STAGE – A Software Tool for Automatic Grading of Testing Exercises

Case Study Paper

Sebastian Pape[*], Julian Flake[+], Andreas Beckmann[+], Jan Jürjens[+]

[*]Goethe University Frankfurt
[+]TU Dortmund

May, 20th, 2016
ICSE – SEET '16, Austin, Texas USA

Outline

Motivation

- Course on Software Engineering
  - Model-based development
  - Quality management (testing)
- 15 weeks with 90-minute-lectures and 45-minute-tutorials
- 200 computer science undergraduates

technische universität dortmund

Motivation for automatic tool:

- Correcting homework is a time-consuming and error-prone task
- Automatic assessment has same the level of detail for all students
- Students may repeat the exercises as often as they like

Participation

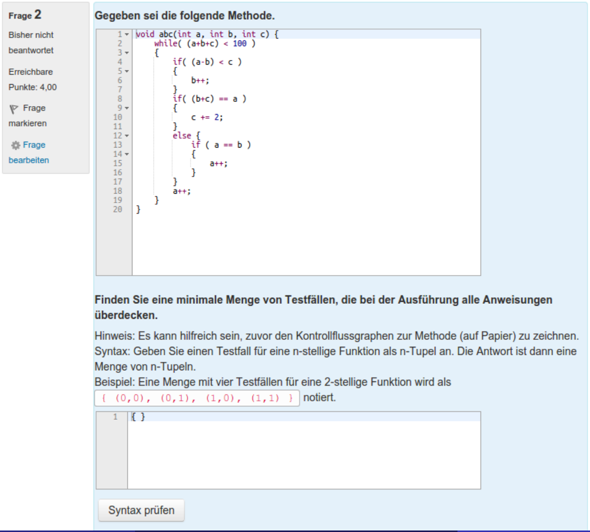
- 1st use: Winter Semester 2013/14 (1 / 6 online)
- 2nd use: Winter Semester 2014/15 (2 / 6 online)

Table: Number of students participating in exercises

Exercise	1	2	3	4	5	6
Regular, WS 13/14	127	122	101	**99**	118	52
Regular, WS 14/15	121	147	144	146	**148**	**81**
Addit., WS 14/15	**64**	**83**	**113**	**108**	–	–

Sample Exercise: Task

Give a *minimal set of test cases* that reaches a *full statement coverage*.



Frage 2

Bisher nicht beantwortet

Erreichbare Punkte: 4,00

Frage markieren

Frage bearbeiten

Gegeben sei die folgende Methode.

```
void abc(int a, int b, int c) {
    while( (a+b+c) < 100 )
    {
        if( (a-b) < c )
        {
            b++;
        }
        if( (b+c) == a )
        {
            c += 2;
        }
        else {
            if ( a == b )
            {
                a++;
            }
        }
        a++;
    }
}
```

Finden Sie eine minimale Menge von Testfällen, die bei der Ausführung alle Anweisungen überdecken.

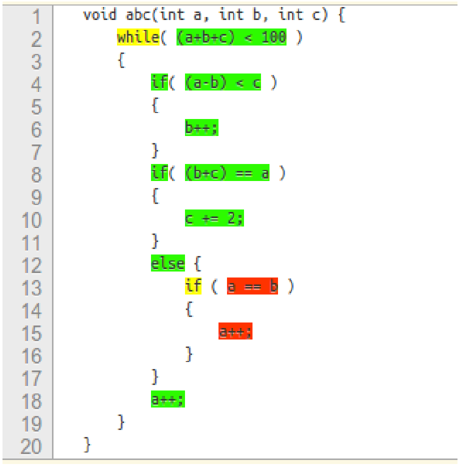
Hinweis: Es kann hilfreich sein, zuvor den Kontrollflussgraphen zur Methode (auf Papier) zu zeichnen.
Syntax: Geben Sie einen Testfall für eine n-stellige Funktion als n-Tupel an. Die Antwort ist dann eine Menge von n-Tupeln.
Beispiel: Eine Menge mit vier Testfällen für eine 2-stellige Funktion wird als `{ (0,0), (0,1), (1,0), (1,1) }` notiert.

```
{ }
```

