



Nokia Networks and Agile Development

Last updated: August 2006

Public

Currently in Nokia Networks



- 15 projects use Scrum

- Projects range from:

| | | | |
|----------------------|---------|---|-------------|
| • Environments: | Java | ↔ | Traditional |
| • Size: | 10 | ↔ | 150 |
| • Development sites: | 1 | ↔ | 5 |
| • Products: | New | ↔ | Old |
| • Countries: | Finland | ↔ | China |

- Support from a centralized project -> Flexible R&D project

History



- Project “Flexible R&D” started in beginning 2005.
 - One of the goals:
“Find **1 or 2 projects** to pilot “Agile development””.
 - Result:
 - April 2005: 4 projects
 - July 2005; 6 projects
 - December 2005: 9 projects
 - March 2006: 13 projects
 - Original target failed.
 - Reason: Too much interest.
- Constant growing of interest made only “minimal support” possible.

Flexible R&D project



• Jan 2005 → now Flexible R&D project → x4

- What does Flexible R&D do:

- Give trainings: Introduction, Scrum, TDD, A-TDD, CI, Retrospectives
- Facilitation: Retrospectives, Requirement workshops, Iteration planning
- Share Information: Wiki, newsgroups, newsletter
- Organize gatherings: Scrum
- Create communities and user groups: CI, TDD, Scrum
- Coaching: Test-driven-development
- Project support: Problem solving, discussions

- No enforcement, just optional support.

Process



- Projects select their own process
 - **Scrum** been selected almost everywhere
 - Team gradually took **more agile practices** in use
 - Process was **adapted** over time
- No company-wide agile process defined
 - ➔ **values, principles** and some **interfaces** defined.
- Two musts:
 - Short iterations
 - Retrospectives
- Important: **Not iterative when slide**

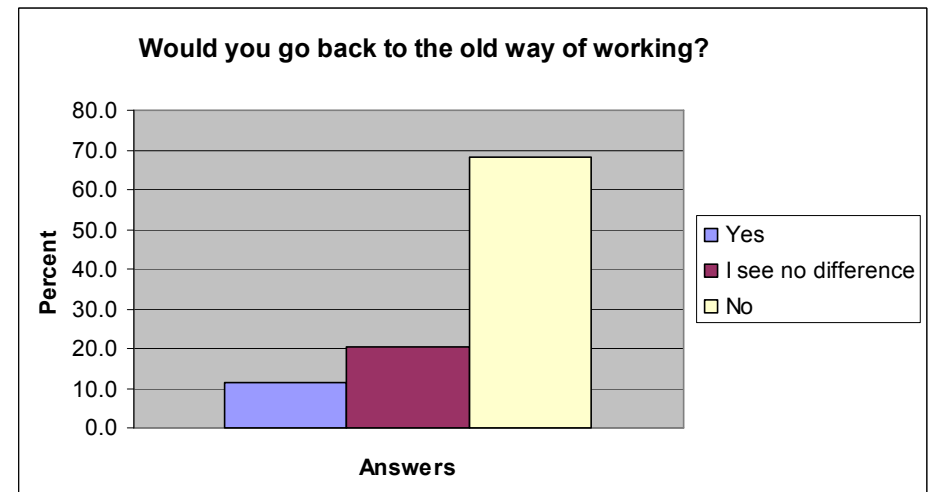
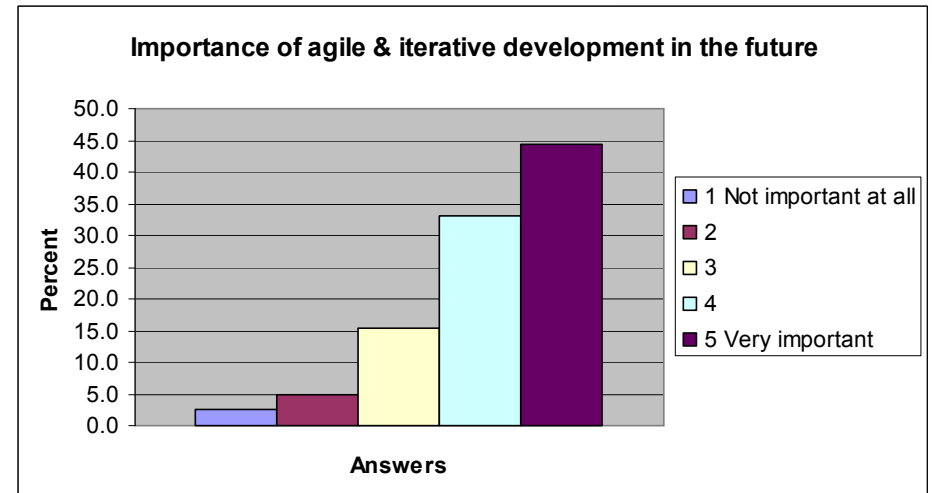
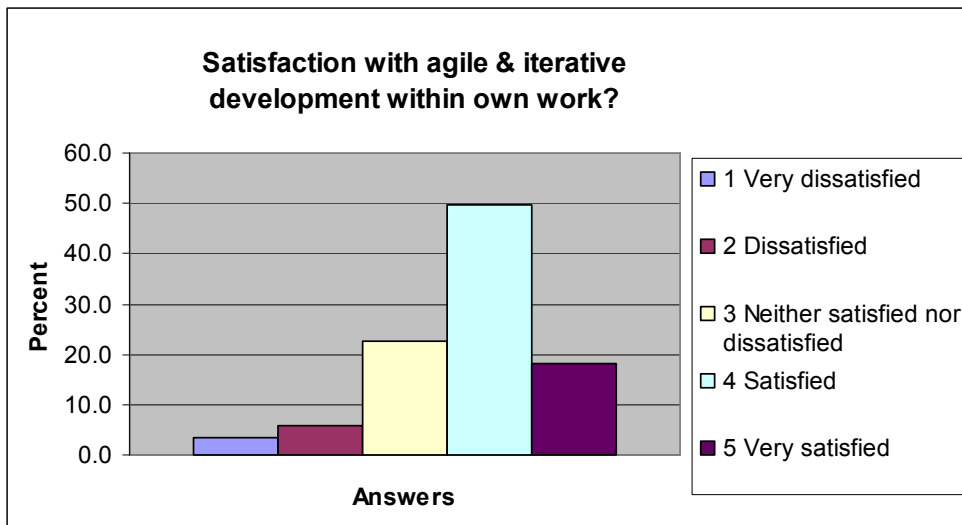
Success



- **No enforcement:** let the projects chose
- Usage of external **consultants**
- Much **open information**
- Linking people together, **communities**
- Provide the **support** for projects when they need
- Focus on both **technical** and **people/organizational** issues

- The two musts
 - Iterations -> Quick feedback
 - Retrospectives -> Inspect and adapt

Survey for people in projects





Background info

Public

No Agile & Iterative when:



- You know when you are not doing iterative development when:
 - Iterations are longer than 2-6 weeks
 - Team tries to complete specifications before programming
 - An iteration does not include testing
 - Iteration does not produce workable code
 - Detailed (task level) plans and accurate estimates are expected at the beginning of a project
 - The iteration plan doesn't reflect what team is doing
- You know when you are not doing agile development when:
 - There is little co-operation within the team
 - Design and code is produced in individual effort
 - Progress is measured by hours spend or documents created instead of working code
 - Builds are done once in three weeks.