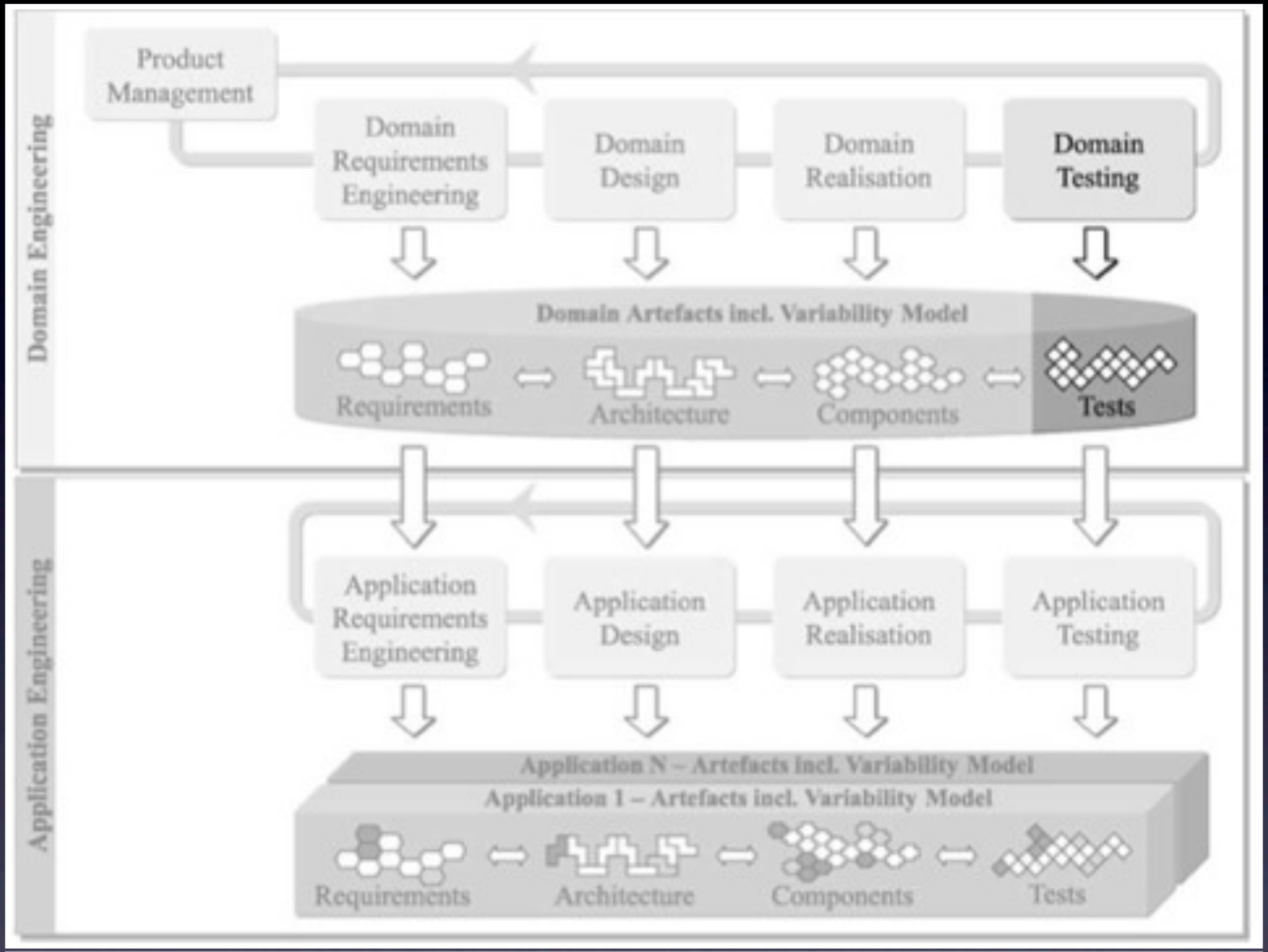
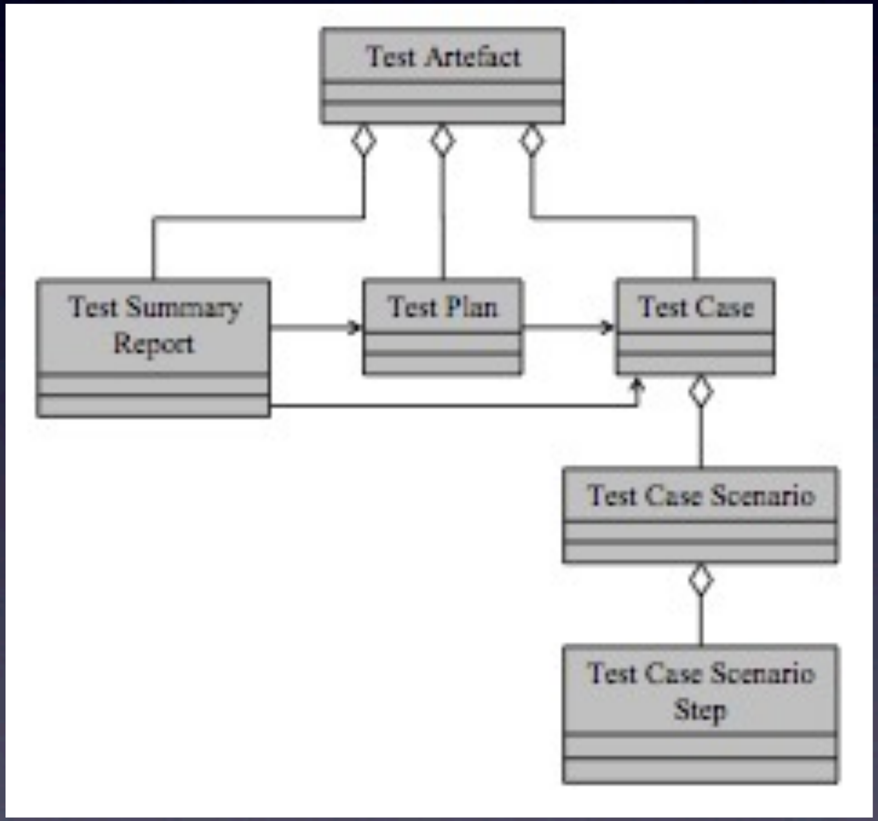
Testing in SPL

