



Enterprise Agility: Being Agile in the Waterfall Sandbox

Philip Japikse

Phil.japikse@pinnsg.com

MVP, MCSD, MCDBA, CSM, CSP

Principal Consultant

Pinnacle Solutions Group

ST. LOUIS DAY OF DOT NET

WE WOULD LIKE TO THANK OUR SPONSORS

PLATINUM



GOLD



SILVER



BRONZE



ST. LOUIS DAY OF DOT NET



Who am I?

- Principal Consultant,
 - Pinnacle Solutions Group, Inc.
- Microsoft MVP
- MCSD, MCDBA, CSM, CSP
- Enterprise Application Architect
- Trainer/Mentor/Speaker
- Director, Cincinnati .NET User's Group
- Contributing Author – www.nplus1.org

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.

<http://agilemanifesto.org>

Scrum Review

- Scrum is a framework for developing complex products and systems¹
- Scrum is based on:
 - Self Managed Teams
 - Iterative Development
 - Transparency

¹Ken Schwaber, “Agile Development With Scrum”

The People

- “Pigs” – Committed to the Project
 - Product Owner
 - Scrum Master
 - The Development Team
- “Chickens” – Interested in the Project
 - Users
 - Managers
 - Others

Scrum Master

- Ensures and Enforces Scrum
 - Values
 - Practices
 - Rules
- Not an “HR” position
 - Coach
 - Mentor

Product Owner

- Sole person responsible for the Product Backlog
 - Keeps the backlog up to date
 - Updates prioritization as necessary
- All project items and priorities funnel through the Product Owner

The Development Team

- Cross Functional
 - Coders
 - Testers
 - Designers
- Must adopt a “We” mentality
 - “Our” code, not “my” code
 - Considerate and Respectful
 - Located together in an open development space
- Sink or Swim, it’s a team effort

Product Backlog

- The Product Requirements
- Each Item consists of
 - Description
 - Priority
 - Estimate
- Dynamic, can be modified at any time
 - By the Product Owner
- Also includes triaged bugs

Sprint Backlog

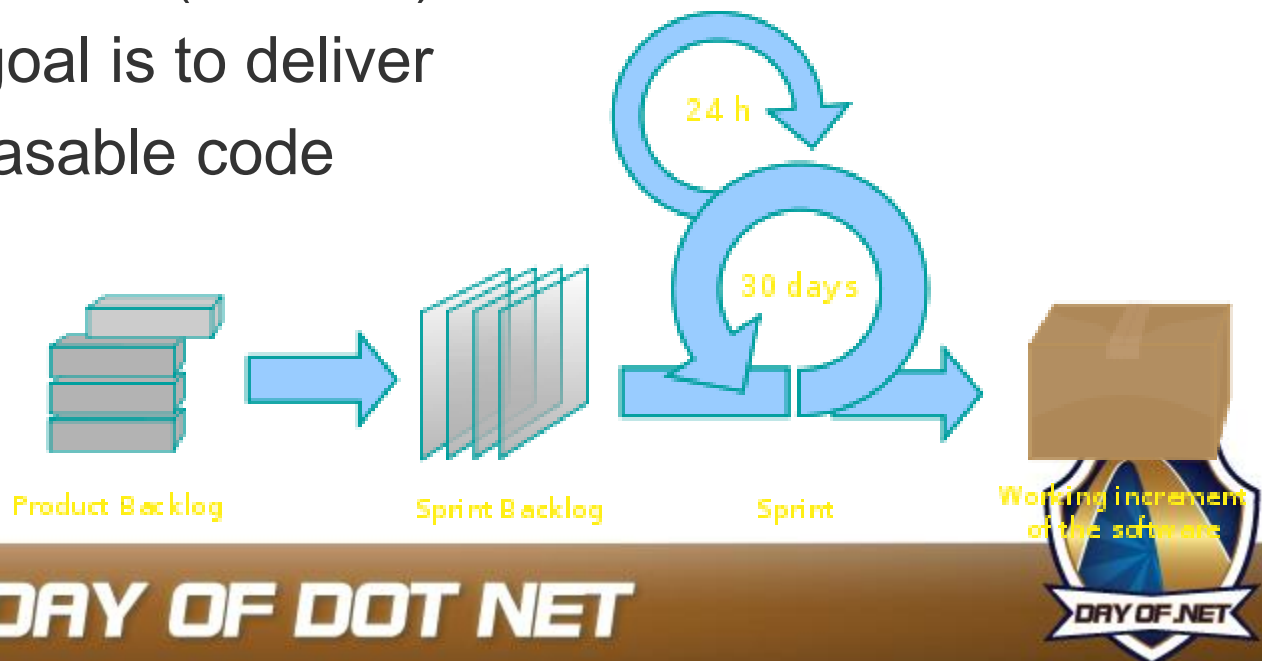
- Defines the work items for the sprint
- Controlled by the team
- Items from Product Backlog can be added

