Coding for the Future

The IDE (R)evolution for the Next Generation

John Ceccarelli
NetBeans Director
Oracle
Sept, 2014

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1 The Road to Here

2 Recent Trends in Tools

3 Opportunities for Growth and Innovation

4 What is NetBeans/Oracle Doing?

Program Agenda

1 **The Road to Here**

2 Recent Trends in Tools

3 Opportunities for Growth and Innovation

4 What is NetBeans/Oracle Doing?

The Road to Here

- Java IDE Wars of the 2000s
  - Open source IDEs took much of the revenue out of tools
    - Consolidation around Eclipse and NetBeans
  - IntelliJ proved elite tools could still make money

- Non-IDE tools reduced importance of IDEs
  - "Project" info stored/shared with Ant, Maven, Hudson/Jenkins
  - Devs can now use tool of their choice on shared projects

- Other ecosystems (Apple, Microsoft, IoT)
  - Tools as table stakes, not as revenue driver
    - MS Studio the notable exception, but Microsoft charges for everything

Program Agenda

1 The Road to Here

2 **Recent Trends in Tools**

3 Opportunities for Growth and Innovation

4 What is NetBeans/Oracle Doing?

IDE Coming to Your Language Soon

- Main trend in last 8 years is Java IDEs supporting scripting languages
  - 2002 – Java developers use emacs!
  - 2007 – Ruby developers use TextMate!
  - 2008 – PHP developers use notepad++!
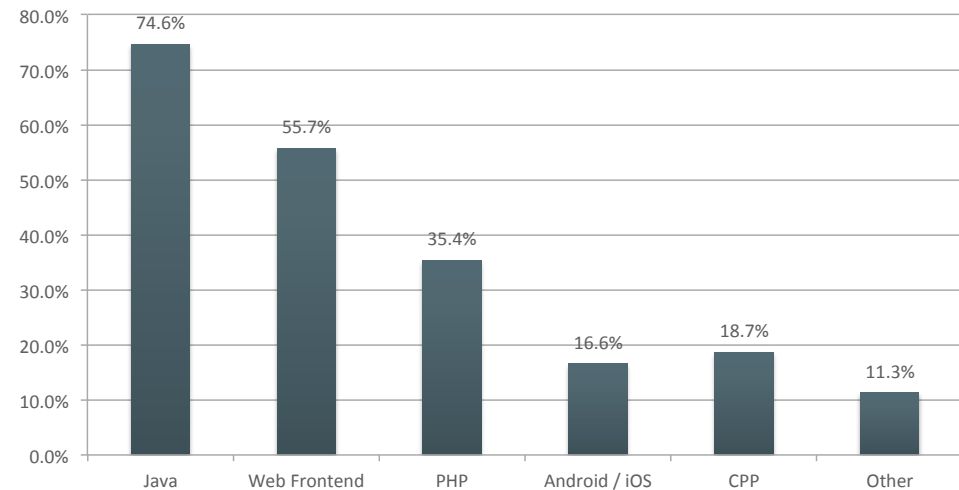 - 2014 – JavaScript developers use SublimeText!

Code completion? Blah!

Polyglot Is Here to Stay

- Strong uptake of NetBeans for non-Java languages

- JavaScript is everywhere

- JavaScript is difficult to tool
  - Loosely typed
  - Proliferation of JavaScript frameworks

- Continue to focus on Java back-ends with JavaScript clients

Languages Used for Development with NetBeans
NetBeans 7.4 Satisfaction Survey

Java	74.6%
Web Frontend	55.7%
PHP	35.4%
Android / iOS	16.6%
CPP	18.7%
Other	11.3%

Is Java Tooling Mature?

