AUTOMATIC VERIFICATION OF ARCHITECTURE REQUIREMENTS

BOSCH

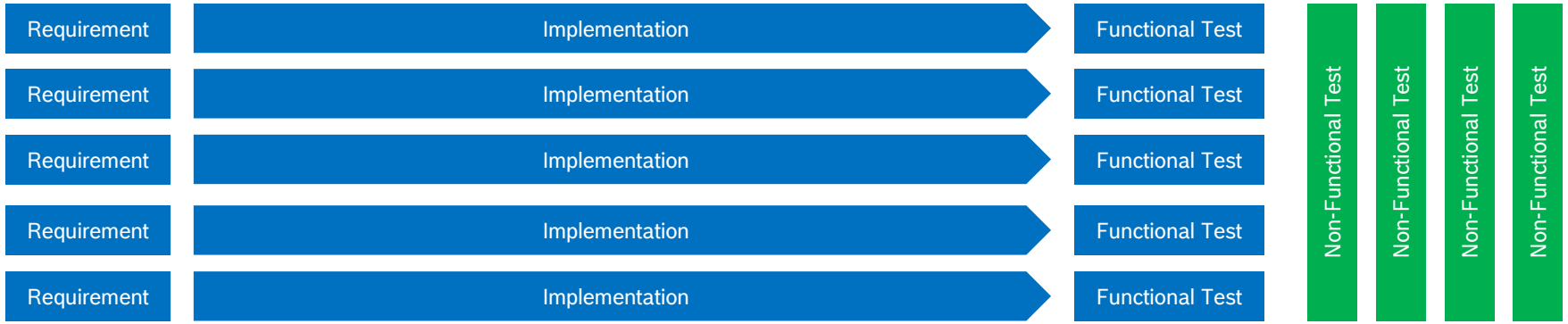
Automatic Verification of Architecture Requirements
Content

- Background
  Paradigm shift towards ”Continuous Software Engineering”
- Problem Statement
  Pile of Guidelines, Trainings, Focus
- Solution
  Automation and Short Feedback-loop

Summary: When shifting to agile, continuous software engineering we automate more. However many non-functional requirements, like architecture requirements are still given to the developers as a pile of documents. These can also be automated with a short feedback-loop.

