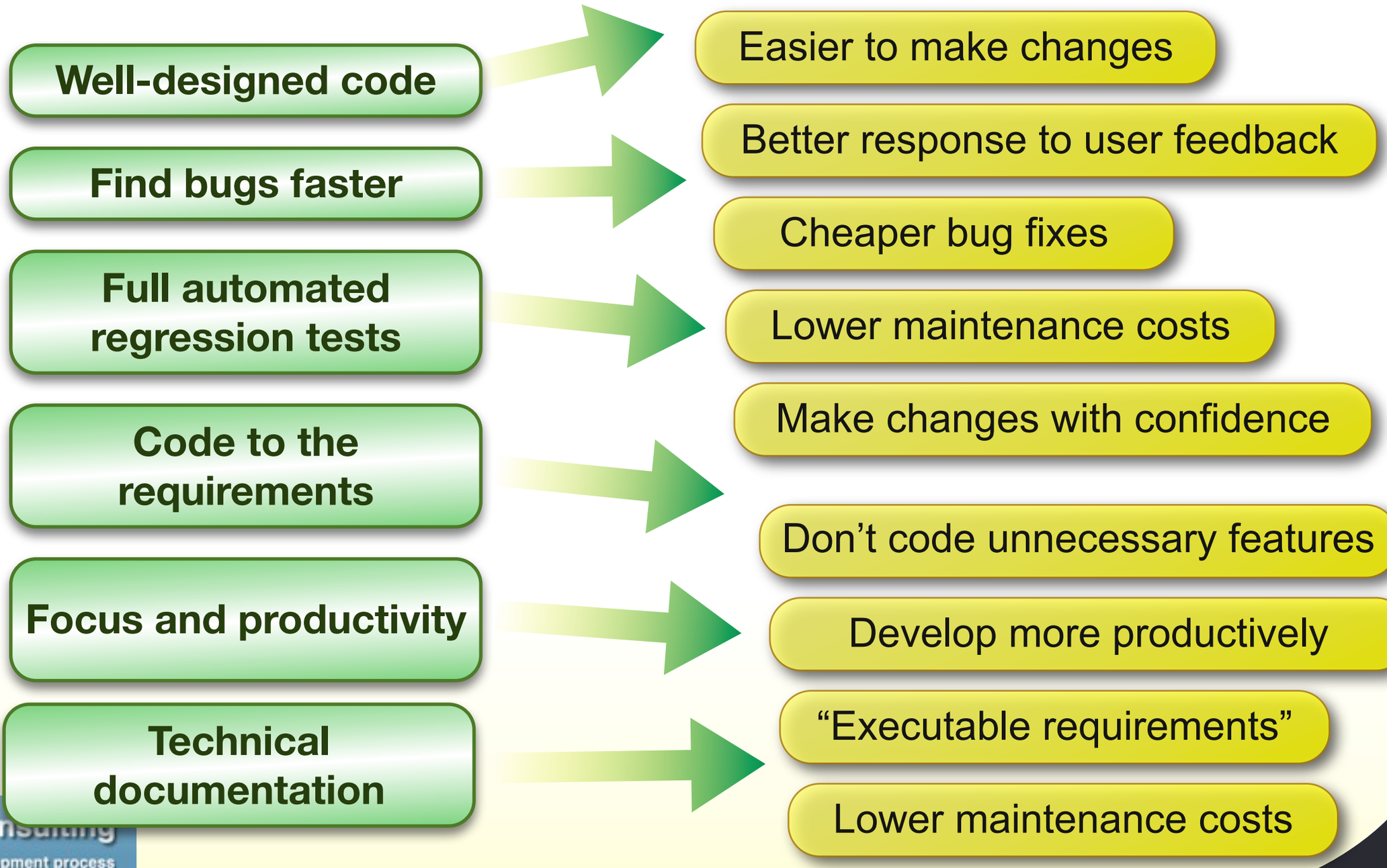


Towards better testing practices

- How can good testing practices help?
  - ☑ Reduce bugs
  - ☑ Write better-designed code
  - ☑ Have more confidence in our code
  - ☑ Make changes more easily
  - ☑ Meet user requirements more accurately
  - ☑ Lower maintenance costs

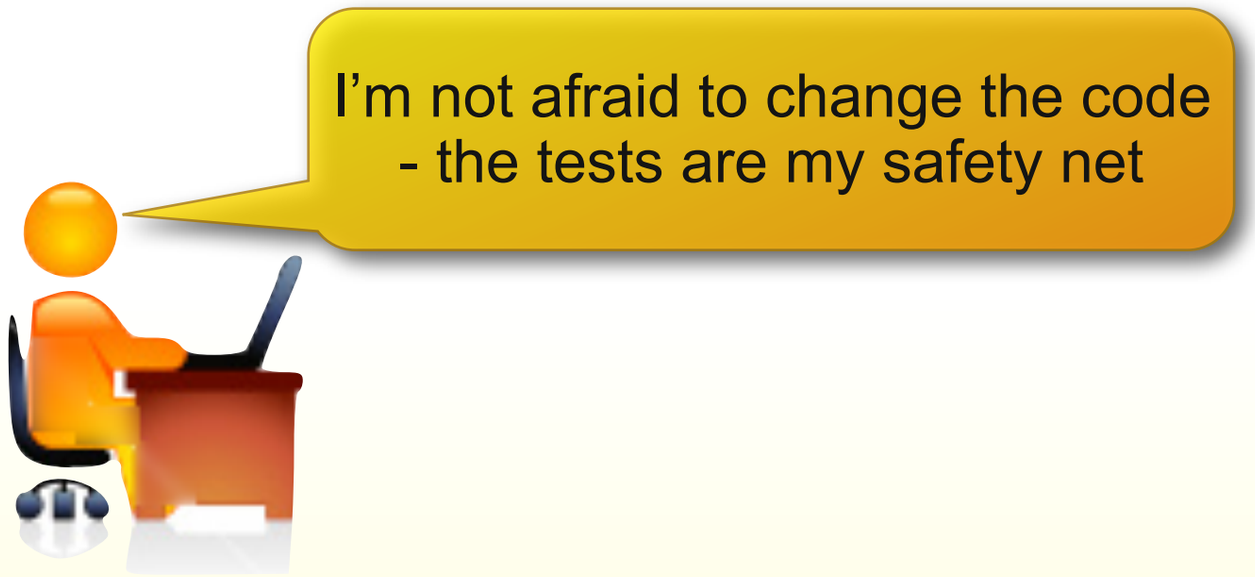
Towards better testing practices

- How can good testing practices help?



