



Mantid

Requirements in a long term agile project

Nick Draper

Tessella

www.mantidproject.org

Overview

- Requirements in Agile Projects
- Requirements throughout Mantid
- The problems
- Our solution



Key Principles of Agile projects

- Active user involvement is imperative
- Requirements evolve but the timescale is fixed
- Capture requirements at a high level; lightweight & visual
- Develop small, incremental releases and iterate
- Focus on frequent delivery of products
- Apply the 80/20 rule
- Testing is integrated throughout the project lifecycle – test early and often

Key Principles of Agile projects

- Active user involvement is imperative
- Requirements evolve but the timescale is fixed
- Capture requirements at a high level; lightweight & visual
- Develop small, incremental releases and iterate
- Focus on frequent delivery of products
- Apply the 80/20 rule
- Testing is integrated throughout the project lifecycle – test early and often

Key Principles of Agile projects

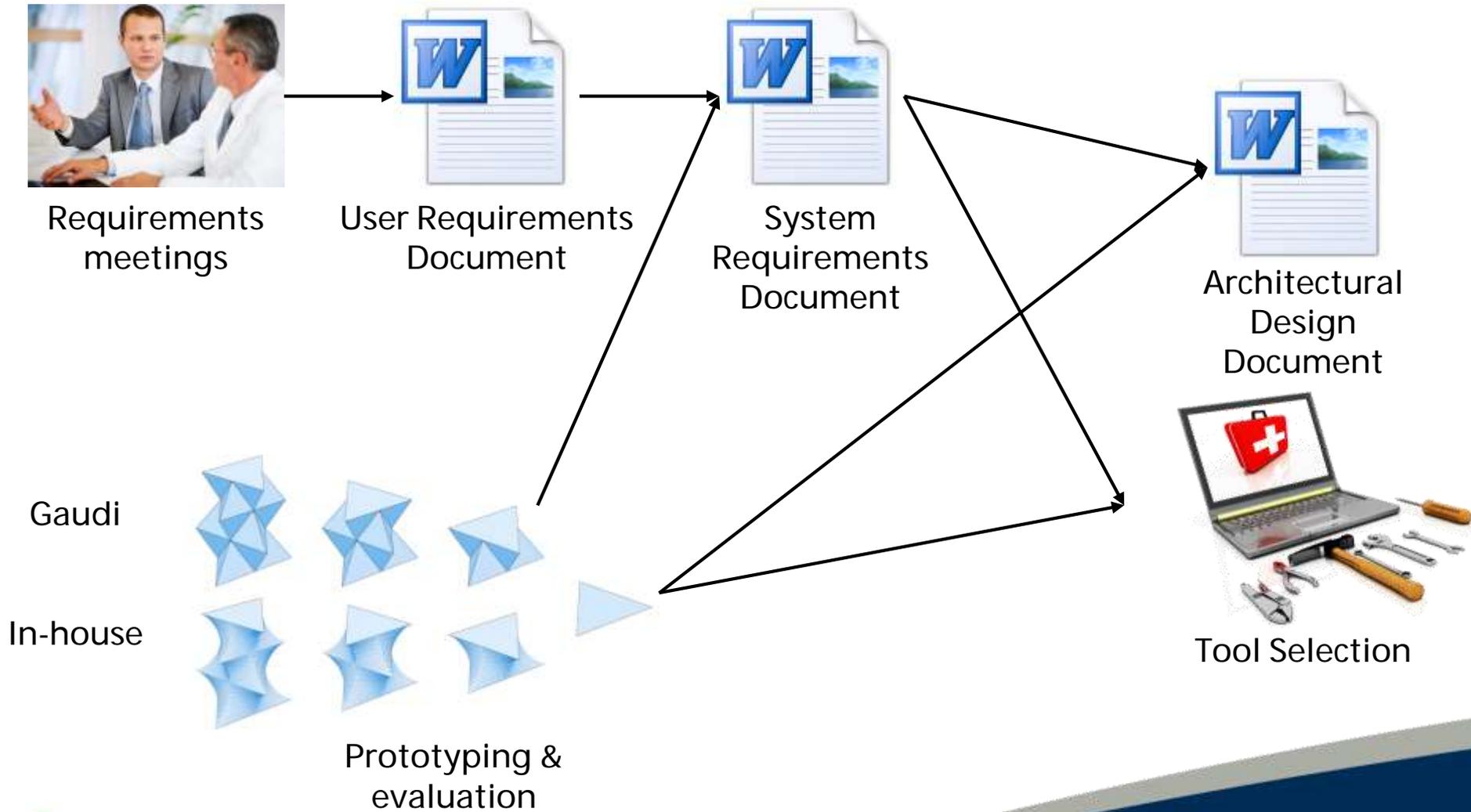
- Active user involvement is imperative
- Requirements evolve but the timescale is fixed
- Capture requirements at a high level; lightweight & visual
- Develop small, incremental releases and iterate
- Focus on frequent delivery of products
- Apply the 80/20 rule
- Testing is integrated throughout the project lifecycle – test early and often