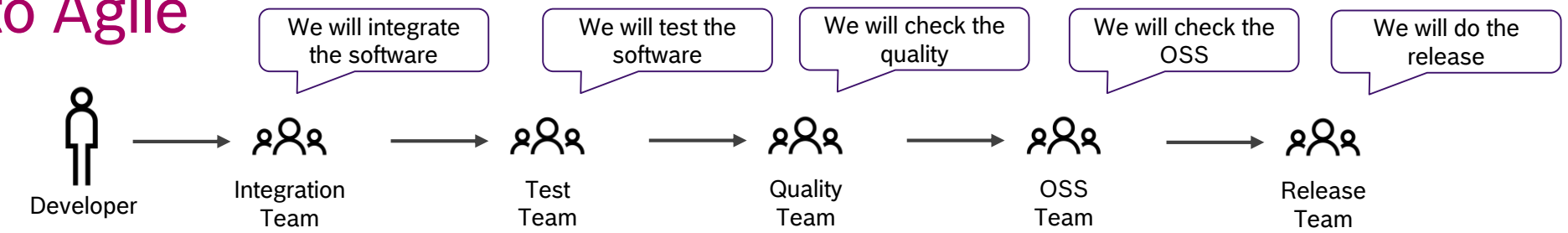
BOSCH

Background
Test of functional and non-functional requirements



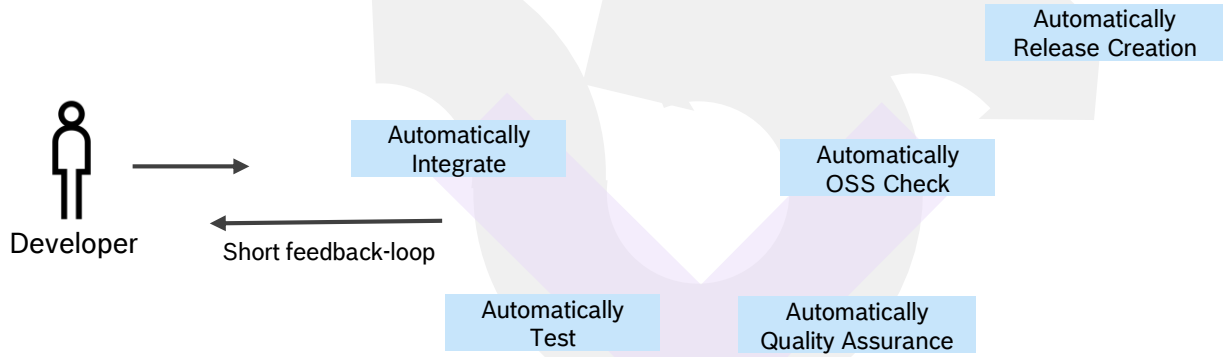
Non-functional tests like performance, security, open source legal compliance, etc. must be performed for every added or changed feature.

Background
Waterfall to Agile



Traditional

Agile



Modern agile software development is running in iterations with automation and short feedback-loops. The developer makes sure the code is integrated, tested and quality assured.

