

# Типовые **проблемы** разработки ПО в больших проектах



Rustam Mehmandarov  
*@RMehmandarov*

# Typical **Challenges** in Software Development at a Large Scale



Rustam Mehmandarov  
*@RMehmandarov*



# Hello!

I am **Rustam Mehmandarov**



computas



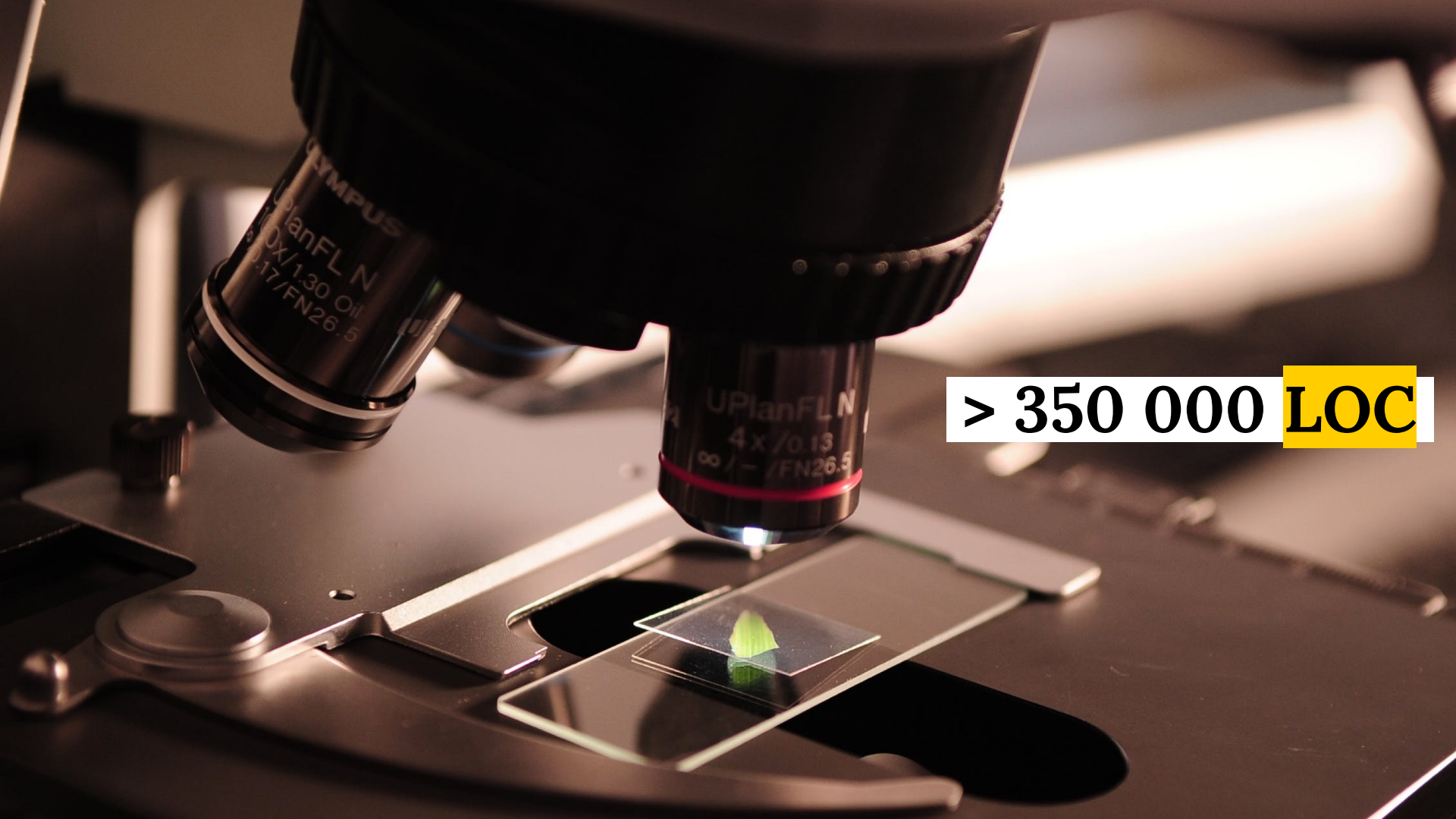
javaBin



JavaZone



Java™  
Champions

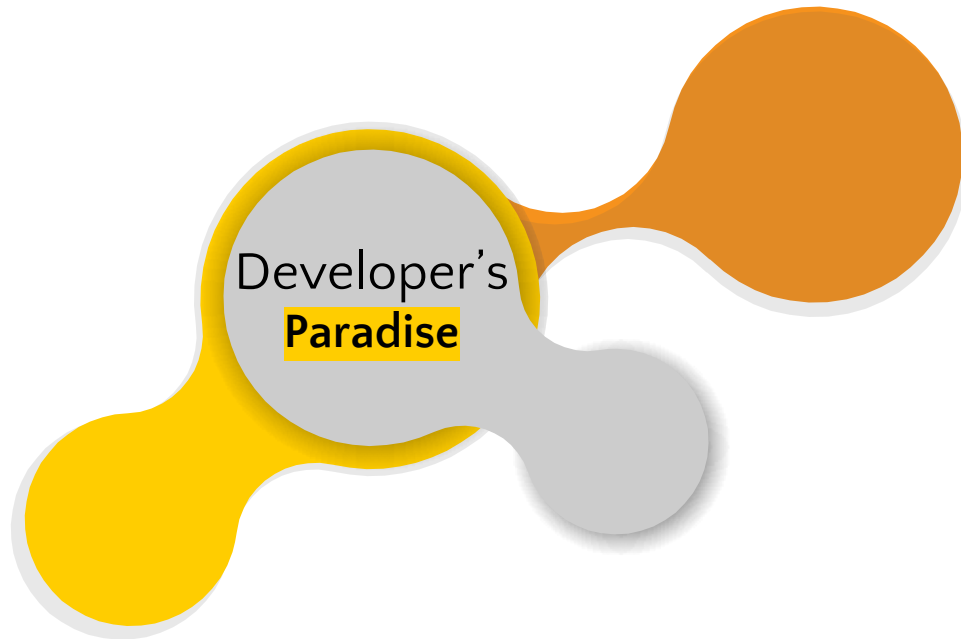


> 350 000 LOC



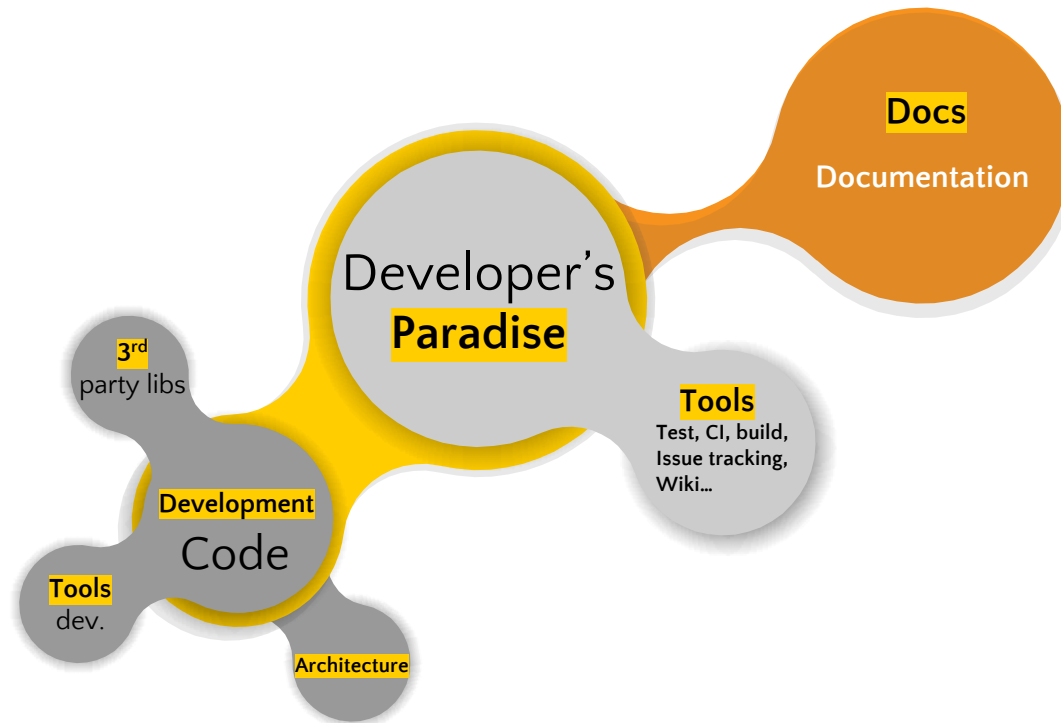


# A **paradise** ...or a nightmare





# The **moving** parts





# Meet My Little Helpers







**90s...**

...for the developers





