UNIVERSITY OF TWENTE.

everett.

TRUSTED TO KNOW | HOW

MOVING TOWARDS CONTINUOUS DELIVERY

INTRODUCING TEST AUTOMATION WITH FITNESSE IN SYSTEM INTEGRATION PROJECTS

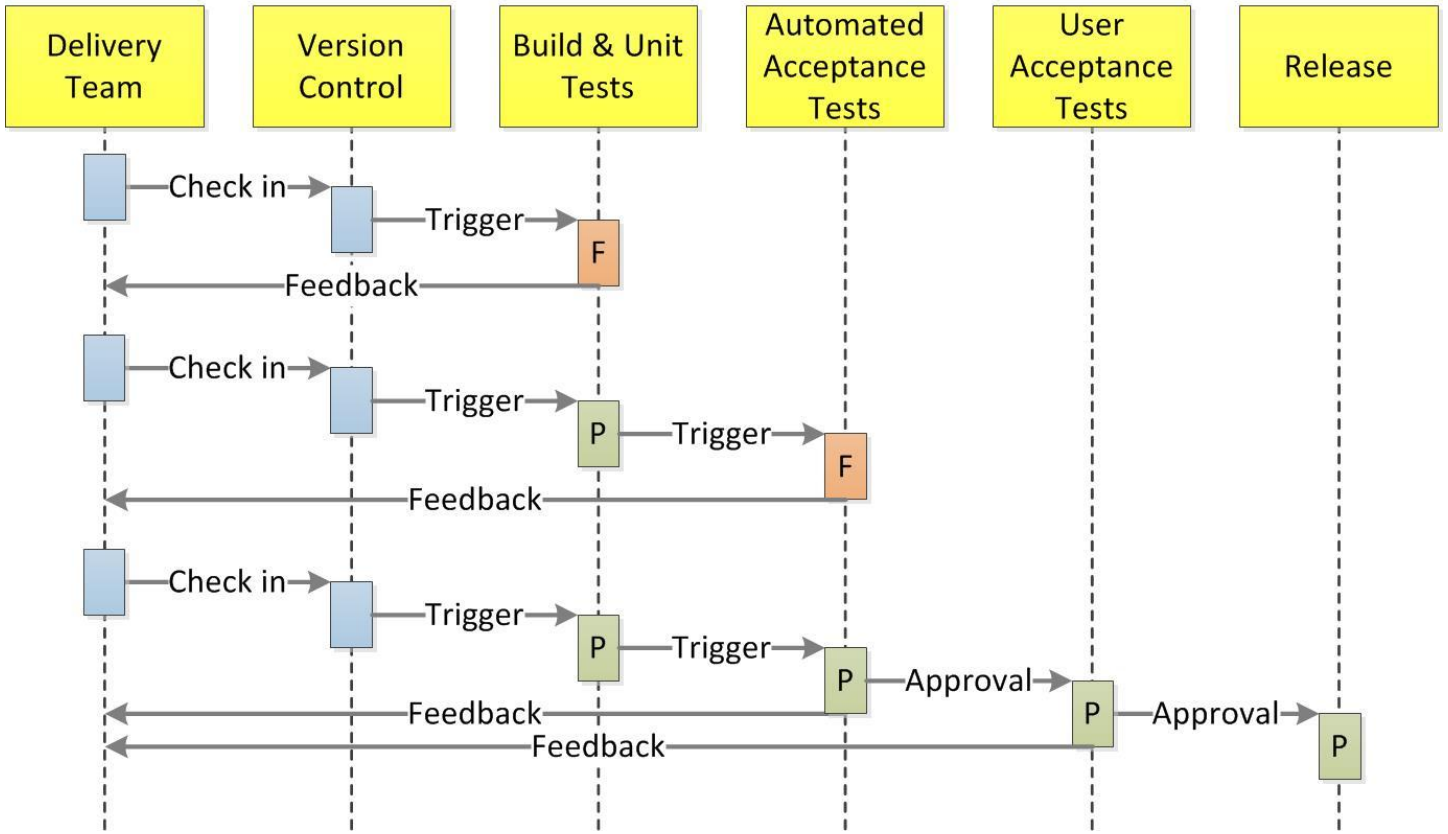
SANDRA DRENTHEN
s.drenthen@student.utwente.nl /
s.drenthen@alumnus.utwente.nl

