SFWR ENG 3S03
Software Testing

A. Marinache

Outline

Preliminaries

What Is Software
Testing?

Challenges of
Software Testing

Simple Example
Illustrating the
Challenge of
Testing

Roadmap

SFWR ENG 3S03
Software Testing

Alicia Marinache

Department of Computing and Software, McMaster University
Canada L8S 4L7, Hamilton, Ontario

SFWR ENG 3S03
Software Testing

A. Marinache

Outline

Preliminaries

What Is Software
Testing?

Challenges of
Software Testing

Simple Example
Illustrating the
Challenge of
Testing

Roadmap

Exciting Exercise:
https://bit.ly/40kzgjk

Introduction to Testing and Measurement
➡Preliminaries

Example (Explosion of the rocket Ariane 5 –June 4, 1996–)

On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in a failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded.

Example (Mars Polar Lander crash – December 3, 1999 –)

The Mars Polar Lander was a robotic spacecraft lander launched by NASA on January 3, 1999 to study the soil and climate of a region near the south pole on Mars. After the descent phase was expected to be complete, the lander failed to reestablish communication with Earth. Moments later it crashed into Mars at a destructive speed of about 22 meters per second. [Cow08]

Other famous (more recent) examples

- August 2003 – the Northeast blackout

- March 2014 – Nissan recall for airbags malfunction

- May 2015 – Airbus crash

- March 2017 – Bitcoin Unlimited node crash

- June 2024 – MS/CrowdStrike Global Outage

- Good testing is difficult

- It requires careful planning and documentation

- There exists a large number of test techniques that we will study

- We will discuss the major classes of test techniques with their characteristics

An obvious definition

Definition

Software Testing is the process of executing a program on some inputs in order to find errors.

Leaves questions unanswered

- How do we choose the inputs?

- How do we recognise errors?

Doesn't cover all testing

- Not all testing is at the program level

- Possible to test the code without executing it

- Can test for reasons other than fault discovery

Introduction to Testing and Measurement

➡ What Is Software Testing?

A better definition

Definition

Software Testing is the process of evaluating software in order to derive an estimate of whether or not it meets some criteria

➡️What Is Software Testing?

SFWR ENG 3S03
Software Testing

A. Marinache

Outline

Preliminaries

What Is Software
Testing?

Challenges of
Software Testing

Simple Example
Illustrating the
Challenge of
Testing

Roadmap

Still some questions unanswered

- What is evaluated?

- When is it evaluated?

- How is it evaluated?

- What criteria must it meet?

- How do we know if it meets the criteria?

Introduction to Testing and Measurement

E. Djikstra's famous quote (1969)

"Testing shows the presence, not the absence of bugs"

`https://en.wikiquote.org/wiki/Edsger_W._Dijkstra#1960s`

SFWR ENG 3S03
Software Testing

A. Marinache

Outline

Preliminaries

What Is Software
Testing?

Challenges of
Software Testing

Simple Example
Illustrating the
Challenge of
Testing

Roadmap

What *can* testing do then?

- Testing can show software *works*
  - i.e. that it sometimes does what we want

- Testing can show software *doesn't work*
  - i.e. that it sometimes does something bad

- Testing can *reduce the chance* that software doesn't work
  - i.e. it can build confidence that key properties hold

Would you...?[Bab82]

`http://bit.ly/3C3yjST`

Would you...?[Bab82]

`http://bit.ly/3C3yjST`

- What is the quality of code? studies indicate 10 to 50 errors per KLOC during in-house testing, and 0.5 to 3 errors per KLOC in released products

Introduction to Testing and Measurement

➥Challenges of Software Testing

SFWR ENG 3S03
Software Testing

A. Marinache

Main challenges

- Software complexity

- Software visibility

- Nature of the development process

Several other factors add to the challenges

- The culture within software companies

- Managers dismiss the seriousness of testing

- Large number of programming languages

- Advances in technology

Introduction to Testing and Measurement
➡ Simple Example Illustrating the Challenge of Testing

Example

Write a set of test cases to adequately test the following quite simple Java program. You were given the following requirements:

> *The program reads three integer values from an input dialog. The three values represent the lengths of the sides of a* **triangle**. *The program displays a message that states whether the triangle is* **scalene**, **isosceles**, *or* **equilateral**.