# 90s Development

# Java in Notepad/Emacs/Vim





# 90s Versioning

# Visual SourceSafe

# CVS

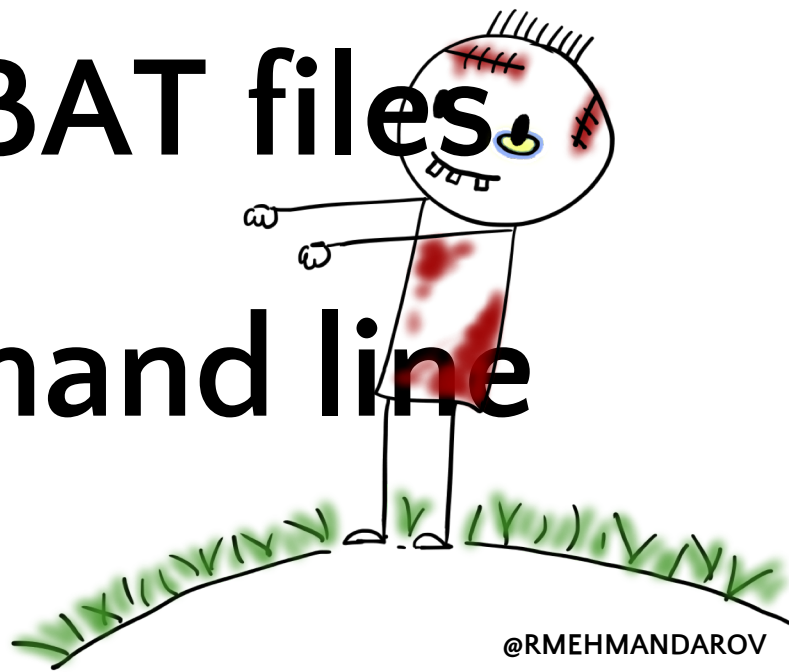




# 90s Compile and Build

# CLASSPATH in BAT files

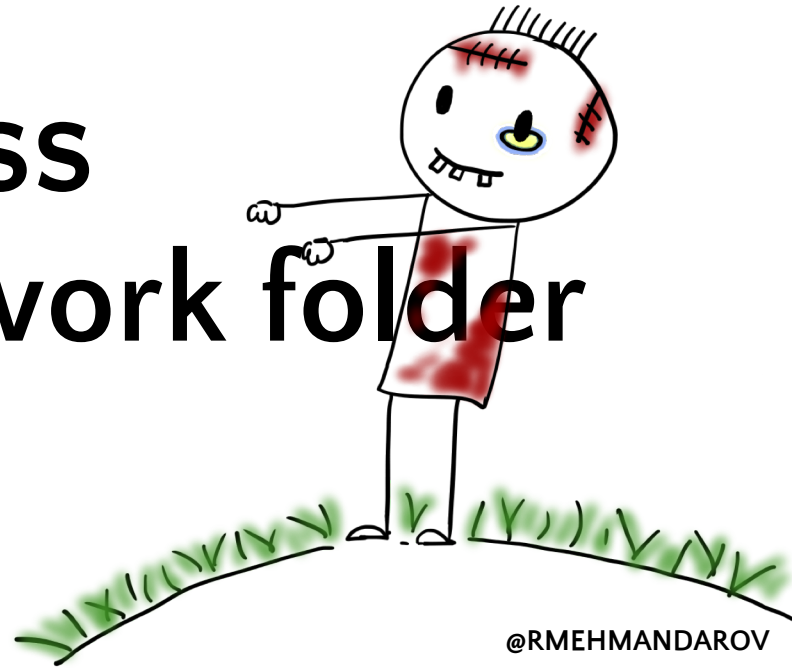
# javac and command line





# 90s Issue Tracking

**MS Access  
..in a shared network folder**







**brian wisti**

@brianwisti

Follow



YOU ARE IN A LEGACY CODEBASE

> RUN TESTS  
YOU HAVE NO TESTS

> READ SPEC  
YOU HAVE NO SPEC

> WRITE FIX  
YOU ARE EATEN BY AN ELDER CODE  
HACK.

