An Empirical Study on Design Quality Improvement from Best-Practice Inspection and Pair Programming

Dietmar Winkler, Stefan Biffl
Vienna University of Technology,
Institute of Software Technology and Interactive Systems

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

Motivation

- In general, early detection and removal of defects, e.g., in the design phase, helps increase software quality and decrease rework effort and cost.
- Specifically, the design phase offers leverage to improve software quality, when the requirements “meet” the opportunities and constraints of engineering solutions.
- Analytical Quality Assurance typically uses reviews and inspections for systematic V&V in early software development phases.
- Pair Programming is a constructive approach including implicit quality assurance.
- *RQ:*
  - *How effective can inspectors be at detecting defect compared to programmers who find defects as by product of their construction activities?*
  - *How to add benefits of inspection to pair programming?*
  - *Evaluation of this new integrated pair programming approach in an empirical study.*
- Initial study to find out whether the new approach is worthwhile investigating.

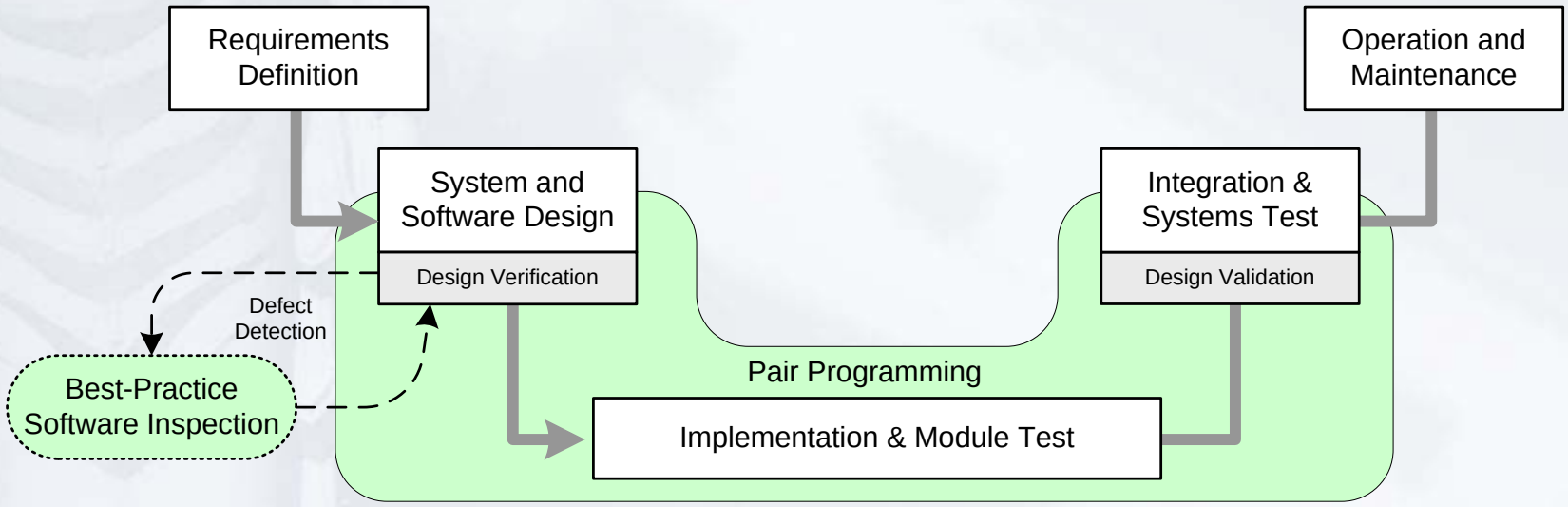
Analytical vs. Constructive Quality Assurance

Software Inspection

- Analytical Best-Practice Approach.
- Early in the Software Life-Cycle.
- Systematic quality assurance activity.
- Additional Effort for Defect Detection.

Pair Programming

- Constructive Approach
- Including implicit quality assurance activities.
- Design – Implementation – Testing.
- Defect detection as by-product of code construction.



Best Practice 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.

