Getting Past the Documentation Conundrum

We're going to try something called Agile Programming.

That means no more planning and no more documentation. Just start writing code and complaining.

I'm glad it has a name. That was your training.

AGILE 2014 ORLANDO
THE DUDE IN FRONT OF YOU

• About the speaker
  – Matt Badgley, CSM, CSPO, SPC, & ABC
  – Agile and Product Consultant @ VersionOne
  – A career that is old enough to drink

• Contact Information
  • matt.badgley@versionone.com
  • @agilebaconbeer @versionone
  • http://www.linkedin.com/in/mattbadgley
POINTS OF DISCUSSION

Contention of Agile and Documentation
POINTS OF DISCUSSION

Tips For how Agile and Documentation can get along ...
POINTS OF DISCUSSION

Agile Requirements
EXERCISE - DOCUMENTATION YOU USE OR CREATE

• Write down the types of documentation that you use and or create
  – Use post-it notes and sharpies – use one post-it per documentation type

• When finished, bring up and place your documentation in the right project state column
That is, while there is value in the items on the right, we value the items on the left more.

– Manifesto for Agile Software Development
What does the Agile Manifesto and the general Agile Software Development Community say about documentation?
DOESN'T PRESCRIBE ANYTHING
JUST BARELY GOOD ENOUGH

Chart the Original Work of Scott Ambler © 2005
VALUE . . .

IDEA

DESIGN & CODE

TEST

VALUABLE, WORKING FEATURES

INVESTMENT (-$)

OPERATING EXPENSE (-$)

OPERATING EXPENSE (-$)

THROUGHPUT (+$)
FACE-TO-FACE
6. The most efficient and effective method of conveying information to and within a
development team is face-to-face conversation.

– Manifesto for Agile Software Development
Why do We Document?
SHARE INFORMATION TO DELIVER
DOWNSTREAM USAGE
**Exercise - What is your Doc For?**

- Back up on our board, there are three new horizontal lines:
  - Top Lane = Compliance / Audit
  - Middle Lane = Share Information to Deliver
  - Bottom Lane = Down Stream Usage

- Sort the documentation in each column into their right horizontal lane
Tips for Agile Documentation

• Focus on The Truth
• Document Decisions
• Who needs to know?
• Use your writers appropriately
  – Coders code, Tech Writers write
• Favor Accuracy and Brevity
• Use Self-documenting Code
  – It’s the tests stupid
• Make Documentation Transparent
• Do the Minimum Required (a.k.a. JBGE)

Thanks to Jason Tee and Scott Ambler for this list
WHAT ABOUT COMPLIANCE & AUDIT

HYA

Have You Asked?
"Repent of your Traditional sins! Only through Agile you can save your projects!"
WHAT ARE AGILE REQUIREMENTS?

• JBG
• Multiple forms
  – User Story (XP, Kent Beck, Mike Cohn, etc.)
  – Ultra-light Use Case (Alistair Cockburn)
  – Story Mapping (Jeff Patton)
• Decomposed to the point to be “delivered” in an iteration
• Bill Wake’s INVEST
• Provide insight into the problem or need a customer needs – the value to deliver
• Tokens for conversations of the past of the future
HOW MUCH DETAIL IS REQUIRED?

It varies...
HOW TO JBGE YOUR REQUIREMENTS

• Five Why your documents

• Use a Wiki (with usage stats)

• Define “Ready”
AGILE REQUIREMENTS LIFE-CYCLE
EXAMPLES OF “READY”

• User Story defined
• User Story Acceptance Criteria defined
• User Story dependencies identified
• User Story sized by Delivery Team
• Scrum Team accepts User Experience artifacts
• Performance criteria identified, where appropriate
• Person who will accept the User Story is identified
• Team has a good idea what it will mean to Demo the User Story
**Life of a Story**

1. Start with a Title and add a concise description

   **New User Registration**
   As a general user, I need to be able to enter my email address, my first name, last name, and a password so that I can securely access the site.

2. Add other relevant notes, specifications, or sketches

   **New User Registration**
   
   | Email Address: | test@test.com |
   | Last Name:     | Smith         |
   | First Name:    | Jane          |
   | Password:      | **********    |
   
   Register  | Cancel

   Need to provide a password strength indicator.

3. Before building software, write the acceptance criteria (how do we know when we’re done?)

   **Scenario: Enter Valid Information**
   
   Given I have entered test@test.com into email field
   And I have entered Smith into last name field
   And I have entered Jane into first name field
   And I have entered goodpassword into password field
   When I press Register
   Then The result should be a confirmation screen
   And An email is sent to test@test.com
   And A user record is created

This is modeled after Jeff Patton’s, “Stories gain detail over time” slide.
TAGRI!
CUSTOMERS CAN'T FIGURE OUT OUR USER INTERFACE.

THEY SHOULD READ THE MANUAL.

OUR MANUAL IS MORE CONFUSING THAN OUR USER INTERFACE.

THEY CAN USE OUR ONLINE SUPPORT DATABASE.

THAT'S MORE CONFUSING THAN OUR MANUAL.

WE HAVE NO MONEY TO FIX ANY OF THAT.

IN SITUATIONS LIKE THIS, I LIKE TO GO TO MY SPECIAL PLACE.

SOMEDAY I HOPE TO HAVE A SPECIAL PLACE THAT'S BIG ENOUGH FOR MY ENTIRE BODY.

PROBLEM SOLVED.
THEY AREN'T GOING TO READ IT (TAGRI)

Examples of TAGRI
• Requirements never read by Development
• PowerPoint Architecture
• Documentation After The Fact (DATF)
• The Magical Process Doc
• Mongol the User Manual

How to prevent TAGRI
• Treat Documentation Like any other Backlog Item
• Create for clear audience
• Always question how it’s going to be used
• Long-life vs. Short-life
• Just Barely Good Enough
• Single information source
EXERCISE - RATIONALIZE YOUR DOCUMENTATION

• Working together – for each section
  – sort the pieces of documentation with the most used at the top of the section and least used at the bottom
  – Remove those documents that everyone agrees would be better handled via verbal communications
  – Flag those documents/diagrams that should be part of Ready criteria – simply write an “R” in the bottom corner
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Let me say thank you to . . .

Agile Alliance for the time and effort to put together such a great conference and giving me the honor to speak.

VersionOne for having a mission of helping teams envision and deliver great software, and letting me be a part of it.

YOU for taking the time to participate and listen. Without you we don’t get better.
CREDiTS AND RESOURCES

• Picture of Dogs Playing, Murphy and Winston Stuart
• Manifesto for Agile Software Development, http://agilemanifesto.org
• Jason Tee, Project Documentation and Agile Development, http://www.theserverside.com
• Scott Adams, Dilbert, http://www.dilbert.com/
• Bill Murray, Picture from the motion picture Meatballs
• Pictures of Mike Cohn, Kent Beck, Jeff Patton, Bill Wake, Alistair Cockburn, Ron Jeffries, Dean Leffingwell, and Scott Ambler taken from the net
• Timezone Map from http://timeanddate.com
• Definition of Done from GDSteam’s Flickr Stream, https://www.flickr.com/photos/gdsteam/14041010536/