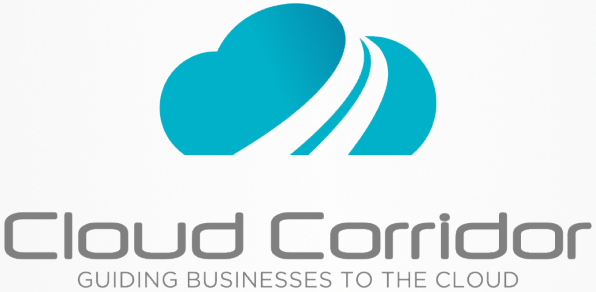


Continuous Integration

What you (probably) don’t know and why you should care?

Who am I?

Łukasz Rybka

Co-lecturer
(2014)

Co-lecturer
(2015)

Lecturer
(2015)

Trainer
(2016)

CTO
(2017)

Software Architect
(2017)

+ Physicist by education

+ With IT industry for 10+ years

Agenda

// Continuous Integration – what it is and what it's not?

// What kind of problems does it solve?

// How to apply the rules of CI to your day-to-day development life?

// Continuous Integration servers – what are these for?

// Patterns and anti-patterns of working with teams

// I know CI – what's next for me?

Disclaimer

// Have a question? Please ask! ☺

// Slides are for me to navigate trough the presentation, not much of a value to the audience in them.

// Everything I say is opinionated

// Truly sorry for my English, I’m not a native speaker

Question

Who has non-academic experience?

Question

Who has commercial experience?

Question

Who knows Agile and Scrum methodologies?

Agile Manifesto

Motivation

„A man who dares to waste one hour of time has not discovered the value of life”

