

## DayTick - Immerse Yourself in Code for a Day

The participants learn to use the **Tick-the-Code** rule cards in their work. The theory parts emphasize the importance of a Quality Attitude. Several rules are practiced and the results measured. Everybody can directly see the improvement in their peer review skills.

If investing a whole day for a training seems too much, think again. With a focused training day, you can show you're willing to invest more than the absolute minimum for quality. You can send the right message. You can show that you as a manager have the Quality Attitude already.

### Target Audience

The training course is aimed at anybody producing software source code. Beginners, seasoned programmers and gurus alike can benefit from the courses. Even testers can benefit.

### Objective

The participants learn how to check software source code in an efficient and effective way, which enables them to use it regularly, i.e. to make it into a habit with ever growing gains.

The participants leave the course more quality-conscious.

### Prerequisites

The participants should be familiar with a programming language (the rules are guaranteed to work with C/C++/Java/C# code). Many other languages have been tried and as long as it is a language using text statements most of the rules will work. Languages like Python, Perl, and Visual Basic have been used, for example.

### Duration

The course lasts for one whole day (7h + lunch break).

### Class Size

5-10 participants is optimal.

### Trainer

The course trainer is Miska Hiltunen.

### Language

The course is held in English, Finnish or German.

### Location

The training course is held in-house.

### Material

All participants get to keep

- 4 rule cards containing 24 rules,
- a summary card and
- a "Commitment to Quality" certificate.

### Teaching Method

- The participants learn to tick by ticking for real.
- They get to practice on 15 rules.
- Quality theory is intertwined with the practical exercises throughout the course.

### Topics

- **The Vicious Circle of Busyness:** How time or the lack of it affects quality?
- **Dilemma of Quality:** How to make sustainable improvements?
- **Paradigm Switch:** Still Wanna Shoot a Bear?
- **Feelings in Software Development:** Why do I feel anxious or bored at work?
- **No Causality:** Software IS Different (Boris Beizer)
- **Quality Attitude:** How to Ensure Quality in Practice?
- **Measure the Improvement:** How much better do you check afterwards?
- **Ideal Development Week:** How often should you check?
- **Checking Method Comparison:** Just-changes, walkthroughs and spec checks

#### Optionally, time permitting

- The Principles behind the Rules
- Errors and Freedom of Choice
- Measuring Software: Standardize the Checking
- All That's Urgent Isn't Important
- We're All in the Same Boat