



Creation of a Serious Game For Teaching Code Review

Presenter: Barış Ardiç



Barış Ardiç



İrem Yurdakul



Eray Tüzün

Outline

- I. Serious Games
- II. Code Review Process
- III. Workflow of the study
- IV. Learning Objectives
- V. Game Flow
- VI. Preliminary Experiment
- VII. Future Directions



Serious Games

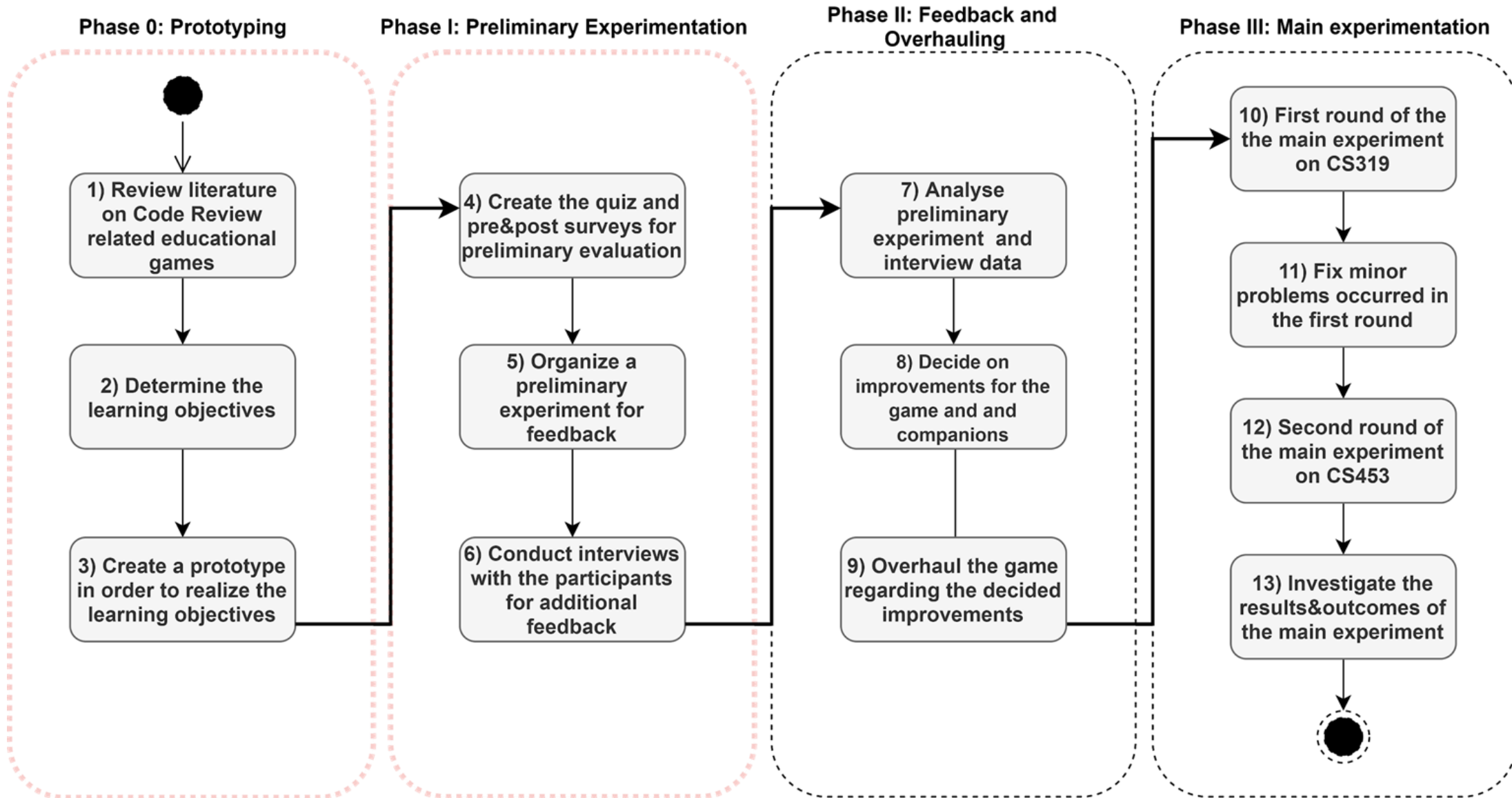
- A game with a different main purpose than entertainment.
- A concept similar to gamification.
- Game elements are parts of the core functionality.
- Aim is to provide a platform that introduces code review while allowing students to practice.

