TEST ARCHITECTURE

G. Molines

2017-2018

Université Nice Sophia Antipolis

POLYTECH NICE-SOPHIA

INTRODUCTION

Software quality

- What is it?

Software quality

- What is it?
- No Defects?

Software quality

- What is it?
- No Defects?
- Or….

- Known defects?

Known defects???

- Goals of quality are:
  - Know and document bugs
  - Verify them for regression
  - Find workarounds
  - Feed more requirements

How do you **choose** your defects?

How do you **choose** your defects?

- Dev methodology
- Team culture
- Good practices
- Correct soft. Architecture
- Early Integration

- And…

How do you **choose** your defects?

- Dev methodology
- Team culture
- Good practices
- Correct soft. Architecture
- Early Integration
- And…

- **Testing**

Types of tests

- Unit-tests
- Integration tests
- Acceptation tests

- …ok…. What else?

Types of tests

- Unit Tests
- Integration Tests
- GUI Tests
- Non-regression Tests
- Coverage Tests
- Load Tests
- Stress Tests
- Performance Tests
- Scalability Tests
- Reliability Tests
- Volume Tests
- Usability Tests
- Security Tests
- Recovery Tests
- L10N/I18N Tests
- Accessibility Tests
- Installation/Configuration Tests
- Documentation Tests
- Platform testing
- Samples/Tutorials Testing
- Code inspections

Seriously, you do all of this stuff???

How the hell do you get organized???

Seriously, you do all of this stuff???

How the hell do you get organized???

TEST ARCHITECT

TEST ARCHITECT

Test architect role

- Choose test tools
- Define practices
- Build base frameworks → reusability
- Find test data
- Global test bucket consistency, strategy

Test Tools

Criteria

- Language
  - Eg: java -> jUnit and variants (HttpUnit, etc.)
- Type of app
  - Eg: desktop, mobile
- Layer where test is applied
  - Eg: UI → Selenium, RFT and other robots
- Goals
  - Eg: Perf -> jMeter
- Phase
  - Eg: functional vs system, see later

Define Test Practices

- Coding guidelines
- Test inventory tracking
  - Inventory tooling (RQM and the likes)
- Manual vs automated
  - ROI
  - Exploratory

Define Test Practices

- Policies
  - Eg: each new defect gets a testcase
- Testability
  - Eg: naming UI elements
- Repeatability
  - Setup / cleanup guidelines
  - Or: VM snapshoting

Test Data

- Representative samples
- Customer data
  - through support ticket
- Adhoc data
  - Eg: to cause error conditions
- Need be managed
  - Inventoried
  - Anonymized (confidential info)
  - Setup (Eg: prod-like DB)

