ITP2200
Introduction to Software Testing

Bogdan Marculescu

Høyskolen Kristiania

Lecture 6
Integrating testing in the software development process

So, more programming this week?

Høyskolen Kristiania

What do we remember from last time?

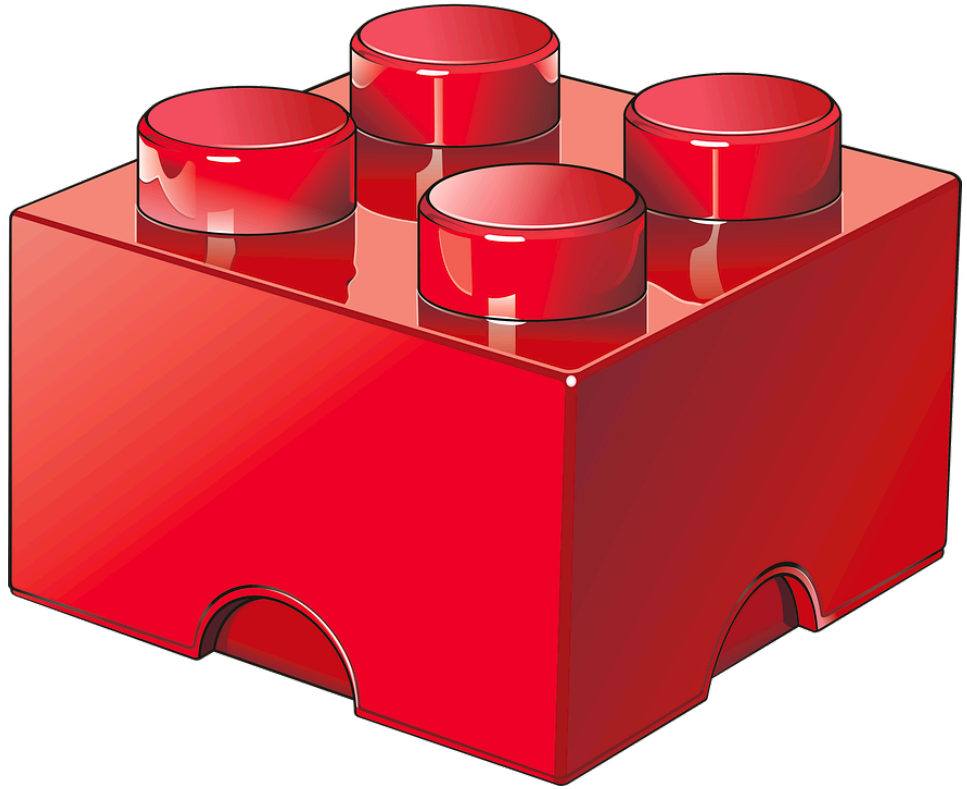
Høyskolen Kristiania

Beizer levels

Level 0 - There's no difference between testing and debugging.
Level 1 - The purpose of testing is to show that the software works.
Level 2 - The purpose of testing is to show that the software doesn't work.
Level 3 - The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.
Level 4 - Testing is a mental discipline that helps all IT professionals develop **higher quality software**.

Høyskolen Kristiania

Software Development Process



Unit test:

- Each component (class, method, etc.)
- Functionality, inputs, behaviours
- Input partitioning
- Coverage criteria

Høyskolen Kristiania

Software Development Process

Quick primer on how software development works:

1. Requirements analysis and specification
2. System and software design
3. Intermediate design
4. Detailed design
5. Implementation
6. Integration
7. System deployment
8. Operation and maintenance

