Agile Software Requirements

Matthew Renze

Iowa State University

COMS 409 – Software Requirements

Purpose

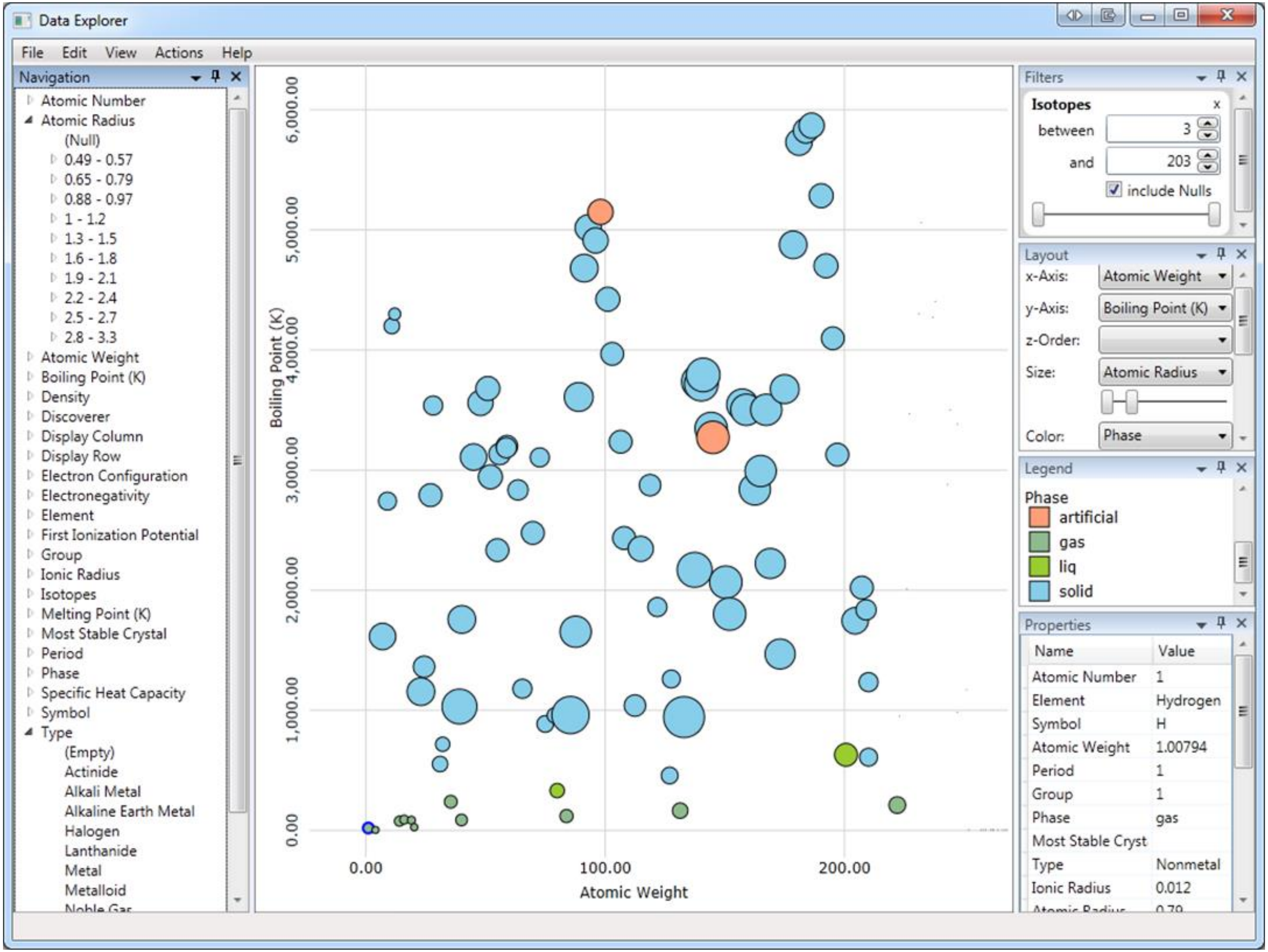
- Introduce you to Agile software development
- Discuss Agile software requirements

