Generation of Test Cases

for Executable Business Processes from Classification Trees

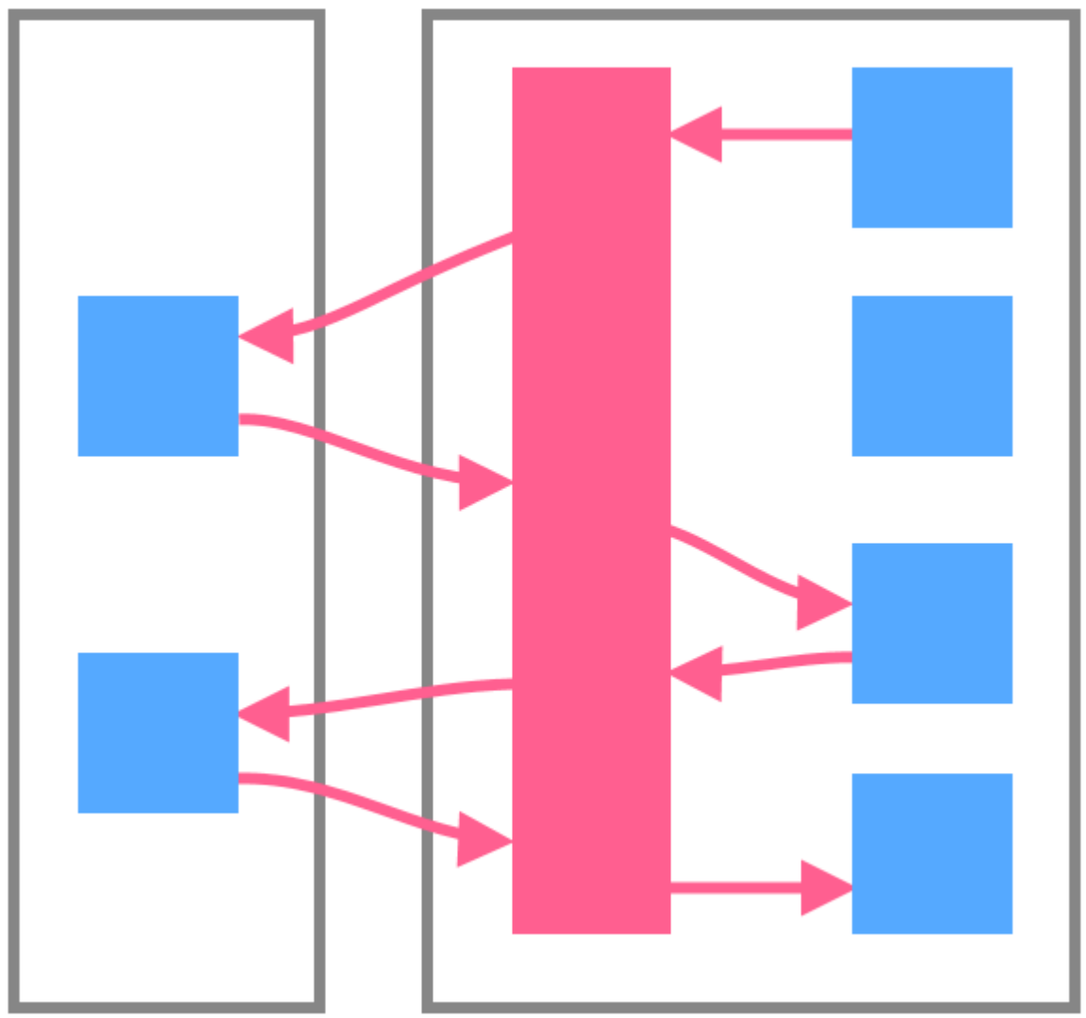
> Thilo Schnelle

> Daniel Lübke

innoQ

Basics

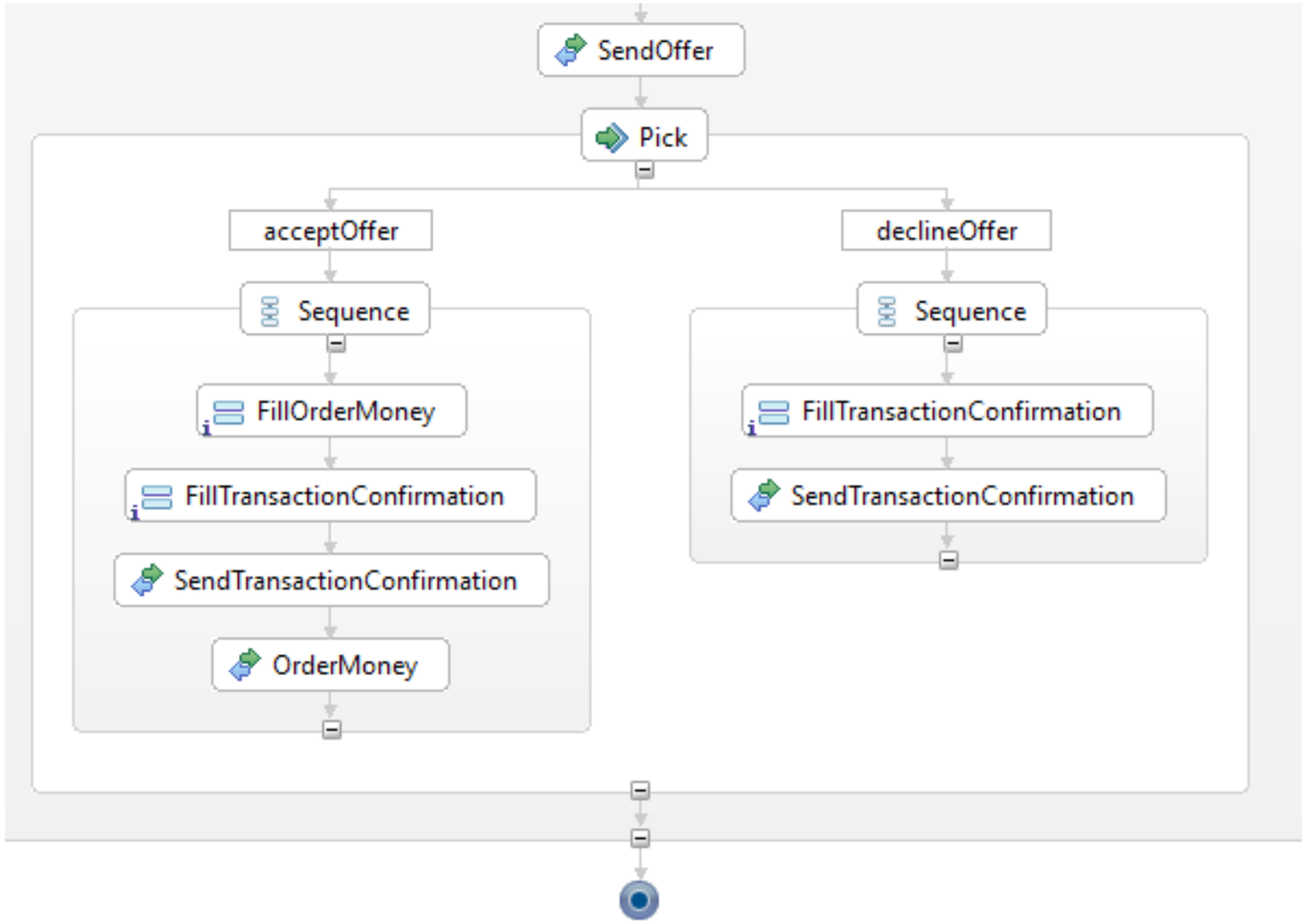
Business logic



BPEL

- Orchestration is a recurring standard task

  ⇒ Business Process Execution Language



Testing BPEL

- Complex as other programming languages

  ⇒ Requires testing

Testing BPEL

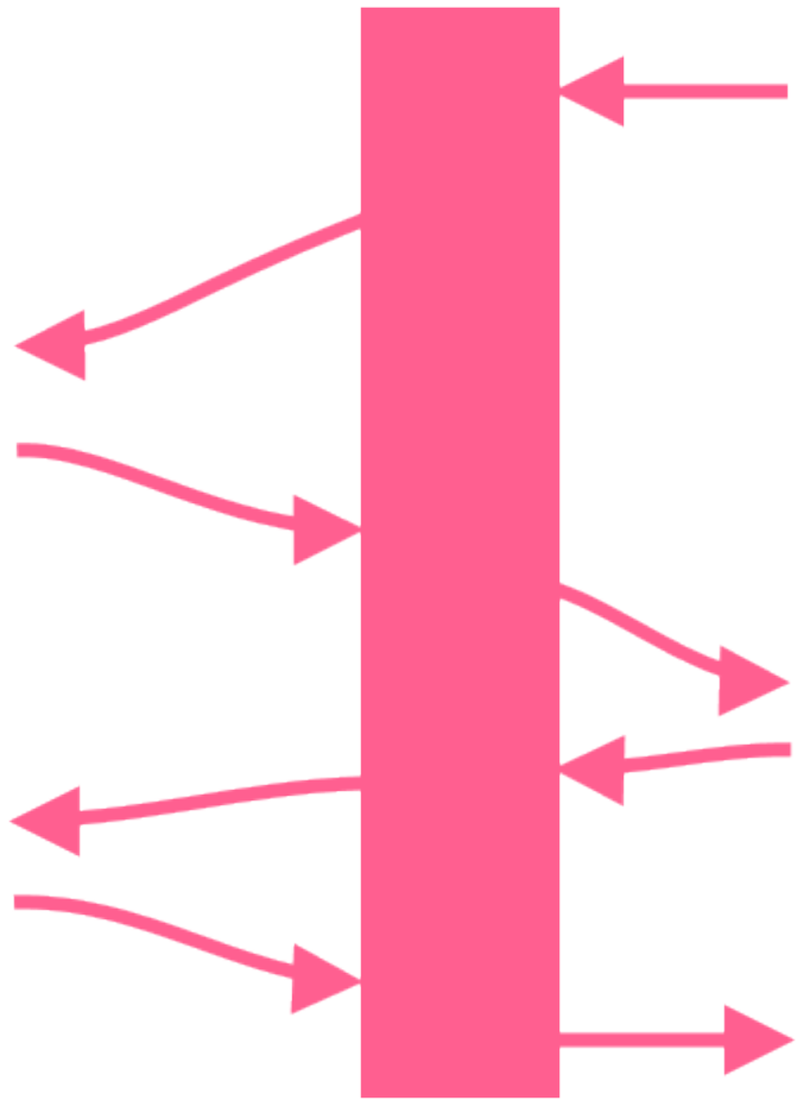
- Complex as other programming languages

