“We’re in the danger zone.”

That’s how senior Rodwitt Lai describes the Senior Design project he and fellow team members Eric Ngeo and Joel Frankford are working on in the Department of Computer Science. The course, led by Lecturer Peter Froehlich, matches real-world clients with teams of seniors who spend a semester working on technical problems no one else has been able to solve.

Lai’s team is improving the image analysis software used by the National Institute on Aging (NIA). “They have all these images of cells,” Frankford explains, “and they use a computer program to classify them as healthy or unhealthy cells. The user needs to make sure the software is functioning correctly and, if necessary, identify where in the process the computer makes a mistake. We’re building the software to do that.”

The danger? “The work we are doing is under very strict guidelines because we are changing code that is in active use,” Lai explains. “We could introduce problems that could crash the entire system.”

The other team of three has the opposite scenario. Their project, proposed by an independent software engineer in California, wants an “add-on” that will give users of the Web browser Firefox document-sharing capabilities.

“We’re making our own side program and plugging it into the Firefox browser,” explains Firefox team member Ben Schuster. Unlike the NIA project, this one allows Schuster and teammates Ayse Sabuncu and Ryan McLelland to work with whatever programming language they choose.

Nevertheless, the Firefox team ultimately has to sweat it out for three weeks before figuring out how to get the add-on to actually appear in the Firefox browser. “We’re on a good road toward finishing now,” reports Sabuncu, late in the semester. For the NIA team, success is defined as their ability to make progress.

“The most valuable lessons are learned through mistakes,” says Froehlich, who does not require successful completion of the project to pass the course or even earn the highest grade. Instead, he teaches students to focus on making tangible steps toward completion. It’s a win-win, he says. Clients benefit from seeing their projects advance. And the students, who don’t receive any payment for their work, reap the rewards of learning through mistakes as well as successes.

“The advantage of computer science is that you can make progress in little pieces,” Froehlich says. “You can’t build a useful bridge in little pieces, but you can build useful software that way.”

—AR