Code Review Process

- A manual inspection of source code by developers other than the author.
- Frequently used practice in software development
- Improves code quality if done **right**.
- Often not addressed in software engineering curriculums.

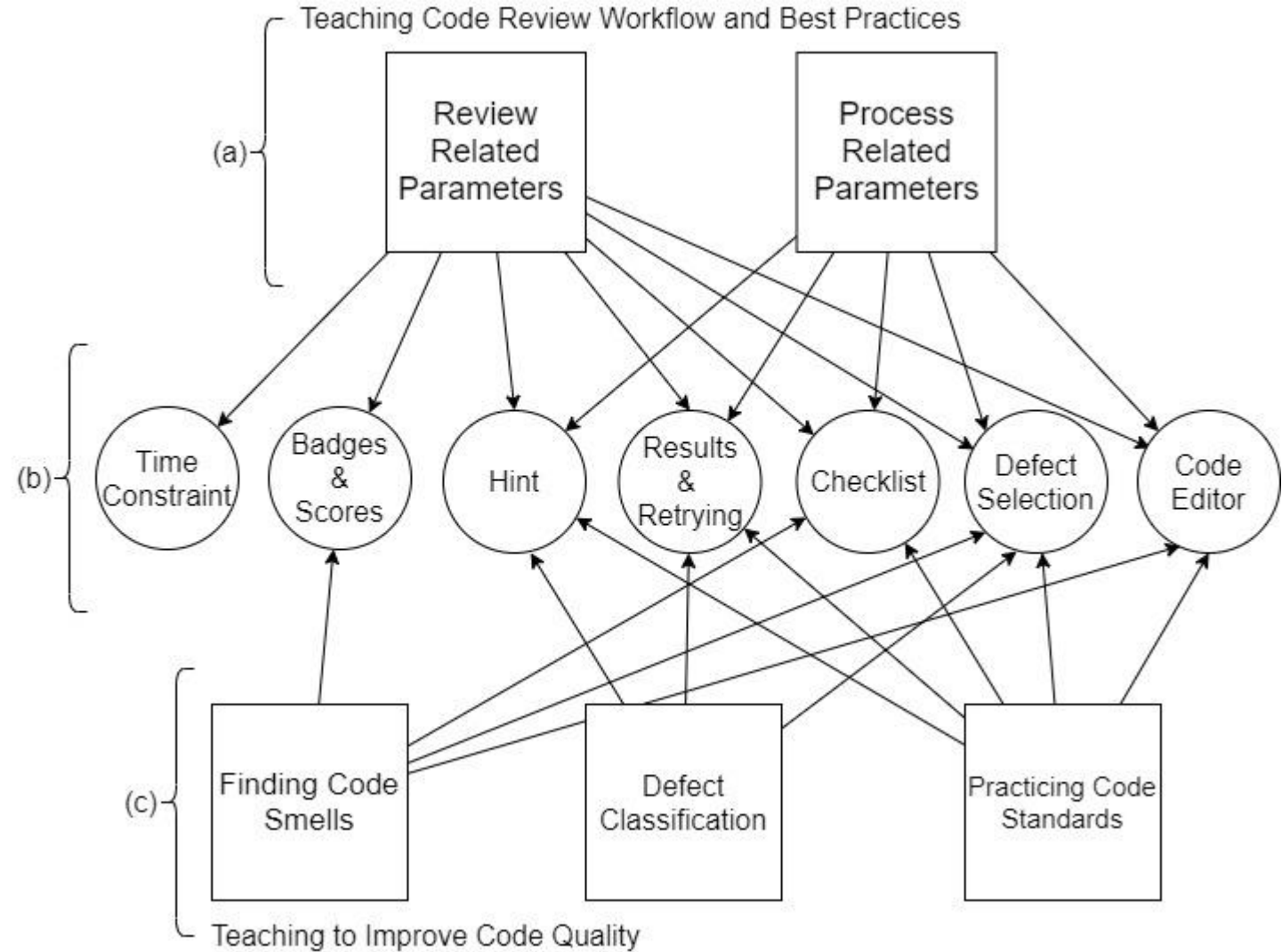


Workflow of the Study



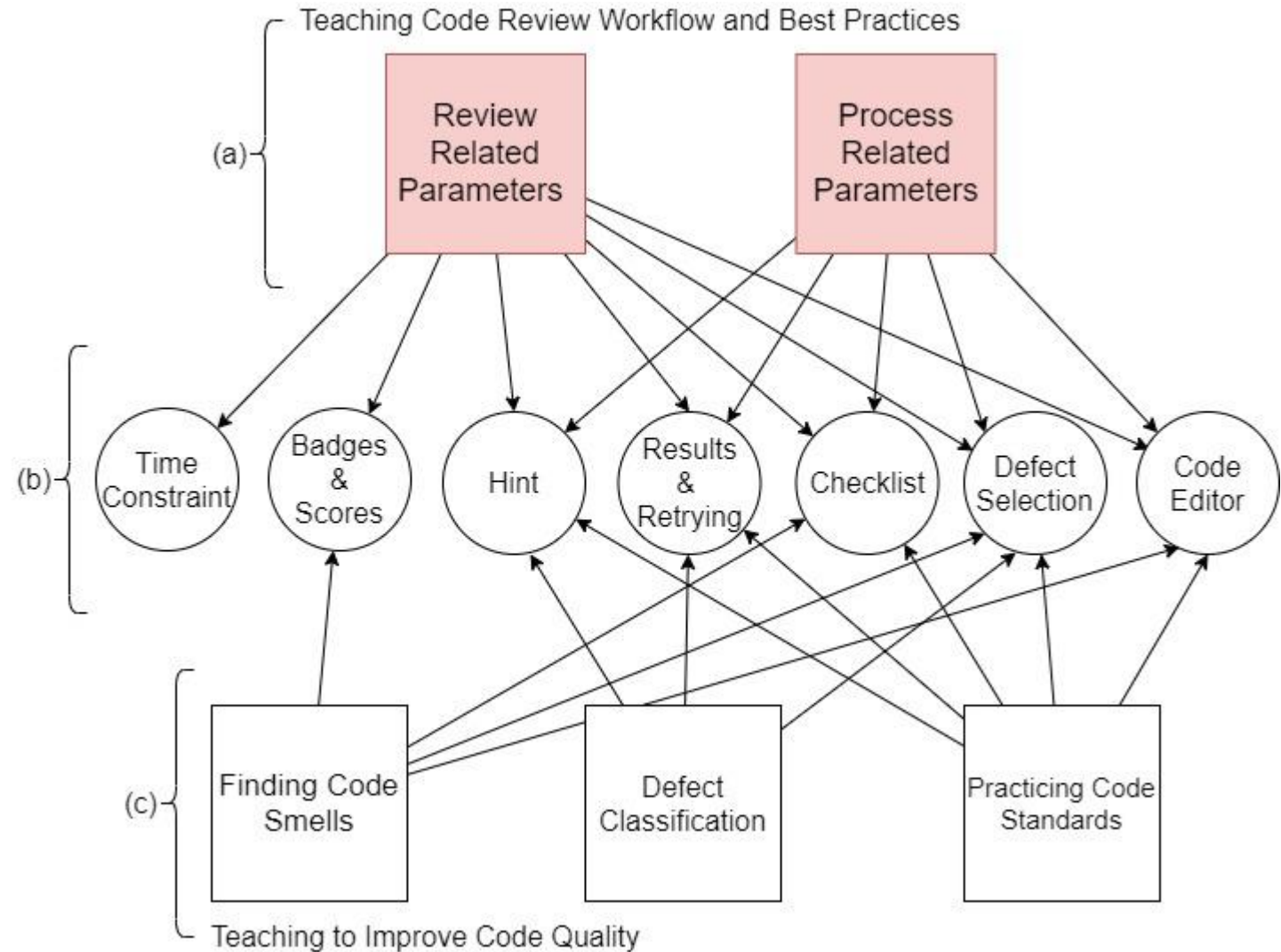
Learning Objectives

- Main objectives are rows A and C.
- B consists of game concepts.
- Objectives are realised by game concepts.