richard.torkar@gmail.com


Domain Engineering
Product Management
Domain Requirements Engineering
Domain Design
Domain Realisation
Domain Testing
Domain Artefacts incl. Variability Model
Requirements
Architecture
Components
Tests
Application Engineering
Application Requirements Engineering
Application Design
Application Realisation
Application Testing
Application N – Artefacts incl. Variability Model
Application 1 – Artefacts incl. Variability Model
Requirements
Architecture
Components
Tests

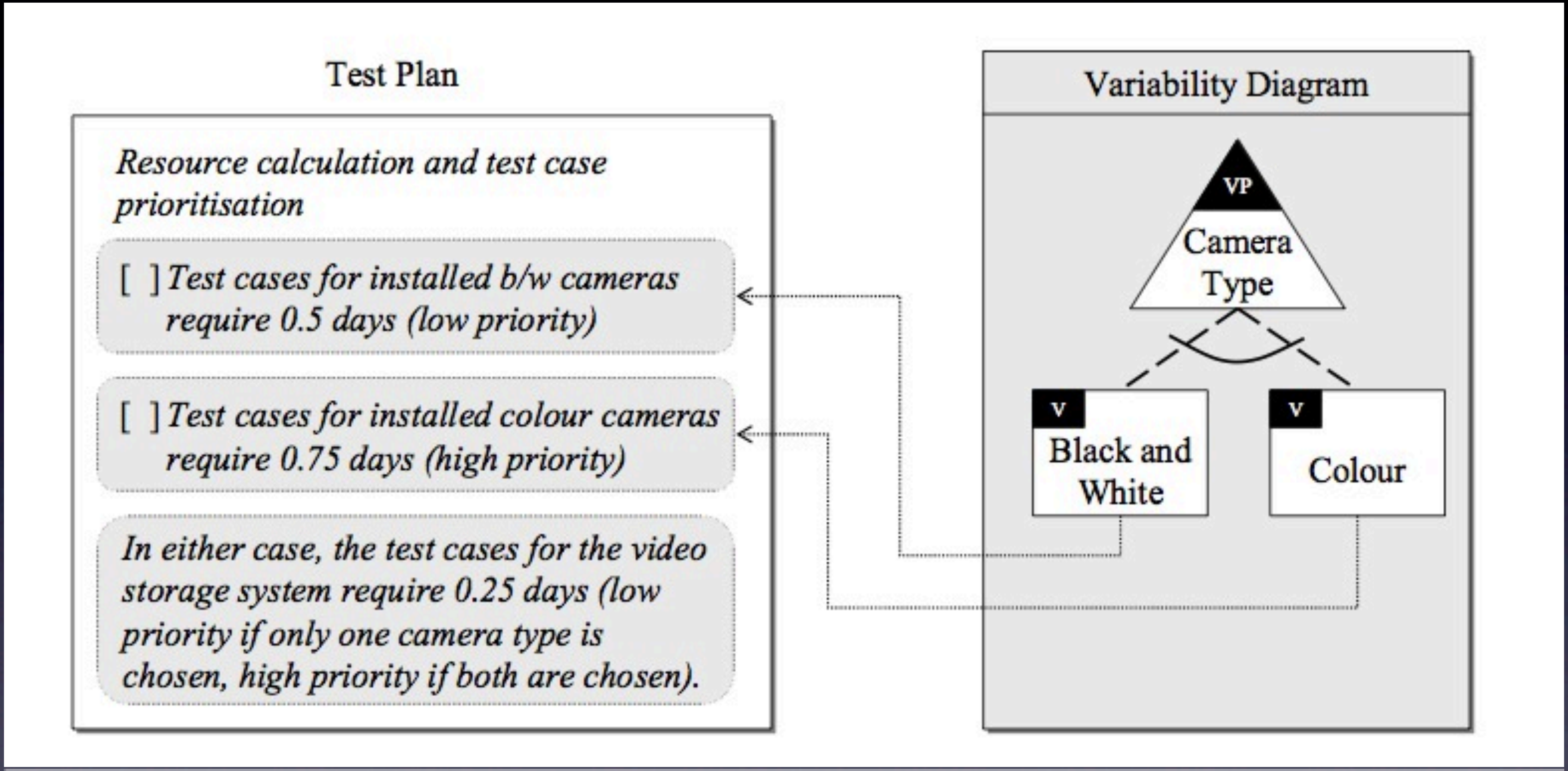
Some definitions

- Taken from IEEE STD 829-1998
    - Test plan
    - Test case
    - Test case scenario
    - Test summary report



Test plan

- Resource consumption/allocation
- Common and variable test cases to perform
- Priorities (of tc)
- Tool support


Test Plan
Resource calculation and test case prioritisation
[ ] Test cases for installed b/w cameras require 0.5 days (low priority)
[ ] Test cases for installed colour cameras require 0.75 days (high priority)
In either case, the test cases for the video storage system require 0.25 days (low priority if only one camera type is chosen, high priority if both are chosen).
Variability Diagram
VP
Camera Type
V
Black and White
V
Colour