Overview

- What is Agile?
- Waterfall vs. Agile
- User Stories
- Embedded Documentation
- Non-functional Requirements
- Q & A

About Me

- Independent software consultant
- 14 years of professional Agile software development experience
- Data-driven desktop, server, and web apps
  - Web-based GIS data warehouse
  - Energy data ETL application
  - Global data management system
  - Intelligent lighting control systems
  - Open source data explorer



Isotopes x

between 3

and 203

☑ include Nulls

x-Axis: Atomic Weight

y-Axis: Boiling Point (K)

z-Order:

Size: Atomic Radius

Color: Phase

Phase

- 🟧 artificial
- 🟩 gas
- 🟩 liq
- 🟦 solid

Name	Value
Atomic Number	1
Element	Hydrogen
Symbol	H
Atomic Weight	1.00794
Period	1
Group	1
Phase	gas
Most Stable Cryst	
Type	Nonmetal
Ionic Radius	0.012
Atomic Radiu	0.79

Education

- BS in Computer Science
- BA in Philosophy
  - Minor in Economics
  - Focus on Artificial Intelligence and Machine Learning
- AS in MIS
- AS in Business Administration

IOWA STATE UNIVERSITY

DMACC
DES MOINES AREA
COMMUNITY COLLEGE

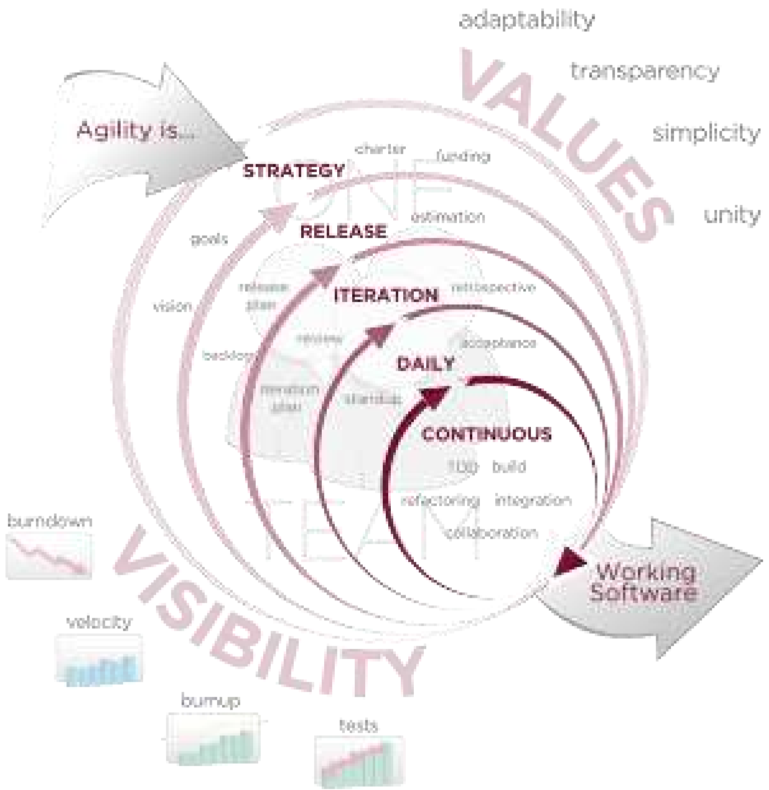
Public Speaking

- Events:
  - Iowa Code Camp
  - Nebraska Code Camp
  - Iowa .NET Users Group
  - Agile Iowa
- Topics:
  - Lean / Agile
  - Data Analysis
  - Patterns, Practices, and Principles

What is Agile?

What is Agile?

