

# Maven Archetypes

knowledge base and RAD tool

**Peter Rybár**

`pr.rybar@gmail.com`

# Problems

- Slow **project startup**
  - Unclear **technology stack**
  - Low **development speed**
  - Missing **project documentation**
  - Non-uniform **build, release, deploy process**
  - Low **code quality**
- 

No **proto projects** or **skeletons** !

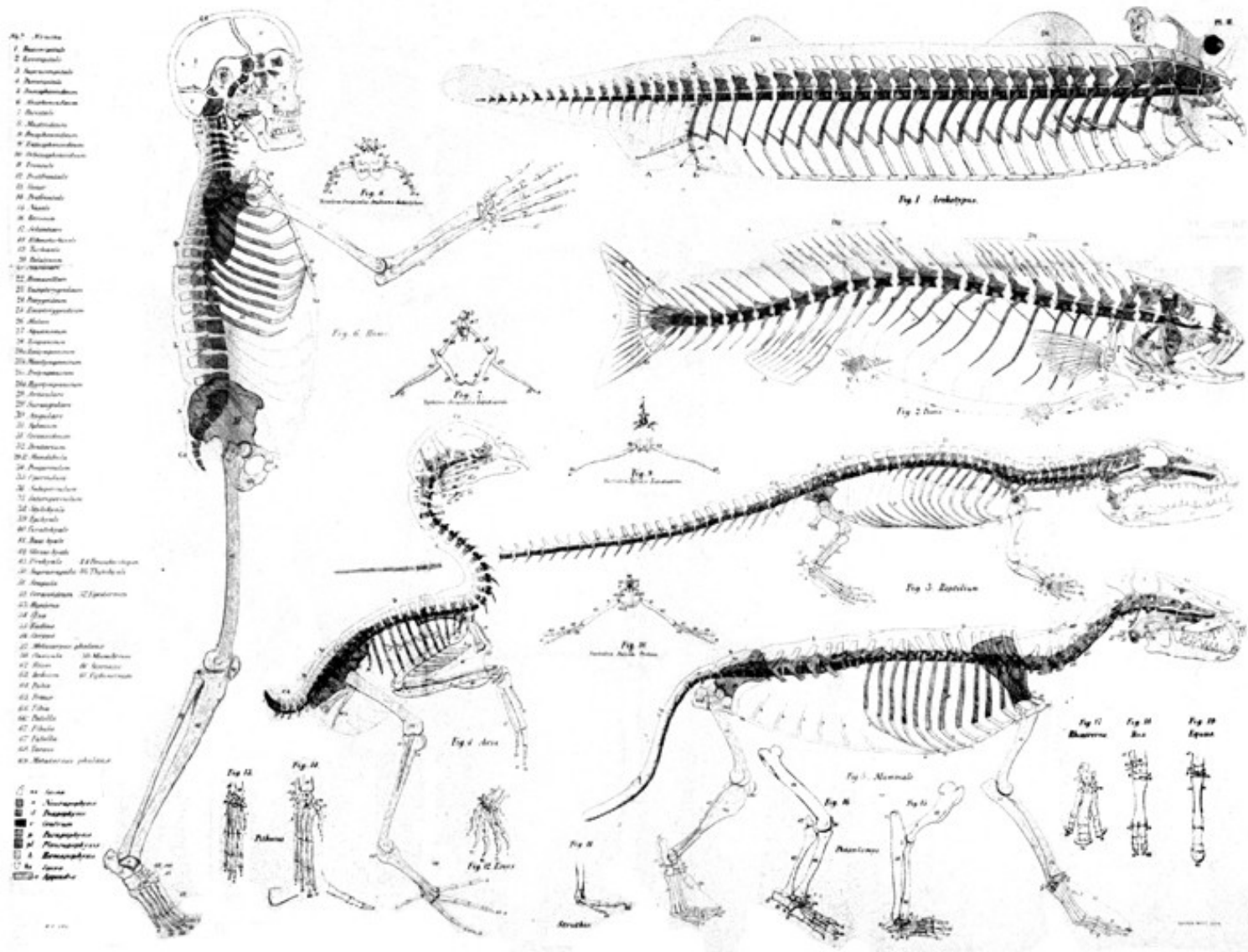
# Build tools and prototyping

- Ant (2000)
  - Granddaddy of Java Build Tools
  - Scripting in XML
  - Very flexible
- Ant+Ivy (2004)
  - Ant but with Dependency Management
- **Maven** (2005)
  - **Archetypes**
  - **Multi-module support**
- Gradle (2008)
  - Attempt to combine Maven structure with Groovy Scripting
  - Easily extensible
  - Immature

# We use Maven

- All build systems are essentially the same:
  - Compile Source code
  - Copy Resource
  - Compile and Run Tests
  - Package Project
  - Deploy Project
  - Cleanup
- **Maven describes the project**
  - No need to script a build, no concept of a conditions
  - **Plugins: tomcat, liquibase, ...**
  - **Archetypes**

# Archetypes



# Archetype

**ar·che·type** /'ärk(i),tīp/

Noun

A very **typical example** of a certain thing.

