

# BAs in the Agile Project Environment



**MAY 4, 2017**  
**SCOTT HINKSON**

**PMP, CERTIFIED SCRUM MASTER (CSM),  
PMI-AGILE CERTIFIED PRACTITIONER (ACP)**

# Summary

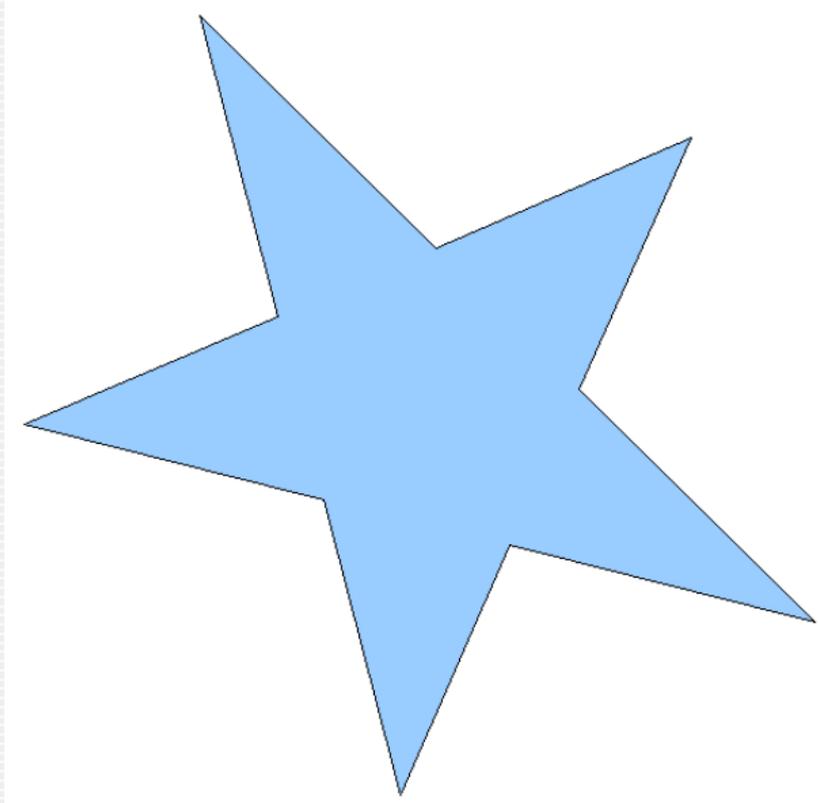


- #1 Roles and responsibilities of a BA in Agile
- #2 Discover a sampling of BA tools and techniques found in Agile
- #3 Discuss the tight coupling between the BA and Product Owner in Agile

# Requirements Gathering Problem



- How many points does this star have?

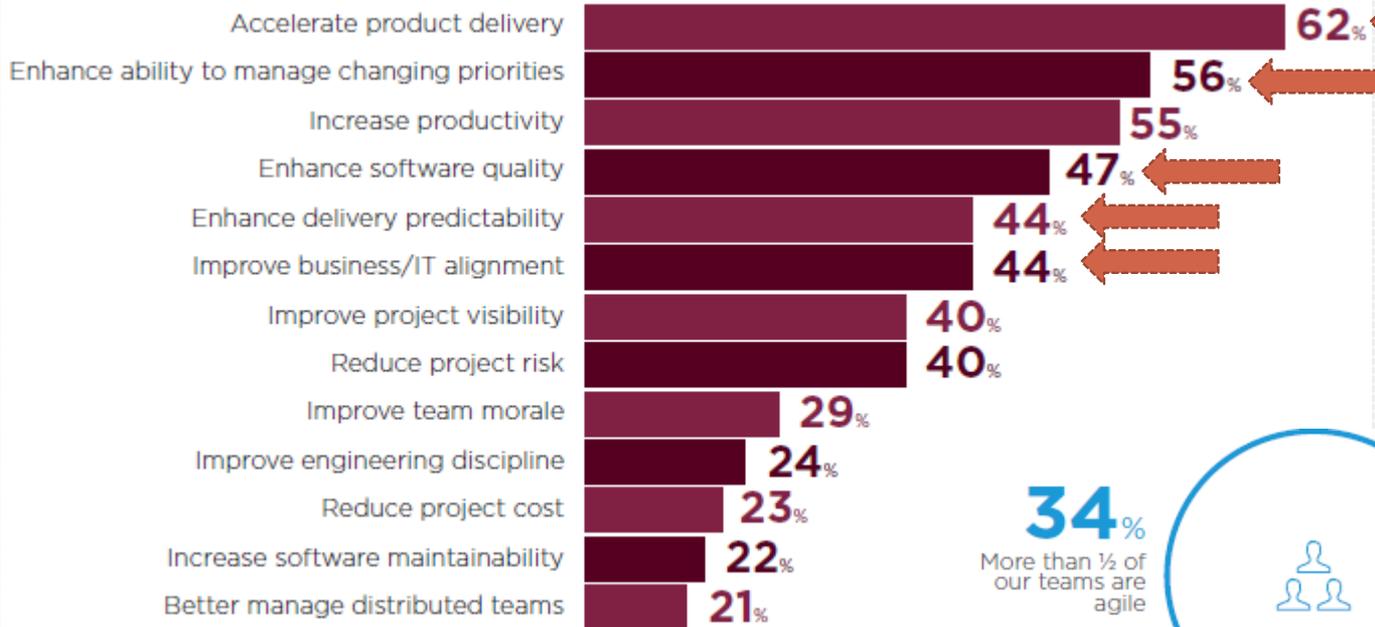


# What is Agile and Why Do We Use It?



- A **collection** of software development methods based on iterative development
  - Requirements and solutions evolve
  - Focus on delivering “value” sooner and with higher quality
  - Customer directly involve in defining and validating (only they understand what “Done” looks like)
- Most organizations—need:
  - Faster product delivery
  - Delivers a minimum viable product quickly
    - ✦ Fail faster – Low Risk, Lower Cost
  - Increased visibility of project team work – Allows for better project coordination
- Practical problems
  - Requirements risks continue to be the top challenge in any development effort
  - Traditional projects struggle with ambiguous goals, objectives, and requirements
    - ✦ How often does the customer really know and are able to communicate exactly what they want?
  - There is a focus on being able to adjust to change - ADAPTABILITY

