



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

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/>



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

The screenshot shows the Pex Coding Duel for fun website. At the top, there's a navigation bar with "My Duels", "Settings", and "Sign In". The main header features the "Pex" logo and "Coding Duel for fun". Below this, there's a "Random Puzzle" button and a "Learn" section with "APCS" and "New" tabs. A notification says "1,858,343 clicked 'Ask Pex!'". There are also "C#", "Visual Basic", and "F#" buttons. The main content area contains a puzzle description: "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". The puzzle code is as follows:

```
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;
    }
}
```

Below the code, there's an "Ask Pex!" button and a message: "Done. 4 interesting inputs found. How does Pex work? Permalink". A summary message says: "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		

At the bottom, there's a "Pex and Moles" section with a "Like Page" button (4.8K likes) and a "Tweet" button.

Pex (Microsoft)

The diagram illustrates the Code Defenders game interface. At the top, the title "Code Defenders" is displayed, followed by the subtitle "A Mutation Testing Game". A blue "Enter" button is positioned below the subtitle. The main area is divided into three sections: "Attack!", "Class Under Test", and "Defend!".

The "Attack!" section contains the following code:

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

The "Class Under Test" section contains the following code:

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

The "Defend!" section contains the following code:

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

Blue arrows indicate the flow of the game: a horizontal arrow points from "Attack!" to "Class Under Test", and a vertical arrow points from "Class Under Test" to "Defend!".

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
 - $\forall \text{identifier} \in \text{simple-expression} \bullet \text{formula}$
 - $\exists \text{identifier} \in \text{simple-expression} \bullet \text{formula}$)
- For example:
 - $\forall x \in \text{Students} \bullet x.\text{age} \geq 16$
 - $\exists x \in \text{Students} \bullet x.\text{age} = 16$

A lesson in writing formal specifications



Let's try something different....

kahoot.it

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 specifications interpretable in Java
- will lean more towards the “engagement” aspect
- <https://git.science.uu.nl/impresshs/javawlp>

```
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