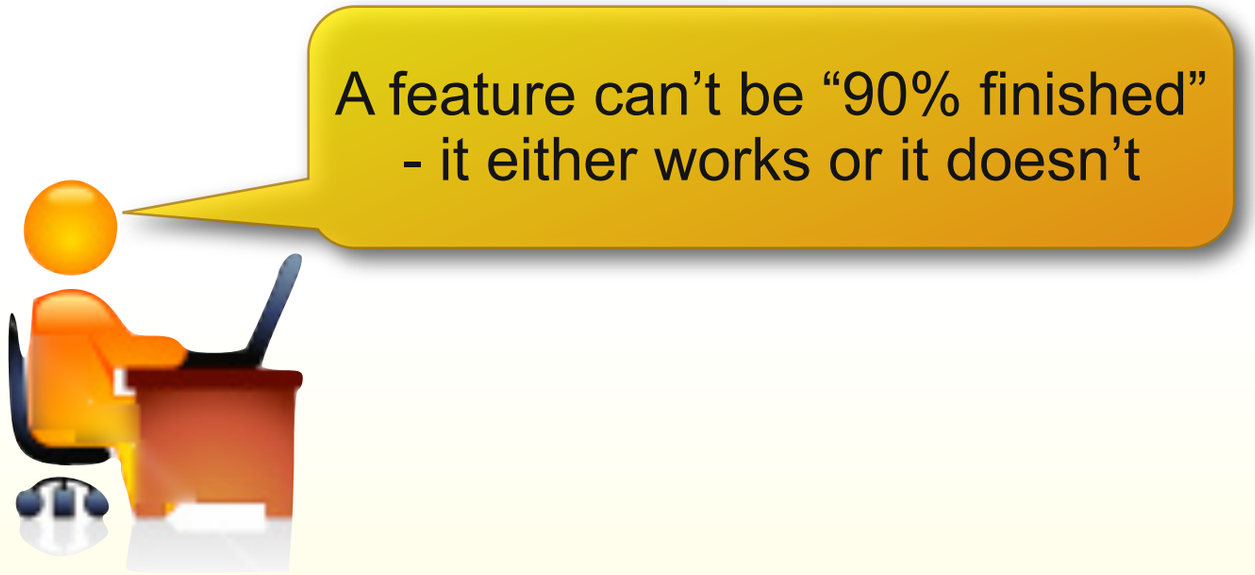
Towards better testing practices

- More flexibility
  - ☑ Testable code is easier to change
  - ☑ Full regression tests avoid introducing errors



Towards better testing practices

- Better visibility
  - Tests are “executable requirements”
  - Automated acceptance tests measure progress



Summary

Behaviors	Failed	Pending	Time (sec)
4	1	1	0.738

Stories Summary

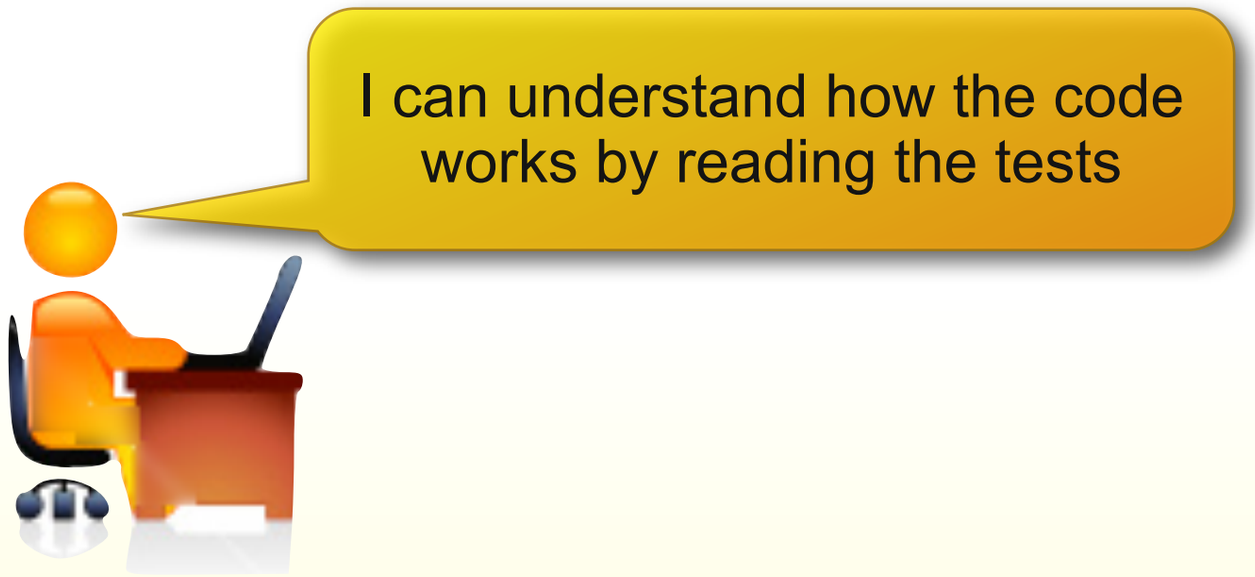
Stories	Scenarios	Failed	Pending	Time (sec)
2	4	1	1	0.738

