How to Rate Programming Skills in Programming Experiments? A Preliminary, Exploratory Study based on University Marks, Pretests, and Self-Estimation

Sebastian Kleinschmager, Stefan Hanenberg
University of Duisburg-Essen, Germany

PLATEAU Workshop @ SPLASH
2011-10-24 Portland, Orgegon

Structure of this task

1. Motivation
2. Research question & results
3. Experiment
   - Experiment results
   - Threats to validity
4. Summary & conclusion

Motivation

- Different motivations
  - need for grouping criteria
  - scientific doubts that we currently have valid criteria
  - ...we are frequently asked to apply such criteria

Motivation - Need for grouping criteria

- We all believe that...
  - ...there are developers which differ with respect to development speed, coding quality, etc.
  - ...empirical studies are desirable that are performed (mainly) on good developers
  - ...education might be an indicator for the quality of developers
- But how to identify them?

Motivation – Scientific Need

- Empirical studies in software engineering are
  - Typically quite trivial in their experimental design
  - Blocked designs in SE hardly used – and hardly possible
  - Currently, empirical studies frequently do post-hoc grouping of subjects
  - Current dominating problem: large deviation
- Internal validity of experiments often problematic
  - Empirical comparisons require different groups
  - Similarity of groups cannot be decided upfront
- External validity unclear
  - Criteria are missing to what extent subjects are representative

Motivation – Personal Motivation

- We are frequently (always?) asked by reviewers to rate developers via questionnaires
- We have no idea what those questionnaires should tell us
  - Does „number of years in industry“ imply good programming skills?
  - Does „number of years in industry“ imply steep learning curve?
  - Does „100.000 JAVA-LOC“ imply better programming skills than someone who has done 5.000 LOC BASIC?

- We have doubts that...
  - ...there are (currently) scientific valid criteria that we can applied
  - ...„*often mentioned*“ criteria reduce the problem of assigning subjects
  - ...„*often mentioned*“ criteria increase the external validity

Motivation – Personal Motivation

- But...
  - We have done such questionnaires in our experiments
- Idea
  - Test, whether such criteria reveal differences in development skills

Motivation – Personal Motivation

- We are frequently asked by reviewers to do pre-tests with subjects
- We have no idea how to design such pre-tests
  - Example: *If the programming task is to implement a loop*
    - *is „implement a loop“ a valid pre-test?*
    - *can something completely different be used as a pre-test?*

- Idea
  - Test, whether an assumed pre-test (taken from experiment tasks) would have revealed difference in programming skills

Initial Considerations

- We have two criteria that can be directly extracted from our questionnaires
  - University mark
  - Self-estimation
- Due to the nature of our experiments (multiple programming tasks), we can
  - extract one task
  - check whether the task's result is an indicator for the other tasks

Initial Considerations

- **Remark**
  - Our view was that self-estimation does not represent a valid criteria – we used it only for comparing the results „to something“

Research question

- **Does one of the following criteria reveal significant differences in development skills?**
  - University marks
  - Self-estimation
  - Pre-tests
- **Result**
  - No

Approach

- Two previous studies
  - static type systems [StuchlikHanenberg@DLS11], 20 students
  - aspect-oriented programming [KHJ@ESEM09] 21 students

- Both studies
  - measured development time until completion
  - measured Java development times + *something else*

- Our interpretation
  - Java development time as an indicator for programming skills

Approach

- Separating subjects (per experiment) into according to criteria into quartiles

- Detecting, whether there are significant difference between quartiles (development time)