  ⇒ Requires testing

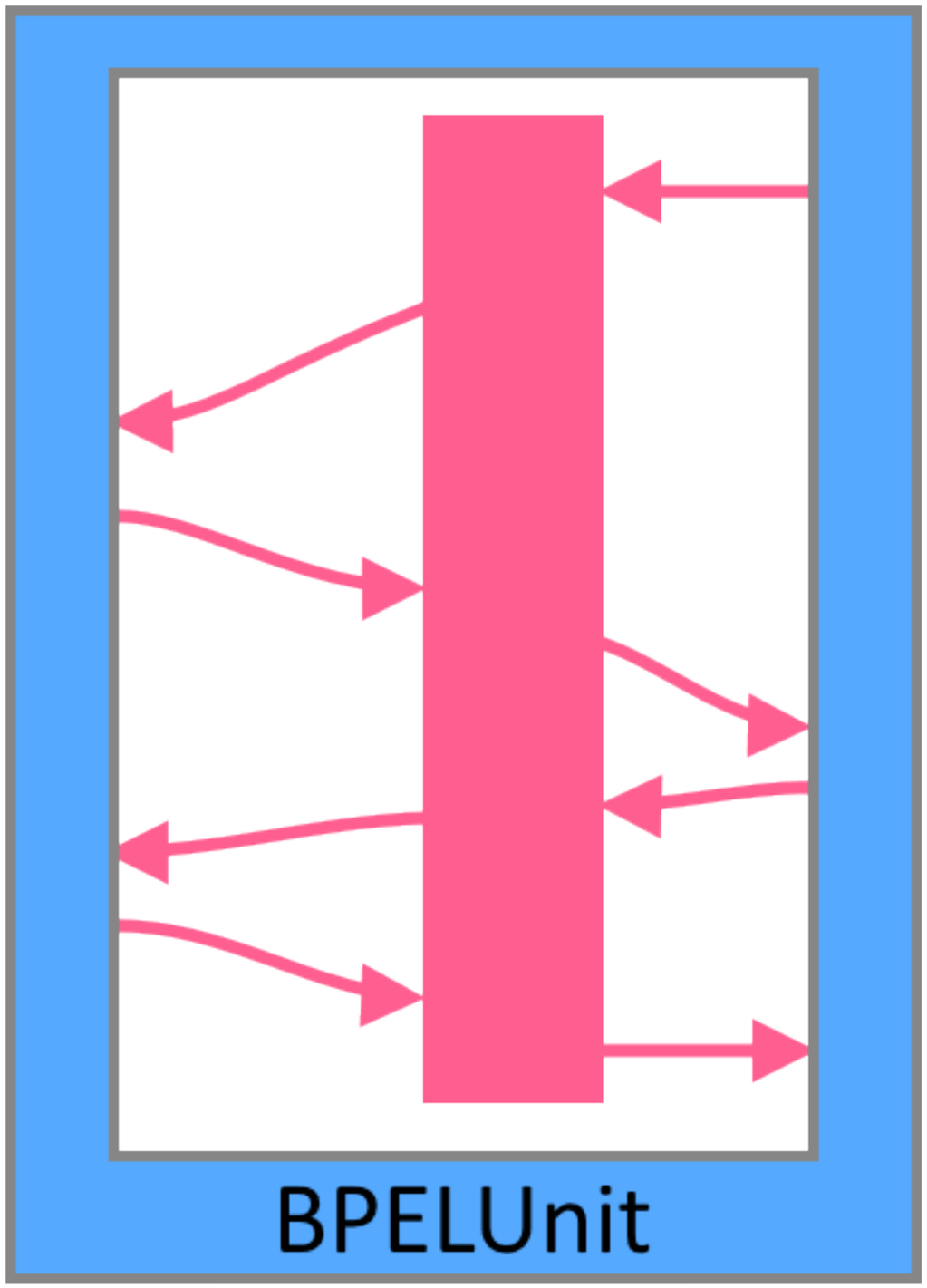
- WSDL: Clear interfaces

  ⇒ Convenient for testing

Testing w/o real services



BPELUnit



Message syntax

```
<tes:receiveSend service="tns:ApprovalService" port="ApprovalPort"
operation="approveLoan">
    <tes:send fault="false">
    </tes:send>
    <tes:receive fault="false">
    </tes:receive>
</tes:receiveSend>
```

BPELUnit - Send

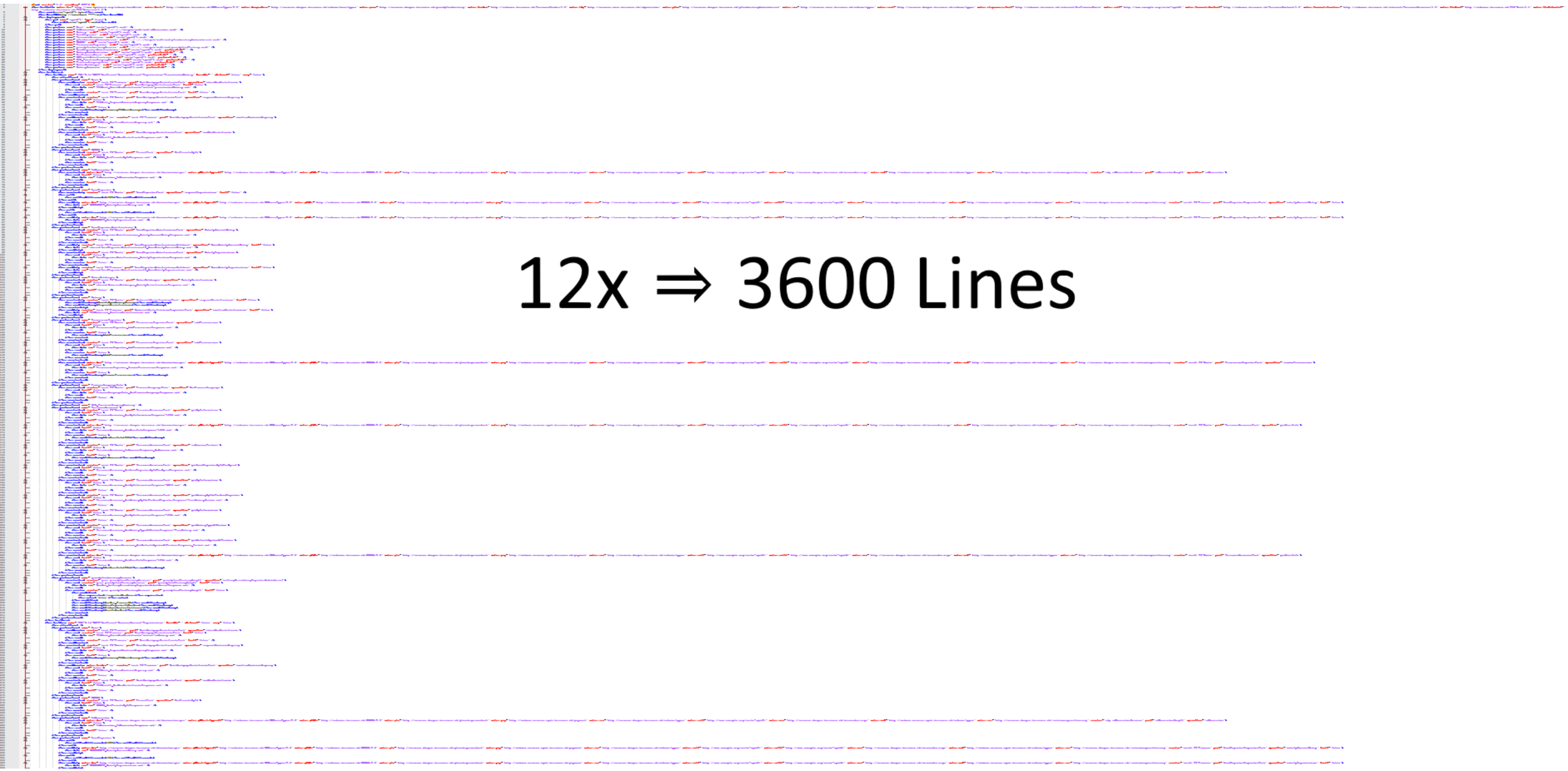
```
<tes:receiveSend service="tns:ApprovalService" port="ApprovalPort"
operation="approveLoan">
    <tes:send fault="false">
        <tes:data>
            <tns:approveLoanResponse>
                <tns:granted>true</tns:granted>
                <tns:customerID>10001</tns:customerID>
            </tns:approveLoanResponse>
        </tes:data>
    </tes:send>
    <tes:receive fault="false">
    </tes:receive>
</tes:receiveSend>
```