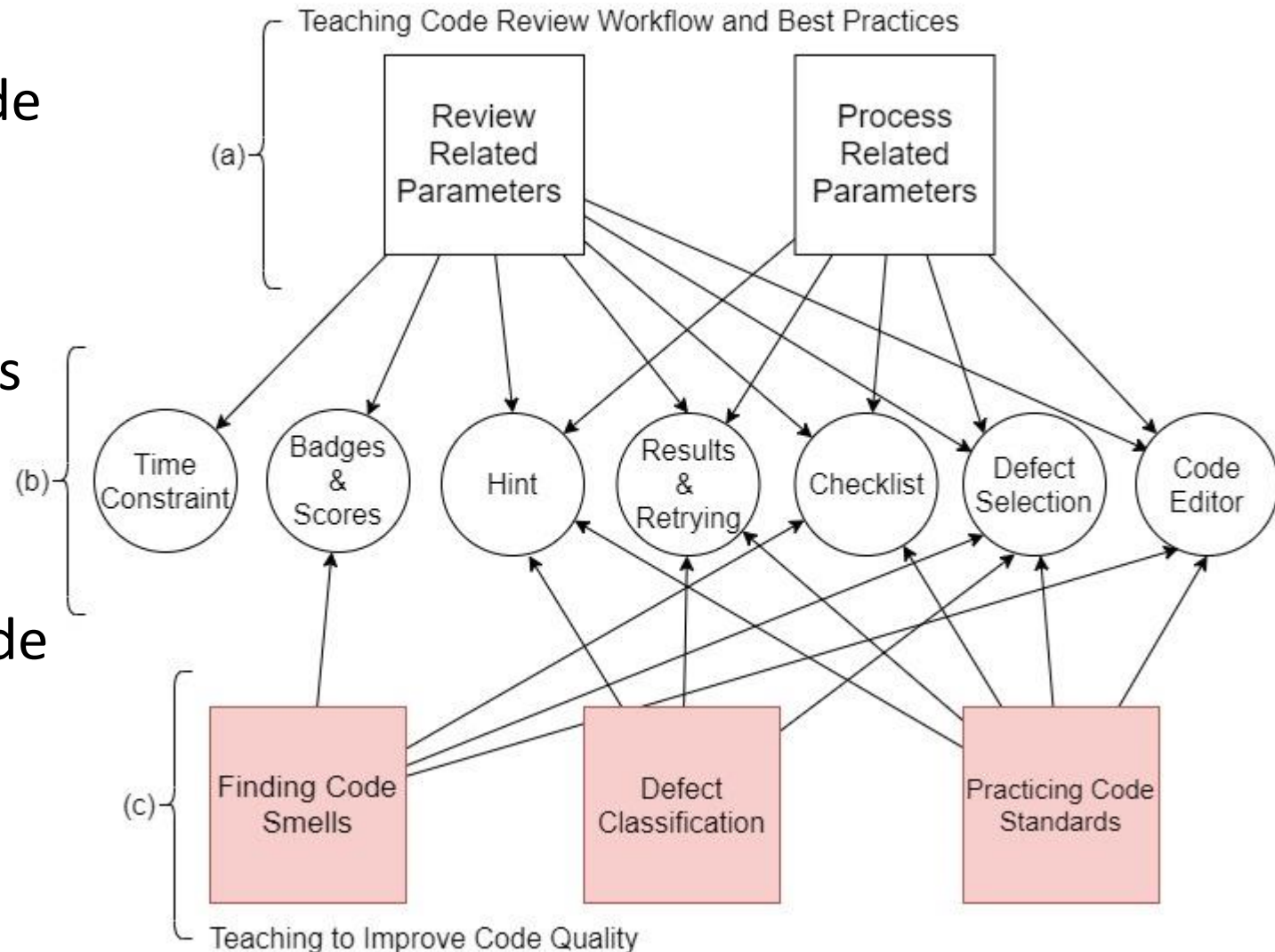
Learning Objectives

- Review Related Parameters:
 - Review size
 - Time spent on review
- Process Related Parameters:
 - Review workflow
 - Actors and responsibilities



Learning Objectives

- Finding smells improve code quality.
- Defect classification^[4] helps communication.
- Code standards help provide evolvability.



[4] M. Mantyla and C. Lassenius, "What types of defects are really discovered in code reviews?" *IEEE Trans. Software Eng.*, vol. 35, no. 3, pp. 430–448, 2009.

Game Flow

← Defects you find will go here!

| | | |
|---------------------|--------------------------------|-------|
| Error Lines : 17-23 | Data and resource manipulation | 👁️ 🗑️ |
| Error Lines : 10-10 | Compare | 👁️ 🗑️ |
| Error Lines : 8-8 | Immutable | 👁️ 🗑️ |
| Error Lines : 6-6 | Element Type | 👁️ 🗑️ |

SUBMIT SELECTION DESCRIPTION GUIDE CHECKLIST HINT SEE ANSWERS

Last level of the experiment is **HERE** and about prime numbers!!

PS: If a line of code is missing you should select the nearest encapsulating object structure, eg the nearest curly brackets, for example if a return statement is missing inside an "if" statement you should select the whole if statement. The reason of the defect for such cases should be "Data and resource manipulation"

```
1
2 public class Primes{
3     public static void main(String[] args) throws Exception {
4
5         final int twoToThePowerThirty = 1073741824;
6         List listOfPrimes = new ArrayList();
7         String title = "Printing all integer prime numbers";
8         title.toUpperCase();
9         System.out.println(title);
10        for (int number = largestPrimeSoFar; number<=Integer.MAX_VALUE; number++){
11            if (isPrime(number)) {
12                System.out.println(number);
13                listOfPrimes.add(number);
14            }
15        }
16        int isLarger = (listOfPrimes.get(listOfPrimes.size()-1)>twoToThePowerThirty) ? 1 : 0;
17        switch (isLarger) {
18            case false:
19                System.out.print("Largest prime integer is not bigger than 2^30");
20            case true:
21                System.out.print("Largest prime integer is bigger than 2^30");
22                break;
23        }
24    }
25    /*
26     * Prime number is not divisible by any number other than 1 and itself
27     * @return true if number is prime
28     */
29    public static boolean isPrime(int number) {
30        for (int i = 2; i < number; i++) {
31            if (number % i == 0) {
32                return false; // number is divisible so its not prime
33            }
34        }
35        return true; // number is prime now
36    }
37 }
38
```