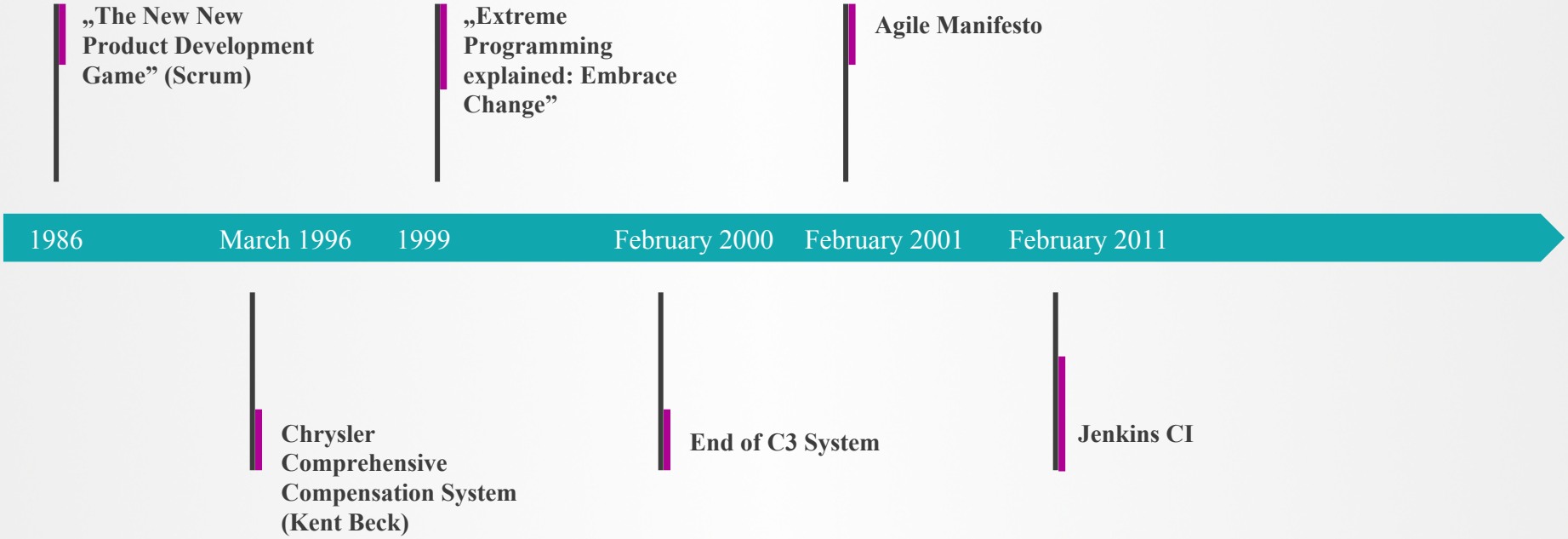
Charles Darwin

Motivation

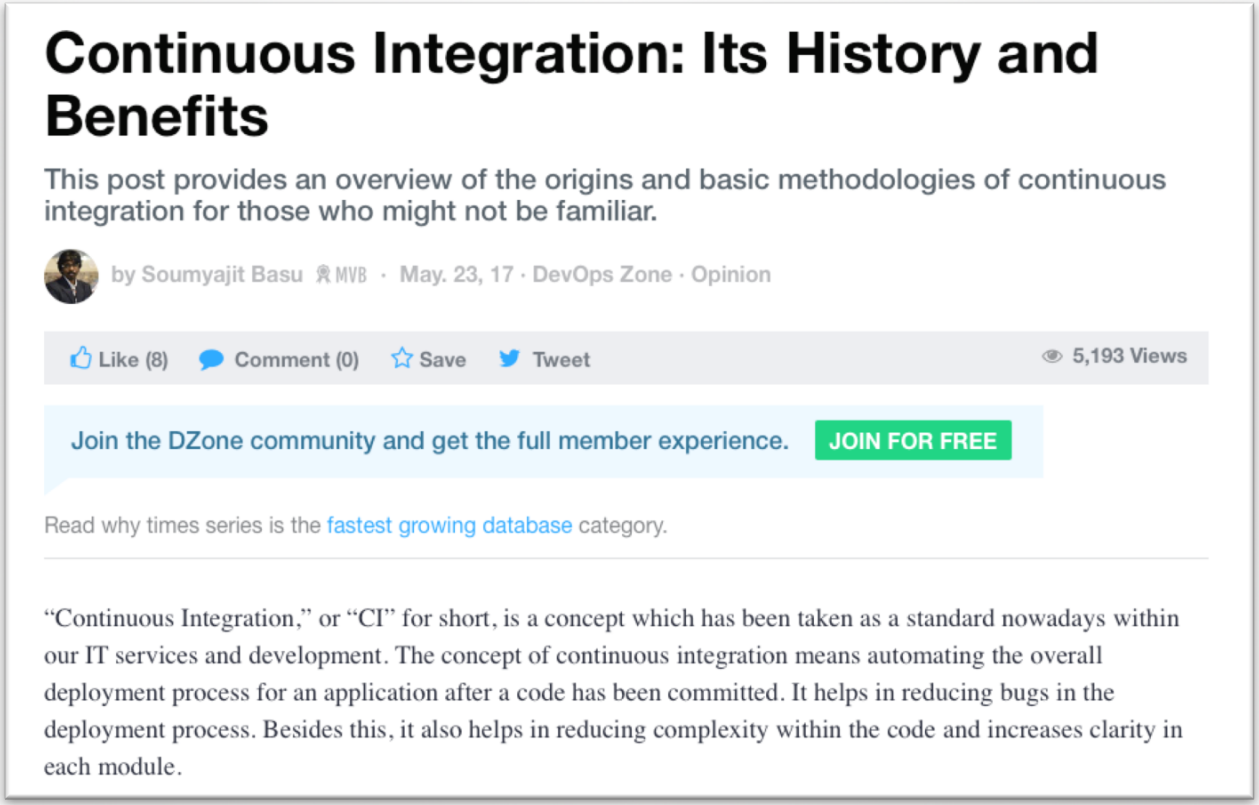
„I choose a lazy person to do a hard job. Because a lazy person will find an easy way to do it.”

Bill Gates

A bit of history…



Common Misconception



Continuous Integration: Its History and Benefits

This post provides an overview of the origins and basic methodologies of continuous integration for those who might not be familiar.

by Soumyajit Basu MVB · May. 23, 17 · DevOps Zone · Opinion

Like (8) Comment (0) Save Tweet 5,193 Views

Join the DZone community and get the full member experience. JOIN FOR FREE

Read why times series is the fastest growing database category.

"Continuous Integration," or "CI" for short, is a concept which has been taken as a standard nowadays within our IT services and development. The concept of continuous integration means automating the overall deployment process for an application after a code has been committed. It helps in reducing bugs in the deployment process. Besides this, it also helps in reducing complexity within the code and increases clarity in each module.

Other incorrect definitions

// „Tool for running unit tests”

// „Tool for deploying newest changes to development and production environments”

// „Tool for static code analysis”

// „Tool for preparing binary distribution of our product”

What it actually is?

// Originated in Extreme Programming (XP)

// **Practice in software engineering**

// Assumes frequent integrations next, smaller changes in code with main development branch

// Commonly teams assumes multiple integrations per day, about 1 integration per programmer each day

What are the benefits

// Avoiding „merging hell”