Specifications Summary

Specifications	Failed	Pending	Time (sec)
0	0	0	0.0

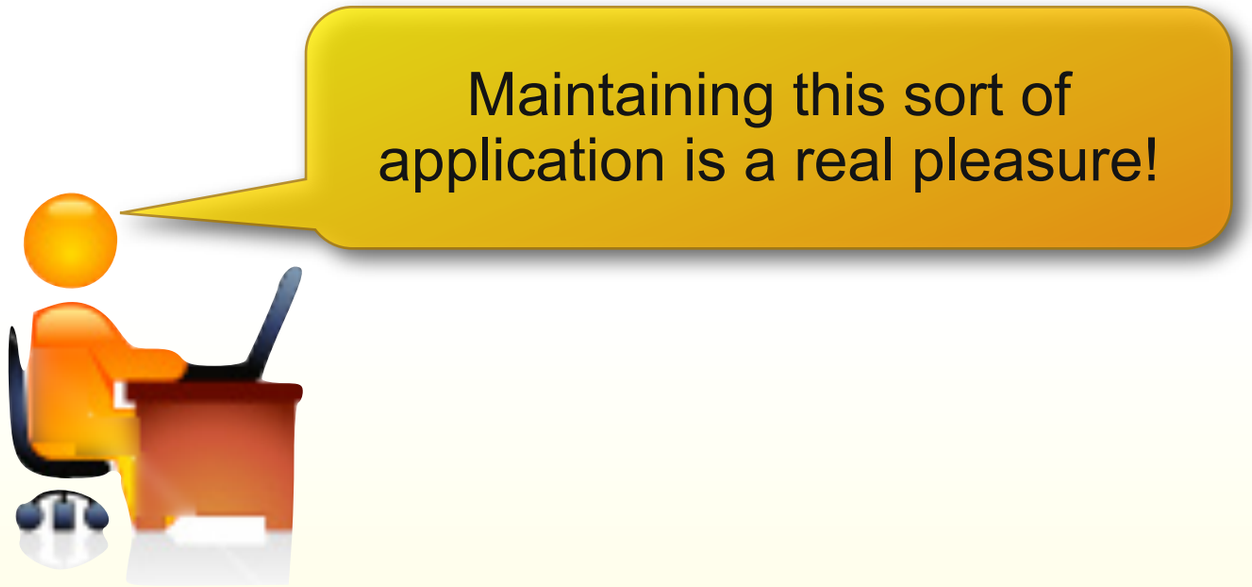
Towards better testing practices

- Documentation
  - Tests are “living documentation” of your code
  - Always accurate and up-to-date



Towards better testing practices

- Lower maintenance costs
  - Less bugs, found faster
  - Changes are easier to make

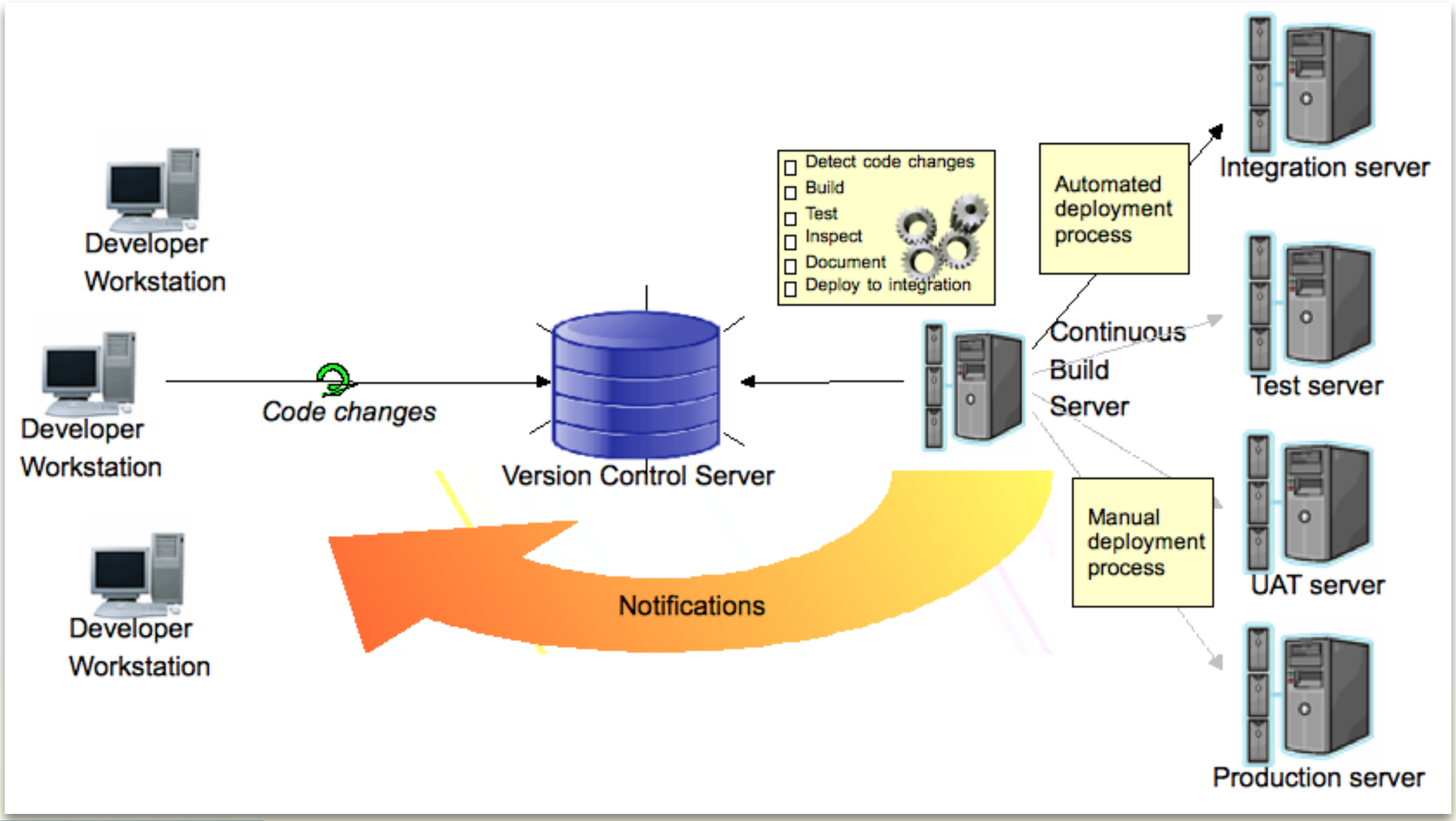


Automating the build process

- Continuous Integration - what's the issue?
  - Traditional development cycles are bad for your health:
    - Integration is long and difficult
    - Poor visibility on development progress
    - Functional tests are done too late
    - Raised issues are harder to fix
    - The client gets a sub-optimal product

