

# EDUCATIONAL PERSPECTIVES IN SOFTWARE FACTORY

**LEARNING** is about personal change. Formal higher education in computer science gives an opportunity to invest time and effort into learning – in other words, to make the personal change possible. In order to accumulate enough expertise to really be an expert in any given, a multitude of educational methods are used. There is a place for lectures, lab exercises and open-ended assignments; bits and pieces of information. But there is also a need for authentic learning situations that glue everything learned into a meaningful whole. Many times in formal education, these authentic learning possibilities are too rare. Such learning possibilities within the relatively safe formal context are indeed craved by the learners themselves. It is easy to see how inherently motivating and rewarding genuine learning possibilities are, when everything just “clicks” – real people, real demands, real facilities, and real need to showcase your talent, knowledge, creativity and perseverance.

At the Department of Computer Science, University of Helsinki, we now have a place for such a deep learning experience: Software Factory and a Software Factory project. It is a hands-on, outcome-driven personal growth experience for the students. It is not just a physical place to work but also a challenge and an opportunity. It allows the learners to put every piece of their accumulated expertise into practice; a challenge to exceed oneself. Moreover, it is an attitude: “Show me what you really can do!”

Software Factory has started in January 2010. It has been an instant success, even though the activity is still in the pioneering stage. What are the reasons behind the success?

## **Software factory project is intense**

It is clear that any project in the Software Factory requires devotion of time and effort. Everyone needs to declare their commitment to the project by signing into the four- or five-days-a-week work cycle for seven weeks. The Factory is an office space, and it is equipped with real tools and facilities, precisely what you could expect to find in an up-to-date software business. Such an authentic setting does not undermine the academic side of the project work but is beneficial to the learners’ self-confidence, since it provides them with real experiences and exposure to the same facilities as the businesses in the field.

When the activity in itself is the motivation, there is no need for external, artificial assessment of learning. Projects are inherently and intrinsically motivating as every software project in the factory has a potential path to commercialization, spin-offs and startups.

## **Software factory is about interpersonal interaction**

Interaction between people and other social skills are required for every expert in a highly complex field such as software business. Every project is a group effort. On the lines of best examples of cooperative learning (e.g. Johnson & Johnson 1989), software factory emphasizes positive inter-

dependence between the team members while requiring individual accountability. In addition, there is a cultural context as the factory has been multi-cultural from the beginning, and aims to be global in the very near future. Team members work side-by-side in close cooperation, but relying also on dispersed resources.

## **Software factory empowers the learner**

In formal education, often a rigid structure is imposed to the learning process, leaving little control to the learners themselves. Structure equates to teacher control, dialogue to negotiated control, and autonomy to learner control (Dron 2007). There is a place for structure, dialogue and autonomy, so none of them is undesirable in learning. But there is an added benefit for the learners if the locus of control can be shifted dynamically. As Dron puts it, “the ideal would be to allow the learner to choose whether and when to delegate control at any point in a learning transaction”. This is precisely the educational structure of the software factory – true learner-driven learning with always-available non-invasive support for technical, managerial and leadership challenges.

## **Software Factory’s global extension takes the learning to a new level**

While software development today is by definition globally distributed, also Software Factory has plans to set up satellite sites around the world. This is likely to take the learning even further as the global space will increase the level of authenticity significantly. Challenges discovered in industry include communication and coordination issues as well as cultural themes, which are likely to shape the outcome. Later this year, our sister-site at Madrid will be established and the Born Global thinking will begin to take shape. We believe that this is also increasing students’ motivation toward the maximized learning experience. We also believe that having students passing the Software Factory experience are better equipped in facing the volatility and inherent dynamics of the industrial setting once graduated.

## **References**

- Dron, J. (2007). Designing the Undesignable: Social Software and Control. *Educational Technology & Society*, 10 (3), 60-71.
- Johnson, D.W. and Johnson, R. 1989. *Cooperation and competition. Theory and research*. Edina: Interaction Book Company.



**Jaakko Kurhila, PhD**, is the Head of Studies at the Department of Computer Science, University of Helsinki. His earlier career work included research on various topics of educational technology, especially collaborative and social software. Nowadays, he eagerly promotes greatness in education.

