

# The late life of software

How to ensure projects happily reach retirement

---

**Simone Stefani**

Software Engineer @ Slagkryssaren AB

# Brief history of commercial software

- Software used to be expensive, written by experts and used by experts.
- Through better tooling we decreased cost of building software products.
- Software got cheap, written by everybody and used by everybody.
- Many new businesses can access and use software products.

**What about maintenance?  
And code quality?  
And security?**

We ended up ignoring a large part of the software lifecycle.

# The long-term problem

Software development is a never-ending task. However small businesses often cannot commit to expensive maintenance practices.

Missing and vulnerable dependencies, broken release scripts. Development environment hard to recreate.

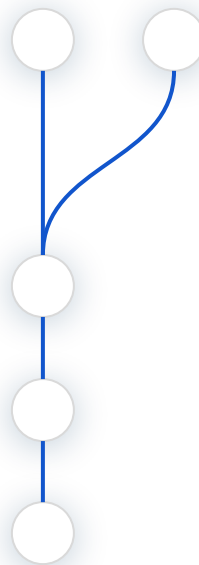
Software ages quickly, becomes legacy in a matter of a few years.

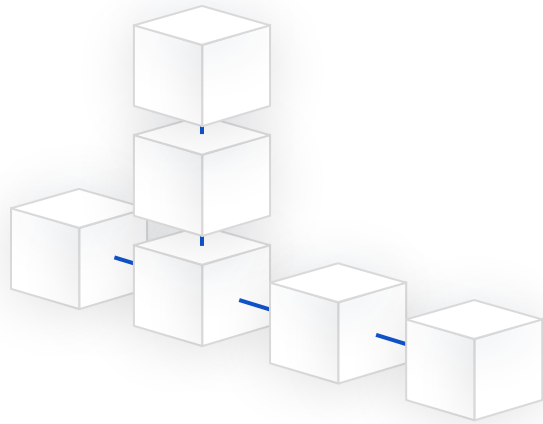
1.

## Project as versioned code

Everything needed to develop and run a project is codified and handled under version control.

Care for a well curated Git tree which clearly expresses the history of the project.





## 2.

# The 1 hour setup rule

A new developer should be able to start working on a project not more than 1 hour after beginning the assignment.

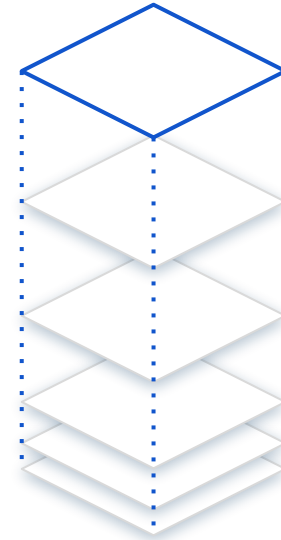
All our projects contain **Docker** configurations which allow us to quickly reproduce a development environment.

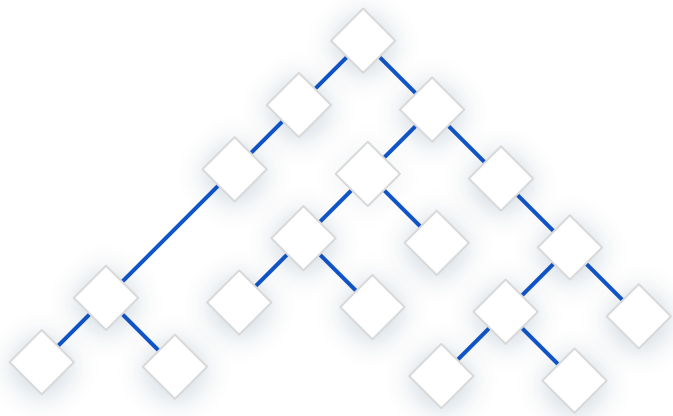
### 3.

## Automate it!

Automation, especially through CD/CI pipelines, allows to scale responsibilities.

Linting, testing, building and deploying are all handled by CD/CI tools. No employee should become the single point of failure of a project.





4.

## Keeping up with dependencies

Automation tools constantly inspect project dependencies and report security vulnerabilities.

We ease the task of keeping dependencies up-to-date with automatic "bump" pull requests.



# A final note

No matter how great the devOps toolbox, **simplicity** and **pragmatism** are key qualities of long living software which continues to be pleasant to work with.