CONTENT

- Background
- Test Strategy
- Case Study
- Conclusion
- Questions

BACKGROUND

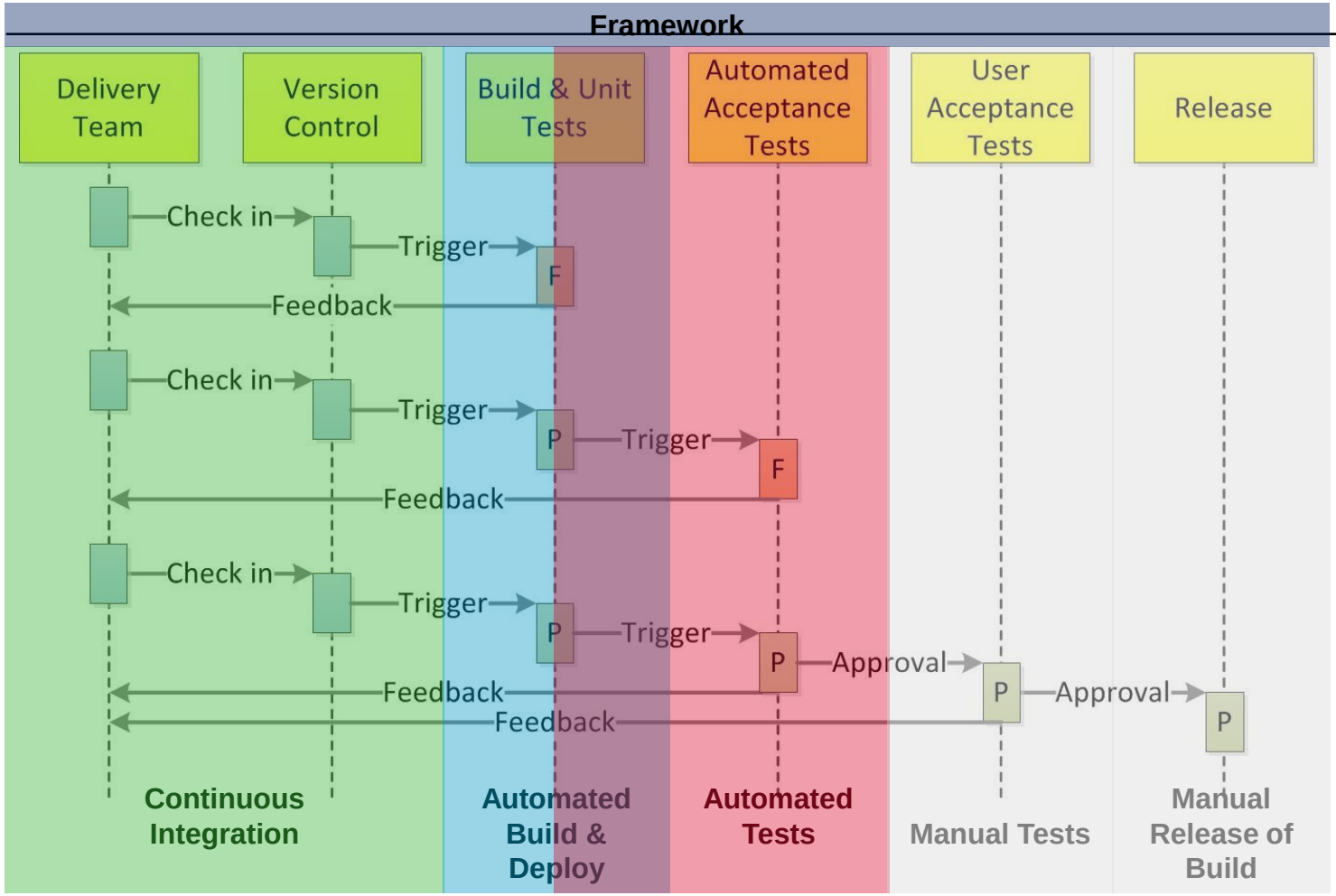
CONTINUOUS DELIVERY
THEORY



[source]: J. Humble and D. Farley, Continuous delivery: reliable software releases through build, test, and deployment automation.,Addison-Wesley Professional, 2010.

CONTINUOUS DELIVERY

THEORY



IDENTITY SOLUTIONS

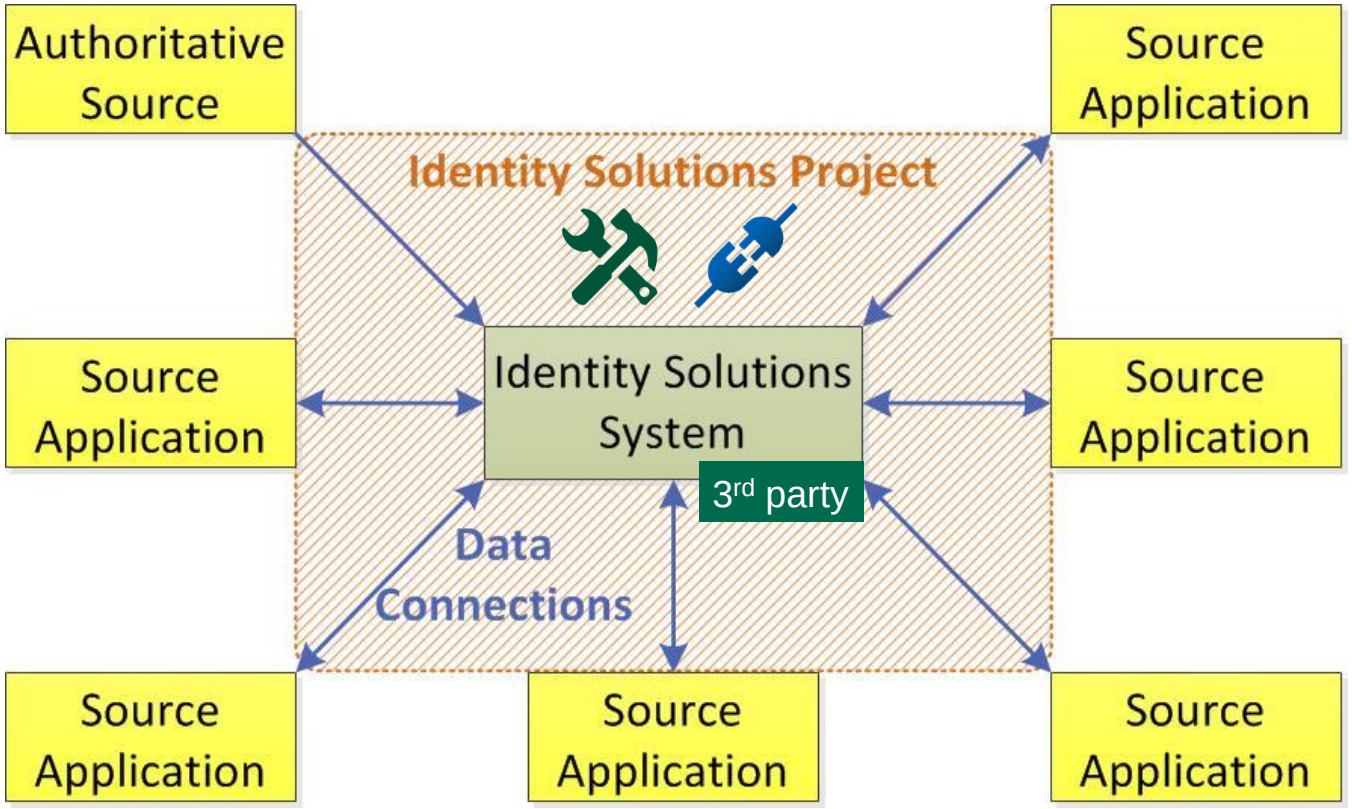
IDENTITY & ACCESS GOVERNANCE

- **Main goal:**
  - **Be in control** of access rights & being able to **prove it**

- **Offers the client:**
  - **Correlate data**
  - Specify/check **rules**
  - Specify/run **certifications**
  - Possibility to implement **provisioning**
  - Possibility to define **Workflows**

