Skoltech

Skolkovo Institute of Science and Technology

Course Organisation

Foundations of Software Engineering

FSE v2020.1

Alexey Artemov, Fall 2020

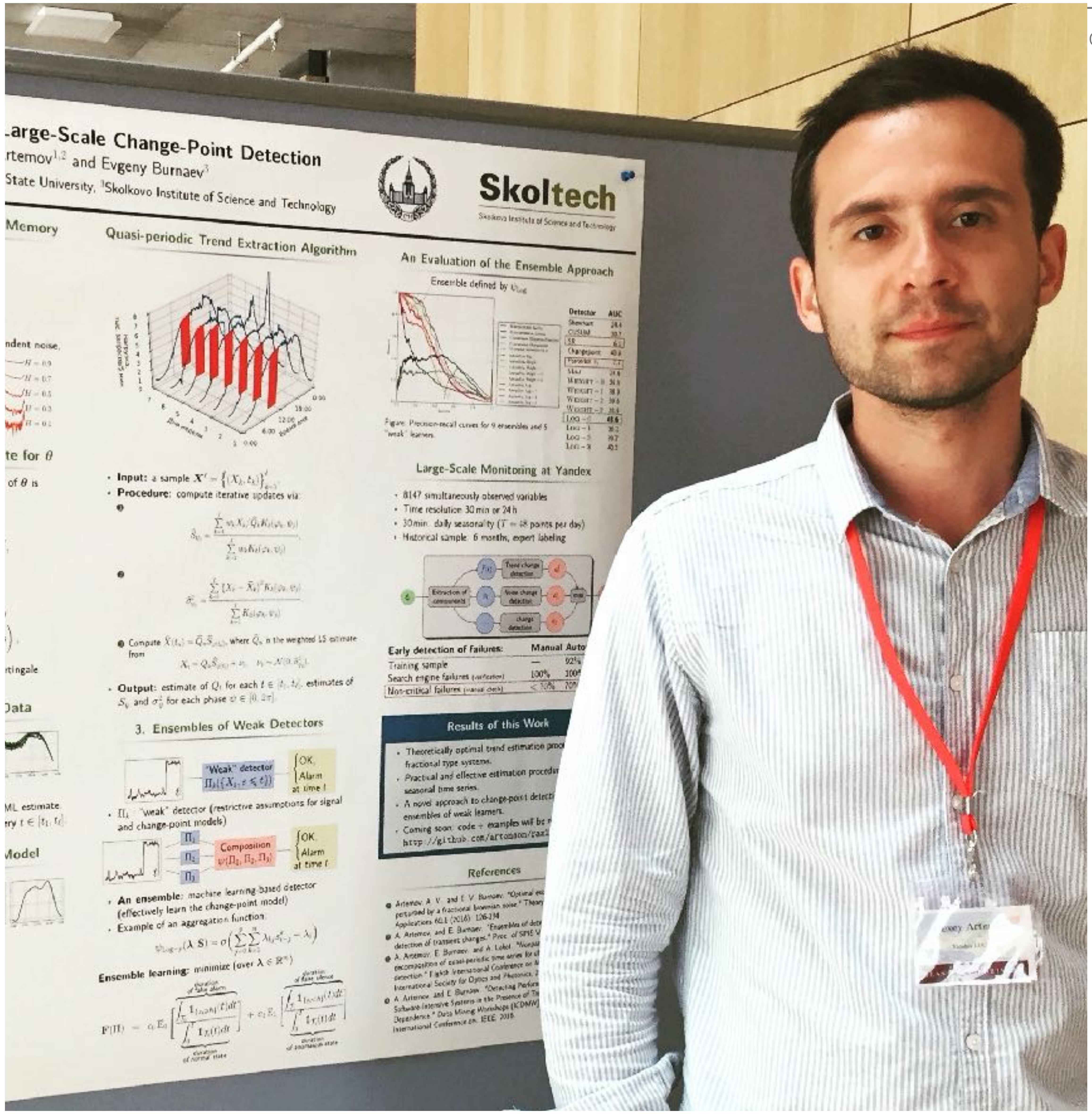
Skoltech

chnology

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

Emil Bogomolov

Vlad Ishimtsev

Arseniy Bozhenko

Outline

§1. Organisation

Why learn software engineering at a Data Science program?

