Continuous Integration: Aspects in Automation and Configuration Management

Christian Rehn

TU Kaiserslautern

January 9, 2012

Overview

Questions

- How to do integration in an agile setting?
- Why is “traditional” integration infeasible for agile development?
- What aspects have to be considered...
  - ... w. r. t. automation?
  - ... w. r. t. configuration management?

Context

Agile Software Development

- Agile Software Development means less documentation
  - Especially no “big upfront design”
- Agile practices needed to compensate lack of planning
- Some practices also valuable in plan-driven projects

XP Practices

XP Practices

Continuous Integration – Overview

Continuous Integration is. . .

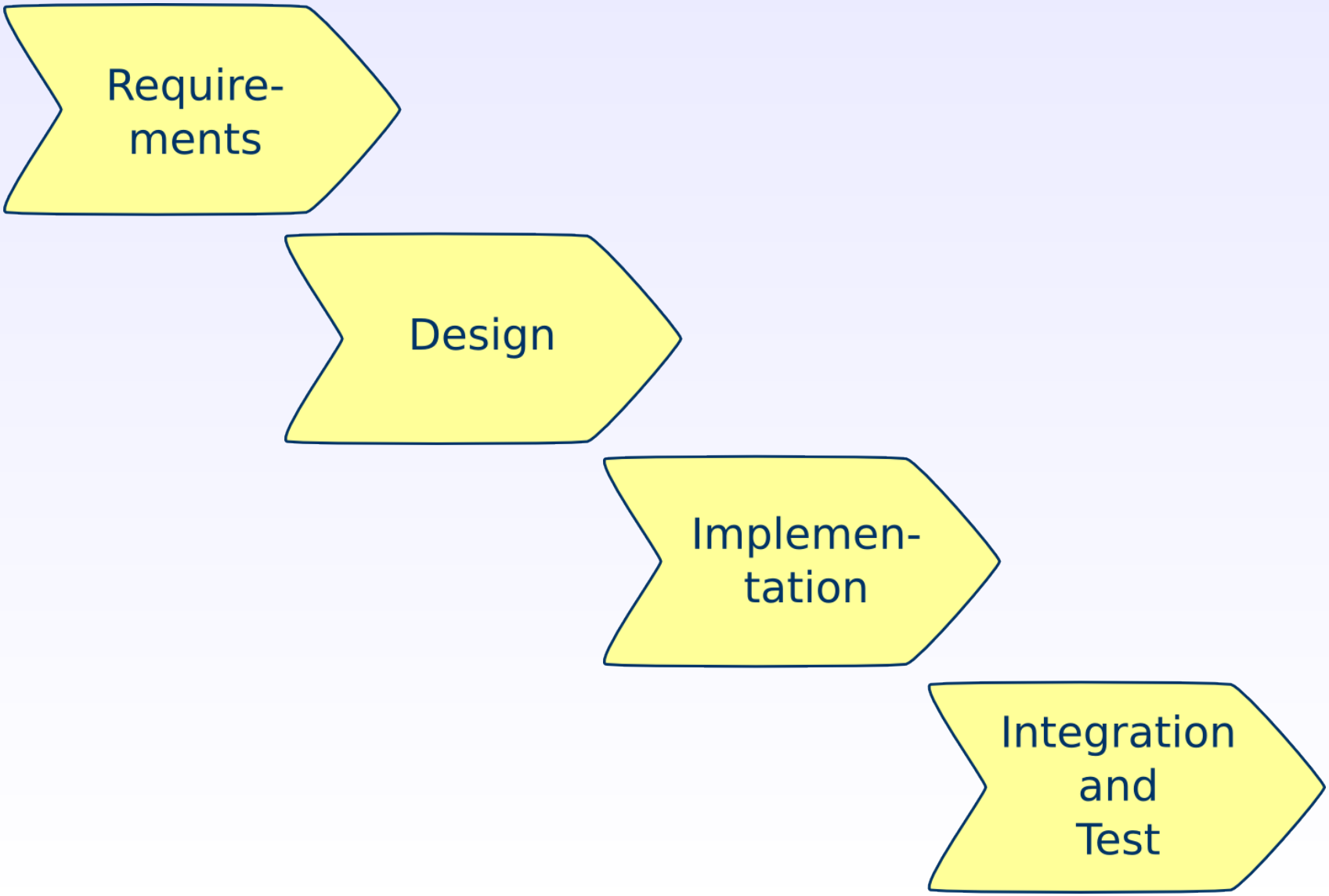
- . . . an agile integration technique
- . . . about integrating every day
- . . . tool supported (everything is automated)
- . . . widely adopted in practice
  - In agile and plan-driven projects

