



Workshop: Learn from infamous software errors.



Why this workshop?

Sloppy code is error prone. Just the smallest typo in code can lead to serious errors and severe consequences. But it is not only about errors and bugs, imperfections in the code can have severe consequences as well. Imperfections in design and code increase the probability of bugs but also increase complexity.

These imperfections in design and code are called technical debt and the consequence of high levels of technical debt is a serious decrease in productivity and maintainability.

There are many causes of technical debt, like working under time pressure causing engineers to take the short cut instead of building the right sustainable solution.

Technical debt is one of the reasons that accuracy and discipline are so important in software development, even software professionals themselves in many cases do not realize the consequences of their short cuts, especially when these imperfections accumulate.

A continuous high level of software quality is the best assurance for a low number of bugs and a sustainable pace of development.

In this workshop you will discover the diversity of software quality which is so much more than testing and process compliance only.



The Method

The workshop “Learn from infamous software errors.” presents the participants a number of press-articles written about symptoms of the A2LL system. Participants will analyze the press articles and will, in small groups, reconstruct possible causes of the described symptoms.

Participants will learn what possible consequences of lack of quality are and thereby their consciousness about software quality will increase.

During the analysis and discussions the 1+3 SQM will be used as a reference in which Product Quality, Design Quality, Code Quality and Organizational Quality will be addressed.

The A2LL system was developed more than a decade ago commissioned by the German government and its purpose was to handle calculation and payment of unemployment premium.

Have fun and learn from mistakes made in the past!



Program

Block-1: Introduction - The A2LL project – 1+3 SQM

The organization and set-up of the development project of A2LL are retrieved and discussed from the information as provided by the press articles.

The 1+3 SQM is introduced and explained.

Block-2: What were the consequences of the lack of quality of A2LL?

Press articles are read and discussed in small workgroups (3-4 people) to get an overview of the flaws and consequences of poor quality of A2LL.

Each group will present their findings.

Block-3: What were possible causes of the lack of quality of A2LL and what can we learn?

Using the 1+3 SQM as a reference model possible causes for poor quality of A2LL and learnings are discussed by the same small workgroups. Additionally the question ‘what can we learn from A2LL?’ is discussed and answered.

Each group will present their findings.



For whom is this workshop?

The level of software quality breaks or makes a product or service, not only from customer or user point of view, as well from development point of view.

Accumulation of many imperfections, small in itself, can have major consequences. In this workshop these consequences are made explicit by the use of an example of software development in which things went severely wrong.

For **software developers** and **test automation engineers** discipline, even under high time pressure, is important. They need to be conscious of the consequences of their daily made decisions in engineering software to be able to stick to their discipline.

For **project managers**, **product managers** and **business managers** to understand what the consequences of low quality and high levels of quality debt can be. Such they understand that the development team ‘needs’ time and effort to do their job as they should do.

Workshop: 6-12 participants

Duration: 4 hours.

Hand-out: the book “What is Software Quality?”

More information: ger@cloudtsoftwareconsulting.com

WHAT IS SOFTWARE QUALITY?

Understanding what really matters in software development.

