

Requirements Engineering II

Traceability and Software
Execution

This presentation will introduce...

... ideas how to automatically identify requirement traces

- Justification
- Problem statement
- High-level idea
- Maintenance classification
- Approach by Cherrada & Glinz in detail
- Other approaches

Justification

- Requirements are a high-level specification of code
- Dependencies of code to requirements and vice versa
- No technical background needed, larger audience

= Requirements integral part of re-engineering

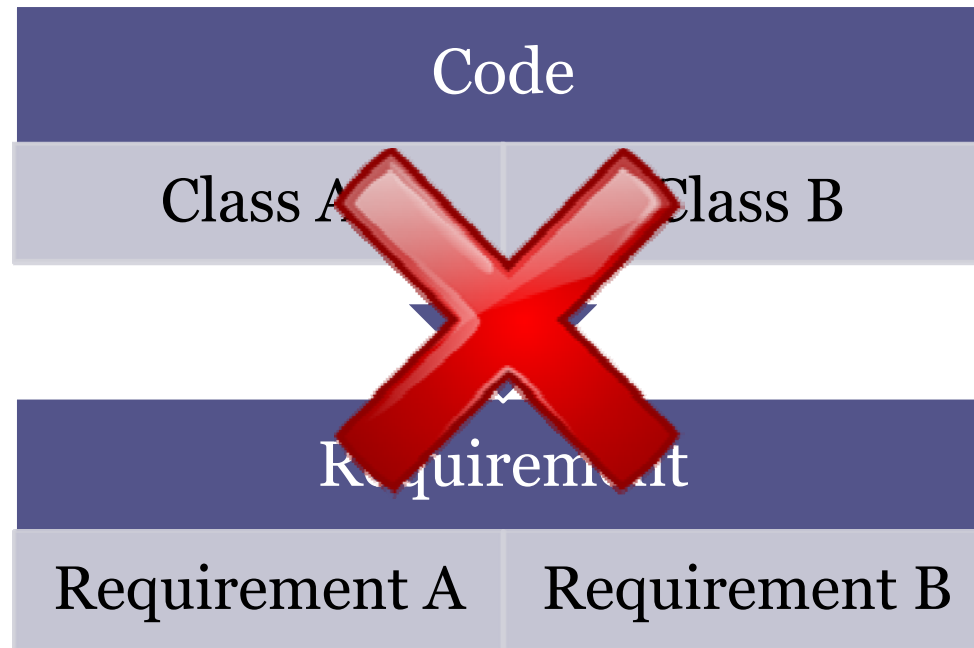
Problems

1. Developers are lazy
2. Requirements are mostly created manually and in independent system
3. In large systems, requirements are complex
4. Requirements are less detailed than code
5. Problem of outdated requirements is long-term

= Lazy developers hate manually updating complex requirement specifications

High Level Idea

There is no direct connection between code and requirement

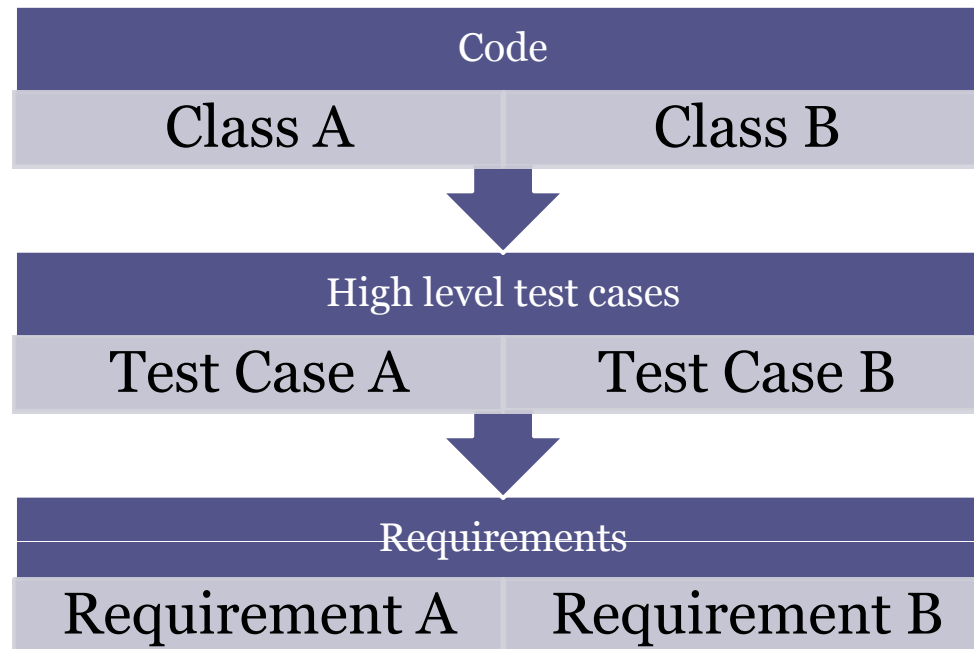


High Level Idea (2)

- No 1:1 relation between code and requirements
 - One function can satisfy multiple requirements
 - + one requirement can describe multiple functions
 - + requirements are not necessarily a function

= impossible to identify for a computer

High Level Idea (3)



Maintenance Classification

... and its impact on test cases

- Reductive
- Corrective
- Enhance, Performance
- Grooming, Preventive

There are others, but they are not detectable or not feasible to detect

Approach by Cherrada & Glinz in detail

- Idea: Manual links between High Level Test cases
- Automated scan of changed tests using algorithm
 - Classification of maintenance
 - 5 rules (explained in next slide)
- Result: Hints/instructions to add, update or delete requirements

Approach by Cherrada & Glinz in detail

- 5 rules, iterate through all requirements and assign points
5 = affected, 3 = unknown, 1 = not affected
 1. If a requirement is related to only 1 test, and that test changed
 2. If a requirement is related to a set of tests, and none changed
 3. If a requirement is related to many tests, and only 1 test changed
 4. If many requirements are related to one test and that test changed
 5. Else assign 3 points to requirement

Even more automated approach

by Egyed and Grünbacher

- Code and Requirements bi-directionally affected
- Common ground in code = common ground in requirements
- Using trace analysis on called test cases
 - Link requirements and code segments
 - Identify dependencies between requirements
- Drawback: Initial dependency information has to be entered manually