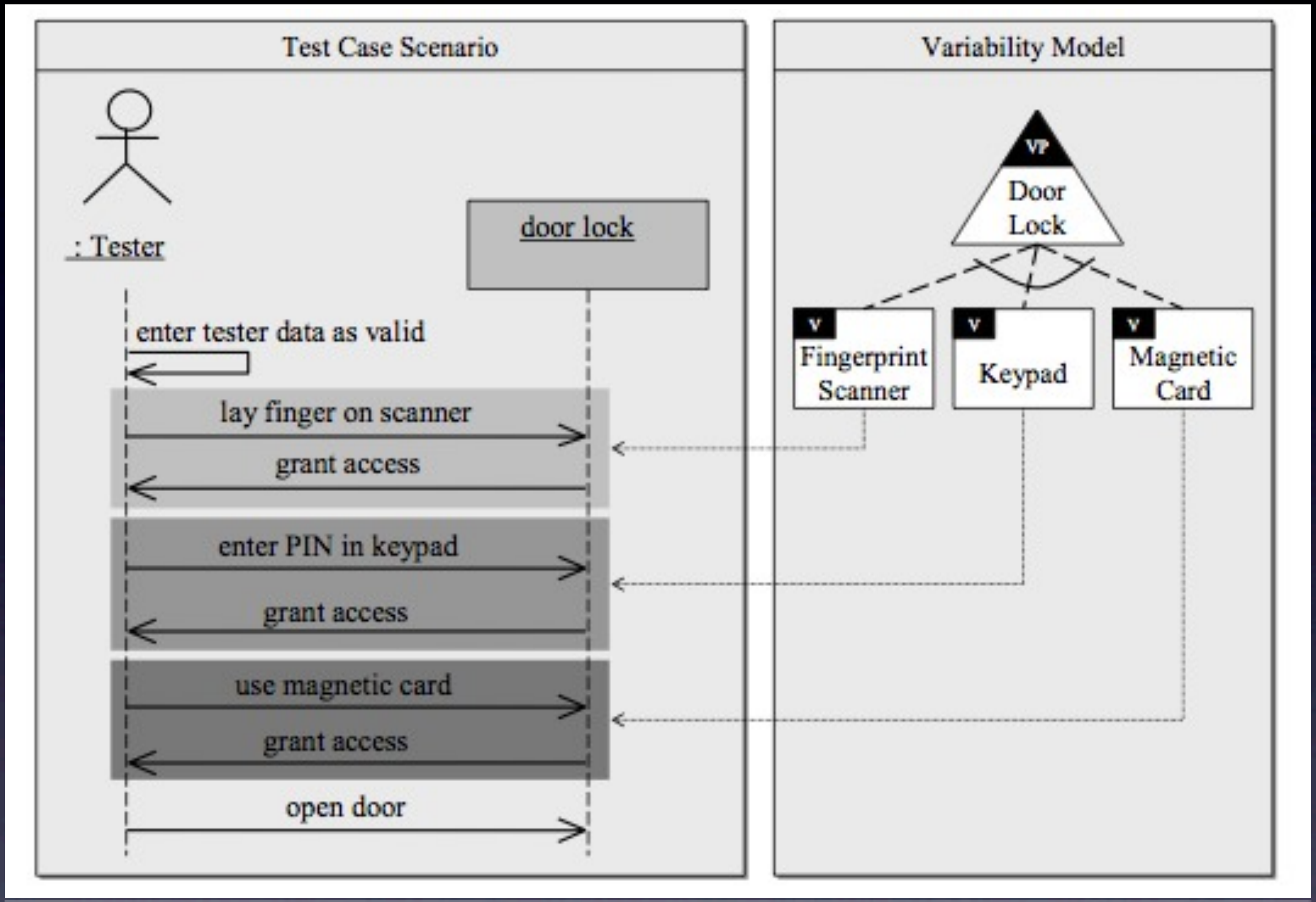
Test plan example

Test case

- Test data
- Test environment
- Fail-pass criteria
- Written in structured natural language