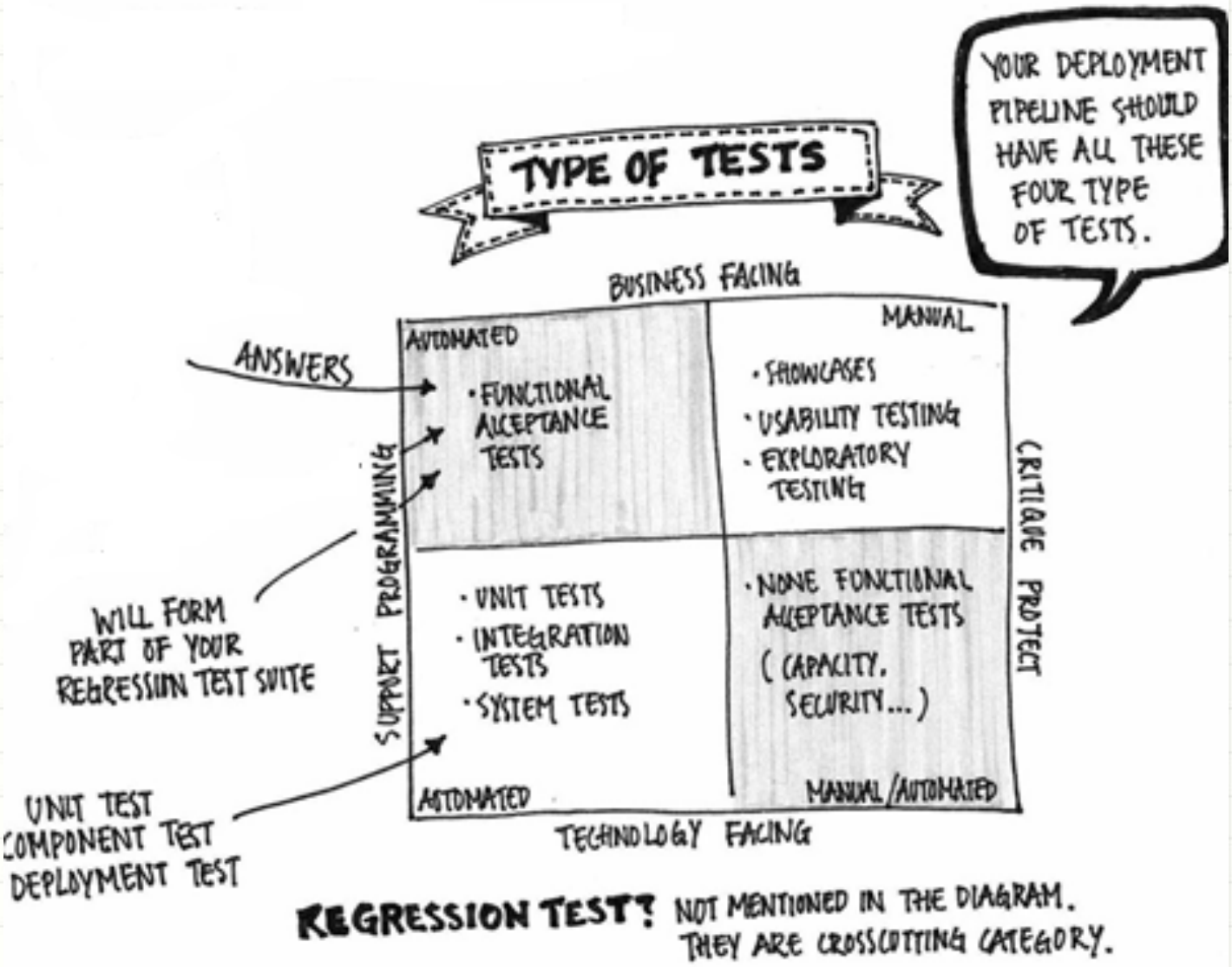
Test Strategy

- Aka: where and when to test
- Sources of information
  - User defect reports
  - Coverage data
  - User story / functional understanding
- Measure effectiveness
  - # defects / regressions found
  - Time to find next defect
  - Time spent in maintenance

Test Dimensions

Can be organized based on:

- Product area
- Goals (find defects, measure perf., …)
- Team in charge
- Dev. Phase
- Execution time