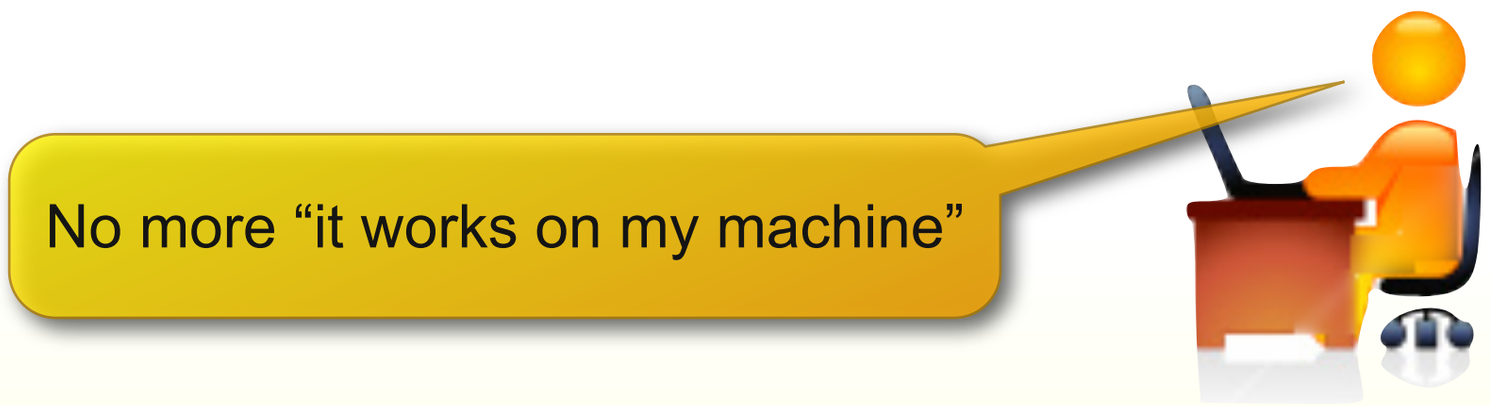
Automating the build process

- Continuous Integration - what's involved?



Automating the build process

- Continuous Integration - why bother?
  - Smoother integration process
  - Automatic regression testing
  - Regular working releases
  - Earlier functional testing
  - Faster and easier bug fixes
  - Better visibility



Automating the build process

- Continuous Integration - what you need

Automated build process (e.g. Maven)

Automated tests (JUnit, Selenium, easyb...)