- All of the basics now available for free in most IDEs
  - Editor with code completion and refactoring
  - Debuger, Profiler, VCS integration, Maven support

- Most feature requests are enhancements, not evolution
  - WYSIWYG editor for my client technology
  - Support for my new language or framework

- Much more important than new features are:
  - Top-notch support for new JDK versions
  - Performance and stability of key features

Proliferation of Web IDEs

Proliferation of Web IDEs

- Mostly for web client technologies
  - Code can be run directly in browser
  - Don't need back-end
- Haven't taken off for compiled languages
  - Needs back-end – more on that later
- Just duplicating an IDE in the browser bad idea
  - You lose performance, feature richness, offline mode
  - What do you gain?

Program Agenda

1 ▶ The Road to Here

2 ▶ Recent Trends in Tools

3 ▶ Opportunities for Growth and Innovation

4 ▶ What is NetBeans/Oracle Doing?

Managed Systems

- Cloud systems that provide ready-to-use deployment services
  - No more maintaining app servers, deploying WARs, etc.
  - Deployment environment, runtime platform, lifecycle management, and standard functionality provided out of the box
  - Only do coding for customization and extension

- Examples:
  - Lower end: Amazon Elastic Beanstalk, Google AppEngine
  - Higher end: MBaaS, SalesForce.com

- Hard to emulate on local machine
  - Built-in debuggers, code editors for making small iterative changes a must

Combination with Developer Services

- Continuous Integration + Code Analysis + Git
  - Isolate builds for individual developers
  - Rule-based propagation based on same code analysis as in tools
    - Know when coding that something will fail
    - Configurable rules for your enterprise

- Continuous Integration + Indexing + Maven
  - Stop indexing static libraries!
  - Produce index during build-time
  - Store with Maven artifacts

Combination with Your Source Forge

- Run code analysis on your Git repository
- Tell you when browsing code:
  - This method called by 200 classes, be careful editing it!
  - Here are all the classes implementing this method
  - The test for this class fail 50% more often than the average for this project
  - Here is the test for this method
  - Etc.

Configuration Management

- Dev environment config is painful, but most of us put up with it
  - Reasonable investment on long-term project
- What if you change projects all the time?
  - Sustaining orgs, contractor-heavy shops, outsourcing firms
- Encapsulate development and execution environments
  - Access and start coding right away
  - From local IDE or web-based editor

Configuration Management Challenges

- What to use for encapsulation?
  - Full cloud-based VM
    - One VM per developer gets expensive
    - Local IDE with remote source directory problematic
  - Containers on hosted cloud service
    - Is it really the same as my production environment?
    - Can I get enough density with big enterprise stacks like WebLogic?
    - What about security?
    - Local IDE with remote source directory problematic
  - Locally downloaded containers
    - Can I get enough density with big enterprise stacks like WebLogic?
    - Local IDE with containerized source directory potentially problematic

Who Is Well Positioned to Deliver?

- Sweet spot is the intersection between various cloud services
  - Source management, IaaS, PaaS, Bug Tracking, Continuous Integration
  - To play in this sandbox you have to:
    - Have your own full-featured cloud
    - Partner with others

- IaaS providers such as Amazon and Google moving aggressively up-stack
  - Drinking smaller player's milkshakes

Program Agenda

1 The Road to Here

2 Recent Trends in Tools

3 Opportunities for Growth and Innovation

4 What is NetBeans/Oracle Doing?

NetBeans as the Engine of Cloud Developer Services

- Splitting core libs into headless services
  - Splitting out UI from logic
  - Adjusting to be multi-tenant

- First round of services
  - Source indexing and dependency analysis
  - Error checking/hints
  - Code completion
  - Maven-based build and deploy

- Not forking NetBeans!
  - Everything is in the public repositories

Where Will NetBeans Cloud Services Show Up?

- Oracle Developer Cloud Service
  - Syntax-aware source browsing

- Tailwind Code Editor
  - Part of Oracle Developer Cloud Service
  - In-place editing and deployment on Oracle Compute VMs

- Various Oracle Cloud Services
  - Spot-editing of source and configuration files

- Taking it slow
  - Listen to our customers, internal and external
  - Build what makes sense

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.