Teaching tomorrow's computer programming methods

"Programmers that work for companies use what is called test-driven development," Goodwin said to forskning.no.

This method is based on dividing the programming into smaller tests. Each test is solved separately.

The programmer starts by creating a quick and dirty solution. It can be both inelegant and slow. But then the programme is gradually improved through trial and error.

This method yields results much more quickly than the traditional way, which involves deep analysis of the problem before a single line of programming code is written.

Absent in current teaching

Goodwin compares the two methods to building houses. Formerly, a carpenter built the entire house. Today the work is done by a team, and each person does one small part of the job.

"Both approaches are valid. The problem is that only one approach is used right now in teaching," Goodwin says.

Training tool

Goodwin’s research group has used test-driven development to give students with what they need most: lots of training.

Starting in the autumn of 2014, students will have to tackle programs that test student assignments. They will get instant feedback and will not have to wait for grades from teachers.
The educational programme will provide a competitive framework in which students can compare results, both based on speed and performance.

"We want to leverage the students' competitive instincts and their enjoyment of games," Goodwin says.

**Detects errors**

The programs Goodwin and his group have created are more than the usual debuggers found in most computer programs. Everyone has seen the red exclamation mark you get when you put a comma in the wrong place in Excel.

These errors are like spelling in natural language. But it does not help to have perfect spelling if the statement is meaningless.

"Programs can be free of coding errors, but may not solve the problem. These higher-level errors can be detected with our training programs," Goodwin says.

**Preparing for the business world**

The training programs must be tailored to the individual task. This requires substantial resources.

"We are applying for funding from the Ministry of Education and Research. In addition, we are looking for cooperation with other institutions," Goodwin says.

"The idea is that when students are fully trained, they will be familiar with the methods used in the business world," he says.

He is not afraid of competition from programmers in India or Eastern Europe.

"Companies need managers who know this craft, both in defining the tasks and in verifying the results," Goodwin says.

**Classical education**

"The business world is first and foremost in need of good programmers," says Dag Langmyhr, a lecturer in computer science at the University of Oslo.

"Although our students have a more classical education, after a short time in the business world, they will have learned test-driven development."

**Other types of programs**

The University of Oslo does not currently use educational programmes of the type Goodwin has developed, Langmyhr explains. Students receive instant feedback in other ways.

"Students receive test data for their assignments. These test data give known results. That way, students can compare their own results to the gold standard and immediately see if the program works," he says. It is also common for larger problems to be split into smaller parts that are assigned separately.

**Resource-intensive**
Langmyhr believes that the learning tool developed at the University of Agder is a good way of teaching, but he observes that it must be time-consuming for the instructors.

"In addition to creating the problems for each of the tasks, the instructors must split the problems into a number of sub-problems that can be solved on their own," he says.

"That means the programs simulate the relationship with the customer in test-driven development. This approach requires close monitoring of the students, but it is a great way to learn," Langmyhr says.

However, Langmyhr stresses that it is important that students receive feedback from the teacher, not just from a computer program.

"Feedback from the teacher can give the student a deeper understanding of what can be improved, and helps them understand to the less well defined qualities of programming," he says.

-----------

Read the Norwegian version of this article at forskning.no [9]