

Hackers Gotta Eat

Building a Company Around an Open Source Project

Kohsuke Kawaguchi
@kohsukekawa



Hello World

Kohsuke

- Hacker
- Created Hudson while working at Sun Microsystems
- Founded the Jenkins project
- Created InfraDNA after leaving Oracle
- Merged with CloudBees
- Now, CTO of CloudBees

Jenkins

- Open source automation server
- 140k known "master" instances
- Running on 500k+ machines worldwide
- Helps over 1M people implement continuous integration and continuous delivery

CloudBees

- Deep DevOps and Jenkins expertise.
- Enterprise-grade platform.
- Expert-level support.
- Major contributor to the Jenkins project

Open Source Business Models



Professional Services



Professional Services

- Consulting on best practices
- Custom, per-company, development on top of the project

Enterprise Products



Enterprise Products

- Build once and sell multiple times
- Enterprise software does not sell itself
 - OSS is good enough
 - Need critical mass, story, & sales
- Recurring revenue vs one-time revenue

Software as a Service



Software as a Service

- We'll run a version of the open source project for a monthly fee
- Pros:
 - Easy to understand product for most customers
 - Economies of scale for running the project.
- Cons:
 - Undervalued "I can run X on my workstation, it's not that hard."
 - Can easily spiral into custom requirements per-customer

Support



Support

- Provide experts-as-a-service!
- Pros:
 - Valued by larger organizations.
- Cons:
 - You need a bigger organization, process, and system
 - You have to hire the experts!

Hiring from
the project

Pros: Hiring from the project

- Straightforward to find motivated and qualified developers
- Open source developers already know how to work together (usually)
- Employing developers gives them more time to solve big problems
 - Paying developers to work on their passion project can give them the adequate support to make big break-throughs
- Often bring in sales leads :-)

Pros: Hiring from the project

- Straightforward to find motivated and qualified developers
- Open source developers already know how to work together (usually)
- Employing developers gives them more time to solve big problems
 - Paying developers to work on their passion project can give them the adequate support to make big break-throughs
- Often bring in sales leads :-)

Cons: Hiring from the project

- Can cause ambiguity for developers
 - "Am I on work-time or my-time?"
 - "The company wants to do X, but I want Y."
- Can risk the company being seen as "co-opting" the project.
 - When all the most vocal contributors are now employees...
- Can remove otherwise active participants from the project
 - Especially if you hire people and put them on Enterprise products

Interacting with
the project

The background features a dark blue area on the left, transitioning into a light blue curved shape in the center, and a cyan curved shape on the right. The text is positioned on the dark blue area.

Interacting with the Project

- Define the boundaries within the company for interacting with the project
- Ensure the company is "on the same page" before spinning cycles in the project
 - Volunteer contributors don't have patience for inconsistencies from a company
- Accommodate community feedback in development time

Development and Collaboration



Development and Collaboration

- Best to follow the Red Hat model and "work upstream" as much as possible
 - Use "upstream" repositories
 - Manage backlogs (when possible) in "upstream" issue trackers
 - Follow "upstream" review and merging practices
- Determining transparency in product roadmap can be a challenge
 - Teams might be working on upstream and "enterprise" functionality in the same sprint

Branding, Trademarks, and Legalese



Branding, Trademarks, and Legalese

- Being recognized as "the X people" is beneficial
 - This is a big part of how people perceive value in what you offer
- **Who owns the trademark?**
 - Extremely important to have clarity on where trademark ownership lies.
 - Company must respect the mark's usage restrictions and guidelines.

Branding, Trademarks, and Legalese

- Licenses 101
 - Comply with existing licenses of software used/extended.
 - Explain to employees copyright ownership of their work.
 - Determine suitable licenses for new software created and open sourced.
- "Open Source" vs. "Free Software"
 - Important to understand the differences.

Why Bother

Open Source is
"eating the world"



Open source is "eating the world"

- For more and more organizations open source is a **better** choice
 - More flexible
 - Higher quality
- Lower barrier to adoption means a larger potential market for companies built around open source projects

We can solve better problems



We can solve better problems

- Participation exposes the project to a broad set of perspectives
 - Open source users and contributors will use software very differently than you might expect.
- Open source means many "itches" and low-hanging fruit are resolved by contributors.

We can solve better problems

- Participation exposes the project to a broad set of perspectives
 - Open source users and contributors will use software very differently than you might expect.
- Open source means many "itches" and low-hanging fruit are resolved by contributors.

