Purpose

• Intro to Agile and Scrum
• Prepare you for the industry
• Questions and answers
Overview

• Intro to Agile
• Waterfall vs. Agile
• Intro to Scrum
• Scrum Team
• Scrum Events
• Scrum Artifacts
• Q & A
About Me

• Independent software consultant
• 14 years of professional 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
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
About Me

• Agile coaching and mentoring
• Regular public speaking
• Open-source software
Introduction to Agile
What is Agile?

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

What is Agile?

Agile is *not*:

• A software development methodology itself

• A silver bullet for all your software woes

Source: http://www.best-story.net/userfiles/silver-bullets.jpg
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
Scrum

• Small teams
• Three roles
• Product backlog
• Sprints
• Daily stand-up

Extreme Programming (XP)

- Small teams
- Customer representative
- Iterative development
- User stories
- Many practices
  - Pair programming
  - Test-driven development
  - Continuous Refactoring

Source: http://static.caloriecount.about.com/images/medium/doritos-extreme-tortilla-chips-29310.jpg
Kanban

- Visualize the workflow
- Limit work-in-progress
- Manage flow
- Feedback loops
Lean

• Eliminate waste
• Focus on value
• Reduce inventory and cycle times
• Continuous process improvement

Agile Methods Used

- **Scrum/XP Hybrid**: 11%
- **Custom Hybrid**: 9%
- **Scrumban**: 7%
- **Kanban**: 4%
- **Don’t Know**: 4%
- **XP**: 2%
- **FDD**: 2%
- **Lean**: 2%
- **Other**: 2%
- **Agile Unified Process**: 1%
- **Agile Modeling**: 1%
- **DSDM Atern**: 1%

*Original Source: Version One - State of Agile Survey 2012*
The Waterfall Method
The Agile Method

- Frequently deliver small incremental units of functionality
- Define, build, test and evaluate cycles
- Maximize speed of feedback loop
Waterfall Assumptions

• All requirements for a project can be defined given enough time
• Changes to requirements will be small and manageable
• Architecture and planning can create predictable system integration outcome
• Software unknowns can be eliminated on a predictable schedule
Agile Assumptions

• Requirements are just assumptions until they are validated
• Requirements can and will changes over time
• We need continuous integration of all of the pieces to avoid late-integration issues
• There is a high degree of uncertainty when creating new software projects
Waterfall Constraints

- **Scope**: Fixed
- **Resources**
- **Schedule**: Estimated
- **Waterfall**
  (plan driven)
Agile Constraints

Scope

Resources

Schedule

Waterfall (plan-driven)

Fixed

Estimated

Agile (value-driven)

Resources

Schedule

Scope
Agile Constraints

• Fixed team size
• Fixed releases
• Estimated features
• Team controls quality

![Diagram showing the balance between Agile (value-driven) and constraints: Fixed team size, Fixed releases, Estimated features, and Team controls quality. The diagram illustrates the relationship between Resources, Schedule, Fixed, Estimated, and Scope.]
## Waterfall vs. Agile Processes

<table>
<thead>
<tr>
<th>Process</th>
<th>Waterfall</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure of Success</td>
<td>Execution of the plan</td>
<td>Working code</td>
</tr>
<tr>
<td>Management Culture</td>
<td>Command and control</td>
<td>Self-organization</td>
</tr>
<tr>
<td>Requirements and Design</td>
<td>Big and upfront</td>
<td>Just-in-time</td>
</tr>
<tr>
<td>Coding / Implementation</td>
<td>Code first and test later</td>
<td>Code and test together</td>
</tr>
<tr>
<td>Testing and QA</td>
<td>Big, planned, and test last</td>
<td>Continuous and test early</td>
</tr>
<tr>
<td>Planning and Scheduling</td>
<td>Large detailed plan</td>
<td>Short, iterative increments</td>
</tr>
</tbody>
</table>
Context is Important

- Neither Agile nor Waterfall is better than the other
- Context is critical
- Examples:
  - Novel web app => Agile
  - Space shuttle control software => Waterfall
Intro to Scrum
What is Scrum?

• Rugby term
• Process framework
• Agile methodology
• Empirical Process Control vs. Command and Control

What is Scrum?