Basics of Continuous Integration

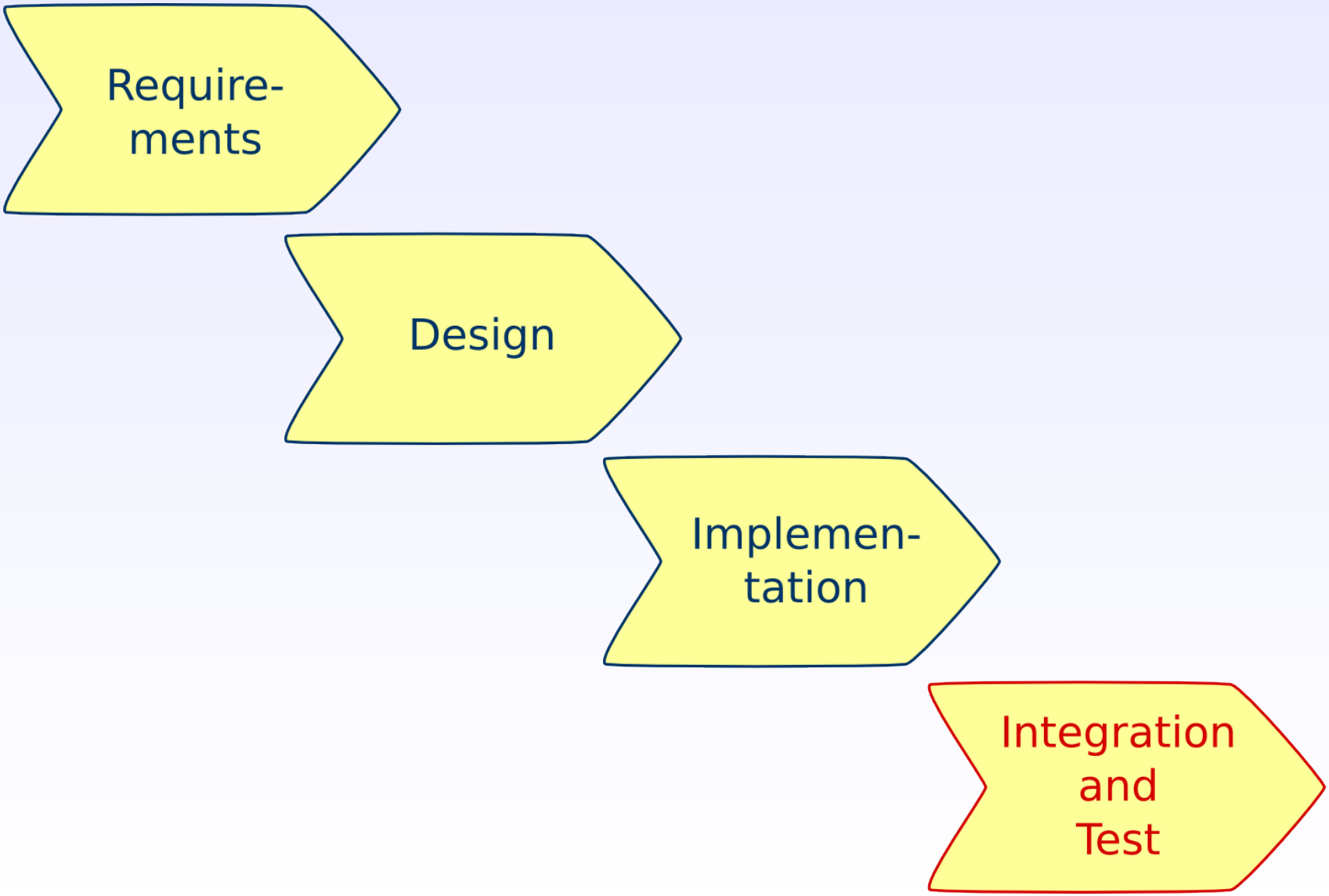
Question

Why is “traditional” integration infeasible for agile development?

