Early Software Product Improvement with Sequential Inspection Sessions: An empirical Investigation of Inspector Capability and Learning Effects

Dietmar Winkler, Bettina Thurnher, Stefan Biffl

Institute of Software Technology and Interactive Systems
Vienna University of Technology

dietmar.winkler@qse.ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at

Motivation

- The construction of high-quality software products requires (a) professional approaches (e.g., processes and methods) and well-trained engineers.
- Early detection and removal of defects, e.g., in the design phase, helps increase software quality and decrease rework effort and cost.
- Prior empirical studies showed that UBR (software inspection with a usage-based reading technique approach) can focus on most important use cases and spot on the detection of crucial and important defects.
- Inspection promises to be a vehicle to support learning.

Questions:

→ How is the impact of inspector qualification on inspection performance?

→ Is there any notable difference of learning effects regarding inspection performance in a sequence of sessions?

Defect Detection with Inspection

- Software Inspection …
  - is a static analysis technique to verify quality properties of software.
  - does not require executable code (applicable to design documents).
  - focuses on defect types and location in the inspected object.
  - Guidance of inspectors with reading techniques and guidelines (how to traverse a software document).

- “Best-practice” approach: Usage-Based Reading (UBR)
  - Well-investigated reading technique approach.
  - Goal: focus on most important defects first (classes “crucial” and “important”).
  - User focus: use cases lead the inspection process.
  - Application of use cases and scenarios from requirements documents in a pre-defined order (prioritized by a group of experts) to design documents.

Learning with Inspection

- Inspection supports learning due to
  - a systematic and structured process (inspection process) which is repeatable and traceable
  - Active guidance to support individual inspectors in defect detection tasks (guidelines, checklists, etc.)
- We refer “learning” as an improvement of individual inspection performance in a sequence of inspection sessions within a similar application domain.
- Research Questions:
  - Is there any difference of inspection performance regarding system complexity and inspector capability?
  - Can we identify differences of gained additional inspection experience in a second inspection session?

Dependent Variables and Hypothesis

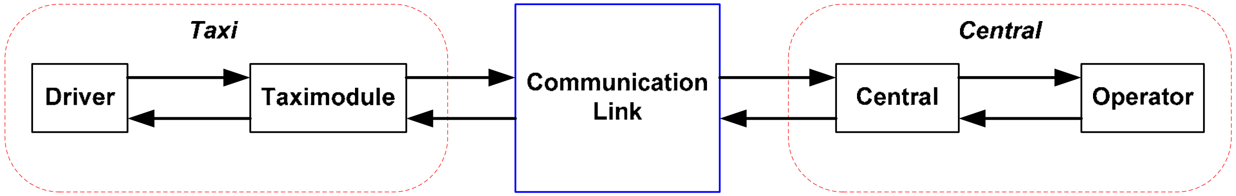
- Inspection effort includes individual preparation time and inspection duration (we did not consider inspection pre-work, e.g. use case prioritization).
- Effectiveness is the number of defects in relation to the overall number of seeded (important) defects.
- Efficiency is the number of defects found per time interval (e.g., defects found per hour)
- False Positives is the share of "wrong defects found" by individual inspectors.

Hypothesis:

- Effectiveness and efficiency will increase in a second inspection session.
- False positives will decrease in a second inspection session.
- Inspectors will perform better in the less complex part of the system.

Experiment Description

- The system represents a snapshot of the development process of a taxi management system including requirements and design documents and source code fragments.
- Two parts of the system



- at different levels of system complexity (amount of inspection material). Complexity (Central) > Complexity (Taxi).
- Total number of 56 seeded important defects within the design specification and the source code.
- Three experiment phases processed:
  (a) training & preparation, (b) individual inspection, and (c) data submission.