Høyskolen Kristiania

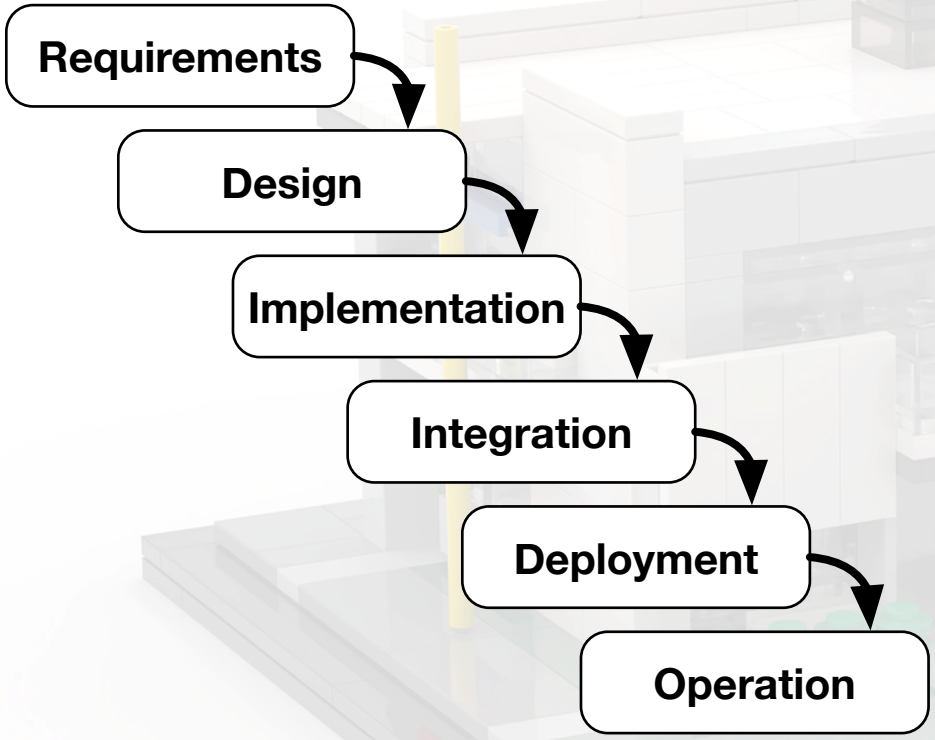
Software Development Process

Process(es):

1. “Waterfall”
2. Incremental
3. Spiral model
4. Agile

Høyskolen Kristiania

Software Development Process

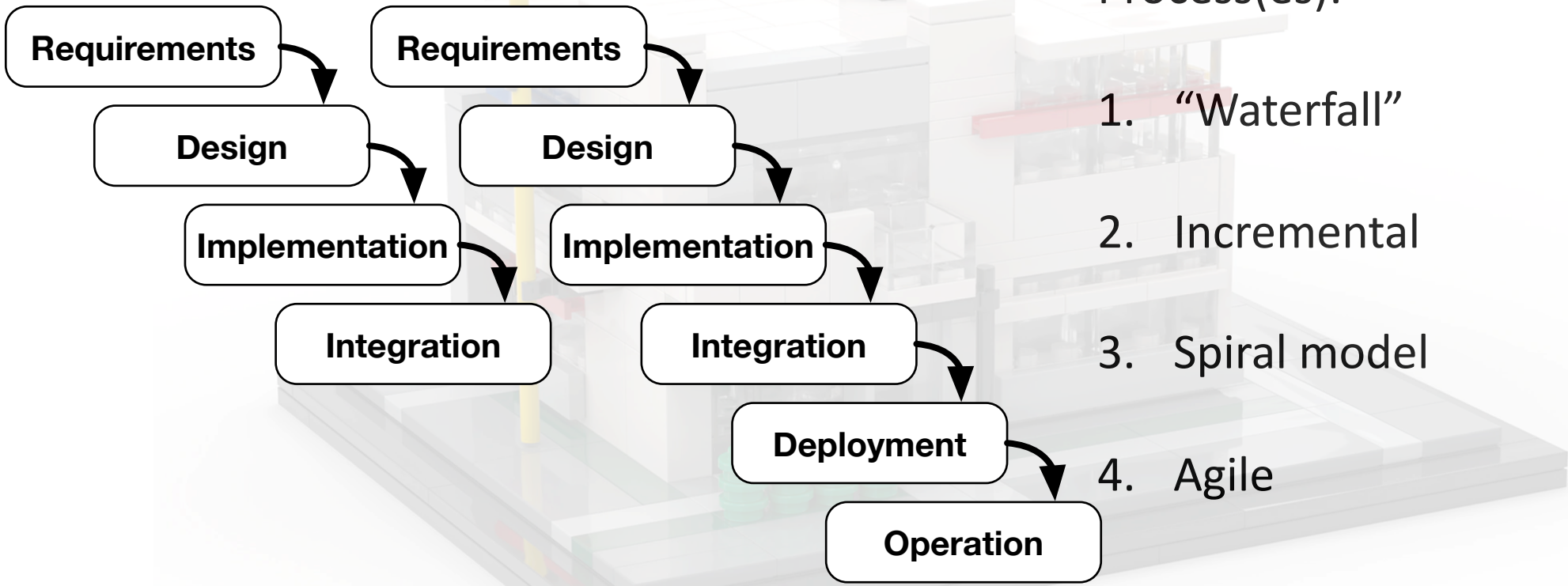


Process(es):