Waterfall


Require-
ments
Design
Implemen-
tation
Integration
and
Test

Waterfall


Require-
ments
Design
Implemen-
tation
Integration
and
Test

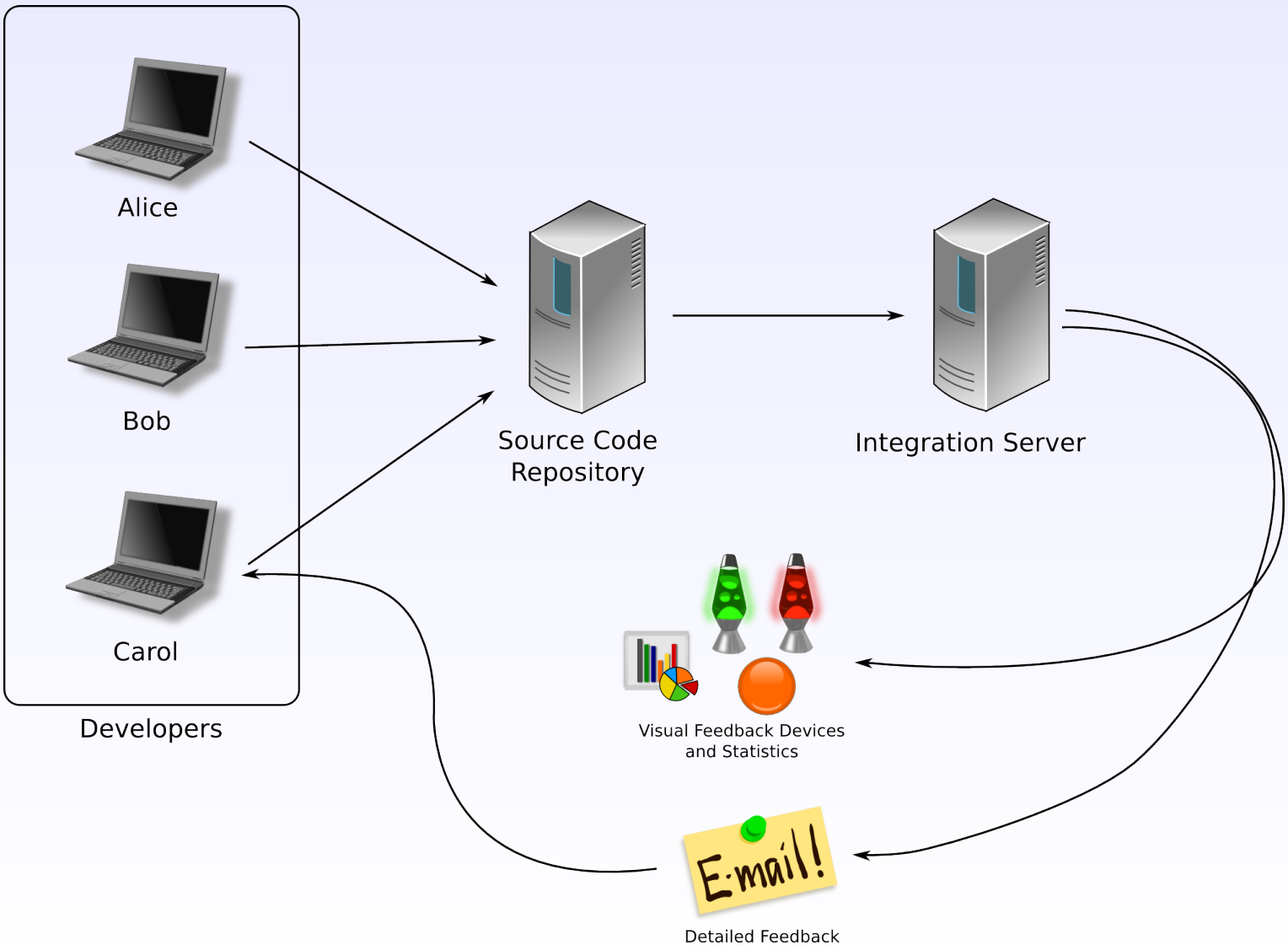
Problems with the Integration Phase

- Certain problems are detected late
  - Misunderstandings of technology
  - Independently developed components not fitting together
  - Wrong assumptions
  - Misunderstandings between developers
  - ...
- In waterfall-like processes mitigated by detailed upfront design

Question

How to do integration in an agile setting?