Approach - Example Task

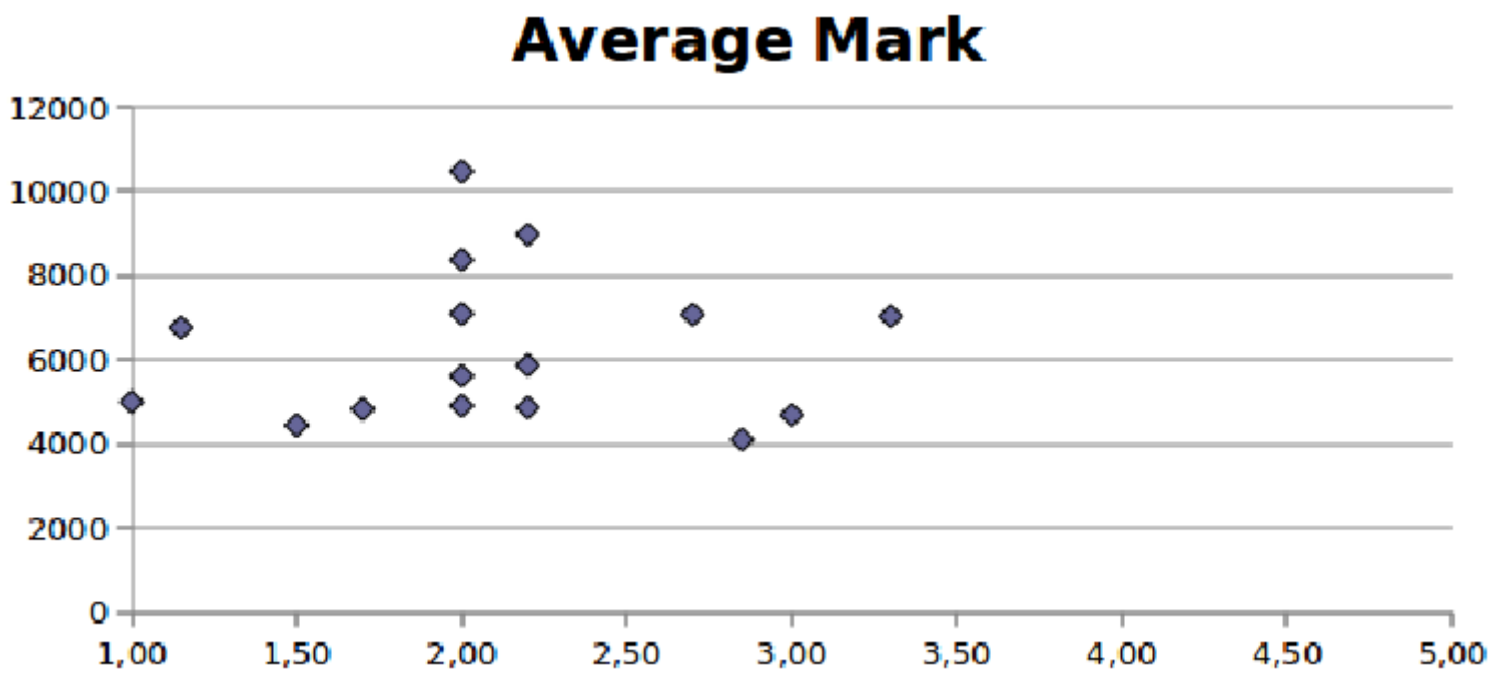
- **Example (2nd experiment, task 4)**
  - *Implement a method that stores a long ball between two players (which are either midfield players or forwards) of the same team*
- **Example solution**

```
void doLongBall(Player p1, Player p2) {
  if (p1.team != p2.team) throw new SoccerException("...");
  if (p1 instanceof Midfielder) {
    if (p2 instanceof Midfielder) {
      ((Midfielder) p1).longBall++;
      ((Midfielder) p2).receivedLongBall++;
    } else if (p2 instanceof Forward) {
      ((Midfielder) p1).longBall++;
      ((Forward) p2).receivedLongBall++;
    } else throw new SoccerException("...");
  } else throw new SoccerException("...");
}
```

Threats to validity

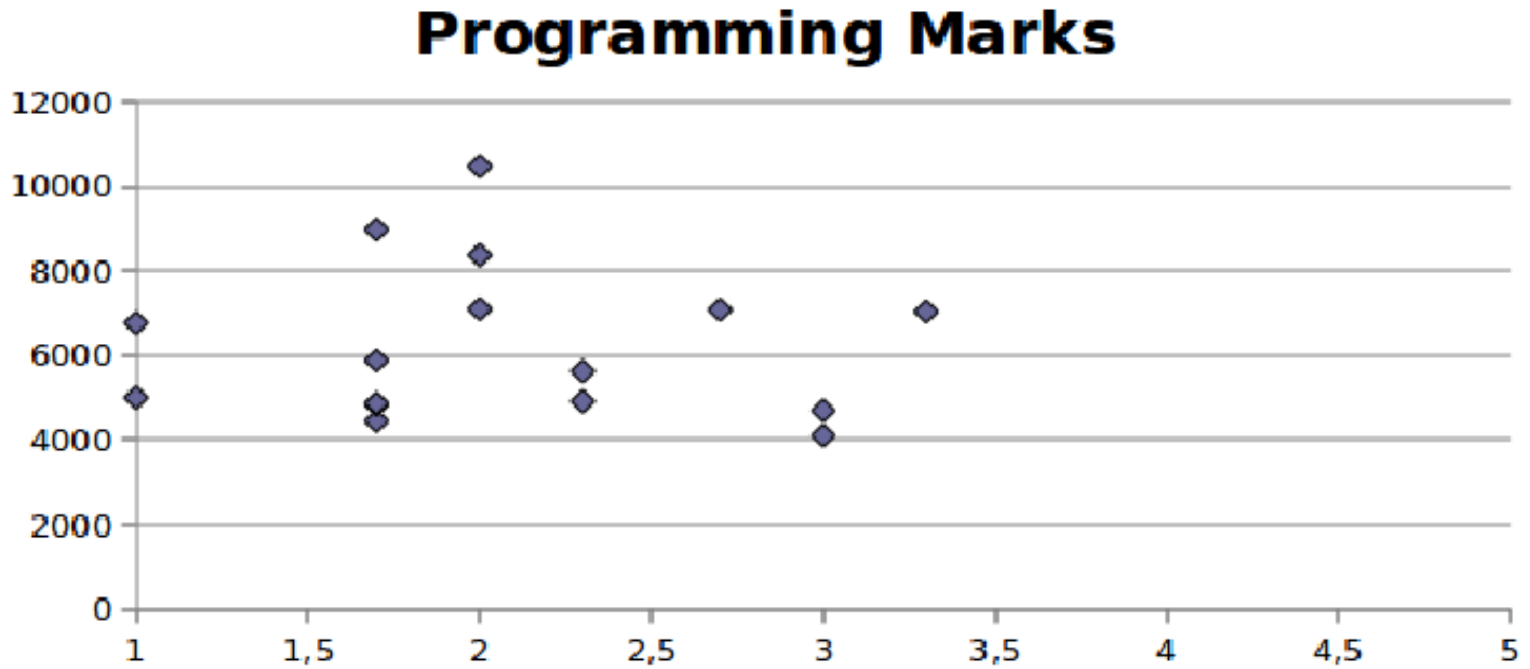
- Many, many, many
- …
- Small sample size
- Post-hoc testing
- Multiple U-Tests
- ....