9:30 PM - 25 Aug 2014

7,422 Retweets 5,389 Likes



154



7.4K



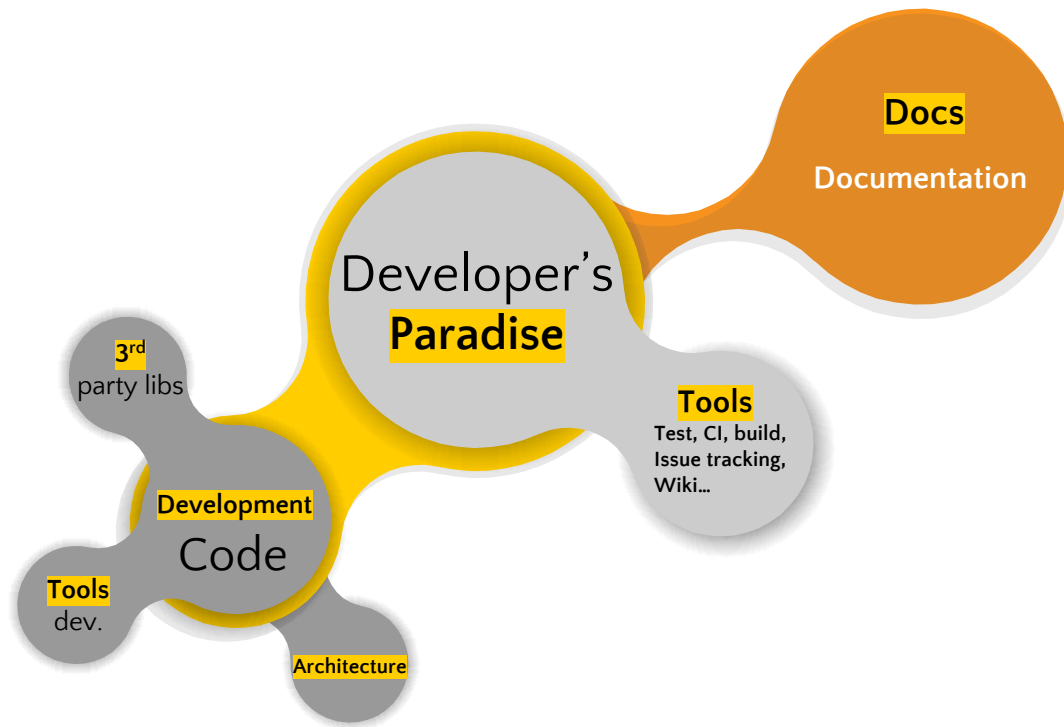
5.4K







# The **moving** parts





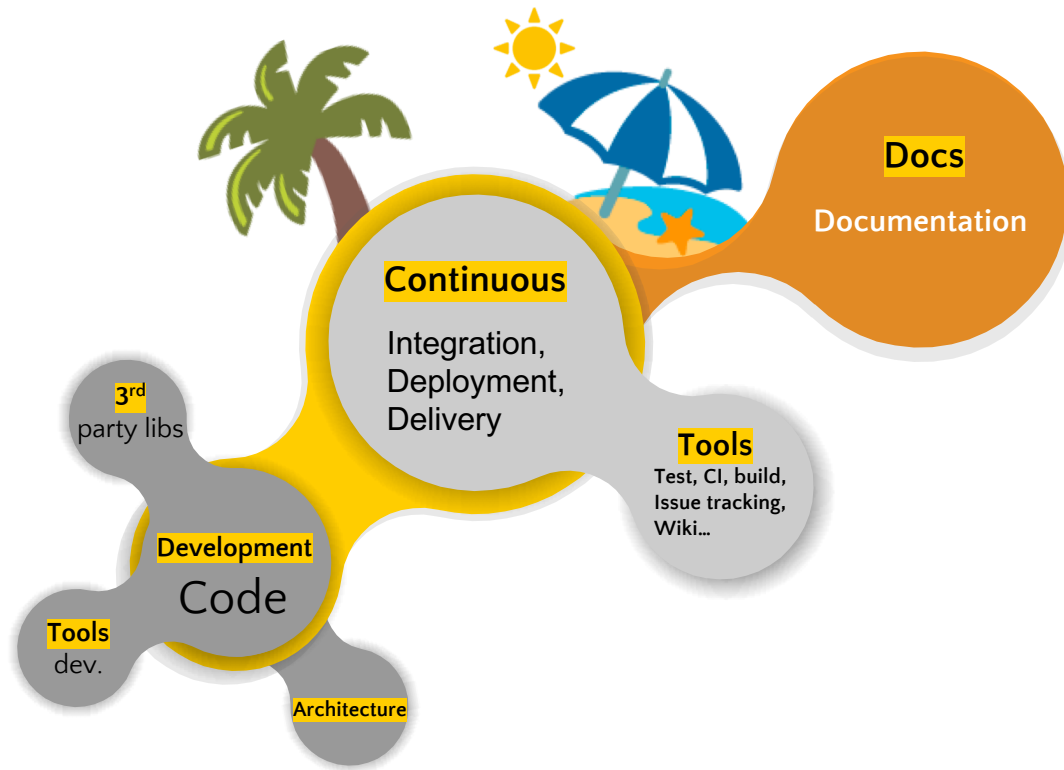
# Continuous ...