1. “Waterfall”
2. Incremental
3. Spiral model
4. Agile

Høyskolen Kristiania

Software Development Process

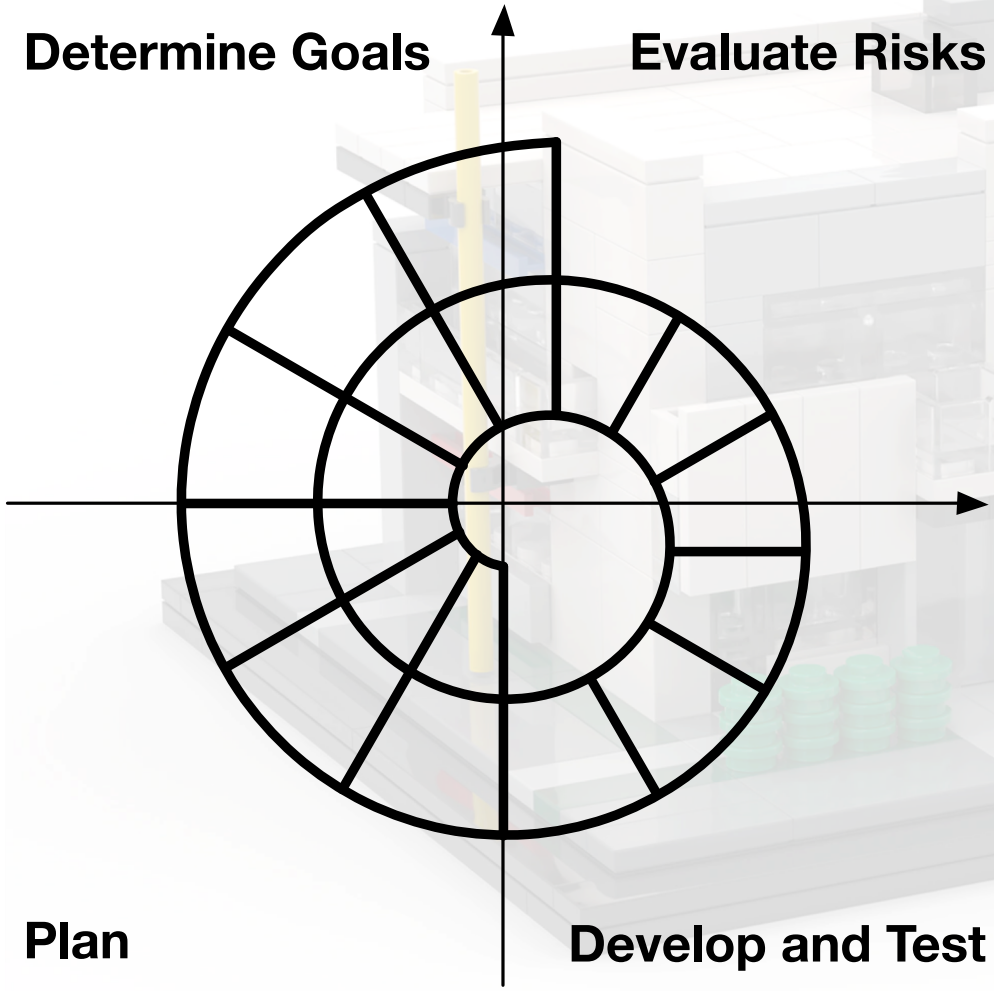


Process(es):

1. "Waterfall"
2. Incremental
3. Spiral model
4. Agile

Høyskolen Kristiania

Software Development Process



Process(es):

1. “Waterfall”
2. Incremental
3. Spiral model
4. Agile

Høyskolen Kristiania

Testing a Growing Product

We’ve done right so far

Existing code works

But more functionality needs added (there’s always more stuff to add)

Høyskolen Kristiania

Regression Testing

Re-test modified software

- Ensure new changes do not break existing functionality
- Automated
- Balance between more tests and manageable test suite size
- Evaluation Criteria: (wait… we know this, right?)

Høyskolen Kristiania

Regression Testing

```java
package ex06;

public class Time {
    int hours;
    int minutes;
    int seconds;

    public Time(int hours, int minutes, int seconds){
        this.hours = hours;
        this.minutes = minutes;
        this.seconds = seconds;
    }

    public static Time convertSeconds(int seconds){
        int sec = seconds % 60;
        int hour = seconds / 60;
        int mins = hour % 60;

        hour = hour / 60;

        return new Time(hour, mins, sec);
    }
```

How can we improve this?