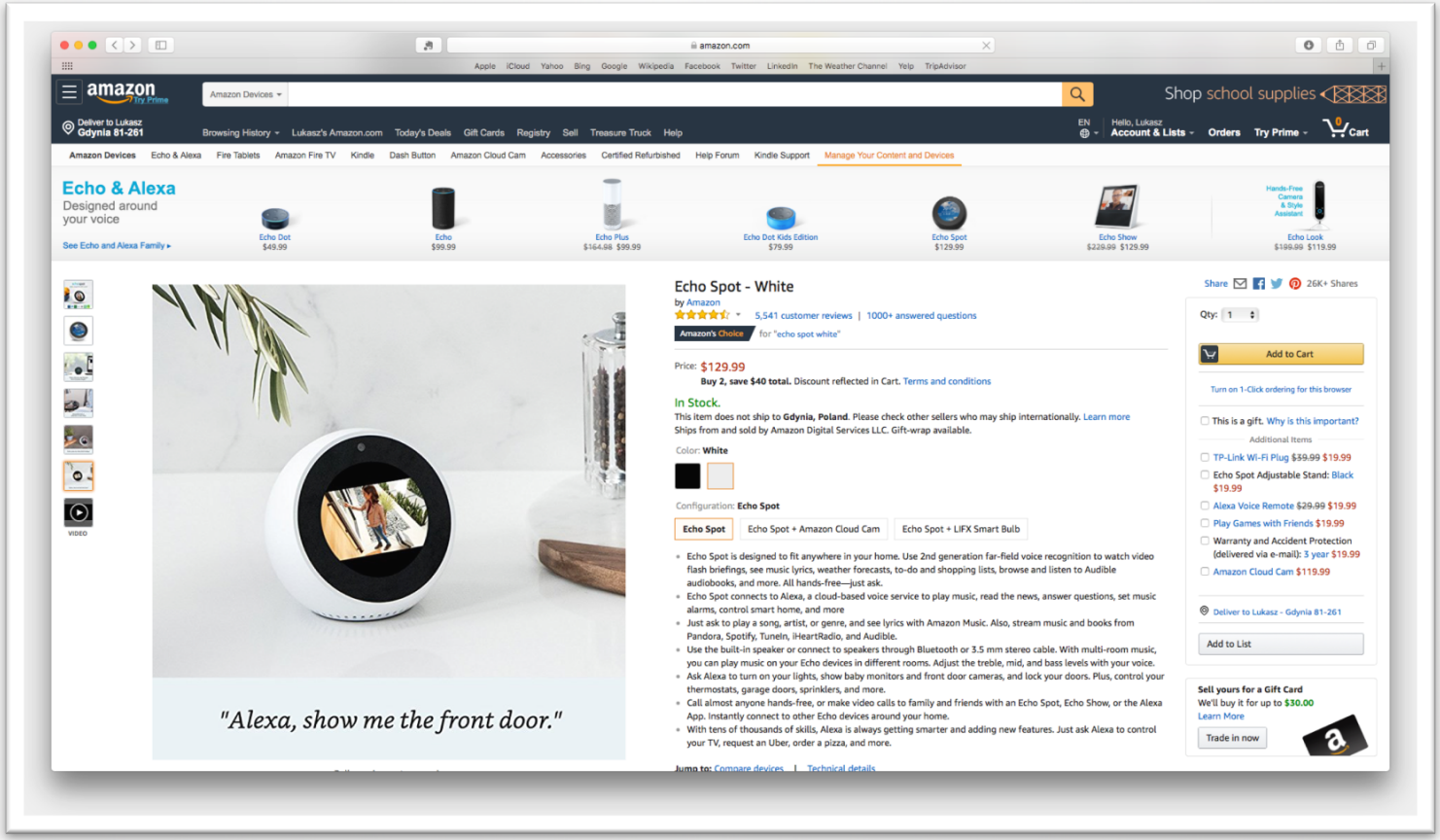
// Helps to quickly catch architectural errors

// Makes conversation between development team and product owner a lot easier

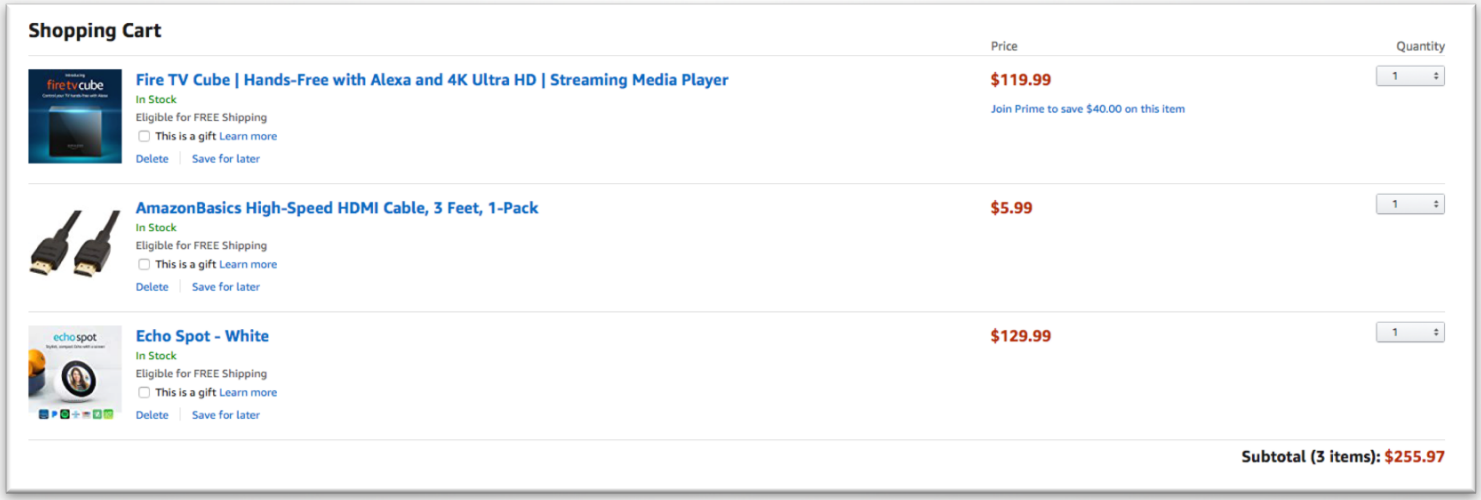
// Introduces new, more efficient and standarized process for working with tasks

Example task/issue

Example task/issue



Example task/issue



Typical questions from management

„How long will it take?”

„When can I see it?”