...Integration, Deployment, Delivery



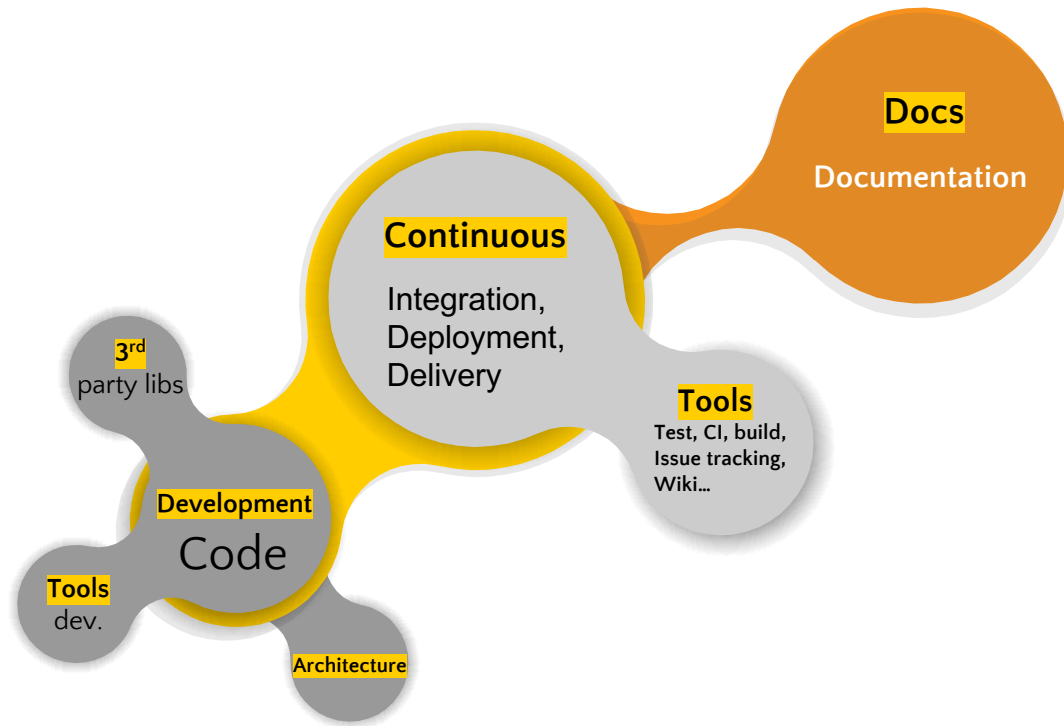


# The **moving** parts

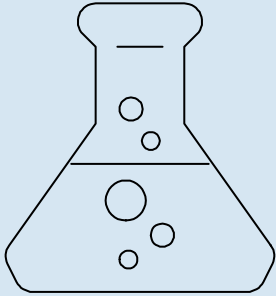




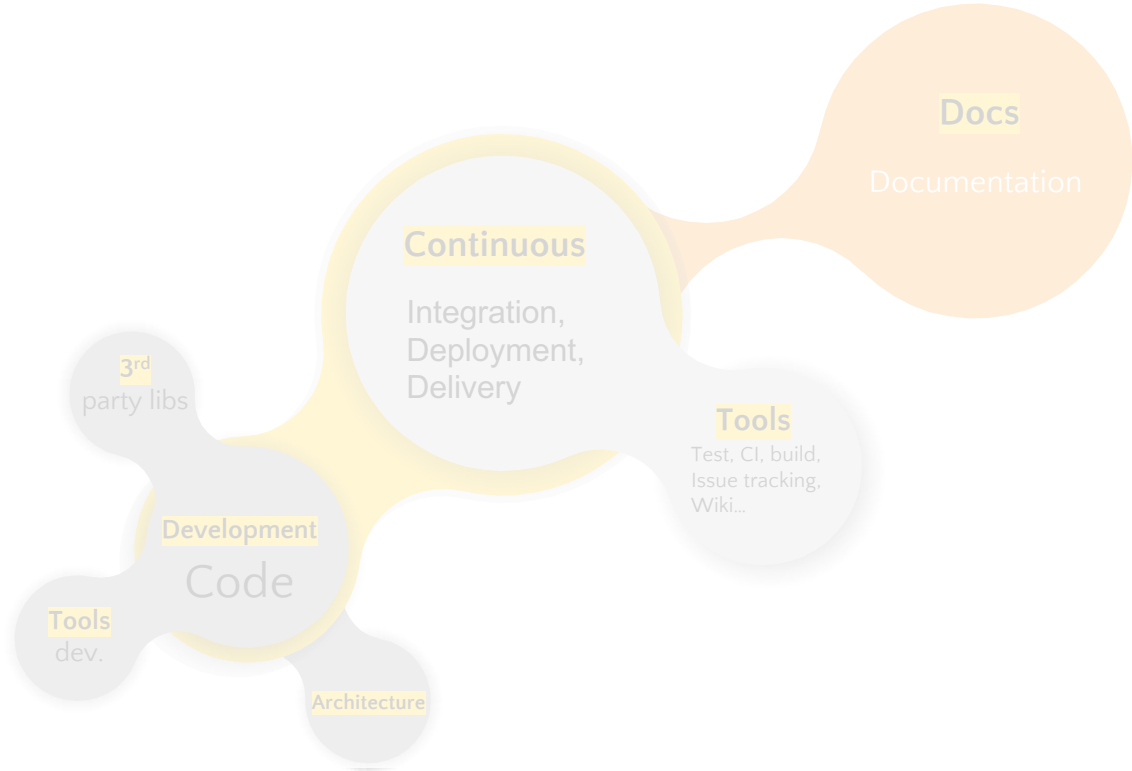
# The **moving** parts



1.



# Development





**Code quality**



Code **quality**

# Code Standard





# Code **quality**

# Encoding



**Encode this**

**Sjåførlærer**

**Blåbærsyltetøy**

**Привет!**

**Skjærgårdsøl**



Code **quality**

# MIME Type



Code **quality**

# Code Review

# Development tools





# Development tools

# Code Versioning



# Development tools

# Branching



# Development tools

**Complexity,  
Testing**



OS Open Source This organization gathers projects that w...

Key: default

- Projects
- Issues
- Quality Profiles
- Rules
- Quality Gates
- Members



XL > 500k 7 |

Languages

Java	1.3k
XML	1.1k
JavaScript	277
Web	260
C#	110
Python	82
C	81
PHP	53
PL/SQL	43
C++	39