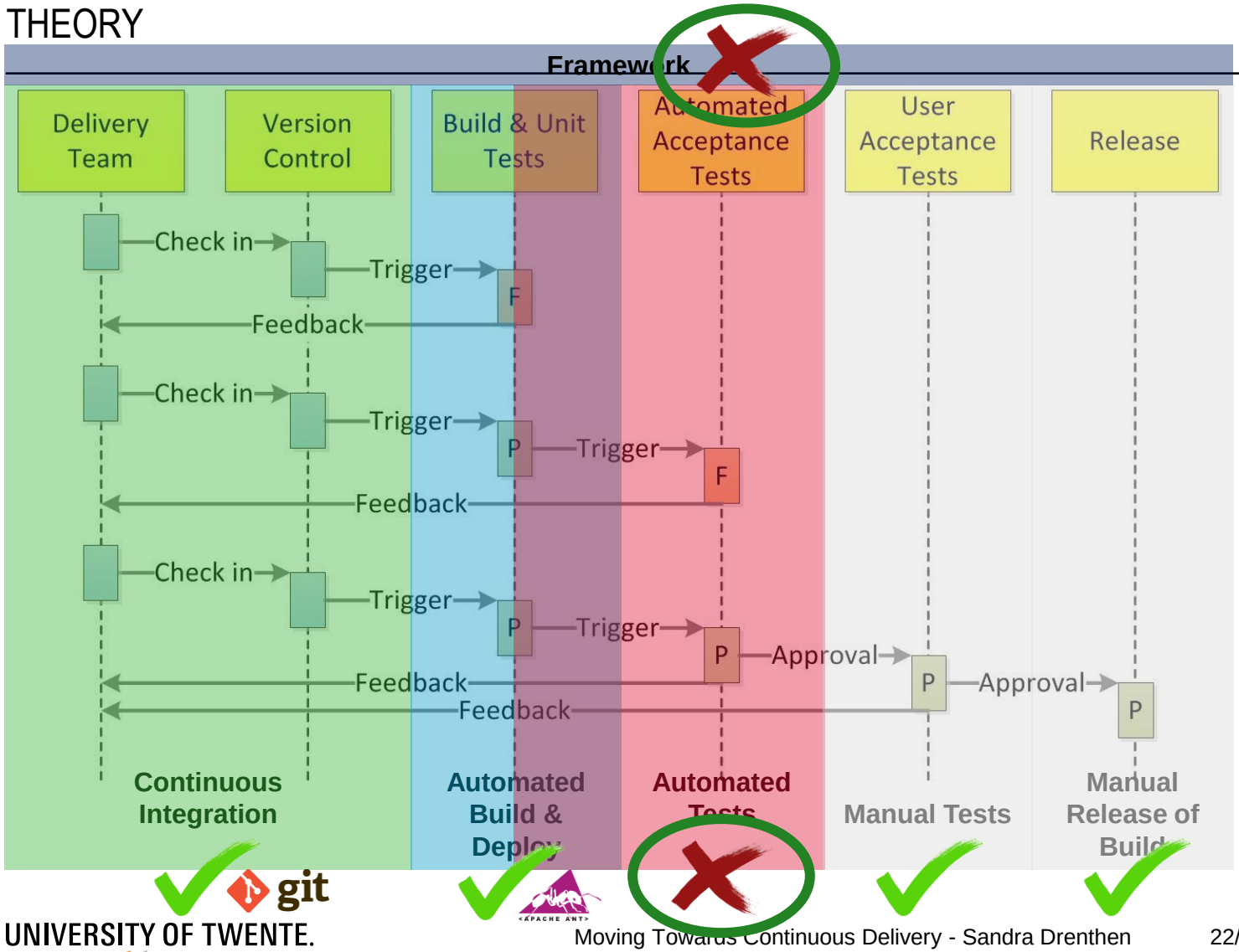
IDENTITY SOLUTIONS

IDENTITY & ACCESS GOVERNANCE



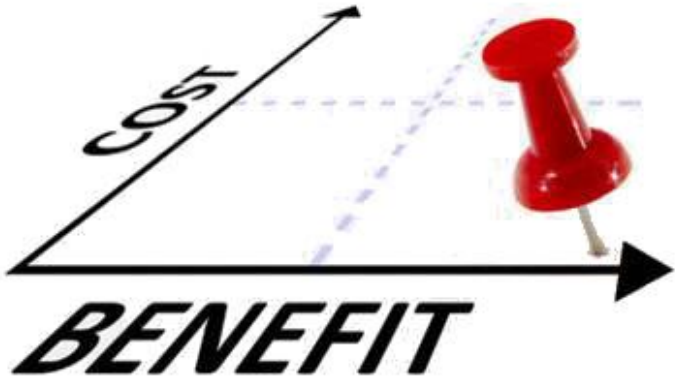
CONTINUOUS DELIVERY

THEORY



GOALS

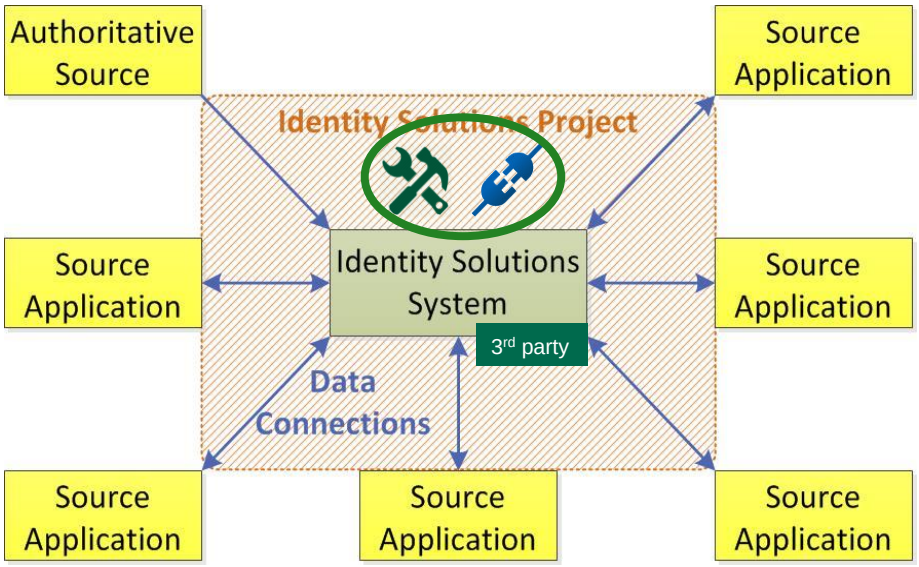
- **Automate and improve process of software delivery**
  - By introducing Continuous Delivery
- **Reduce costs of testing**
  - By re-running/re-using tests
- **Enhance quality of product**
  - By finding faults earlier
  - By having better test-documentation



TEST STRATEGY

TEST AUTOMATION
WHAT DO YOU WANT TO AUTOMATE?

- **In our case, most risks are in:**
  - **Configurations**
  - **Integrations**
- **Automated Tests on:**
  - **Functional level**



[must-read]: B. Marnick, "When Should a Test Be Automated?," in *Proceedings of The 11th International Software/Internet Quality Week*, San Francisco, 1998.

TEST AUTOMATION
TEST DATA

- **Type of test-data**
  - **Production** data
  - **Masked** data
  - **Generated/Fictional** data

- **In our case:**
  - **Production** data with extra constraints

TEST AUTOMATION
HOW DO YOU COMMUNICATE WITH SUT?