Continuous Integration



Benefits

- Better communication among developers
- Fast feedback on integration problems
- Defects found earlier and fixed faster

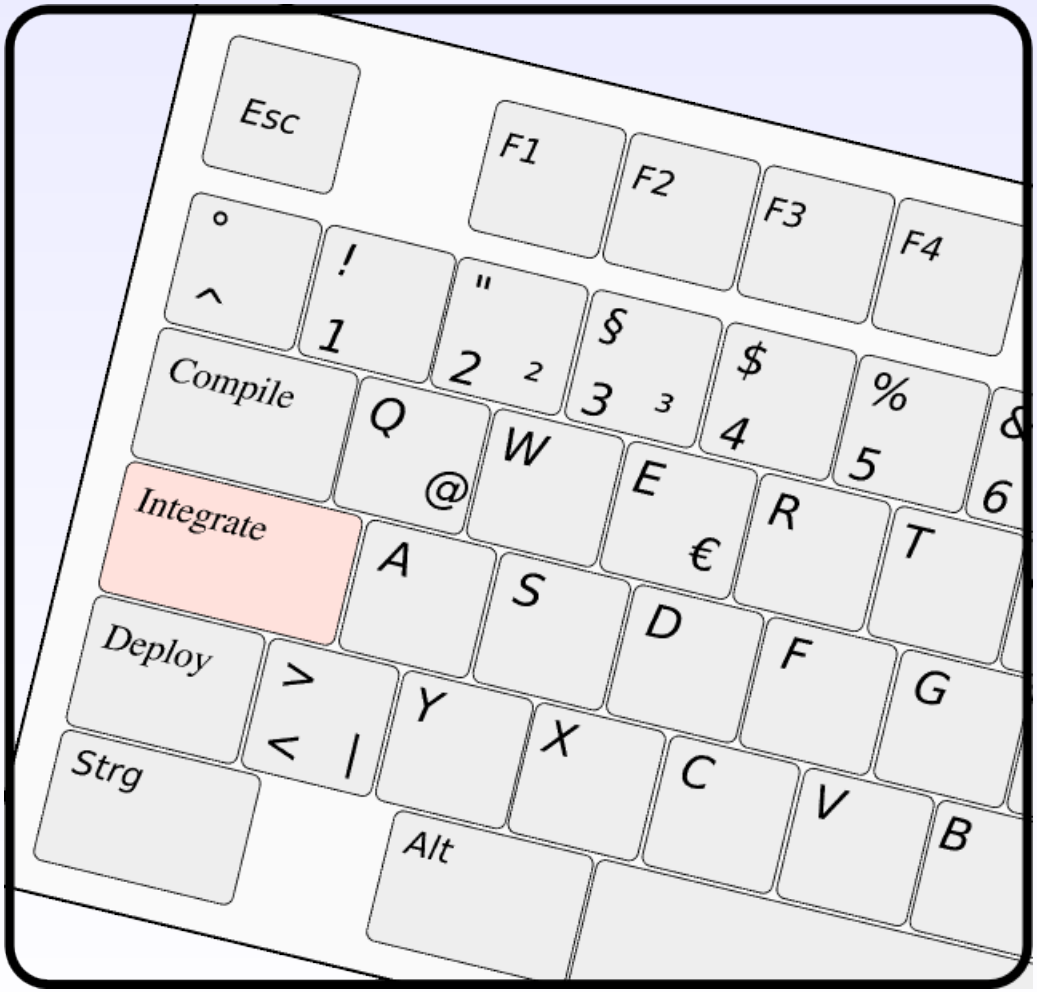
- Effort more predictable (no hard to predict integration phase)
- Reduced risk and higher quality
  - Less residual defects
  - Always a fully integrated product