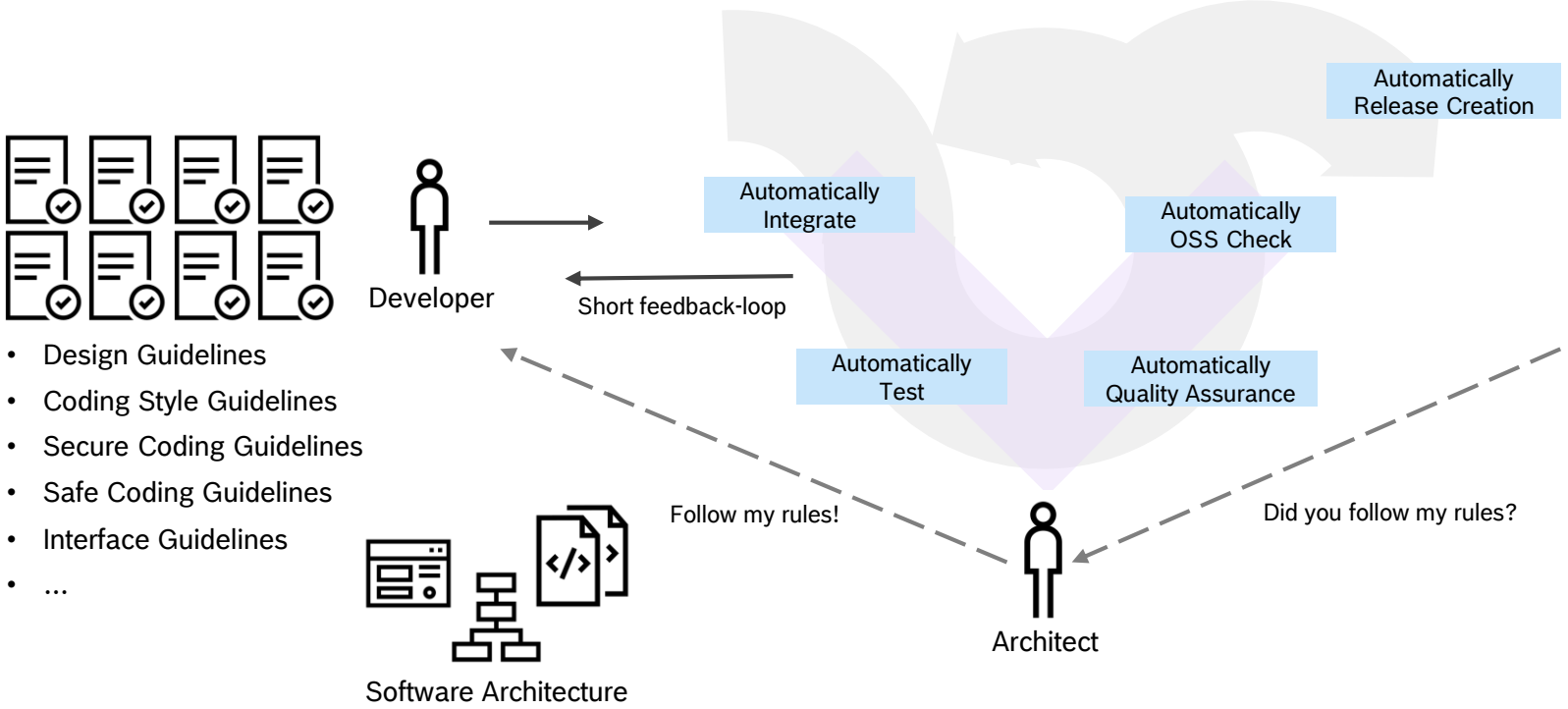
Background
Examples of Automation

- Integration
  - Automatic continuous integration in Git/Gerrit/Jenkins
- Test
  - Automatic verification of unit-test results and code coverage
  - Automatic validation of requirements through system-tests
- Quality Assurance
  - Automatic verification of boot performance
  - Automatic verification of memory budget
- OSS
  - Automatic scan of code in Black Duck®

Several integration, test and quality assurance activities are automated. This makes it possible to have fast short feedback-loops to the developer.

Problem Statement

But what about Architecture Rules and Guidelines?



- Design Guidelines
- Coding Style Guidelines
- Secure Coding Guidelines
- Safe Coding Guidelines
- Interface Guidelines
- ...

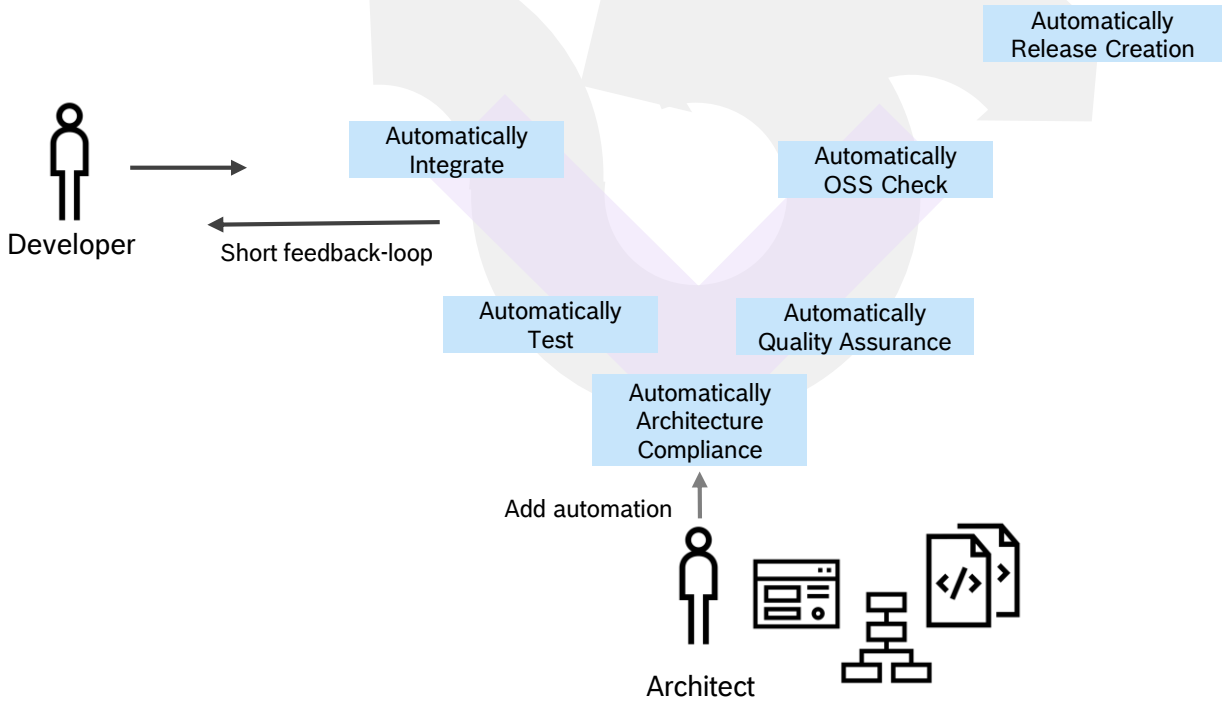
The architecture rules, models and guidelines are usually not checked automatically. Instead a large pile of guideline documents and architecture diagrams are given to the developers.