Source code repository

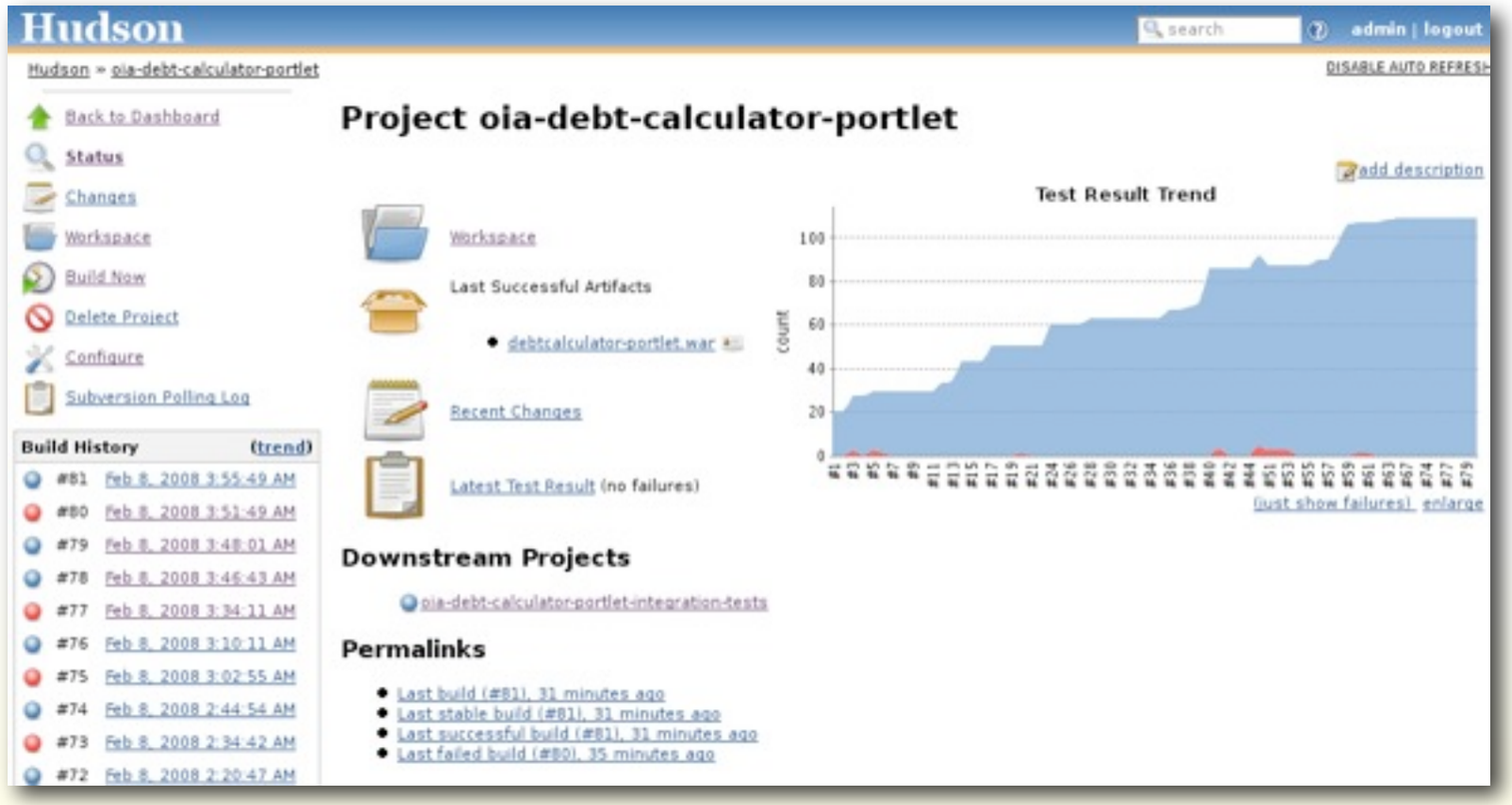
Continuous Build Server

Automating the build process

- Continuous Integration - what can it do?
  - Raise integration issues - fast!
  - Monitor your build process
  - Monitor and report on code quality and code coverage
  - Build promotion and release management
  - Automated deployments

Automating the build process

- Looking for a good O/S Continuous Integration tool?
  - Try Hudson!
    - Easy to set up and configure
    - Good build and code quality metrics
    - Lots of plugins

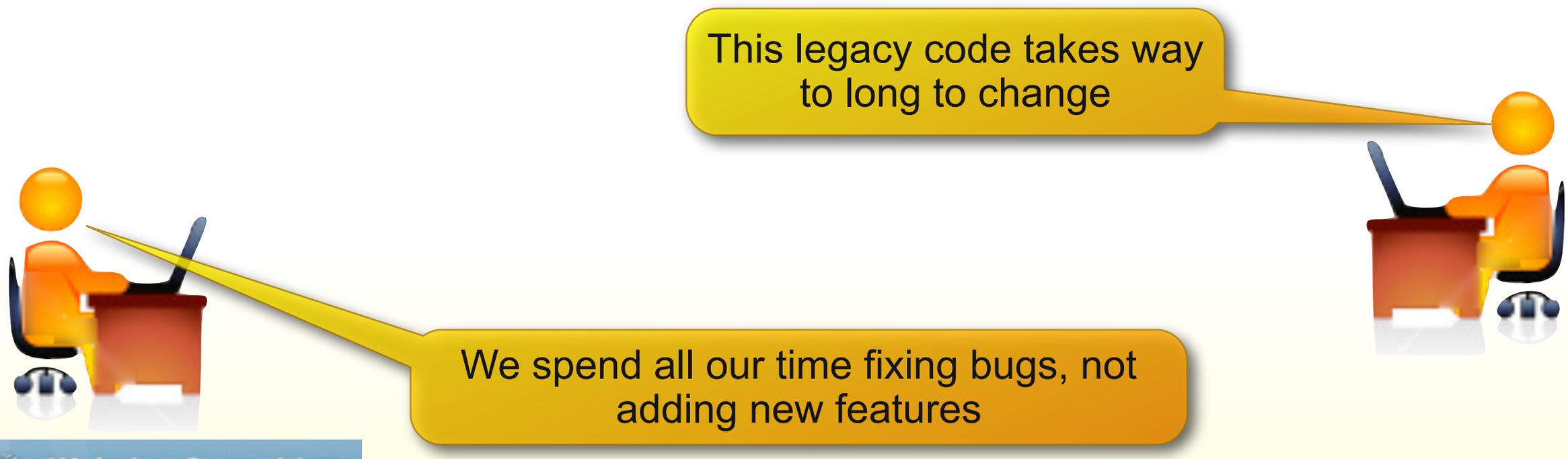


Automated Code Quality

- Why use code quality metrics
  - Better quality code
  - Enforce corporate coding standards
  - Detect potential bugs
  - Code is easier to maintain
  - Train new staff
  - Keep technical debt down

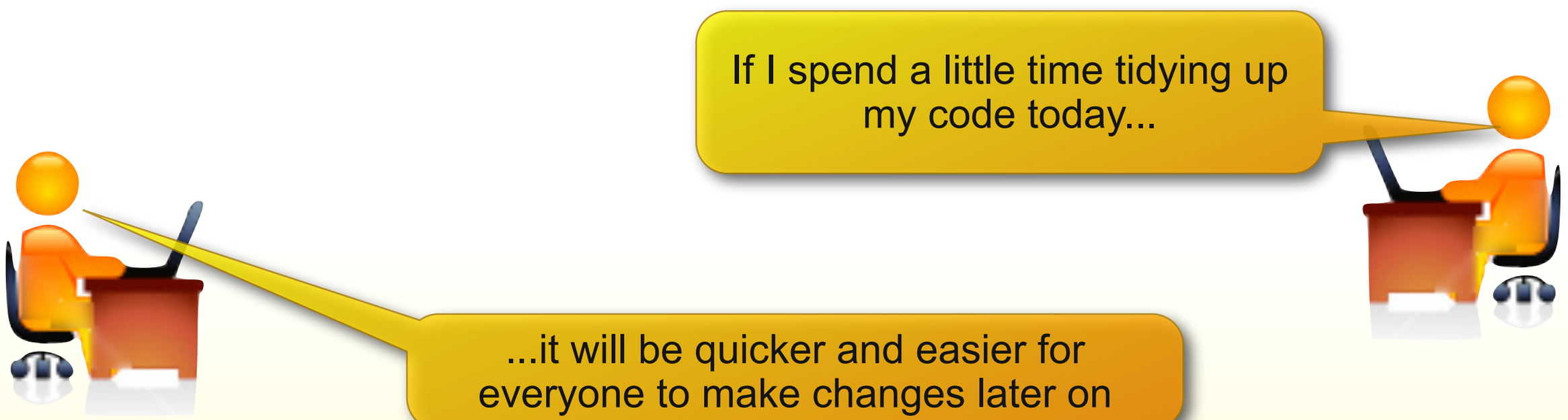
Automated Code Quality

- What is technical debt?
  - The cost of poor quality code:
    - Harder to make changes
    - Too much time spent fixing bugs
    - Takes too long to add competitive new features