# Agile Prevalence in Today's World

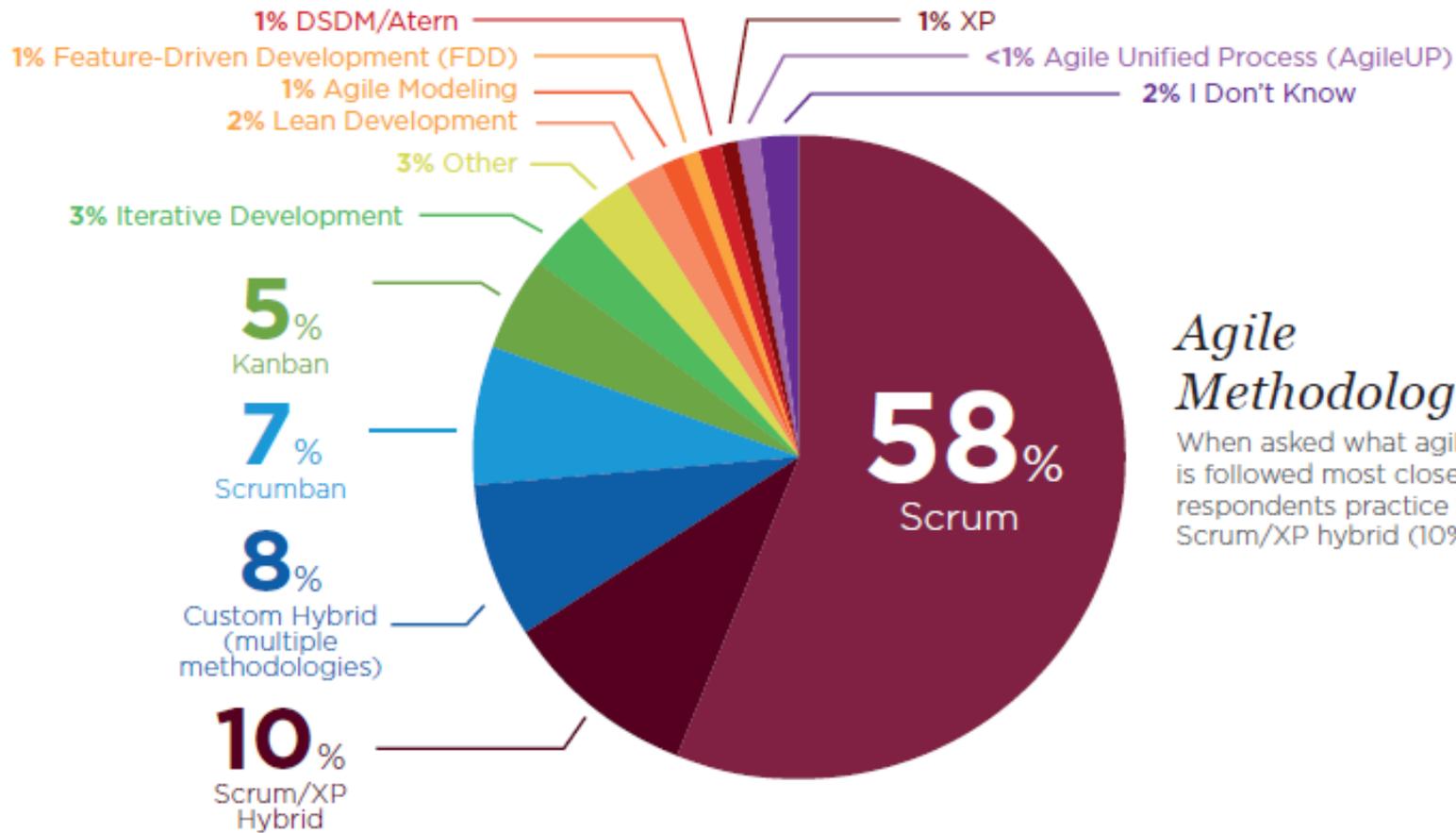


- Directly attributable and supported by to BA role and techniques on a project

- Out of 3,880 global respondents from organizations <1000 people to 20,000+ people, 43% of organizations have majority of teams as Agile



