

# Software Engineering revisited



Traditional engineering disciplines are based on physics. Rules that have been unchanged for many years. Software "engineering", however, is based on whatever we think is good and, unfortunately, it is not an exception we disagree about what is good or bad. Many "real" engineering projects are simply copies of other existing projects. The difference between two similar bridges is minor, they are just copies of the same design. However, the level of effort to build a copy of an existing bridge in a new location cannot be overlooked.

If the level of effort to start using a finished piece of software was comparable to building a copy of a bridge, things would move more slowly and more stable. But we are constantly developing new products and enhancing existing products. The fact that we can innovate and distribute new software at an amazingly fast rate is a major detriment to our industry. Combined with rapid changing technologies as well as increasing pressure on schedule and budget, software projects remain increasingly unpredictable. This industry is still far away from being an engineering discipline, and the demand for qualified software professionals is increasingly high.

That software engineering is difficult to teach is widely understood and much discussed. Less well understood, and certainly less discussed, is the inverse of this - that software engineering is a particularly difficult subject to learn. More than any other subject in computing, software engineering reveals wide differences in ability leaving a long 'tail' of less able students struggling with basic material. This is particularly serious because, for most graduates in computer science, software engineering will form the core of their professional practice.

It is concluded that there is a high need in industry to (re-)educate software professionals in basic and advanced software engineering principles. That is why a coherent curriculum of instructor-led courses was developed addressing these topics. Intended audiences are:

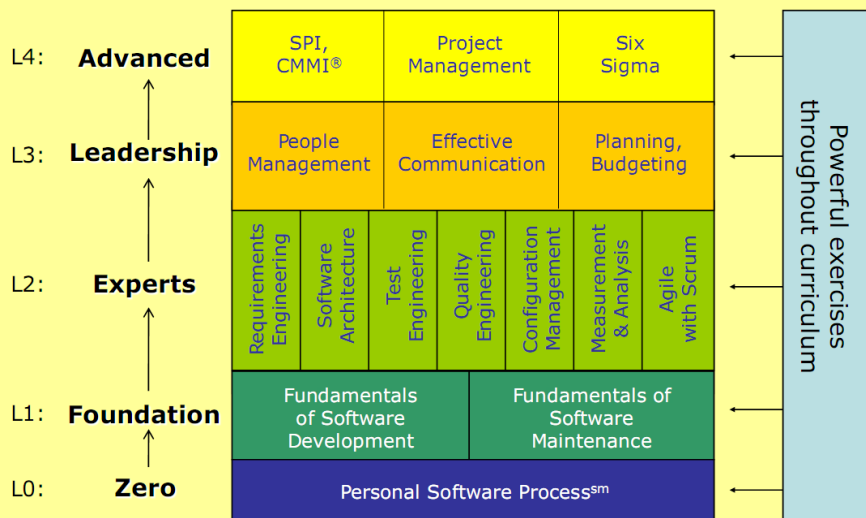
- Software manufacturer organizations wanting to improve their organizational performance and increase job satisfaction of their (potential) employees;
- Software professionals wanting to increase their market value.

# Curriculum



The curriculum exists of 16 instructor-led courses and is derived from internationally accepted standard references such as PMBOK, SWEBOK, publicly available work from the SEI, and books from internationally recognized authors. All exercises are based on practical experiences from our (senior) instructors. They are very challenging for the course participants, who will work in small teams to find appropriate solutions. Lectures and exercises are optimally balanced (50/50).

To provide optimum flexibility to learning, courses are available at five defined training levels. With this modular approach to learning participants can pick and choose those topics that meet both the individual's and your organisation's training needs.



## L0: Zero Level

A must for every professional software engineer involved in programming.

## L1: Foundation Level

A series of courses that covers the basics of software development and software maintenance. The main reference for the course materials is the Guide to the Software Engineering Body of Knowledge. This guide was developed by the IEEE Computer Society and established for the first time a baseline for the body of knowledge for the field of software engineering.

## L2: Expert Level

A series of technical courses for those wanting to become an expert in a specific area. Participants receive a thorough introduction to basic concepts and available standards, methods, and techniques.

## L3: Leadership Level

A series of non-technical courses that enable experts to understand the basics of managing teams and assure effective teamwork.

## L4: Advanced Level

These courses cover the basics of (software) process improvement, project management, and Six Sigma initiatives.

All courses are available in English, German and Dutch with native speaking instructors. Class sizes are between 10 and 20 participants. On request, courses can be customized for in-house training.