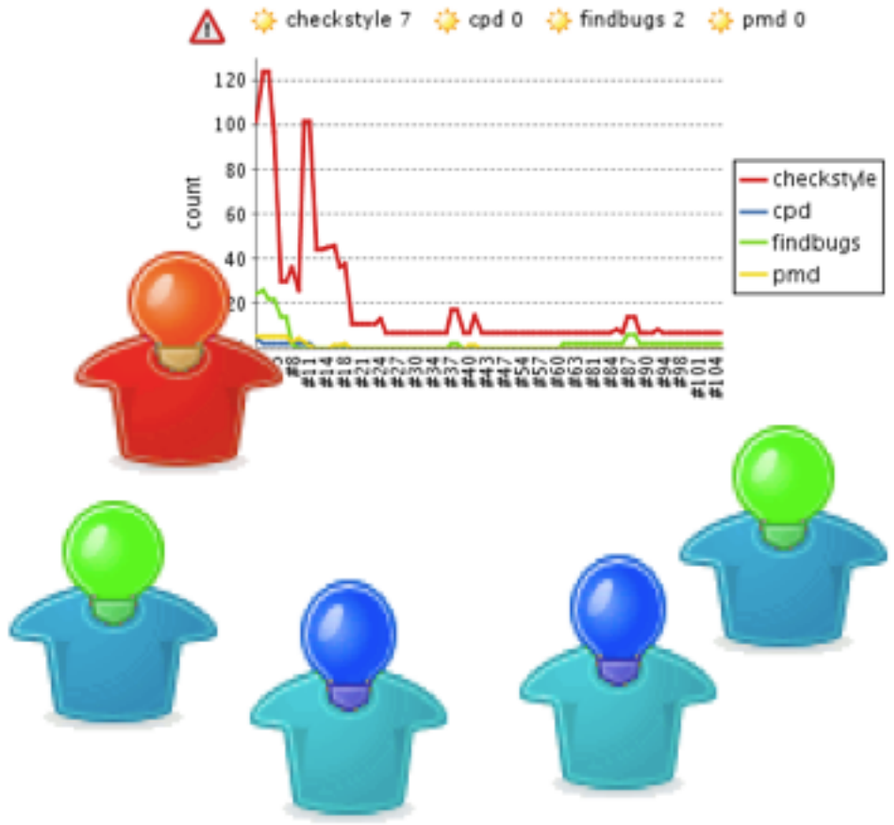
Automated Code Quality

- How do we pay off technical debt?
  - Enforce coding standards
  - Teach developers good coding practices
  - Spend time keeping the code clean (refactoring)



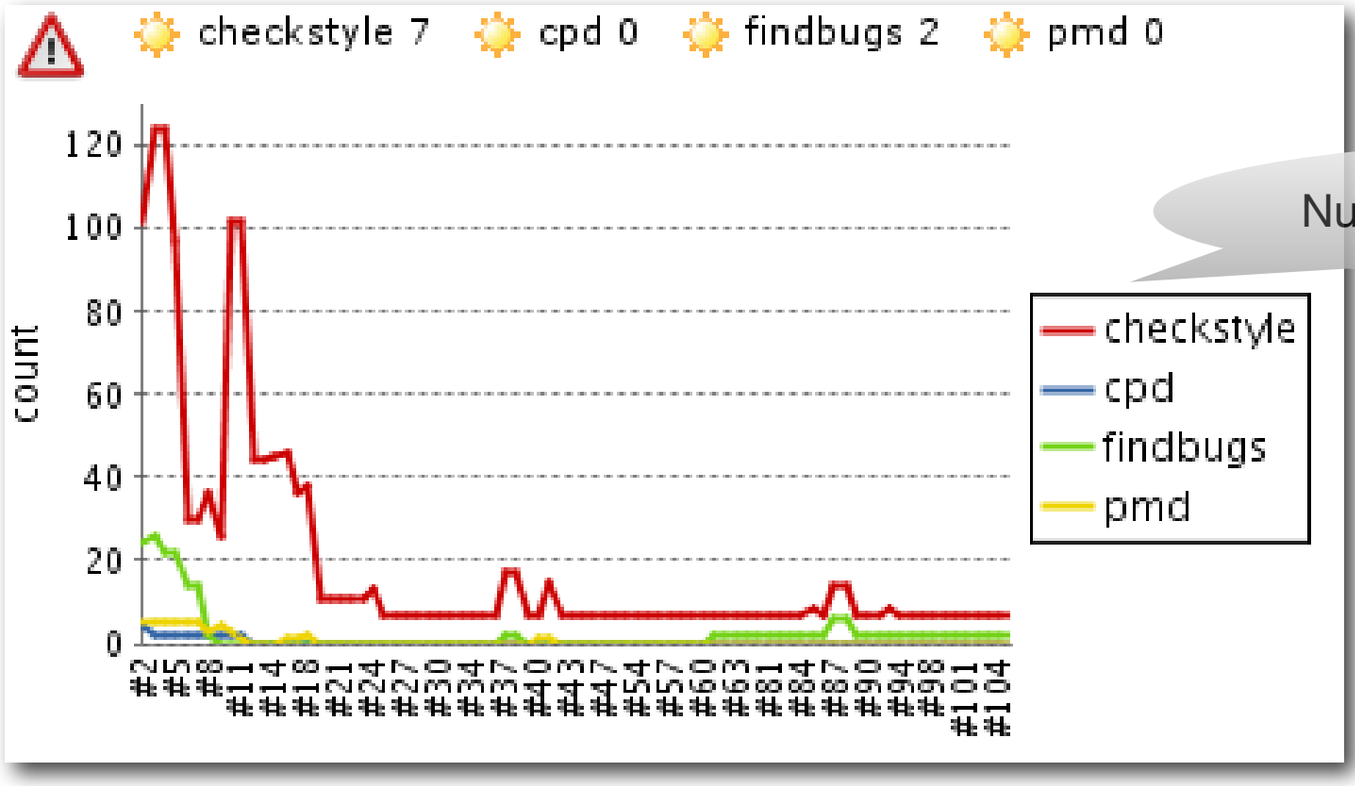
Automated Code Quality

- Team code reviews
  - Review code as a group
  - Long and slow if done manually
  - Benefits greatly from the use of tools