BPELUnit - Receive

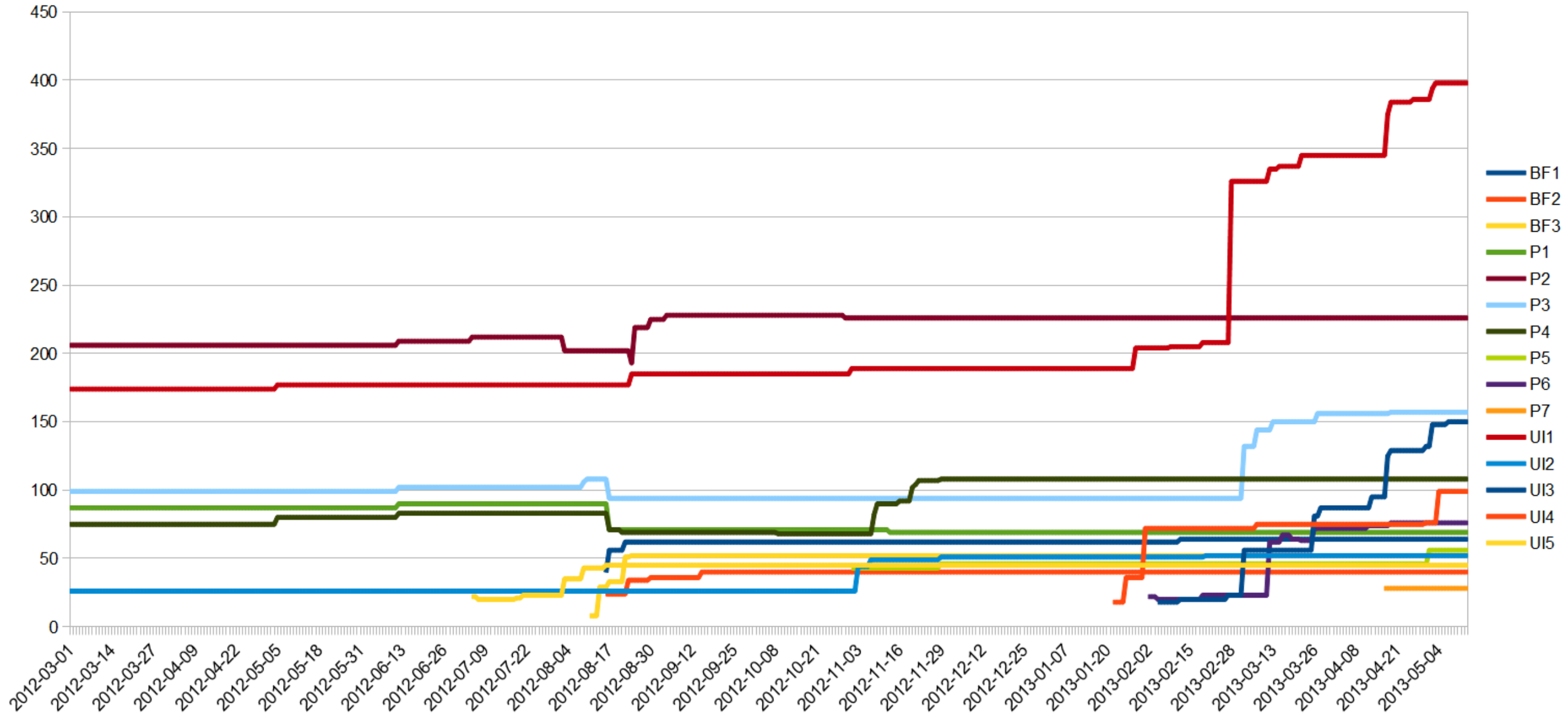
```
<tes:receiveSend service="tns:ApprovalService" port="ApprovalPort"
operation="approveLoan">
    <tes:send fault="false">
        <tes:data>
            <tns:approveLoanResponse>
                <tns:granted>true</tns:granted>
                <tns:customerID>10001</tns:customerID>
            </tns:approveLoanResponse>
        </tes:data>
    </tes:send>
    <tes:receive fault="false">
        <tes:condition>
            <tes:expression>//tns:amount</tes:expression>
            <tes:value>100000</tes:value>
        </tes:condition>
        <tes:condition>
            <tes:expression>//tns:numPDFs</tes:expression>
            <tes:value>1</tes:value>
        </tes:condition>
    </tes:receive>
</tes:receiveSend>
```

Potential for improvement?

Testsuite size



Activity count



Difficult to maintain

- Lots of semantically connected XML

Difficult to maintain

- Lots of semantically connected XML
- But: No syntactical connection

Difficult to maintain

- Lots of semantically connected XML
- But: No syntactical connection
  - Same message in different test cases

Difficult to maintain

- Lots of semantically connected XML
- But: No syntactical connection
  - Same message in different test cases
  - Testdata (e.g. userId)

Difficult to maintain

- Lots of semantically connected XML
- But: No syntactical connection
  - Same message in different test cases
  - Testdata (e.g. userId)
  - Similar messages to different partners

Difficult overview

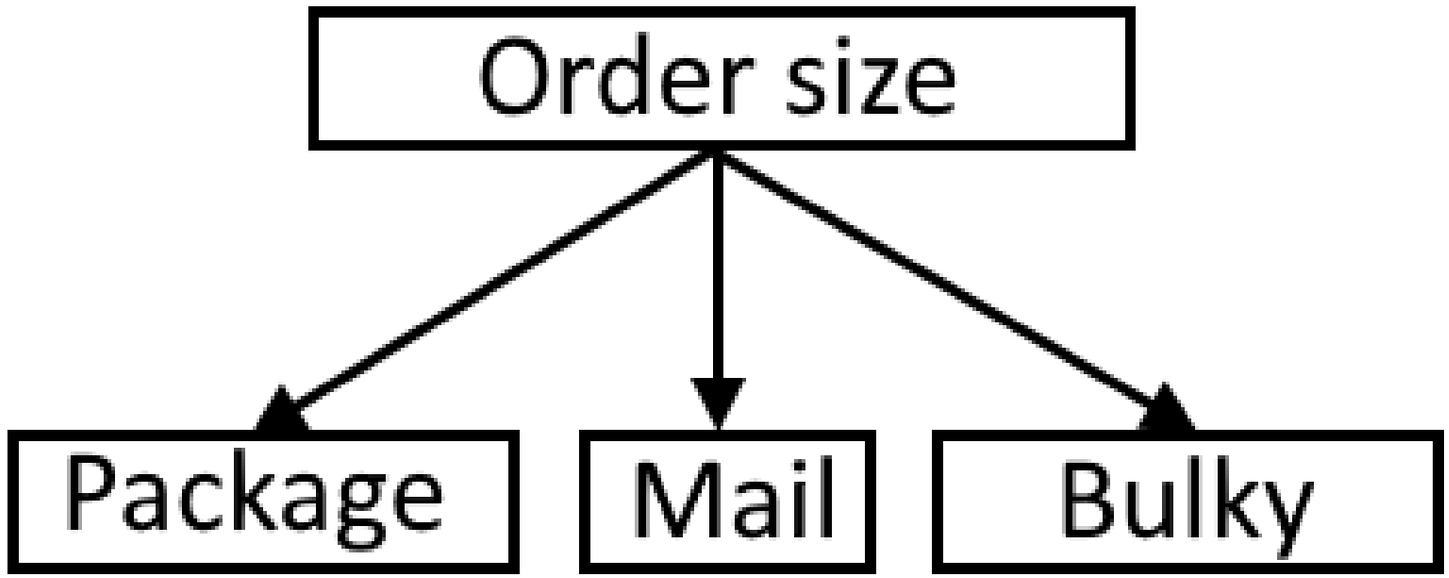
- Lots of code in many test cases

Difficult overview

- Lots of code in many test cases
- Complete requirements coverage?