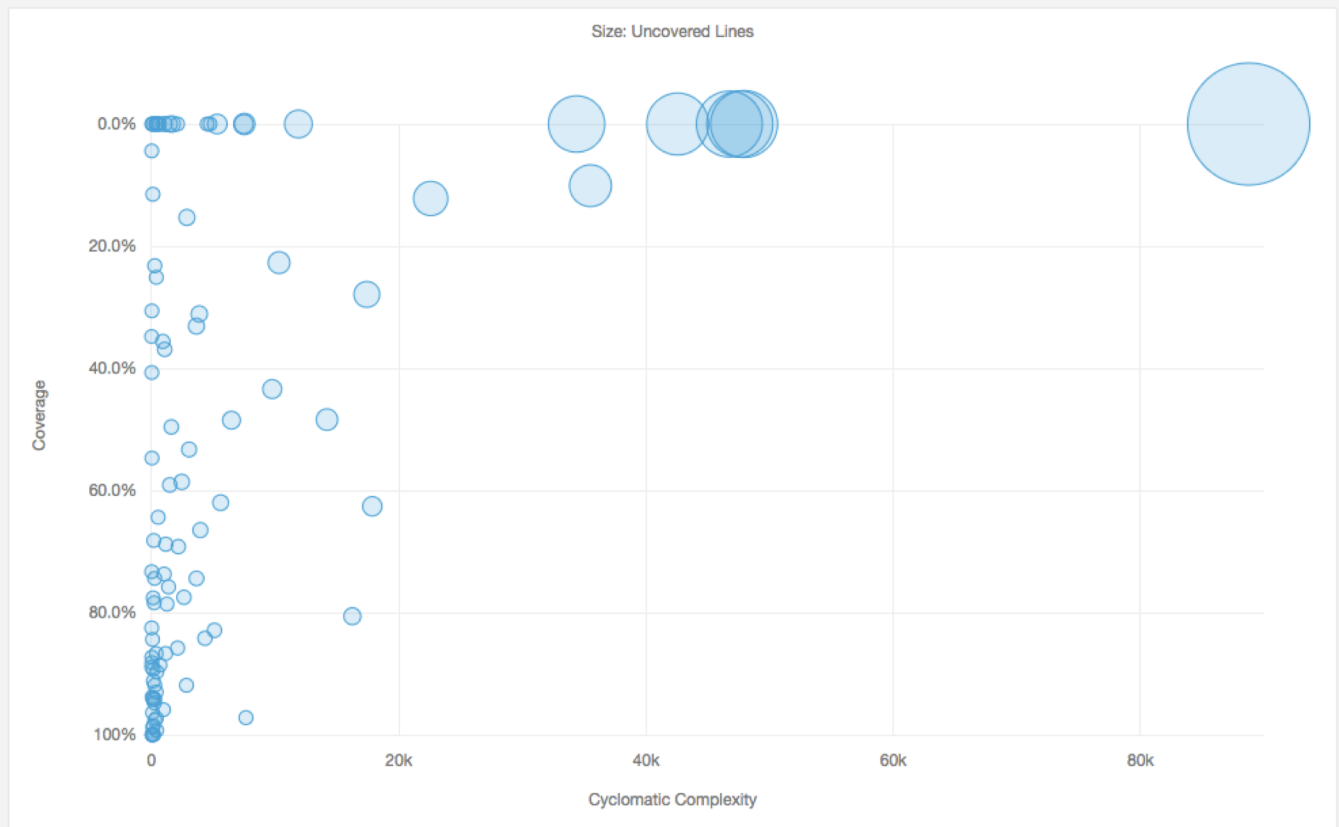
Search

Tags

java	3
java8	2
aws	1
aws-lambda	1
backend	1
coverage	1
eclipse	1
hibernate	1
javascript	1
jax-rs	1

Search

Perspective: Coverage Sort by: Last analysis date Search by project name or key 1251 projects



Quality Gate

**Failed**

<p><b>B</b> Security Rating on New Code is worse than A</p>	<p><b>C</b> Reliability Rating on New Code is worse than A</p>	<p><b>B</b> Maintainability Rating on New Code is worse than A</p>	<p><b>75.8%</b> Coverage on New Code is less than 80.0%</p>
---	--	--	---

Bugs Vulnerabilities

Leak Period: last 30 days

<p><b>185</b> <b>E</b> Bugs</p>	<p><b>180</b> <b>E</b> Vulnerabilities</p>	<p><b>3</b> <b>C</b> New Bugs</p>	<p><b>4</b> <b>B</b> New Vulnerabilities</p>
-------------------------------------	--	---------------------------------------	--

Code Smells

<p><b>77d</b> <b>A</b> Debt <small>started 6 years ago</small></p>	<p><b>2.3k</b> Code Smells</p>	<p><b>11d</b> <b>B</b> New Debt</p>	<p><b>163</b> New Code Smells</p>
--	------------------------------------	---	---------------------------------------

Coverage

<p><b>62.6%</b> Coverage</p>	<p><b>4.9k</b> Unit Tests</p>	<p><b>75.8%</b> Coverage on <b>1.2k</b> New Lines to Cover</p>
----------------------------------	-----------------------------------	--

Duplications

About This Project

PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It supports Java, JavaScript, Salesforce.com Apex, PLSQL, Salesforce.com Visualforce, Apache Velocity, XML, XSL. Additionally it includes CPD, the copy-paste-detector. CPD finds duplicated code in Java, C, C++, C#, Groovy, PHP, Ruby, Fortran, JavaScript, PLSQL, Apache Velocity, Scala, Objective C, Matlab, Python, Go, Swift and Salesforce.com Apex.

No tags

**M** 83k

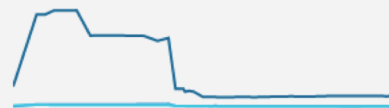
Lines of Code

Java 80k

XML 3.5k

Web 12

Project Activity



March 26, 2018

**6.3.0-SNAPSHOT**

March 26, 2018

Project Analyzed

March 26, 2018

**6.3.0**

## Display Mode

 Issues EffortType Clear Bug 6 Vulnerability 1 Code Smell 72Severity Clear Blocker 6  Minor 90 Critical 15  Info 0 Major 74

## &gt; Resolution

 Bulk Change

to select issues to navigate 2 / 6 issues

PMD Core / src/.../java/net/sourceforge/pmd/cpd/FileReporter.java

 Use try-with-resources or close this "FileOutputStream" in a "finally" clause. ... 2 years ago L49   
  Bug Blocker  Open Not assigned 5min effort Comment cert, cwe, denial-of-service, leak

PMD Core / src/.../java/net/sourceforge/pmd/cpd/SourceCode.java

 Use try-with-resources or close this "BOMInputStream" in a "finally" clause. ... last year L106   
  Bug Blocker  Open Andreas Dangel 5min effort Comment cert, cwe, denial-of-service, leak

PMD Core / src/.../pmd/lang/dfa/report/ReportHTMLPrintVisitor.java

 Use try-with-resources or close this "BufferedWriter" in a "finally" clause. ... last year L47   
  Bug Blocker  Open Andreas Dangel 5min effort Comment cert, cwe, denial-of-service, leak

PMD Java / src/.../sourceforge/pmd/lang/java/rule/AbstractCmBlockAwareRule.java

## Resources should be closed

squid:S2095

## Resources should be closed

 Bug Blocker Main sources cert, cwe, denial-of-service, leak Available Since 05/20/2015 SonarAnalyzer (Java) Constant/issue: 5min

Connections, streams, files, and other classes that implement the `Closeable` interface or its super-interface, `AutoCloseable`, needs to be closed after use. Further, that `close` call must be made in a `finally` block otherwise an exception could keep the call from being made. Preferably, when class implements `AutoCloseable`, resource should be created using "try-with-resources" pattern and will be closed automatically.

Failure to properly close resources will result in a resource leak which could bring first the application and then perhaps the box it's on to their knees.

## Noncompliant Code Example

```
private void readTheFile() throws IOException {
    Path path = Paths.get(this.fileName);
    BufferedReader reader = Files.newBufferedReader(path, this.charset);
    // ...
}
```



## Development **tools**

---

# Static Code Analysis



# Development tools

# Plug-ins

# Libraries by 3rd party





## Libraries by 3rd party

**Known Issues or  
Vulnerabilities?**



## Libraries by 3rd party

# Licenses

- Apache?
- GPL?
- \$\$?





Carlo Morelli

2 days ago · Java, Maven, Mojo, Spring-boot

## Audit licenses in your Java dependencies

Package management on the JDK since the introduction of Maven and Gradle is a goodness that many other dev platforms still can't grasp. I'm thinking of you, pip+virtualenv hacky weirdness :)

Recently as work I was tasked with the generic task of auditing the project code for possible Open Source license violations. At first this raised some eyebrows among us, as one thing that I rarely look at though is what license the library is released under. Also, nowadays we use these very rapid and easy to use frameworks like Spring Boot, where more and more dependencies are pulled without us end-users really controlling them.

Unfortunately, for a library user, a license is VERY important. It is also important for the code that you write, but the license of the dependency could even affect the licensing scheme of your own code. I'll explain better in a moment.

