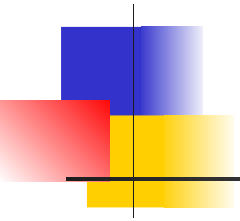


Introduction to the Course

Individual Software Process

Description in Course Catalog

กระบวนกำรพัฒนำซอฟต์แวร์สมัยใหม่ กำรพัฒนำแบบ วนรอบและแบบค่อยเป็นค่อยไป กำรวำงแผนและประมำณ

โครงกำรเดี่ยว กำรจัดกำรเวลำ กำรติดตำมเวลำ คุณภำพรหัส

โปรแกรม กำรปรับปรุงรหัสโปรแกรม กำรตรวจสอบรหัส

โปรแกรม กำรควบคุมรุ่นของรหัสโปรแกรม กำรทดสอบ

ซอฟต์แวร์เบื้องต้น กำรพัฒนำซอฟต์แวร์ภำยใต้กรอบงำน

Modern software development process, iterative and incremental development, individual project planning and estimation, time management, tracking time, code quality, code refactoring, code review, source code version control, introduction to software testing, software development under a modern framework.

Purpose of This Course

Developers work on projects in teams.

They apply a process to their projects.

Individual Software Process - skills, knowledge, and habits to be an effective developer alone or on a team.

Workgroup Software Process - how to work effectively on a (larger) team. Apply other process areas.

SKE technical courses - the knowledge you need

Topics

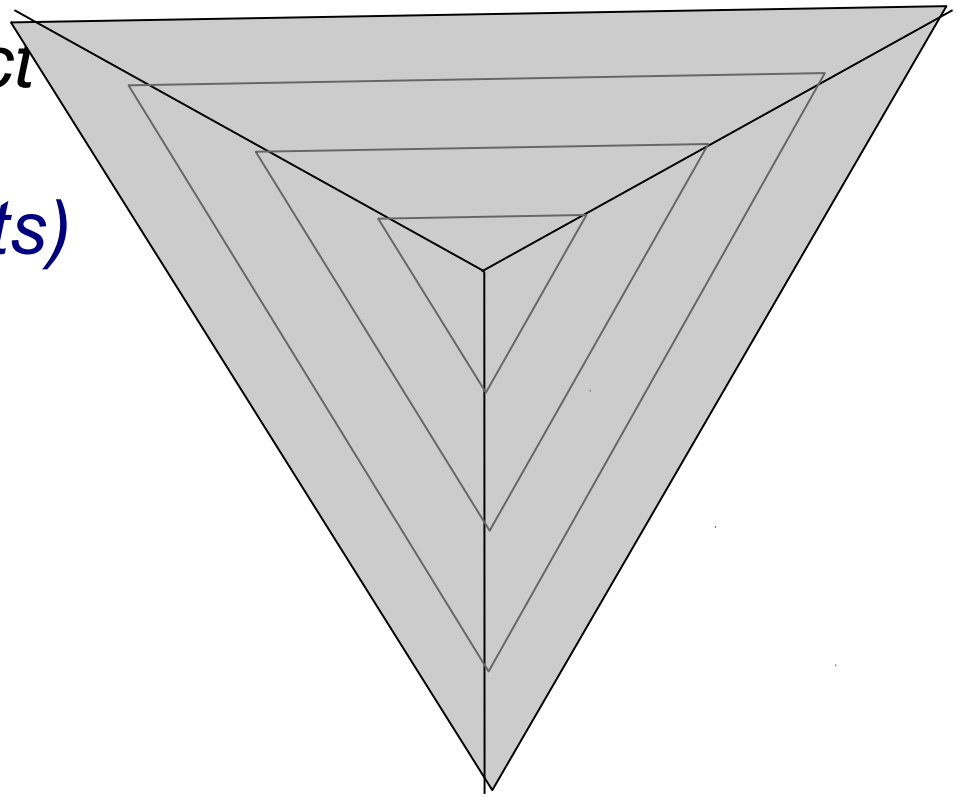
Conceptual Knowledge	Skills	Technology	Habits
Software processes Process areas and practices Iterative & Incremental dev, Agile concepts Waterfall	Estimation Planning Tracking Work Testing Reviews of design & code Build Management Refactoring Retrospective	Git Python unittest Persistence Task boards Issue tracking Automation, CI Build tools	Clean Code Quality Focus Self-learning Communication skill Time Mgmt.