# Prevalence in Today's World



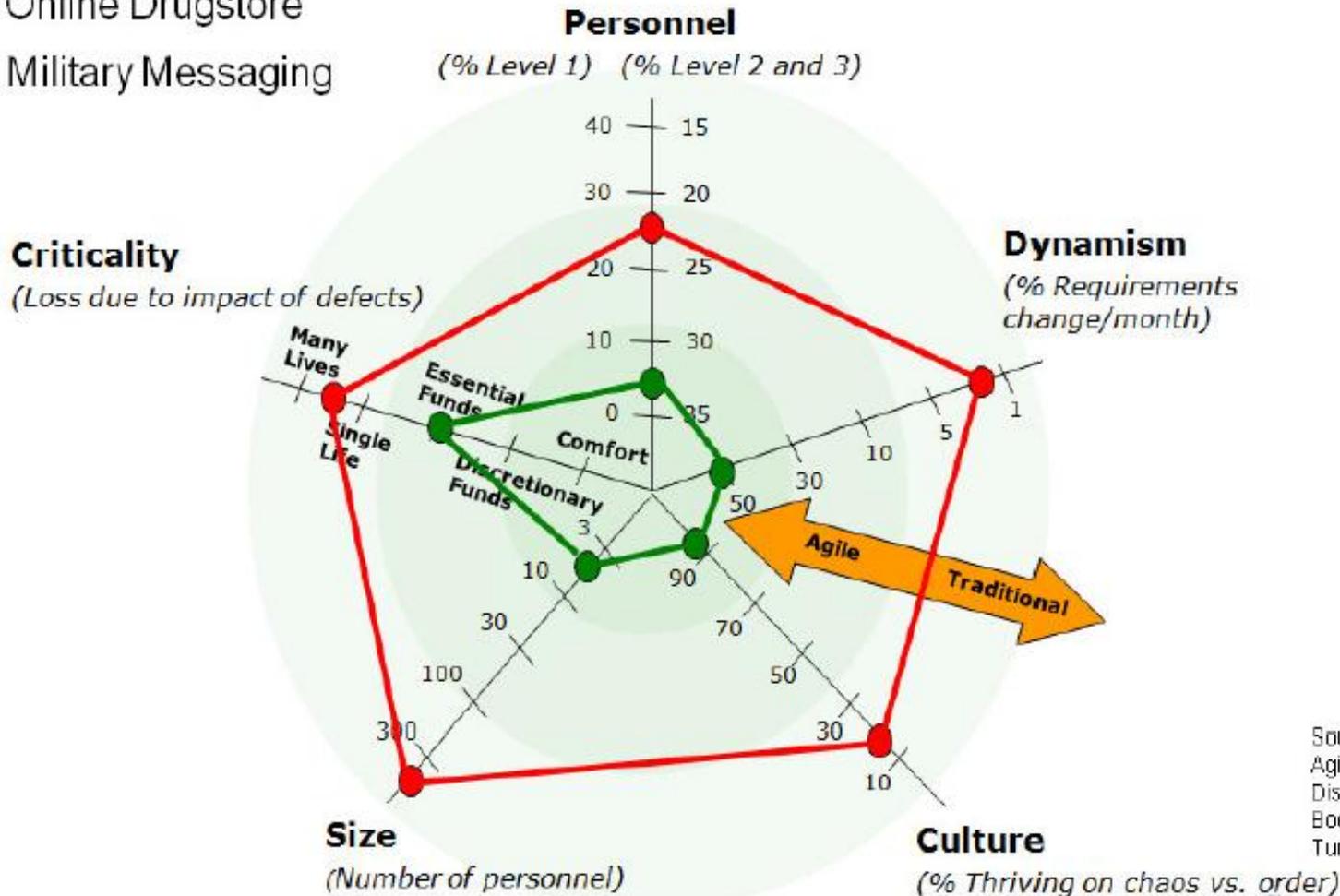
## *Agile Methodologies Used*

When asked what agile methodology is followed most closely, nearly 70% of respondents practice Scrum (58%) or Scrum/XP hybrid (10%).

# Is it Suitable? Try a Suitability Filter



- Online Drugstore
- Military Messaging



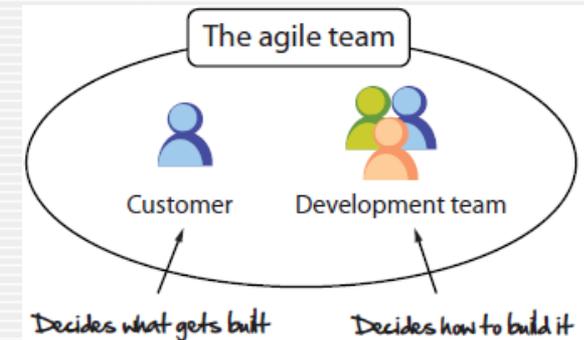
Source: "Balancing Agility with Discipline" Barry Boehm, Richard Turner

# Business Analyst Agile Roles & Responsibilities



# BAs in Agile: Pre-Project and During

- Many Agile advocates (usually in XP) believe the BAs are a role that can be dispensed with as the Customer can advocate their own requirements
  - Difficult to create synergies between different stakeholder groups that require one “Product Owner” voice
    - ✦ May have a process piece that needs analysis
    - ✦ Does the customer know how to author an implementable idea?
  - Little alignment to strategic goals, objectives, or business problem
    - ✦ BAs focus on the value proposition like a “business coach”
- During a project, BAs need to switch to a technology enablement role
  - Product Owner proxy (if required)
  - Story identification/prioritization
  - Story dependency mapping
  - User/persona identification and application
  - User Interface elaboration



# Relevant Components of the Agile Manifesto



- [www.agilemanifesto.org](http://www.agilemanifesto.org)
- Uncovering better ways of developing software by doing it and helping others do it
- 4 key 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

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

# Business Analyst Roles and Responsibilities in Agile



- In an Agile environment, the success of the business analyst relies increasingly on such interpersonal skills as communication, facilitation, coaching and negotiation
  - Still required to understand the broader business need and interpret in ways that make sense to team members focused on technology
- The ability to hold successful conversations replace the need for formal, detailed, long term artifacts such as requirement documents
- Analysts are active participants, if not the actual facilitators of planning, analyzing, testing, and demonstrating activities
  - Ensure product roadmap clearly defines the product's strategic alignment to the business need

# Agile Requirements Definition



- To effectively deal with scope on an Agile project, specifications must be considered in two dimensions:
  - Breadth first (use case diagram can be good)
  - Depth second
- Dealing with the breadth of the solution helps the team understand scope and cost and will facilitate estimating and release planning
  - Frames the boundaries of the project and helps to manage the organization's expectations.
  - Much smaller investment of time and resources than dealing with the entire depth
- The details are most likely to evolve as we progress through the project so defining them early has less value.