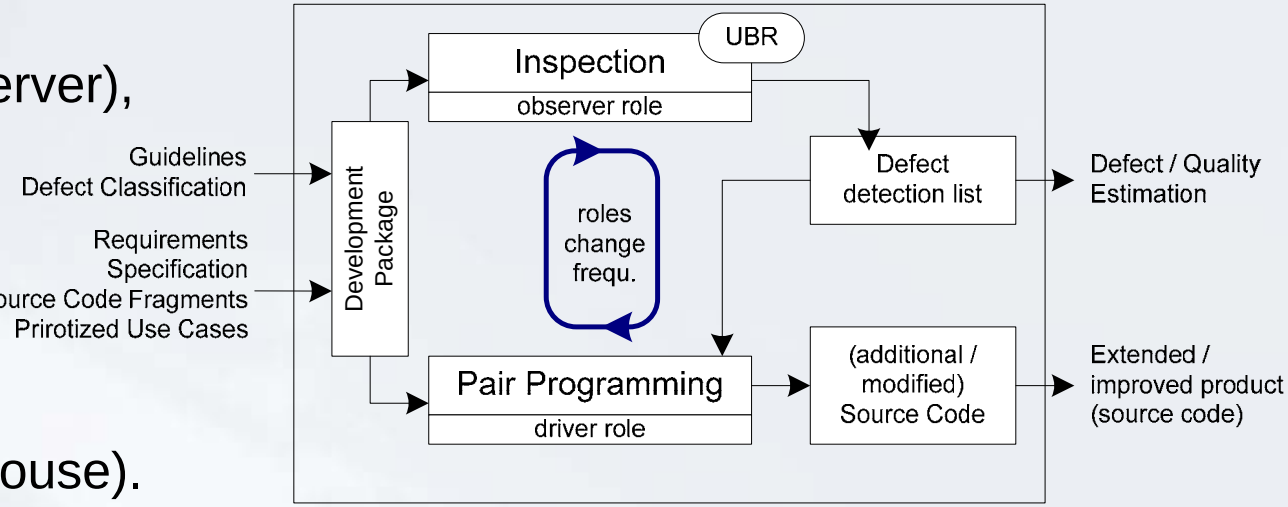
- **“Best-practice” approach: Usage-Based Reading (UBR)**
  - Well-investigated established approach.
  - Goal: focus on most important defects first (classes “crucial” and “important”).
  - User focus: use cases lead the inspection process.

  - Application of use cases and scenarios from requirements documents in a pre-defined order (prioritized by a group of experts) to design document.
  - Active guidance through guidelines and prioritized use-cases.

Pair Programming with Inspection

Pair Programming

- PP involves 2 persons (driver/observer),
  - Driver: implementation role.
  - Observer: supporting role.
  - Roles may change frequently.
- sharing a common development environment (screen, keyboard, mouse).



Integrated Pair Programming Approach

Integrated PP approach:

- More systematic defect detection approach.
- Active support with reading techniques and guidelines.
- Focus on most important use cases (prioritization).

➲ Comparison of best-practice inspection and the new integrated PP approach according to defect detection capability in an empirical study.

Research Questions

- **General idea:** Integrating inspection in PP leads to more structured defect detection approaches, improves overall defect detection capability, and software product quality.

1. Hypotheses for **natural work units** (individual inspectors vs. pairs)
   - H1.1: Effectiveness (PP) > Effectiveness (UBR): source code documents
   - H1.2: Effectiveness (PP) < Effectiveness (UBR): natural-language text documents.
   - Note: higher overall effort applying PP, because of different “team size” (2 persons) and focus on code construction (defect detection as a by-product).