Project Goals

- Goals
 - Consolidate the data reduction/analysis software for neutron scattering without restricting the needs of the instrument scientists
- Key requirement
 - Create a Data Analysis framework
 - not instrument or technique/dependent
 - Cross-platform
 - Windows, Linux, Mac
 - Easily extensible
 - Open source



Mantid in the beginning



Mantid in the beginning

Good

- Wide cross section of scientists
- Separated high level from details
- Built relationships
- Documented and agreed requirements

Requirements

Bad

- Tough to separate vital detail from the rest
- Stakeholder coverage

Prototyping

- Learning from other projects
- Evaluation of current software
- Gathering best practice

- Accuracy vs. time
- Bias

Mantid early development

- 1 month iterations
- Development
 - High risk design features
 - Plug in architecture
 - Python Integration
 - Vital common data structures
 - Workspace2D
 - Workspace History
 - Instrument Geometry
 - Common algorithms
 - LoadRaw
 - Rebin
 - ConvertUnits
- Unit testing, continuous integration from the start

Evaluated by:

- Lead Scientist
- Project Manager
- Project Team

Mantid early development

Good

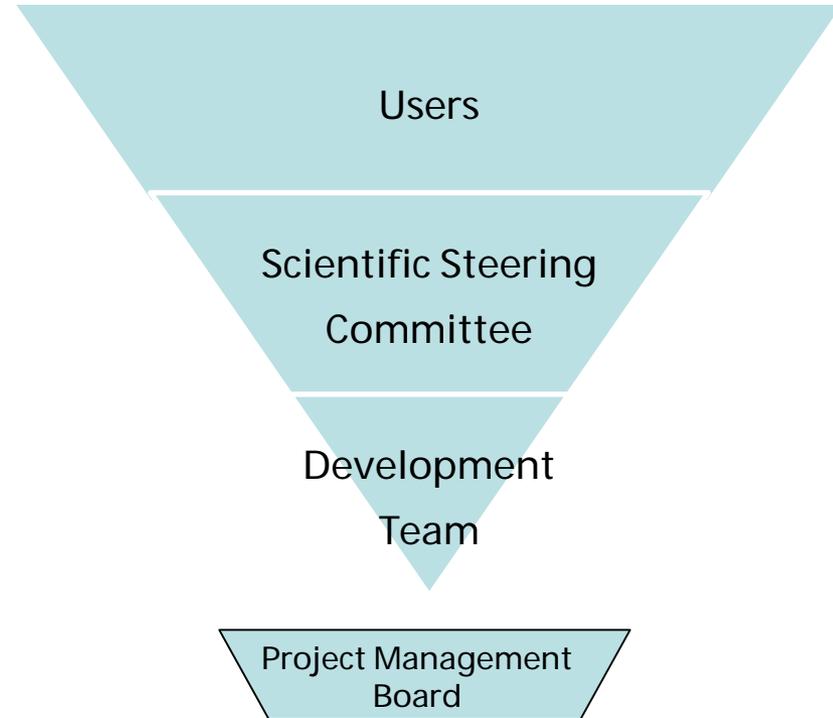
- Rapid development
- De-risking design decisions
- Validation of framework approach
- Gathered momentum within the team

