Improving Engagement of Students in Software Engineering

Tanja Vos, Open Universiteit
Wishnu Prasetya, Universiteit Utrecht

Tanja.Vos@ou.nl, S.W.B.Prasetya@uu.nl
https://impress-project.eu/

European Commission
ERASMUS+

Project 2017-1-NL01-KA203-035259

IMPRESS

- A recently started EU-funded project aiming at improving students' engagement in Software Engineering courses through gamification.

- Sept. 2017 – Aug. 2020

Software engineering

- Customers want to have quality products
- Bosses want to make money
- Engineers want to build wonder

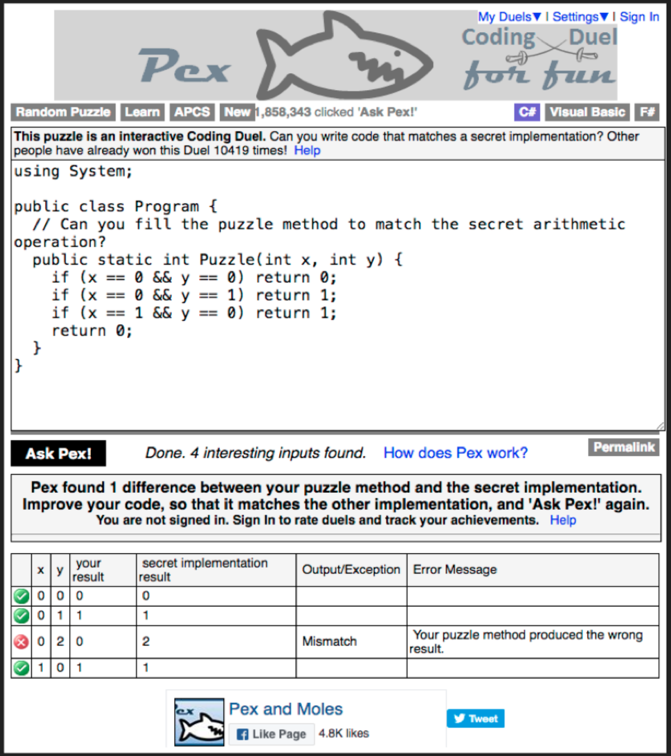
Teaching software engineering

- waterfall, iterative, agile
- 14 UML diagram types
- 23 design patterns
- over 80 refactorings
- ...

IMPRESS

- Can gamification improve the engagement in SE courses?
- Different level of gamification:
  - Gamified class room SE quizzes
  - Serious SE education game
  - SE education games
- Two additional aspects: integrated analytics and AI/automation to reduce teachers' effort.

Examples of SE education games



1,858,343 clicked 'Ask Pex!'

This puzzle is an interactive Coding Duel. Can you write code that matches a secret implementation? Other people have already won this Duel 10419 times! Help

```
using System;

public class Program {
  // Can you fill the puzzle method to match the secret arithmetic
operation?
  public static int Puzzle(int x, int y) {
    if (x == 0 && y == 0) return 0;
    if (x == 0 && y == 1) return 1;
    if (x == 1 && y == 0) return 1;
    return 0;
  }
}
```

