TDC 2010

sonarsource

Ambiente de integração contínua

Ferramentas e soluções para diminuir a complexidade do desenvolvimento

Helder da Rocha (helder@argonavis.com.br)

Helder da Rocha

- **Java**, since 1995
- **Objective-C** & iPhone since 2008
- HTML, JavaScript, Web
- www.argonavis.com.br
- www.helderdarocha.com.br

Objetivos

- Como lidar com a complexidade crescente do **processo de desenvolvimento**
- Que **ferramentas** e **soluções** existem para isso?

Lei da preservação da complexidade

- A complexidade não desaparece!
- Ela se **muda para outro lugar** e se esconde
  - e continua a crescer por lá

Para onde foi a complexidade?

- Com orientação a objetos, ferramentas, frameworks, etc. **programar ficou mais simples**
- Mas a complexidade aumentou nos **processos** e **configuração** dos ambientes
  - Manter repositórios, versões, dependências
  - Configuração, metadados, mapeamentos, ligações
  - Integração, protocolos, adaptadores, filtros, proteções
  - Processo, build, documentação, testes

Como diminuir a complexidade?

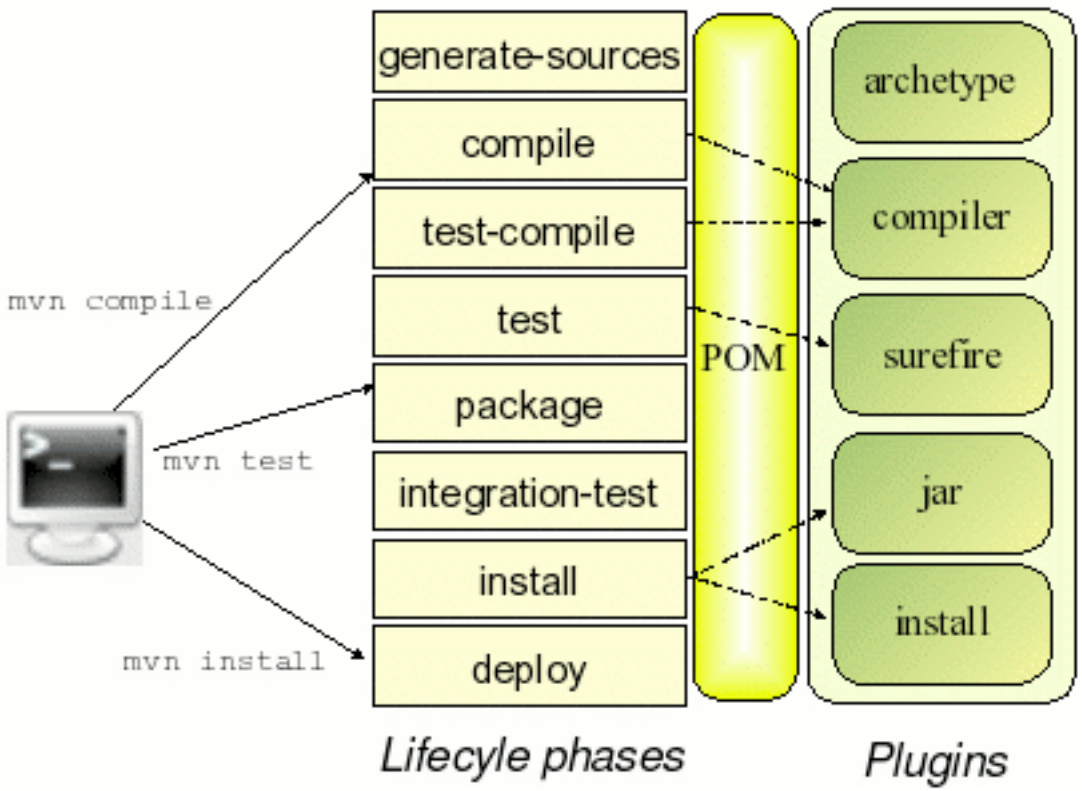
- Nos processos: mais feedback
  - Metodologias de desenvolvimento ágeis
  - Ferramentas para **organizar e simplificar** o processo
  - Ferramentas para **coletar e organizar** informações automaticamente: **métricas**, **qualidade**, prazos, etc.
- Na configuração: menos duplicação de esforços
  - **Automatizar processos** como build, deploy, versionamento, etc.
  - **Integrar ambientes** e ferramentas
  - **Gerar código**

Ferramentas

- Build
  - Ant / Ivy / Maven
- Qualidade e métricas
  - PMD / Checkstyle / Findbugs
  - Sonar
  - JUnit / Cobertura
- Documentação
  - Javadoc via Ant e Maven, Graphviz
- Geração de código, configuração
  - XSLT / XPath

Maven (maven.apache.org)