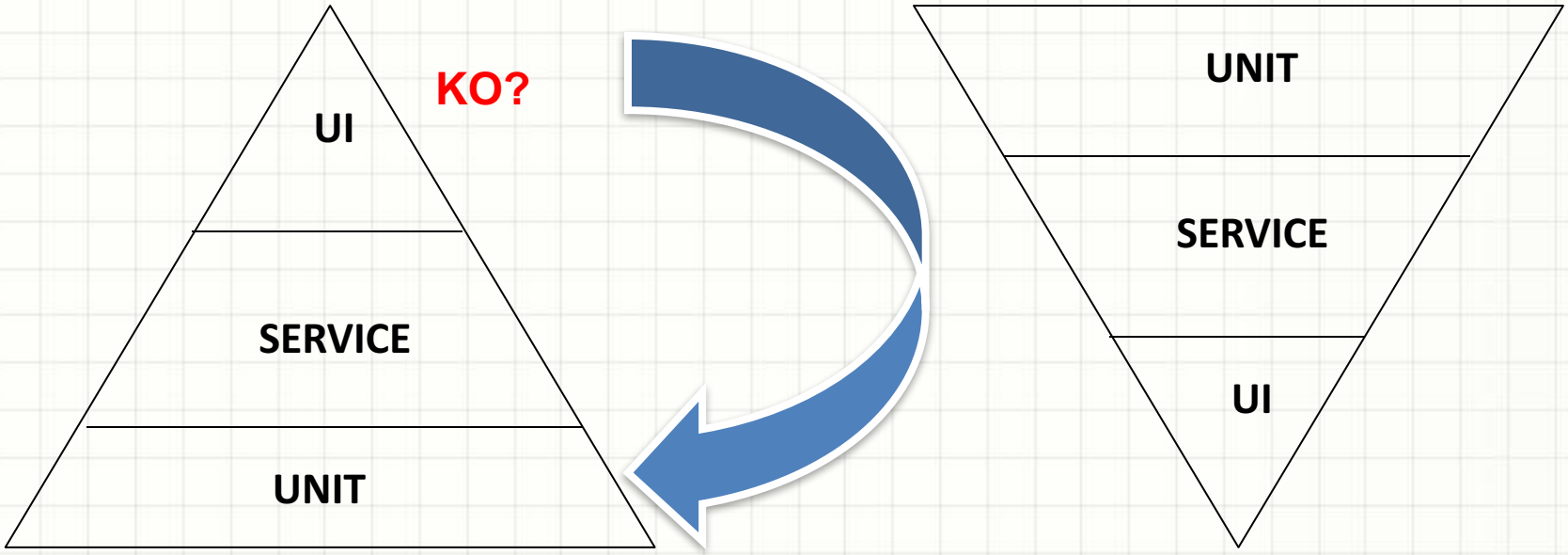


(From: Nhan Ngo)

Test Sizes

- Number
- Execution time
- Depth of paths traversed

> *"In particular I always argue that high-level tests are there as a second line of test defense. If you get a failure in a high level test, not just do you have a bug in your functional code, you also have a missing unit test. Thus whenever you fix a failing end-to-end test, you should be adding unit tests too."*
>
> *Martin Fowler*