We can solve better problems

- Participation exposes the project to a broad set of perspectives
 - Open source users and contributors will use software very differently than you might expect.
- Open source means many "itches" and low-hanging fruit are resolved by contributors.

We can solve better problems

- Participation exposes the project to a broad set of perspectives
 - Open source users and contributors will use software very differently than you might expect.
- Open source means many "itches" and low-hanging fruit are resolved by contributors.

We can solve better problems

- Participation exposes the project to a broad set of perspectives
 - Open source users and contributors will use software very differently than you might expect.
- Open source means many "itches" and low-hanging fruit are resolved by contributors.

Blue Ocean

Blue Ocean

- New user experience for Jenkins
- jenkins.io/projects/blueocean
- Built from the ground up for Continuous Delivery and modern developer needs

Dashboard

New Pipeline

Favorites

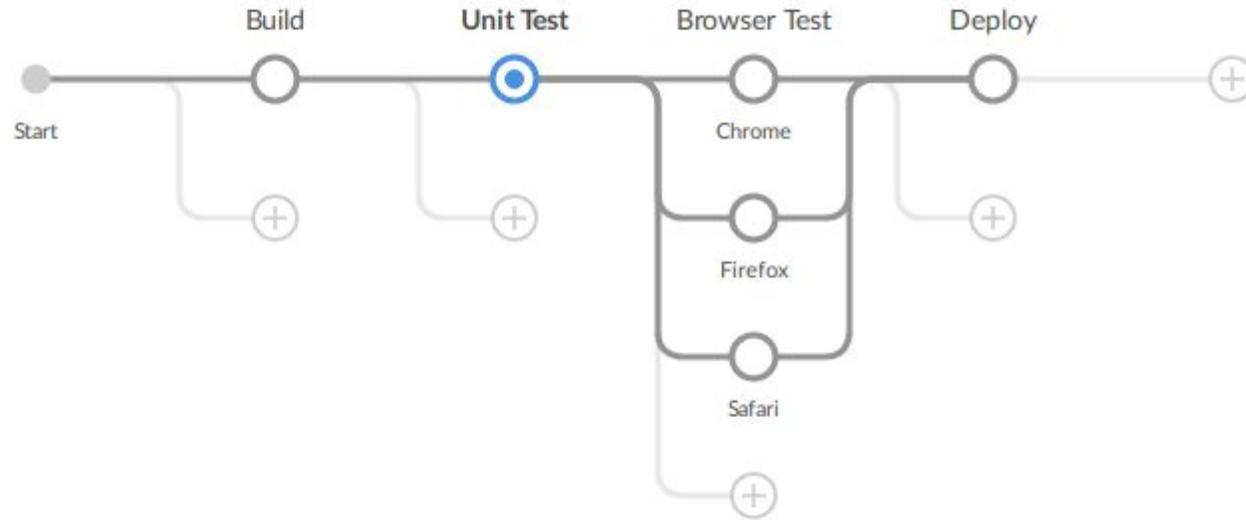
<input checked="" type="radio"/> Core / jenkins	🔗 master	🔗 #6049dfe	🔍 ⭐
<input type="radio"/> Infra / azure	🔗 master	🔗 #48bc9ac	🔍 ⭐
<input checked="" type="radio"/> Infra / jenkins-infra	🔗 production	🔗 #6f3fab8	🔄 ▶ ⭐
<input checked="" type="radio"/> Infra / infra-statistics	🔗 master	🔗 #328ddd6	🔄 ▶ ⭐

Name	Health	Branches	PR	
Core / jenkins	🔴	14 failing	30 failing	⭐
Core / pom	🔴	-	1 passing	⭐
Core / remoting	🌩	3 failing	3 failing	⭐
Core / winstone	🔴	1 passing	1 passing	⭐

Create Pipeline

New

Load/Save



← Unit Test



Print Message

Running unit test suite

Shell Script

```
./gradlew -Si check
```

+ Add step

Benefits to the Project



Benefits to the Project

- More people, not just code contributors, can help move the project forward
 - Designers
 - Writers
 - Many, many, more
- Corporate involvement lends credibility to "main street" businesses
 - The company can become a lightning rod for bringing new corporate contributions

Hackers Gotta Eat

Hackers Gotta Eat

- Open source software is valuable.
- People's time is valuable.

Questions?

cloudbees®



Thank you



Boundaries and Challenges



Boundaries and Challenges

- Becoming an "Elephant in the Room."
 - If the majority of active contributors are employees, it can discourage other contributors and cause mistrust.
- Challenging to invest in deeper architectural changes.
 - Hard to "sell" internal refactorings and updates.