UNDERSTAND AGILE
START TO FINISH

* ONE DAY
* ONE TRACK

7:00	Registration	
7:45	Initial Remarks	Phil Japikse
8:00	Intro to Scrum	Mark Windholtz
9:00	Engineering Practices	Jim Weirich
	Break	
10:15	Five Things to do to be more Agile	Brian Prince
11:15	Why Agile Fails	Chris Nelson
	Lunch and Panel Discussion	
1:30	An Agile Success Story	Sean Heuer
2:30	Enterprise Agility	Phil Japikse
3:30	Retrospectives	Joe Obrien
4:30	Closing Remarks and Raffle	


Microsoft


ATLASSIAN


gaslightsoftware


Apress
THE EXPERT'S VOICE


SDS
Strategic Data Systems


PROFITABILITY.net


MAX
max technical training


Seapine Software


telerik
deliver more than expected

PINNACLE SOLUTIONS GROUP, INC.

Microsoft GOLD CERTIFIED Partner

telerik deliver more than expected

Enterprise Agility:
Applying Agile in a Waterfall World

By
Philip Japikse
Phil.japikse@pinnsg.com
MVP, MCSD.Net, MCDBA, CSM, CSP
Principal Consultant
Pinnacle Solutions Group

Who am I?

- Principal Consultant, Pinnacle Solutions Group
- Microsoft MVP
- MCSD, MCDBA, CSM, CSP
- Member Telerik Community Speaker Bureau
- Enterprise Application Architect
- Trainer/Mentor/Speaker
- Lead Director, Cincinnati .NET User's Group
- Founder and President, Agile Conferences, Inc
- Contributing Author –
  - www.nplus1.org
  - "C# 2010 All In One" (Wiley)

Summary

- Scrum is about
  - Setting attainable goals
  - Preventing death marches through time boxing
  - Transparency
- Scrum is not about
  - Coding standards, development techniques or processes
- Scrum is a framework that promotes interaction, communication, and teamwork

Can't we all just get along?

- Teams don't work in isolation
- Teams must interact with many other groups in the enterprise that
  - Typically are not agile and/or
  - Have no desire/ability to become agile

Making Scrum work

- Courtesy and Respect
- Don’t just assume they “don’t get it”
- Be “Agile” in interactions

- Disclaimer: Some of the concepts in the following slides are not traditional Scrum

Inter-Team Communication

- Host meetings with representatives from all affected teams on a regular schedule
  - Development team reports:
    - High level progress status
    - Reaffirms architecture
  - Other teams report:
    - Status of infrastructure required for release
    - Any changes to external requirements
- Meet more often as release gets closer

Release Planning

- Enterprise projects
  - Usually consist of multiple sprints
  - Require a great deal of coordination between teams
- Product Backlog must be
  - Complete* (still subject to change)
  - Prioritized (all items, not just top *n*)
- Time-box the release
  - Priorities and scope *will* change
  - Estimates will be wrong
- Involves Product Owner, Architect(s), Security, Infrastructure, QA, Someone who knows MS Project, etc

Users

- Most users/customers don't understand software development
- Used to waiting months/years to see projects delivered
- Coaching is required
  - Product Owner is their single Point Of Contact
  - User Testing of Sprints is a new concept

User Testing

- User Testing is used to validate the state of the software after every sprint.
  - Key Users should be testing the codebase from the previous sprint
  - The Team (via the Product Owner) must fully disclose what they believe to be working and not working
- Users can enter *potential* defects into the tracking system

QA/Testers

- Best if QA is part of The Team
  - Corporate Silos can prevent this
- QA/Testers are used to the waterfall approach
  - Development creates something, throws it over the wall
  - QA tests it, throws it back
  - Ad infinitum
- Must adopt a different approach to testing

Sprint QA Testing

- As soon as the Sprint Backlog is determined:
  - Begin creating Test Plans for items in the sprint
  - Create/Update Integration Test Plans for current and previous Sprints
- When developers believe they are “Done”
  - QA Reviews Unit Tests
    - Validate that they are testing the requirements
- Bottom line, QA should be *Pro*active, and not *Re*active

Bug Triage

- Bug triage meetings happen immediately after the Daily Standup
- Triage Team
  - Lead QA, Architect/Dev Lead, Product Owner
- Bugs are marked for either:
  - Sprint Backlog
  - Product Backlog
    - Bug
    - Change Request

Swim Lanes

- Instead of Burn Down Charts
- “Stolen” from Kanban
- Tasks/Features move from
  - In Queue
  - In Process
  - Ready for QA
  - Ready for UAT
  - Ready for Release

9122 Montgomery Rd • Suite 201 • Cincinnati, OH • 45242
Phone:513-619-6323 • Fax:513-791-5202

Refining Requirements

- A good requirement is one that you can wrap a test around
- All Backlog items need to be defined well enough that a:
  - Developer can understand and code the intent
  - QA Resource/Tester can validate the code
- Incomplete items are removed

Wireframes

- Used to visually layout the User Interface
- All proposed screens
- Important to not look “finished”
- Tools:
  - www.mockupscreens.com
  - www.balsamiq.com

User Stories

- User Stories
  - As an [X] I Want [Y] So That [Z][1]
    - X is a role
    - Y is a feature
    - Z is the benefit

[1] http://dannorth.net/introducing-bdd

- As an Account Manager, I want to be able to Edit a Customer's Address so that we can Effectively Communicate with them
- Includes success criteria

Success Criteria

- Must be testable
- Use Given/When/Then syntax
  - Given 2000 customers
  - When selecting one
  - Then the form should open in < 1 second

9122 Montgomery Rd • Suite 201 • Cincinnati, OH • 45242
Phone:513-619-6323 • Fax:513-791-5202

Context Specification[1]

- When Editing a Customers Address
  - It Should Load in < 1 sec with 2000 customer records
  - It Should allow an Account Manager to edit the address

[1] Behavior Driven Development (Code Magazine) – Scott Bellware

9122 Montgomery Rd • Suite 201 • Cincinnati, OH • 45242
Phone:513-619-6323 • Fax:513-791-5202

Defining “Done”

- All (Dev, Users, QA, etc) must agree on definition of Done
  - Developer
    - Unit Tests, Documentation, Code Reviews, etc
  - QA
    - Integration Testing, Black Box Testing, etc
  - Users
    - UAT
- Will be different based on the product
  - NASA vs XBOX

Test/Behavior Driven Development

- Development needs to be Test Driven
  - QA personnel need to understand what that means
- Successful T/BDD development teams build confidence in themselves and with others
  - QA shouldn't have to test that
    - Math.Add(2,3) returns 5
  - QA can focus on the bigger picture
    - Making sure the requirements are met
    - Integration Testing

Sprint 0

- Also referred to as the Foundational Sprint
- Occurs before full Team is formed
  - Product Owner, Application Architect
- Used for:
  - Configuration (e.g. Build Server, developer Virtuals)
  - Product Backlog creation
  - Acquiring Funding
  - Release/Hardware planning
  - Assembling the Development Team

Verification Sprint

- Occurs after code “chill”
- Used for:
  - Security audits
  - Performance/Load/UAT/Integration testing
  - Deployment documentation
- Team uses this time to work on:
  - Required documentation, improving Unit Tests, etc.
  - NOT refactoring application code

Contact Me

- phil@skimedic.com
- www.skimedic.com/blog
- www.twitter.com/skimedic