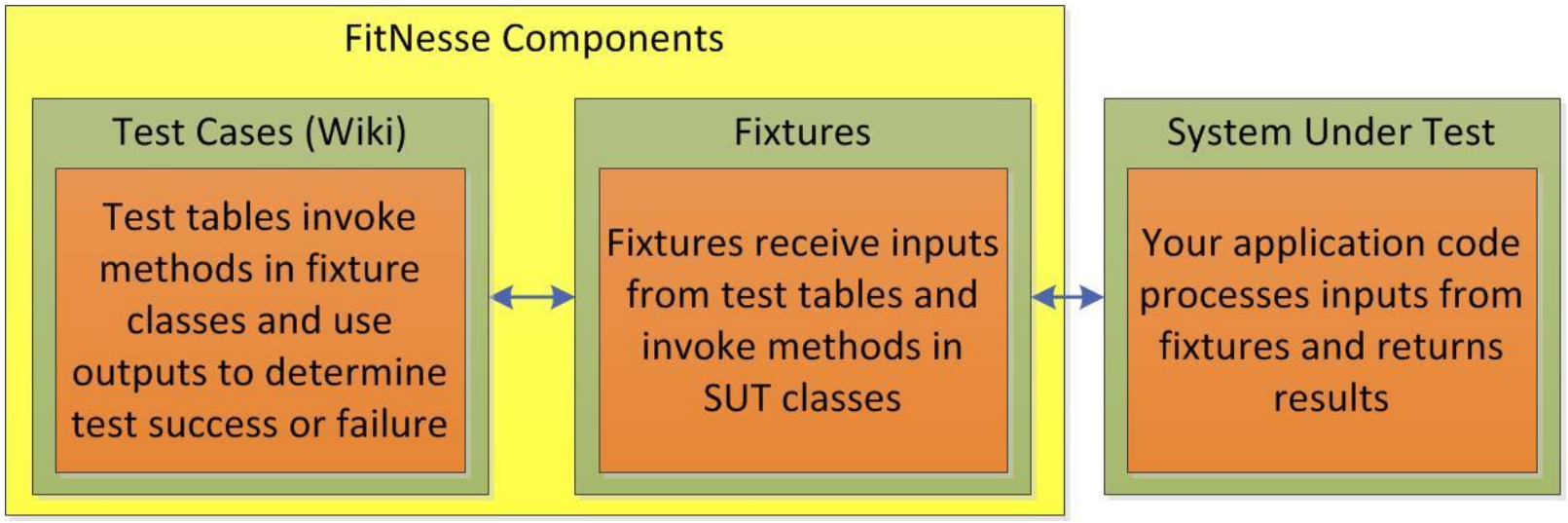
- **Communicate with SUT**
  - **Find tools** that are capable do this and try a **proof of concept**.

- **Our choice:**
  - Because **functional**, communicate with the **API** of the SUT
  - Use the **FitNesse tool**

TEST AUTOMATION
FITNESSE SELLING POINTS

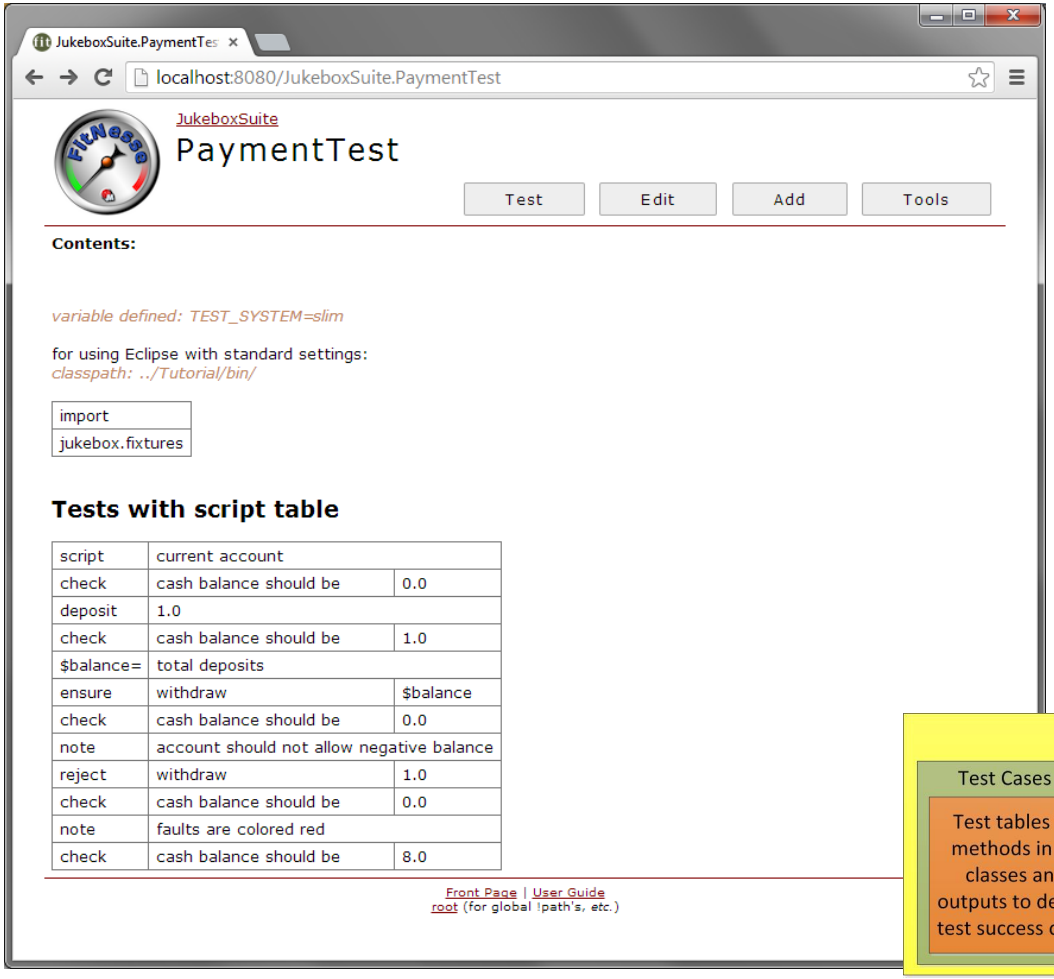
- **Automated acceptance testing framework**
  - Use it **everywhere**
    cross-platform, lightweight, open-source, standalone
  - **Easy** to use and learn
    wiki and java syntax
  - **Readable** tests
    Tests are wiki tables
  - Define tests **together**
    business users can write tests