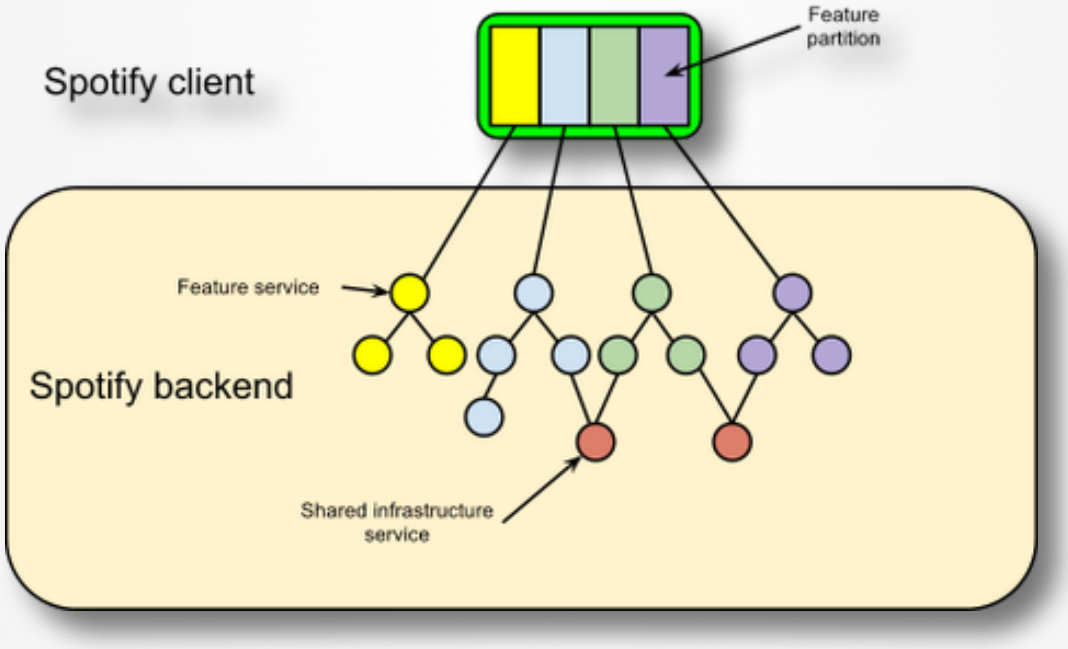
„We only asked for one checkbox, why do you need 3 weeks for it?”

„Continuous” Approach

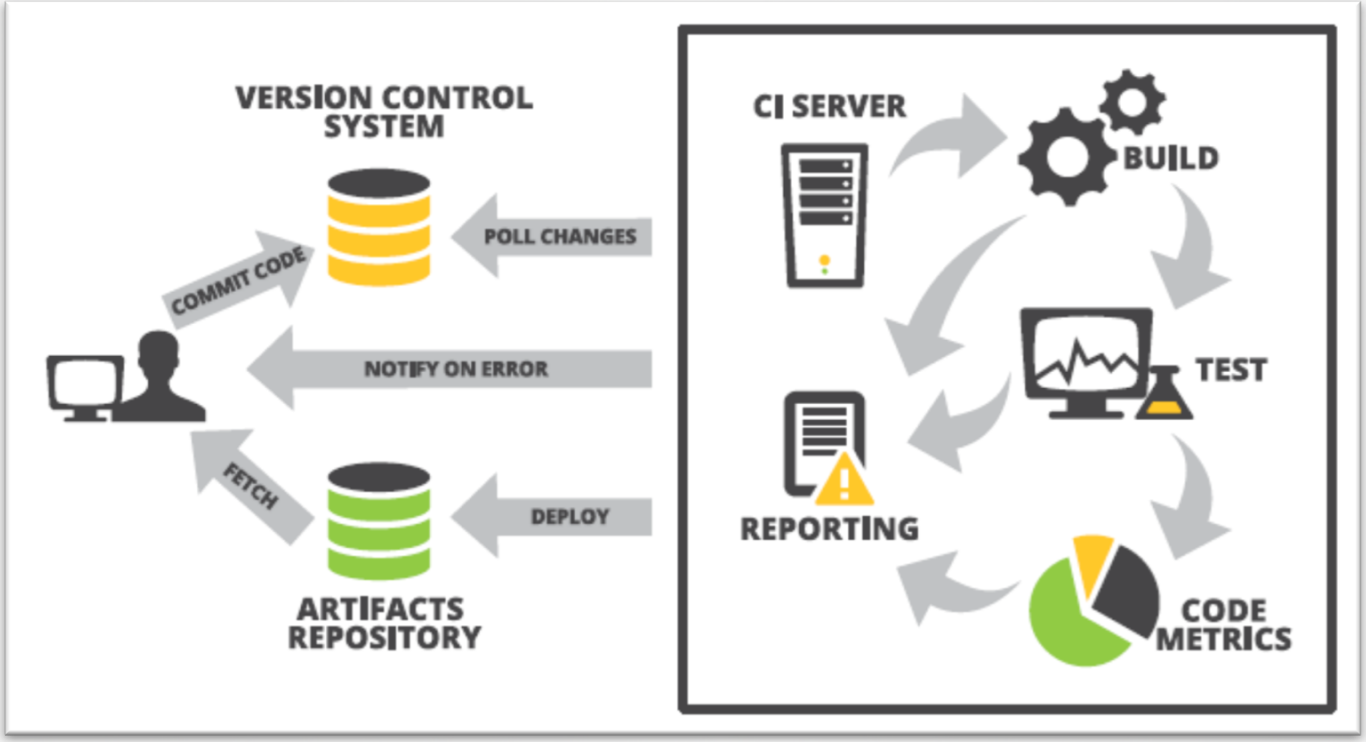
Typical concerns

„What if I don’t want to ship unfinished feature to clients and deployment is very close?”