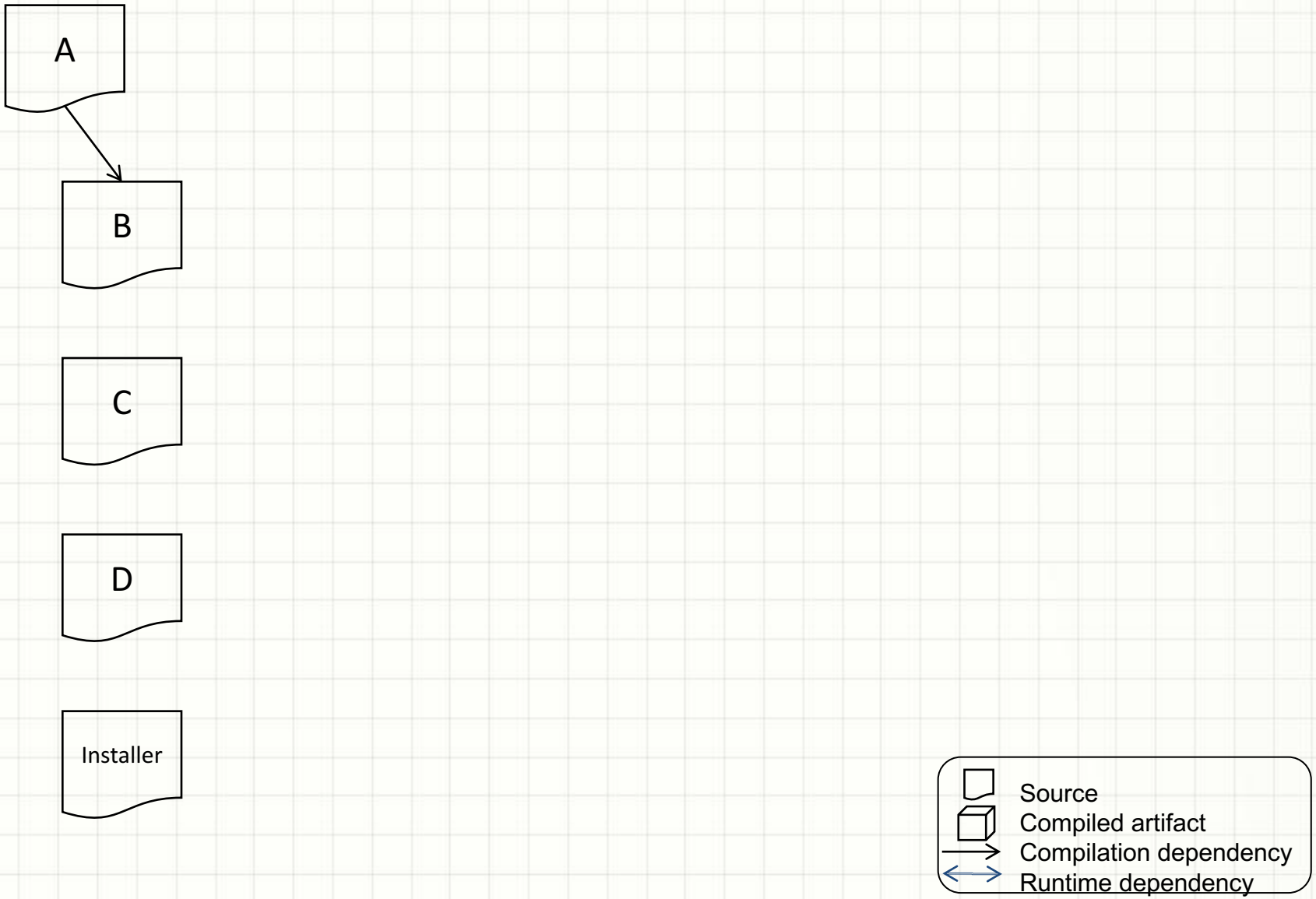


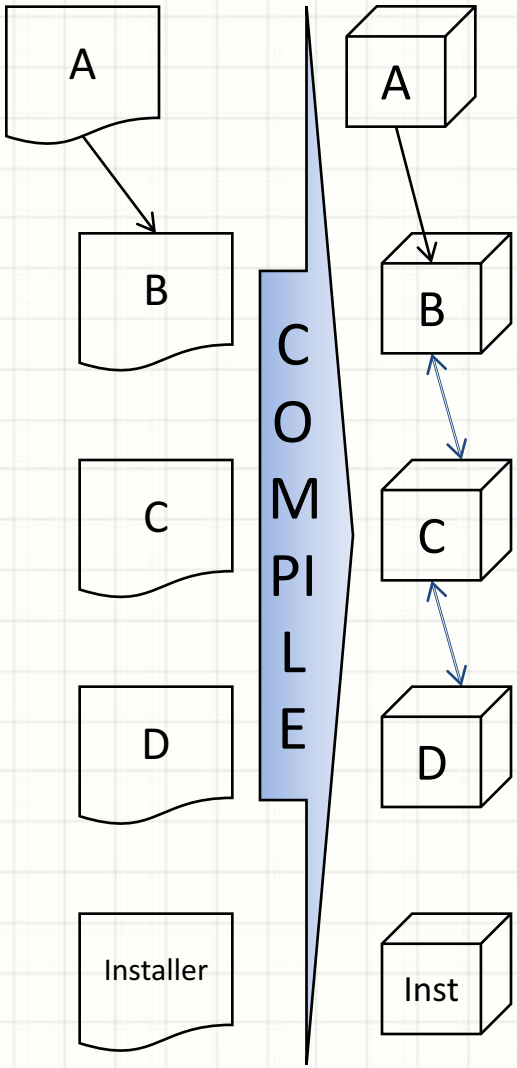
Volume

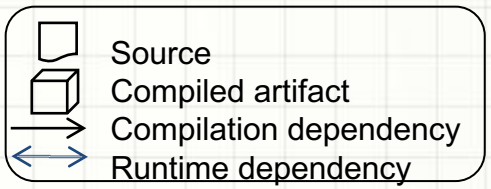
