Agility and FOSS

whoami

Haïkel Guémar

Fedora Packager since 2006

Senior Software Engineer @ SysFera

Certified "Scum" Master

Values

Agile Manifesto

4 values

1. Individuals and interactions over processes and tools

2. Working software over comprehensive documentation

3. Customer Collaboration over contract negotiation

4. Responding to change over following a plan

FSF Four Freedoms

1. The freedom to run the program for any purpose

2. The freedom to study how the program works and change it to make it do what you wish

3. The freedom to redisitribute copies so you can help your neighbor

4. The Freedom to improve the program and release your improvements to the public

Two attitudes

Agile manifesto describe how to build better software

FSF Four Freedoms how to respect users freedom

Common points

- Similar Ethics

- Focus on teamwork

- Communication as pivot

Technically

- Iteration vs "Release early, release often"

- Empowerment of knowledge: "Software Craftmanship vs meritocracy"

FLOSS achievements

- open code

    ◦ code review "Given enough eyeballs, all bugs are shallow"

    ◦ effective collective ownership

- broader community

    ◦ team ⊂ community

Agile for FLOSS hackers

Principles

- iterative and incremental development

- self-organizing and cross-functionnal teams

- adaptative planning and delivery

- timeboxed approach

Many Methods

- eXtreme Programming

- Crystal methods family

- Kanban

- Scrum (the trendy one)

eXtreme Programming

Created in 1999 by Kent Beck

eXtreme Programming

4 values:

- Communication

- Simplicity

- Feedback

- Courage

- Respect (added in *Extreme Programming Explained, 2nd ed*)

eXtreme Programming

12 practices:

- Fine scale feedback
    - Pair programming
    - Planning game
    - Test-driven development
    - Whole team
- Continuous process
    - Continuous integration
    - Refactoring or design improvement
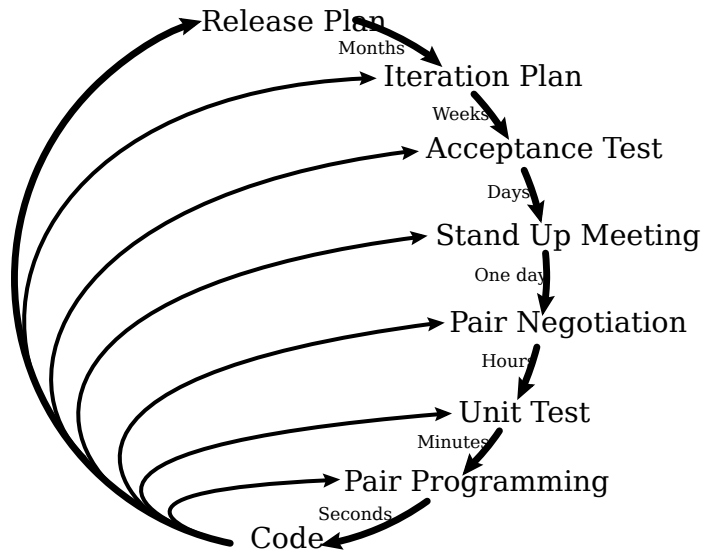   - Small releases

eXtreme Programming

12 practices:

- Shared understanding
    - Coding standards
    - Collective code ownership
    - Simple design
    - System metaphor
- Programmer welfare
    - Sustainable pace

eXtreme Programming

Planning/Feedback Loops

Release Plan

Months

Iteration Plan

Weeks

Acceptance Test

Days

Stand Up Meeting

One day

Pair Negotiation

Hour

Unit Test

Minutes

Pair Programming

Seconds

Code

Scrum

An iterative and incremental agile software development *framework* coined by Jeff Sutherland and Ken Schawber in 1995

Scrum

It has roles:

- Product Owner: the guy who sets the vision

- Scrum Master: the sheep herder

- Team: people who do the actual works

- Stakeholders: people who pay/use your product

Scrum

And meetings (don't worry, they're timeboxed !)

- daily *scrum*

- sprint planning

- sprint review (aka demo)

- retrospective

Scrum

And artefacts:

- Product backlog

- Sprint backlog

- Burndown chart

Scrum

The process:

Product Backlog   Sprint Backlog   Sprint   Working increment of the software

24 h

30 days