Embedded Systems Software Engineering

Jarosław Kuchta

Agile Programming methods
Classic methods weaknesses

• Large overhead for documentation
• Concrete results are lately achieved
• It is difficult to specify requirements at the early project stage
• Developers lose a contact with the client for a long time
• User requirements meeting is not guaranteed
Agile programming

• Agile Manifesto
  – Individuals and interactions over processes and tools
  – Working software over comprehensive documentation
  – Customer collaboration over contract negotiation
  – Responding to change over following a plan
12 principles behind Agile Manifesto (1)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Software life cycle
SCRUM

• Sprint (short) iterations
• Only 2-4 weeks for one iteration
• Short daily meetings (15 min)
• Three roles:
  – Product Owner
  – Scrum Master
  – Team (utterly self managing)
• Product backlog
Agile programming characteristics

**Pros**
- Time effective (only development & testing)
- Easily fitted to requirements changes
- Can be used when requirements are unknown

**Cons**
- Process depends on people – risk of team change
- No possibility to return to the project after a long break - lack of documentation
- The possibility of serious design errors (e.g. special and emergency situations)
- High cost of changes during software usage
Testing in agile programming

• New or modified code cannot be placed in repository until it passes all the tests.
• Only under this condition effective team work is possible.
• Acceptance tests are designed basing on user stories.
• No user story is considered to be implemented until it passes all the acceptance tests.
Testy akceptacyjne

• Testy akceptacyjne są tworzone na podstawie opowieści użytkownika.
• Klient formułuje scenariusze testowe.
• Jedna opowieść może mieć wiele testów akceptacyjnych.
• Implementacja opowieści nie jest zakończona dopóki nie przejdzie wszystkich testów akceptacyjnych.
Conclusions

• Agile Programming is more effective than classic methodologies (e.g. RUP),
• Agile Programming can assure software quality under some key conditions.
Key conditions for quality

• Close cooperation (communication) with customers in the whole development time.
• Release objectives based on real customers’ needs.
• Keeping good engineering practices.
• Keeping a schedule of frequent releases (a continuous work in the same project)
• Reliable test writing and test passing before each release.
• Passing acceptance tests performed by users.
Agile programming usage

• Simple applications
• Low risk at project failure
• Continuous product maintenance
Sources

• http://www.xprogramming.com
• http://www.extremeprogramming.org
• http://www.agile.com
• http://www.pairprogramming.com
• http://www.martinfowler.com
• http://www.xp123.com
• http://www.thoughtworks.com
• http://www.rolemodelsoftware.com
• http://www.industriallogic.com/xp
• http://www.agilealliance.org