Testing Highly Complex System of Systems: An Industrial Case

Nauman Bin Ali, Kai Petersen, Mika V. Mäntylä
<mika.mantyla@aalto.fi>

Systems of Systems = Set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities

Examples

- Integrated Undersea Surveillance System (IUSS) and Sound Surveillance System (SOSUS)
- Global Positioning System (GPS)
- Telecommunication network
  - Base station (communicate with phone)
  - Administration (disable phone)

Characteristics

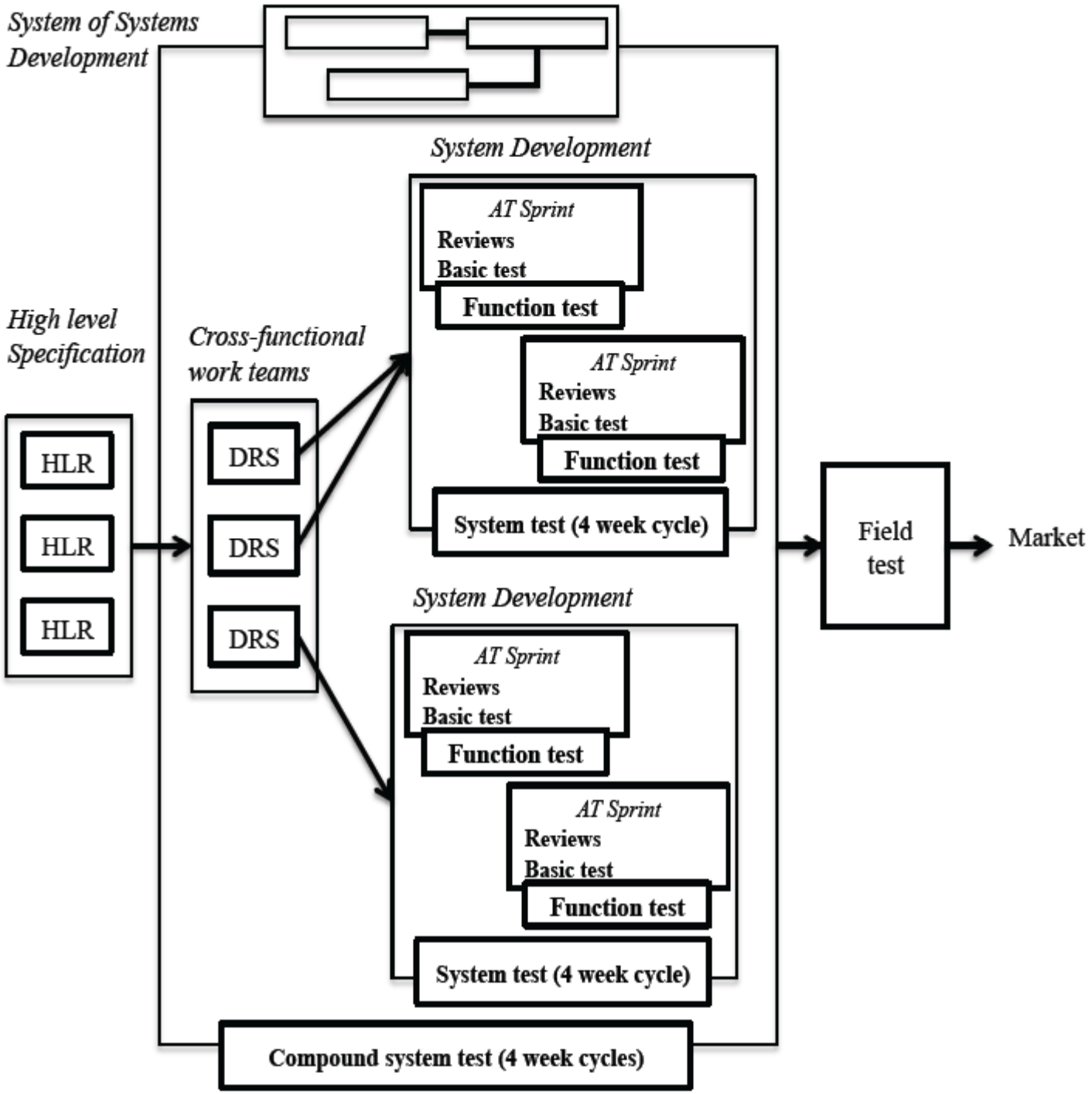
- C1 Operational independence
- C2 Managerial independence
- C3 Integration of system into system of systems
- C4 SoS comprised of complex systems
- C5 System suppliers deliver systems for integration
- C6 Complete technical overview of SoS and system