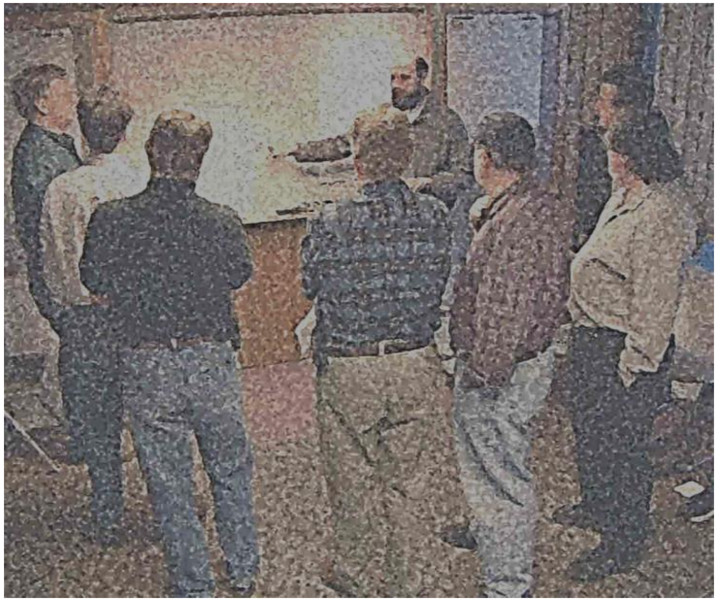
- Started with the Agile Manifesto
  - 4 value propositions
  - 12 principles
- Common set of practices across several methodologies



Source: Wikipedia

Agile Values

- Individuals and interactions
  - over processes and tools
- Working software
  - over comprehensive documentation
- Customer collaboration
  - over contract negotiation
- Responding to change
  - over following a plan



Source: http://agilemanifesto.org/

12 Principles of Agile

1. Continuous delivery of value
2. Embrace changing requirements
3. Frequent deployment
4. Customer collaboration
5. Motivated individuals
6. Face-to-face conversation

12 Principles of Agile

7. Working software as measure of progress
8. Sustainable development
9. Technical excellence
10. Simplicity
11. Self-organization
12. Continuous improvement