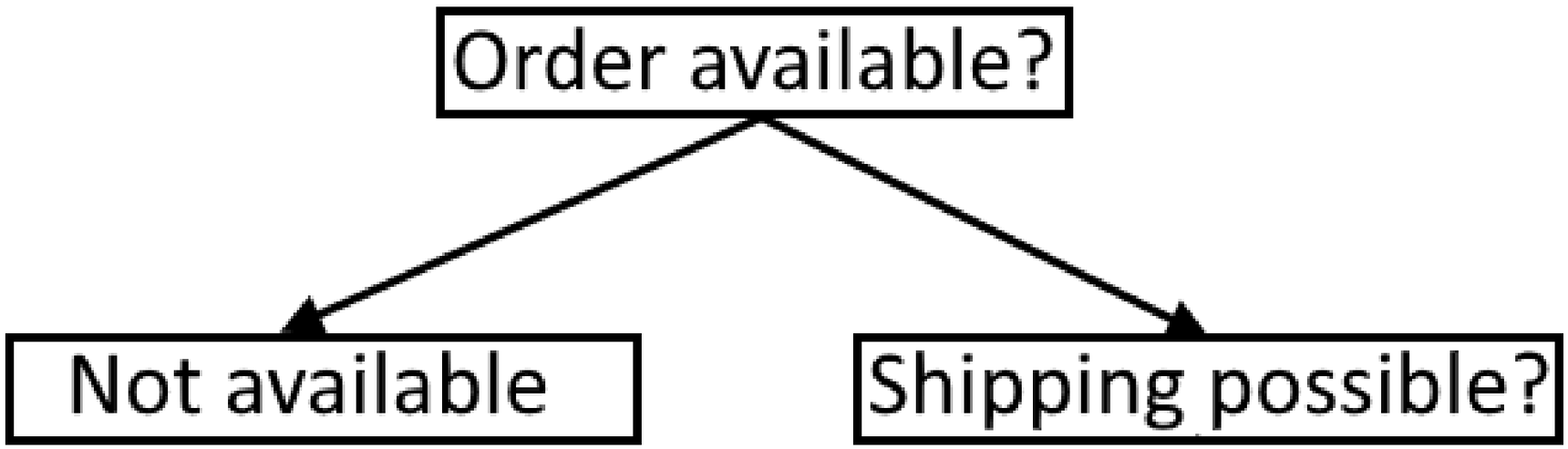
Creating an overview

- Partition values into categories



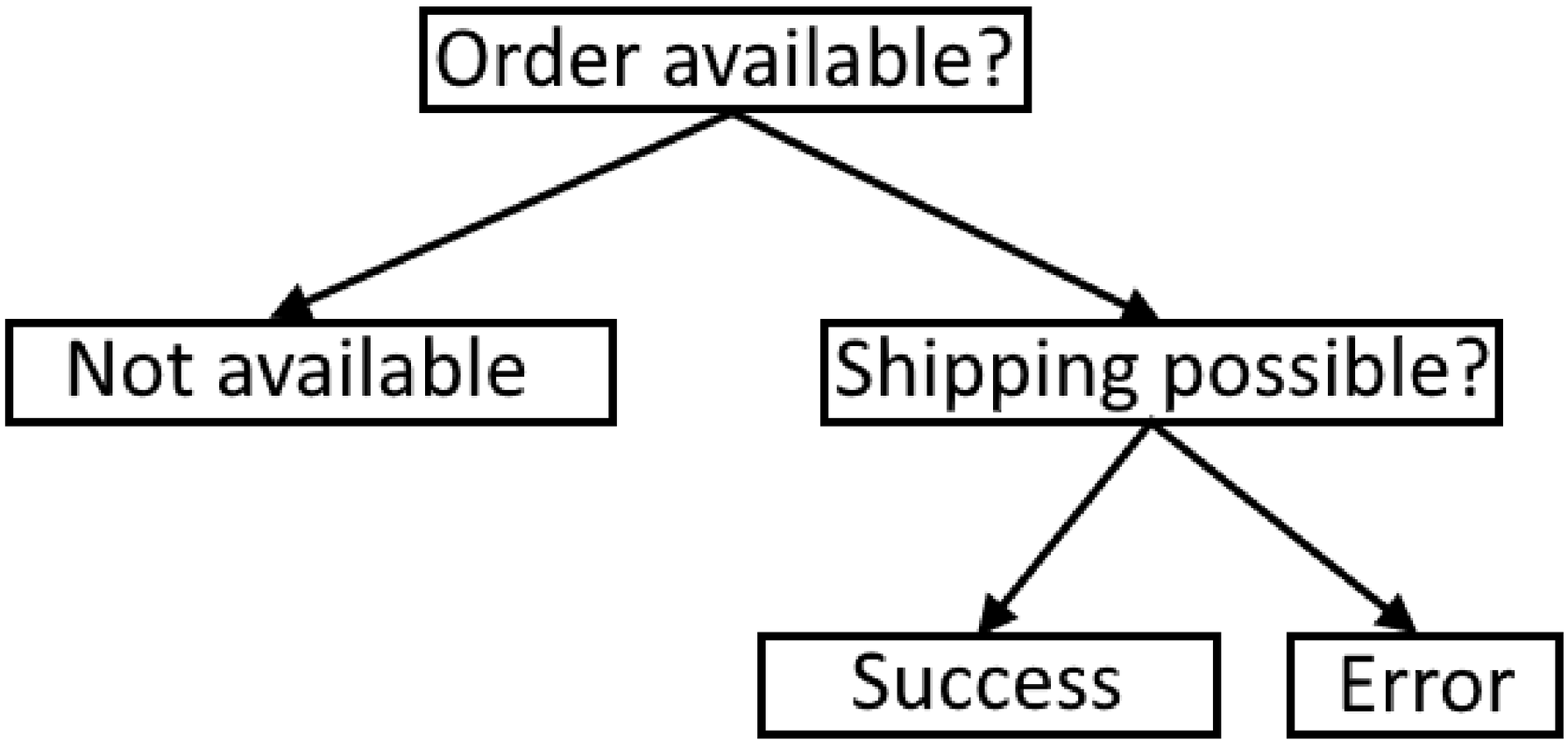
Classification tree

- Split categories into subcategories

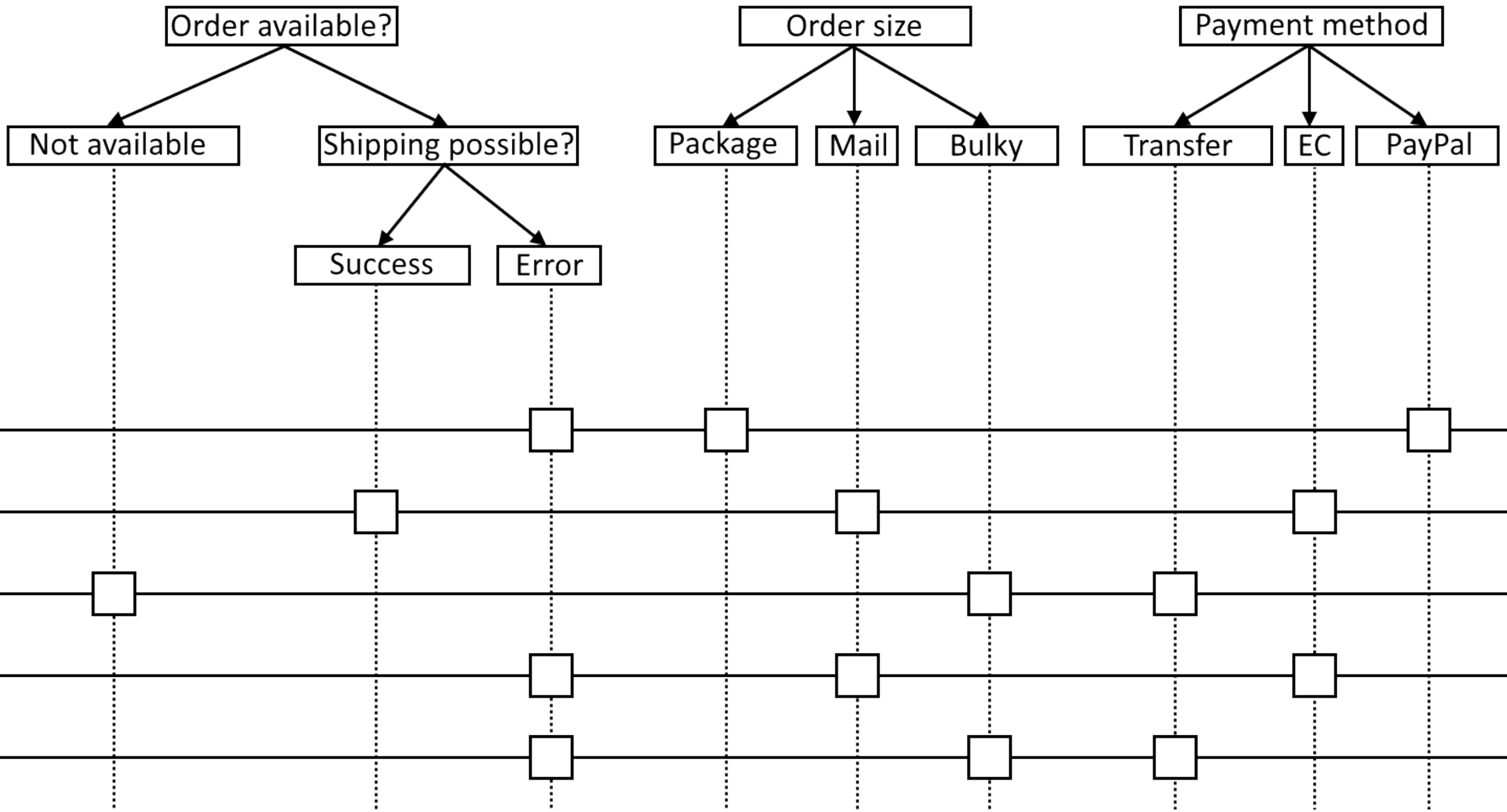


Classification tree

- Split categories into subcategories



Test cases

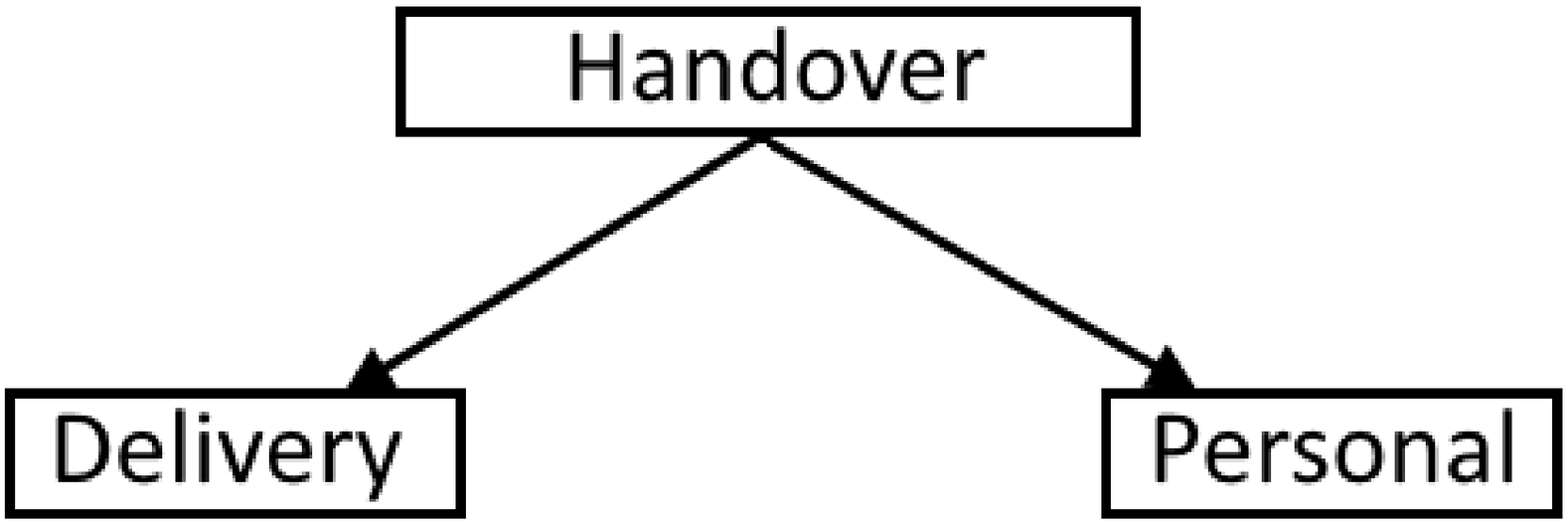

Order available?
Not available
Shipping possible?
Success
Error
Order size
Package
Mail
Bulky
Payment method
Transfer
EC
PayPal

