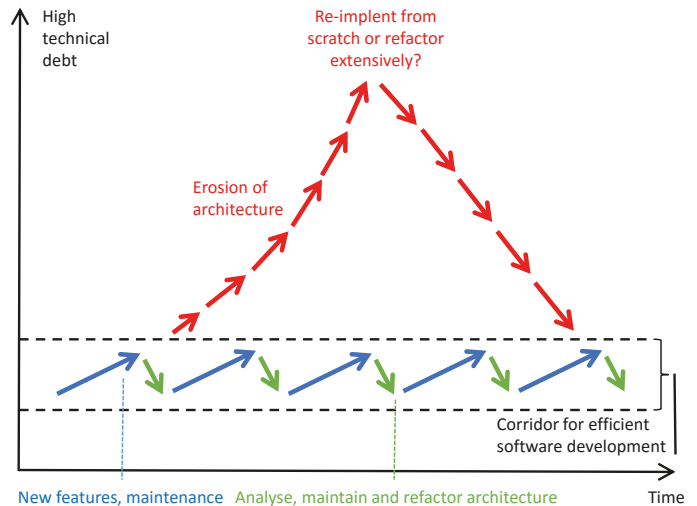# ARCHITECTURE ANALYSIS

These symptons are well known: The older a software system gets, the longer it takes to implement changes to it. This not only stresses developers, but also the customer's wallet. Architecture analysis adresses methods for uncovering problems and letting software developers go against it efficiently.

Architecture violations, clashing with the style, redundant or dead code, violations against metrics, hidden or cyclic dependencies and other problems can be avoided by regular analysis. This increases code quality.

If you buy or get delivered code, you should always check for quality and potential risks in software-architecture.

High technical debt

Re-implent from scratch or refactor extensively?

Erosion of architecture

Corridor for efficient software development

New features, maintenance   Analyse, maintain and refactor architecture   Time

„If you think good architecture is expensive, try bad architecture"
Brian Foote and Joseph Yoder

Your benefit

- Your software is easy to maintain even after any years.
- New functions can be implemented easily – almost like in a completely new system.
- Developers work on new features instead on hard defect analysis.
- Developers are satisfied and productive.
- You know if your architecture can handle future requirements.
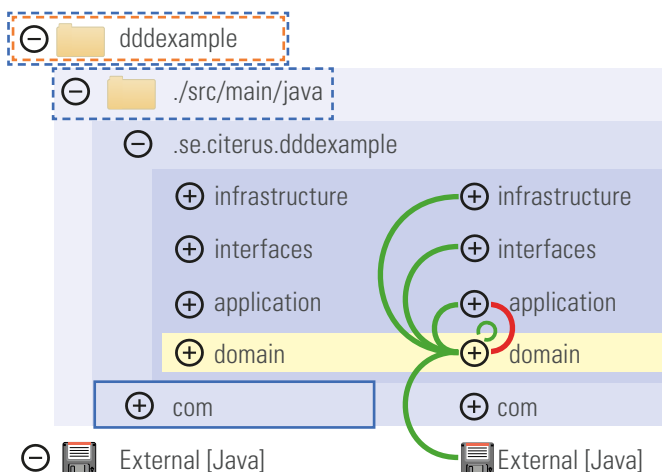
Services

- Analysis of technical debt using static code analysis and ATAM (Architecture Tradeoff Analysis Method).
- Elaboration and prioritisation of improvement actions.
- Evaluation and introduction of tools for static analysis of software architecture.
- Integration of static architecture analysis into the continuous integration process.
- Efficient documentation of the software architecture.