- Ensure that any data stored in it is usable
- Ensure that any data is accessible when appropriate (and only when appropriate)
- Add additional functionality

Høyskolen Kristiania

Regression Testing

```java
package ex06;

public class Time {
    int hours;
    int minutes;
    int seconds;

    public Time(int hours, int minutes, int seconds){
        this.hours = hours;
        this.minutes = minutes;
        this.seconds = seconds;
    }

    public static Time convertSeconds(int seconds){
        int sec = seconds % 60;
        int hour = seconds / 60;
        int mins = hour % 60;

        hour = hour / 60;

        return new Time(hour, mins, sec);
    }
```

Private!

Example:
Make constructor private!

In groups (self-assigned)

Take 5 minutes and discuss:

Benefits?
Drawbacks?
What tests would you add?
What tests should be removed?

Pick a representative and share your findings!

Høyskolen Kristiania

Regression Testing

Changes can

- **Correct** known faults
- **Improve** existing functionality or performance
- **Adapt** existing structure and functionality
- **Prevent** future problems

Høyskolen Kristiania

Software Development Process

So,

Nothing old broke,

Is what is new any good?

Høyskolen Kristiania

Integration Testing

Assumptions:

Individual components are tested

Integration testing: ensuring that the combination of these components is also tested.

Høyskolen Kristiania

Integration Testing

Check that the various components:

1. Connect!
2. Interact appropriately
3. Require any changes to be performed

Høyskolen Kristiania

Integration Testing

```
1   package ex06;
2
3   public class Meeting {
4       Person owner;
5       Date date;
6       Time time;
7       int duration;
8
9   @   private Meeting(Person p, Date d, Time t){
10          this.owner = p;
11          this.date = d;
12          this.time = t;
13      }
14  }
```

The Person class is not ready yet!

- what inputs should it have?
- What methods should it have?
- Is currently designed behaviour appropriate?

Høyskolen Kristiania

Integration Testing

```
1   package ex06;
2
3   public class Meeting {
4       Person owner;
5       Date date;
6       Time time;
7       int duration;
8
9   @   private Meeting(Person p, Date d, Time t){
10          this.owner = p;
11          this.date = d;
12          this.time = t;
13      }
14  }
```

The Person class is not ready yet!

- what inputs should it have?
- What methods should it have?
- Is currently designed behaviour appropriate?

Høyskolen Kristiania

Integration Testing

Integration:

- Can be done on incomplete systems
- Some components can be replaced with Stubs (or Mock-ups in other contexts).
- Gives rise to a new problem: what is a good order to integrate components in.

Høyskolen Kristiania

Høyskolen Kristiania

Questions so far?

Software Development Process

Quick primer on how software development works:

1. Requirements analysis and specification
2. System and software design
3. Intermediate design
4. Detailed design
5. Implementation
6. Integration
7. System deployment
8. Operation and maintenance

Høyskolen Kristiania

Test Process

Quality

- At all the levels presented there!

- At each step – there are activities that can improve quality, and support or enable better testing

- It’s not just code being assessed!

Høyskolen Kristiania

Beizer levels

Level 0 - There's no difference between testing and debugging.
Level 1 - The purpose of testing is to show that the software works.
Level 2 - The purpose of testing is to show that the software doesn't work.
Level 3 - The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.
Level 4 - Testing is a mental discipline that helps all IT professionals develop **higher quality software**.

Høyskolen Kristiania

Høyskolen Kristiania

Questions so far?

Exercise for the Seminar

Rooms:

-A3-01

-A4-01

-A4-03

-A5-18

Høyskolen Kristiania

Exercise for the Seminar

RECAP!

- What tests do you have for the Time and Date classes?
- What coverage did you achieve?
- Explain your testing approach!

Høyskolen Kristiania

Exercise for the Seminar

RECAP!

- Ensure that the Time and Date classes only contain valid info (have tests to show that)

- Change those classes where appropriate!

Høyskolen Kristiania

Exercise for the Seminar

Write up tests that

- Fail before you update the Date and/or Time classes

- Pass after those classes are changed

Høyskolen Kristiania

Exercise for the Seminar

If possible, use your implementation from the previous seminar/OOP.

Focus on writing the tests!

Høyskolen Kristiania