Test case scenario

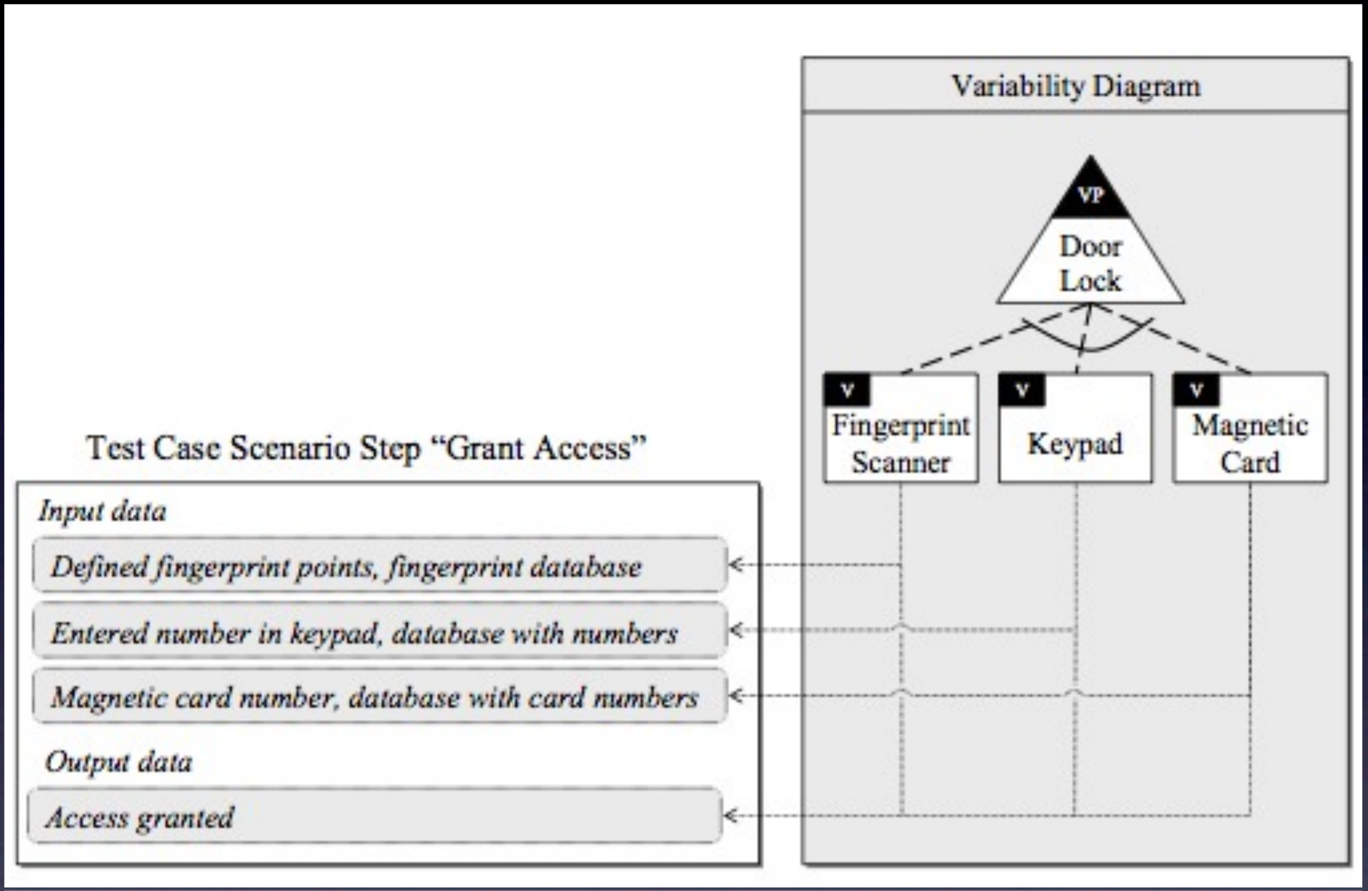
- Three types
  - Common to all intended apps
  - Specific for one variant
  - Specific for two or more variants



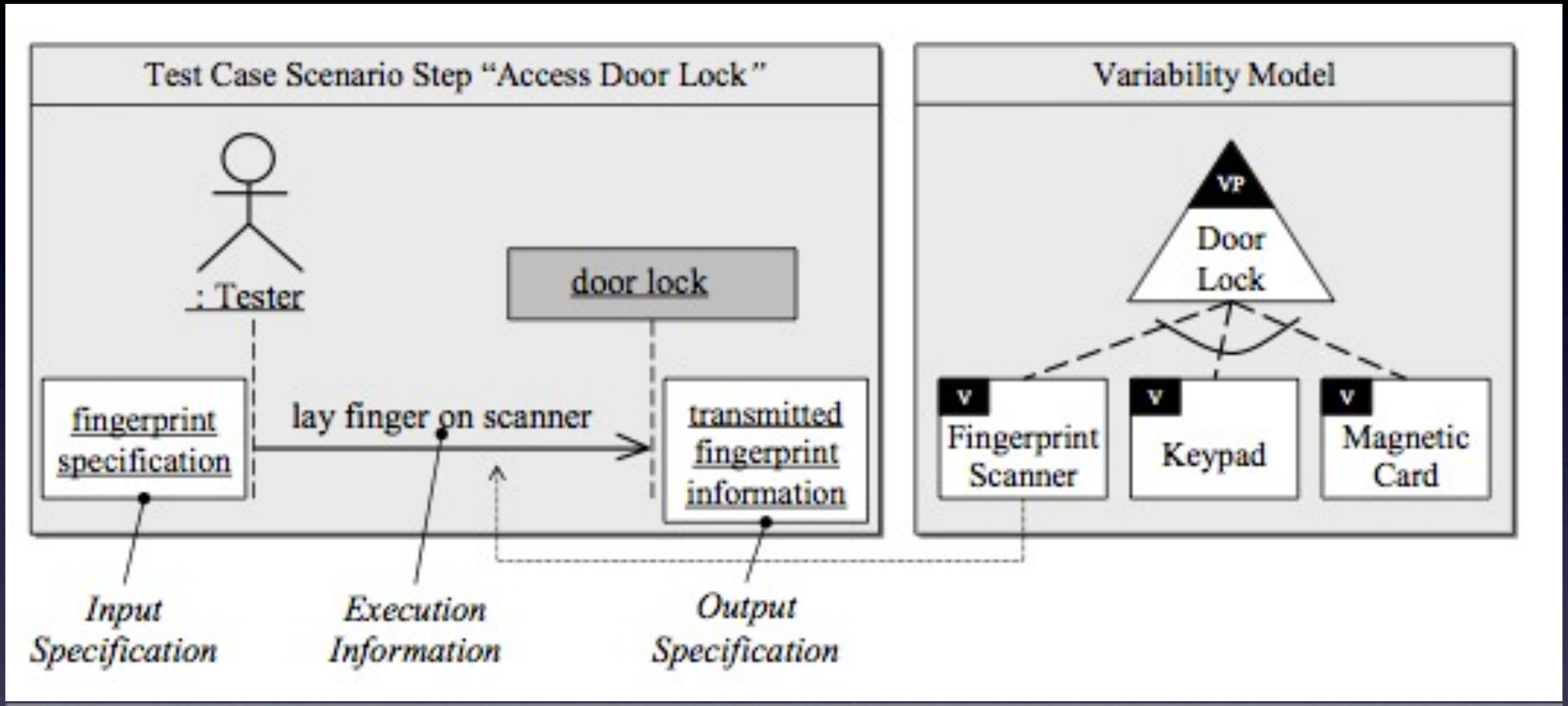
Test case scenario

Test case scenario step

- Input (specification)
- Output (specification)
- Execution info (guidance on how to perform the step(s))