- Gerencia builds e repositórios

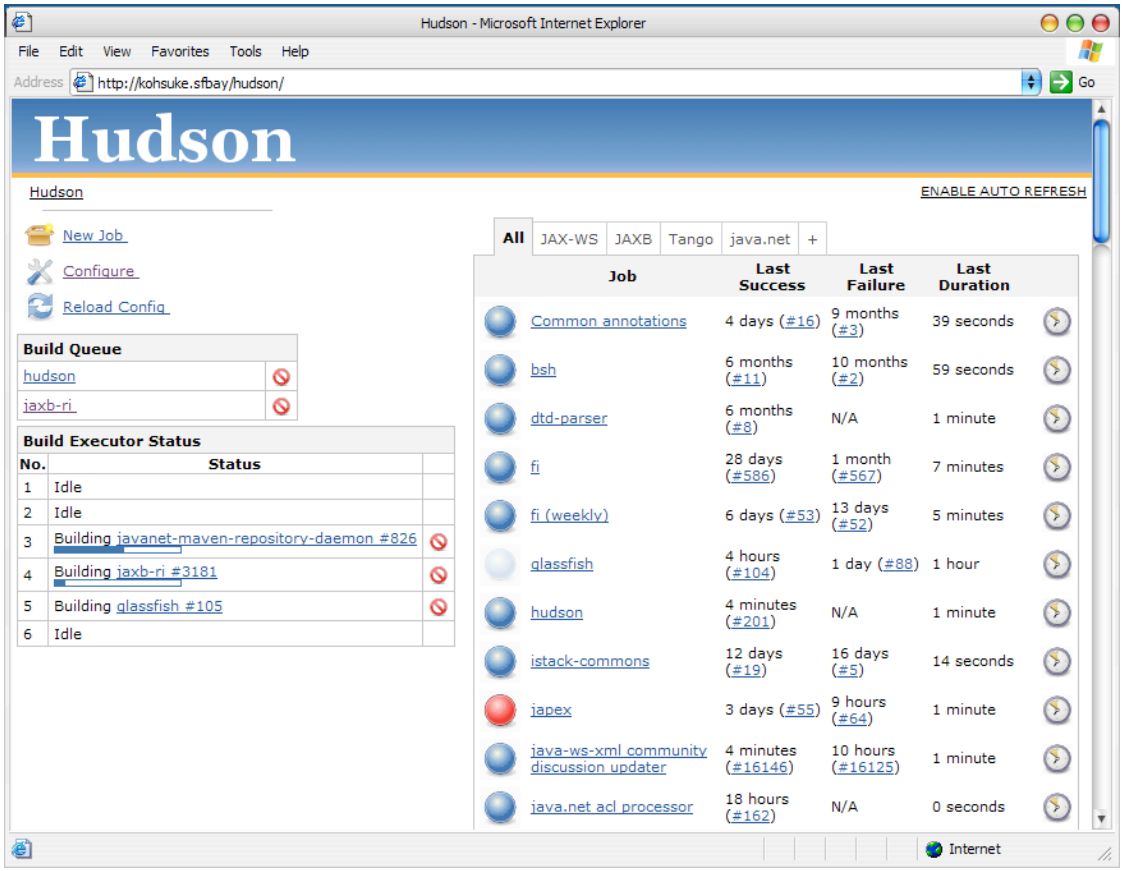


Ant + Ivy (ant.apache.org)

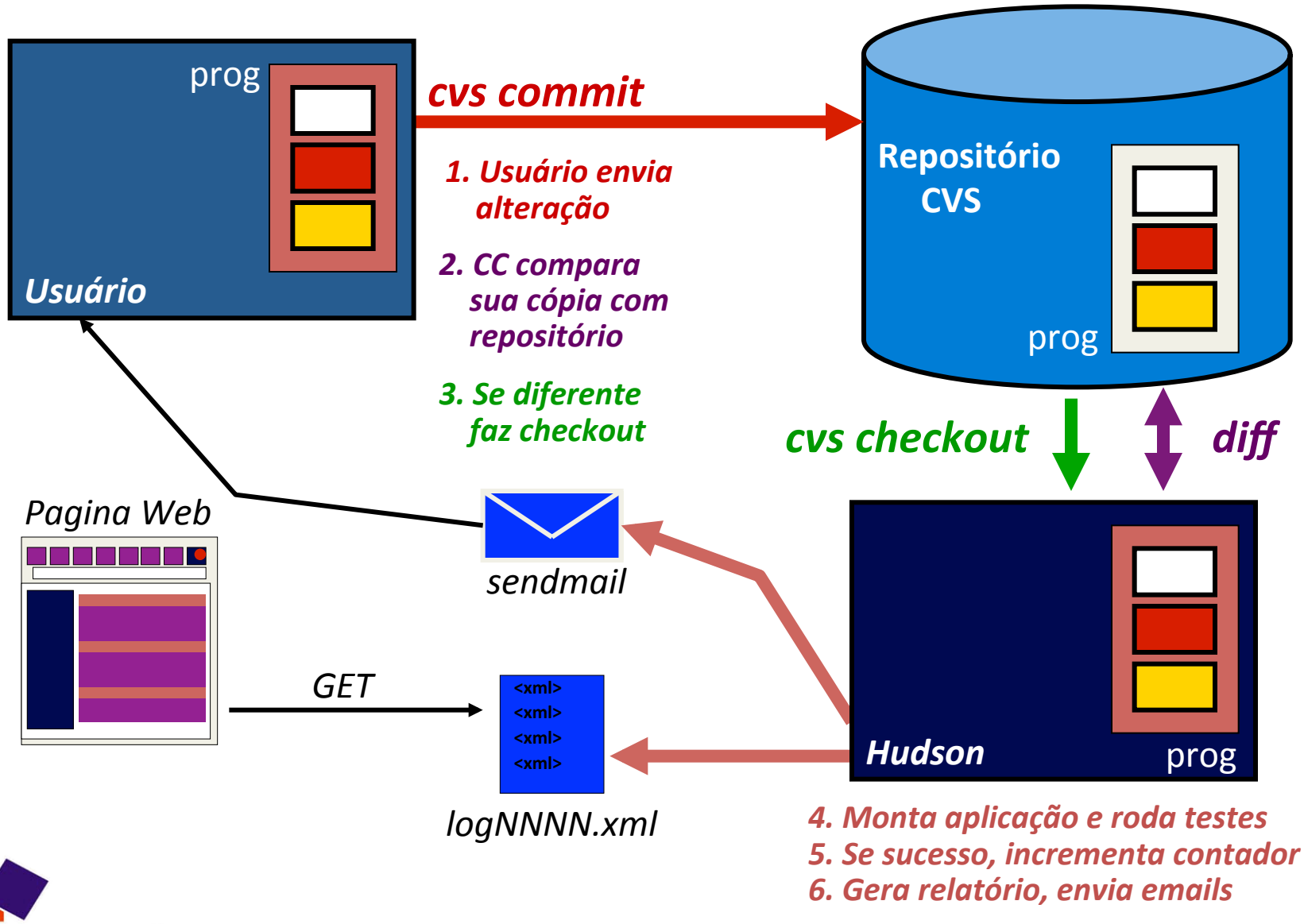
- Gerencia builds (Ant) e repositórios
  - Alternativa ao Maven
  - Podem ser integrados a um ambiente Maven