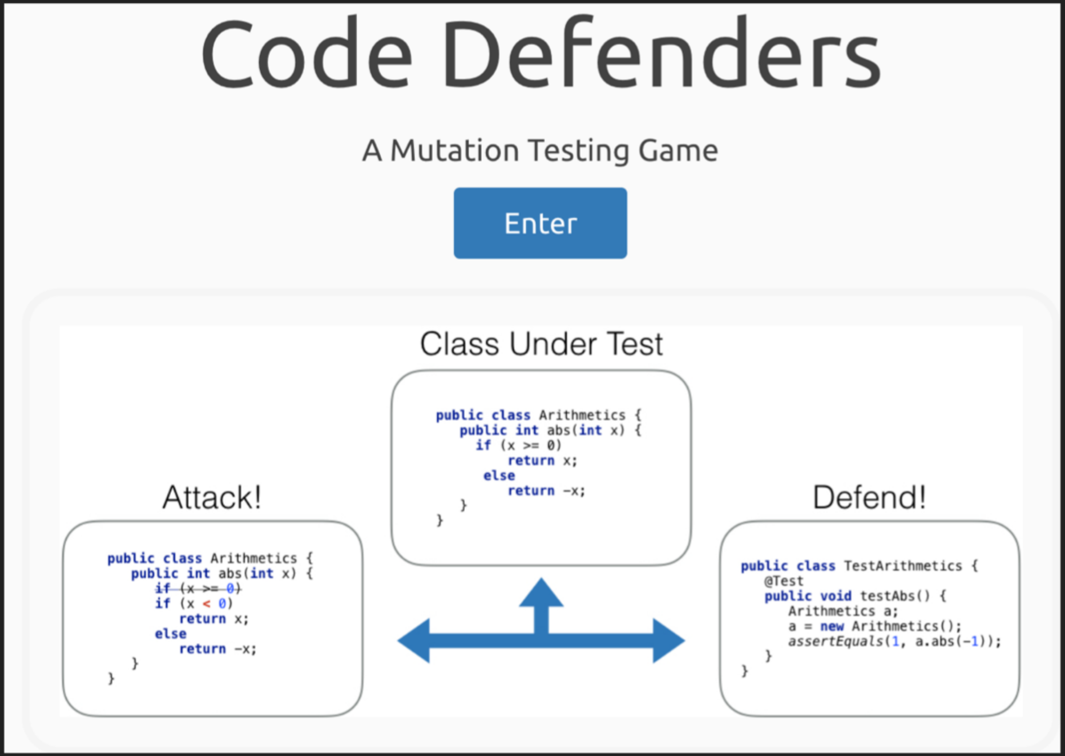
Done. 4 interesting inputs found. How does Pex work?

Pex found 1 difference between your puzzle method and the secret implementation.
Improve your code, so that it matches the other implementation, and 'Ask Pex!' again.
You are not signed in. Sign In to rate duels and track your achievements. Help

	x	y	your result	secret implementation result	Output/Exception	Error Message
✓	0	0	0	0		
✓	0	1	1	1		
✗	0	2	0	2	Mismatch	Your puzzle method produced the wrong result.
✓	1	0	1	1		

Pex and Moles
Like Page 4.8K likes
Tweet

Pex (Microsoft)



```
public class Arithmetics {
   public int abs(int x) {
     if (x >= 0)
         return x;
      else
         return -x;
   }
}
```

```
public class Arithmetics {
   public int abs(int x) {
      ~~if (x >= 0)~~
      if (x < 0)
         return x;
      else
         return -x;
   }
}
```