Conditions

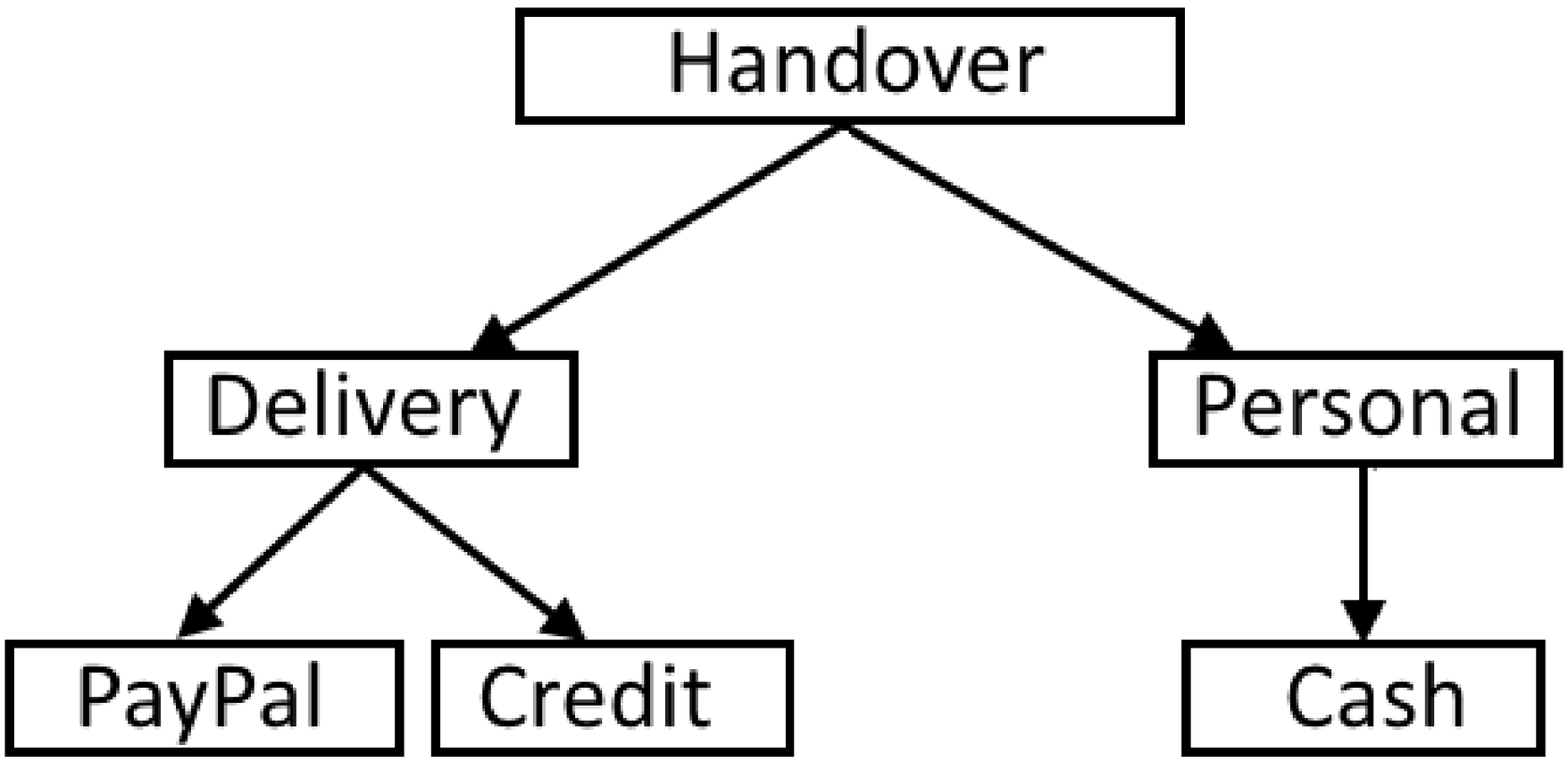
- Prohibit certain value combinations

handover:delivery AND payment:cash

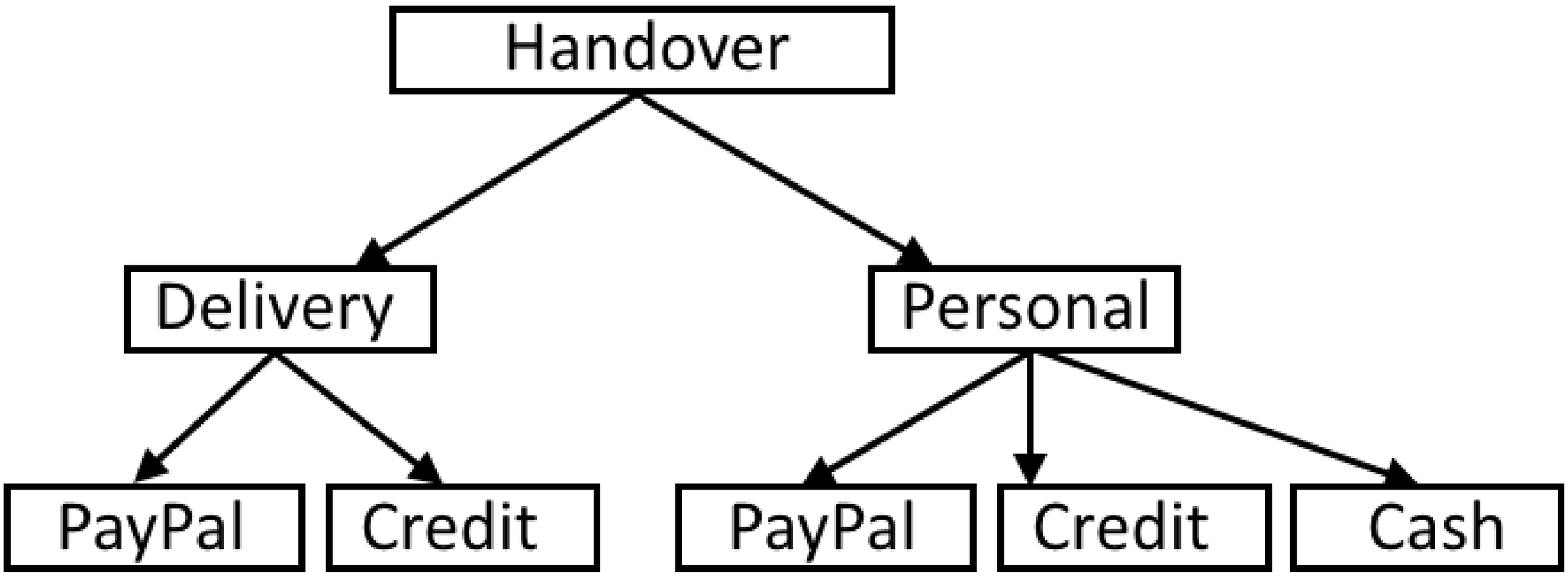
Conditions – By categories



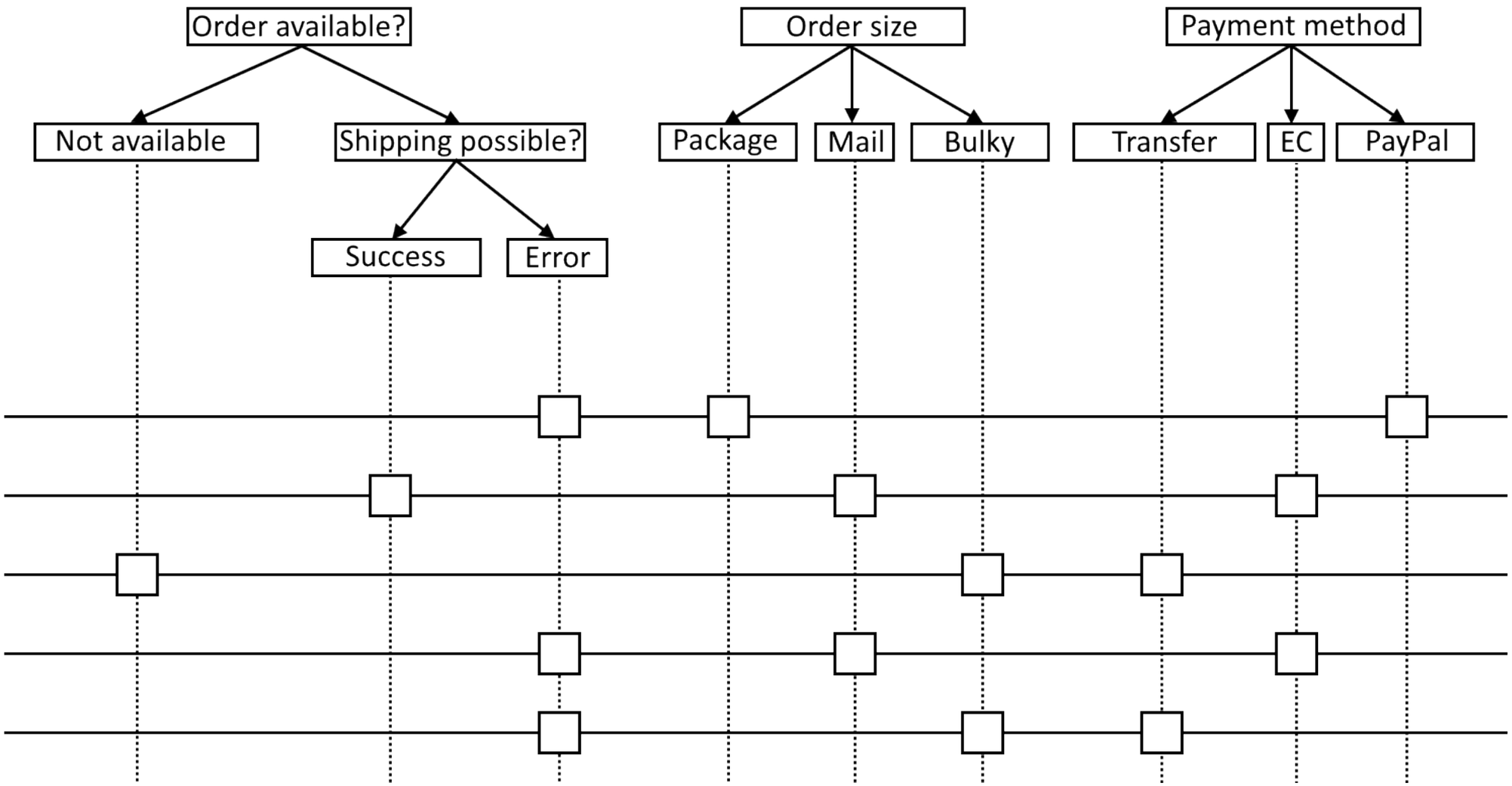
Conditions – By categories



Conditions – By categories



Classification



handover:delivery AND payment:cash

Problem: Relevance

- Classification is worthless if not updated

Problem: Relevance

- Classification is worthless if not updated