Hudson (https://hudson.dev.java.net/)

- Integração contínua + Feedback contínuo

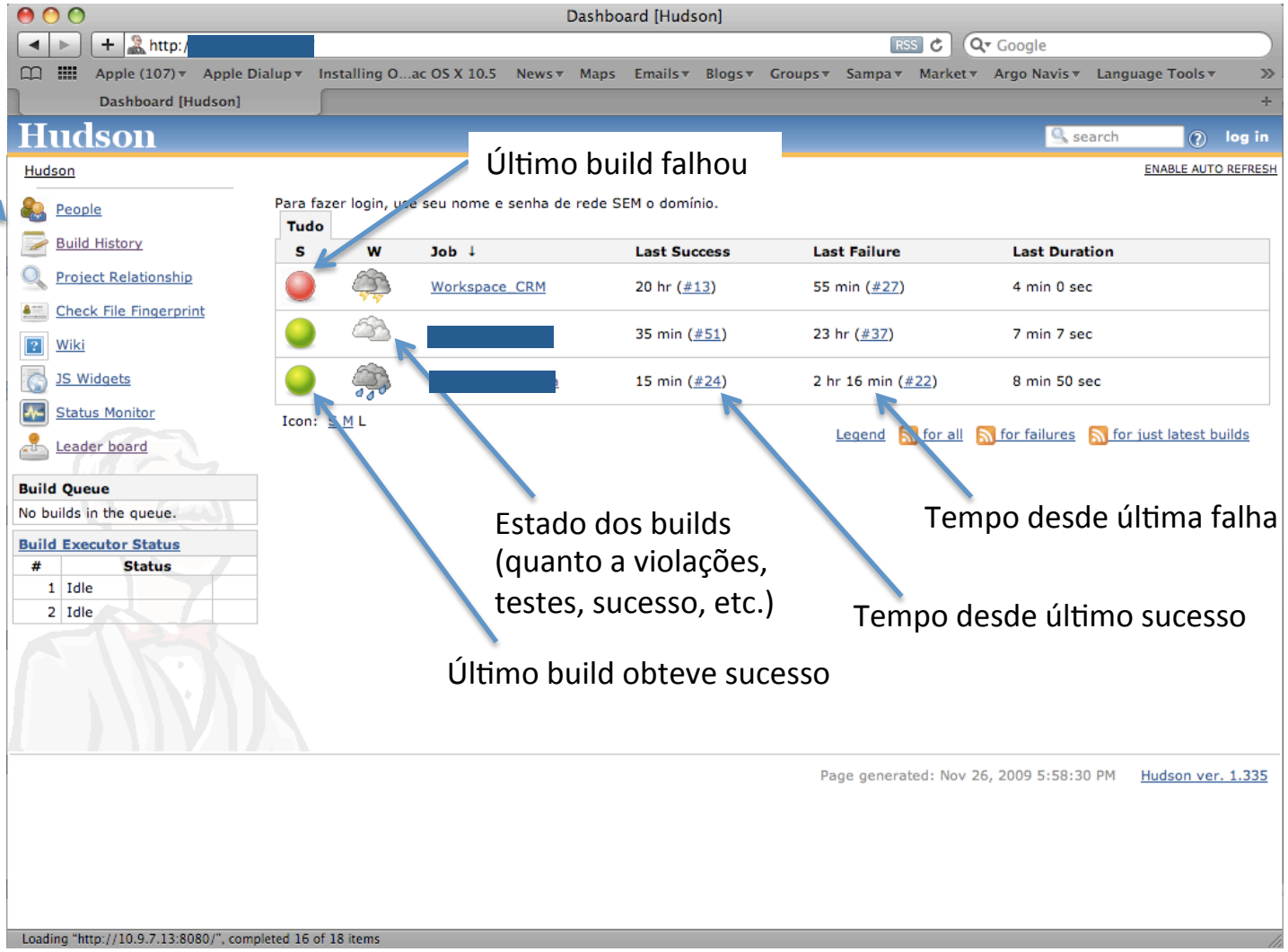


Integração contínua



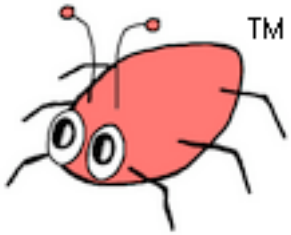
Hudson: dashboard

Usuarios CVS que fazem commit ou montam o projeto

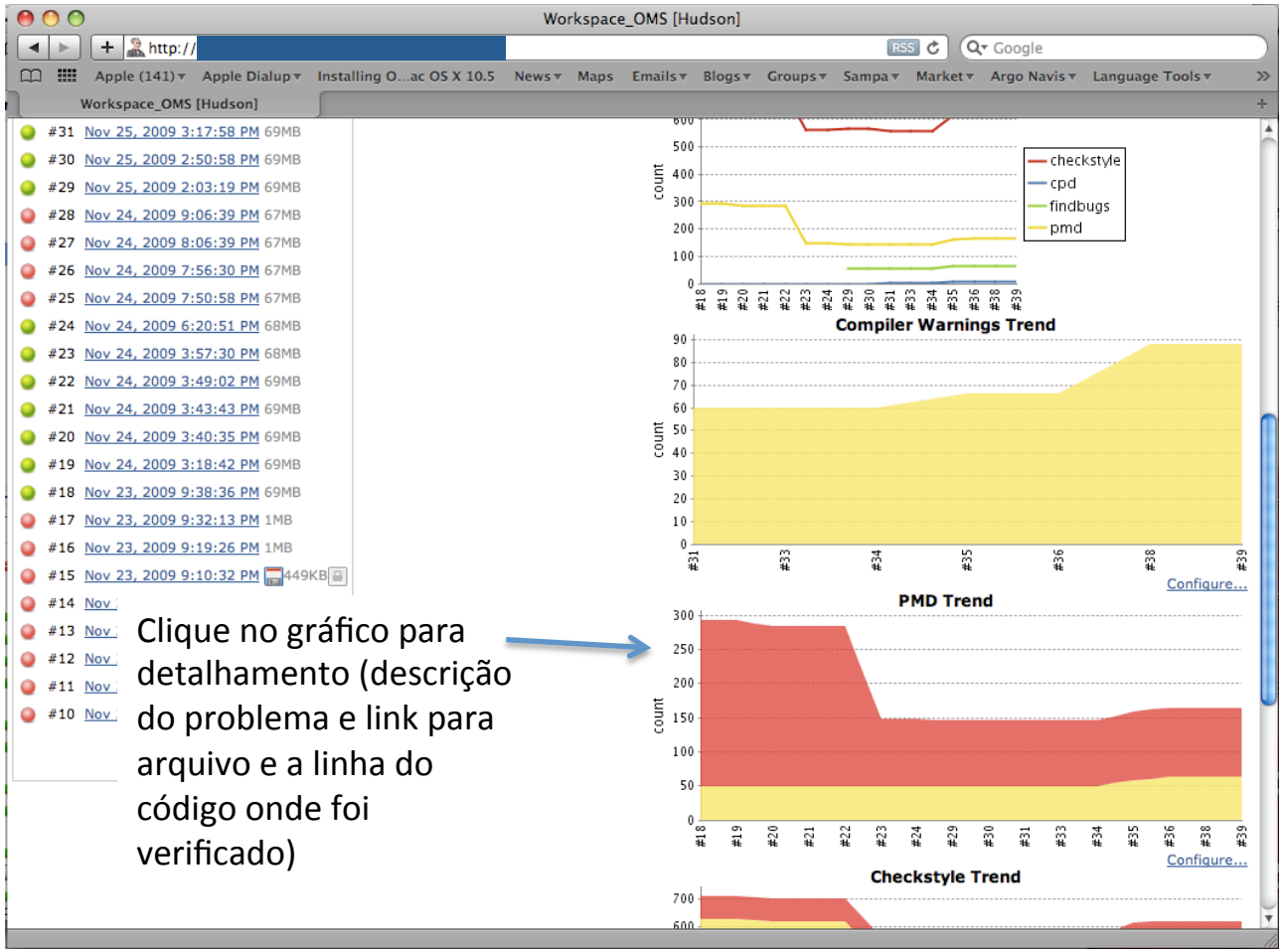


PMD + Checkstyle + Findbugs

- Ferramentas de análise estática
- Verificam **qualidade** do código
- Geralmente são incluídas no ambiente de desenvolvimento (Eclipse, Netbeans)
  - Mas também podem ser executadas via Ant, Maven e incluídas na integração contínua
- **Sonar** e **Hudson** são ótimos para apresentar resultados de PMD, Checkstyle e Findbugs
- Onde encontrar
  - http://pmd.sourceforge.net/
  - http://checkstyle.sourceforge.net
  - http://findbugs.sourceforge.net/