Question

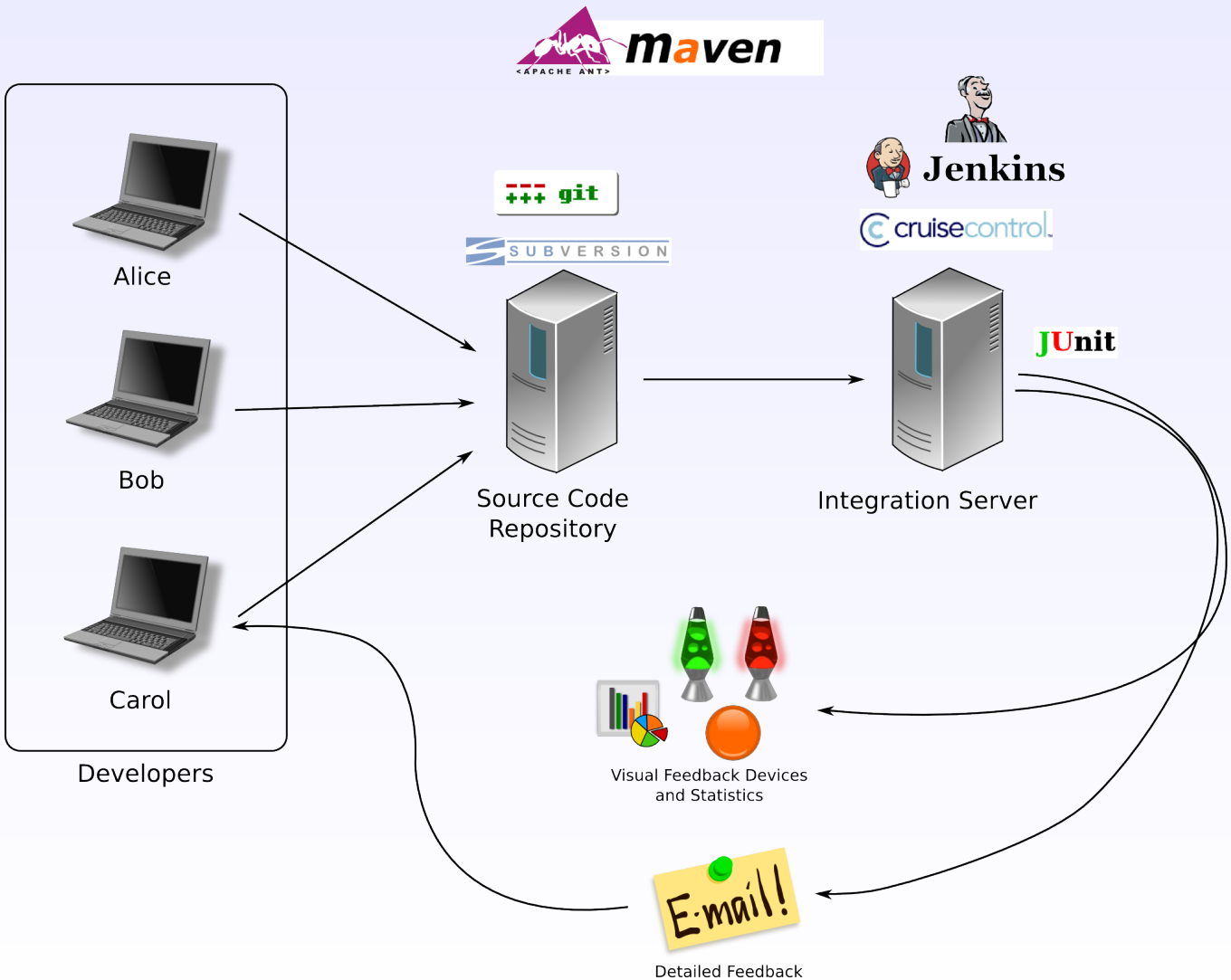
Now let's briefly have look at the mentioned aspects *automation* and *SCM*.

