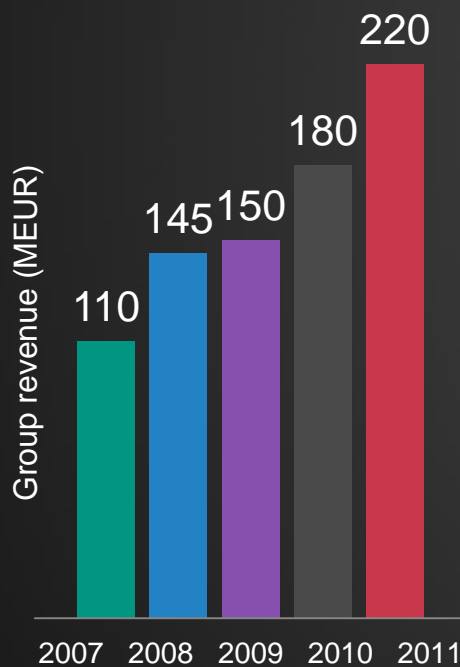

Continuous Delivery and Test Automation in Agile SW projects with Robot Framework

7.6.2013 *Antti Pohjonen*

Trusted Partner in Nordics

Knowit AB is listed on the Nordic Exchange in Stockholm

1800 Experts in
*Sweden, Finland, Norway,
Denmark, Russia and
Estonia*



We develop our customers **competitiveness**

Shorter lead times and lower costs in product development through quality management and flexible resourcing

Test &
Development

Better decisions based on refined business information.

Decision

Improved sales, collaboration and communication with advanced web solutions.