An **original** that has been imitated.

Synonyms

**prototype**

# Maven archetypes

- **Project** skeletons, **structure** creation
- Dedicated to **rapid project startup**
- Maven modules – **application layers**
- **Tuned technology** stack
- Global project **architecture**
- Cumulate company **best practices**
- Cumulate company **know-how**

# Maven archetypes

## **proto** project

- **Proto(a)** indicating the first or earliest or original
- Project – **set of maven archetypes**
- **Uniform look** for projects !
- How will the projects looks like ?





# proto

- Create project root (Maven reactor)

```
mvn archetype:generate \  
    -DarchetypeGroupId=proto.reactor \  
    -DarchetypeArtifactId=proto-reactor-archetype
```

```
cd <project_dir>
```

# proto

- Create **project modules**
- Every project compose of logical project components like:
  - **Persistent layer** (ORM)
  - **Application logic** (core)
  - **Web user interface** (thin client)
  - **Desktop user interface** (fat client)
  - **Command line interface** (CLI)
  - **Web service interface** (REST, SOAP)

# proto

- Create project modules according to the project requirements.

```
mvn archetype:generate \  
    -DarchetypeArtifactId=<module_artifact> \  
    -DarchetypeGroupId=<module_artifact_group>
```

...

# proto

```
|-- README
|-- INSTALL
|-- LICENSE
|-- pom.xml
|-- docs
|-- data_liquibase
|-- data_orm
|-- core
|-- ui_cli
|-- ui_swing
|-- ui_web_gwt
|-- ws_rest
|-- ws_soap
```

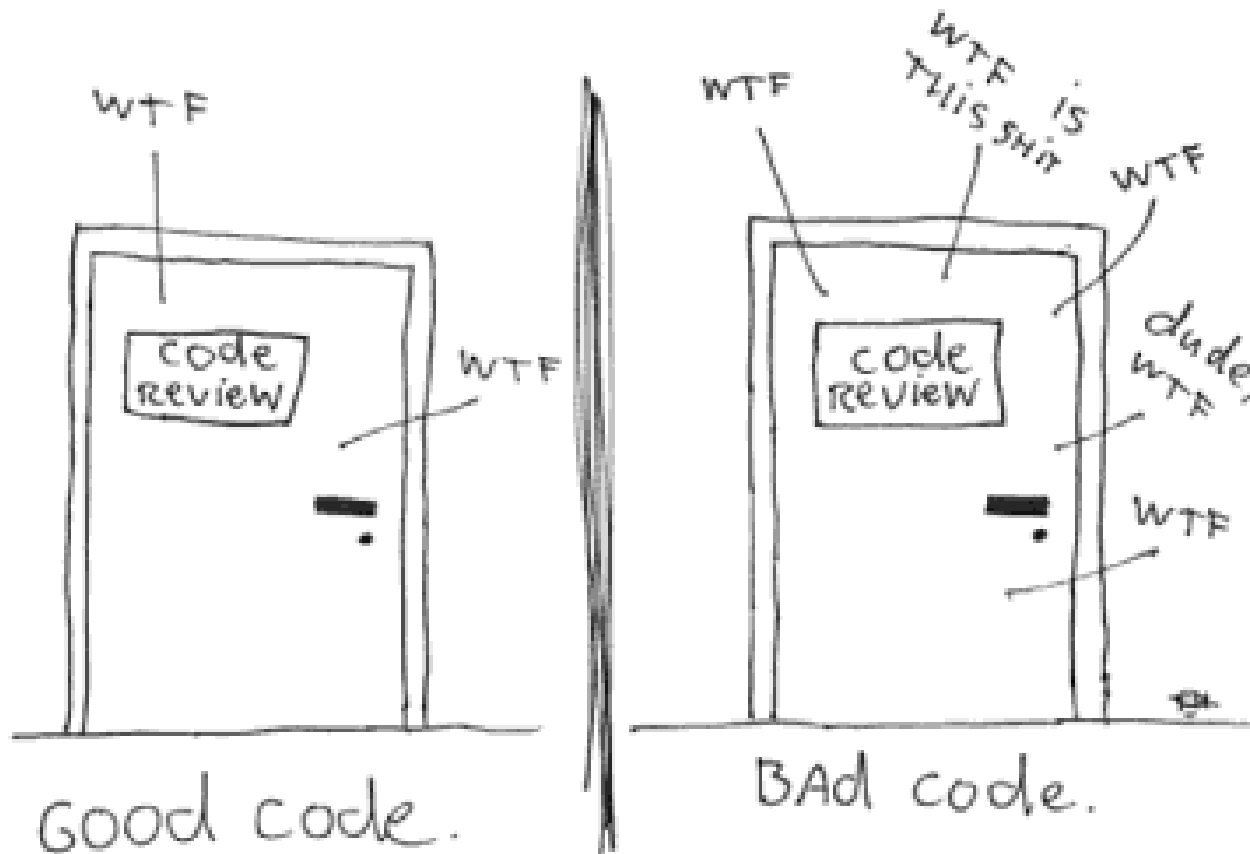
# proto

```
|-- README
|-- INSTALL
|-- LICENSE
|-- pom.xml
|-- docs
|-- src
|  |-- site
|      |-- site.xml
|      |-- xhtml
|          |-- index.xhtml
|          |-- page.xhtml
|-- data_liquibase
|-- data_orm
|-- core
|-- ui_cli
|-- ui_swing
|-- ui_web_gwt
|-- ws_rest
|-- ws_soap
```