Automation

The Integrate Button



Tools


maven
Jenkins
git
SUBVERSION
cruisecontrol
JUnit
Alice
Bob
Carol
Developers
Source Code
Repository
Integration Server
Visual Feedback Devices
and Statistics
E-mail!
Detailed Feedback

An Ant Build Script

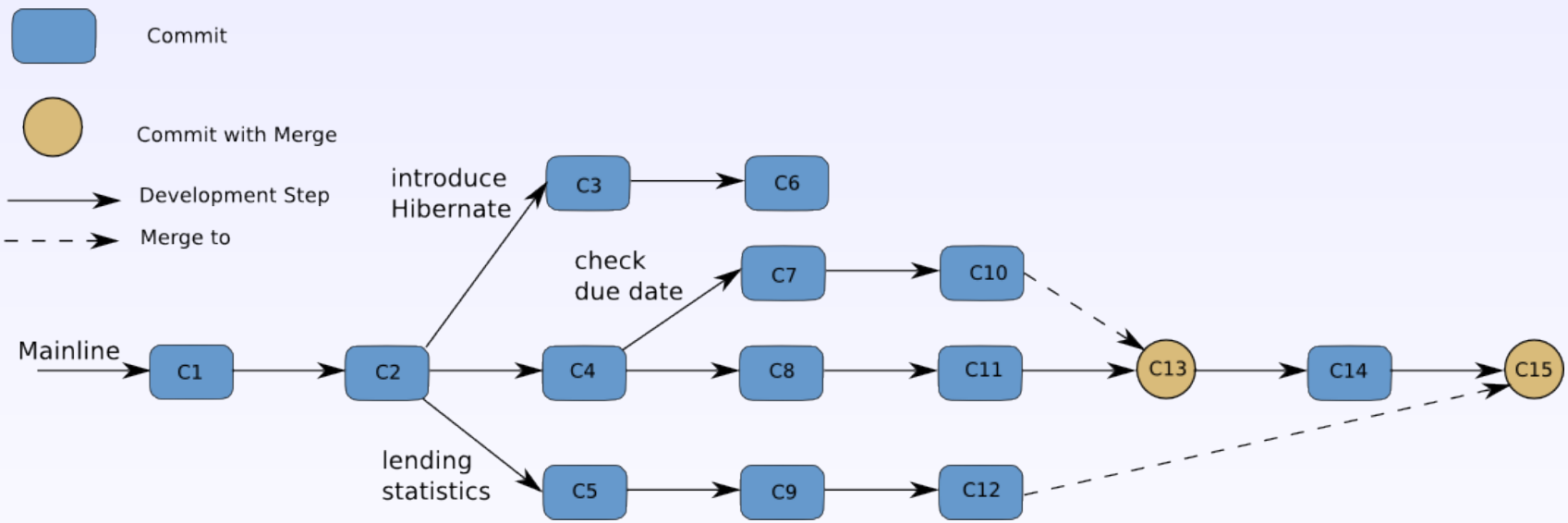
```
<?xml version="1.0"?>
<project name="MyLibrarySoftware" default="integrate">
    ...

    <target name="integrate" depends="clean,checkout,compile" description="Checkout, compile and test
        everything">
        ...
        <junit printsummary="yes" haltonfailure="yes">
            ...
        </junit>
    </target>

    ...
</project>
```