- Often this is neglected

Problem: Relevance

- Classification is worthless if not updated
- Often this is neglected

⇒ Enforce updates through advantages

Solution: Generation

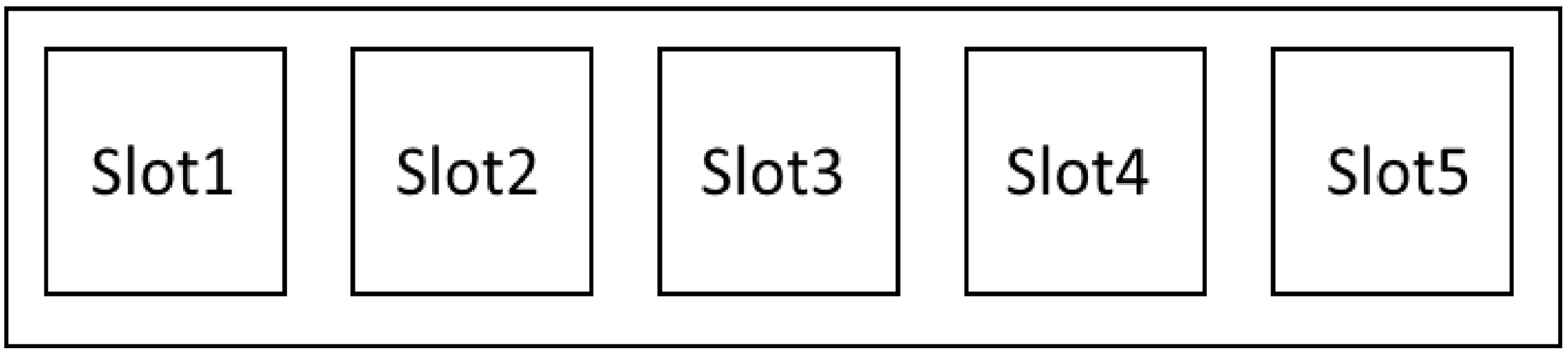
- Generate test suites from the classification

Solution: Generation

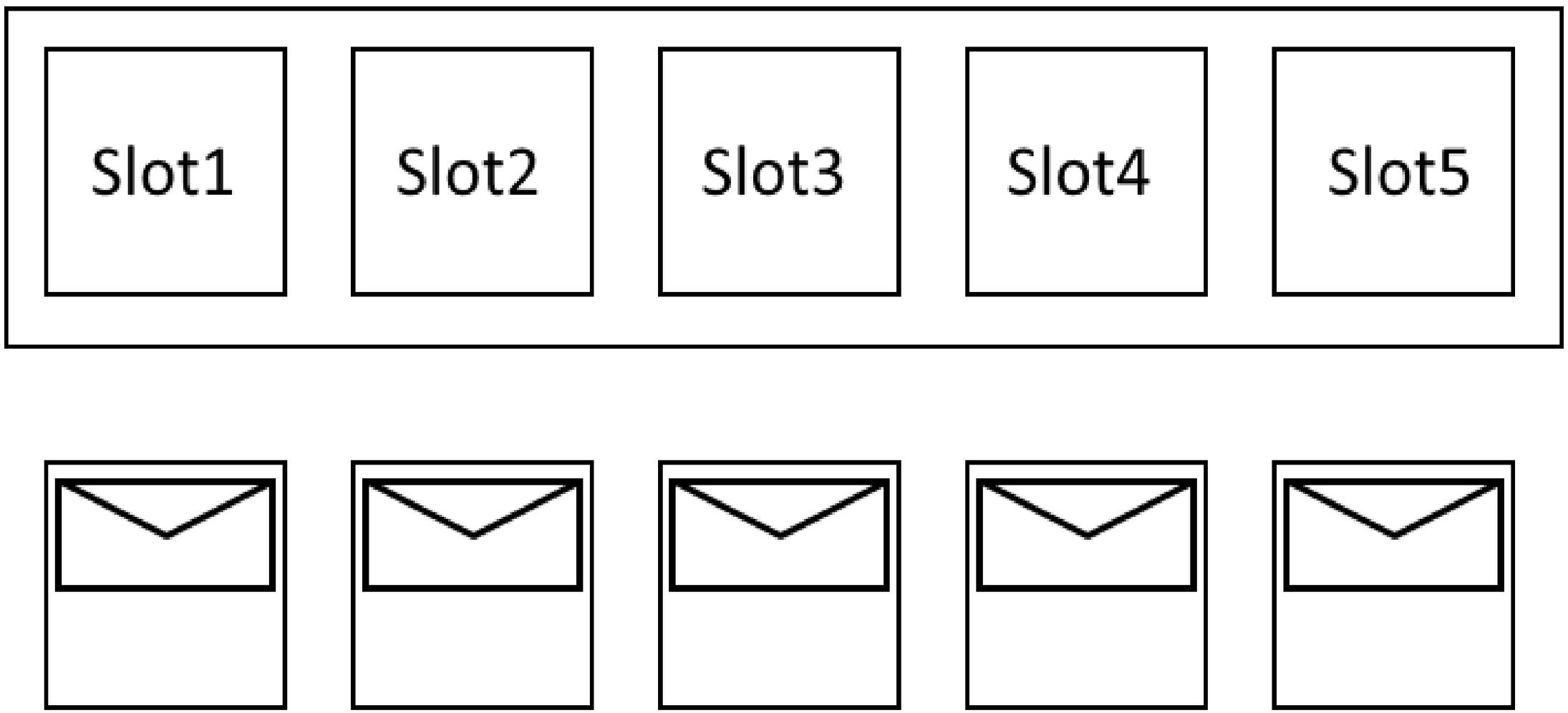
- Generate test suites from the classification
- Needs test fragments
  - Technical information
  - Example data