Variability Diagram
VP
Door Lock
V
Fingerprint Scanner
V
Keypad
V
Magnetic Card
Test Case Scenario Step "Grant Access"
Input data
Defined fingerprint points, fingerprint database
Entered number in keypad, database with numbers
Magnetic card number, database with card numbers
Output data
Access granted

TC scenario step



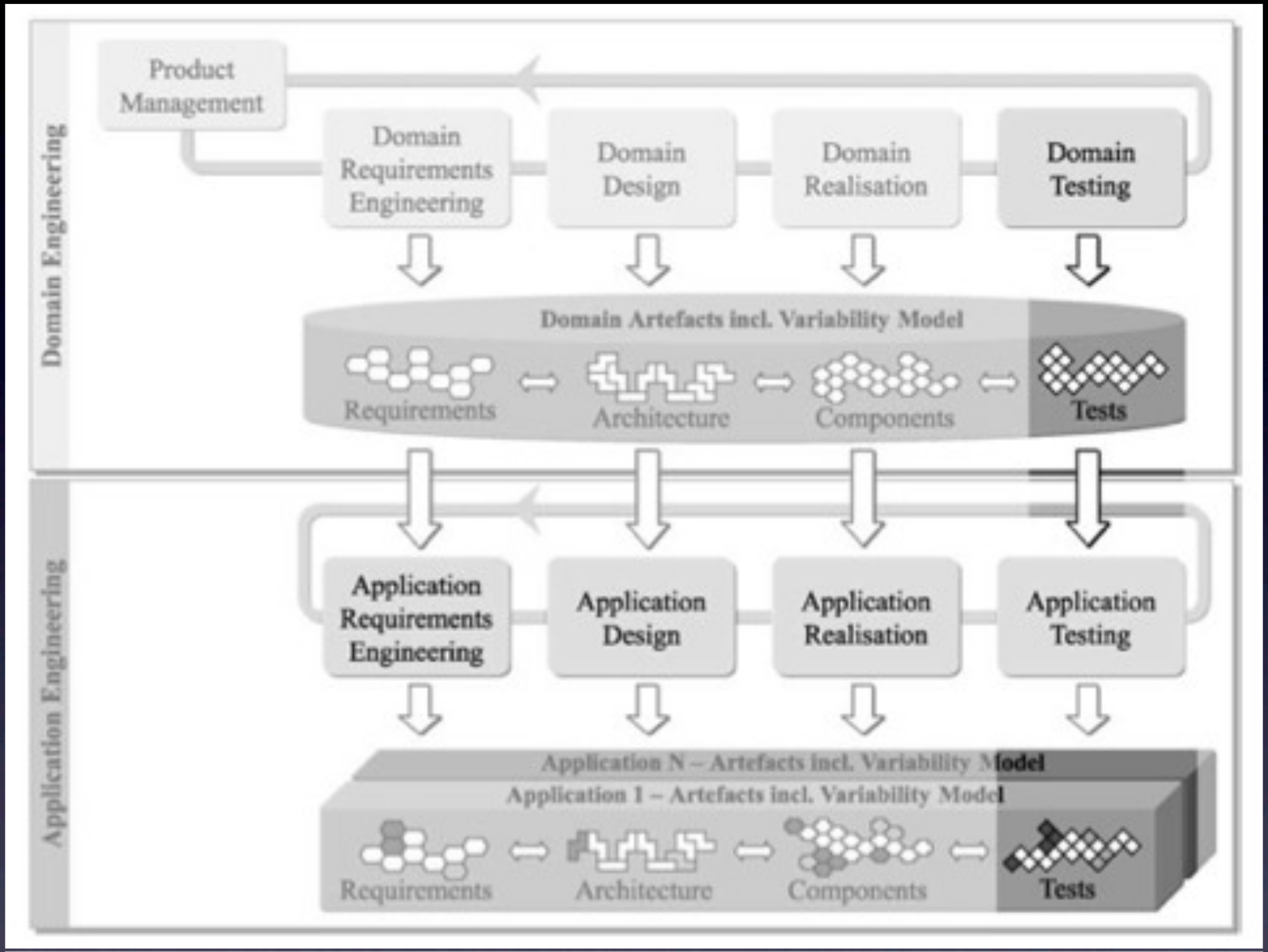
TC scenario step (sequence diagram)