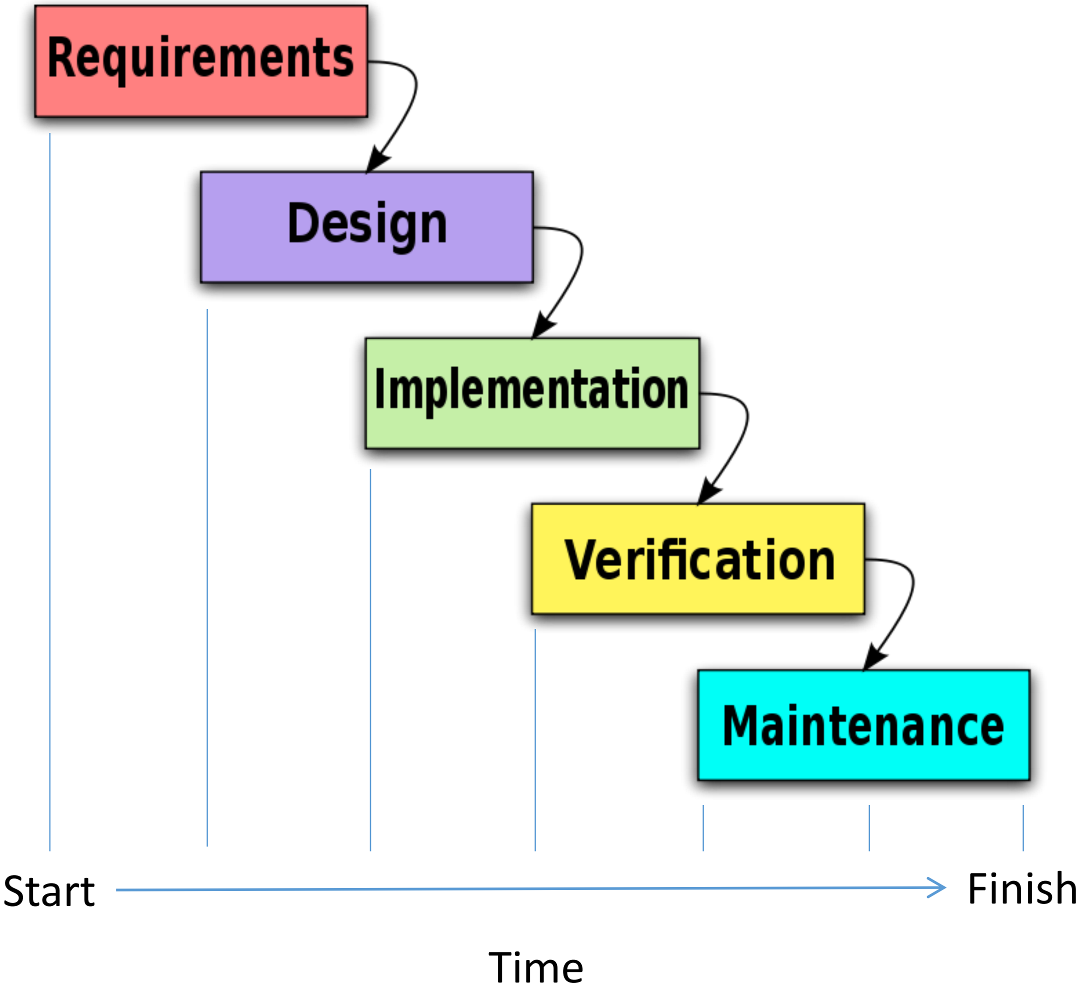
Agile Methodologies

- Scrum
- XP
- Kanban
- Lean
- And many more…

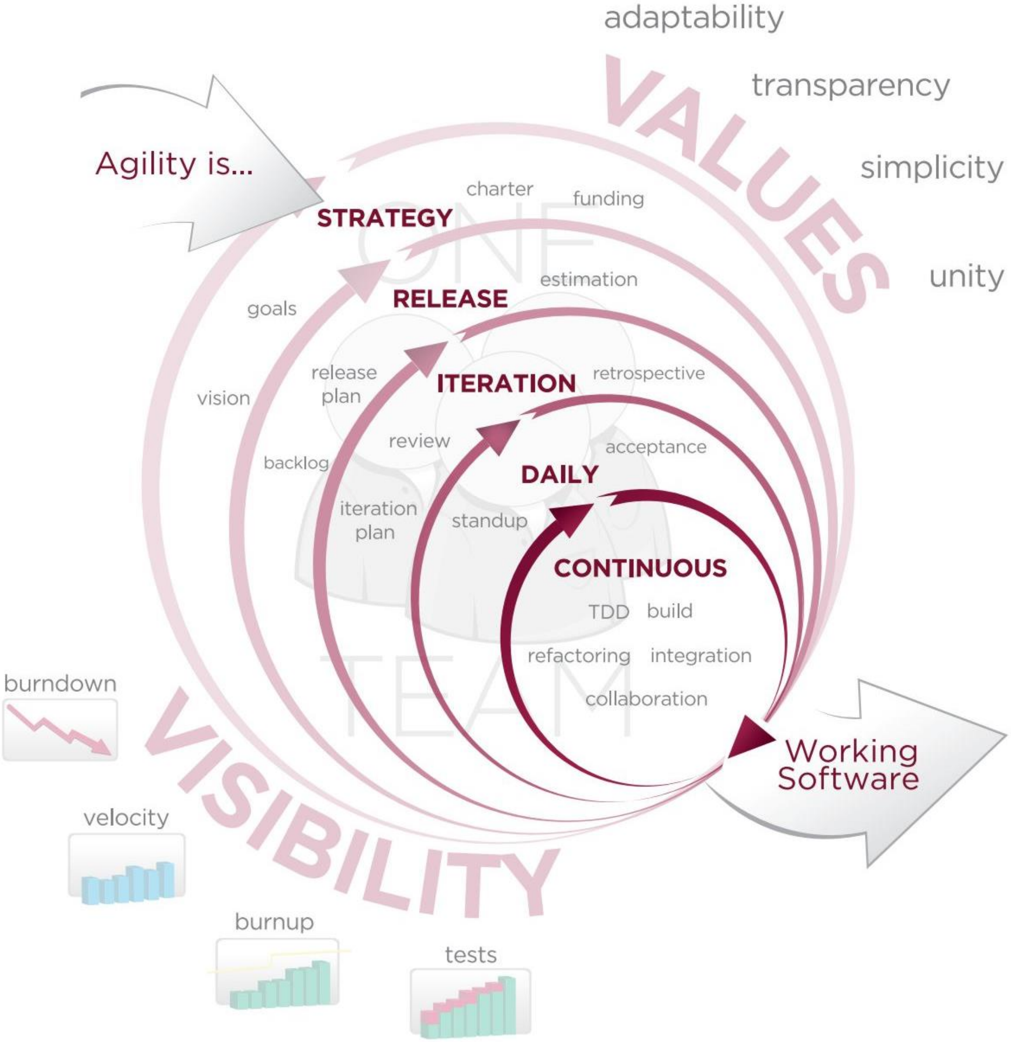
Source: http://parkertoddloesch.files.wordpress.com/2011/09/umbrella.jpg