# BAs in Waterfall vs. Agile: A brief comparison



Traditional Projects	Agile Projects
Moderate focus on providing value at all steps	Increased focus on delivering value to the customer
<b>Timing of Activities:</b> Requirements Gathered/Documented Mostly at the Beginning of the Project	<b>Timing of Activities:</b> Requirements Gathered/Documented continuously throughout project
Focus on comprehensive detailed documentation and “sign off” sought. Examples are use cases, business and functional requirements, UI specs, etc.	Focus on the just the right amount just in time, “low-fidelity” documentation and requirements meet business needs through validation. Examples are User Stories.
Focus on completeness of the requirement in language	Focus on prototyping solutions quickly to test for <b>shared understanding</b>
Customer is not part of the project team	Customer is integral to the project team
Requires ability to comprehensively document all requirements at once for all development	Requirements ability to break problems into smaller chunks focusing on risk/ priority
Focus on what the requirement says	Focus on “fit for purpose/use” in the solution

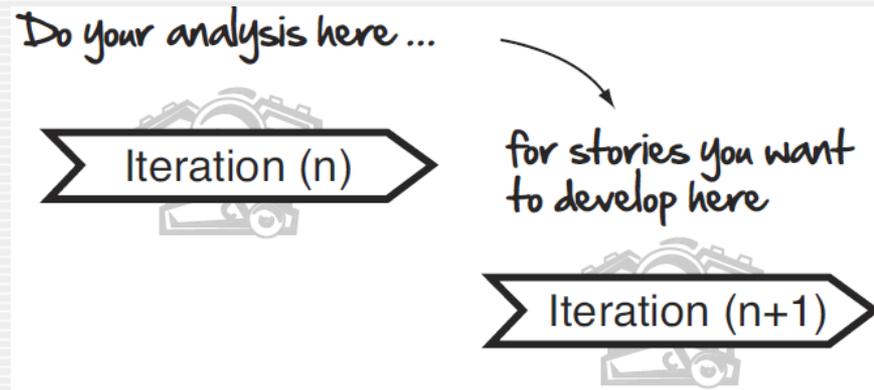
# Agile Business Analyst Tools and Techniques (a Sample)



# Common Principles to Agile Analysis



- There are two key concepts to agile analysis:
  - **Just-enough** - about doing whatever it takes to make the work ready—nothing more, nothing less
    - ✦ Small co-located team with customer onsite won't need a lot of formal documentation
    - ✦ Medium sized team, perhaps geographically dispersed, might need more
    - ✦ Large teams with distributed teams will need more
    - ✦ No one right level of detail for agile analysis. There is only what is right for you and your project
  - **Just-in time** - about doing the deep-dive analysis on your user story just before you need it (usually the iteration before)



# Backlog



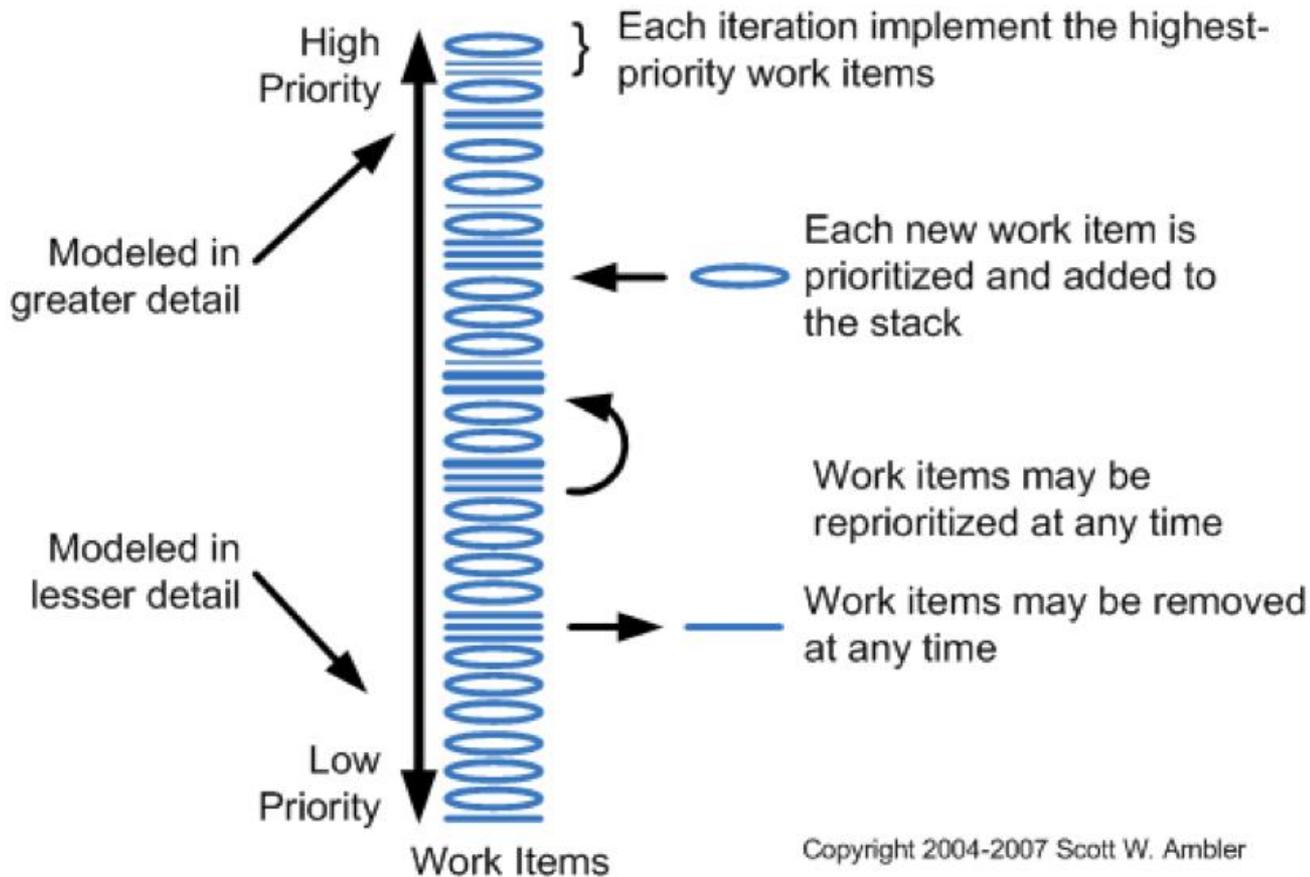
- A list of features or technical tasks which the team maintains and are known to be necessary and sufficient to complete a project or a release
  - If at any time a task or feature becomes known that is considered necessary to the project, it should be added to the backlog. Vice versa, items should be removed when no longer required.
- The backlog is the primary point of entry for knowledge about requirements
  - The single authoritative source defining project work to be done
- Typically the following is on a backlog (Product Backlog Items):
  - **Feature** (Epic, User Story) → representing functionality the customer desires
  - **Task** (something we need to perform to create the product or run the project)
  - **Defect** (mismatch from customer requirement)