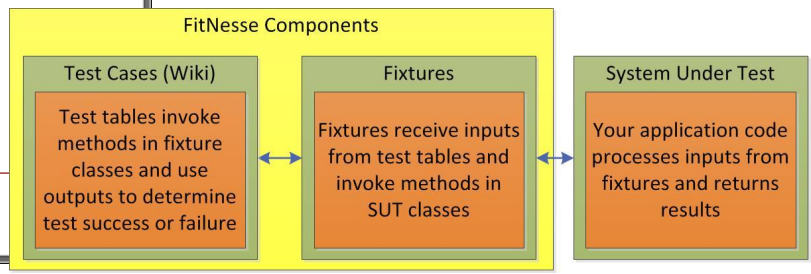
TEST AUTOMATION
FITNESSE COMPONENTS



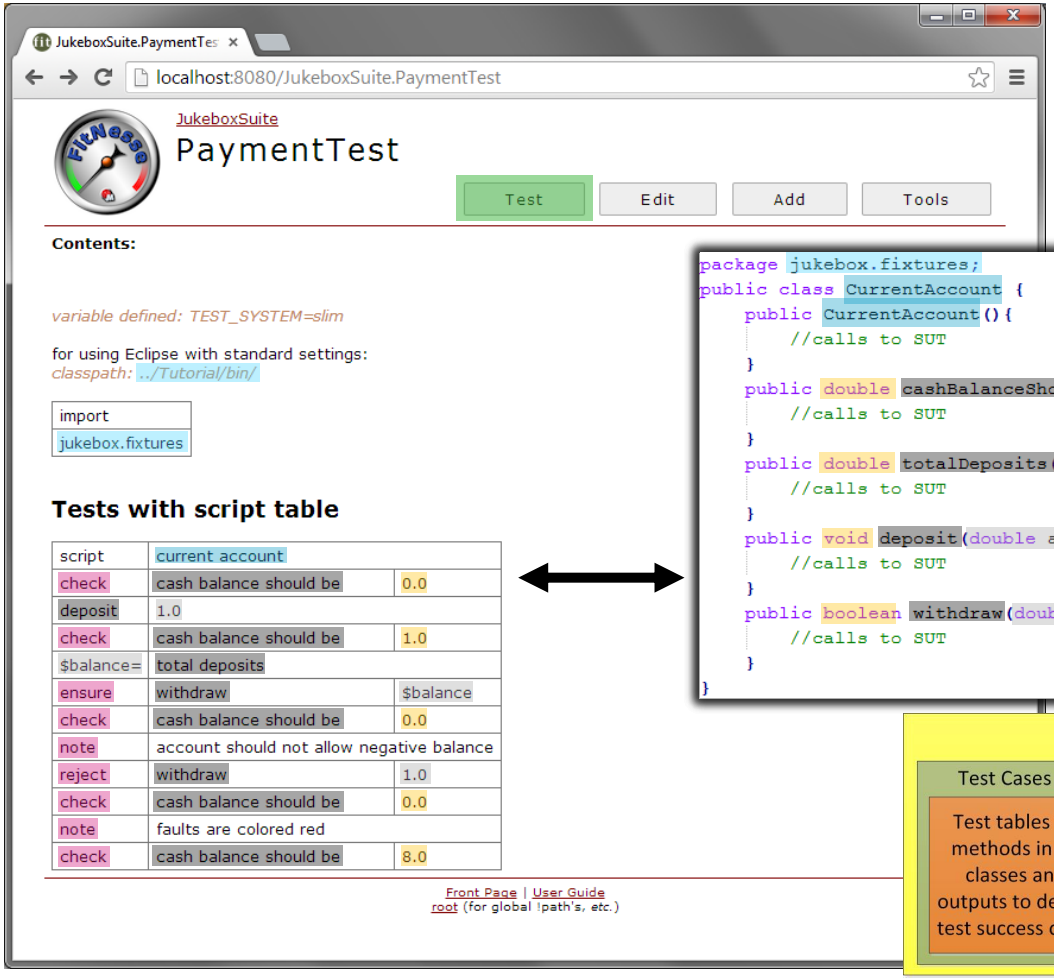
[source]: M. Sorens, "Acceptance Testing With FitNesse, The Overview," Juli 2013. [Online]. Available: https://www.simple-talk.com/dotnet/.net-tools/acceptance-testing-with-fitnesse,-the-overview/