Defect Selection

The drop-down menu represents the defect classification taxonomy



Error Lines : 40&43

Error Reason ▾

Evolvability Defects

Structure

Visual Representation

Functional Defects

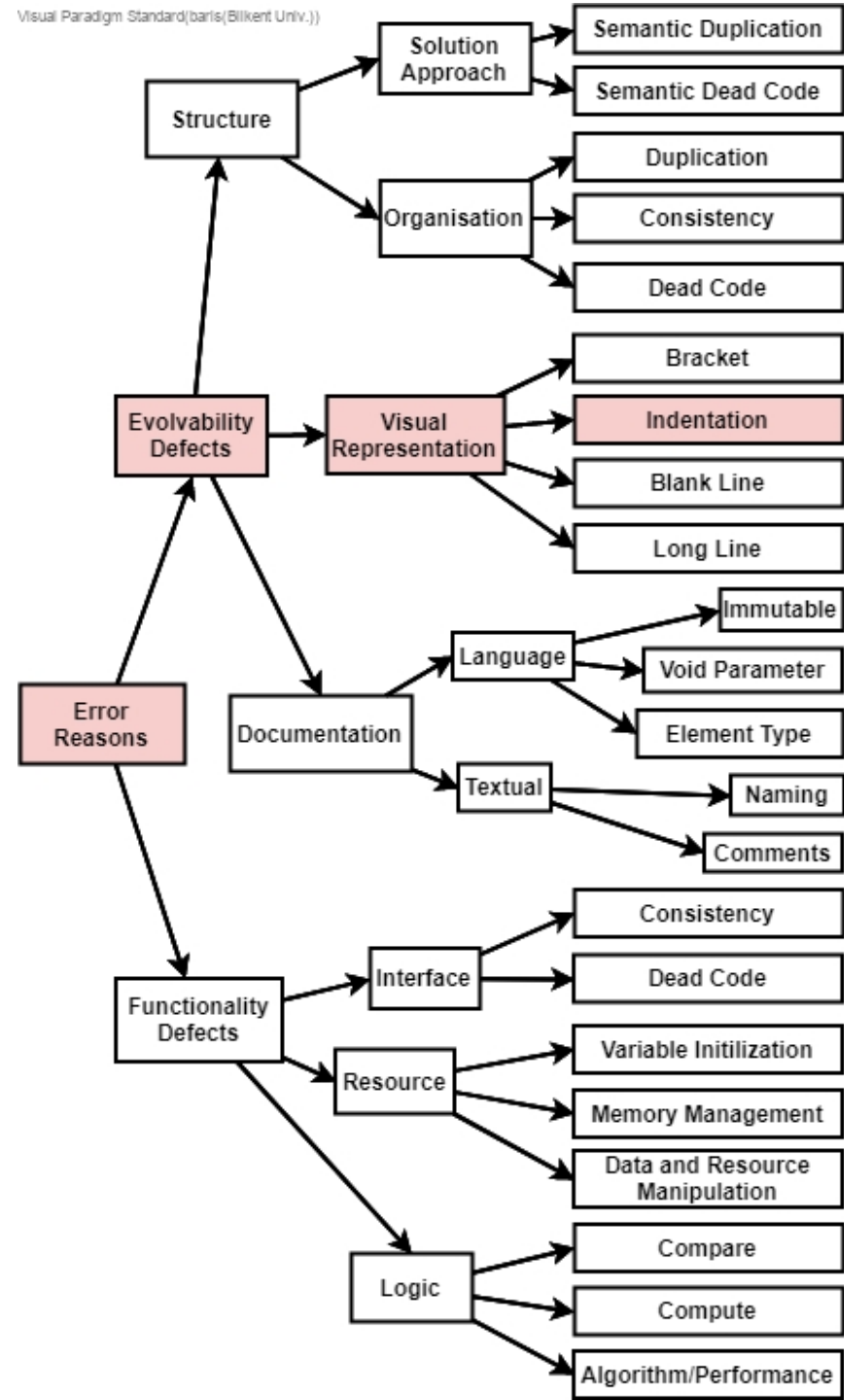
Documentation

Bracket

Indentation

Blank line usage

Long line



Validation Strategy



- Not a straightforward process.
- Lack of validation is prevalent in the field.
- We came up with a strategy inspired by social sciences.
- Created an experiment for measuring differences after playing.
- Test -> Play -> Test