# Backlog Management



- Also called “Pruning” or Grooming” the Product Backlog
  - The team (or part of the team including the product owner) meet regularly to "groom the product backlog", in a formal or informal meeting which can lead to any of the following:
    - ✦ removing user stories that no longer appear relevant
    - ✦ creating new user stories in response to newly discovered needs
    - ✦ re-assessing the relative priority of stories
    - ✦ assigning estimates to stories which have yet to receive one
    - ✦ correcting estimates in light of newly discovered information
    - ✦ splitting user stories which are high priority but too coarse grained to fit in an upcoming iteration

# Backlog Management



<http://www.agilemodeling.com/essays/amdd.htm>

- In Scrum, the top X amount of items would form the *Sprint Backlog* (aka *Iteration Backlog*) based and would be sized based on the team's velocity
  - How much work they can get done in an iteration measured typically in story points

# Kanban Boards



- Methodology for managing the flow of work to allow for change
  - Often the first “quick win” Agile tool to start with
- Key principles:
  - Visualize the work
  - Limit work in progress
  - Focus on flow
- Kanban (“signboard”) Characteristics
  - Does not require fixed iterations
  - Team only works on a fixed number of items at any one time → Called “Work In Progress” (WIP) limits. Recommended 2 to 3 items per person
  - Work may only begin on a new item when it is required to maintain flow downstream and the previous item is complete
  - Kanban board needs to be “groomed” similar to a backlog

# Kanban Boards



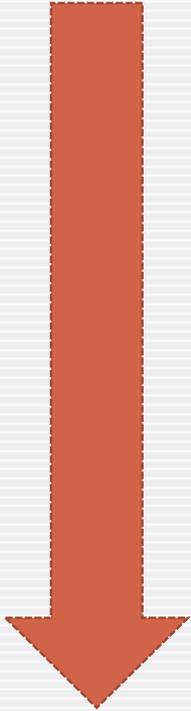
- Use different colour stickies for:
  - Features/stories
    - ✦ Include brief description, date started date finished, and priority
    - ✦ Must indicate who is working on it now
  - Defects
  - Tasks
- For all of the above, find a way to visually indicate (smaller sticky, dots):
  - Completed
  - Blocked



# Kanban Queues



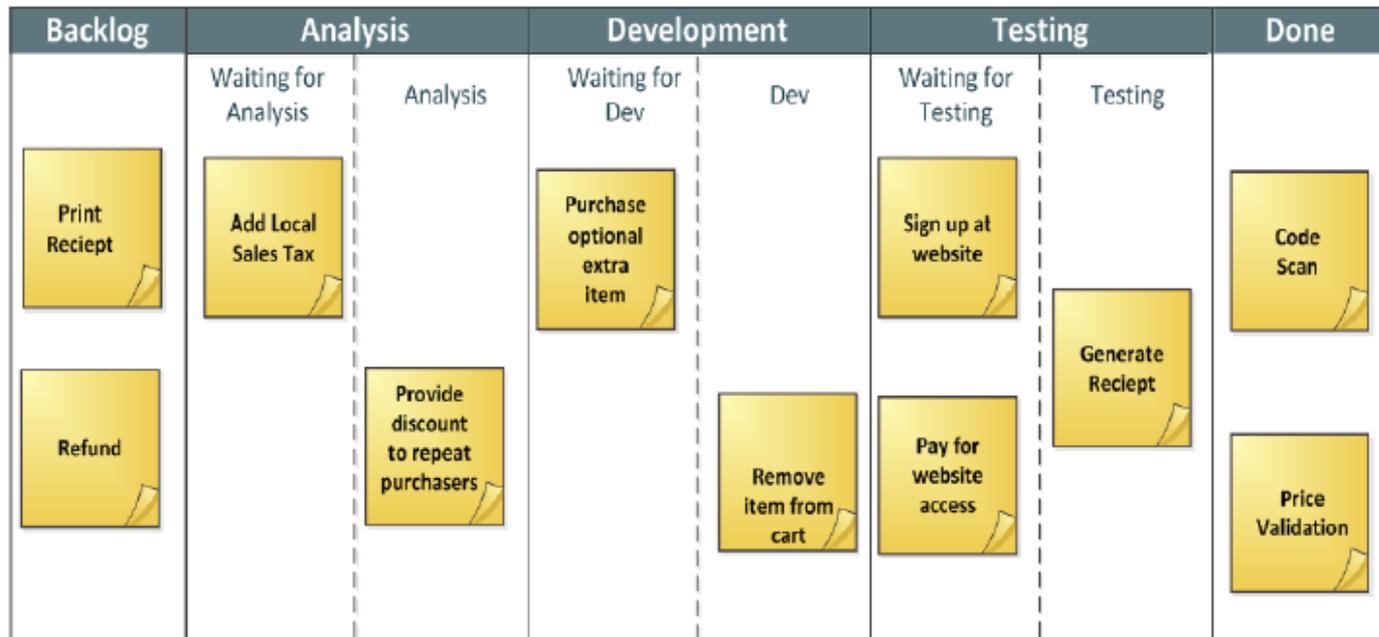
- Uses queues to control work in progress
  - Take from the previous queue and move a new queue without skipping steps. A **common** example is below:
    - ✦ **Product Backlog** (list of features, defects, or tasks)
    - ✦ Waiting for Analysis
    - ✦ **In Analysis**
    - ✦ Waiting for Development
    - ✦ **In Development**
    - ✦ Waiting for Testing
    - ✦ **In Testing**
    - ✦ Ready for Implementation
    - ✦ **Implemented**
    - ✦ **Done**



