

Researchers hope to transform software engineering training

August 16 2012

Ten percent of software projects are abandoned, one-third are not successful and more than half end up late or over budget. But technology isn't to blame. People are. Now, researchers led by a San Francisco State University professor hope to improve the effectiveness of software engineering teams by learning what causes them to fail.

Failures in communication, organization and teamwork are the primary cause of problems in software engineering, says Dragutin Petkovic, professor and chair of SF State's [Computer Science](#) Department. The high rate of failure, he says, indicates a glaring need to rethink how software engineers are trained and managed, to ensure they develop the teamwork skills necessary to thrive in the working world.

To that end, Petkovic is leading a study that will learn what affects whether software engineering teams succeed and use that information to develop new software models that can objectively predict whether a team is likely to fail. That software will then be shared with other educators and industry leaders so they can monitor the effectiveness of their training and improve and build on the researchers' work.

Knowing which teams might fail could allow [software companies](#) to intervene and improve a project's chances of success. In order to learn which factors affect a team's outcome, Dragutin and his colleagues -- Kazunori Okada, associate professor of computer science at SF State, Shihong Huang, associate professor of computer science and engineering at Florida Atlantic University, and Rainer Todtenhoefer, professor of

[applied science](#) at Fulda University in Germany -- are looking to their students.

Assessments of how well software [engineering students](#) work in teams are typically done subjectively, for example by asking students to complete a questionnaire following the conclusion of the class, Petkovic said. His project will be the first to use objective and quantitative measures to assess and predict success in teamwork learning.

"Imagine if you could measure, during the class, how often teams use software tools or how often they meet, for example, and a computer would objectively tell you which teams are likely to fail," Petkovic said. "This is a dramatic and, I would say, radical solution to the problem. You could then use this 'early prediction' to adjust your teaching and help potentially failing teams. Or in a work environment, alert the management to address the issue before it is too late."

Dragutin, Huang and Todtenhoefer simultaneously teach a course in which students learn teamwork skills while developing a software project. For the study, researchers will collect data, such as how often team members e-mail each other or meet, how much time they spend coding and how long it takes for problems to get resolved. The aim is to develop software models that can automatically determine team success as well as rank the factors that influence the learning process, leading to better teaching methodology.

While some teams in the class are made up of students from a single university, others are comprised of students from all three institutions, mirroring a [software engineering](#) industry in which those working together on a project are more likely to be separated by several time zones than a few cubicles.

"Students have to learn how to work with a peer they have never met, in

a global collaboration setting," Huang said.

Once researchers have built the database, they can share it with others who can input their own data to evaluate their teams or broaden the study into other fields. Petkovic said he has already heard from mechanical engineers who would like a similar project to examine that field.

"We want to give advice to teachers and industry leaders on how to manage their teams," Petkovic said. "Research overwhelmingly shows that it is 'soft skills,' how people work together, that are the most critical to success."

Provided by San Francisco State University

Citation: Researchers hope to transform software engineering training (2012, August 16)
retrieved 21 June 2024 from <https://phys.org/news/2012-08-software.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.