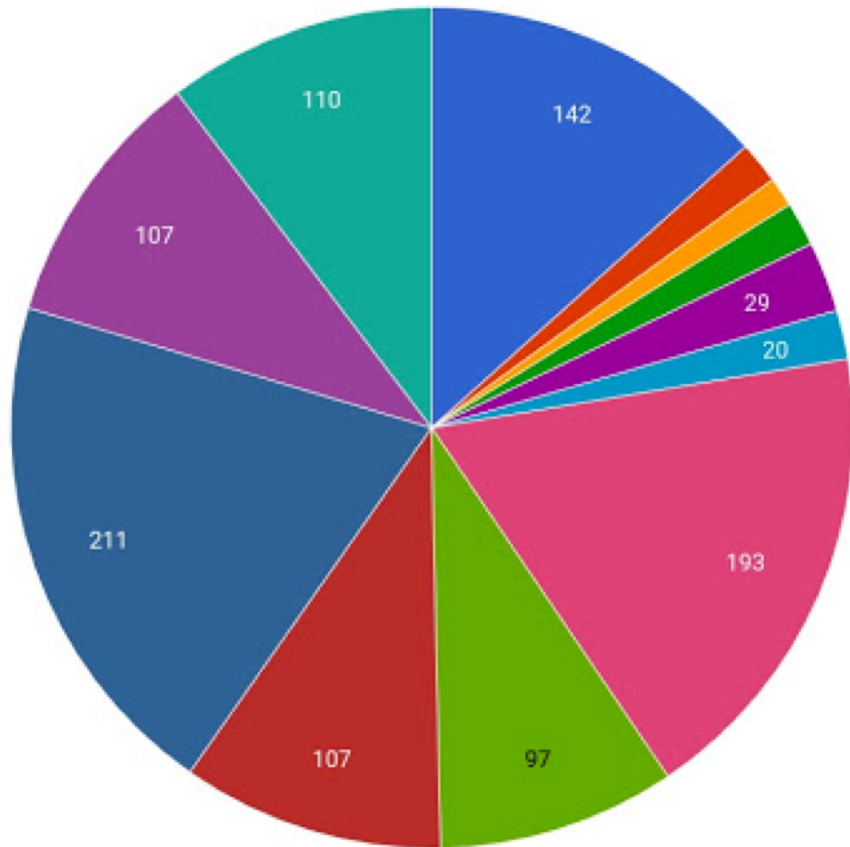
### The license spectrum is wide

I don't want to make a blog post about licenses, I am not an expert. Keeping it simple: beside commercial "Closed Source" code (e.g. the Microsoft Windows code) the Open Source code can have a varying degree of copyleft. There are *permissive licenses*, like **MIT**, **Apache 2.0**, and **BSD**, which allow any fair use of their code (linked or embedded). There are *less permissive licenses* like the **LGPL** and the **GPL with classpath exception** that allow the code to be linked by any other code (so in Java terms, you can have dependencies with those licenses), but not embedded (or, more likely copy-pasted). Finally there are licenses like the **GPL** and the **AGPL** which are considered *strong copyleft* and viral: if your code links a library with GPL code, then your code too should be licensed as GPL. This of course is acceptable if you are developing a open-source project, but it may be not if you are developing a commercial application.



# Libraries by 3rd party

Updated?  
Maintained?  
Compatible?  
...with each other



- heap buffer overflows
- global buffer overflows
- stack buffer overflows
- use after frees
- uninitialized memory
- stack overflows
- timeouts
- ooms
- leaks
- ubsan
- unknown crashes
- other (e.g. assertions)



**Packaging -> Delivery -> Deploy**

**Automated?**  
**Manual?**



**Packaging -> Delivery -> Deploy**

**Deploy to  
Dev, Test, Staging, Prod**



**Packaging -> Delivery -> Deploy**

# Tools

The good, the bad, the ugly...

# Architecture





# Architecture

**Supports continuous deploy?**





# Architecture

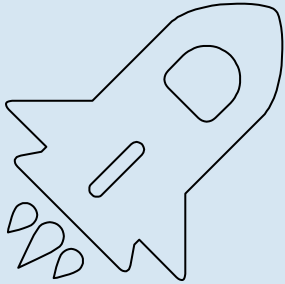
**Application architecture**  
**Integration architecture**



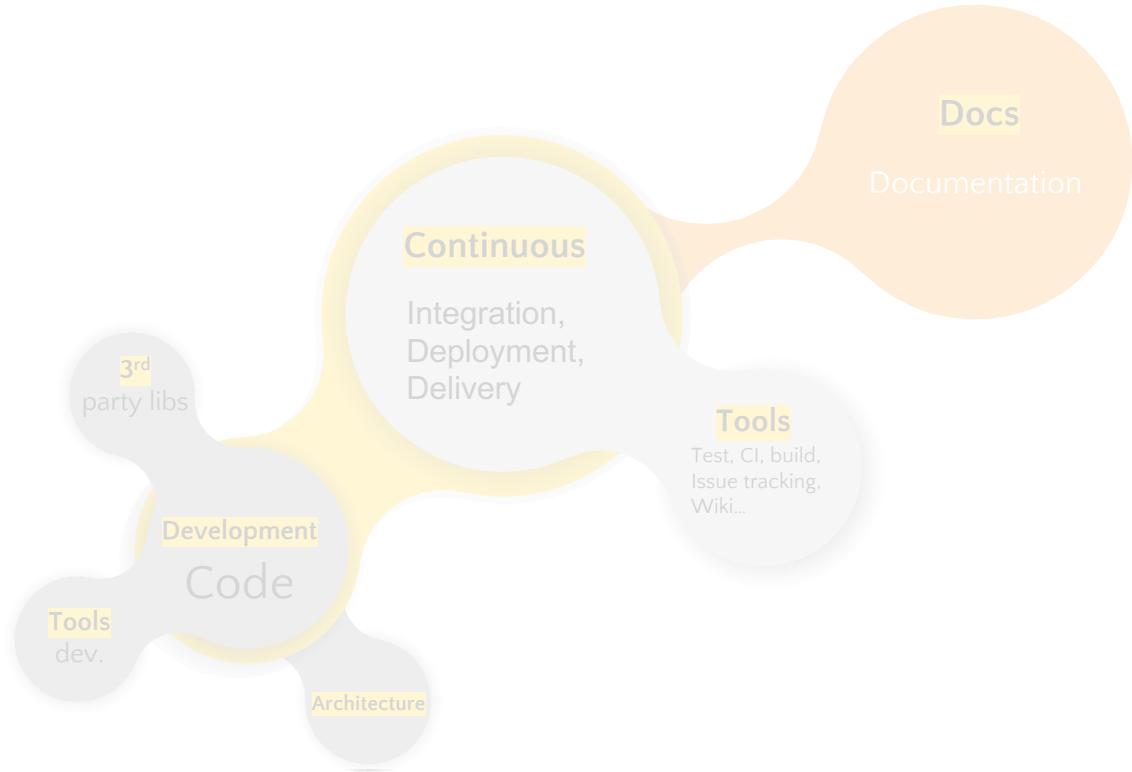
# Architecture

## Code package structure

2.