Waterfall vs. Agile

Waterfall


Requirements
Design
Implementation
Verification
Maintenance
Start
Finish
Time

Agile



Source: Wikipedia

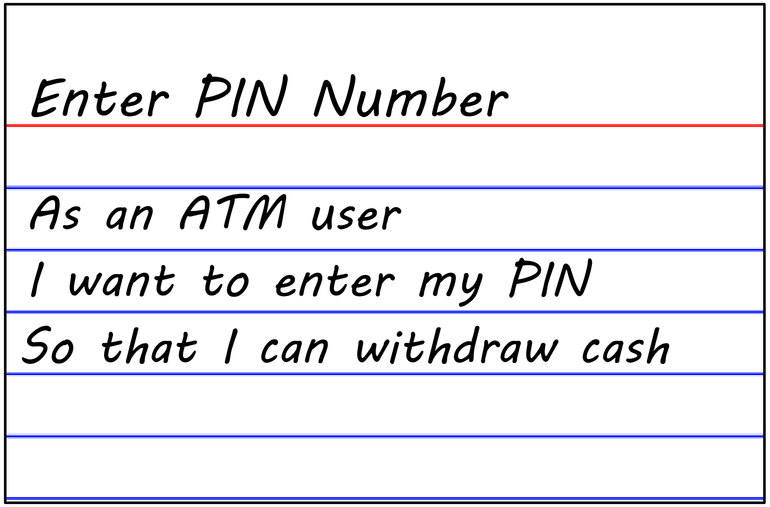
Waterfall vs. Agile Processes

	Waterfall	Agile
Measure of Success	Execution of the plan	Working software
Management Culture	Command and control	Self-organization
Requirements / Design	Big and upfront	Just-in-time / minimal
Code / Implementation	Code first and test later	Test and code together
Testing and QA	Big test plan / test last	Test early / continuously
Planning and Scheduling	Large detailed plan	Short, iterative increments

User Stories

User Story

- Short description of functionality that will provide value to a user
- Contains:
  - Title
  - Description
  - Acceptance Criteria
- Placeholder for a conversation to occur



Enter PIN Number

As an ATM user

I want to enter my PIN

So that I can withdraw cash

User Story Example

Title: Enter Personal Identification Number (PIN)

Description:

As an ATM user

I want to enter my PIN

So that I can withdraw cash

Acceptance Criteria:

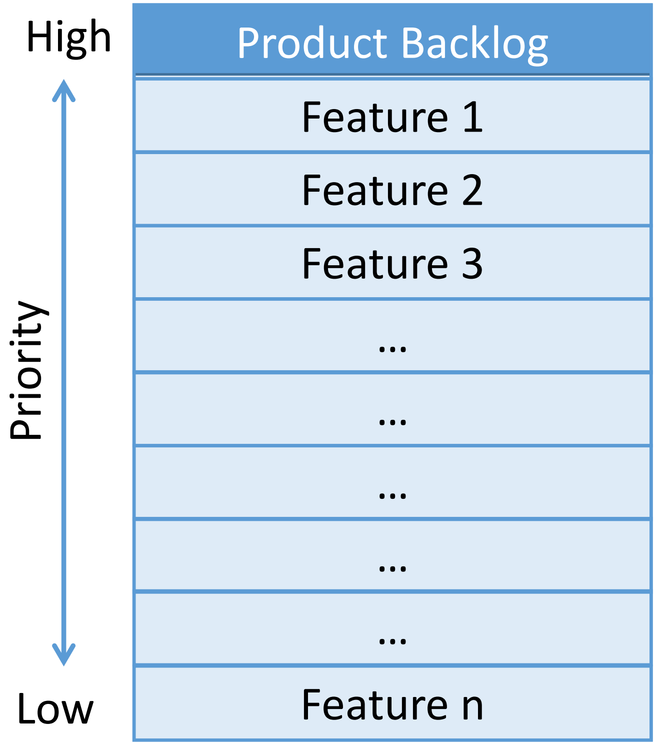
PIN must be four digits long

PIN must not allow alpha or special characters

PIN must be entered within 30 seconds
or the transaction will be canceled

Agile Requirements

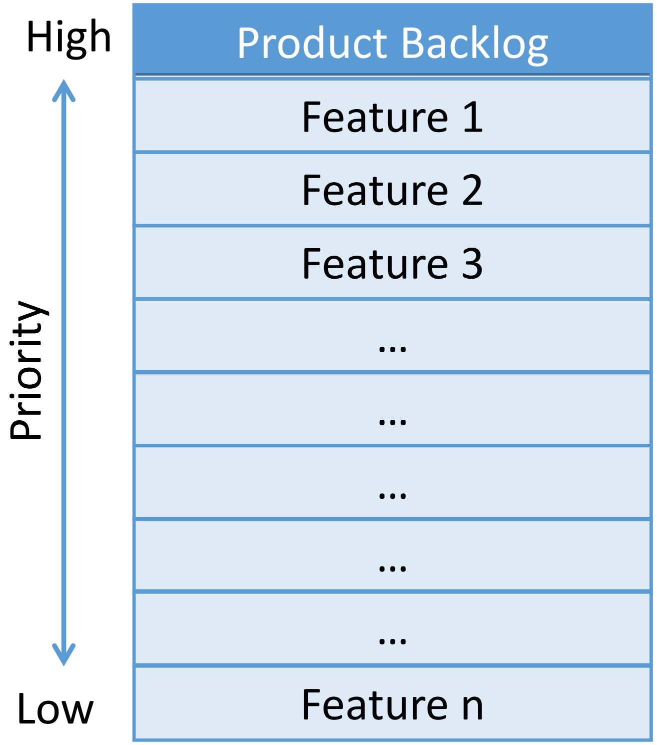
- Feature requests captured in user stories
- User stories are prioritized in product backlog
- Work on user stories in priority order



Product Backlog
Feature 1
Feature 2
Feature 3
…
…
…
…
…
Feature n

Agile Requirements

- For each user story:
  - Gather requirements through collaboration
  - Implement functionality using Agile practices
  - Requirements become embedded in code