Test fragments

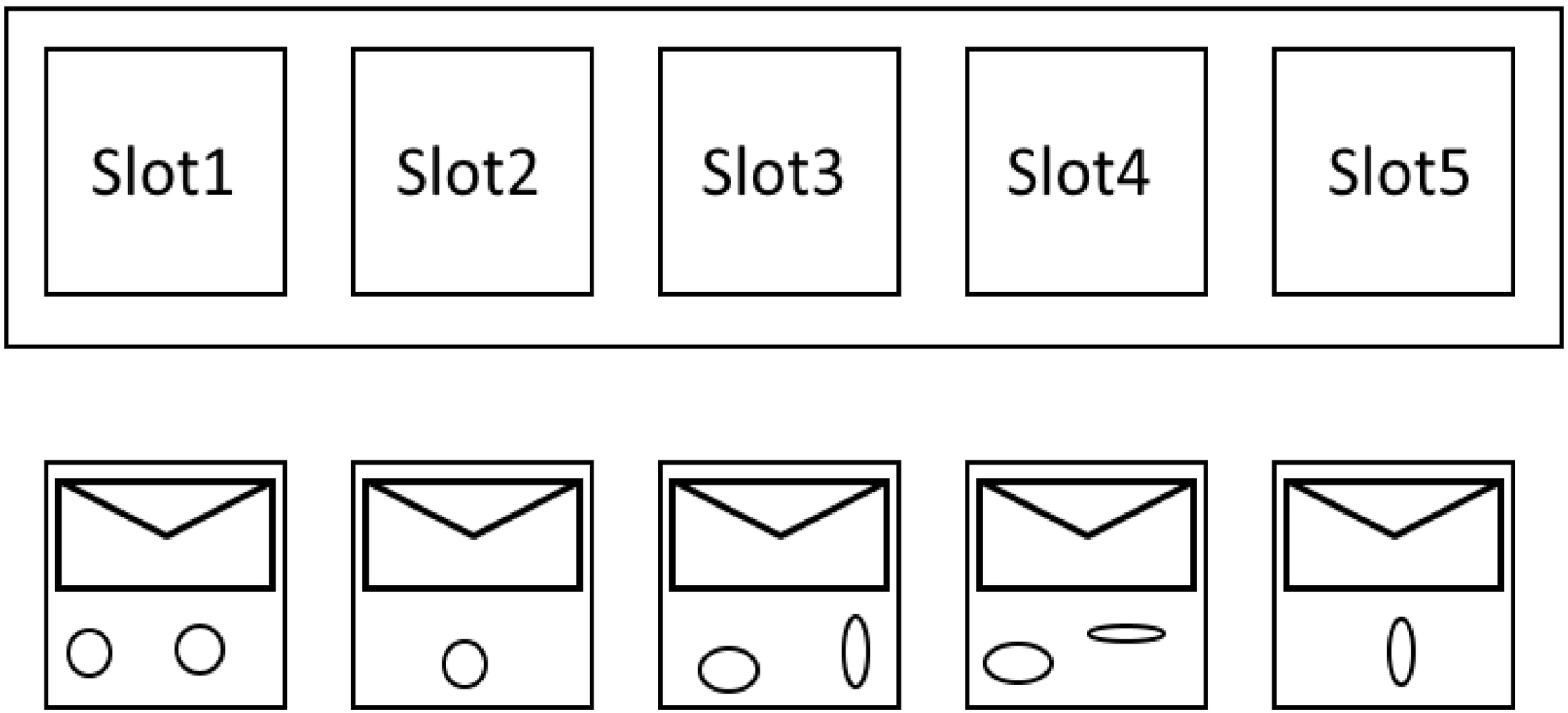
Test fragments



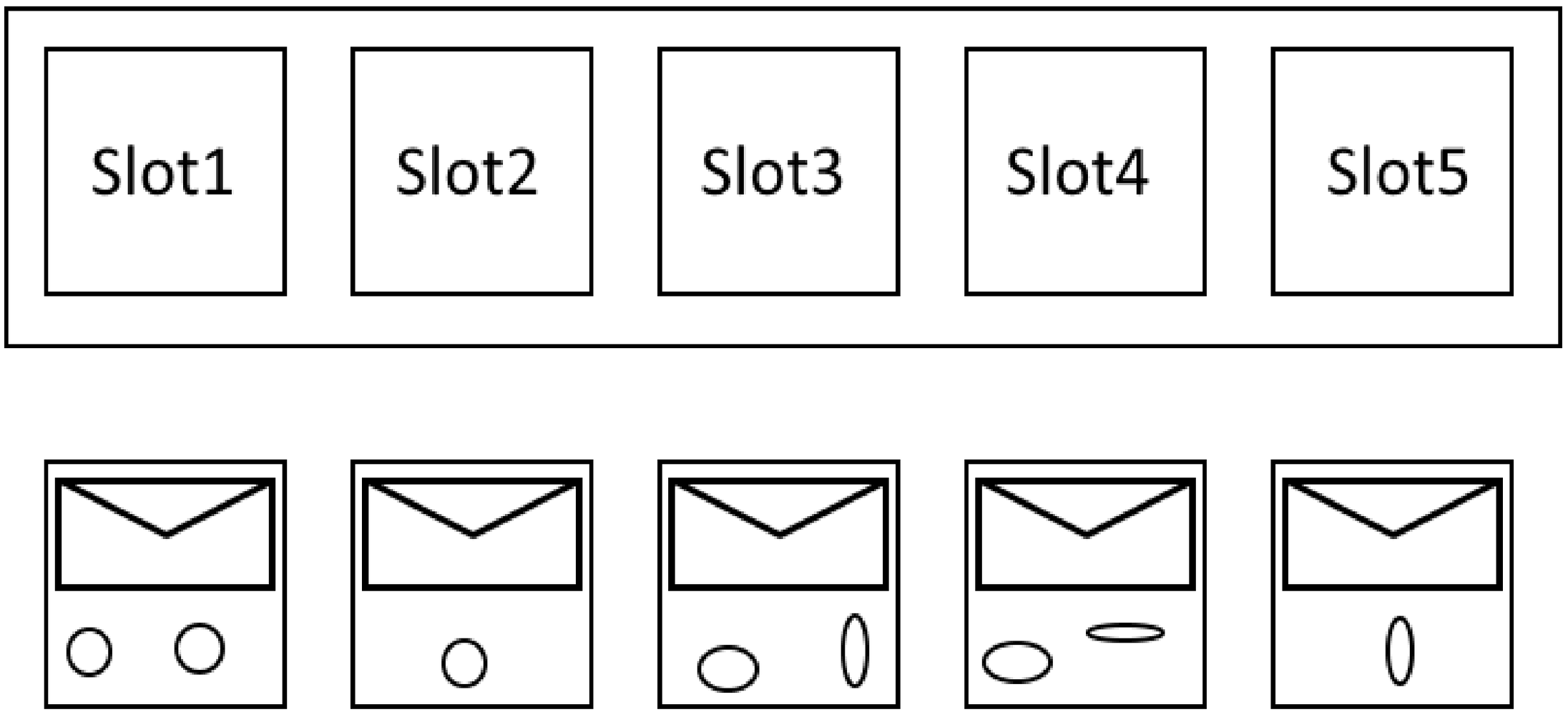
Test fragments



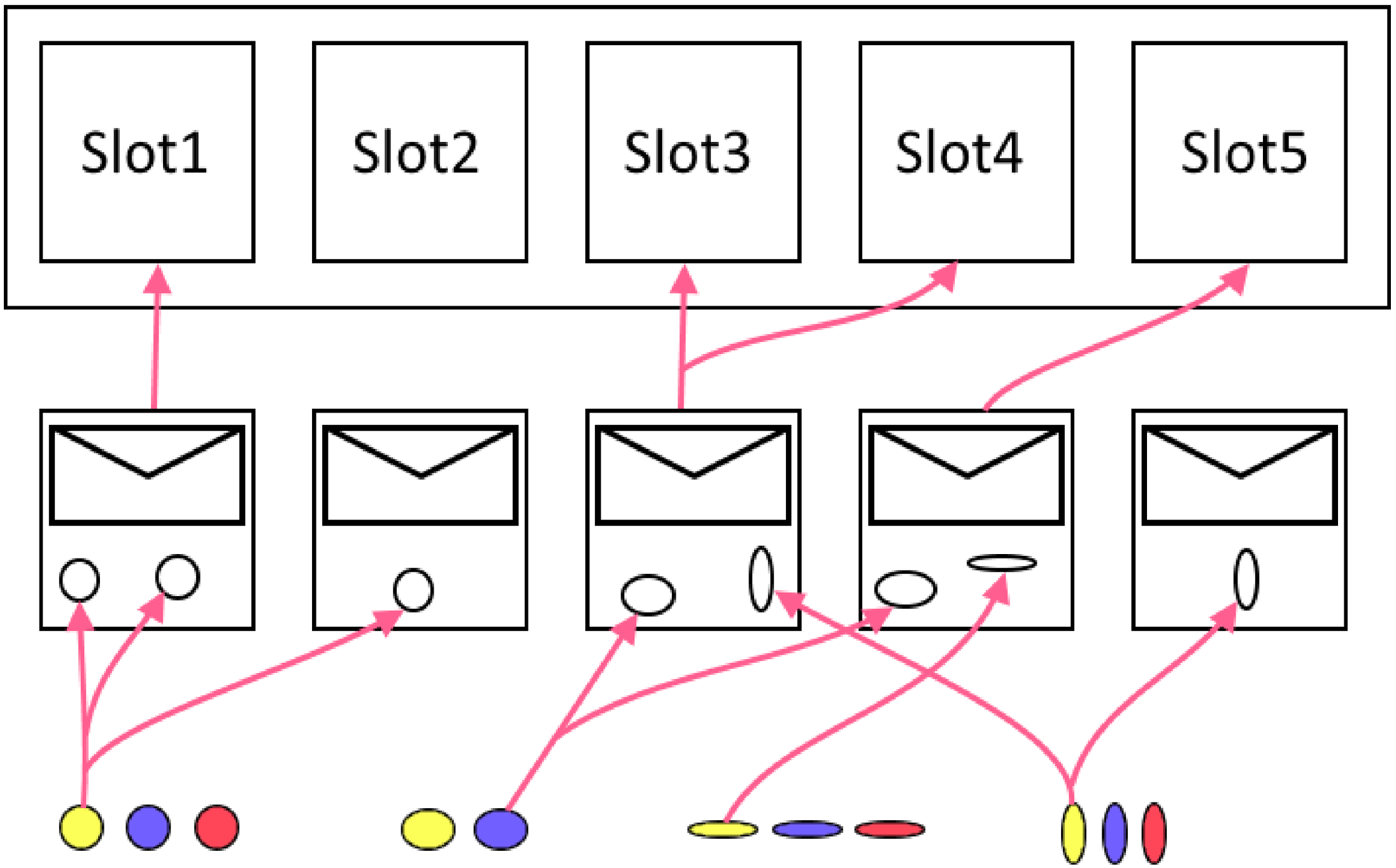
Test fragments



Test fragments



Test fragments



Implications of changes

- Test fragments
- Classification

Implications of changes

- Test fragments
  - Change syntax of a message once
- Classification

Implications of changes

- Test fragments
  - Change syntax of a message once
- Classification
  - New mappings
  - Change mappings
  - Create new test fragments