Research questions and data

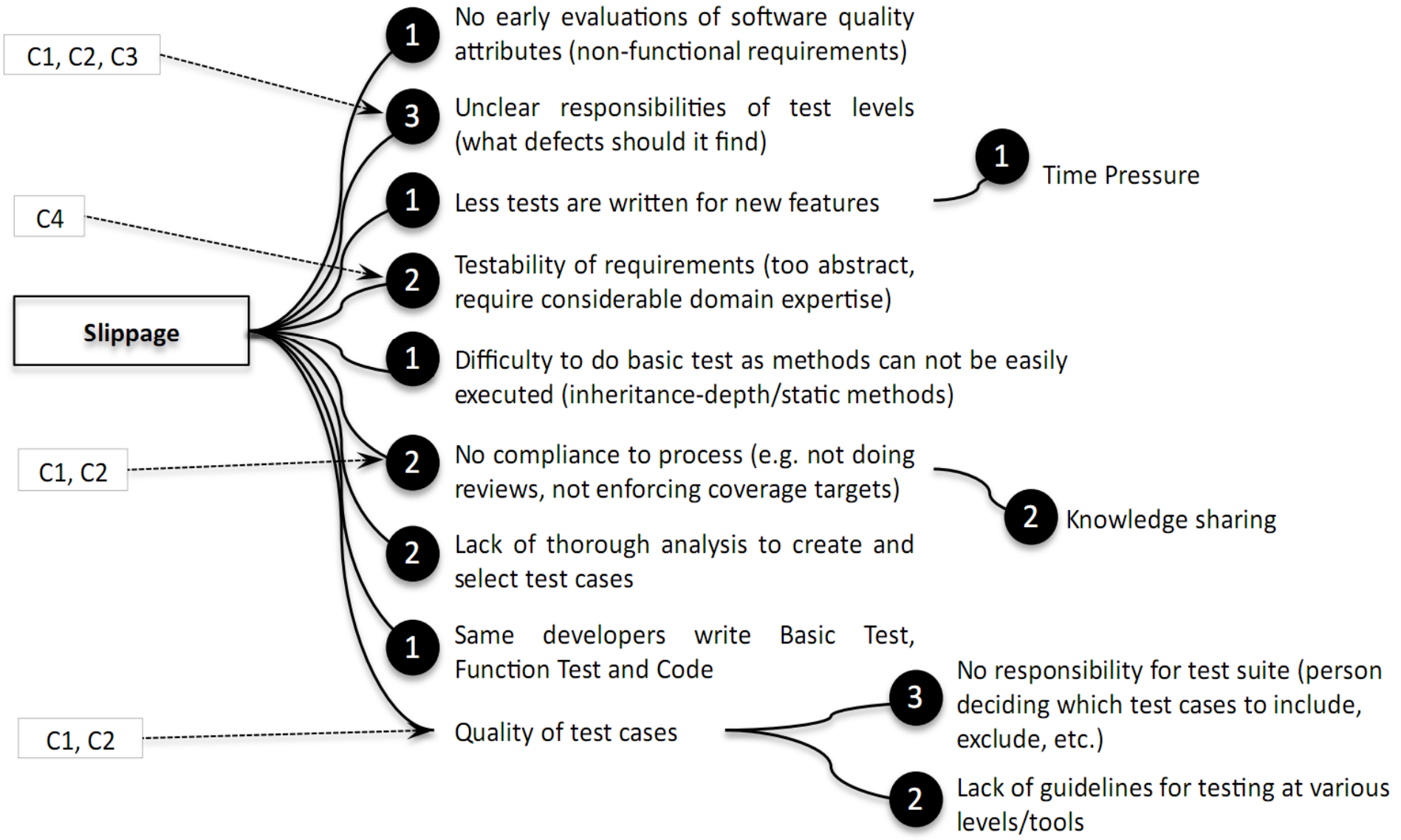
- RQ1: How is SoS testing done in industry?
- RQ2: What are perceived and measured challenges when testing SoS and how are they different from testing challenges of other contexts?
- RQ3: What potential solutions do practitioners see in order to address the challenges identified?

