Propositions

to accompany the dissertation

Studies into Interactive Didactic Approaches for Learning Software Design Using UML

Dave R. Stikkolorum

1. Software design heavily depends on abstraction skills. An instrument that measures software design skills should correlate with the performance on abstraction tasks. – This Dissertation

2. The integration of gamification in software design education introduces extra possibilities for self-assessment which supports students in learning software design. – This Dissertation

3. Growing student numbers provide a challenge to lecturers and curriculum designers. Automating the evaluation of design assignments supplement the in-class feedback from lecturers. We need to implement guiding, educational agents that enrich self-assessment tools for software design education. – This Dissertation

4. Students predominantly use a depth first or breadth first strategy when creating software designs. Recognising strategies helps lecturers to provide better feedback during a task instead of post-task. – This Dissertation

5. By observing the peer reflections of students during class assignments in software design courses, lecturers can better understand the struggles students have during these tasks. At the same time, by discussing their software designs students better understand their own decisions. – This Dissertation

6. Many textbooks and lectures on software design have two major problems. Firstly, they use analysis problems for exercising software design. This mixes domain with solution concepts. Secondly, they use database-oriented problems for exercising class design. This mixes data modelling with object structure modelling.

7. Many students struggle with determining whether a concept should appear as a class or as an attribute in a UML class diagram. We need to expand the list of points of concern with future research to better understand student struggles. This leads to improved educational interventions.

8. It is difficult to grasp how students reason and make decisions. Understanding students better would improve our didactic approaches. There is a need for research in cognitive theories on learning abstract tasks such as software design. Therefore researchers in computer science should collaborate more with researchers in the field of psychology.

9. For a wider acceptance of (UML) modelling in industry, software design should be integrated into Agile software development processes, starting with university projects. The integration leads to a better understanding of the relationships between the stages of development and the appropriate modelling techniques. This approach engages students into modelling and motivates them to propagate this during their professional life.

10. Computing science should be a mandatory part of one of the graduation profiles of Dutch secondary schools.

11. While a critical attitude towards science is part of the scientific discipline itself, the criticism of science by non-experts disproportionately damages the trust of society in the value of science.