Feature toggle – Spotify architecture



Automation


VERSION CONTROL SYSTEM
POLL CHANGES
COMMIT CODE
NOTIFY ON ERROR
FETCH
DEPLOY
ARTIFACTS REPOSITORY
CI SERVER
BUILD
TEST
REPORTING
CODE METRICS

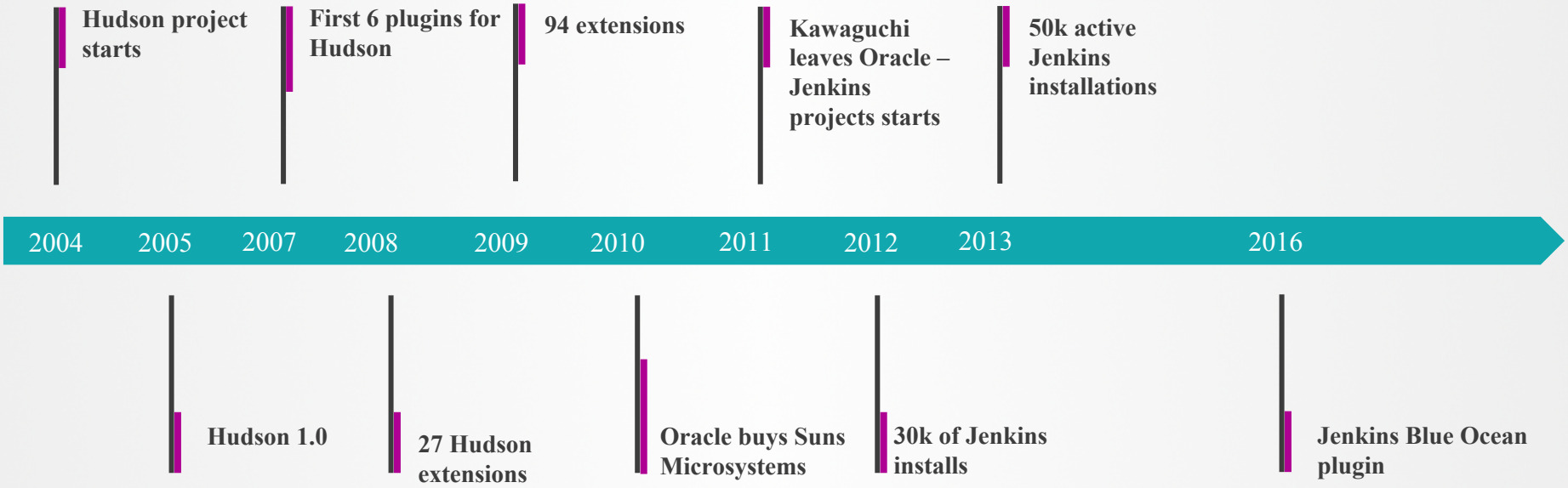
Continuous Integration servers

Jenkins

„I had this great idea of a web framework, and I wanted an excuse to actually implement it. This was around the time when everyone had his own framework, so I suppose I couldn't help myself."

