Maintainability is defined as the ease with which a software can be modified. The main-

- and conform to changing requirements.
- add functionality,
- improve quality attributes (e.g. performance),
- correct faults,
- If the software fails to finish before the requirements change it becomes obsolete

Case Study
Throughout a software development project a lot of time is spent on maintenance. Therefore code that is easy to maintain is desirable. A case study with an Austrian development team that changed its development strategy from the waterfall model to Scrum was conducted. The development of the code maintainability was analyzed when using the waterfall model and when using Scrum as development strategy. The analyzed software project started in July 2008 with the waterfall model as development strategy. In June 2009 the team switched to an agile development strategy, Scrum. The target of the analyzed software project is implementing a patient data management system (PDMS) as a part of a greater software project. The analysis includes six months of development using the waterfall model and twelve months using Scrum. The code of the development team that was implemented in the time between November 2008 and May 2010 was analyzed.

Waterfall Model
- Sequential development process
- Customers are never able to specify all requirements up front
- Requirements change during a software project
- No feedback from customers
- No means to respond to changes after the requirements specification
- Customers are almost never able to specify all requirements up front

Scrum
- Iterative development process
- The development team has to collaborate closely
- Takes the evolutionary nature of requirements, resources and competition into account
- Requirements are captured in user stories and prioritized in the product backlog
- When requirements change new user stories are added to the product backlog
- Customers are part of the development process through constant review meetings
- Customers control the development by prioritizing the product backlog items

Maintainability
Maintainability is defined as the ease with which a software can be modified. The maintainability indicates how complicated it is to modify the code to

- correct faults,
- improve quality attributes (e.g. performance),
- add functionality,
- and conform to changing requirements.

Measuring Maintainability
Changes in one part of the code should not unexpectedly or unintentionally affect other parts of the code. The robustness against the negative impact of the ripple effect is called logical stability. Elish and Rine found that the metrics coupling between object classes (CBO), response set for a class (RFC), and weighted methods per class (WMC) are good indicators of logical stability of classes. Highly cohesive classes are said to have high maintainability and reusability. Therefore the metric tight class cohesion (TCC) is also used for the analysis. The metric values are normalized by the number of classes.

Trend Analysis and Statistical Analysis
The blue line represents the development of the average metric values per class during the waterfall model. The red line represents the development during Scrum. The trendlines indicate that the average metric values per class do not increase as much when using Scrum as when using the waterfall model as development strategy. The statistical method used for validation is a non-parametric analysis of variance (Friedman’s ANOVA) with post-hoc tests (Wilcoxon signed-rank tests with Bonferroni correction).

The statistical analysis shows that the CBO metric values did significantly change while using different development strategies (X²(2) = 11.143, p < 0.05). The null hypothesis is rejected for the CBO metric.

The RFC metric values did also change significantly (X²(2) = 12.286, p < 0.05). The null hypothesis is rejected for the RFC metric.

The WMC metric values did also significantly change (X²(2) = 12.286, p < 0.05), but the results of the post hoc tests are ambiguous. The null hypothesis is not rejected for the WMC metric.

The TCC metric values did also significantly change (X²(2) = 8.857, p < 0.05), but the results of the post hoc tests are ambiguous. The null hypothesis is not rejected for the TCC metric.

Conclusion
- The agile operation method of Scrum has an effect on the development of the CBO, RFC, WMC, and TCC metric values.
- Scrum does not increase the CBO and RFC metric values as much as the waterfall model.
- Scrum does a better job in preserving maintainability of the source code than the waterfall model.
- No unambiguous conclusion can be drawn from the results regarding the WMC and TCC metric values.