• Iterative / incremental
• Frequent delivery
• Working software
• Continuous feedback
• Dev-team oriented
• Consists of:
  • Team
  • Events
  • Artifacts

Source: http://www.telegraph.co.uk
Three Pillars of Scrum

• Transparency
• Inspection
• Adaptation
Transparency

• Visibility into the project
• Information radiators
• End-of-sprint demos
• Common definition of “Done”
Inspection

• Frequent inspection of progress
• Detection of undesirable variance
• Not frequent enough to cause interference
• Requires the right people doing the inspection
Adaptation

• Adjustments are made to correct undesired variances
• Adjusting earlier minimizes further deviations
• Inspection and adaptation occurs during scrum events
• Purpose is continuous improvement
Scrum Team
Scrum Team Roles

• Development Team
• Product Owner
• Scrum Master
Development Team

• Does work and creates value
• Small teams (7 ± 2)
• Self-organizing
• Cross-functional
• No titles (preferably)

Product Owner

• Responsible for ROI of project
• Represents users and stakeholders
• One person, not a committee
• Manages product backlog

Scrum Master

• Responsible for scrum process
• Facilitates ceremonies
• Coaches the team
• Removes impediments
• Is not a project manager

Source: http://www.cbssports.com/
Scrum Events
Scrum Events

• Sprint
• Sprint Planning
• Daily Stand-up Meeting
• Sprint Review
• Sprint Retrospective
Sprint

• Time box of iteration
• Fixed interval
• 1-4 weeks
• Team works on items in sprint backlog
• Ends with potentially shippable software

Sprint Planning

• Start of sprint
• Goals defined
• Tasks identified
• Timeboxed (2hr / wk)
• PO presents prioritized backlog items
• Team pulls items into sprint backlog

Daily Stand-up Meeting

• Occurs each day
• Short daily meeting
• Synchronize the team
• Timebox 15 min / day
• Same place and time
• It is not a status update meeting

Daily Stand-up

Three questions:
1. What have you done since the last stand-up?
2. What do plan to do today?
3. Do you have any impediments / blockers?
Sprint Review

• End of sprint
• Opportunity to inspect
• Timebox (1 hr/wk)
• PO identifies what has been completed
• Dev team demos completed functionality
• Stakeholders observe and provide feedback

Source: http://pic.twitter.com/gfzYl0XG8c
Sprint Retrospective

• End of sprint
• Inspection and adaptation
• Timebox (3 hr / 4 wk)
• Two questions:
  • What went well in the past sprint?
  • What could be improved for the next sprint?
• Continuous improvement

Source: http://spin.atomicobject.com/2014/04/07/improve-retrospective/
Scrum Artifacts
Scrum Artifacts

• Product Backlog
• Sprint Backlog
• User Stories
• Scrum Board
Product Backlog

- List of features for product
- Ordered by business value or ROI
- Highest priority on top
- Create and deliver features in order
- Owned by product owner
Sprint Backlog

• List of features to be completed during sprint
• Owned by the development team
• No work can be added without team’s approval
• Highly visible to everyone

<table>
<thead>
<tr>
<th>Sprint Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature 1</td>
</tr>
<tr>
<td>Feature 2</td>
</tr>
<tr>
<td>Feature 3</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Feature n</td>
</tr>
</tbody>
</table>
Backlog Items

- User Stories
- Bugs / Defects
- Architecture / Infrastructure
- Technical Debt
- Research (aka. Spikes)
User Story

• Short description of functionality that will provide value to a user

• Contains:
  • Title
  • Description
  • Acceptance Criteria

• Placeholder for a conversation to occur

<table>
<thead>
<tr>
<th>Enter PIN Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>As an ATM user</td>
</tr>
<tr>
<td>I want to enter my PIN</td>
</tr>
<tr>
<td>So that I can withdraw cash</td>
</tr>
</tbody>
</table>
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
Scrum Board

- Tool to visualize progress within sprint
- User stories and tasks written on post-it notes
- Tasks moved from:
  - To do
  - In progress
  - Done

The Whole Process

Product Backlog → Sprint Backlog → Sprint

Conclusion
Agile is:

Individuals and interactions
  over processes and tools

Working software
  over comprehensive documentation

Customer collaboration
  over contract negotiation

Responding to change
  over following a plan
Scrum is:

3 Roles
  • Development Team
  • Product Owner
  • Scrum Master
Scrum is:

4 Events

• Sprint Planning
• Daily Stand-up Meeting
• Sprint Review
• Sprint Retrospective
Scrum is:

4 Artifacts
- Product Backlog
- Sprint Backlog
- User Stories
- Scrum Board
Questions / Feedback

• What was one thing you thought was valuable?
• What is one thing you would change?
Contact Info

Matthew Renze
• Twitter: @matthewrenze
• Email: info@matthewrenze.com
• Website: www.matthewrenze.com

Renze Consulting
www.renzeconsulting.com