



The 7digital and Esendex Prize for Software Quality

We're looking for evidence of the principles and practices that we know help us make good quality software. We don't expect you to have done all of these but including any in your code, report or demonstration at the Open Day will certainly get you closer to our short list.

You can find out more about these topics by searching on Google, reading books and blogs, or contacting us via Henrik. Some good authors and bloggers we'd recommend are Kent Beck, Robert C. Martin (also known as Uncle Bob), Martin Fowler and Bill Wake.

Code Quality

- Good naming
- Formatting and conventions
- Methods and classes to describe behaviour over comments
- Small classes with single responsibilities
- Awareness of SOLID principles in general
- Short methods with few parameters
- Reduced amounts of nesting
- Awareness of duplication both in terms of code and behaviour duplication
- Pair programming

Code Architecture

- Loose coupling (inverted dependencies)
- Good agile design (low rigidity, low fragility, low viscosity, reusability, no needless complexity or repetition)
- Well organised projects and files
- Granularity

Testing

- Unit tests
- Test driven unit tests
- Integration tests
- Acceptance tests
- Smoke/Blackbox/End-to-end tests
- Characterisations tests of any existing code
- Continuous integration
- User testing

Team Organisation

- A clearly defined shared team goal or mission
- Team structure (roles and responsibilities are understood even if temporary or frequently changing)
- Use of source control
- Use of work in progress boards, Kanban or SCRUM
- Agile planning techniques (breaking into story tasks, planning poker)
- Prioritisation of some form

Knowledge Sharing

- Regular updates (regular "stand-ups", online or otherwise, collaboration tools, etc)
- Pair programming
- Regular deliverables and demonstrations - continuous delivery