Bad

- No wider stakeholder engagement
- Little immediate use
- Low visibility to senior management

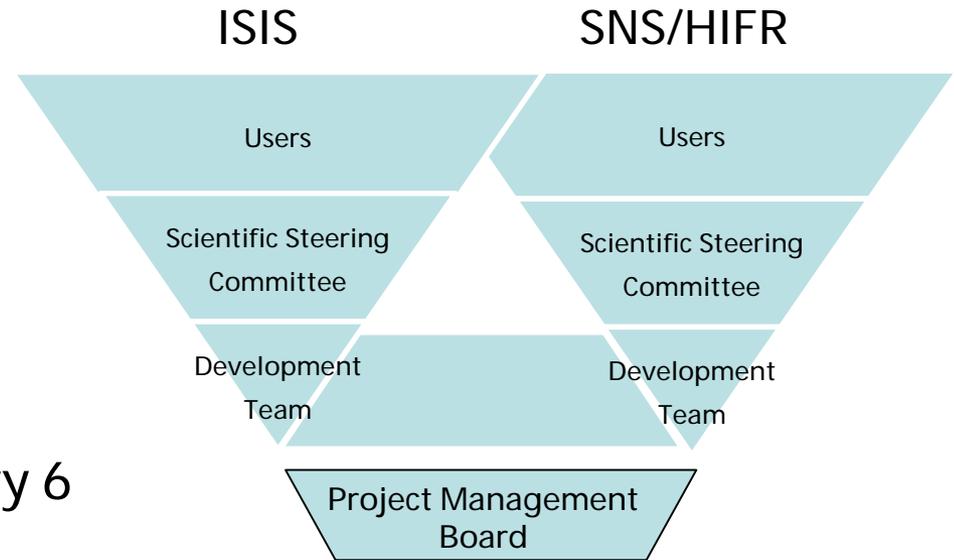
Improving Stakeholder engagement

- Scientific Steering Committee
- 1 -> 3 month releases
 - Better fit with cycle timings
- User Training
- Release Notes



Expanding the project

- Scientific Steering Committee
 - Every 3 months
 - Local scientific steering committees
 - Joint teleconferences every 6 months
 - Two way information flow
 - Project progress -> scientists
 - Requirements & priorities -> dev team



Problems

- The two facilities were in two very different situations
 - ISIS had been involved since the start, and were looking to extend capability.
 - SNS were finding how to fit Mantid to their needs
- Information did not flow reliably beyond members
 - Progress
 - Requirements & Priorities
- How to handle strategic and longer term tasks?
 - Requirements
 - Process
 - Visibility

Annual Joint Scientific Steering Committee

- Extract longer term requirements
 - Ask less often!
 - Place greater importance on the event
 - Spend time, listening and feeding back
 - Deliver on your promises
- Encourage wider collaboration
 - Face to face meetings lead to much more agreement
 - The joint meeting allow time for other conversations with remote colleagues



Release Presentations

- Just before each release
- Everyone invited
 - But participation optional
- Development team present progress
- Different focus topic each time
 - Covers complete functionality in that area, not just improvements
- Meet the team at the end
 - Opportunity to chat to the scientists and solve any problems.

Conclusion

- A good overview at the start of an Agile project is essential
- Requirements gathering approaches need to change during a long term project
- Do not assume that others will pass the message of the good work you are doing onwards
- One collaborative project is a great opportunity to found others