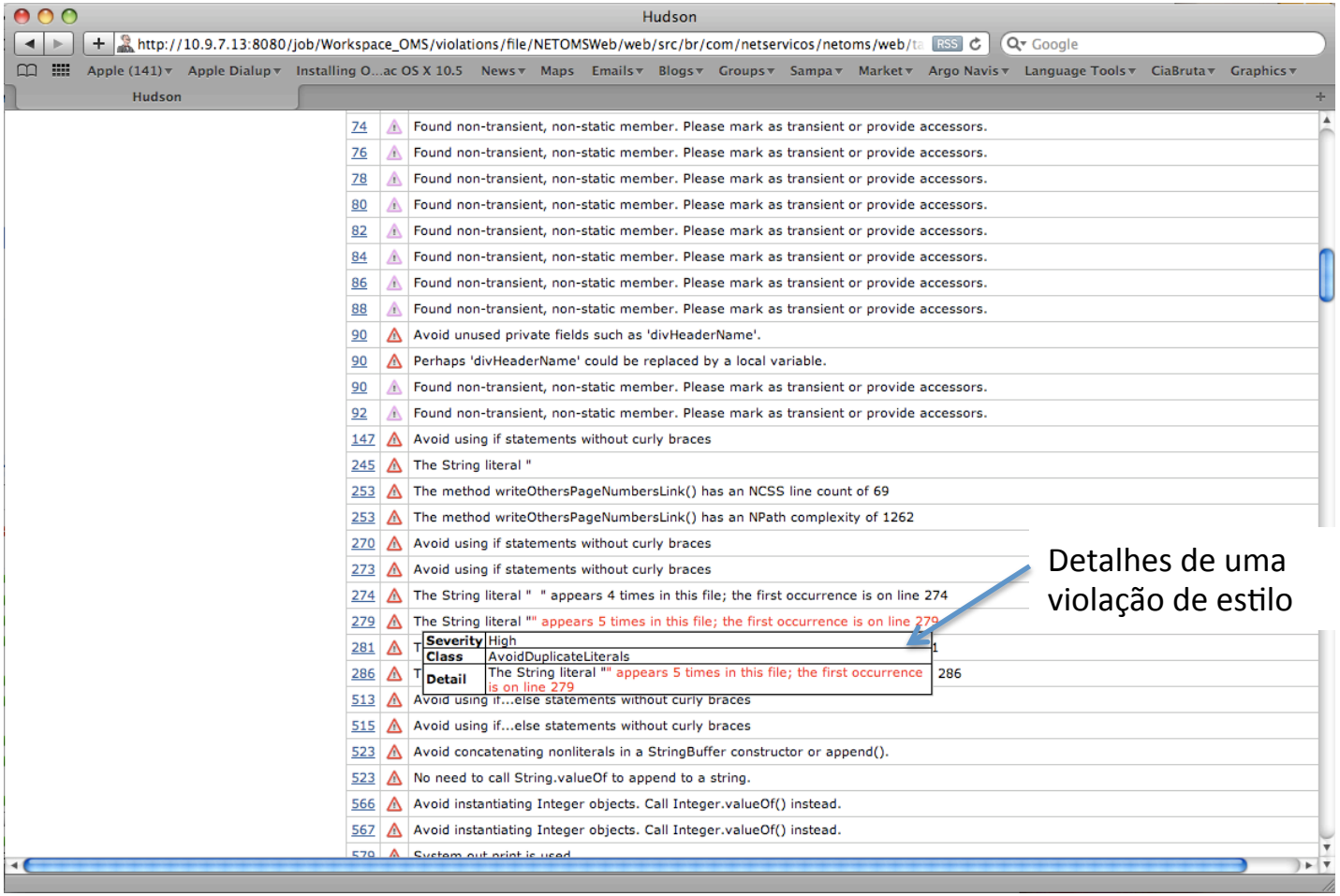


Gráficos de tendências: Hudson



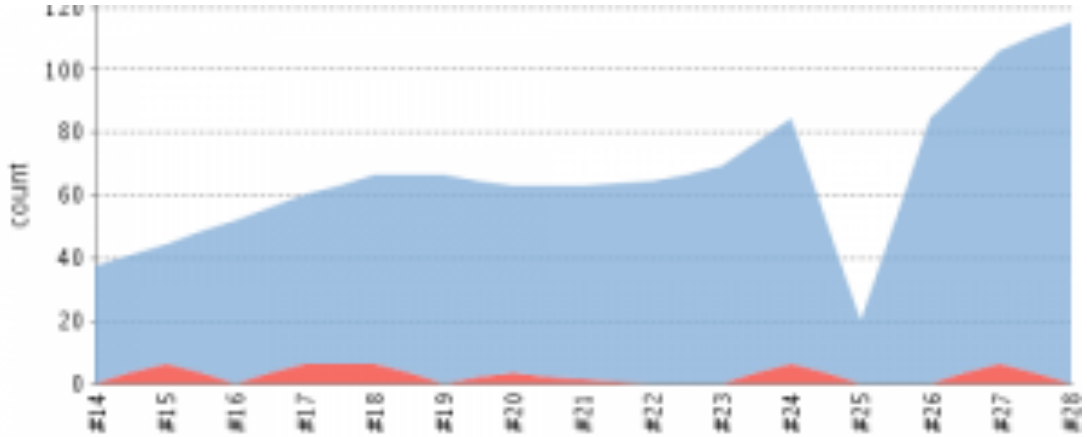
Clique no gráfico para detalhamento (descrição do problema e link para arquivo e a linha do código onde foi verificado)

