Skoltech

Skolkovo Institute of Science and Technology

Course Organisation

Foundations of Software Engineering

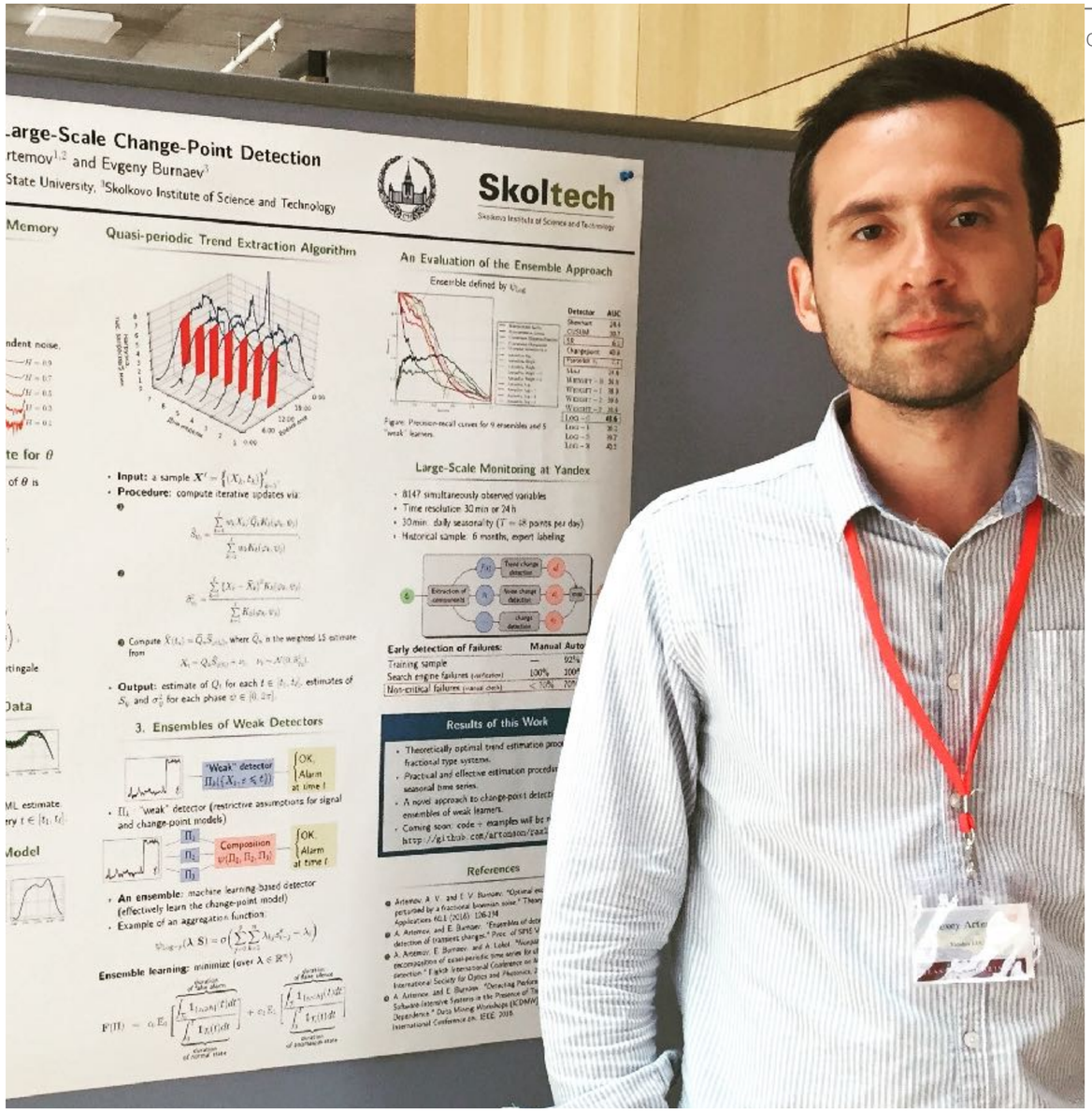
FSE v2021.1

Alexey Artemov, Fall 2021

Your instructor

Alexey Artemov, Ph.D.

- 2002—2006 LIT 1533, *Software Engineering*
- 2006—2012 Lomonosov MSU, *Physics*
- 2010—2012 Yandex Data School, *Data Science*
- 2011—2017 Yandex, Yandex Data Factory, Yandex Self-Driving, *Computer vision*
- 2012—2017 IITP RAS, Ph.D., *Statistics/Data Science/Software*
- 2017—now Skoltech, *Computer vision*
- **Core:** software, statistics and data science, computer vision
- **At Skoltech:** leading a team of 8 Ph.D., 12 MSc. students, >15 papers (5 Core A*)



Your TAs

Skoltech
Skolkovo Institute of Science and Technology

Emil Bogomolov

Slava Yarkin

Arseniy Bozhenko

Katya Voloshina

Outline

Skoltech
Skolkovo Institute of Science and Technology

§1. Organisation [15 min]

1.1. Why learn software engineering at a Data Science program?

1.2. Course outline

1.3. Course assessment

1.4. Disclaimers

§2. Course project [5 min]

2.1. Why course project?

2.2. Accomplishing the course project.

§1. Organisation

Why learn software engineering at a Data Science program?