Agile Requirements Gathering

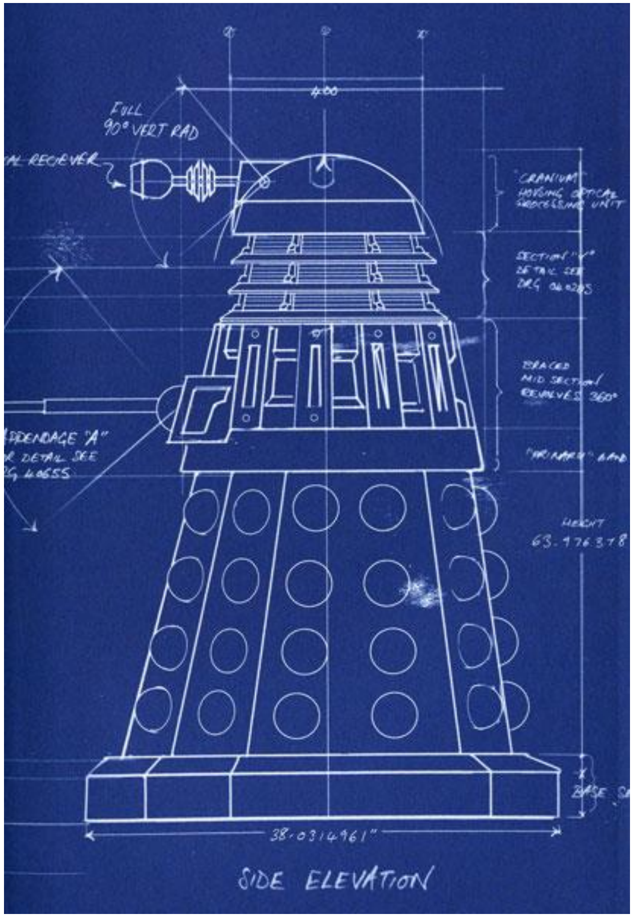
- Minimal documentation
- Communication and collaboration are critical
- Uses whole-team approach
- On-site user representation
- Rapid feedback loop is critical to success

Source: http://www.dvdizzy.com/peanuts-1970s-vol2.html

Embedded Documentation

Waterfall Requirements Documentation

- In Waterfall:
  - Documentation is the blueprint
  - Code is the product being produced
- Documentation is:
  - Often out of date
  - Not read frequently
  - Not executable


400
FULL 90° VERT RAD
AL RECIEVER
"CRANIUM" HOUSING OPTICAL PROCESSING UNIT
SECTION "V" DETAIL SEE DRG 06-0205
BRACED MID SECTION REVOLVES 360°
APPENDAGE "A" R DETAIL SEE RG 4-0655
"PRIMARY" BAND
HEIGHT 63.976378
BASE S
38.0314961"
SIDE ELEVATION

Source: http://www.thinkgeek.com/product/f373/images/16693/

Agile Requirements Documentation

- In Agile:
  - Code is the blueprint
  - Working software is the product being produced
- Code is:
  - Always up to date
  - Continuously read
  - Executable
- Requirements are embedded in the code

Source: http://www.firebox.com/product/3621/Dalek-Replica

Agile Practices for Embedded Documentation

Practice	Purpose
Test-Driven Development (TDD)	Low-level Behaviors
Behavior-Driven Development (BDD)	High-level Behaviors
Domain-Driven Design (DDD)	High-level Policy / Domain Logic
Domain-Specific Languages (DSL)	Human-readable code for business
Clean Code	Reader-centered code for devs.

Non-Functional Requirements

Non-Functional Requirements

- Maintainability
- Performance
- Reliability
- Security
- Testability
- Usability

Agile NFRs

Driven by Users:

- Performance
- Security
- Usability

Driven by Developers:

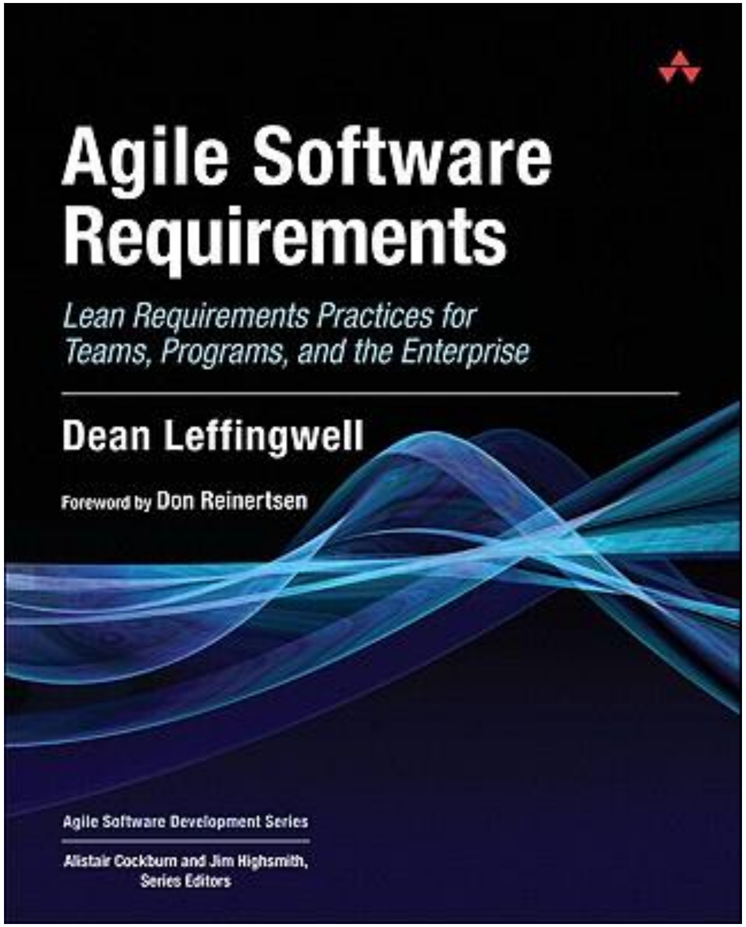
- Maintainability
- Readability
- Testability