Relatórios de violações e falhas de testes



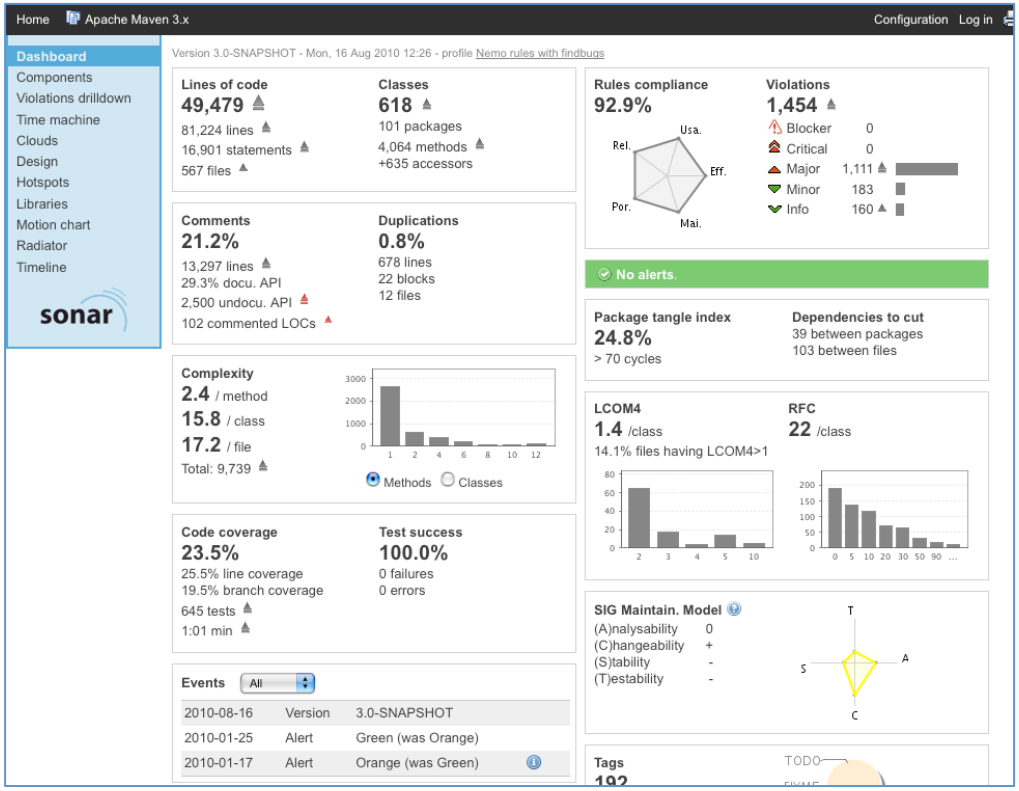
JUnit e Cobertura

- Seu código deve ter testes unitários
- Cobertura identifica o nível de cobertura desses testes (métricas via Sonar)
- Hudson gera gráfico de tendências