TEST AUTOMATION
FITNESSE EXAMPLE



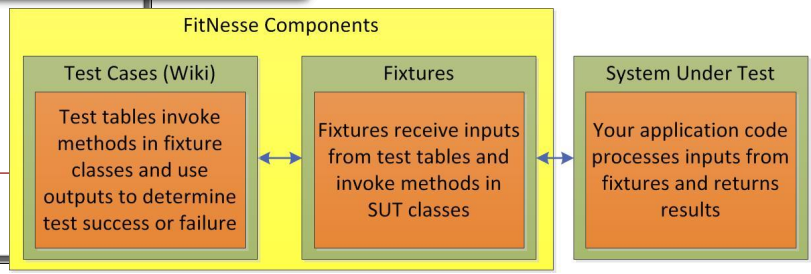


TEST AUTOMATION
FITNESSE EXAMPLE

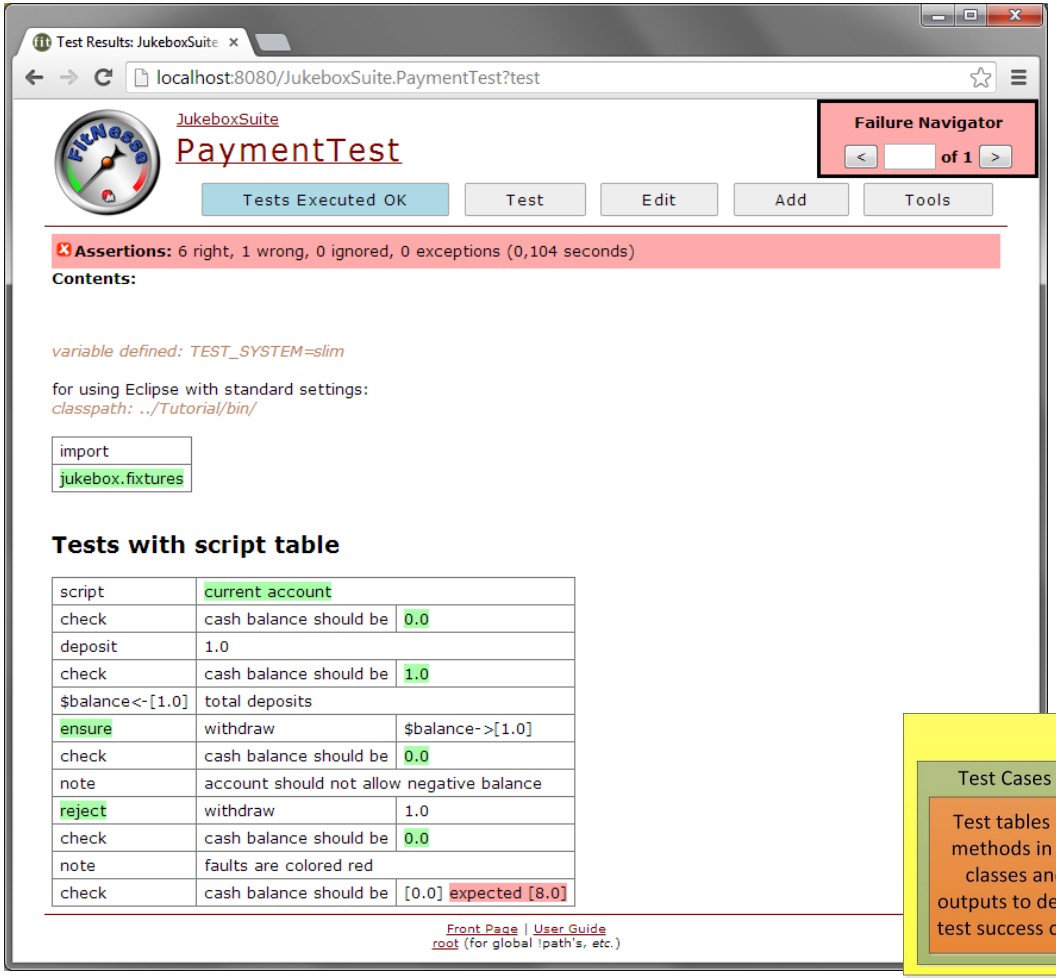


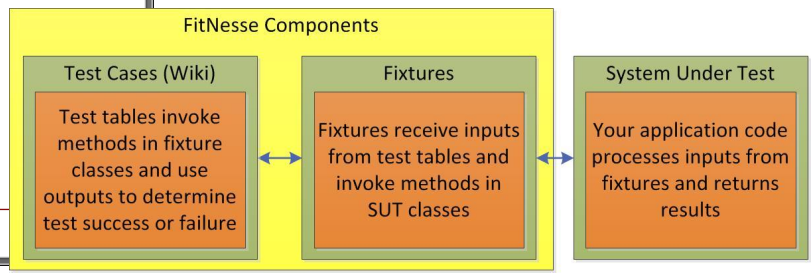

```
package jukebox.fixtures;
public class CurrentAccount {
    public CurrentAccount(){
        //calls to SUT
    }
    public double cashBalanceShouldBe() {
        //calls to SUT
    }
    public double totalDeposits() {
        //calls to SUT
    }
    public void deposit(double amount) {
        //calls to SUT
    }
    public boolean withdraw(double amount) {
        //calls to SUT
    }
}
```

System(s) Under Test



TEST AUTOMATION
FITNESSE EXAMPLE



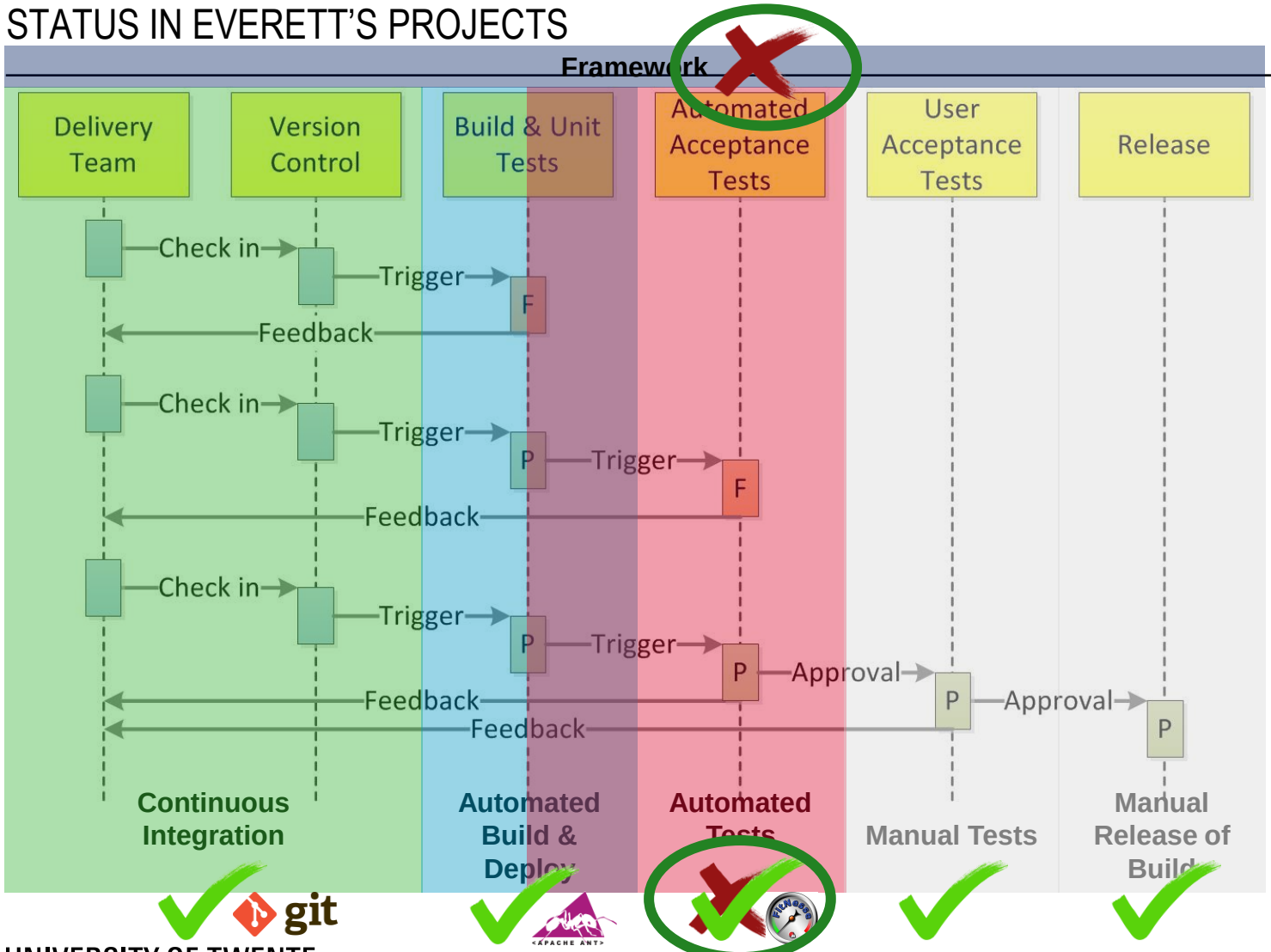


TEST AUTOMATION
FITNESSE USAGE

- **FitNesse Guidelines**
  - Combine with **Test Driven Development**
  - Use FitNesse's **SLIM-test system** (instead of FIT)
  - Use FitNesse's **SetUp/TearDown** pages
  - **Group** tests in suites, based on **functionality**
  - **Work together** by using **GIT** for FitNesse as well