# Kanban Board



- Clear expectations of tasks by all team members due to visibility
- Easy to read/understand
  - Increased visibility to delays occurring
    - ✦ Team “swarms” to removes roadblocks or bottlenecks
  - Allows to identify “inefficiencies” with team members
    - ✦ Could result in clarification of expectations
    - ✦ Reassignment to different types of product work or roles/training/coaching needs





# Task Boards



- Can be for a whole project or per story (as shown below)
- Simplified workflow management by task

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 8 Code the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6
As a user, I... 5 points	Code the... 8 Code the... 4	Test the... 8 Code the... 6	Code the... DC 8	Test the... SC 8 Test the... SC 6

# User Stories



- A lightweight requirements documentation method for **shared understanding** vs. shared documentation
  - Three essential aspects of user stories called *card, conversation, and confirmation*
- **Card:** User stories should be written on small index cards. Write down only so much information, so nobody can mistake it for a complete requirement
  - Placeholder for discussion and a tool for planning
- **Conversation:** Is where we flush out all the details and get to real requirements
- **Confirmation:** Each user story should be verifiable, and we should have a clean way of deciding whether it was implemented correctly and completely

As a <type of user>  
I want <some goal>  
so that <some reason>.



**who** is this story for  
**what** they want to do  
**why** they want to do it

# User Stories – Card General Format



**TITLE** Automatic Updates **PRIORITY** P1

**PERSPECTIVE** AS A TRADER, I WANT TO BE NOTIFIED ABOUT PRICE CHANGES **REQUIREMENT**

**REASON** SO THAT I CAN TRADE PRO-ACTIVELY

MARK 7/10/2008 5

**AUTHOR** **CREATED ON** **ESTIMATE**

Short descriptions of a feature slice

- Usually written on small index cards to encourage the “conversation”
- Use index cards (small recipe cards) to remind teams that the initial goal of the requirements-gathering exercise isn't to get into all the details

- ✦ Write down a few key words to capture the spirit of the feature (breadth first, depth during conversation)

# User Stories - Conversation



- Why conversation? How do we trust the customer agrees to what they said if it wasn't formally documented?
  - The problem with gathering requirements as documentation isn't one of volume—it's one of communication
  - Written language is imperfect and subject to the decoder's interpretation

I didn't say she took the money.

I, didn't say she took the money.    I didn't say it.

I didn't **say** she took the money.    I said something else ...

I didn't say **she** took the money.    But someone else might have!

I didn't say she **took** the money.    She spent it instead.

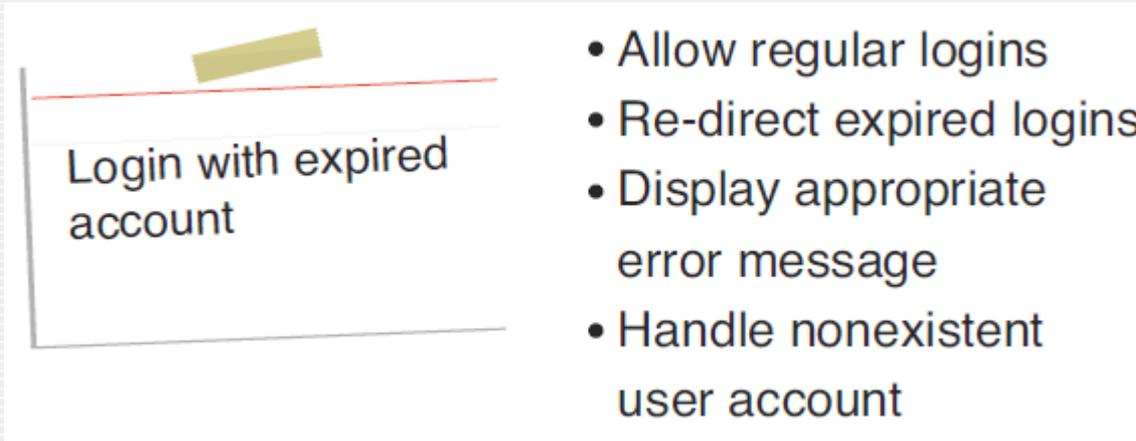
I didn't say she took **the money**.    Nope. Instead she stole my heart and left for San Francisco.

*Words are slippery things!*

# User Stories - Confirmation



- We want the aspects of our requirements to be measurable (testable) – **Acceptance Criteria**
- Need to articulate in small, verifiable chunks



The image shows a user story card with a yellow sticky note at the top. The card contains the text "Login with expired account" and a list of acceptance criteria.

Login with expired account

- Allow regular logins
- Re-direct expired logins
- Display appropriate error message
- Handle nonexistent user account

# User Stories – INVEST acronym



- **Independent**

- Avoid interdependent stories
  - ✦ Prioritization and priority problems

- **Negotiable**

- Details ironed out during conversations

- **Valuable**

- Must be valuable to the user
  - ✦ I.e. From “All error handling and logging is done through a set of common classes” to...
  - ✦ “All errors are presented to the user and logged in a consistent manner”

# User Stories – INVEST acronym



- **Estimatable**

- Must be able to determine how much time it takes
  - ✦ 3 reasons why you cannot
    - Developers lack domain knowledge
    - Developers lack technical knowledge
    - Story is too big

- **Small**

- Must be “right” sized
  - ✦ Team defines
- Epics fall into two types
  - ✦ Compound stories
  - ✦ Complex stories

- **Testable**

- Must be measurable

# User Story – Example when Complete



## Story name: Create work permit

### Description

Before contractors can legally work on the construction site, they need a work permit. This permit is what they will take to the job site when they are ready to begin construction.

### Tasks

1. Create permit page.
2. Save permit to database.
3. Add basic validation.
4. Ignore security (for now).

### Test criteria

1. Requestor can save basic permit.
2. Permit gets saved to the database.
3. Invalid permits are rejected.
4. Permit defaults to next week's start date.