## Tools



# Environments





# Environments

---

# Same routines for deploy?

Dev – Test – Staging – Production



# Environments

---

**Are the environments similar?**

**Dev – Test – Staging – Production**



# Environments

---

# Re-buildable with a script?

Dev – Test – Staging – Production



# Environments

---

**Same physical hardware?**

**O-oh!**





# Environments

## Monitoring

All of the environments!

<https://wired.com/2016/04/google-ensures-services-almost-never-go/>

# SHARE



SHARE



TWEET



COMMENT



EMAIL

WHEN WAS THE last time you needed to Google something and Google wasn't there?

Odds are, you don't remember that ever happening. Sure, there are times when you can't reach Google because your internet connection is down. But Google's primary online services, from its search engine to Gmail to Google Docs and more, are nearly always accessible. The company's Google Apps suite, including Gmail and Docs, was available about 99.97 percent of the time in 2015, according to the company's own numbers. The world pretty much takes this for granted, but it's a remarkable reality. The billions who use Google hardly stop to consider how Google made something so impressive seem so mundane.

Google explains the feat in three words: Site Reliability Engineering. OK, they aren't the best three words. But that's the rather unsexy name Google gave to this seminal philosophy more than a decade ago. It's a rather nuanced

<https://github.com/danluu/post-mortems>

## Table of Contents

---

[Config Errors](#)

[Hardware/Power Failures](#)

[Conflicts](#)

[Time](#)

[Uncategorized](#)

[Other lists of postmortems](#)

[Analysis](#)

[Contributors](#)

## Config Errors

---

[Cloudflare](#). A bad config (router rule) caused all of their edge routers to crash, taking down all of Cloudflare.

[Etsy](#). Sending multicast traffic without properly configuring switches caused an Etsy global outage.

[Facebook](#). A bad config took down both Facebook and Instagram.



**Tools**



# Development tools

---

IDE



# Development tools

## Integrated with

- SonarQube,
- Unit Tests,
- etc.



# Development tools

---

## Checks at Commits



# Build tools

---

**Automated tests on all levels**





# Build tools

---

Jenkins, TeamCity, Travis, etc.