Study Arrangement (2x2 study design)

	Group A	Group B
Session 1	Central (C) (Data Set 1)	Taxi (T) (Data Set 2)
Session 2	Taxi (T) (Data Set 2)	Central (C) (Data Set 1)

- Subjects
  - 104 graduate students in a class on quality assurance and software engineering: 18 less, 22 medium and 12 higher-qualified inspectors per group.
  - The subjects were randomly assigned to 2 groups.
- Data Set 1 (Central part) is the more complex part and Data Set 2 (Taxi) is the less complex part of the system.

Threats to Validity

Internal validity:

- Avoidance of communication between individuals during the study execution.
- Participants could take individual brakes, whenever necessary (break durations noted).
- Limitation of the overall inspection duration (3h for taxi, 5h for central due complexity reasons).
- Experience questionnaire to get an insight on prior knowledge.
- Feedback questionnaire to see if the participants followed the study process properly.

External validity:

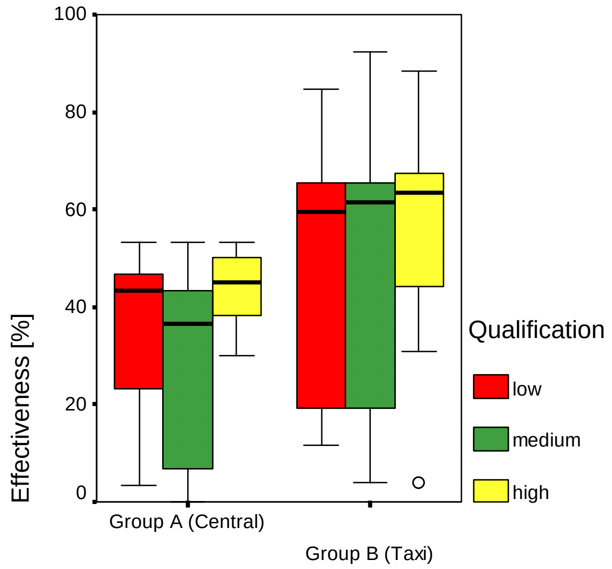
- Well-known application domain to avoid domain-specific interpretation problems.
- Pilot test and reviews to assure correctness of experiment material.
- Control of variables due classroom setting.

Results: Effectiveness

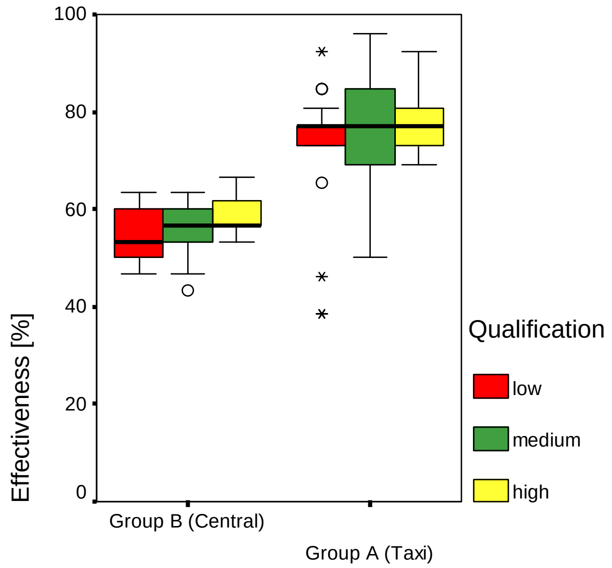
- Effectiveness: number of defects found in relation to the number of seeded defects.

	Session 1		Session 2	
	Central	Taxi	Central	Taxi
Mean	34.2%	51.0%	55.7%	74.6%
Std.Dev.	17.2%	25.7%	5.5%	11.6%

- System Complexity:
  - Significant differences between Group A and B in both sessions.
  - Only small advantages for high-qualified inspectors.
- Similar system parts:
  - significant differences for all qualification classes and both groups.
  - Advantages for less- and medium-qualified inspectors.