Goal of the Course

Understand and consistently apply software development skills used by developers & teams

Dimensions of a Typical Software Project

Build the right product (Features, Requirements)

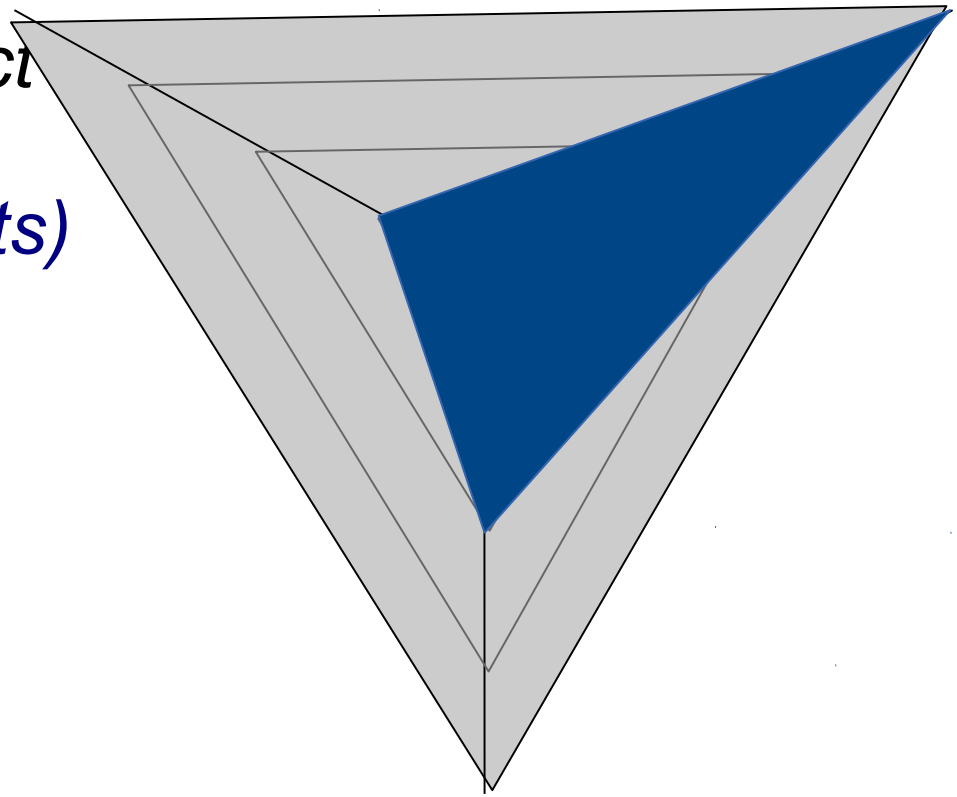


Build the product right (Quality)

Build it fast (Time, Cost)

Focus of this course

Build the right product (Features, Requirements)



Build the product right (Quality)

Build it fast (Time, Cost)

Prerequisites

1. Ability to write O-O style code in Java & Python at level of **Programming 2**.

2. **Git basics**: create & clone a repo, update files, push changes, view changes to files.

3. How to use **command line** to navigate the file system, manipulate files, enter git commands.

4. How to use Github and Github Classroom.

See: **`"Git" topics on https://skeoop.github.io/`**

Programming 2 Skill *Really* is Needed

If anyone has not passed Programming 1 and at least gone through Programming 2 (OOP),

then it is a waste of your time to enroll in ISP.

Pass Programming 1 and 2 **first**.

Then take ISP.

You will learn more.

Work and Grading

1. Weekly assignments - in lab and homework
2. Quizzes
3. Written Exams
4. Programming Exams
5. Small team project - a web application

Approximate Grading Scale

A	85% and above
B	75% - 85%
C	65% - 75%
D	55% - 65%
F	less than 55% overall
	or written exam average < 50%
	or lab exam average < 50%

To pass you must average >= 50% on written exams and lab (programming) exams.

Why? You must know concepts and how to use them.

The Rules are Strict *(sorry)*

1. No copying
2. Do assigned reading & work
3. Write good quality code
4. Use the coding standard
5. Install required software on your machine
6. No food in lab (drinks OK)
7. Participate in class

Copying

Copy anything --> Fail (F).

Including Homework.

No second chance.

Write Good Quality Code

1. Write code that is easy to read.
2. Write code that is testable.
3. Consistently use a naming standard.
4. Write meaningful comments.
   For Python, include docstring comments.

No Comments -> No Credit

Bad Coding Style -> No Credit

Online Course Resources

Google Classroom. **`https://classroom.google.com`**

- Enroll using course code:
- Assignments, announcements, feedback, discussion

Github Organization & Classroom: for programming work

- **`https://github.com/orgs/ISP2020`**

Course Material: **`https://cpske.github.io/ISP`**

- Organized by topic, not sequential order

`https://cpske.github.io/ISP/course-urls`