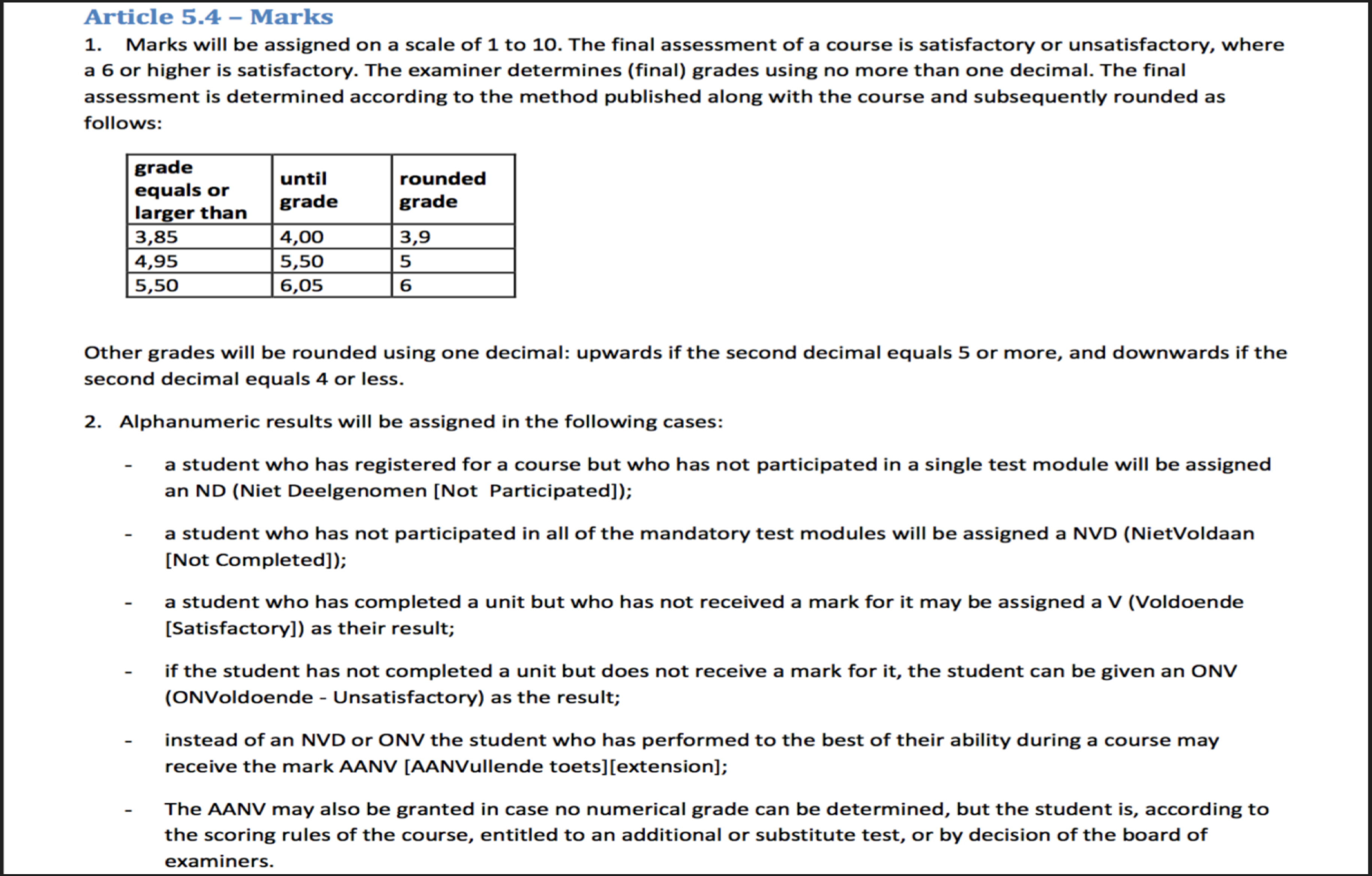
```
public class TestArithmetics {
   @Test
   public void testAbs() {
      Arithmetics a;
      a = new Arithmetics();
      assertEquals(1, a.abs(-1));
   }
}
```

Code Defenders (IMPRESS)

IMPRESS case study: quality assurance

- Customers want to have quality product
- Engineers should not only “develop”
- They also need to **test** the modules they build
- ... and invest in **formalizing** the modules’ specification

Or shall we just leave it informal?



Article 5.4 – Marks

1. Marks will be assigned on a scale of 1 to 10. The final assessment of a course is satisfactory or unsatisfactory, where a 6 or higher is satisfactory. The examiner determines (final) grades using no more than one decimal. The final assessment is determined according to the method published along with the course and subsequently rounded as follows:

grade equals or larger than	until grade	rounded grade
3,85	4,00	3,9
4,95	5,50	5
5,50	6,05	6

Other grades will be rounded using one decimal: upwards if the second decimal equals 5 or more, and downwards if the second decimal equals 4 or less.

2. Alphanumeric results will be assigned in the following cases:

- a student who has registered for a course but who has not participated in a single test module will be assigned an ND (Niet Deelgenomen [Not Participated]);
- a student who has not participated in all of the mandatory test modules will be assigned a NVD (NietVoldaan [Not Completed]);
- a student who has completed a unit but who has not received a mark for it may be assigned a V (Voldoende [Satisfactory]) as their result;
- if the student has not completed a unit but does not receive a mark for it, the student can be given an ONV (ONVoldoende - Unsatisfactory) as the result;
- instead of an NVD or ONV the student who has performed to the best of their ability during a course may receive the mark AANV [AANVullende toets][extension];
- The AANV may also be granted in case no numerical grade can be determined, but the student is, according to the scoring rules of the course, entitled to an additional or substitute test, or by decision of the board of examiners.