Preliminary Experiment Setting

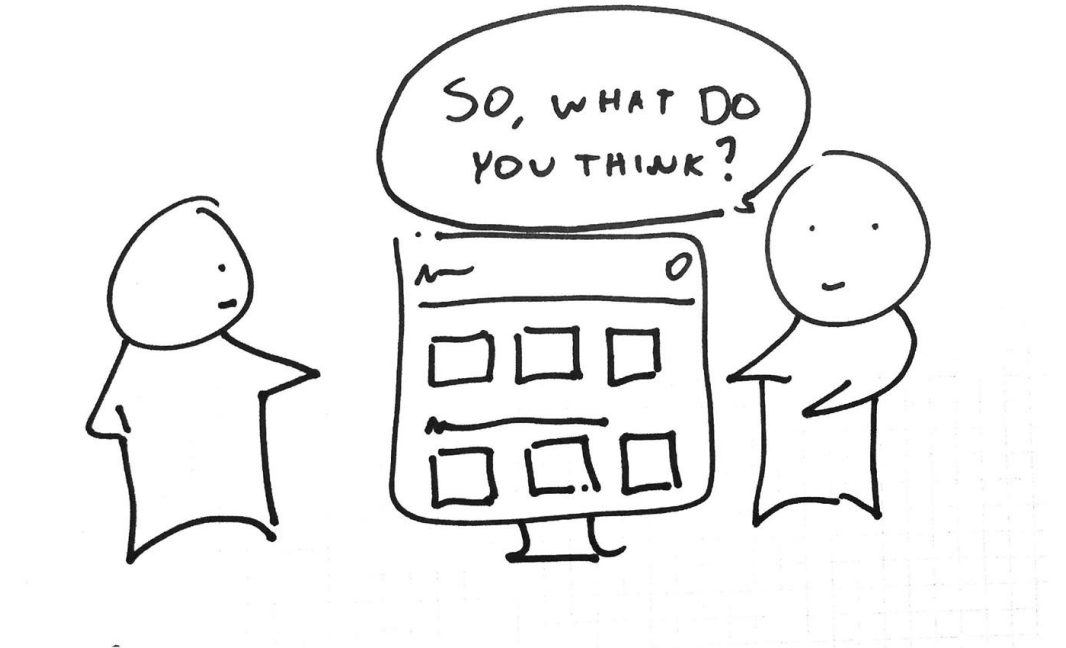
- Pre-Survey : Overall and CR experience
- Pre-Quiz : CR knowledge and programming
- Gameplay: 5 levels
- Post-Quiz: Same as Pre-Quiz
- Post-Survey: Feedback and familiarity
- Interviews: Rate quiz questions

TABLE I
EXPERIMENT SETUP DURATIONS

| Phases of Experiment | Min. Duration | Max. Duration |
|----------------------|---------------|---------------|
| Pre-Survey | 5 min. | 8 min. |
| Pre-Quiz | 19 min. | 34 min. |
| Play Session | 35 min. | 61 min. |
| Post Quiz | 7 min. | 13 min. |
| Post Survey | 5 min. | 20 min. |

Feedback From Participants

- Detected some bugs and inconsistencies.
- Improved the game content.
- Added feature requests like:
 - Multiple reason support
 - Tutorial
- Adjusted the experiment duration by omitting or adding quiz questions based on interviews.



Later Phases

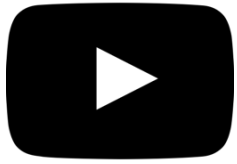
- Enhanced the game according to the preliminary feedback
- Utilized in game as a lab assignment in Bilkent University
- Followed the same experiment format.
- The lab had 132 participants and was received positively.
- Journal paper with the improved game and main experiment statistics and is in review.



Future Directions

- We intend to add collaborative gameplay for knowledge transfer and team building.
- An additional game mode from the authors perspective.
- Code base can be extended for other SE processes.
- More experiments are on their way as well.

Thank You for Listening



Video Demo



Github Page



Full Paper