Recommendations

- MinMax-principle
  - Best case: **w** test cases
    - **w**: number of values in biggest category

Recommendations

- MinMax-principle
  - Best case: **w** test cases
    - **w**: number of values in biggest category
- Based on variables
  - Every instance should be used at least once

Automated definition

- Generator defines new test cases itself
  - Using aforementioned recommendations

Automated definition

- Generator defines new test cases itself
  - Using aforementioned recommendations
    - Combinatorial test design
      - Using all **n**-way value combinations
        - Min. $\mathbf{w}^{\mathbf{n}}$ test cases
          - **w**: Number of values in the biggest category

Validation

Validation - Students

- Experiment with a group of students
  - Overview over functional test coverage increased

Validation - Students

- Experiment with a group of students
  - Overview over functional test coverage increased
  - But also slow down
    - Remembering too many element names

Validation - Students

- Experiment with a group of students
  - Overview over functional test coverage increased
  - But also slow down
    - Remembering too many element names
    - Difficult to recognize fragment connections

Validation - Students

- Experiment with a group of students
  - Overview over functional test coverage increased
  - But also slow down
    - Remembering too many element names
    - Difficult to recognize fragment connections

  ⇒ A GUI would help

Validation - Rw-Process

- Reimplementation of an industry process

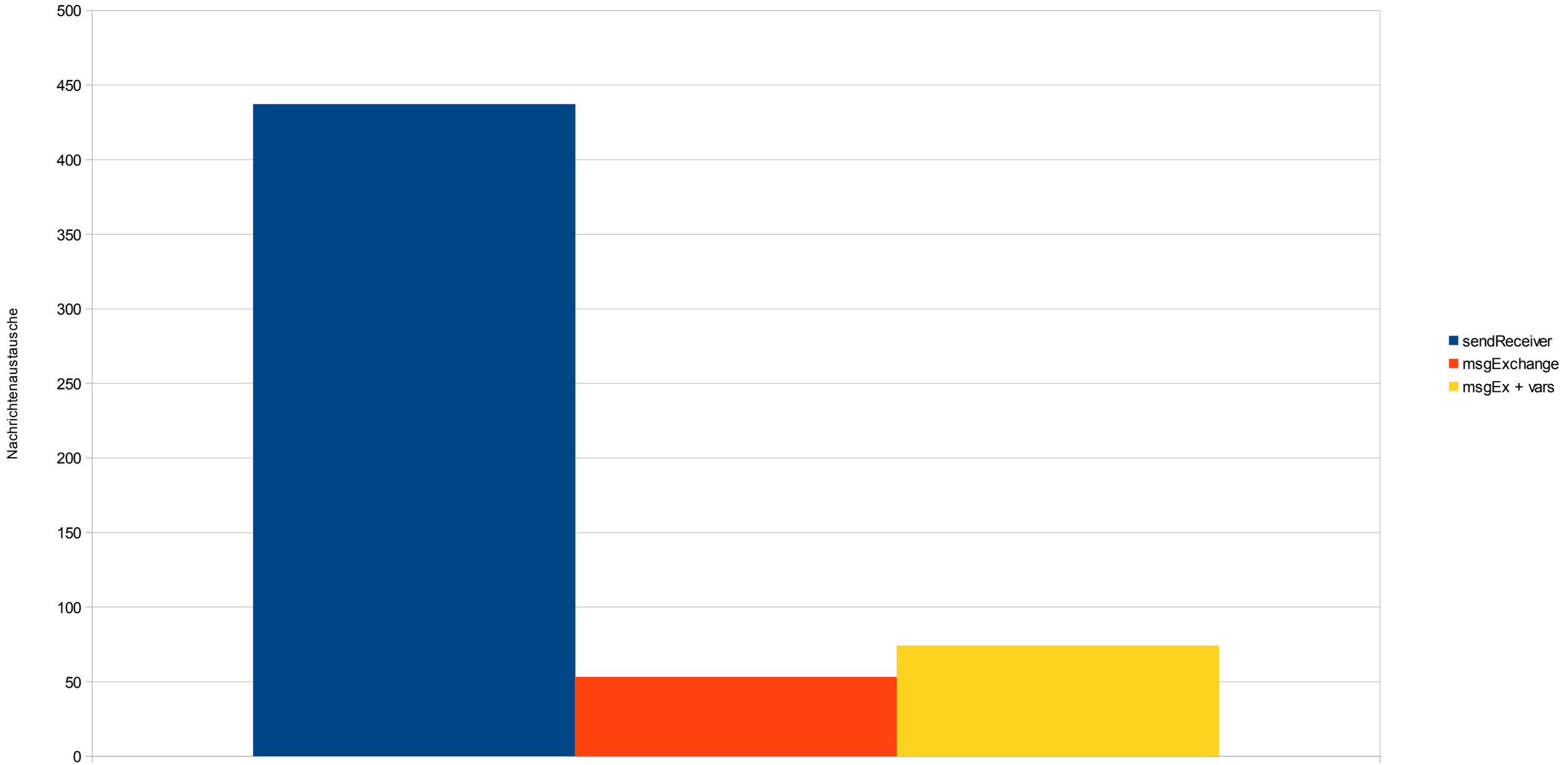
Validation - Rw-Process

- Reimplementation of an industry process
- Measurement criteria
  - Number of message definitions: Places of change

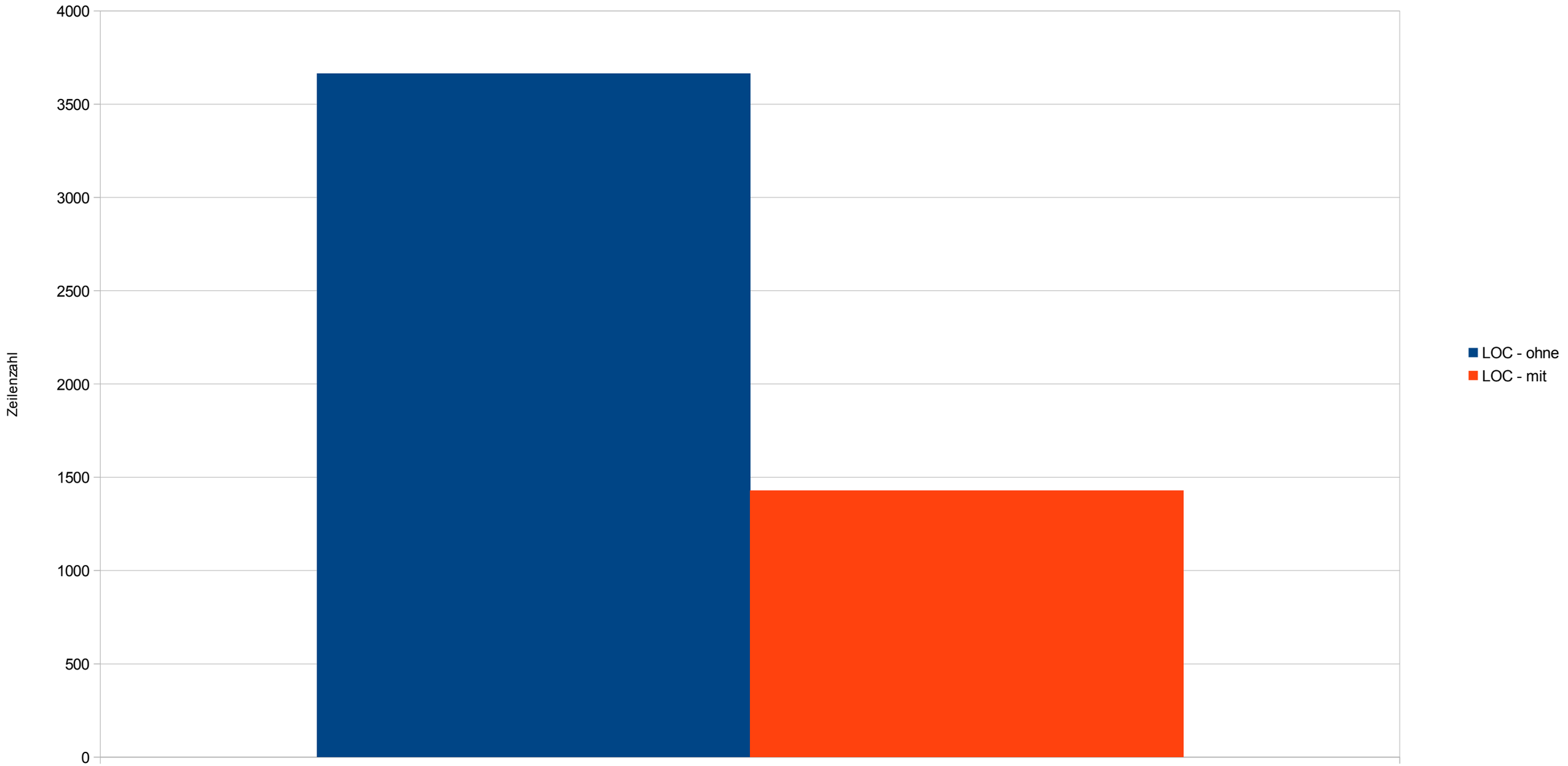
Validation - Rw-Process

- Reimplementation of an industry process
- Measurement criteria
  - Number of message definitions: Places of change
  - Lines of codes: General complexity indicator

Rw-Process - Definitions



Rw-Process – Lines of Code



Summary

- The generator:
  - Increases overview over functional test coverage and supports maintenance
  - Generates test suites from classification and test fragments
  - Successfully rebuilds real world test suites

Thank you!

Thilo Schnelle

thilo.schnelle@innoq.com

Dr. Daniel Lübke

daniel.luebke@innoq.com

@dluebke

http://www.innoq.com