2. Similar hypothesis for “**minimal teams**” (2-person inspection teams vs. pairs).

3. Performance of **nominal teams**:
   Do mixed teams perform better than “best-practice” teams?

System Overview: Taxi Management System

- **System Overview:**

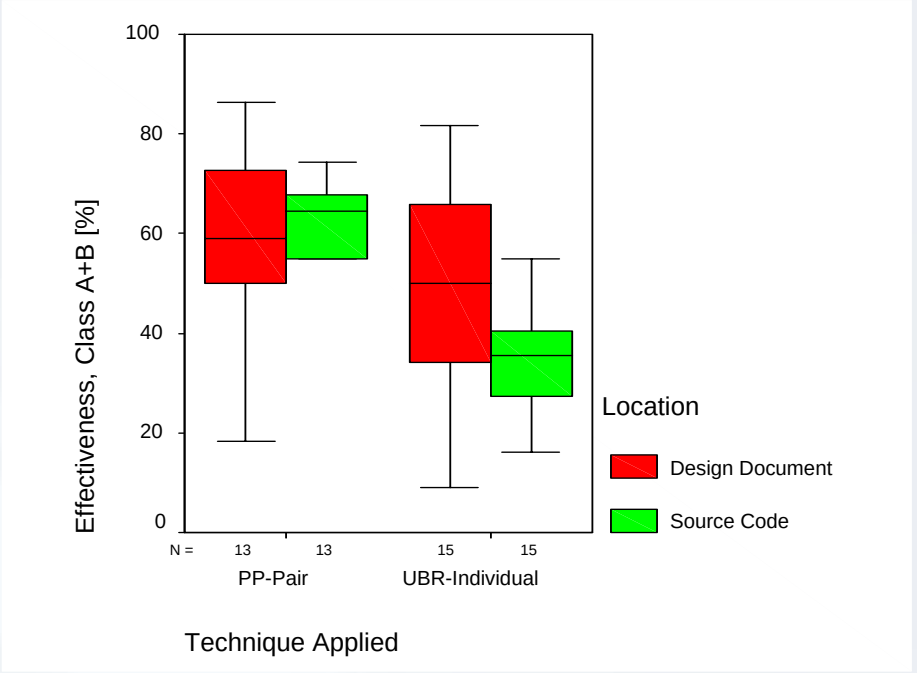
- **Software Artifacts**
  - Textual requirements: 8 pages, 2 component diagrams.
  - Design document: 8 pages, 2 component diagrams and 2 UML charts.
  - Use case document: 24 use cases and 23 sequence diagrams.
  - Source Code: some 1,400 LoC, 9-page description.
  - Guidelines and Questionnaires.

Empirical Evaluation: A Controlled Experiment

- An experiment to investigate defect detection capability of best-practice inspection and an integrated pair programming approach.
- Three **experiment phases** processed**:**
  (a) training & preparation, (b) individual inspection, and (c) data submission.
- **60 Reference Defects**
  - (29 crucial, 24 major, 7 minor) seeded in the design specification and source code.
- **41 Subjects** (experiment participants):
  - graduate students in a class on quality assurance and software engineering
    (15 UBR, 26 pair programmers, i.e., 13 pairs).
- **Effort** for Inspection / Pair Programming
  - Inspection effort includes preparation time (derivation of requirements and system functions, etc.) and individual inspection duration; overall effort: about 4-5 ph (person hours).
  - Pair Programmer effort includes the additional task of code construction;
    overall effort: about 17ph.

Results: Effectiveness of Working Units

- Effectiveness is the number of defects found defects in relation to the number of seeded defects.
- Focus on important defects (risk A+B) and document location (design document, source code).
- Effectiveness (PP) > Effectiveness (UBR) for all defect severity classes and document locations.
- Significant differences for
  - Source Code and
  - Design Document & Source Code.
- No significant differences for
  - Design Document.

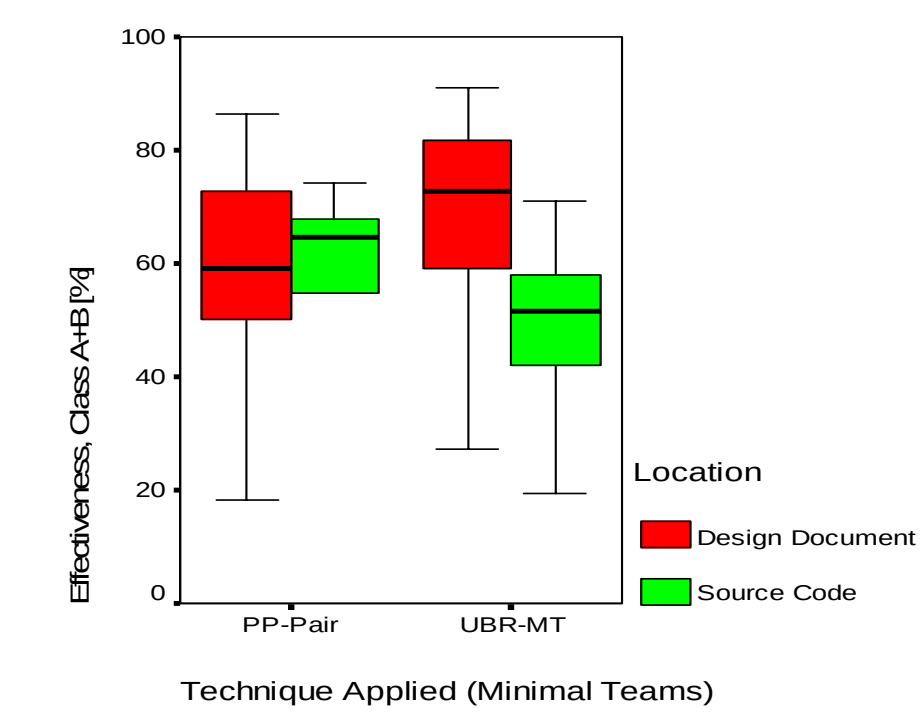


➲ The integrated PP approach outperforms inspection according to source code defects.