- What does the tester need to know to properly test a program for this specification?
- Geometry theory about triangles + a precise requirements (no ambiguity)

EQUILATERAL		All the sides are the same length (shown by th line through each of them) and all the angles a the same length. $\angle A = \angle B = \angle C = 60°$
ISOSCELES		Two sides are equal (shown by the lines). Angles opposite the equal sides are equal. $\angle B = \angle C$
SCALENE		No sides are equal. No angles are equal. $\angle A \neq \angle B \neq \angle C$

Introduction to Testing and Measurement

➡️ Simple Example Illustrating the Challenge of Testing

```
1  /* Specification: The program reads three integer values
      from an input dialog. The three values represent
      the lengths of the sides of a triangle. The program
      displays a message that states whether the triangle
      is scalene, isosceles, or equilateral.
3  */

5  import java.util.Scanner;
   public class TriangleType {

       static Triangle getType(int a, int b, int c)
9      {
           if(a<=0||b<=0||c<=0)
11              throw new IllegalArgumentException("Length
                    of sides cannot be equal to or less
                    than zero");

13         if(a==b&&b==c&&c==a)
               return Triangle.EQUILATERAL;
15         else if((a==b)||(b==c))
               return Triangle.ISOSCELES;
17         else if(a!=b&&b!=c&&c!=a)
               return Triangle.SCALENE;
19         else
               return Triangle.ERROR;
21     }

23     public static void main(String[] args)
       {
25         System.out.println("IS THIS A CORRECT PROGRAM
                WITH REGARD TO THE ABOVE SPECIFICATION?");
           Scanner keyboard = new Scanner(System.in);
27         System.out.println("Read the first side length:
                ");
           int a = keyboard.nextInt();
29         System.out.println("Read the second side length:
                ");
           int b = keyboard.nextInt();
31         System.out.println("Read the third side length:
                ");
           int c = keyboard.nextInt();
33         System.out.println(TriangleType.getType(a, b,
                c));
35     }
   }
37
   enum Triangle
39 {
   ISOSCELES(0),
41 EQUILATERAL(1),
   SCALENE(2),
43 ERROR(3);

45 private int n;
   Triangle(int n)
47 {this.n = n;}
   }
```

Theorem (Triangle inequality)

If a, b, and c are the lengths of the sides of a triangle, then
$a + b > c \ \wedge \ a + c > b \ \wedge \ b + c > a$

$\textbf{Triangle} \implies a + b > c \ \wedge \ a + c > b \ \wedge \ b + c > a$

$\iff \quad \langle \text{ Contrapositive } \rangle$

$\neg(a + b > c \ \wedge \ a + c > b \ \wedge \ b + c > a) \implies \neg\text{Triangle}$

$\iff \quad \langle \text{ De Morgan Law } \rangle$

$\neg(a + b > c) \vee \neg(a + c > b) \vee \neg(b + c > a) \implies \neg\text{Triangle}$

- Moreover, there is an ambiguity in the requirements (short integer, integer, or long integer?)

```java
/* Specification: The program reads three integer values
   from an input dialog. The three values represent
   the lengths of the sides of a triangle. The program
   displays a message that states whether the triangle
   is scalene, isosceles, or equilateral.
*/

import java.util.Scanner;
public class TriangleType {

    static Triangle getType(int a, int b, int c)
    {
        if(a<=0||b<=0||c<=0)
            throw new IllegalArgumentException("Length
                of sides cannot be equal to or less
                than zero");

        if(a==b&&b==c&&c==a)
            return Triangle.EQUILATERAL;
        else if((a==b)||(b==c))
            return Triangle.ISOSCELES;
        else if(a!=b&&b!=c&&c!=a)
            return Triangle.SCALENE;
        else
            return Triangle.ERROR;
    }

    public static void main(String[] args)
    {
        System.out.println("IS THIS A CORRECT PROGRAM
            WITH REGARD TO THE ABOVE SPECIFICATION?");
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Read the first side length:
            ");
        int a = keyboard.nextInt();
        System.out.println("Read the second side length:
            ");
        int b = keyboard.nextInt();
        System.out.println("Read the third side length:
            ");
        int c = keyboard.nextInt();
        System.out.println(TriangleType.getType(a, b,
            c));
    }
}

enum Triangle
{
ISOSCELES(0),
EQUILATERAL(1),
SCALENE(2),
ERROR(3);

private int n;
Triangle(int n)
{this.n = n;}
}
```