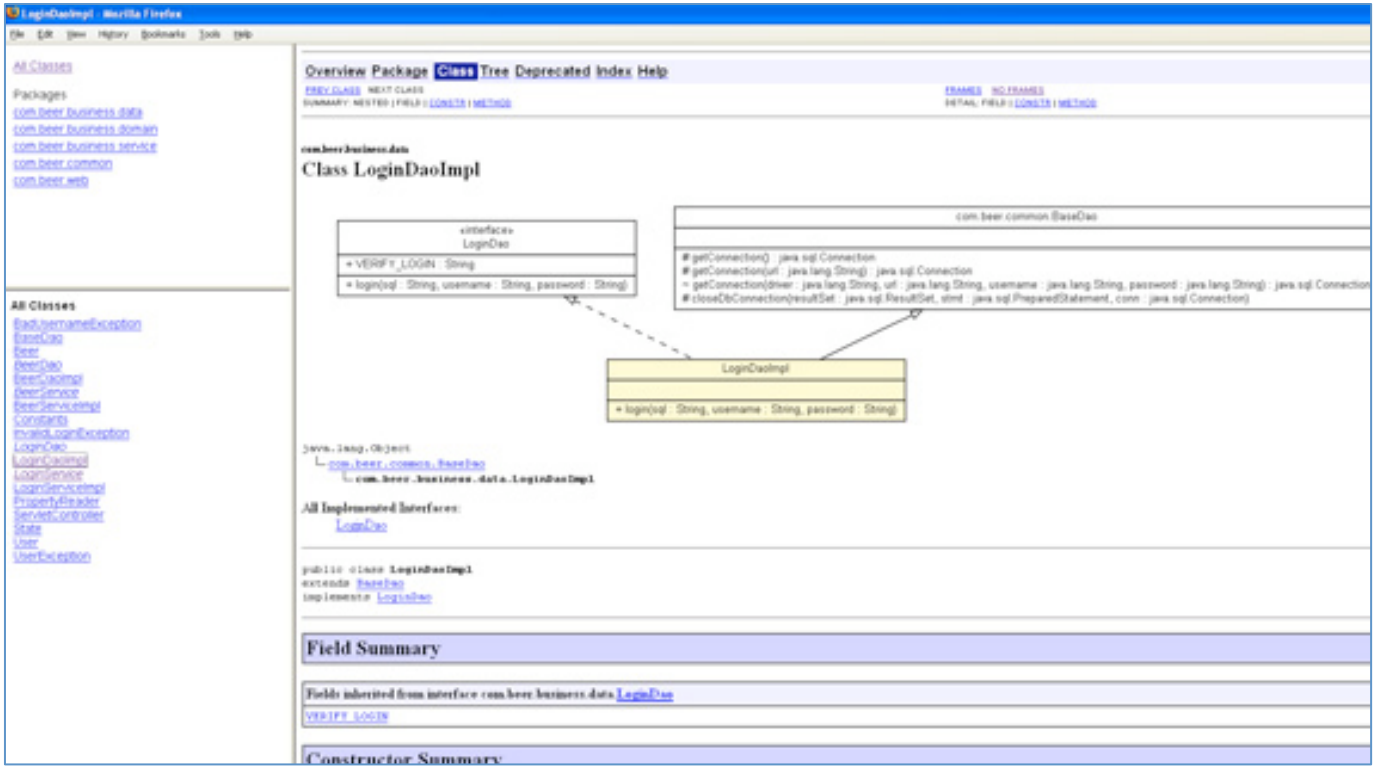
Sonar (www.sonarsource.org)

- Instale no servidor Web; Gere os dados via plug-in do Maven (mesmo em projetos Ant)
  - **mvn sonar:sonar**



Geração de javadoc

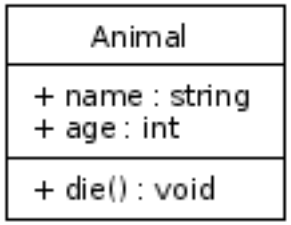
- UML Graph + Javadoc (a cada build)
  - http://www.umlgraph.org
  - http://www.graphviz.org



Graphviz (www.graphviz.org)

- Linguagem declarativa para gerar gráficos
- Exemplo com UML

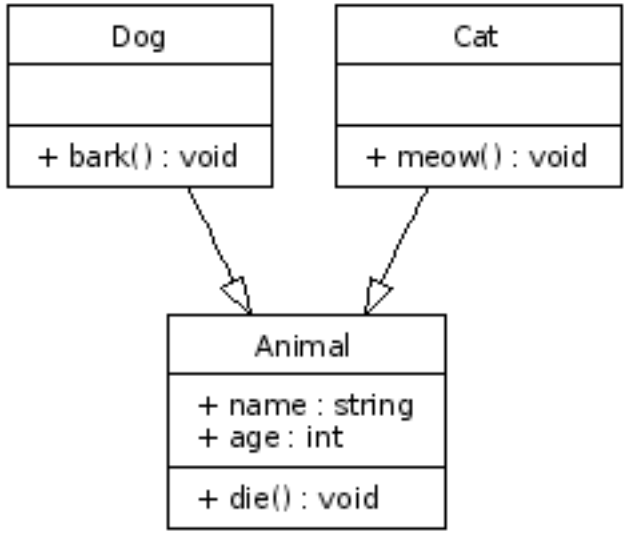
Animal [label = "{Animal|+ name : string\l+ age : int\l|+ die() : void\l}”]



edge [arrowhead = "empty”]