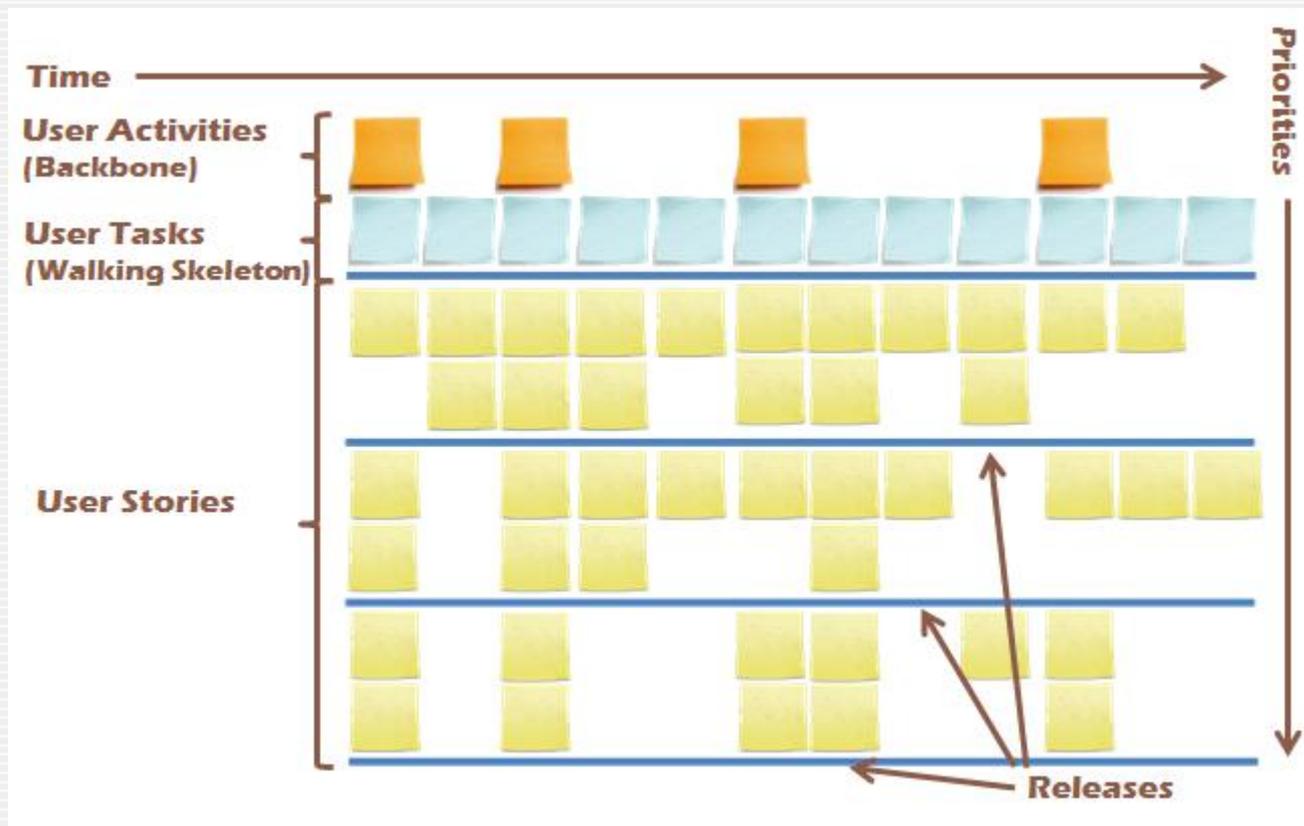
The screenshot shows a web browser window with the address bar containing 'http://workpermitsRUa/create'. The page title is 'Create Safety Work Permit'. The form includes a navigation menu with 'Create', 'Search', and 'Print'. Below the menu are dropdown menus for 'Area' and 'Site', and date pickers for 'From' and 'To'. There are input fields for 'Description of' and 'Equipment'. A 'Status' dropdown menu is set to 'Pending'. A large graphic of a yellow envelope with a red stamp and the text 'Construction Site' is positioned to the right of the input fields. At the bottom of the form are 'Save' and 'Cancel' buttons.

