Agile development
Tools and best practices

Walter Van Herck
October 24, 2013
Outline

• General goals in product development
• Agile manifesto
• Waterfall versus agile development
• Agile tools
• Agile processes
AWESOMENESS

When I get sad, I stop being sad and be awesome instead. True Story.
General goals in product development

• Deliver on a timely basis
• Increase quality and stability
• More efficient work
• Better control of progress and planning
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
Waterfall versus Agile (timeline)

- Planning
- Design
- Development
- Testing
- Debugging
- Releasing

- Planning
- Design
- Development
- Testing
- Debugging
- Releasing

October 24th, 2013
Waterfall versus Agile (roles)

- Project Manager
- Developers
- Testers

Project management
Programming
Testing
Quality control

Project Team
Typical agile development process

- Define use cases
- Prioritize
- Work Breakdown
- Sprint Planning
- Sprint
- Release
Development sprints

- Sprint planning meeting
- Taking ownership
- Development and continuous testing, discussion
- Sprint review meeting (internal release)
Checklist (Joel on Software)

- Do you use source control?
- Can you make a build in one step?
- Do you make daily builds?
- Do you have a bug database?
- Do you fix bugs before writing new code?
- Do you have an up-to-date schedule?
- Do you have a spec?
- Do programmers have quiet working conditions?
- Do you use the best tools money can buy?
- Do you have testers?
- Do new candidates write code during their interview?
- Do you do hallway usability testing?
Agile tools

- Source control: git
- Nightly builds: Teamcity
- Bug and issue tracking: Redmine
- Testing:
  - Unit tests: googletest framework
  - Functional tests with the help of IsGISAXS
- Communication tools:
  - Skype
  - Person-to-person
Sprint 12

17 closed (91%) - 12 open (39%)

Related issues:
- Bug #214: Weird double delete somewhere in boost/gzip while running functional tests on mac+gcc48
- Bug #220: GUI: provide GUI compilation under Qt5/linux
- Bug #229: PythonAPI: provide Py++ code generation under Linux
- Bug #244: OutputDataIOFactory: make consistent [x][y] axis ordering while writing/reading OutputData from *.txt and *.ima files
- Bug #245: Inconsistent fact 2^PI in FormFactorHemiSpheroid
- Bug #246: PythonAPI: resolve bug with 32bit pointer type generated by Py++
- Bug #247: PythonAPI: kvector algebra disappear from python
- Bug #248: Problem with form factor of cylinder
- Feature #60: Framework general: implement simple logging tool with control of output level
- Feature #212: Provide libBornAgainFitCore with ROOT minimizer wrappers
- Feature #216: Provide python example of fitting using external ROOT
- Feature #219: Provide python script editing in GUI
- Feature #255: Building: prevent build failure in the case of ROOT is absent on the system
- Support and Documentation #157: Update NS Theory documents
- Support and Documentation #166: Create architectural overview documentation
- Support and Documentation #200: Doxygen comments for all classes
- Support and Documentation #212: Write description of logicals01 example for User Manual using latex
- Support and Documentation #250: Provide corrected "number of line of code vs time" script and plot
- Support and Documentation #257: Make tarball BornAgain.tar.gz for the Workshop
- Refactoring #204: Implement ISampleVisitor
- Refactoring #223: Get rid from Instance() method for all singletons
- Refactoring #227: Check replacement of Coordinate3D<int> by Geometry::BasicVector3D<int>
- Refactoring #238: Restructure Examples directory
- Refactoring #248: PythonAPI: redone Py++ boost::python API code generation and installation
- Testing #209: Run vaqrind on all FunctionalTests
- Testing #228: Add mesocrystal test in Functional TestCor and TestPyCore
- Testing #233: PythonAPI: learn how to create shared_ptr objects in python and deliver them into C++
- Testing #255: Repair compilation under JCNS linux, OpenSuse 12.3

Redmine -- Impressum
Teamcity

Configure your email notifications on your settings page.
Unit tests
Practices in our computing group

- Sprints of 3-4 weeks
- Keeping track of status, to do, etc.
- Rapid escalation of encountered problems
- Continuous testing (unit and functional)
- Regular meetings
- Accountability and ownership for tasks
- Automatization of release builds
Summary

- Difference between classical waterfall and agile development
- Tools and practices
- Be open to change