Kohsuke Kawaguchi

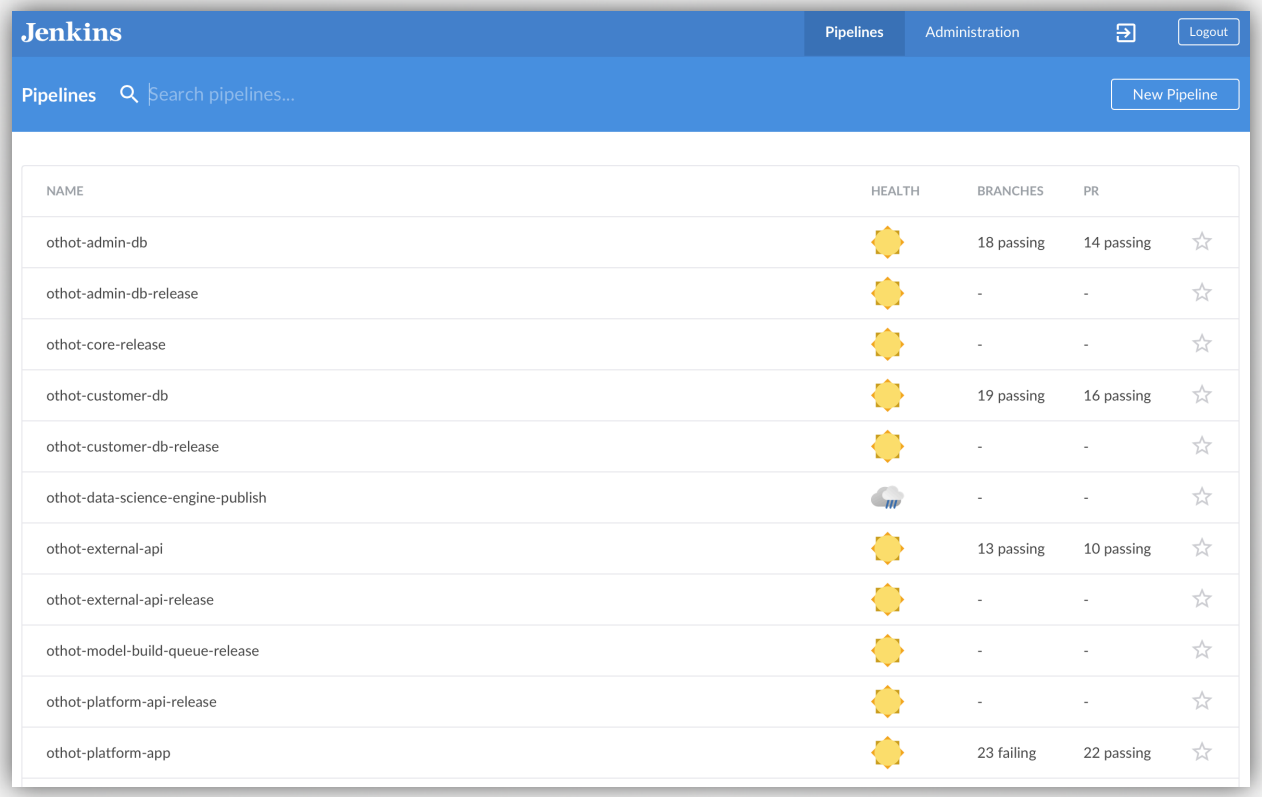
Jenkins history



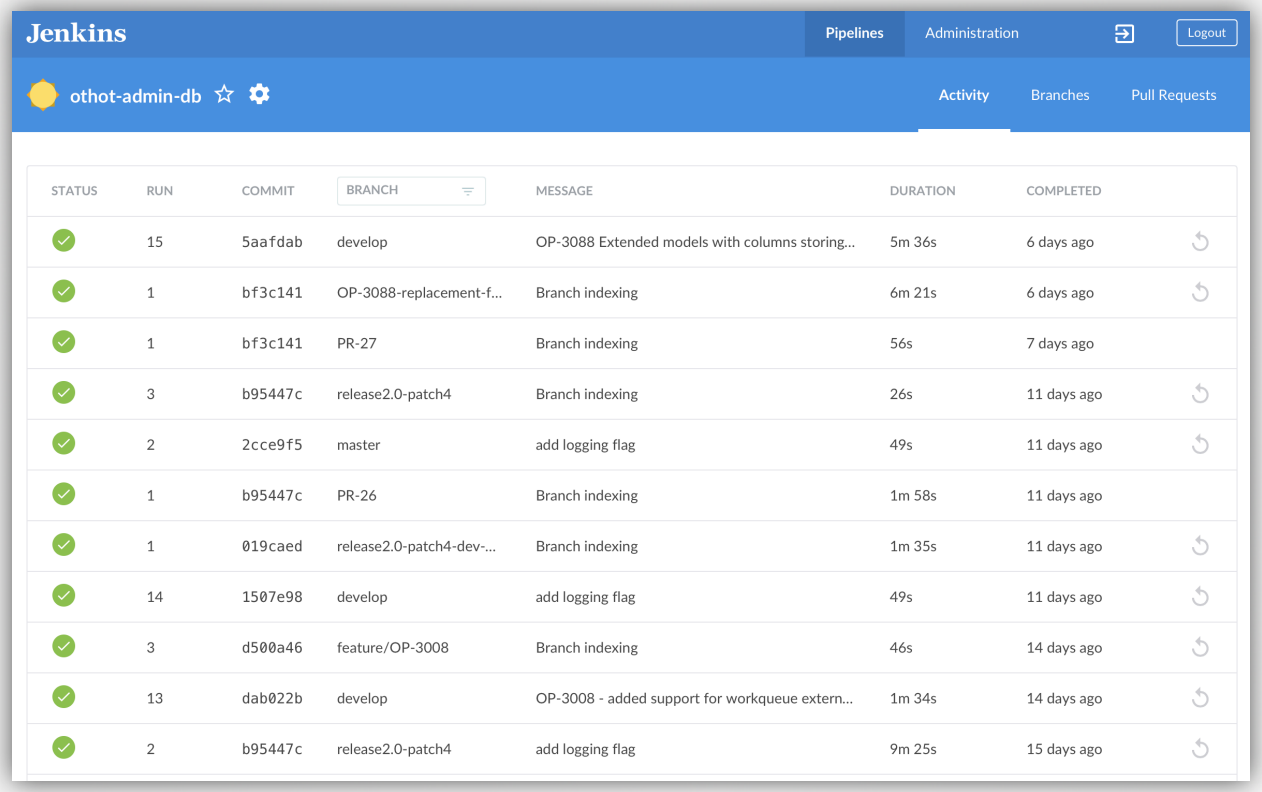
Jenkins features

// Easy to install – Docker image
// Weekly updates
// Build as a group of core plugins
// Huge database of plugins
// Easy to learn and use
// Platform and programming language agnostic
// Elastic
// Built-in scripting mechanism
// Very scalable (master/slave model)