Automated Code Quality

Enforcing coding standards with Hudson



Number of violations over time

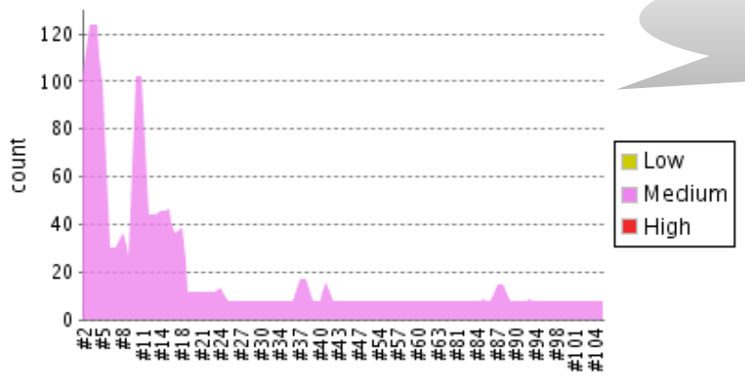
Automated Code Quality

Enforcing coding standards with Hudson

Violations Report for build 105

Type	Violations	Files in violation
checkstyle	7	4
cpd	0	0
findbugs	2	1
pmd	0	0

checkstyle

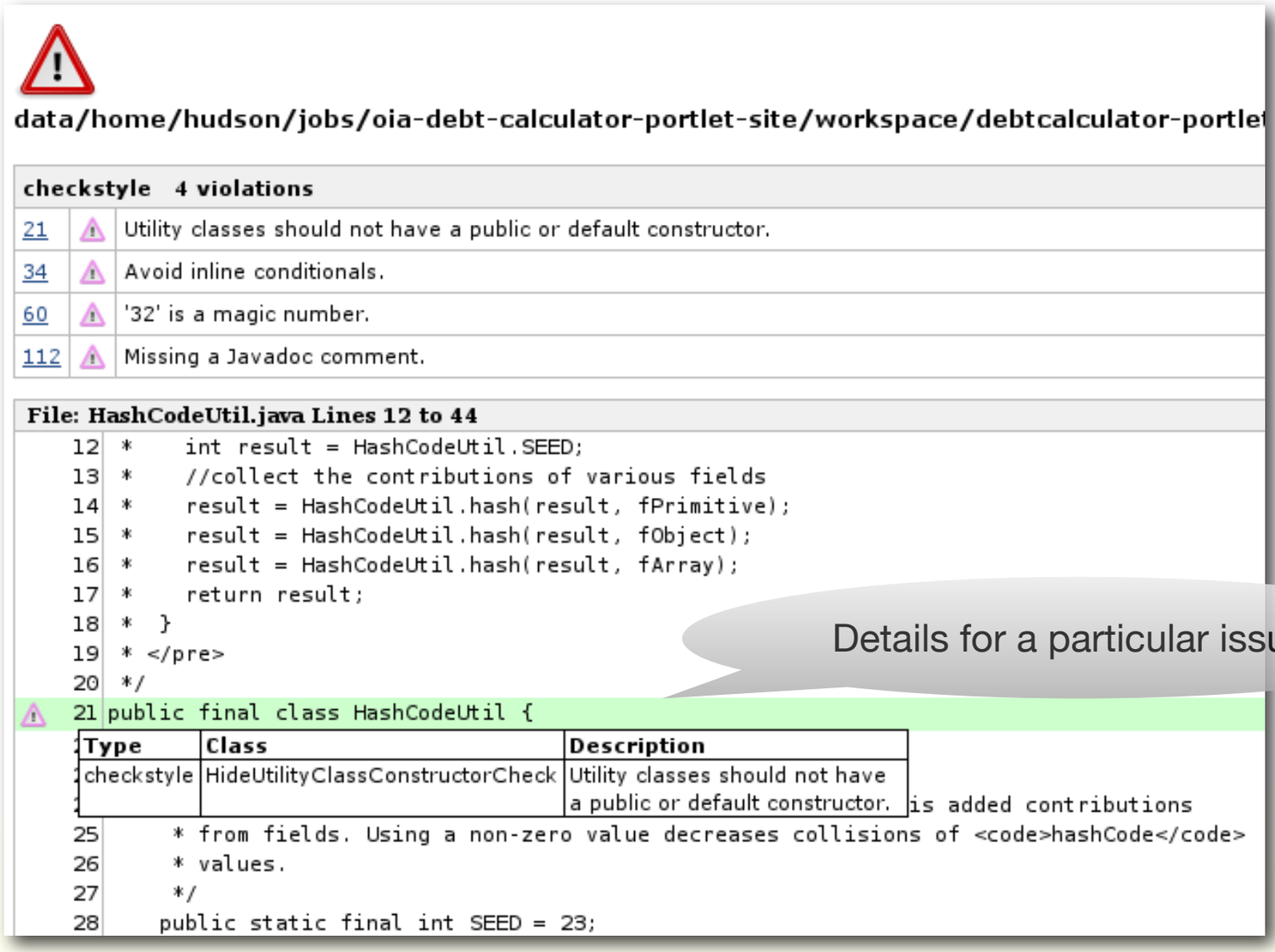


Drilling down

filename
data/home/hudson/jobs/oia-debt-calculator-portlet-site/workspace/debtcalculator-portlet/src/main/java/nz/govt/ird/oia/debtcalculator/portle
data/home/hudson/jobs/oia-debt-calculator-portlet-site/workspace/debtcalculator-portlet/src/main/java/nz/govt/ird/oia/debtcalculator/portle
data/home/hudson/jobs/oia-debt-calculator-portlet-site/workspace/debtcalculator-portlet/src/main/java/nz/govt/ird/oia/debtcalculator/portle
data/home/hudson/jobs/oia-debt-calculator-portlet-site/workspace/debtcalculator-portlet/src/main/java/nz/govt/ird/oia/debtcalculator/portle