➲ Smaller differences for design documents but still advantages for PP.

Results: Effectiveness of “Minimal Teams”

- Comparability in team size → minimal teams.
  - Pair: 2 persons (original work unit).
  - UBR-MT: nominal 2-person team of individual inspectors (randomly assigned).
- Focus on important defects (risk A+B) and document location (design document, design source code).



- Significant differences for
  - Source Code.
- No significant differences for
  - Design Document and
  - Design Document & Source Code.

➲ PP outperforms effectiveness acc. to source code defects.

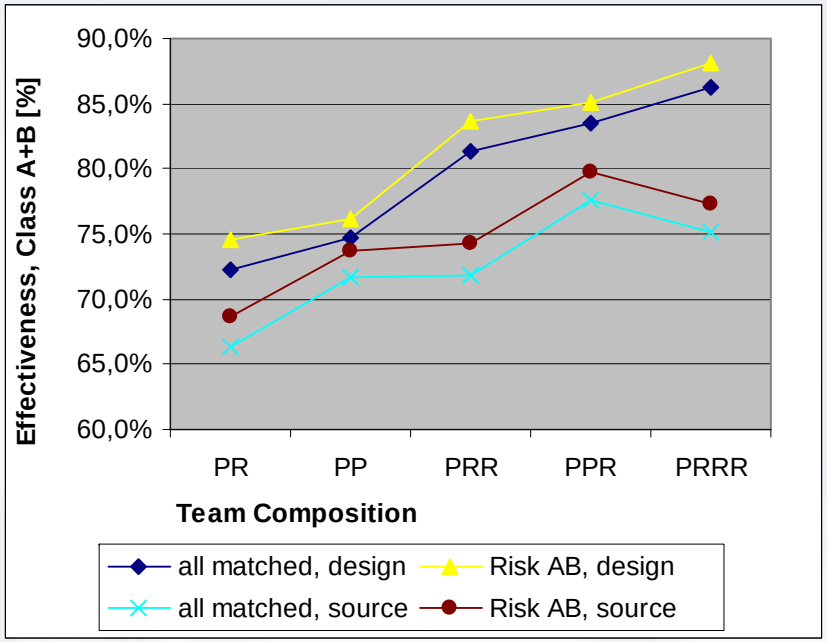
➲ Advantages for UBR-MT according to design document defects.

Results: Team Composition (1 of 2)

- Inspection and Pair Programming focuses on different defect types and defect locations.
- Thus, we expect an improved performance of mixed teams due to synergy effects.

- A “nominal team” is a collaboration of two or more members without interaction.
- Team building: continuous increase of effectiveness for up to 4 team members.

- Increasing effectiveness for design documents (smaller gain including additional pairs).
- Increasing effectiveness for source code including additional pairs and an almost constant value on inspector integration up to 4 team members.
- PRRR: decreasing effectiveness acc. to source code defects.



Note:
P … 1 Pair (=2 persons)
R ... 1 individual Inspector (UBR)

Summary & Further Work

- **Summary**
  - The integration of analytical quality assurance activities (software inspection) improves defect detection in specific document locations.
  - Effectiveness of *natural work units*: Significant differences for source code defects and all document locations. No significant differences for design document.
  - Effectiveness of "*minimal teams*": Improved effectiveness for design defects (but no significant differences) for UBR. Significant differences for PP and source code defects.
  - *Team Composition (Nominal teams)*: a mixed team of UBR and PP participants achieves higher effectiveness according to the individual focus of the technique. Best results for:
    - PPR (team size: 5) for all documents types and for source code documents.
    - PRRR (team size: 5) for design documents.
- **Further work**
  - Replication to achieve higher external validity and to verify the results.
  - Investigation of quality issues of modified/constructed pair programming source code.
  - Investigation of the impact of inspector capability on inspection performance.