- Data1: Software development process descriptions
- Data2: Interviews of practitioners
- Data 3: Defect data and defect slippage through process phases

RQ1: Test Process of the case company



Slippage = Defects are not found where they suppose to be found



Defect Slippage can help in finding process improvement targets

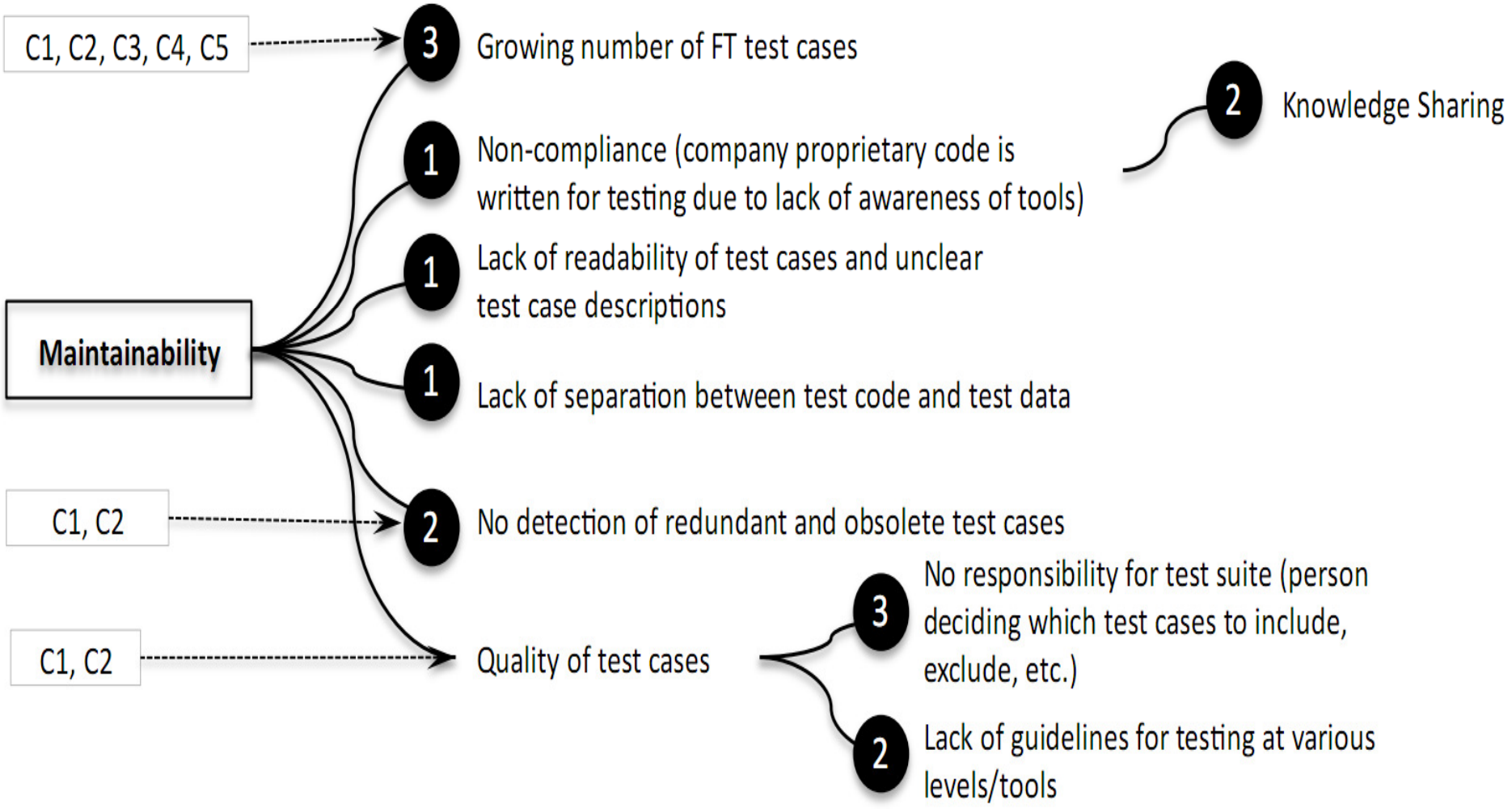
- We want **bold** numbers to be 100.00%
- We want normal numbers to be 0.00%
- Numbers in *italics* are anomalies

