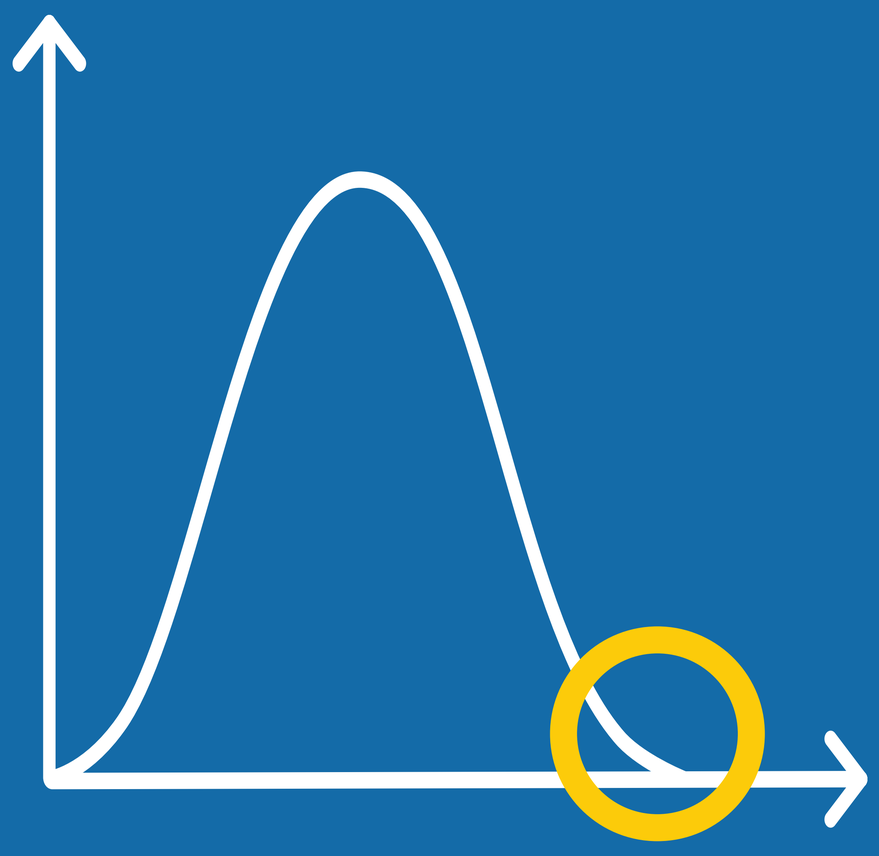

How to tap the minds of the brightest students for RSE?

SCAN ME

Michael Dorner
Florian Angermeir
Davide Fucci
Daniel Mendez

An experience report on integrating research software engineering into a software testing course at Blekinge Institute of Technology

THE COURSE

- Introducing hands-on testing and quality assurance techniques for software systems
- 40 students from the SE and AI B.Sc. programmes
- Each group (4–5 students) must implement a minimum test suite for a CI-pipeline and choose a project focus (e.g., performance testing or algorithmic verification).

THE CODE

- Simulation of information diffusion in code review at Microsoft, Spotify, and Trivago
- Implementation Dijkstra's algorithm for time-varying hypergraphs in Python

THE GOOD

- **Grew a comprehensive test suite** with unit and integration tests to catch regression bugs in the future and fuzzing to catch memory-corruption and safety bugs
- **Improved documentation** to provide context and lower the barrier to the project
- **Minimized code dependencies** and OS/hardware requirements to make it available for all
- Brought **students closer to SE research**
- **Overall positive student feedback**

THE BAD

- **Substantial efforts** (upfront and continous) for improving documentation context and minimizing dependencies and requirements
- **No direct integration of student code** into the project
- Some students felt overwhelmed

THE UGLY

- **Uncertainty about intellectual properties** of code contributions from students to open source projects