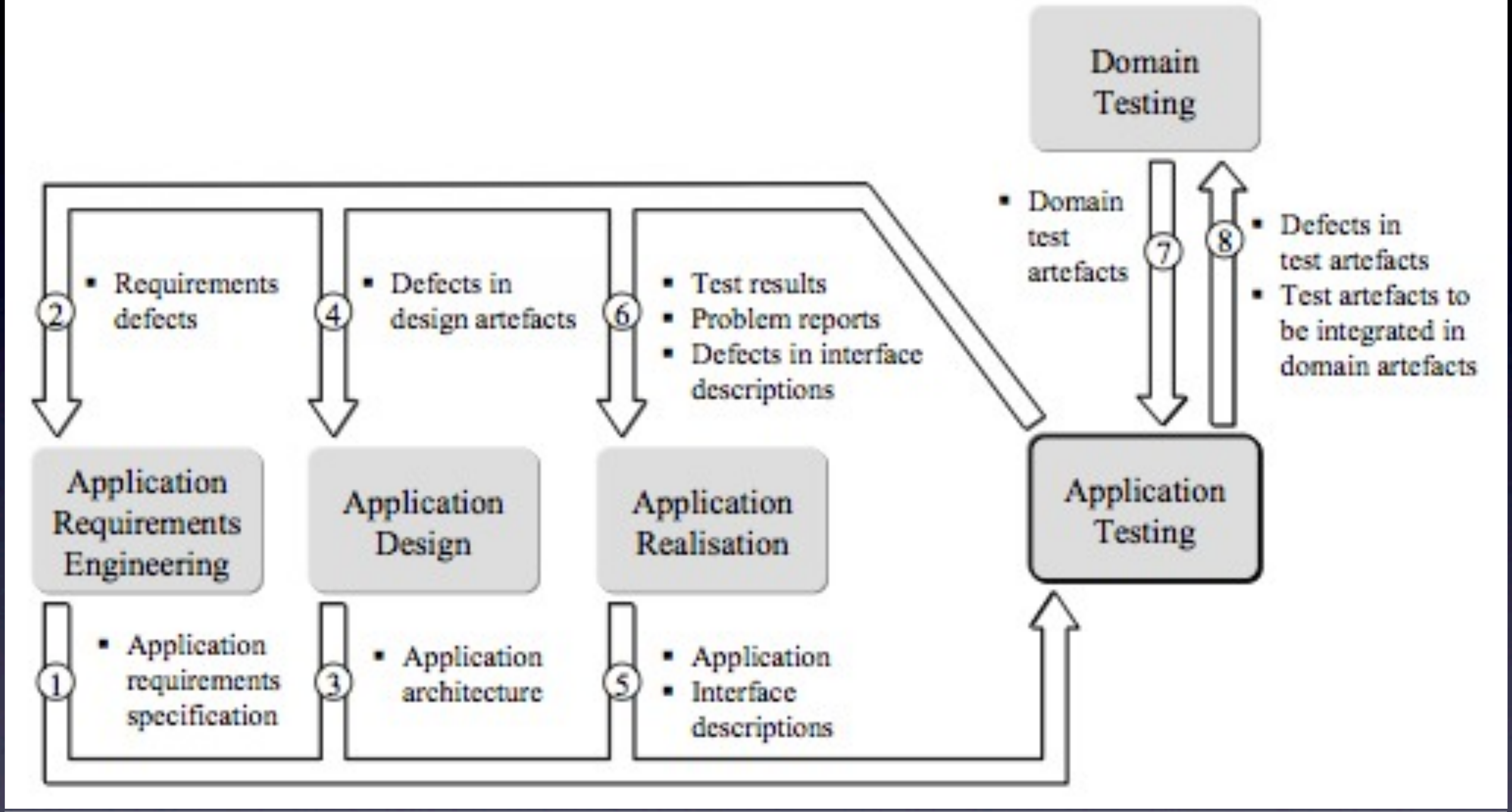
What is different from standard SD?

- Test activities are distributed between domain and individual apps
- Test artefacts contain common and variable parts to be reusable
- Documentation determines which variable parts to use for which config

Common strategies to PLT

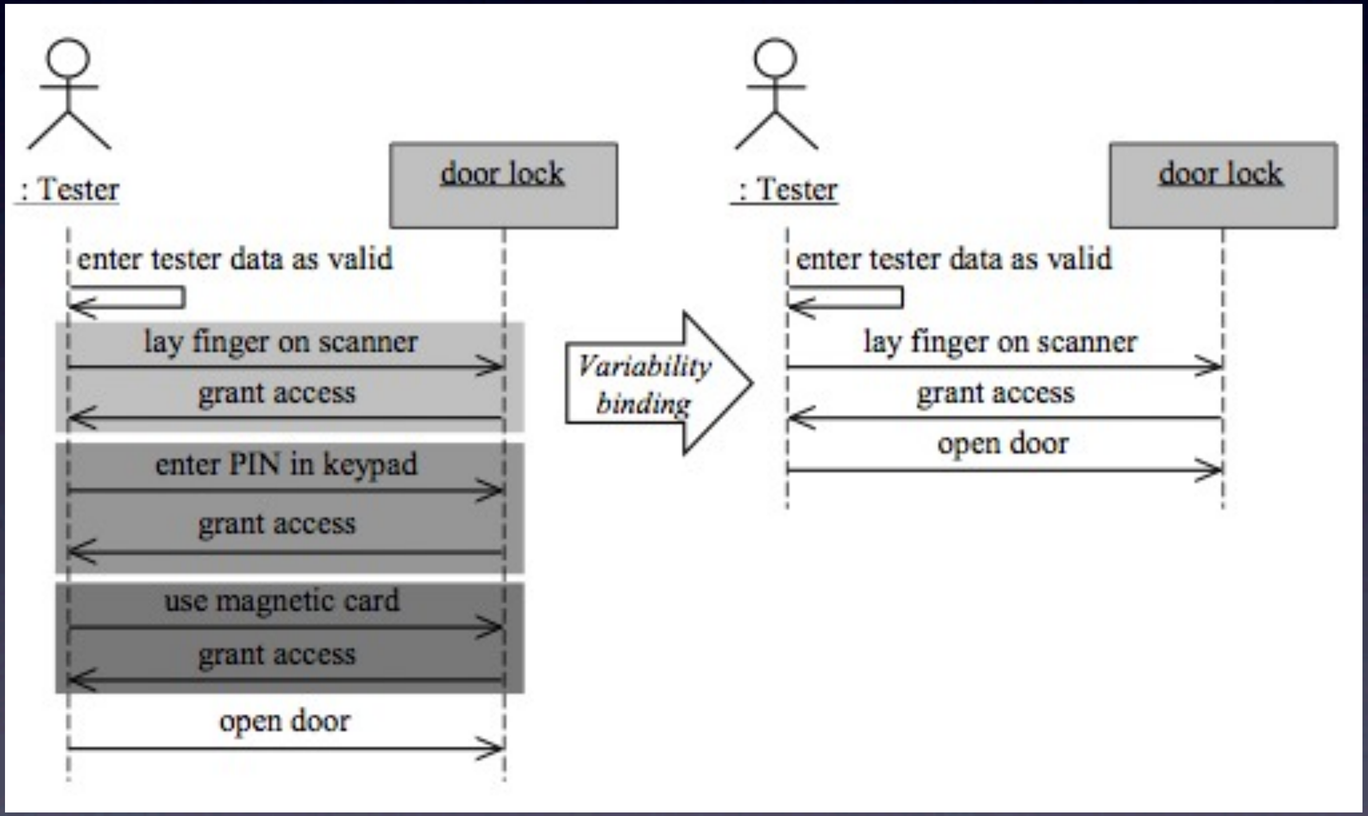
- Brute force strategy (BFS) - perform tests at all test levels and for all possible applications during domain testing
- Pure application strategy (PAS) - Perform tests only in app engineering. Only app specific tests are created and executed. No reusable domain test artefacts are created during domain testing
- Sample application strategy (SAS) - Use one (or a few) sample applications to test the domain artefacts. App testing still required for each app!
- Commonality and reuse strategy (CRS) - Domain testing tests common parts and prepares artefacts for testing variable parts. App testing aims at reusing test artefacts for common parts and reusing the predefined, variable domain test artefacts for test specific applications.