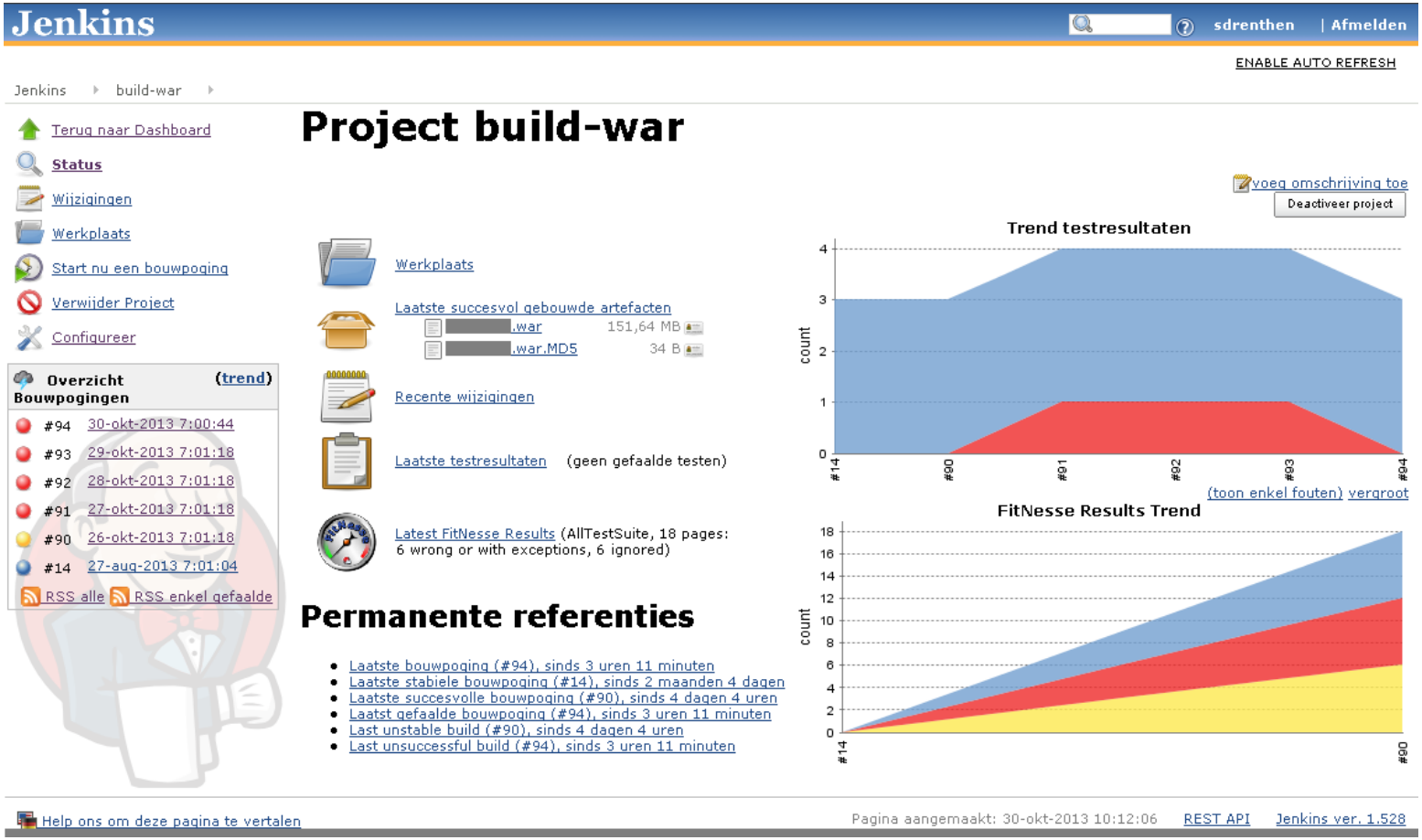
CONTINUOUS DELIVERY

STATUS IN EVERETT'S PROJECTS



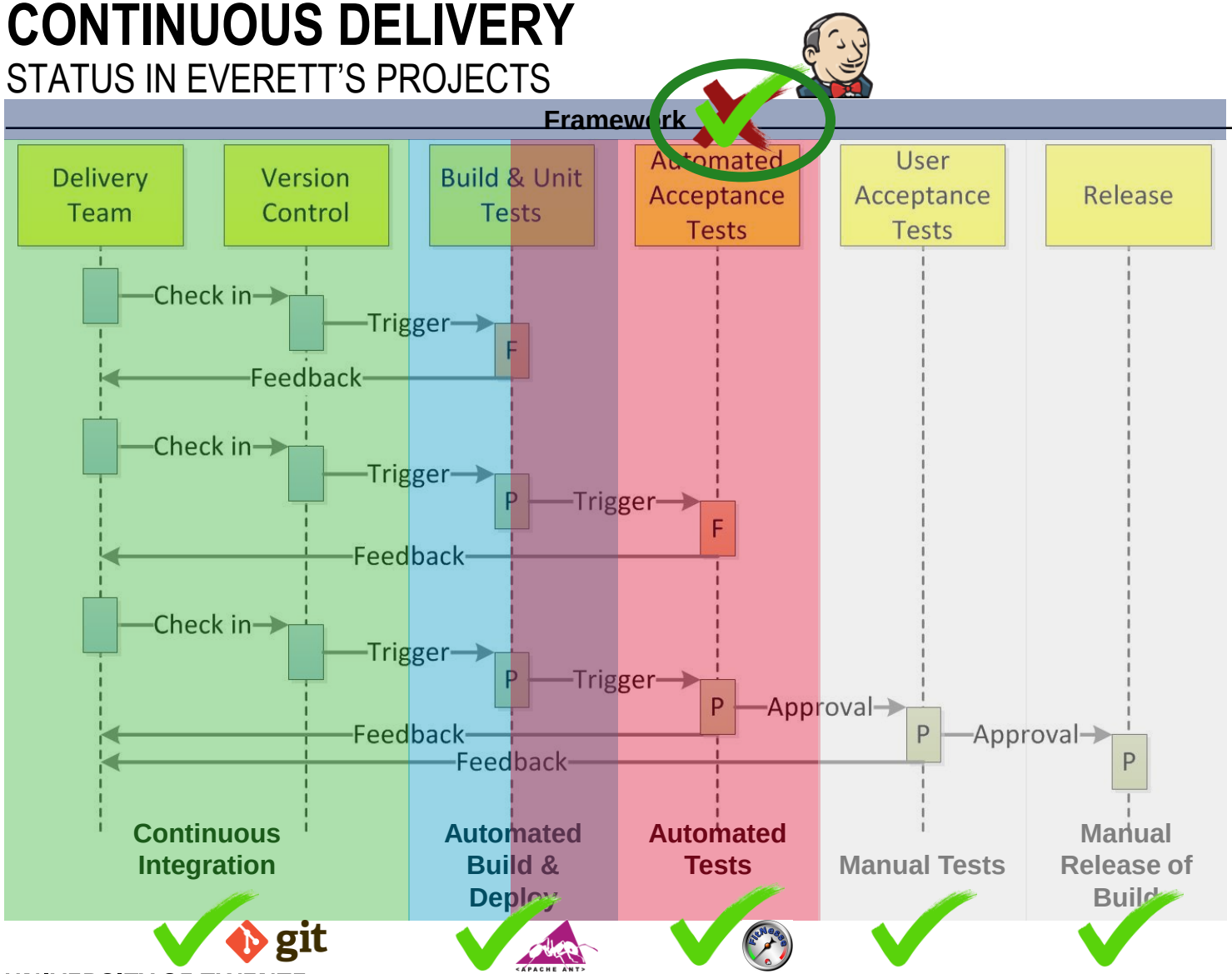
AUTOMATED BUILD, TEST AND DEPLOY

OUR CHOICE: JENKINS



CONTINUOUS DELIVERY

STATUS IN EVERETT'S PROJECTS

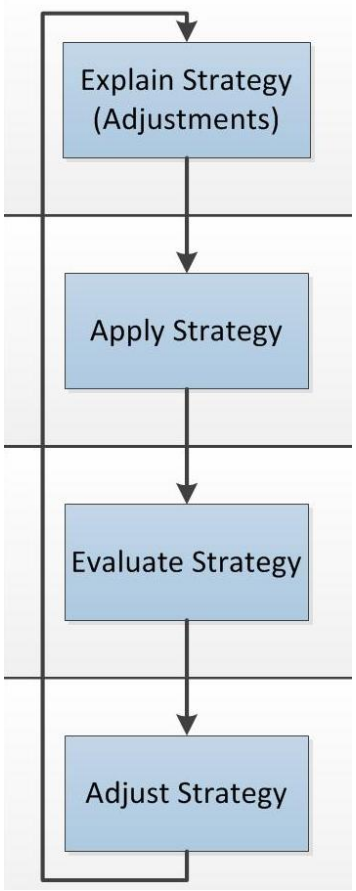


CASE STUDY

CASE STUDY
PROJECT & METHOD

- **Project:**
  - **At a client** of Everett
  - **6 team-members** (3 of Everett, 3 of client)
  - 3 days in a week, **sprints of 2 weeks**

- **Method**
  - Used **5 sprints (= 5x2 weeks)**
    - **Iterative**
    - **Evaluate**: Survey + Interviews



CASE STUDY
ITERATIONS

- **First iteration:**
  - Started with **basic strategy**
  - I created first test with team-member
  - Problem: **public API/Console**

- **Second Iteration**
  - Use **private API (created framework)**
  - I created **demo-tests** on demo environment
  - Chose & Install **Jenkins**

CASE STUDY
ITERATIONS

- **Third iteration:**
  - Team created **first tests** on project
  - Problem: running **wiki isn't updated** automatically
- **Fourth Iteration**
  - **Automatically** pull/push wiki **updates** to GIT
- **Fifth iteration:**
  - Held **Final Interviews**

RESULTS (1)

- **Amount of tests created**
  - **10** (by 4 team-members and me)
- **Benefits**
  - Short **learning curve**
  - **Regression** tests can now be automated