# Problems

- Slow **project startup** – **solved**
- Unclear **technology** stack – **solved**
- Low **development speed** – **solved**
- Missing **project documentation** – **solved**
- Non-uniform **build, release, deploy** – **solved**
- Low **code quality** – **???**

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE



(c) 2008 Focus Shift

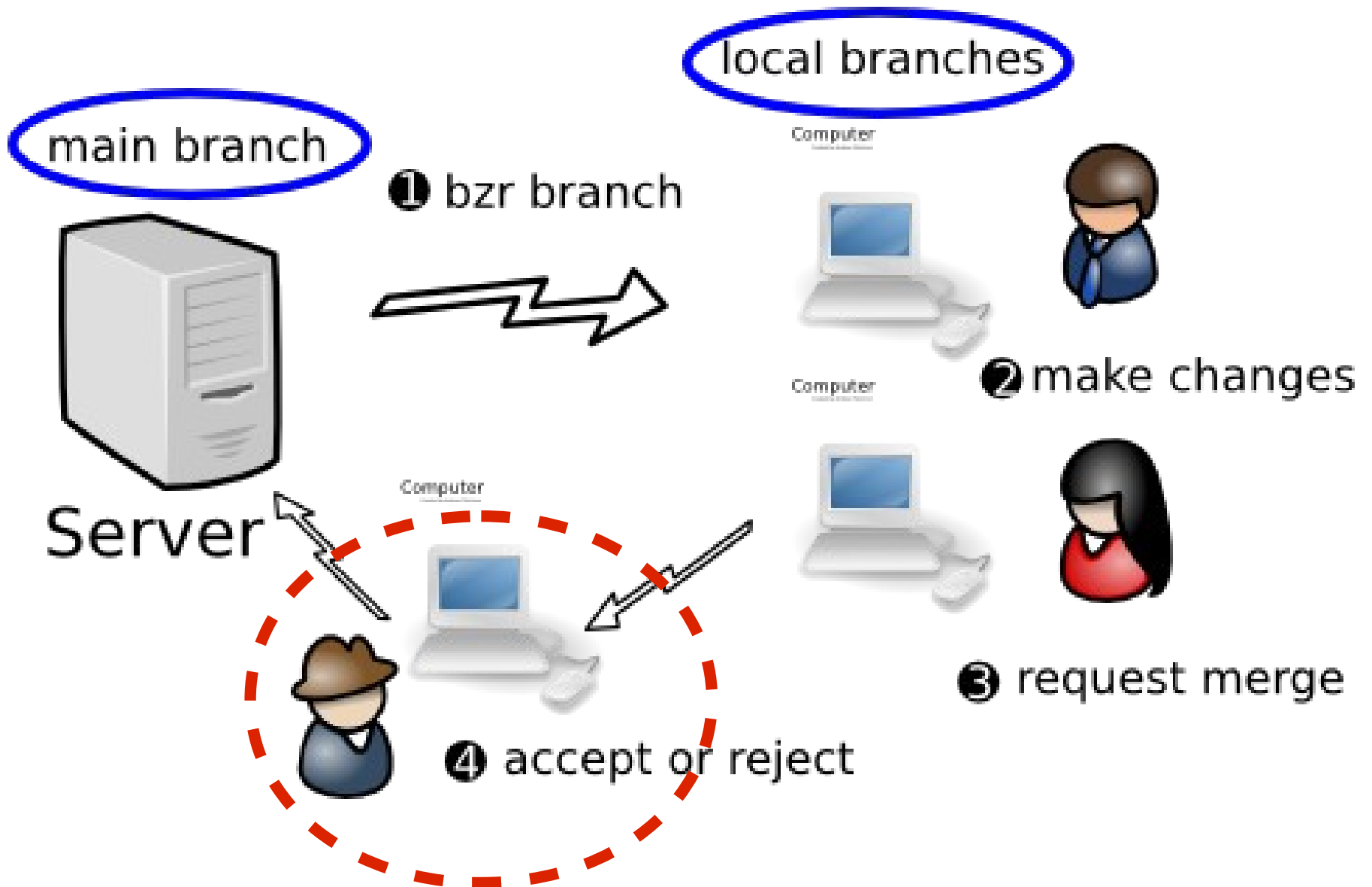
Reproduced with the kind permission of Thom Holwerda.  
[http://www.osnews.com/story/19266/WTFs\\_m](http://www.osnews.com/story/19266/WTFs_m)