Problem Statement
Pile of guidelines and architecture documents

- It is hard for a developer to keep all details in their head all the time
- When one area is facing issues (like security) other areas tend to be missed
- The architect writing the guidelines spend much time presenting and train the developers
- Model/Code GAP becomes large
- It might be a long time until feedback is given

This causes several problems and extra effort both for the developers and the architects.

Solution
Automate the Validation of Guidelines and Models



Use the toolchain to create a feedback loop for the developers. The feedback loop will increase developer knowledge, prevent mistakes and automatically govern the changes.

THANK YOU

Robert Lagerstedt

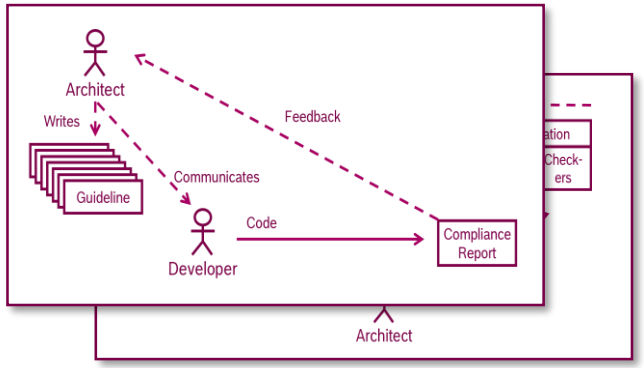
robert.lagerstedt@se.bosch.com

https://www.linkedin.com/in/robertlagerstedt/

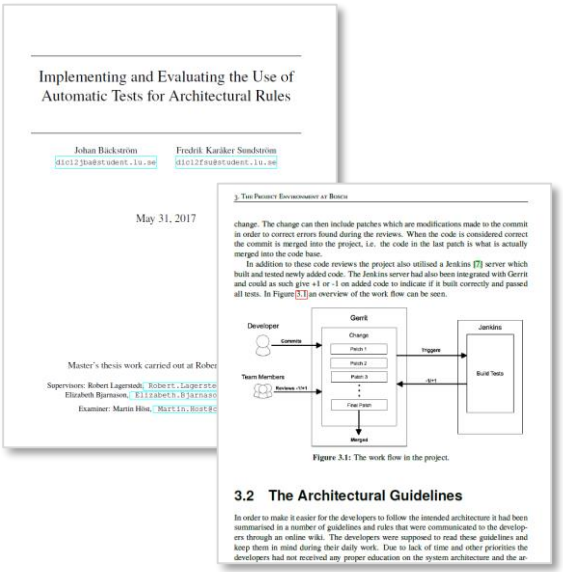
BOSCH

Backup
More detailed information

"Using automated tests for communicating and verifying non-functional requirements"
2014 IEEE 1st International Workshop on Requirements Engineering and Testing (RET) - R. Lagerstedt
https://ieeexplore.ieee.org/document/6908675



"Implementing and Evaluating the Use of Automatic Tests for Architectural Rules"
Master Thesis LTH/Bosch – J. Bäckström, F. Karåker Sundström


Implementing and Evaluating the Use of Automatic Tests for Architectural Rules

Johan Bäckström
Fredrik Karåker Sundström

May 31, 2017

Figure 3.1: The work flow in the project.

3.2 The Architectural Guidelines

Software Center – Continuous Architecture Theme