

## Tick-the-Code

### Training introduced



It is fairly rare that you can teach veterans a completely new and useful skill. Software quality control is often considered a boring and laborious undertaking. Consequently **Tick-the-Code** produces a double surprise. Not only is it previously unknown and yet a useful skill, but its clear, systematic nature makes software quality control positively fascinating.

The training courses teach the participants to find programming errors and unnecessary complexity in source code. Unnecessary complexity causes errors or delays in the future. Removing unclarities *now* reduces misunderstandings *later* and helps keep the software extendable and maintainable.

Small groups - of up to ten people - learn in practice to mark rule violations. 24 dealt-out rules are based on good programming principles, such as clarity, modularity, explicitness, conciseness and integrity. In the training the participants practice with several of the rules and as a result find violations in the source code they've brought. Fixing the source code after the training course will clarify the code considerably. You could say the training course pays for itself right then and there.

In addition to the rules, all participants receive a summary of the training listing the key topics. A durable laminated card includes the rules, the attitude and main purpose of the method. The participants also receive the Commitment to Quality Certificates, meant to inspire them to use the **Tick-the-Code** skill often and regularly also after the training course. In a frame on the wall it is a great reminder of the newly-acquired skill.

The concept of busyness and how the lack of time - especially constant lack of time - affects the quality of work, is discussed on the course. The discussion leads to the core problem of quality improvements: How to establish a lasting change? What kind of quality improvements survive in organizations? Hunting for bears works as a metaphor for a typical process emphasizing error correction. Preventing errors may be less instinctive, but it is also significantly more effective, cheaper and not at all impossible. Can you be happy as a software developer? Doing software quality control?

## Tick-the-Code

- for practical software quality -

### Miska Hiltunen, Head Trainer

- In software since 1993
- M.Sc. (IT) 1996
- 1993-1998 Genera Prosign Oy: software tasks of all sorts
- 1998-2005 Nokia Oy and GmbH: from team leader to quality specialist
- 2006- Qualiteers: freelancer
- Developed **Tick-the-Code**
- Programming experience: C, C++ and Java
- Eloquent, clear and entertaining presenter
- Lives in Germany (1999-)
- Training languages: Finnish, English, German

### Training Courses & Extensions

- **DayTick:** 7h, 5-10 people - immerse yourself in code for a day
- **Licensing** makes it possible to utilize your own trainers.

There are two management extension modules:

- **PoliTick:** 1-2h, no participant limit - inform management about **Tick-the-Code**
- **PlanTick:** 1-2h, no participant limit - ensure successful deployment of **Tick-the-Code**