# Need Gatekeeper

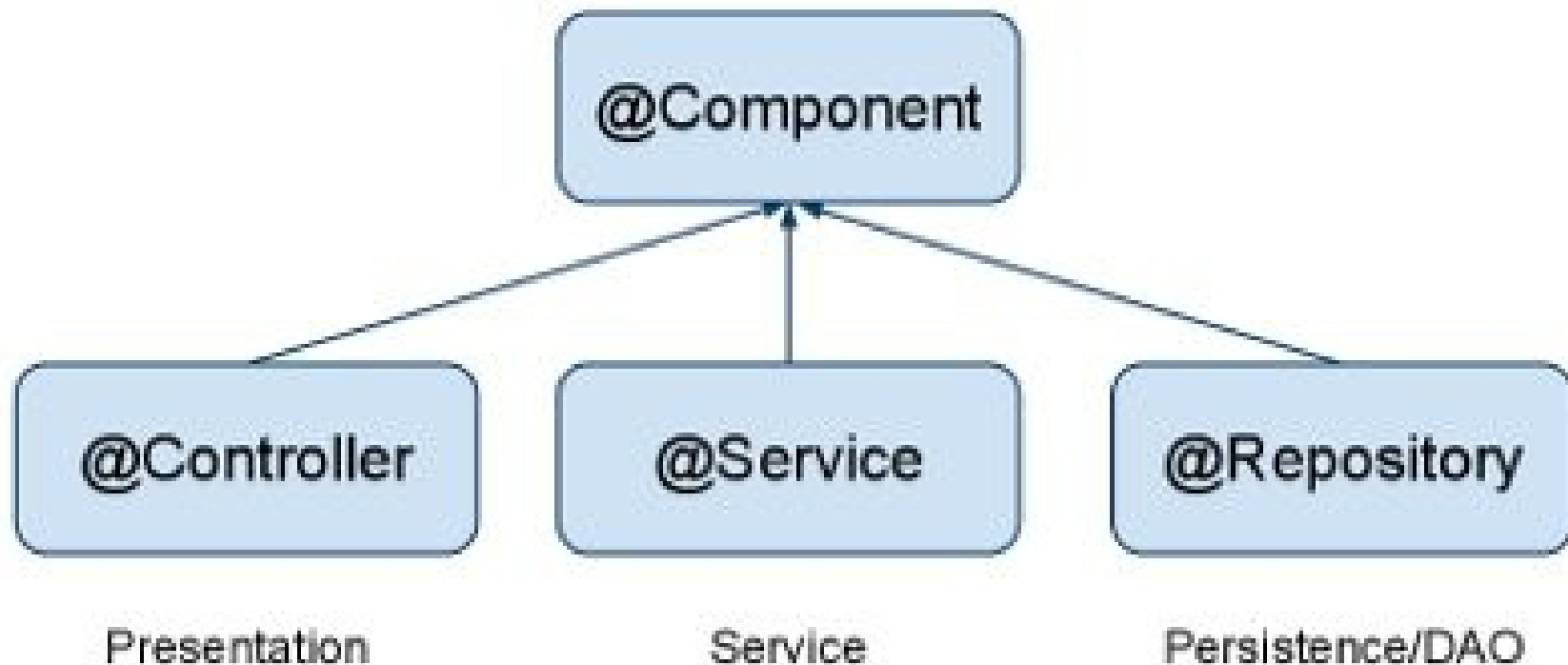


# Human gatekeeper



# Spring

- Spring WS – **pain**, **go REST**
- Spring WS security – **unusable**
- Auto-wiring – **hell**



# Spring > 3.1



<http://cbeams.github.io/spring-3.1-config/#1>



# Need Master