Run time

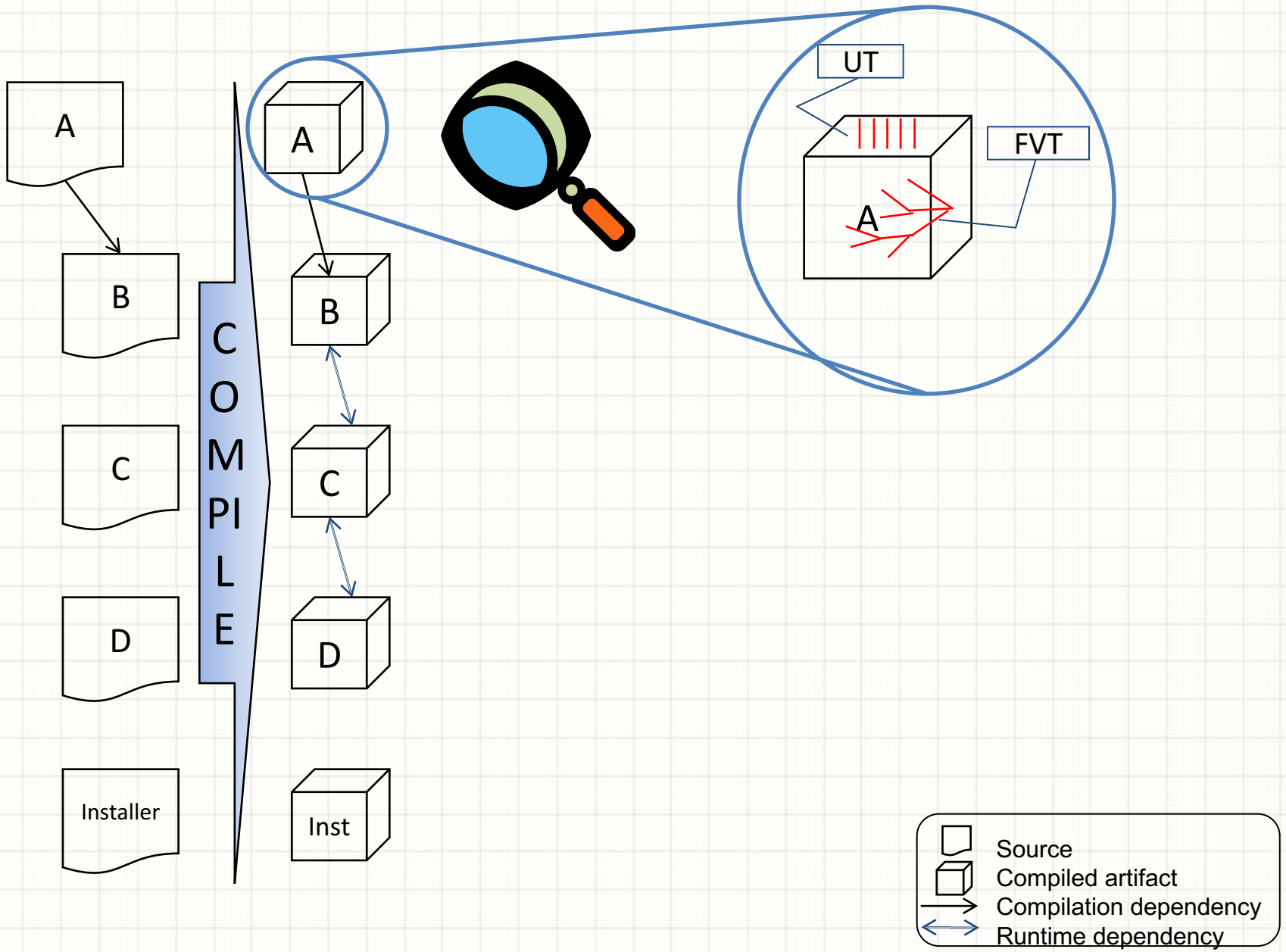
Functional vs. System

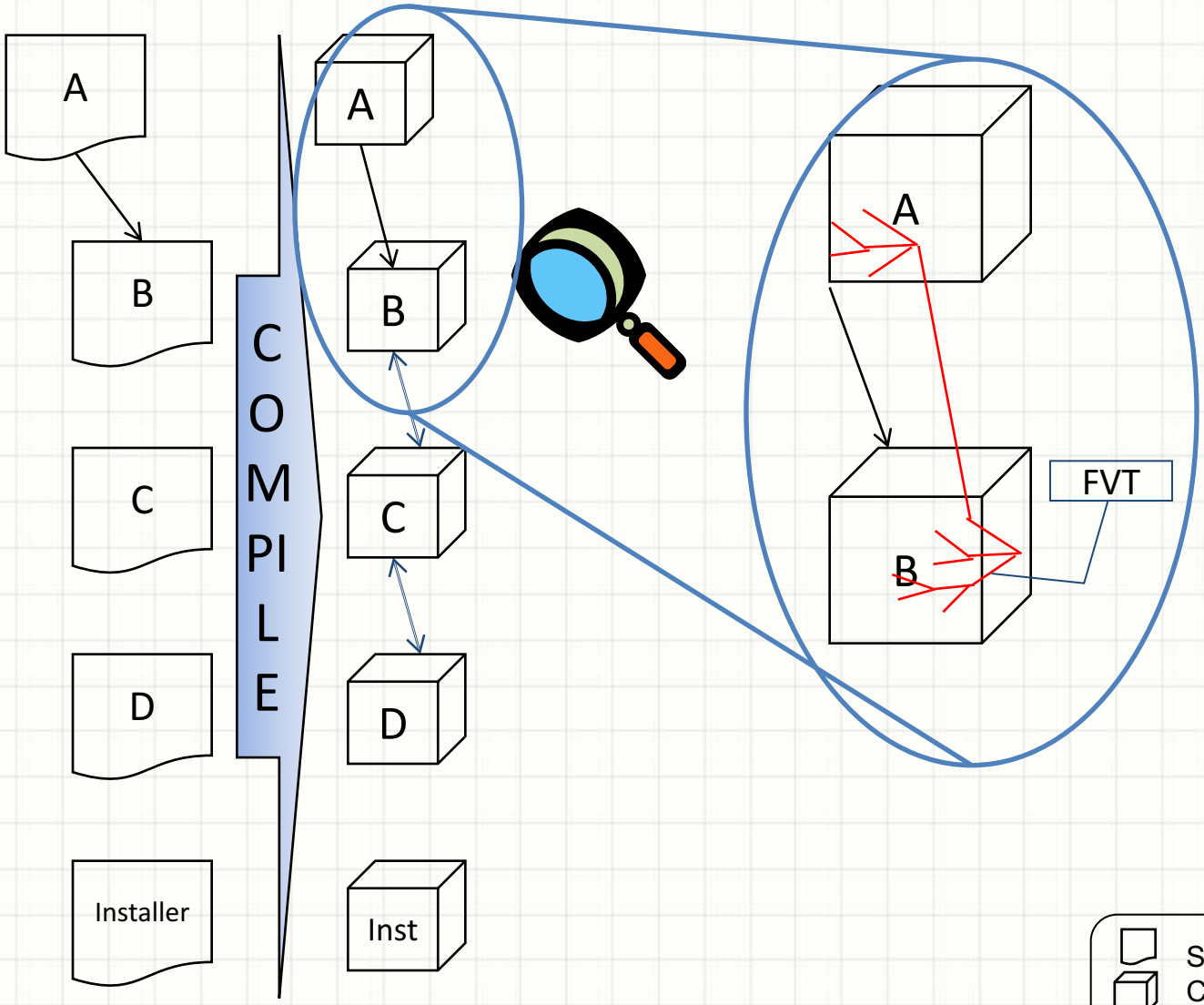
- Some make a big difference between Functional and System testing
- But
- It’s just a continuum