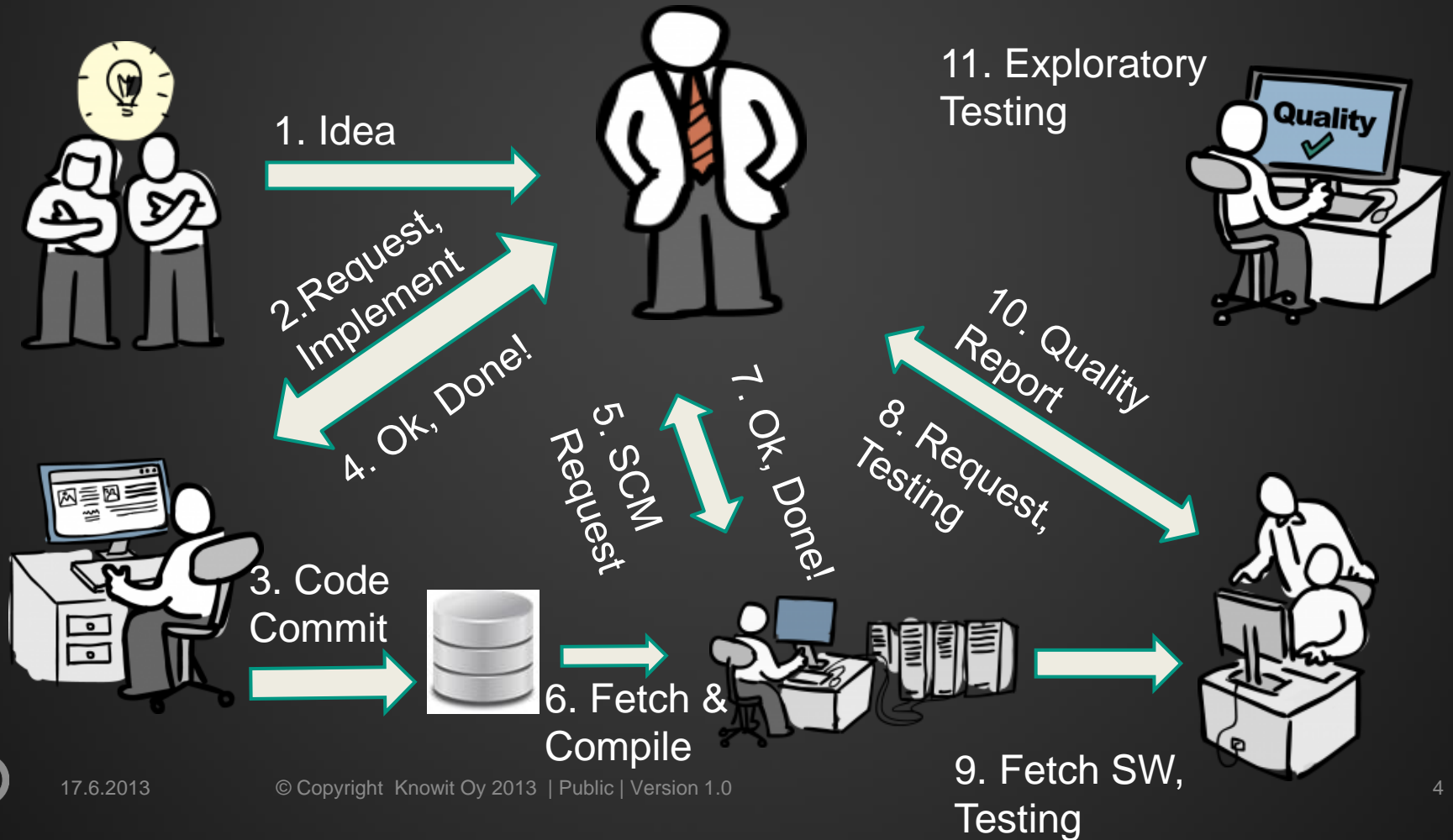
Web

More productivity in information-intensive work with efficient digital tools

Collaboration



The Traditional Way (Simplified!)



How this could be done in a better way?



Overview

- Continuous Delivery
- Test Automation
- Robot Framework
- Case I: A large project from scratch
- Case II: Continuous Improvement
- How to start & build your own Test Automation?



Continuous Delivery

Based on Continuous Integration (CI)

"Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day." - Martin Fowler

Continuous Delivery just adds Test Automation to it

A Pipeline concept, from bottom to upper levels and layers

Focus on speed and automation!!!

Small difference, but significant, to Continuous Deployment



Continuous Delivery



Test Automation

More than just Automated Tests

Fully automated flow (pipeline)

DevOps (Development and Operations) tasks

- Own server and test environment administration
- Means involvement in to company politics

Requires CI system and some kind of test automation framework / tool



Test Automation

Test automation contains steps like:

- Continuous Integration, Commissioning, Integration tests, Functional / System Tests, (User) Acceptance Tests
- And is triggered by external event: scheduled / commit etc.

Test automation developers should have a wide and good understanding over used SW development process

Focus on Building Right It vs. Building It Right.