SCM

Feature Branches

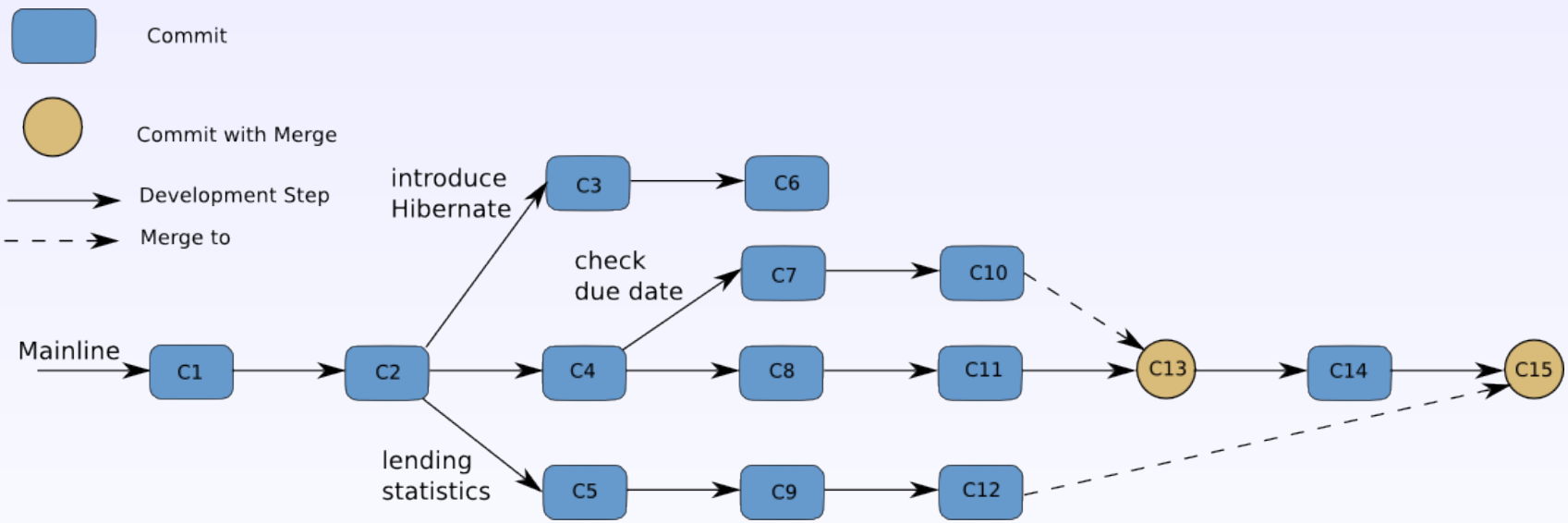


Feature Toggles

- Disable features instead of not integrating them
- Several possibilities
  - No button in GUI
  - Compiler switches
  - Command line options
  - Configuration files
  - ...

```
#ifdef FEATURE1_PRESENT
    CallFeature1();
#endif
```

In Example


Commit
Commit with Merge
Development Step
Merge to
Mainline
C1
C2
introduce
Hibernate
C3
C6
check
due date
C7
C10
C4
C8
C11
C13
C14
C15
lending
statistics
C5
C9
C12

Conclusion

- CI means to integrate several times a day
- CI needs a high degree of automation
- CI has some effects on SCM

- Agile processes require CI
- Plan-driven processes can also benefit from CI

Questions?

Literature

Paul M. Duvall, Steve Matyas, and Andrew Glover.
Continuous Integration: Improving Software Quality and Reducing Risk.
Addison-Wesley Professional, 7 2007.

(for others see seminar paper)