Automated Code Quality

Enforcing coding standards with Hudson



Automated Code Quality

- Code Coverage
  - See what code is being executed by your unit tests.
  - Isolate untested code
  - Can help to estimate if testing practices are being applied

Automated Code Quality

Monitoring Code Coverage with Hudson

```
43        */
44       public Manifest getManifest() {
45   3       if (manifest == null) {
46   0           loadMetaInf();
47           }
48   3       return manifest;
49       }
50
51       public void setManifest(Manifest manifest) {
52   0       this.manifest = manifest;
53   0       }
54
55       public ServletContext getContext() {
56   0       return context;
57       }
58
59       public void setContext(ServletContext context) {
60   3       this.context = context;
61   3       }
62
63       /**
64        * Load the META-INF file.
65        */
66       public void loadMetaInf() {
67   3       String appServerHome = context.getRealPath("/");
68   3       File manifestFile = new File(appServerHome, "META-INF/MANIFEST.MF");
```

Executed code

Unexecuted code

Automated Code Quality

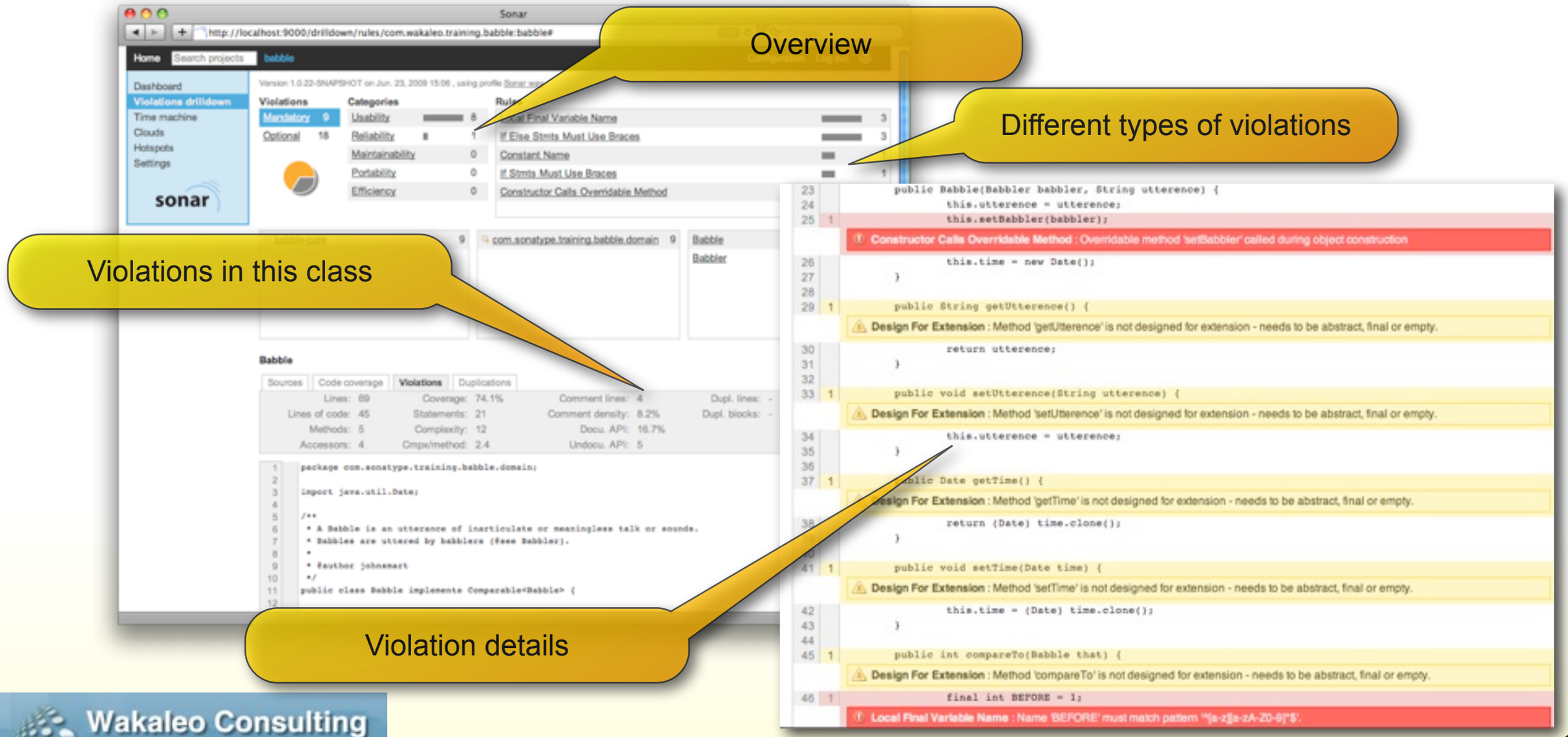
- Code Quality Governance with Sonar
  - Centralized code quality management
  - Works on any Maven project
  - Store code quality metrics in a database
  - Code quality metrics can be consulted on a web site

Automated Code Quality

- Code Quality Governance with Sonar
  - Sonar centralizes many code quality metrics

Automated Code Quality

- Code Quality Governance with Sonar
  - You can drill down to view the details for each type of issue



Summary

- How can you improve the development process?
  - Standardize your build process
  - Improve testing practices
  - Use Continuous Integration
  - Reduce your technical debt
  - Automate, automate, automate!

John Ferguson Smart
Email: john.smart@wakaleo.com
Web: http://www.wakaleo.com
Twitter: wakaleo

Thanks for your attention!