

# Principles of Software Design

## Introduction

Robert Lukočka

lukotka@dcs.fmph.uniba.sk

[www.dcs.fmph.uniba.sk/~lukotka](http://www.dcs.fmph.uniba.sk/~lukotka)

M-255

What is the difference between real world project and your typical class assignment?

## Class assignment.

- The underlying reason on why to do the assignment is not your concern.
- The assignment is clearly stated and complete.
- There are no changes about your assignment during your work.
- You work on your own.
- You work on it for short amount of time, then you forget about.
- The resulting product is small, easy to comprehend by a single person.
- ...

# What we will do here

The aim is to give you a basic idea about software development.

- 1 We discuss various areas of software development.
- 2 How to actually write code under typical real life circumstances.
- 3 Basics of several useful technologies.

# Areas of knowledge [1]

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering models and methods
- Software quality
- Software engineering professional practice
- Software engineering economics

# Areas of knowledge [1]

- Computing foundations
- Mathematical foundations
- Engineering foundations

Supporting:

- Computer engineering
- Systems engineering
- Project management
- Quality management
- General management
- Computer science
- Mathematics

# Areas of knowledge

- The previous courses only touched some of these areas.
- We give a very basic overview of most of these areas.

# What to expect and what should you focus on

- No theorem or proofs.
- Various approaches how to do something.
- Strong and weak sides of each approach, when given approach is appropriate.
- Good practices within chosen approach:
  - amplify the strong sides of the approach
  - manage the weak side
- How given practice affects other areas of software development.



# How to program

You will encounter many programming languages in this course.

- Most examples will be mostly in Python.
- Assignments will be in Java (optionally, you may use Python)

A small introduction to Python as part of this course.

# How to program

You learnt how to program. But to make complex software you need reasonable design. You need to split your problem into small tasks.

- In such a way that each task is easy to handle.
- In such a way that they do not affect each other too much.
- In such a way that each task may be tested well.
- In such a way that that we can get rapid feedback.

Small is way smaller than I typically see in student projects.

# Technologies

Several technologies:

- Git
- Makefile
- Unit testing frameworks
- Docker
- ...

# PTS 1 and PTS 2

This semester we focus mostly on subjects that are of immediate use while working on your projects such as

- Version control systems
- Build automation
- Design
- Automated testing
- Good implementation practices
- ...

# PTS 1 and PTS 2

In PTS 2 the focus will be more broad.

- Handling requirements
- Architecture
- Management and Estimations
- Concurrency and Parallelism
- Databases
- ...

# Grading

- One assignment, 60 points. Deadline is late in the semester so we can go through the topics necessary to finish it.

Assignment 19.11., 26.11.

- Oral examination 60 points (two open questions from a published set of questions).

# Grading

- You have to use Git and Java/Python to complete your assignments (And If you use Python you have to follow style that enforces strong type checking similar to Java).
- After you are done you send entire repository to me and one of your colleagues (first deadline)
- During the next week your colleague reviews the code and you improve it accordingly.
- You send me the whole repository again (second deadline)
- Send the assignments to [lukotka.pts@gmail.com](mailto:lukotka.pts@gmail.com). Use this e-mail address just to send the assignments and maybe for some immediate discussion after I send you my feedback.

# Grading

- Grading:

*A*  $\langle 120, 110 \rangle$

*B*  $(110, 100)$

*C*  $(100, 90)$

*D*  $(90, 80)$

*E*  $(80, 72)$



# Resources

On this frame you usually find materials to learn from. You should read this stuff to prepare for the examination (especially if you missed the course or were sleeping during my presentation :)).

# References I

Sometimes I feel it is necessary to give you the source of some claim (I should do that more often than I do). This will appear here. You don't have to read the content of it.



[SWEBOK V3 -TOC](#)