Software Engineering for K-12 Students

How Much Can We Teach?

Design and Planning

- Two of the key elements of software engineering are design and planning, and we often see a “rush to code.” Can you guide students to plan more?

Agile Methodologies

- Agile methodologies encourage short iterations of analysis, design, code, test, and deploy. While this fits in well with the way many students work, are the projects assigned to K-12 students of a nature where agile methodologies are applicable?

Documentation

- Documentation is essential for a software product, as opposed to a computer program, yet many students don't like to write. Can you get them to do more writing? Can you teach them to think in the kind of depth a good design requires?

Advanced Methologies

- Is there a place for advanced methodologies such as pair programming and test-driven development in the K-12 curriculum?

Object-Oriented Concepts

- Creating an object model can be challenging even for graduate students, and will often be seen as unnecessary for smaller projects. Can K-12 students grasp object-oriented concepts and apply them?

Team Methodologies

- If you assign group projects, do you let the teams self-select or do you assign students to teams? What is your rationale for your choice?