Verification what is done vs. finding bugs

Test Automation development is Continuous activity



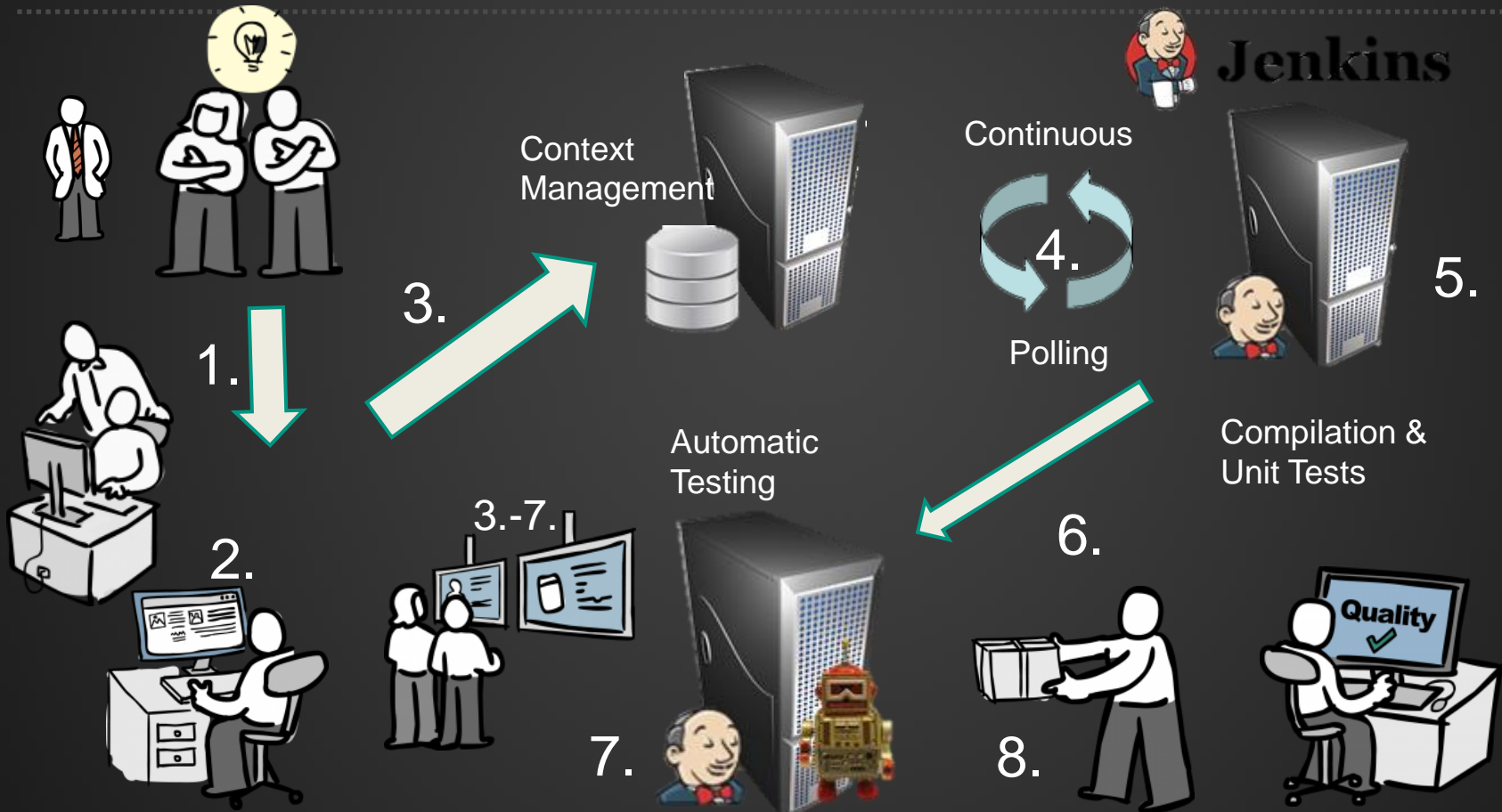
Robot Framework

Generic Test Automation framework

- Many internal & external “ready made” test libraries
- Uses keyword-driven testing approach
- Can be used in vast various ways
 - ATDD, BDD, Data-Driven Tests, Specification by Example etc.
- Implemented with Python
 - Can be extended natively using Python or Java
- Easy integration to other systems
- Open source
 - Sponsored by Nokia Siemens Networks
 - Active development and growing community



Continuous Delivery & Test Automation



Case I: A Large Project from Scratch

Case I: A Large Project from Scratch

Started as automated tests project with Robot Framework as replacement for old testing tool.

- Goal was to make automated tests

Turned to Test Automation project by interest to execute tests automatically (scheduled) in unbiased environment.

- Opportunity and curiosity, free resources
- Gained interest of management and other stakeholders
 - Increased visibility and velocity



Case I: A Large Project from Scratch

Grew larger and more significant with time, more features were injected and requested by management and users.

- Spread around development teams & sites.
- Became de facto, added to Way of Working

Next focus on Continuous Delivery, added Continuous Integration as beginning of pipeline

After implementing CD, we had huge amount of testing data

- Focus moved on faster feedback and reporting



Case I: A Large Project from Scratch

Old product / project had:

- 2000-4000 semi automatic test cases
- Took around 2 weeks to execute
- High dependency to test environments

New product / project:

- over 3500 fully automated test case
- Took around 12 hours to execute
- Test environment independent by design
- Focus moved to reporting
 - Continuous Test Automation development



Case I: A Large Project from Scratch



Jenkins



Case II: Continuous Improvement



Case II: Continuous Improvement

I participated to the project, when the second rewrite and enhancement round of Robot Framework test libraries was starting.

Test Automation was handled by development teams as side activity, taking more time than they dare to spent on it.

Need for a new test automation core team was apparent

New team's role was act as supportive team for SW development teams



Case II: Continuous Improvement

New team's main goals were:

- Take over DevOps tasks
- Maintenance and enhancement of test libraries
- Enhance the Test Automation pipeline concept.
- Make Test Automation solution scalable and productize it.
- Act as "Community of Practise" for different projects



Case II: Continuous Improvement

Main Achievements:

- Developers can focus on their main tasks
- Better visibility and reporting
- Test Automation is spread to multiple new software projects
- Tool and technology changes are made easier
- Trust and "dependency" to Test Automation is increased



How to start & build your own Test Automation?

Little by little...

Aim high, but start from basics and easy tasks

Have fun and make little experiments first!!

Fail often and Fail fast!

Select best and the most suitable tools for your project, evaluate those before making final decision!

For Example:

Use Jenkins as CI system tool and Robot Framework as test automation framework

Learn from others', don't repeat same mistakes again...



Thanks!!

Questions?

 @AnttiPohjonen

LinkedIn: fi.linkedin.com/in/anttipohjonen/