'Should' to 'Did' detect	Rev	BT	FT	ST	CST	FOA	Total slip-page
Rev	**30.76**	7.69	30.76	19.23	0.00	11.53	25.71
BT	*6.15*	**26.15**	32.30	18.46	4.61	12.30	62.85
FT		*7.14*	**50.00**	28.57	14.28		8.57
ST				**88.88**	11.11		1.42
CST					**100.00**		0.00
FOA							**0.00**

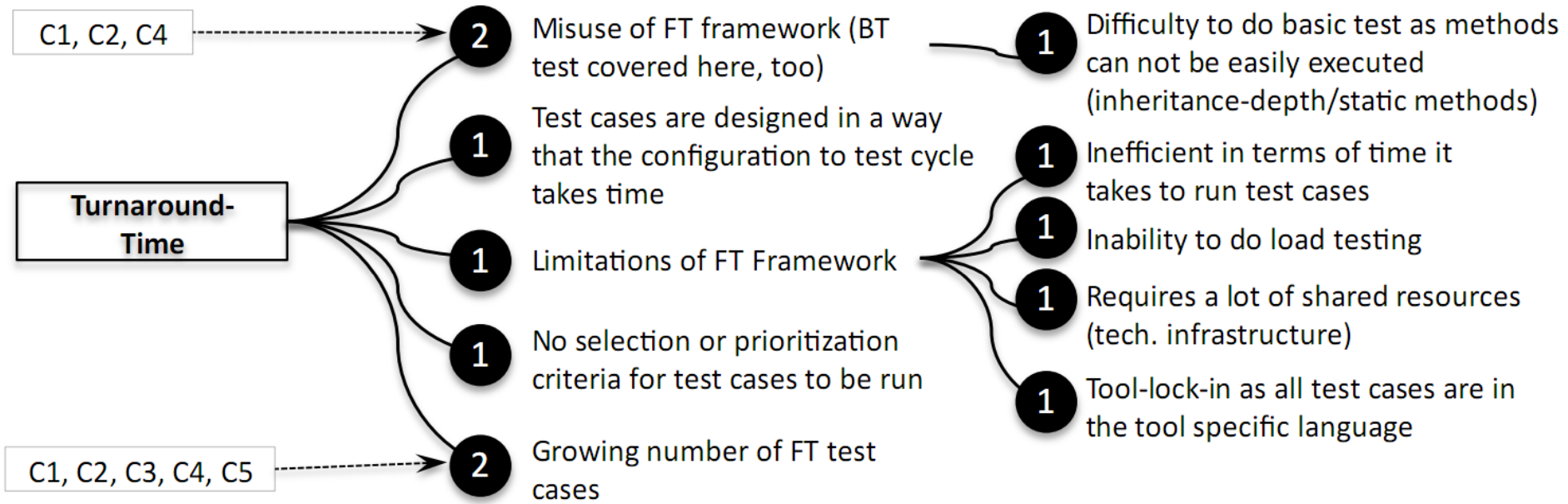
Defect slippage to customer

'Should' to 'Did' detect	Percentage found by Customer
Rev	15.27
BT	63.88
FT	6.94
ST	11.11
CST	1.85
FOA	0.00
Customer	0.92

Maintainability of the test suite



Long turnaround time for testing



Improvements (RQ3)

- Developer QA: **Have more of these!**
  - Code reviews
  - Basic testing
- Function test: **Fix these!**
  - Maintenance responsibility of regression test suite
  - Controlling the size of regression suite
  - Reducing the feedback time for regression suite
  - Guidelines for FT tool usage
  - Technical improvements to FT tool

Improvements (RQ3)

- Testing in all levels
  - Template to Improve the test-case quality
  - Definitions and responsibilities of test levels:
- Requirements engineering and communication
  - Improving the testability of requirements:
  - Improved interaction between teams and cross-functional requirement teams
  - Feature status tracking

Contributions

- A study that focuses on testing challenges and solutions on SoS context
- Improve strong practices with remaining improvement potential, or weak practices?
  - Contradiction between the fault slippage measure and interviews with practitioners.
- Poor maintainability and turnaround time of functional tests are linked
  - Difficult to improve turnaround time if the test code has low maintainability.
  - Low maintainability -> developers add new test cases rather than modify the existing test cases -> Longer turnaround time
- Maintenance of test code is problematic and needs to be studied