Agile Practices Help Sheet

Refactoring

- Refactoring is the name given to the process of improving the design of existing code by making a series of micro-changes that preserve the external behaviour of the code being changed.
- Refactoring is also the third step in the Test Driven Development (TDD) mantra of “red, green, refactor”, which means a continued focus on the quality of code. Refactoring is built into the smallest cycle of work most Agile developers perform.
- Don’t be confused or intimidated by the name: the process of refactoring is the application of a bunch of very *simple* changes (each one a “refactoring”), many of which can be automated through the use of modern IDEs like Eclipse, Visual Studio/Resharperor or IntelliJ.

To give an idea of the simplicity of the work involved by many refactorings, here is a list of some of the most common ones used by developers

- Rename variable
- Rename method
- Extract method
- Extract class

Agile development teams use the continuous application of disciplined refactoring to keep the code quality high. This helps to reduce the amount of technical debt in the system, allowing the code base to adapt easily to changes in requirements.

REFERENCES

Written by Martin Fowler, this is the first book to coin the term “refactoring” and provides an extensive catalogue of refactorings, presented in a common format.