Jenkins interface

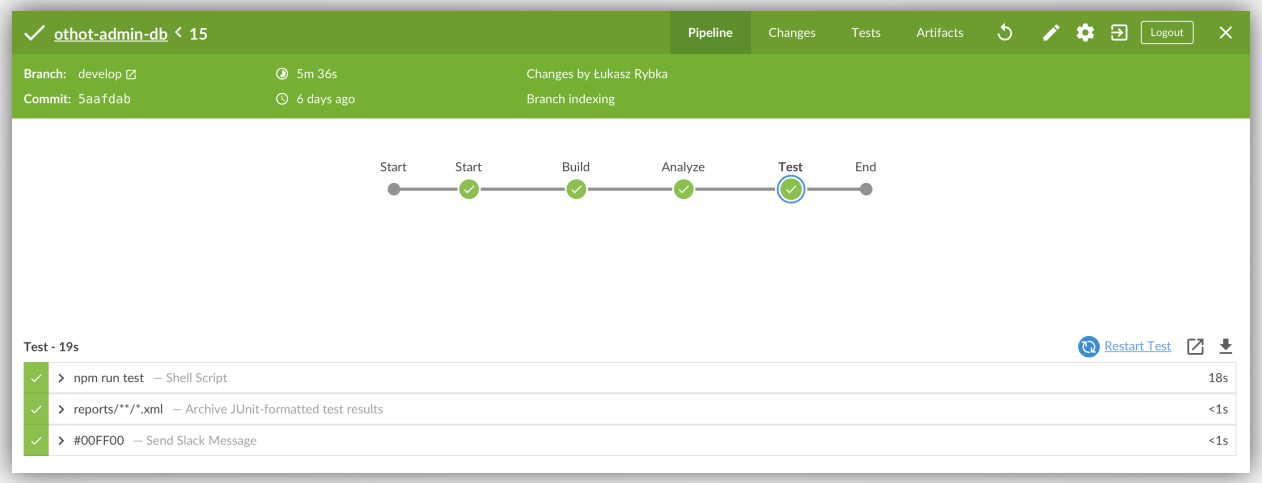


Jenkins interface

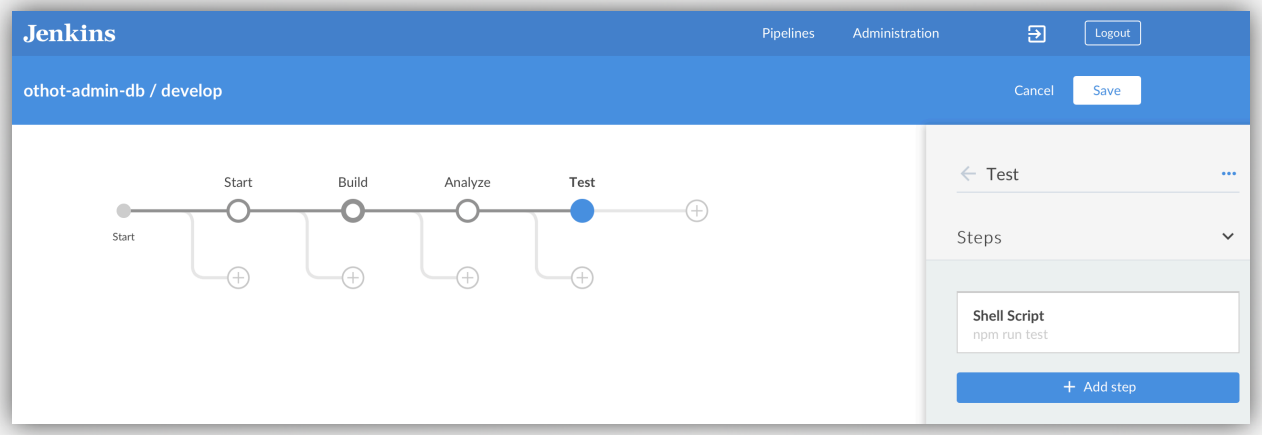


STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	15	5aafdab	develop	OP-3088 Extended models with columns storing...	5m 36s	6 days ago
✓	1	bf3c141	OP-3088-replacement-f...	Branch indexing	6m 21s	6 days ago
✓	1	bf3c141	PR-27	Branch indexing	56s	7 days ago
✓	3	b95447c	release2.0-patch4	Branch indexing	26s	11 days ago
✓	2	2cce9f5	master	add logging flag	49s	11 days ago
✓	1	b95447c	PR-26	Branch indexing	1m 58s	11 days ago
✓	1	019caed	release2.0-patch4-dev-...	Branch indexing	1m 35s	11 days ago
✓	14	1507e98	develop	add logging flag	49s	11 days ago
✓	3	d500a46	feature/OP-3008	Branch indexing	46s	14 days ago
✓	13	dab022b	develop	OP-3008 - added support for workqueue extern...	1m 34s	14 days ago
✓	2	b95447c	release2.0-patch4	add logging flag	9m 25s	15 days ago

Jenkins interface



Jenkins interface



What do I need?

// Version control system (Git)
// Application build tool (Maven)
// Versioning tool (Maven)
// Code review tool (GitHub)
// Scripting language (Maven/bash)
// Artifacts repository (Nexus)
// Testing environment (Google App Engine)