A lesson in writing formal specifications

- We can write **simple expressions**:
  - constants like 1,2,3
  - identifiers like x,y,Students
  - properties, e.g. x.age, y.goal
  - $e_1 \otimes e_2$ where $\otimes$ is $+$, $-$, $*$, $=$, $>$, $\geq$, $<$, $\leq$, $\in$
- A **simple formula** is a simple expression of type Boolean

A lesson in writing formal specifications

- A **formula** is either:
  - a simple formula
  - $\forall$identifier$\in$simple-expression● formula
  - $\exists$identifier$\in$simple-expression● formula)
- For example:
  - $\forall x \in$ Students● $x.age \geq 16$
  - $\exists x \in$ Students● $x.age = 16$

A lesson in writing formal specifications

Let’s try something different....

kahoot.it

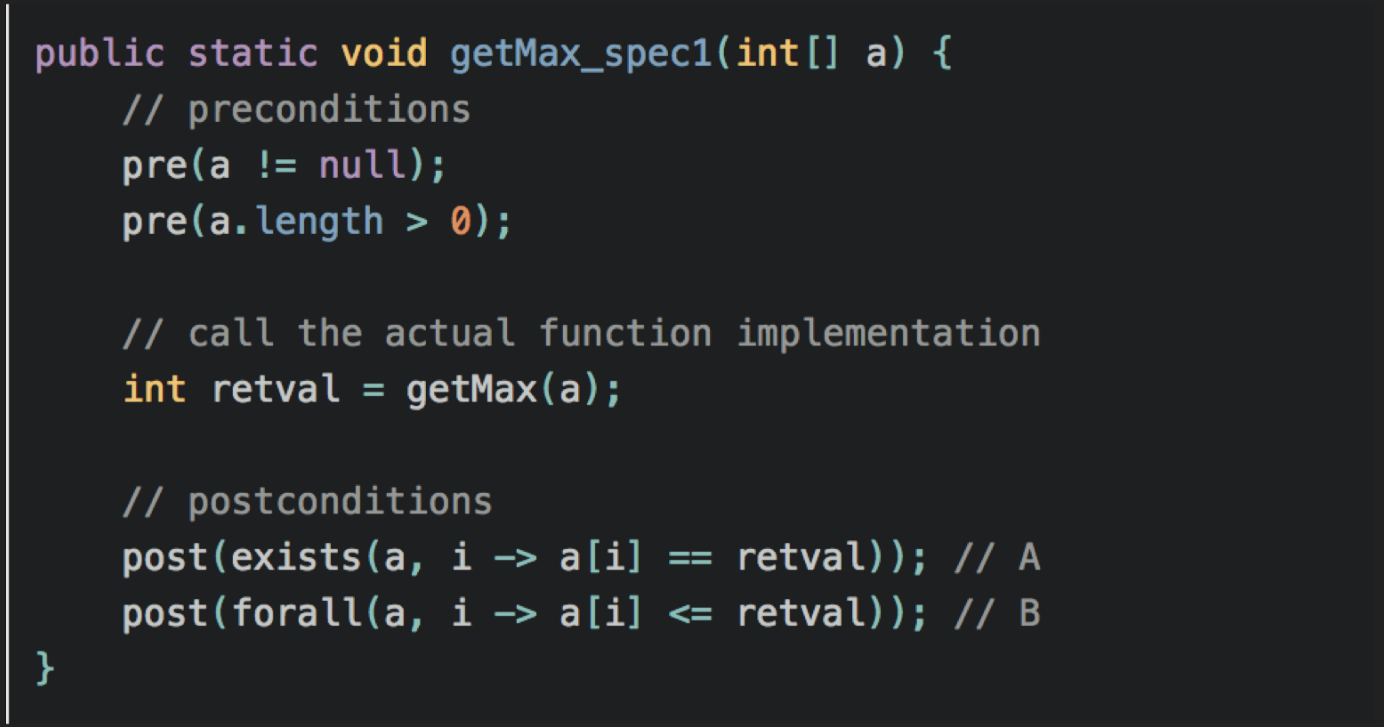
re-order the symbols,
make something
sensical

All students have the same goal.

(assume we have at least one student)

In production: FormalZ game

- a game to train student to write formal specifcations interpretable in Java
- will lean more towards the “engagement” aspect
- https://git.science.uu.nl/impresshs/javawlp



```java
public static void getMax_spec1(int[] a) {
    // preconditions
    pre(a != null);
    pre(a.length > 0);

    // call the actual function implementation
    int retval = getMax(a);

    // postconditions
    post(exists(a, i -> a[i] == retval)); // A
    post(forall(a, i -> a[i] <= retval)); // B
}
```

IMPRESS future work

- Education quizzes and games for Software Engineering, experimenting with the balance between “seriousness” and “excitement”.
- Data analytics.
- Studying these innovations in actual class rooms.
- If you are interested: Tanja.Vos@ou.nl