Session 1



Session 2

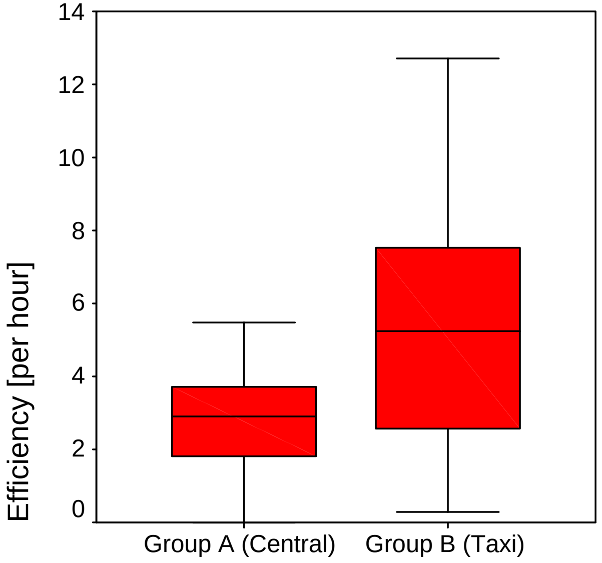
Results: Efficiency

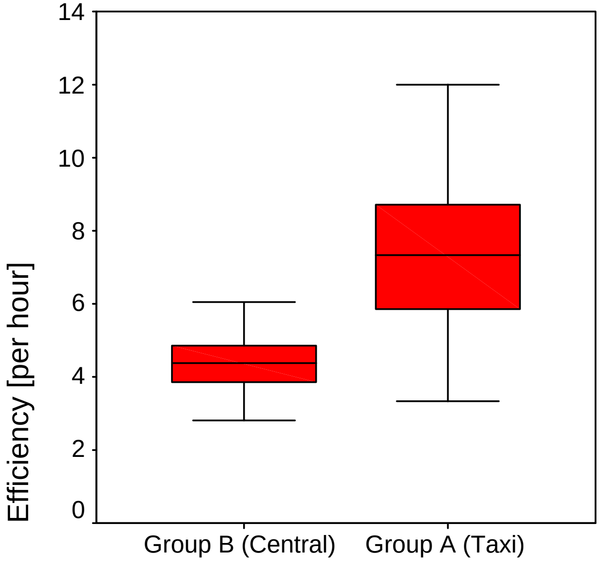
- Efficiency: number of found defects per hour.

	Session 1		Session 2	
	Central	Taxi	Central	Taxi
Mean	2.7	5.3	4.4	7.4
Std.Dev.	1.4	3.1	0.7	2.0

- System Complexity:
  - We observed significant differences between less- and higher qualified inspectors in both session within the more complex part.
  - No significant differences in the taxi part.

- Similar system parts:
  - There is a notable learning effect ($p<0.040$) in session 2 for all qualification classes.
  - We observed the highest improvement factor for medium qualified inspectors and the lowest for higher-qualified inspectors.