Domain Engineering
Product Management
Domain Requirements Engineering
Domain Design
Domain Realisation
Domain Testing
Domain Artefacts incl. Variability Model
Requirements
Architecture
Components
Tests
Application Engineering
Application Requirements Engineering
Application Design
Application Realisation
Application Testing
Application N – Artefacts incl. Variability Model
Application 1 – Artefacts incl. Variability Model
Requirements
Architecture
Components
Tests


Domain Testing
Domain test artefacts
7
8
Defects in test artefacts
Test artefacts to be integrated in domain artefacts
2
Requirements defects
4
Defects in design artefacts
6
Test results
Problem reports
Defects in interface descriptions
Application Requirements Engineering
Application Design
Application Realisation
Application Testing
1
Application requirements specification
3
Application architecture
5
Application
Interface descriptions

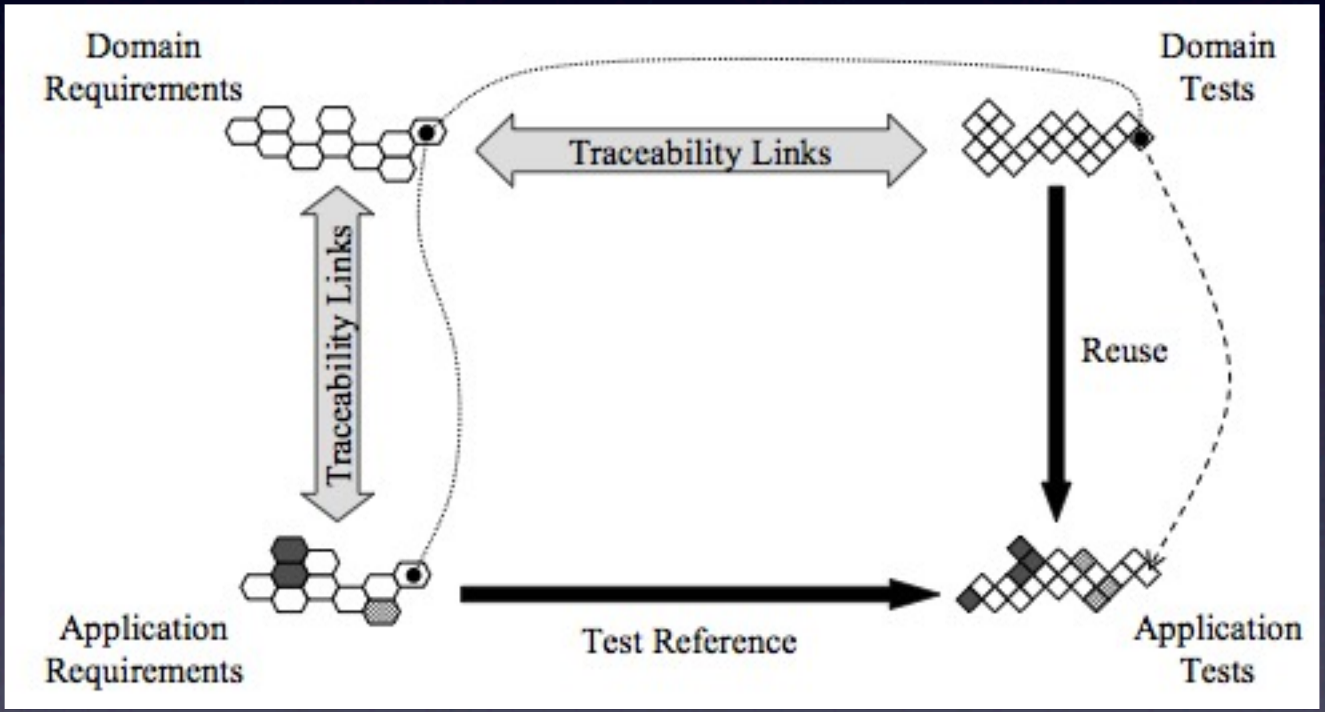
Essential prerequisites

- Dealing with variability
  - Variants and variation points from domain artefacts must be understood
  - Bind variability defined in domain artefacts with application variability model



Essential prerequisites

- Traceability links
    - Retrieve domain artefacts
    - App Req Spec (
        - Var Model Deltas, i.e. the impact of deltas; is it worth implementing?
        - Traceability between domain and application → App Req Spec
        - Document variability point bindings defined in the domain var model → App Var Model
    -