Syntax prüfen

Sample Exercise: Solution

Example answer: $\{(1,2,0),(2,1,1)\}$



```
void abc(int a, int b, int c) {
    while( (a+b+c) < 100 )
    {
        if( (a-b) < c )
        {
            b++;
        }
        if( (b+c) == a )
        {
            c += 2;
        }
        else {
            if ( a == b )
            {
                a++;
            }
        }
        a++;
    }
}
```

Related Work

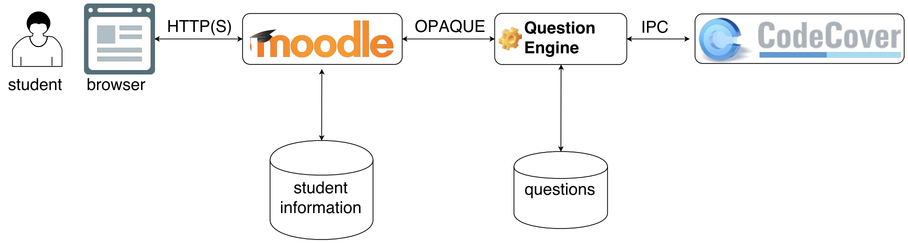
- Lots of work on automatic assessment of programming
  - dating back to 1960s, e.g.
    [Hollingsworth, 1960]
    [Forsythe and Wirth, 1965]
  - aim to develop the programming skills
- Most tools focus on
  assessing the quality of submitted code
- Task for testing is different
- Payed online courses available

Requirements

Requirements besides automatic correction:

1. The tool should improve students' experience
   - by allowing more creative questions
   - by giving detailed feedback on their solution
2. The tool should allow additional exercises for the students
3. A relationship between accounts in the system and the students' matriculation number is needed
4. The source code base which needs to be maintained should be as small as possible
5. The solution should be easily scalable to 400 students

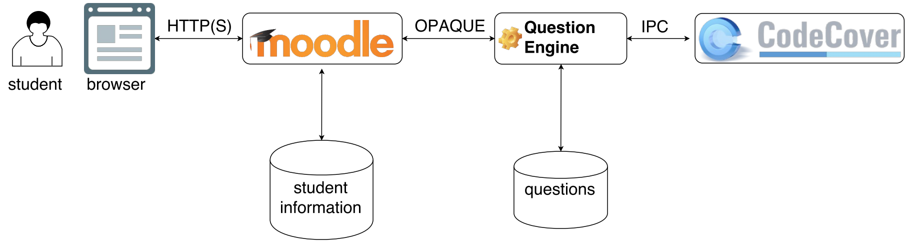
Architecture Overview: Moodle



Most principal decision:

- University's computing center already runs Moodle [Lopes, 2011]
  2 Exercise management
  3 Identity management
- Drawback: Only limited modules allowed to install
  1 Open Protocol for Accessing Question Engines (SOAP-based)
- Question Engine based on Activiti BPMN2.0 Process Engine

Architecture Overview: CodeCover



System building:

- CodeCover measures several code coverage metrics in the context of white-box testing.
  1. provides valuable feedback
  4. Open source under EPL
  4. Was under active development and maintainance

Performance ... later.

Evaluation Questions

- How were the exercises perceived by the students?
- Were there any technical obstacles while working on the exercises?
- ...

Questionnaire

Two versions:

- After exercises
- After feedback

Four parts:

- Demographics
- Likert-type scale questions (perception and time spent)
- Free text fields (improvements and shortcomings)
- Overall grade

Method:

- Voluntary
- Anonymous

$\rightarrow$ Multiple submissions possible

$\rightarrow$ No mapping between exercises and feedback questionnaires

Result:

- 105 completed questionnaires

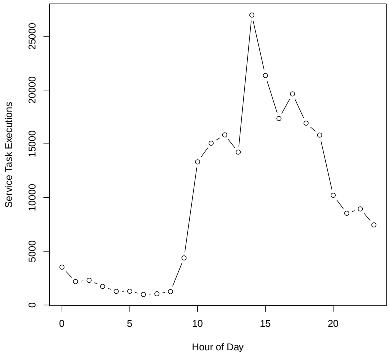
Qualitative Analysis

+ Advantages of digital submission (18)
+ Precise feedback (5)
+ Intuitive usability (5)
- Editing of submitted answers (7)
- Similarity of Assignments (6)
- Bad Performance (5)
- Indifferent Grading (5)
- Dowloading the exercises (4)