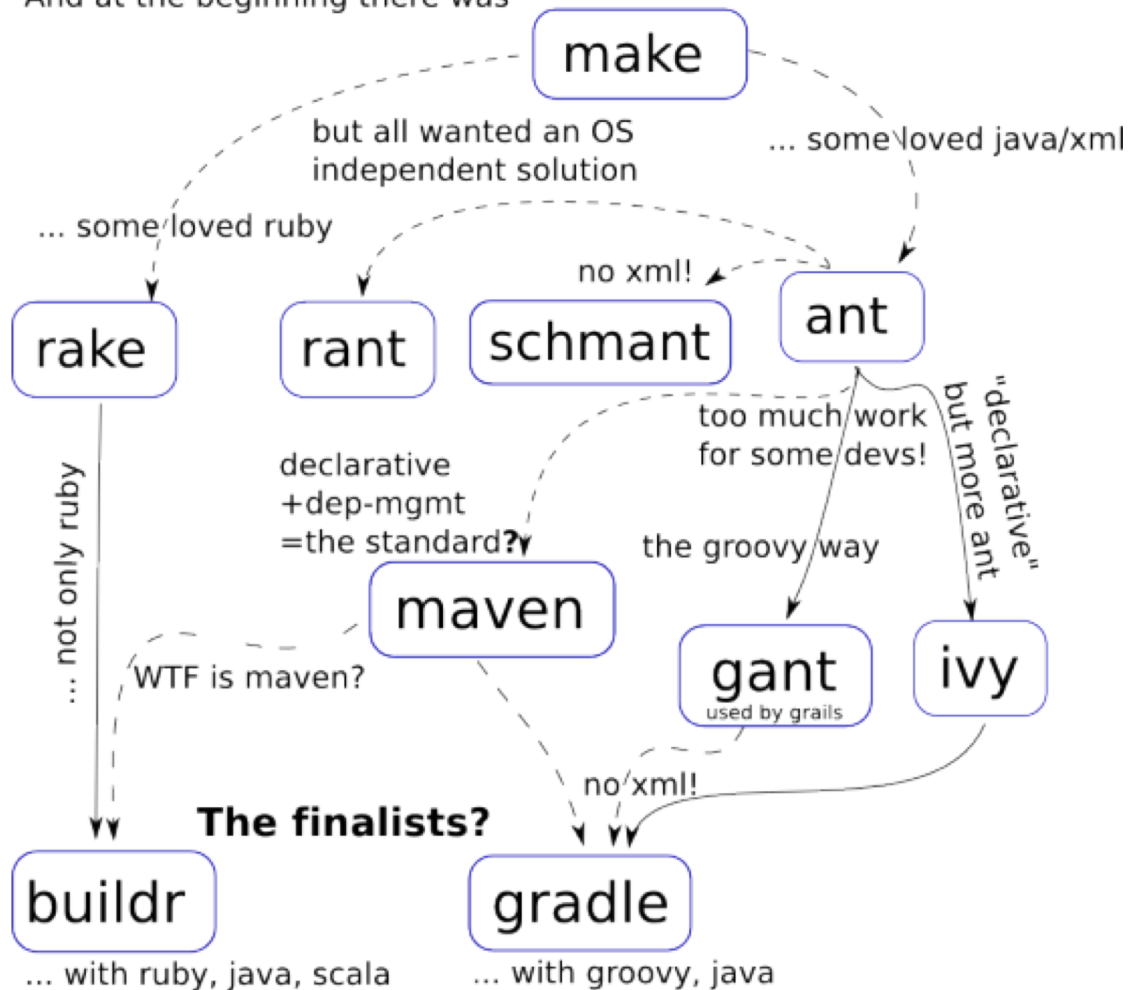


# Build tools

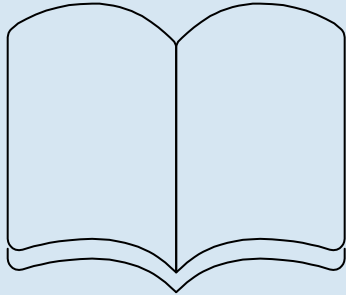
---

**Maven, Gradle, or similar**

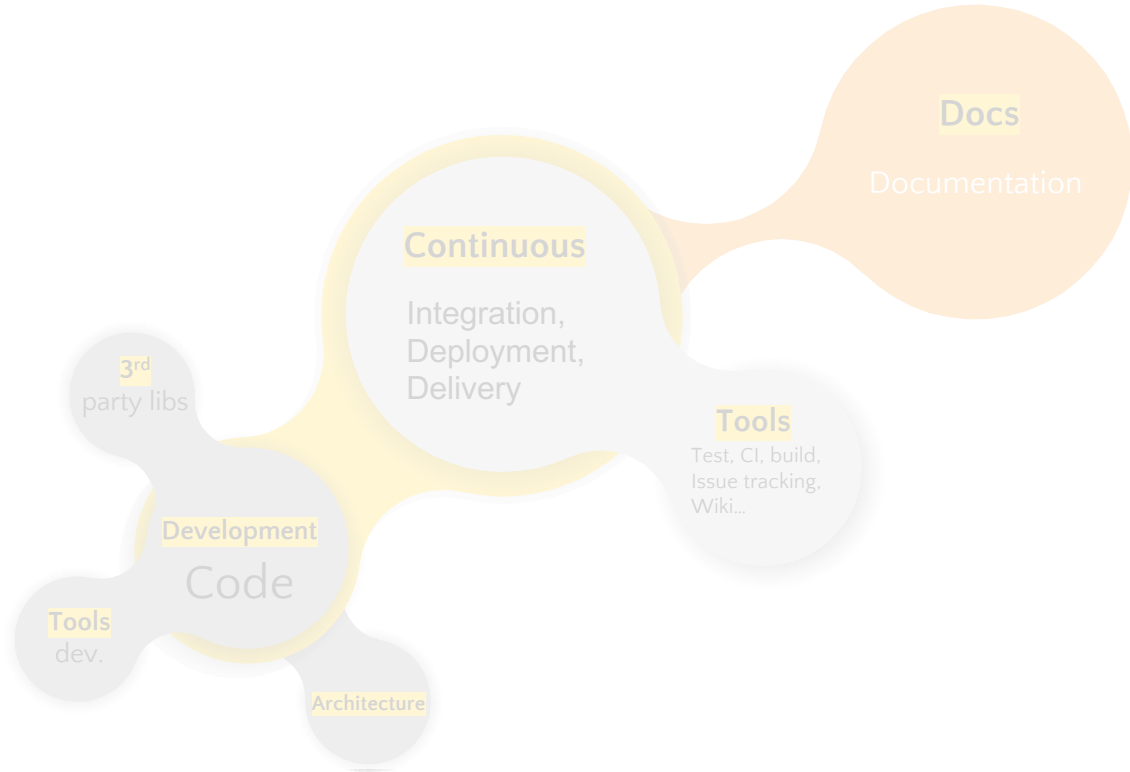
And at the beginning there was



3.

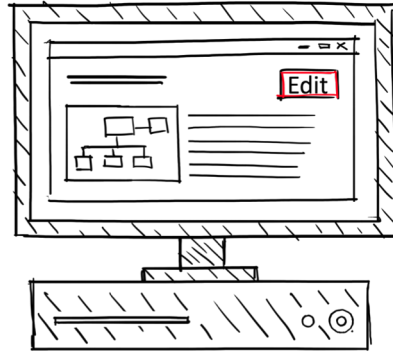


**Docs**





# Tools for collaboration



Wiki,  
Chat, etc.



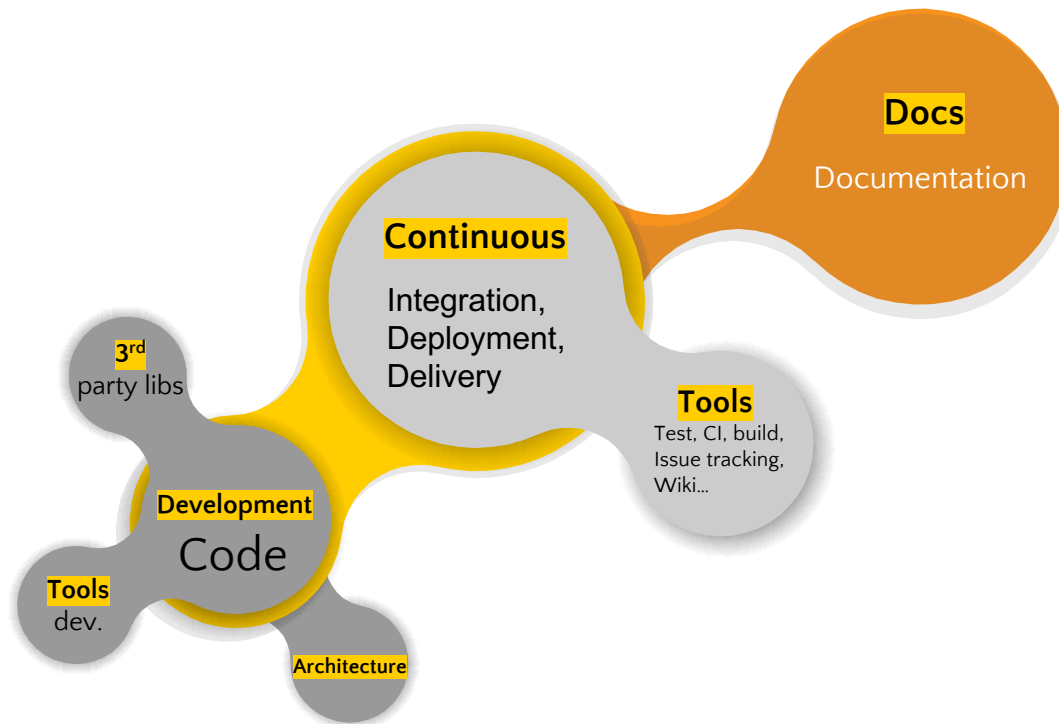
# Tools for collaboration

## Issue tracking





# Remember: The **moving** parts



# Command Line Tools for Your Java Projects

MONDAY, MAY 15, 2017 - 7 MINS

JAVA SOFTWARE DEVELOPMENT ENGLISH

*Getting an overview of your project with some simple command line tools.*

- [Introduction](#)
- [Directory Structure](#)
- [Code Metrics](#)
- [Encoding and MIME types](#)
- [Dependencies](#)
- [SonarQube](#)

## Introduction

This post will give you an overview of some command line tools that will be able to help you to get the feeling on how your project is doing. Most of the tools are widely available in the main Linux distributions and MacOS (some of



## Escaping Developer Nightmares

FRIDAY, DECEMBER 01, 2017 - 10 MINS

[JAVA](#) [FIELD NOTES](#) [SOFTWARE DEVELOPMENT](#) [ENGLISH](#)

*A short write up of the bad things we do in software development and some suggestions on how to fix them.*

- [The Existing State of Affairs](#)
- [The Moving Parts](#)
- [Conclusion](#)

---

Let's take a look into what we can do to achieve a better development environment than an average development project – a project that most of us have seen at some point in our professional lives, or maybe even are a part of right now. We will also look into some tools and patterns that will help us convert those projects into a paradise for the developers.

Just a few decades ago, we were working in ways that might look like unproductive, in the best case. Our development models were predominated





# Thanks!

Any **questions** ?

- ◎ *@RMehmandarov*
- ◎ *rm@computas.com*
- ◎ *<https://mehmandarov.com>*