AI/ML/DS: 2% math, 98% coding stuff

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.1. Why learn software engineering at a Data Science program?

- Most research in CDISE: **programming** (95% of all research in my team)
- Most experiments in CDISE: **computational experiments**
- Most projects in today's ML: **team efforts on software development**
- Most projects in today's computational sciences involve **HPC and heterogeneous computing, complex numerical libraries**
- Most cited papers in ML: **papers with great code**

It's All About The Software

Course outline

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.2. Course outline

Goals of this course:

- Provide an introduction into the **ideas** behind software engineering
  - Build automation, version control, scripting, continuous integration, …
- Learn the **tools** commonly used in software engineering
  - Unix, git, docker, vim, make, UML, …
- Gain the **skills** needed to continue progressing with software development
  - Writing unit tests, building docker images, …

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.2. Course outline

- Course structure for v2021.1: two “blocks”
  - Block 1: Unix fundamentals: Unix local & remote machines, scripting (3 classes)
  - Block 2: Software development in teams: version control, build automation, deployment/dockers, testing, debugging, deployment (6 classes)

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.2. Course outline

Unix stuff | Dev in teams | Project

	Term 1B Week 1	Term 1B Week 2	Term 1B Week 3	Term 1B Week 4	Term 1B Week 8
Tuesday	*(Term 1A)*	Version control	Dependencies, reproducibility, and docker	Deploying software	Project
Thursday	Unix fundamentals: local machine 1	Unix fundamentals: remote machine	Testing software	Project	*(Term 2)*
Friday Lecture	Unix fundamentals: local machine 2	Building software	Debugging software	Project	*(Term 2)*

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.2. Course outline

Structure of typical module (=offline class):

- Demo / lecture [50+25 min, 10 min break]
- Team-based practical/lab [25+50 min, 10 min break] → submit for assessment

Online:

- Read supplementary / watch recordings [30—60 min]
- Offline quiz [should take 15 min, 72 hours] → submit for assessment
- Work on project [should take 60–180 min] → final project

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.2. Course outline

- 🛠️ Notion course page
- ⭕ Github Repository
- ✈️ Telegram Channel
- ✅ Anonymous feedback form

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.3. Course assessment

- The goal of this course is to quickly raise your awareness of baseline techniques and improve knowledge, not evaluate you
- But Education asks us to still somehow do this…

The final grade =
- 40% × Computer labs
- 40% × Final project
- 20% × Test/quiz

Disclaimers

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.4. Disclaimers

Do take this course if (either applies):

- You have (almost) never used Unix or developed industrial software (e.g. for a living)
- You are expected to work a lot with Unix environments and want to optimize your time by learning how to do things (more) efficiently

Do not take this course if:

- You have a lot of prior experience with Unix and wanna refresh / go advanced

- Our poll (canvas link, google form link) may help you decide

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.4. Disclaimers

What you **will not** learn:

- Programming per se; Algorithms (except for a tiny subset); Project Management; Machine Learning / Big Data (no SWE for model deployment)

Ethics:

- There are multiple better ways to do things
- Unix and SWE is infinite (which is cool!) — your instructors are only aware of particular functionality
- Sharing hacks & ideas and contributing improvements is very much welcome!

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.4. Disclaimers

- SWE is like driving a car or working out: the more you practice, the better you get at it
- Lots of exercises in this course are going to be dumb as hell: this is intentional to make you repeat things many many times

§2. Course project

Why course project?

§2. Course project

Skoltech
Skolkovo Institute of Science and Technology

2.1 Why course project?

- Additional educational format used in this course
- Learning by doing
- Putting yourself in real-world[-like] circumstances
- **Trying to do something useful**

§2. Course project

Skoltech
Skolkovo Institute of Science and Technology

2.1 Why course project?

- **Goal: use SWE to improve real-world machine learning code repositories**
  - IN: crappy repo (not reproducible, outdated, hard to build, untested, undocumented)
  - Example: https://github.com/charlesq34/pointnet2
  - OUT: cool repo (environment/docker, automatically buildable, tests, documentation, wrapper scripts, etc.)
  - Example: https://github.com/NVIDIA/MinkowskiEngine
- Project performed in teams of size 2~4
- Each team refactors a single repo
- Results assessed by course instructors

Accomplishing the course project

§2. Course project

Skoltech
Skolkovo Institute of Science and Technology

2.2. Accomplishing the course project

- The goal: NOT to make the **right project**, but to make the **project right**
- Things your instructors and TAs are going to do for you:
  - Provide a set of repos to refactor
  - If needed, provide a server to run experiments
  - Provide continuous feedback

- Things your instructors and TAs are NOT going to do for you:
  - Write code, perform tests, write documentation, or negotiate with code maintainers

Quick questions?