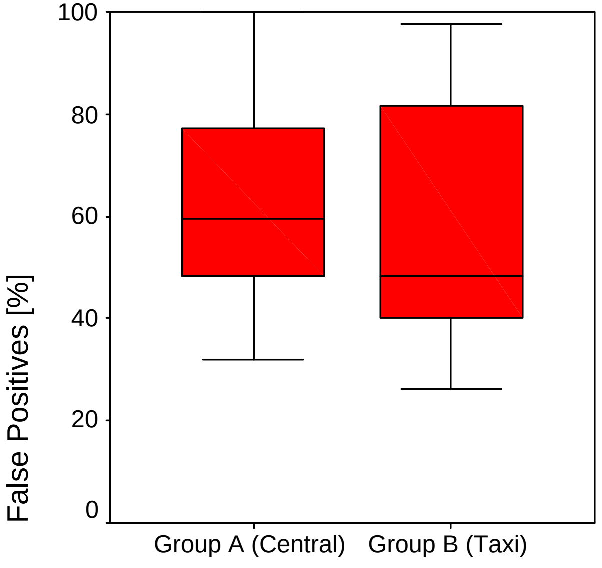
Session 2

Results: False Positives

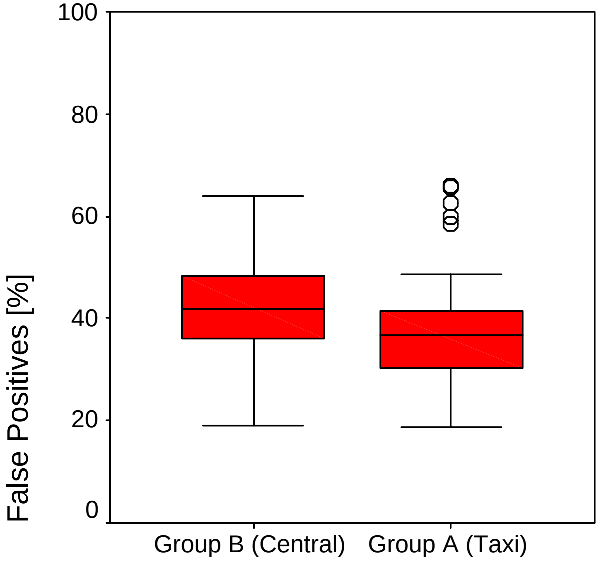
- False Positives (FP): share of “wrong defects found”.

	Session 1		Session 2	
	Central	Taxi	Central	Taxi
Mean	63.3%	36.9%	55.7%	41.7%
Std.Dev.	19.4%	10.9%	21.9%	9.6%

- System Complexity:
  - No significant differences within both sessions.
- Similar system parts:
  - The FP decreased for group A (starting with the more complex central part of the system) and increased for group B (starting with taxi).



Session 1



Session 2

Summary and Further Work

Summary:

- Software inspection is an appropriate method for defect detection in early software development phases and learning.
- The results showed that inspectors who started with a simple (less complex) system part are more successful regarding effectiveness, efficiency and false positives.
- We observed a smaller learning effect for higher-qualified inspectors. One reason might be that they apply their own suitable inspection approach.

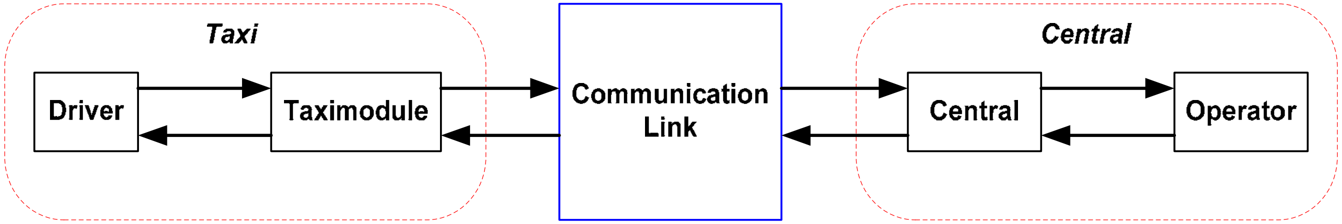
Further Work:

- More detailed investigation of the results regarding inspector qualification, different defect types and document locations will lead to a deeper insight in learning with inspection.

Backup

Study Artifacts

System Overview: Taxi Management System



Software artifacts per part

	Central (DS1)	Taxi (DS2)
Requirements definitions	4 pages 1,400 words 16 use cases	4 pages 1,100 words 11 use cases
Design document	5 pages 1,600 words	3 pages 800 words
Source Code in Lines of Code (LoC)	750 LoC 2,100 words	650 LoC 1,400 words
Common system information	1 component diagram 1 class diagram 11 sequence charts	

Seeded defects per part

	Central (DS1)	Taxi (DS2)	Total
No. of Defects	30	26	56
Share of Defects	54%	46%	100%

Data Set 1 (Central part) is the more complex part (higher amount of inspection material) of the system.