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

§1. Organisation

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

Skoltech

Skolkovo Institute of Science and Technology

Course outline

§1. Organisation

Skolkovo Institute of Science and Technology

1.2. Course outline

Goals of this course:

- Provide an introduction into the **ideas** behind software engineering
    - processes and dev models, version control, continuous integration, …
- Learn the **tools** commonly used in software engineering
    - Unix, git, docker, vim, SRS, UML, …
- Gain the **skills** needed to continue progressing with software development
    - Writing unit tests, managing s/w requirements, drawing use-case diagrams, …

What you **will not** learn:

- Programming per se; Algorithms (except for a narrow subset); Management

Skoltech
Skolkovo Institute of Science and Technology

§1. Organisation

1.2. Course outline

- Course structure for v2020.1: 3 blocks
  - Module 1: Software development processes, requirements and design (4 lectures)
  - Module 2: Introduction into Unix development (3 lectures)
  - Module 3: Software development in teams (4 lectures)

§1. Organisation

Skoltech
Skolkovo Institute of Science and Technology

1.2. Course outline

🟥 Processes, reqs & design

🟦 Unix dev

🟨 Dev in teams

	Term 1A Week 1	Term 1A Week 2	Term 1A Week 3	Term 1A Week 4	Term 1B Week 5	Term 1B Week 6	Term 1B Week 7	Term 1B Week 8
Tuesday Lecture	SDLC	Reqs & Arch	OO Design	Unix Scripting	Testing	IDEs and codestyle	Deployment	
Tuesday Practical	*Project Intro*	*Formulate reqs & Select arch*	*Design*	*Build containers*	*Write tests given code*			
Friday Lecture	Process Models	Unix Local Machine	Unix Remote Machine	Version control				
Friday Practical	*Model selection*	*Try out Unix*	*Try out Unix*					

§1. Organisation

1.2. Course outline

Structure of typical module:

- Pre-recorded lecture [45 – 60 min]
- Live practical [45 – 60 min]
- Live/offline Lab exercise [45 – 60 min] → submit for assessment
- [Wait 1-2 days] Read supplementary [30 – 60 min]
- [Wait 3 days] Offline quiz [should take 15 min, 48 hours] → submit for assessment
- [Wait 6 days] Project assignment [should take 60–180 min] → submit for assessment
- Work on project [should take 60–180 min] → final project + peer review

Skoltech
Skolkovo Institute of Science and Technology

§1. Organisation

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 = 30% × Computer labs 40% × Final project 15% × Team feedback 15% × Test/quiz

Skoltech
Skolkovo Institute of Science and Technology

§2. Course project

Why course project?

Skoltech

Skolkovo Institute of Science and Technology

§2. Course project

2.1 Why course project?

- The core 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: build a web-based search engine for scientific papers in the area of computer science**
  - IN: lots of CS papers
  - OUT: an intelligent interactive search & analysis tool
- Project performed in teams of 3~5 people with distinct roles (e.g., requirements engineer, architect, developer, tester, …)
- Each team implements a particular component (e.g., crawler, UI, SERP, wizard, ranking, …) or feature (e.g., search using an example paper)

Accomplishing the course project

§2. Course project

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:
  - Serve as customers and users: Set up project requirements and provide feedback on your understanding
  - Serve as PMs: Help set up your development processes and resolve issues
  - Serve as admins: Provide a server and help set up your development environment
- Things your instructors and TAs are NOT going to do for you:
  - Write code, perform tests, write a requirements document, or negotiate with customers

Skoltech
Skolkovo Institute of Science and Technology

§2. Course project

2.2. Accomplishing the course project

	Term 1A Week 1	Term 1A Week 2	Term 1A Week 3	Term 1A Week 4	Term 1B Week 5	Term 1B Week 6	Term 1B Week 7	Term 1B Week 8
Development actions	**Milestone 1:** Project Charter, Vision & Scope, Process Model	**Milestone 2:** SRS, System Architecture	**Milestone 3:** Low-level system design	**Milestone 4:** Construction of core functionality	**Milestone 5:** Construction/ testing of core functionality	**Milestone 6:** Construction/ testing of core functionality on production env.	**Milestone 7:** Full function. production deployment & test	**Milestone 8:** Production deployment & user testing
Testing scenario	no user testing	no user testing	no user testing	Developers test core functionality	Testers/users test core functionality	Testers/users test core functionality	Users actively test most of functionality	Final acceptance testing
Milestone name		Life Cycle Objective		Life Cycle Architecture		Initial Operational Capability		User Acceptance Testing