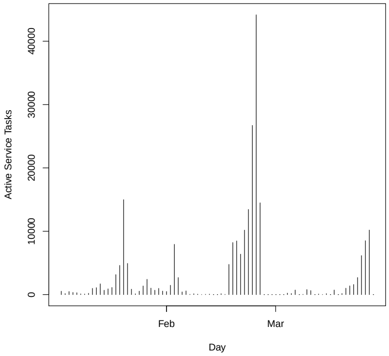
Threats to Validity

- Multiple submissions possible
- Qualitative reviews may be biased by analyzers
- One student gave positive feedback, but bad marks
- Many students did not take part in the last exercise
- Only voluntary feedback
- Qualitative analysis will be skewed towards more negative comments

Performance: Load Distribution

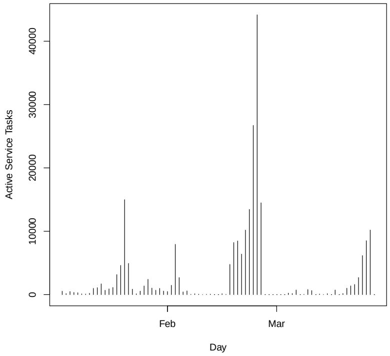


(a) Tasks by Hour of Day

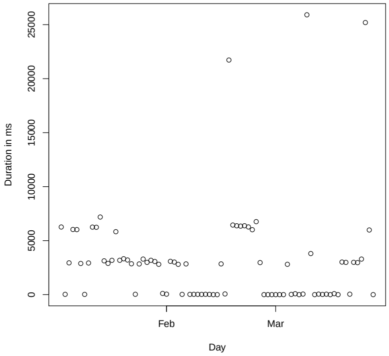


(b) Tasks per Day

Performance: Max. Task Duration



(b) Tasks per Day



(c) Max. Service Task Durations

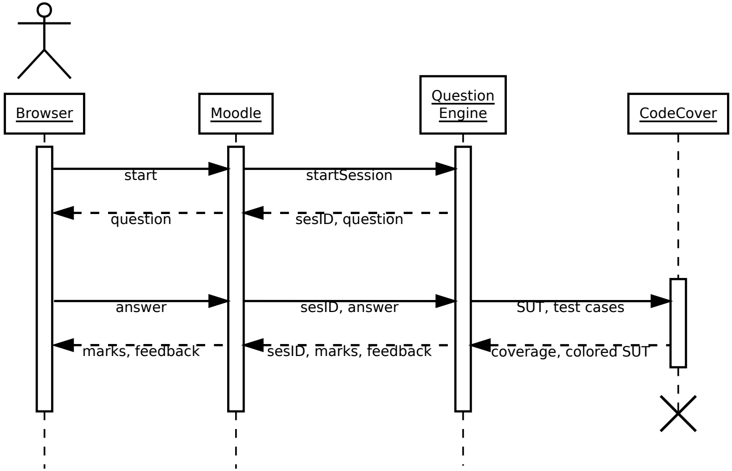
Conclusion

- Most students had a positive or neutral view
- Automation of assessment allows to free up teaching resources
- No serious technical or usability issues
- Feedback seemed helpful for most students, but could be more detailed
- System's performance sufficient