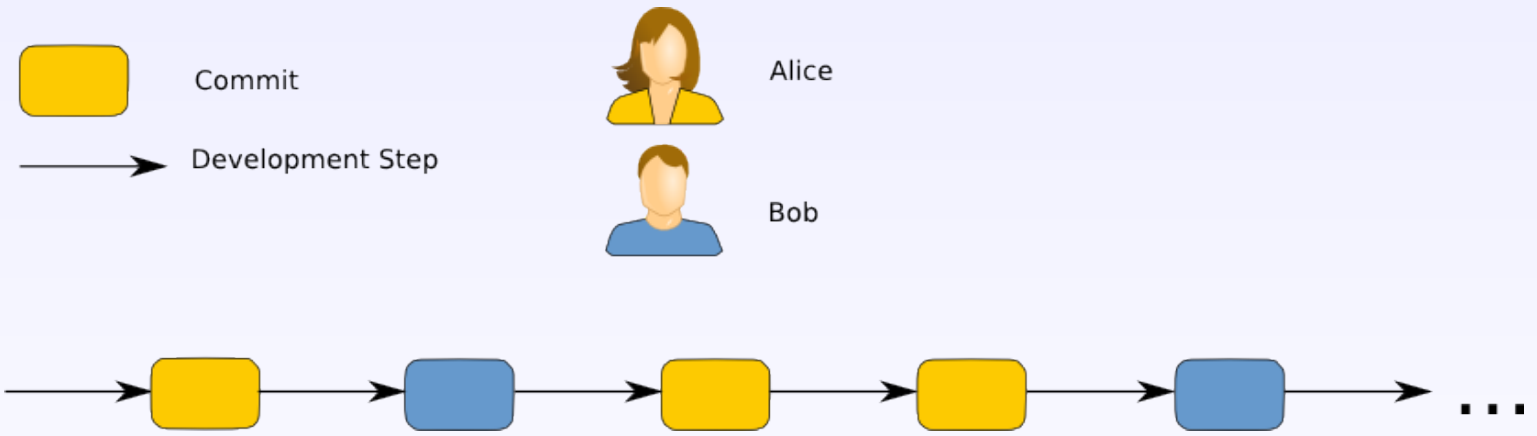
Appendix

Build Automation

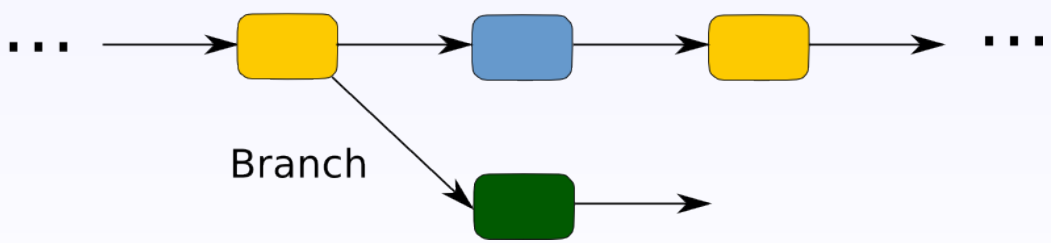
Build

"**build** is much more than a compile [...]. A build may consist of the compilation, testing, [...], and deployment—among other things. A build acts as the process for putting source code together and verifying that the software works as a cohesive unit." [DMG07]

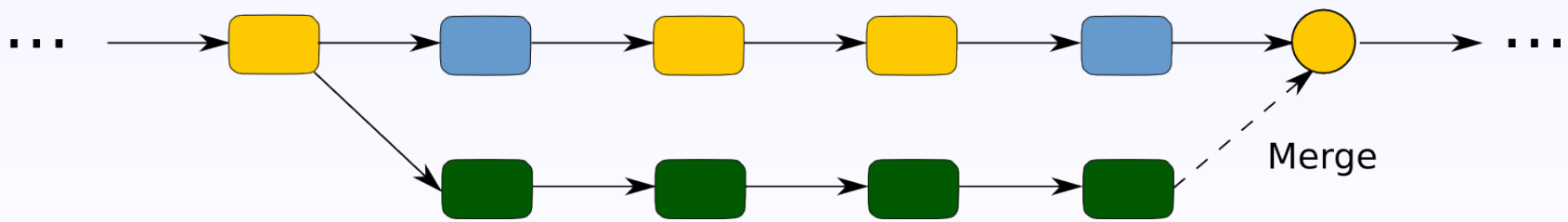
Branching (1/3)



Branching (2/3)



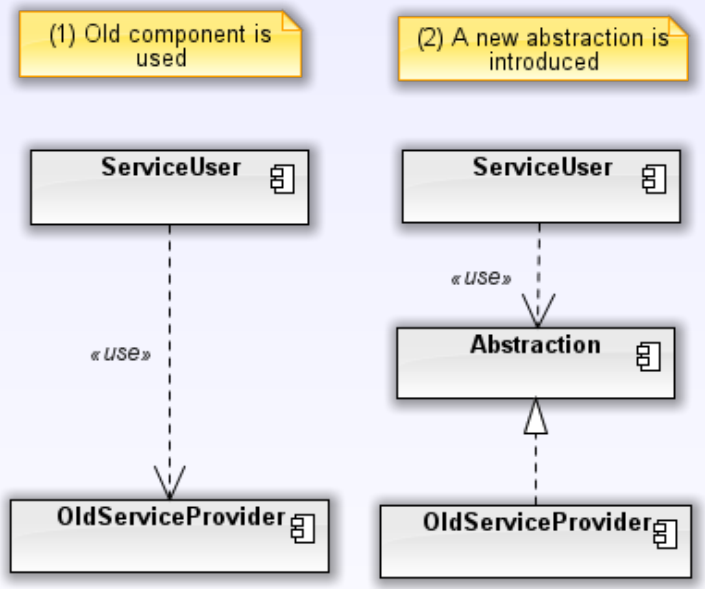
Branching (3/3)



Branching Models

- Branching models tell when to branch and merge
- One often used branching model is called “feature branches”

Branch by Abstraction (1/2)



Branch by Abstraction (2/2)