

CASE STUDY

VAADIN

COMPANY

SCORE CARD

- **Size of development team:** 60
- **Technologies in use:** Jetty, Tomcat, GlassFish, JBoss, WebLogic, WebSphere, Java EE, Spring, Vaadin... and many more
- **Types of apps being built:** various UI and system apps for customer projects
- **App redeploy time before JRebel:** 20s - 1m depending on environment
- **Number of redeploys per hour:** 2 (because they took time)
- **Number of instant reloads with JRebel per hour:** 60
- **Time saved per day using JRebel:** at least 10%

Company Profile

Vaadin Ltd is the company behind the popular **Vaadin Framework**, an open source software development tool for building high-quality web user interfaces for business software. Since 2000, the technology was originally developed based on the programming challenges in a large-scale health care system. Java was chosen to be the platform and the project was open sourced later in 2002. Today, Vaadin is used daily by thousands of professional software developers in more than 170 countries. While the free technology is used all around the world, the company is headquartered in Turku, Finland. Most of Vaadin's 60+ developers work here, but the company also has offices in California and Frankfurt.



OUR VAADIN EXPERT

Vaadin have both a services and a consulting team as well as an R&D/Engineering team which make use of JRebel internally. We caught up with **Petri Heinonen**, a Vaadin Expert, program manager and team lead who has been with Vaadin for over 4 years. Petri works in services on customer projects, building UI and backend systems for customer projects, as well as consulting and training. He has used JRebel for over 2 years now. Projects he works on can range from anything between a week and a couple of years. As a result, Petri has used JRebel across a wide range of application servers, including Jetty, Tomcat (these two are his preference due to size and speed), JBoss, GlassFish and WebLogic.

DEVELOPMENT ENVIRONMENT

Petri's development environment itself also varies. The IDE he uses is Eclipse most of the time, but sometimes the project calls for IntelliJ IDEA. His build environment is usually Maven and Ant-oriented, and less frequently, Gradle. So he uses the JRebel support from the Ant and Maven plugins, so the configuration is all handled automatically on the build side.

JRebel usage

Petri really has road-tested JRebel on many different environments, including IDEs and build tools. We asked him: "Did you have any problems in all these years?"

“ I only had a problem with the JRebel set up once in all my experiences, on using Eclipse and GlassFish. It turned out Eclipse was compiling to a different place to where JRebel was looking, so didn't pick up my changes. Once I fixed the config file, all was working again. I can't say I was happy coding without JRebel during that time - I had to develop with the server in debug mode, but still had to redeploy frequently, which really hurt!

”

Is there anything you want JRebel to do which it doesn't currently do?

“ Of course you still have to redeploy when updating superclasses in JRebel ”

Check out our [JRebel v6 Beta](#), where support is now included for superclass changes! ;)

Time Saved and benefits gained

The redeploy time for Petri varies on the environment and application which he is working on at the time, but all environments he has worked on give some kind of productivity hit as redeploys don't go away easily.



I hate them so much, I actually try to architect my development environment to try to avoid them, particularly if the project uses Liferay. When I'm not using JRebel I redeploy and test my code once every 30 minutes or so, but when I use JRebel, I test every minute as my code is reloaded instantly.



This instant feedback lowers the number of bugs in your code because you test your features instantly, if you can't test instantly you sometimes forget to test some of the things you've effectively 'coded blindly'.



Time Saved and benefits gained

Another area which Vaadin really gain value out of JRebel is in the persistence of state across a class reload. This maintains the point in time which the user is at throughout the reload of the underlying class.

“ State is very important during our testing. To get to a certain place in an application using a more complex UI design, you often have to follow a trail to get into a specific state - sometimes you can get to it easily, in just a few clicks, other times it's very hard - JRebel you can just refresh and maintain state - this is one of the primary reasons as to why I use JRebel. ”

Time can be saved in many ways using JRebel, whether it's the easily-measurable waiting time for a application to redeploy, or the harder to measure early capture of bugs due to the instant testing cycles, or the maintained focus during a class reload.

“ The time saving for me is significant. I'd say around 10% of my time in a day, sometimes more, is saved because of JRebel, depending on the project. ”

Before Vaadin introduced the [Sass](#) (Syntactically Awesome Stylesheets) compiler JRebel also updated the CSS updates in an application, meaning theme changes in an application were seen instantly when they were updated in your IDE.

Summary

JRebel is used during the development of Vaadin as well as the services work to create apps and solutions which use Vaadin. It eliminates the need for constant redeploying, providing **an instant feedback loop between developing code and testing code.**

This means **Petri doesn't have to 'code blindly'** as he can test everything he writes instantly.

Petri enjoys coding **without having to wait 20 seconds to > 1 minute for build/compile and app server restarts**, but the real value for him is the **persistence of state in his app across class reloads.**

This is the feature which saves Petri time and sanity, and is one of the main reasons why he uses JRebel. He would like JRebel to reload superclass changes, and is currently investigating the **JRebel v6 Beta** which does exactly that, among other things.



↙ Contact Us

Twitter: [@JRebel](#)

Web: <http://zeroturnaround.com>

Email: info@zeroturnaround.com

Estonia

Ülikooli 2, 5th floor
Tartu, Estonia, 51003
Phone: +372 740 4533

USA

399 Boylston St., 3rd fl.
Boston, MA, USA, 02116
Phone: 1(857)277-1199

Czech Republic

Osadní 35 - Building B
Prague, Czech Republic 170 00
Phone: +372 740 4533