Joel Spolsky

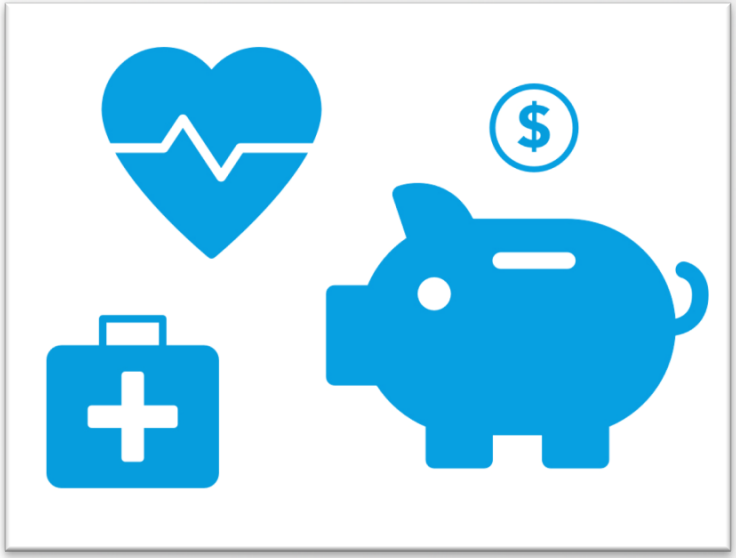
Avram Joel Spolsky (born 1965) is a software engineer and writer. He is the author of Joel on Software, a blog on software development, and the creator of the project management software Trello.[2] He was a Program Manager on the Microsoft Excel team between 1991 and 1994. He later founded Fog Creek Software in 2000 and launched the Joel on Software blog. In 2008, he launched the Stack Overflow programmer Q&A site in collaboration with Jeff Atwood. Using the Stack Exchange software product which powers Stack Overflow, the Stack Exchange Network now hosts over 170 Q&A sites.

Joel Spolsky Test

01. **Do you use source control?**
02. **Can you make a build in one step?**
03. **Do you have daily builds?**
04. Do you have a bug database?
05. Do you fix bugs before writing new code?
06. Do you have an up-to-date schedule?
07. Do you have a spec?
08. Do programmers have quiet working conditions?
09. **Do you use the best tools money can buy?***
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability tests?

Overall benefits

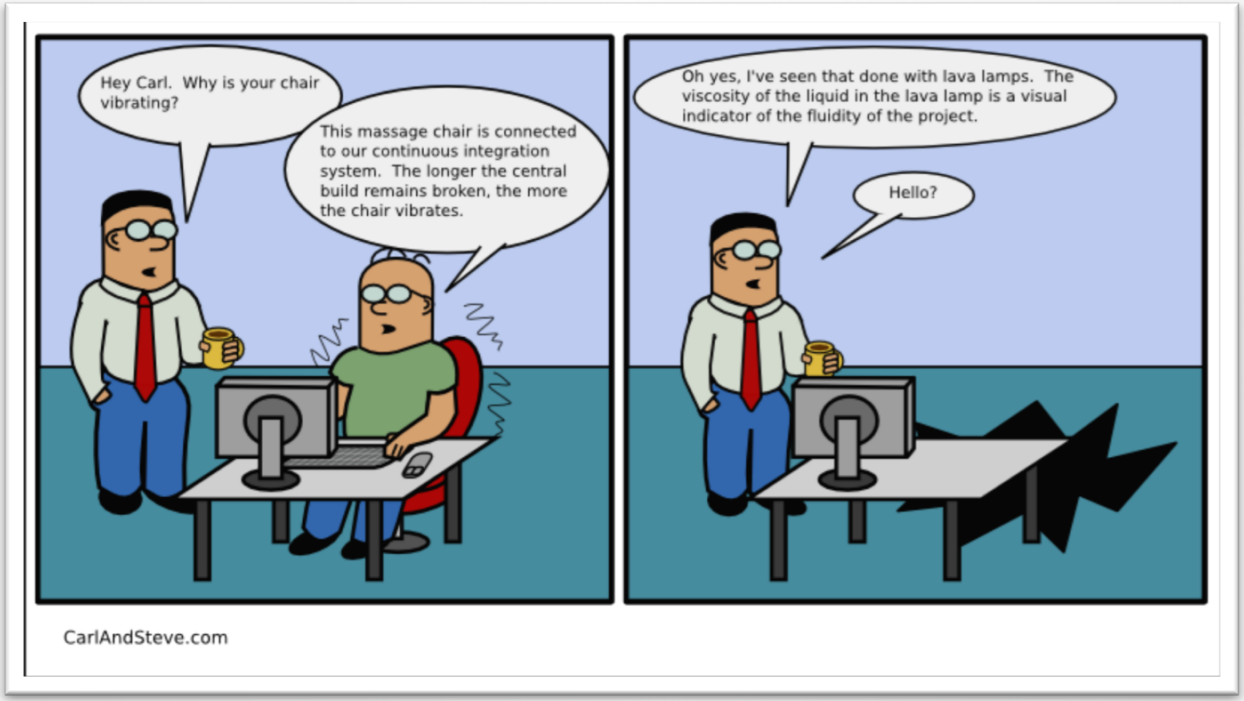
// Speed of delivering new features
// Higher quality
// Better application architecture
// Product more aligned with the vision
// Flexibility in working on single feature by more than one developer
// Easy QA process
// Easy production bugs investigation



What's next?

// Continuous Deployment / Delivery (CD)
// Microservices pattern
// Nightly Builds
// Automated Integration/Functional/… tests
// A/B testing
// …

Digression


Hey Carl. Why is your chair vibrating?
This massage chair is connected to our continuous integration system. The longer the central build remains broken, the more the chair vibrates.
Oh yes, I've seen that done with lava lamps. The viscosity of the liquid in the lava lamp is a visual indicator of the fluidity of the project.
Hello?
CarlAndSteve.com

EXIT
STORM

Contact

http://www.dragonia.org.pl