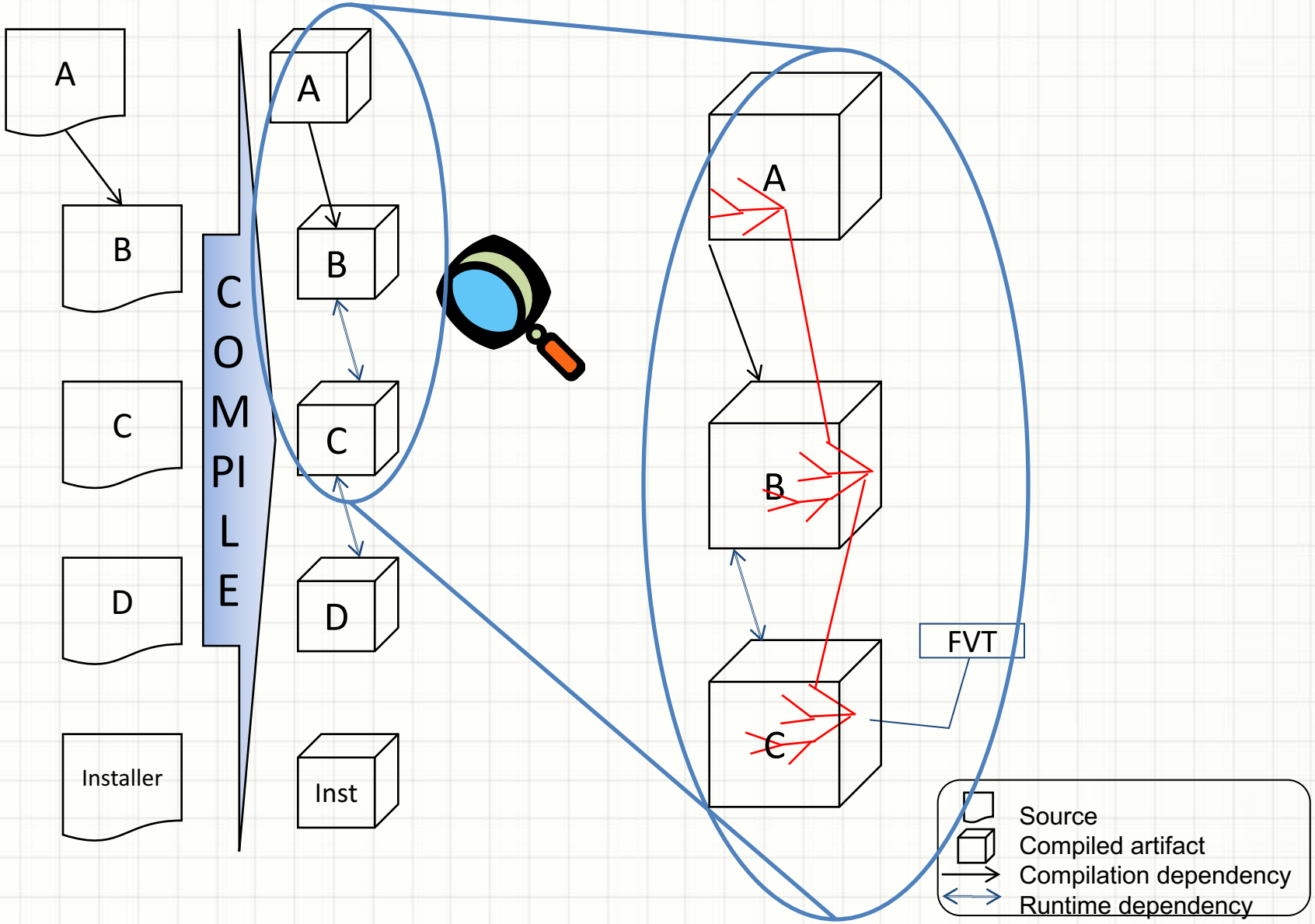

A
B
C
D
Installer
Source
Compiled artifact
Compilation dependency
Runtime dependency