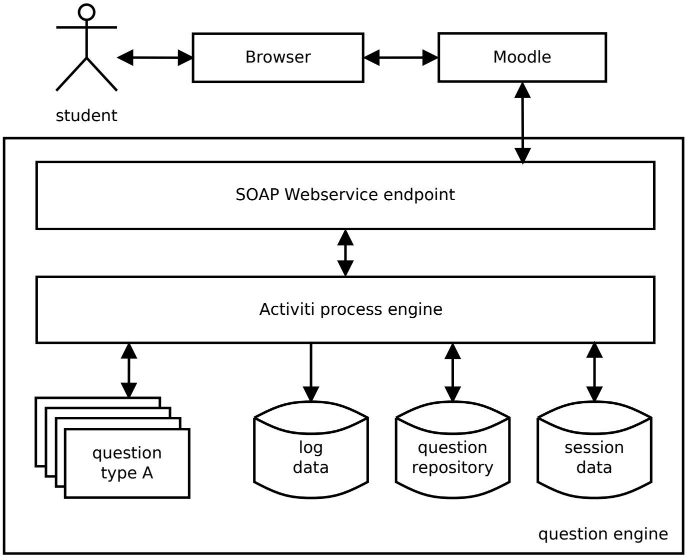
Future Work

- Performance improvements
- Facilitating the editing of answers
- Additional exercises: (e.g. UML modeling, OCL)
- Individual instances for each student
- Use as audience response system during lectures

Messages between browser, Moodle, Question Engine and CodeCover



Components of the Question Engine



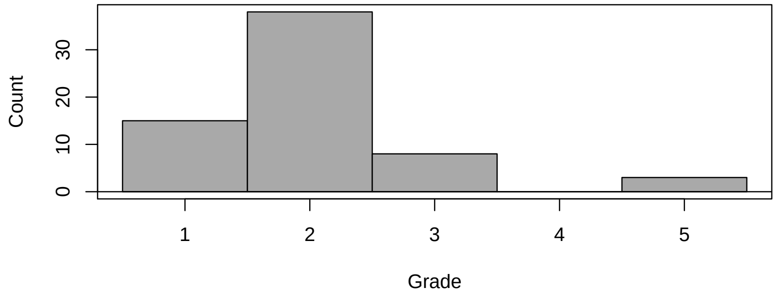
Evaluation Questions

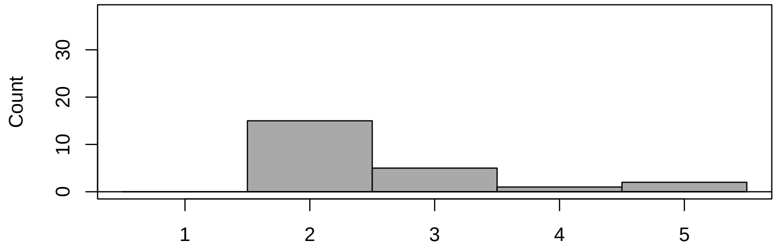
- Were the online exercises more or less demanding than the traditional exercises?
- How were the exercises perceived by the students?
- Would the students prefer more or less online exercises for future lectures?
- Were there any technical obstacles while working on the exercises?
- Were the additional voluntary exercises a helpful addition regarding the preparation for the final exam?

Questions with Likert-type scale

Question	++	+	o	-	- -
Online exercises required more effort than paper exercises.	6	3	18	23	15
The motivation to work with online exercises was higher than with paper exercises.	11	19	14	7	12
When working on the exercises, technical problems occurred.	3	2	4	5	53
The usability of the online system was good.	43	25	4	1	3
The feedback was helpful for understanding the exercise.	11	8	9	4	2
Feedback for online exercises was more detailed than for paper exercises.	3	3	6	5	6
Overall, I preferred the online exercises.	37	27	17	8	10

Grades after Exercises / Feedback





Qualitative Analysis (Overview)

The topics regarding positive effects of the system towards the students are more prominent (55%) than critical topics, which can be interpreted as a positive opinion of the students towards the system.

Table: Result of categorization of Feedback

Content\Technical	Pos.	Neutr.	Neg.	No Feedback
Positive	4	1	0	1
Neutral	3	5	0	1
Negative	6	6	2	5
No Feedback	6	9	7	48

Forsythe, G. E. and Wirth, N. (1965).
Automatic grading programs.
Commun. ACM, 8(5):275–278.

Hollingsworth, J. (1960).
Automatic graders for programming classes.
Commun. ACM, 3(10):528–529.

Lopes, A. P. F. F. (2011).
Teaching with Moodle in higher education.
Technical report, Institute of Accounting and Administration (ISCAP), Polytechnic Institute of Oporto (IPP).