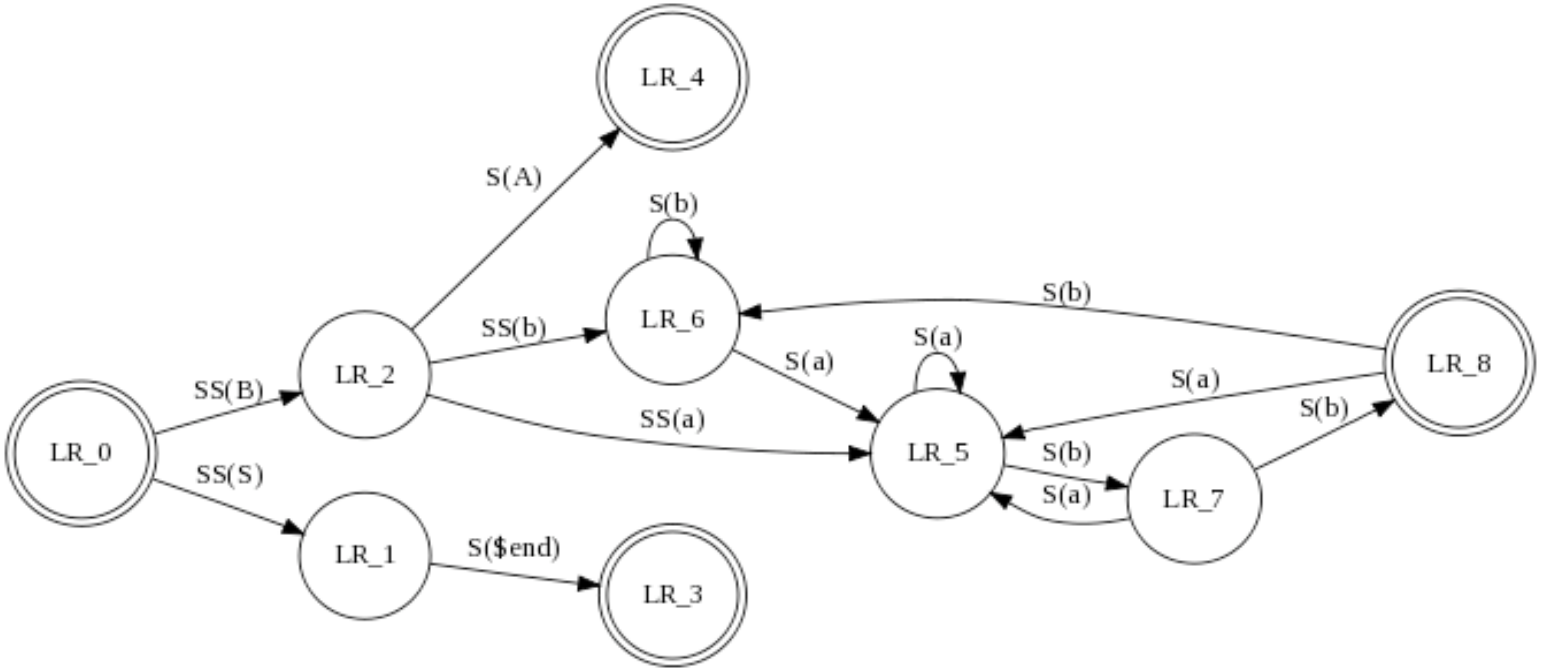
Dog -> Animal

Cat -> Animal



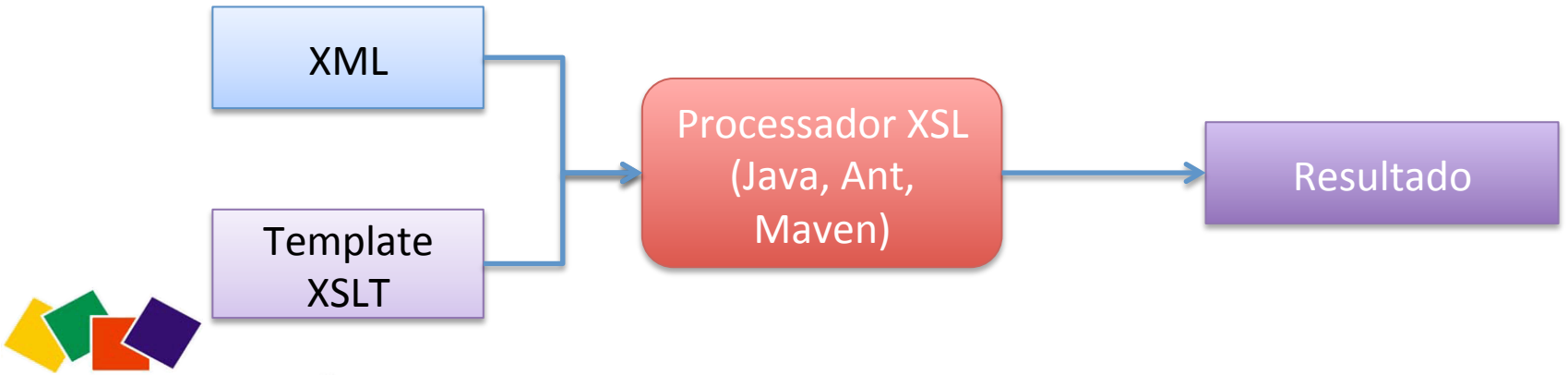
Graphviz

- Graphviz faz bem mais que UML
  - Use para gerar documentação gráfica complexa
  - Escreva “dot language” (ou gere a partir de XML com XSLT/XPath)



XSLT e XPath (www.w3.org)

- Pode ser usado via Java, tarefa do Ant, plug-in do Maven ou linha de comando
- Use XSLT/XPath para extrair dados de arquivos de configuração XML para
  - Gerar outros arquivos (de qualquer tipo)
  - Gerar PDF
  - Gerar documentação com tabelas HTML
  - Gerar documentação gráfica com Graphviz



Conclusão

- Vale a pena investir em tornar seu ambiente de desenvolvimento mais automatizado
  - Menos tempo perdido com **tarefas burocráticas**
  - **Informações atualizadas** e precisas sobre o estado atual do desenvolvimento e **qualidade** do código
  - **Subsídios** para tomadas de decisões em ambientes que adotam metodologias ágeis
- Há várias ferramentas open-source que podem ajudar nessa tarefa, como
  - Sonar, Hudson, Maven, Ant+Ivy
  - XSLT/Framemarker, Graphviz