University Marks



- Quantiles 1&3 bettern than quantile 2!
- No difference between 1st and 3rd!

University Marks



- Quantile 3 better than quantile 2(!)

Pre-test (experiment 1)

Task	P-value	dominating group
Task 1	0.3	-
Task 2	0.6	-
Task 3	0.82	-
Task 4	0.45	-
Task 5	0.88	-
Task 6	0.5	-
Task 7	0.45	-
Task 8	0.71	-
Task 9	0.26	-

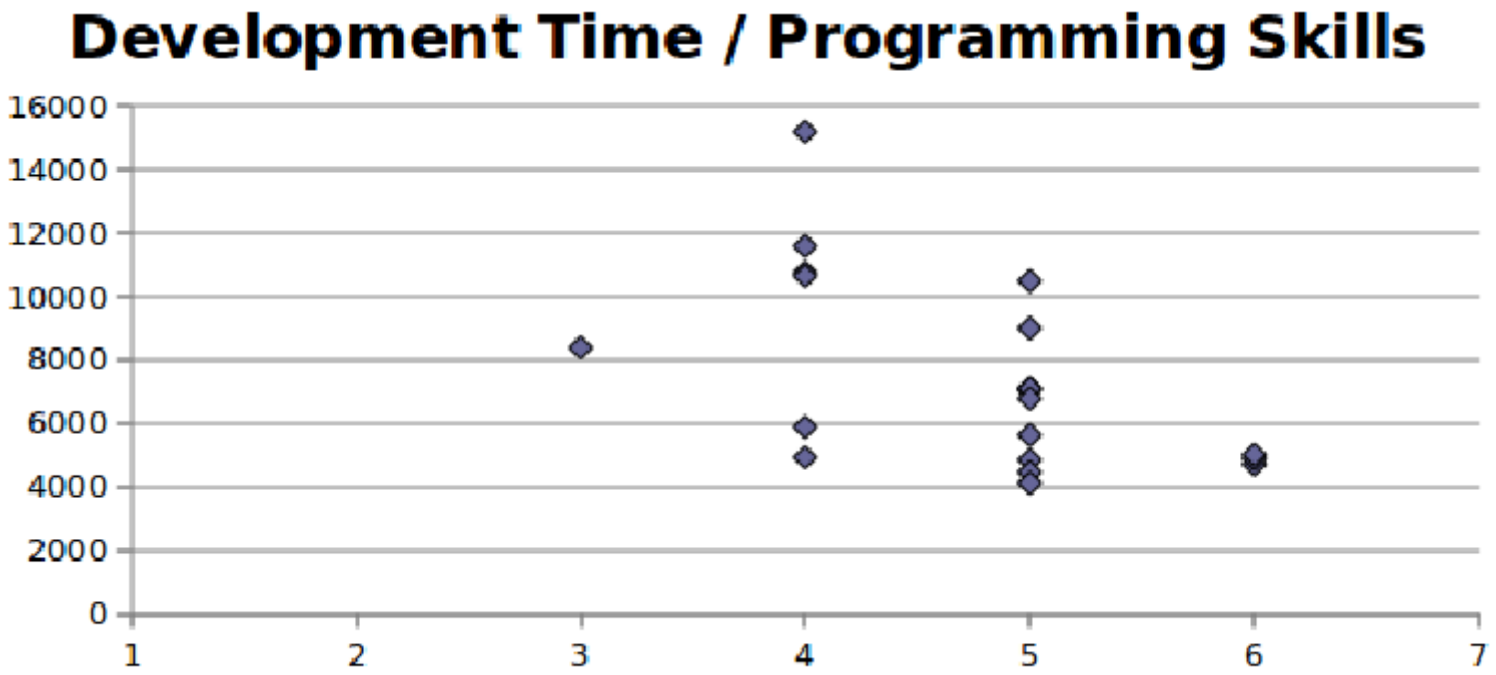
- No significance anywhere...