- **Costs**
  - **Invest time** to create framework, tailor the strategy, automate tests
  - Need **test criteria** early in the sprints
  - Need a **dedicated person** for the continuous delivery process.

RESULTS (2)

- **Compared with previous projects :**
  - Did not yet **spend less time on testing**
    - Due to needed time investments
  - Did not yet **found bugs earlier / found more bugs**
    - Due to few automated tests and still lot of manual tests
  - In a **new project**, the strategy is promising for these goals

- **Reuse strategy and tests in other projects**
  - **Strategy: yes, Except API-framework** when using other software
  - **Test cases: no, Except some parts** when using the same software

CONCLUSIONS

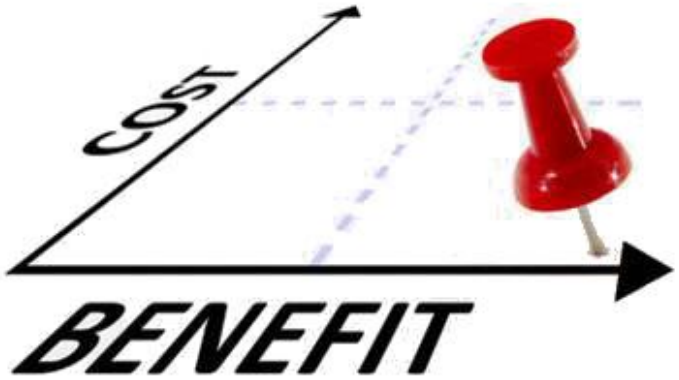
GOALS ACHIEVED?

✓ **Automate and improve process of software delivery**

- By introducing Continuous Delivery

✗ **Reduce costs of Testing**

- By re-running/re-using tests



✓ **Enhance quality of product**

- By finding faults earlier
- By having better test-documentation

QUESTIONS

UNIVERSITY OF TWENTE.

everett.

TRUSTED TO KNOW | HOW

MOVING TOWARDS CONTINUOUS DELIVERY

INTRODUCING TEST AUTOMATION WITH FITNESSE IN SYSTEM INTEGRATION PROJECTS

SANDRA DRENTHEN
s.drenthen@student.utwente.nl /
s.drenthen@alumnus.utwente.nl