1. Do you have a test case that represents a valid scalene triangle? You need to make sure that the cases you enter can correspond to a triangle.
2. Do you have a test case that represents a valid equilateral triangle?
3. Do you have a test case that represents a valid isosceles triangle?
4. Do you have at least three test cases for #3 such that you have tried all three permutations (such as, 3,3,4; 3,4,3; 4,3,3)?
5. Do you have a test case in which one side has a zero value?
6. Do you have a test case in which one side has a negative value?
7. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is equal to the third?
8. Do you have at least three test cases for #7 such that you have tried all three permutations (for example, 1,2,3; 1,3,2; and 3,1,2)?
9. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is less than the third (such as 1,2,4 or 12, 15,30)?
10. Do you have at least three test cases for #9 such that you have tried all three permutations (for example, 1,2,4; 1,4,2; and 4,1,2)?
11. Do you have a test case in which all sides are zero (0, 0, 0)?
12. Do you have at least one test case specifying non-integer values (such as 2.5,3.5,5.5)?
13. Do you have at least one test case specifying the wrong number of values (two rather than three integers, for example)?
14. For each test case did you specify the expected output from the program in addition to the input values?

```java
// Specification: The program reads three integer values
//     from an input dialog. The three values represent
//     the lengths of the sides of a triangle. The
//     program displays a message that states whether the
//     triangle is scalene, isosceles, or equilateral.
//
import java.util.Scanner;
public class TriangleType {

    static Triangle getType(int a, int b, int c)
    {
        if(a<=0||b<=0||c<=0)
            throw new IllegalArgumentException("Length
                of sides cannot be equal to or less
                than zero");

        if(a==b&&b==c&&c==a)
            return Triangle.EQUILATERAL;
        else if((a==b)||(b==c)||(c==a))
            return Triangle.ISOSCELES;
        else if(a!=b&&b!=c&&c!=a)
            return Triangle.SCALENE;
        else
            return Triangle.ERROR;
    }

    static boolean isItTriangle(int a, int b, int c)
    {
        boolean triangle = !(a+b <= c || a+c <= b || c +
            b <= a || a<=0 || b<=0 || c<=0);
        if (triangle)
            System.out.println("It is a triangle: ");
        else
            System.out.println("The entered 3 sides
                cannot form a triangle.");
        return triangle;
    }

    public static void main(String[] args)
    {
        boolean isTriangle;
        System.out.println("IS This PROGRAM correct WITH
            REGARD TO THE ABOVE SPECIFICATION");
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Read the first side length:
            ");
        int a = keyboard.nextInt();
        System.out.println("Read the second side length:
            ");
        int b = keyboard.nextInt();
        System.out.println("Read the third side length:
            ");
        int c = keyboard.nextInt();
        isTriangle = TriangleType.isItTriangle(a, b, c);
        if (isTriangle) {
            System.out.println(TriangleType.getType(a,
                b, c));
        }
    }
}

enum Triangle
{
    ISOSCELES(0),
    EQUILATERAL(1),
    SCALENE(2),
    ERROR(3);

    private int n;
    Triangle(int n)
    {this.n = n;}
}
```

SFWR ENG 3S03
Software Testing

A. Marinache

Outline

Preliminaries

What Is Software
Testing?

Challenges of
Software Testing

Simple Example
Illustrating the
Challenge of
Testing

Roadmap

1. Testing Principles, Techniques, and Methods
2. Testing Metrics
3. Reviews and Evaluating Testing Methods
4. Static Analysis
5. Testing Strategies and Plans
6. Non-Functional Testing Techniques
7. Measurement in Software Engineering
8. Security Testing
9. Mobile and Web Testing

References I

SFWR ENG 3S03
Software Testing

A. Marinache

Outline

Preliminaries

What Is Software
Testing?

Challenges of
Software Testing

Simple Example
Illustrating the
Challenge of
Testing

Roadmap

📄 Robert Baber, *Software reflected*, North-Holland
Publishing Company, 1982.

📄 Keith Cowing, *Nasa reveals probable cause of mars
polar lander and deep space-2 mission failures*, March
28 2008.

📄 Norman E. Fenton and Shari Lawrence Pfleeger,
Software metrics: A rigorous and practical approach,
third ed., CRC Press, 2014.