Burn Down charts

- Display of:
 - What's been accomplished
 - What is remaining
- Updated daily
- Transparent to all (Pigs AND Chickens)

Classic Scrum Lifecycle

- Sprint Planning Meeting (2-8 hours)
- Sprint (7-30 days)
- Daily Standup (15 minutes every day)
- Sprint Review (4 Hours)
- Sprint Retrospective (4 Hours)
- Each Sprint's goal is to deliver potentially releasable code



Sprint Planning

- Separated into two parts
 - Select Items to pull from the Product Backlog
 - Consider only prioritized and well defined items
 - How will it be developed
 - High level architecture
 - Get clarifications from Product Owner
- End result is Sprint Backlog

Iterations/Sprints

- Predetermined duration of work
 - Time boxed, not Scope boxed
- Length must stay consistent throughout the development life cycle
 - Personal recommendation is 30 days

Daily Standup

- Held every day. No EXEPTIONS!
- Only PIGS can speak
- Three Questions
 - What did you do yesterday
 - What are you going to do today
 - What's holding you back
- If meeting is lasting longer than 15 minutes, your talking too much!
 - Sidebars

Sprint Review

- Attended by Pigs and Chickens
- Team Demonstrates the Product
- Product Owner reviews:
 - What was accomplished
 - What (if anything) was deferred
 - What remains on the Product Backlog
 - Updated Release schedule

Sprint Retrospective

- Identify
 - Successes
 - Areas for Improvement
- Inspect Everything
 - People
 - Relationships
 - Processes
 - Tools
- Tackle the most relevant 1-2 items

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

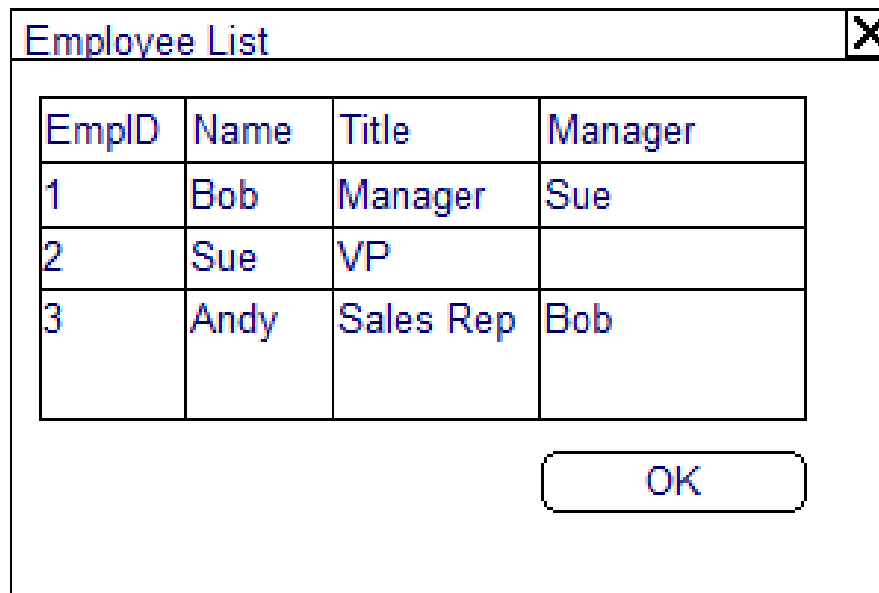
- Disclaimer: Some of the concepts in the following slides are not traditional Scrum

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”



A wireframe of a dialog box titled "Employee List" with a close button (X) in the top right corner. The dialog contains a table with four columns: "EmpID", "Name", "Title", and "Manager". The table has three rows of data. Below the table is an "OK" button.

EmpID	Name	Title	Manager
1	Bob	Manager	Sue
2	Sue	VP	
3	Andy	Sales Rep	Bob

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

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, etc

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

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

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 Driven Development

- Development needs to be Test Driven
 - QA personnel need to understand what that means
- Successful TDD 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

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 *Proactive*, and not *Reactive*

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

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

Questions?



Contact Me

- phil.japikse@pinnsng.com
- www.pinnsng.com
- www.japikse.blogspot.com
- www.twitter.com/skimedic
- (513) 619-6323