A
B
C
D
Installer
COMPILE
A
B
C
D
Inst

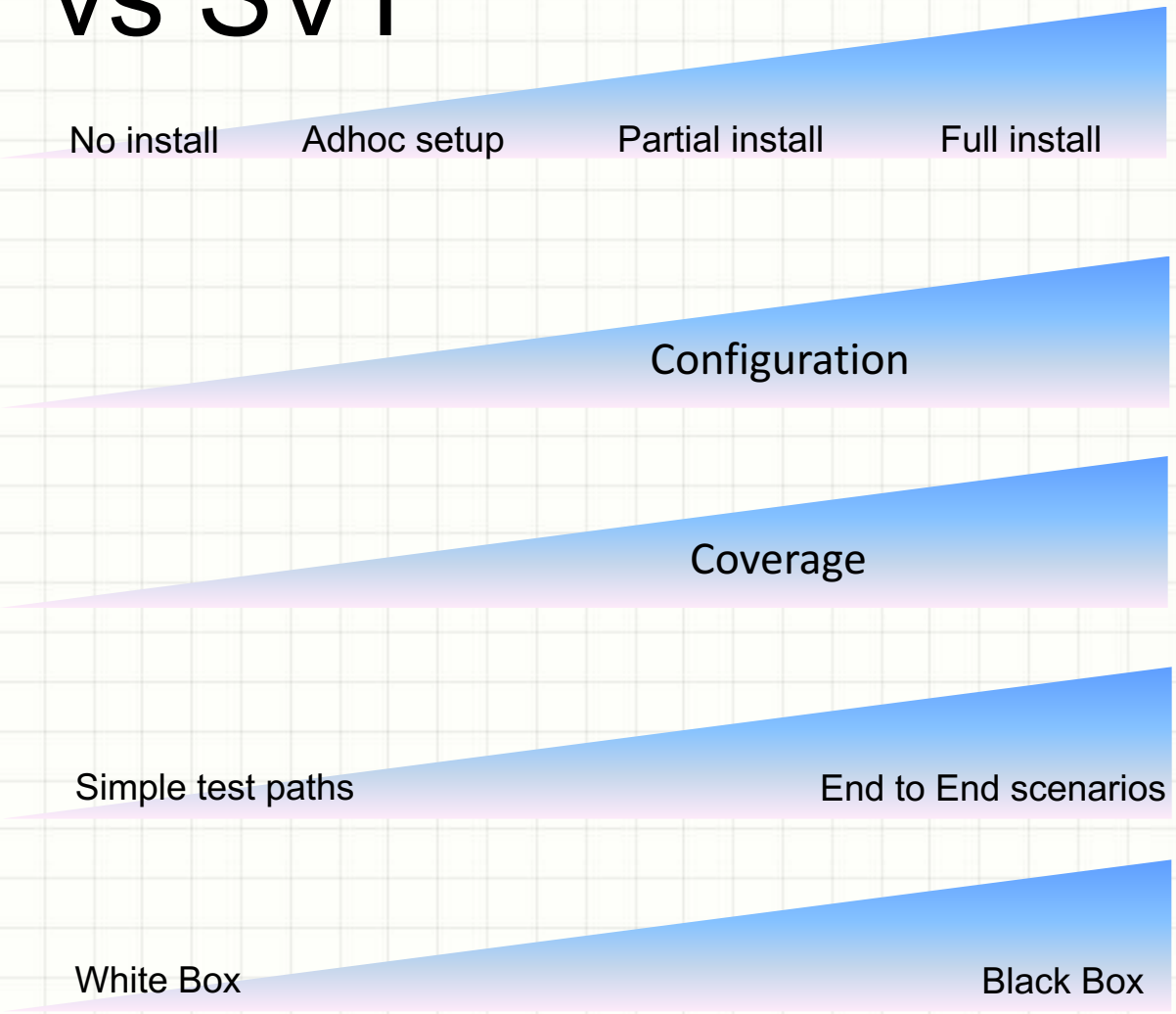

Source
Compiled artifact
Compilation dependency
Runtime dependency


A
B
C
D
Installer
COMPILE
A
B
C
D
Inst
UT
FVT
A
Source
Compiled artifact
Compilation dependency
Runtime dependency


A
B
C
D
Installer
COMPILE
A
B
C
D
Inst
A
B
FVT
Source
Compiled artifact
Compilation dependency
Runtime dependency


A
B
C
D
Installer
COMPILE
A
B
C
D
Inst
A
B
C
FVT
Source
Compiled artifact
Compilation dependency
Runtime dependency

FVT vs SVT


No install
Adhoc setup
Partial install
Full install
Configuration
FVT
Coverage
SVT
Simple test paths
End to End scenarios
White Box
Black Box

QUESTIONS?

Next two weeks

- This week:
  - TD: Mutants
- Software factory
  - Functional & Integration Tests
  - TD: finish Mutants, start tests on TCF

APPENDIX