Pre-test (experiment 2)

Task	P-value	dominating group
Task 1	0,44	-
Task 2	0,02	best dev.
Task 3	0,26	-
Task 4	0,23	-
Task 5	0,12	-

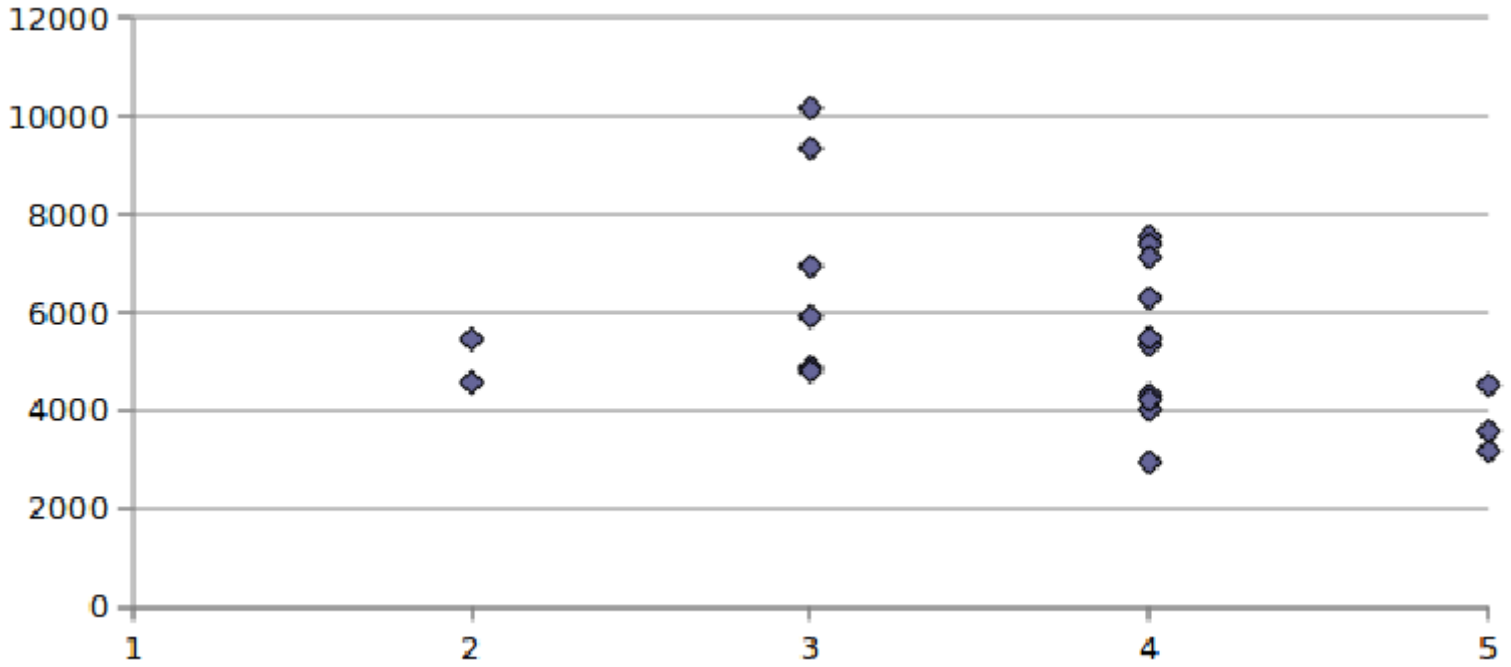
- Task 2 showed effect (5 „potential candidate“)

Self-estimation



- 2nd quantile worse than 1st , 3rd better than 2nd

Self-estimation (2nd Experiment)



- Group 5 better than group 1,2

Conclusion

- None of those criteria really showed a large effect
- We should continue with random experiment designs

How to Rate Programming Skills in Programming Experiments? A Preliminary, Exploratory Study based on University Marks, Pretests, and Self-Estimation

Sebastian Kleinschmager, Stefan Hanenberg
University of Duisburg-Essen, Germany

PLATEAU Workshop @ SPLASH
2011-10-24 Portland, Orgegon