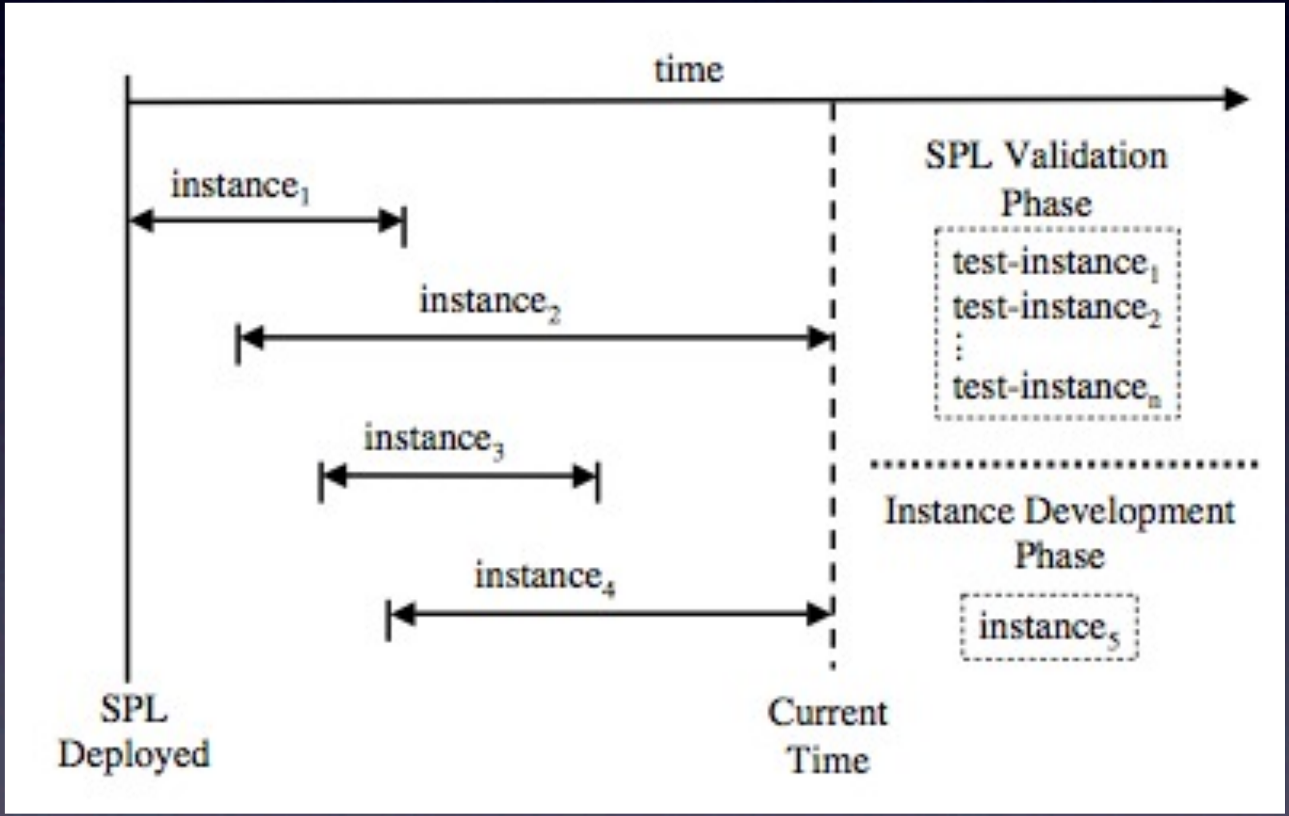


App var tests

- Specific tests for detecting issues connected to variability
- Variant absence test (VAT) makes sure that the app does not include unnecessary vars
  - unit - IFDEF statements
  - integration (component focus) - link time, load time, run-time config
  - system - verifies major system features that span multiple components
- Regression tests...

How much do we cover?

- Test coverage?
  - Sample input domain (very much as sampling in experiments)
  - Variable strength array (some vars may be more likely to interact?)
  - Cumulative test coverage



Cohen, M. B., Dwyer, M. B., and Shi, J. 2006. Coverage and adequacy in software product line testing. In *Proceedings of the ISSTA 2006 Workshop on Role of Software Architecture For Testing and Analysis* (Portland, Maine, July 17 - 20, 2006). ROSATEA '06. ACM, New York, NY, 53-63. DOI= http://doi.acm.org/10.1145/1147249.1147257

Q and A

- Which artefacts should be tested in domain engineering and which ones in app engineering?
  - Generate test paths w/o variability as intermediate artefacts to test commonalities
  - Derive test paths for specific app instances
  - Derive test cases to check for presence and/or absence
  - Connect tests to specific app instances and create modified regression test suites

Q and A (cont)

- How do we facilitate the reuse of SPL test artefacts?
  - Explicitly define variability in domain tc
  - Generate test paths w/o variability as intermediate artefacts to test commonalities
  - Connect tests to specific app instances and create modified regression test suites

Q and A (cont)

- Ensuring correct variability bindings?
  - Test for presence and/or absence of variants in an app

Things to do...

- Quality assurance?
  - Integration and regression tests?
  - Review techniques?
- Model-driven development?
  - And esp. how do we introduce formalism to this?
- Evolution?
- Multiple product lines and variability across these?
- Tool support
  - Traceability
  - Connection between domain and app
- Ps improvement and assessment
  - Using different processes for different apps or, even, domains?
- Economics
  - Predictive models (does it pay off?)
  - ROI

Thesis proposals

- Currently four thesis proposals announced
  - Proof reuse to verify SPLs
  - Model-based dev of SPLs
  - Comparing product maps and feature diagrams
  - Impact of core product in Δ-modelling
- See course page for more info
- Contact schaefer@chalmers.se if interested!

Additions to course

- Dec. 8th, 10.00, Dr. Ina Schaefer will come and present her research and talk about the possibilities students have in this field!