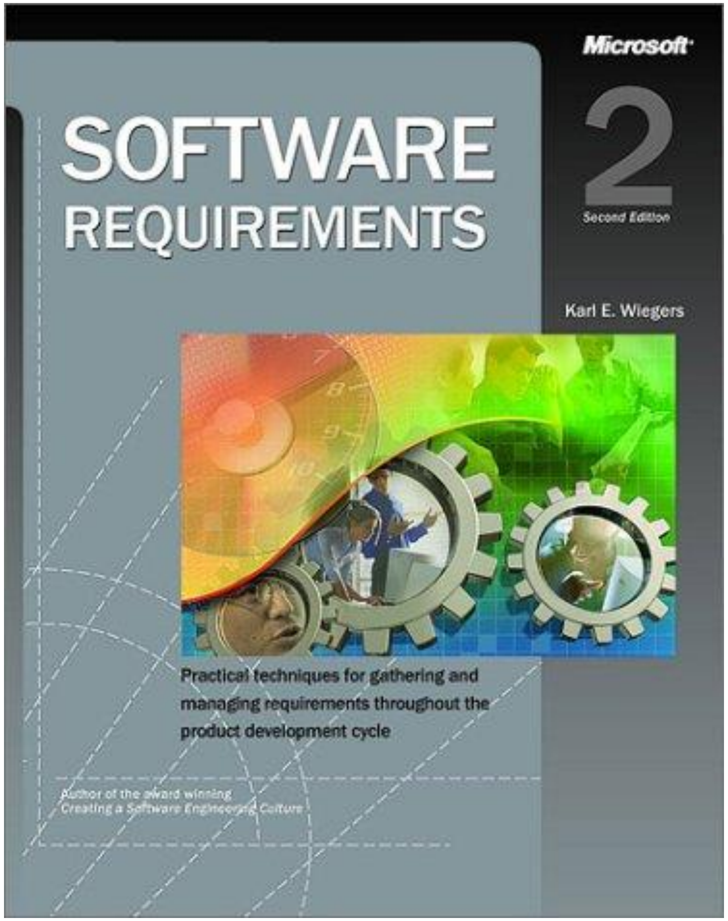
Conclusion

Conclusion

- What is Agile?
- Waterfall vs. Agile
- User Stories
- Embedded Documentation
- Non-functional Requirements

Recommended Reading


Agile Software Requirements
Lean Requirements Practices for Teams, Programs, and the Enterprise
Dean Leffingwell
Foreword by Don Reinertsen
Agile Software Development Series
Alistair Cockburn and Jim Highsmith, Series Editors


Microsoft
SOFTWARE REQUIREMENTS
2
Second Edition
Karl E. Wiegers
Practical techniques for gathering and managing requirements throughout the product development cycle
Author of the award winning Creating a Software Engineering Culture

Contact Info

Matthew Renze
info@renzeconsulting.com

Renze Consulting
www.renzeconsulting.com

Data Explorer
http://www.data-explorer.com

Q & A