# Story Mapping

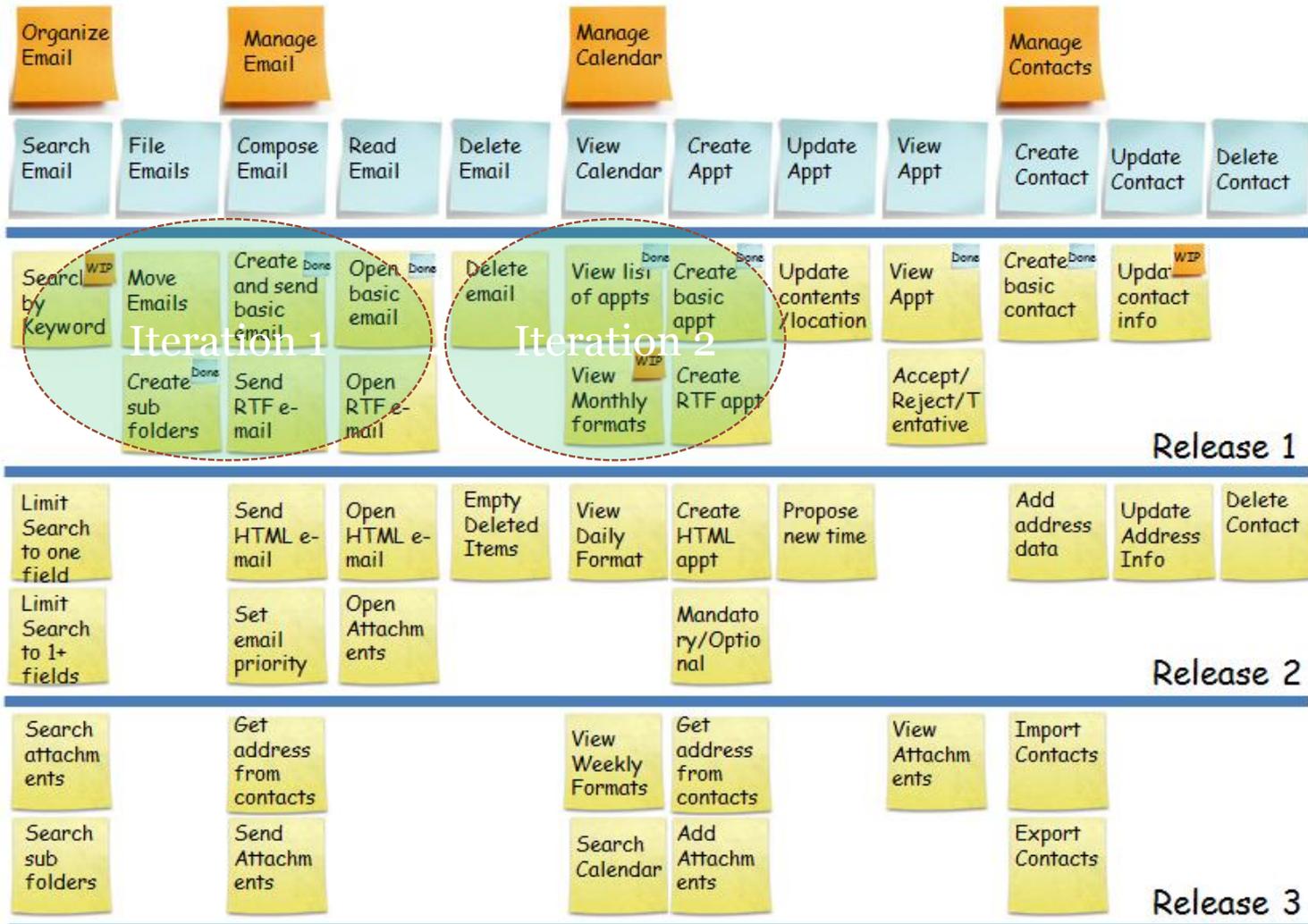


## ○ Story Mapping

- ✦ Graphical representation of stories along a time continuum used to identify risks and dependencies amongst and between the user stories to optimize the value delivered by each incremental story implementation

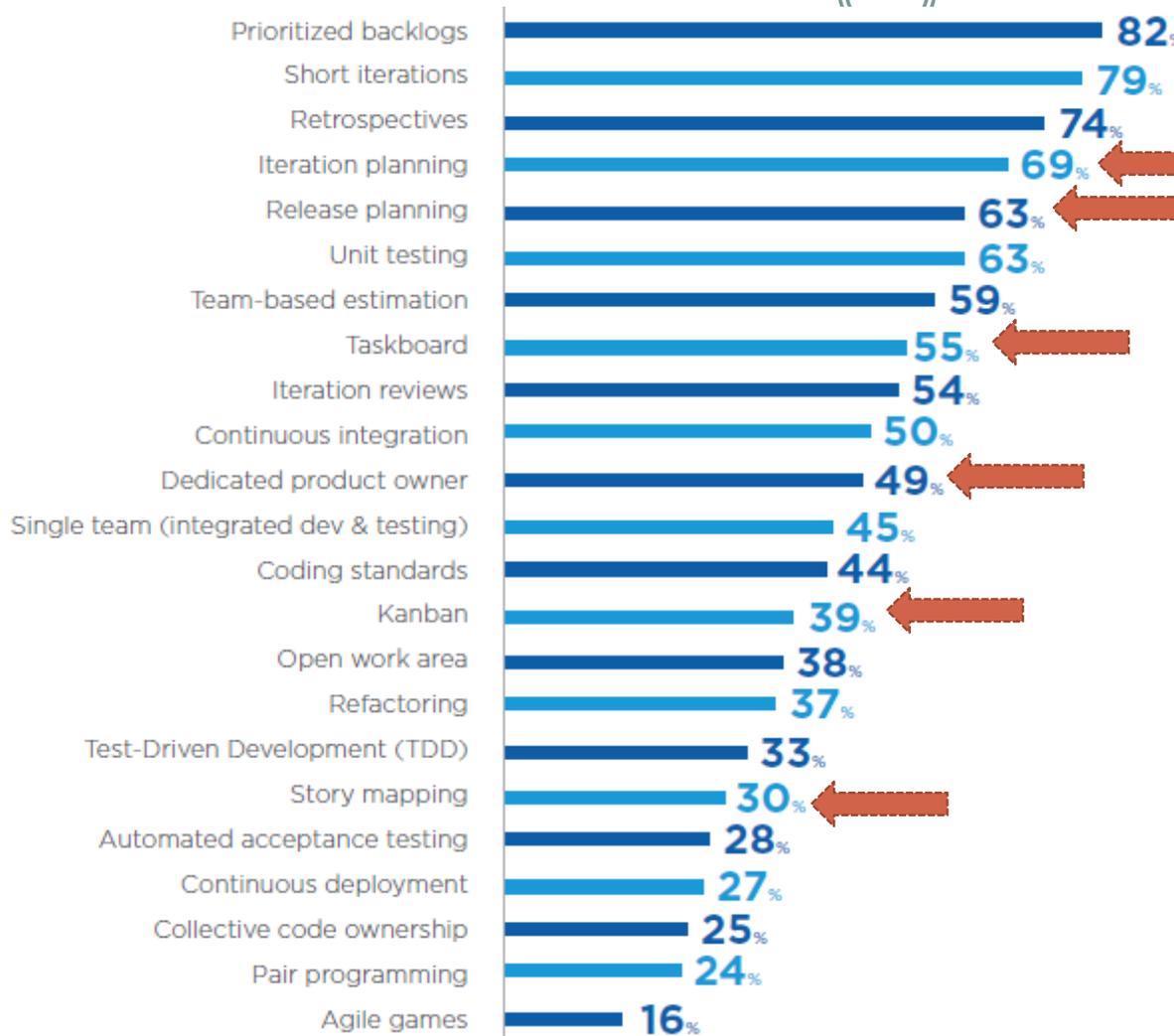


# Story Mapping



- Priority of features move left to right (time) and release management top down as they are decomposed into stories per feature
- Product roadmaps can be created using this way (project and post-project release management)

# Prevalence in Today's World – Tools and Techniques



- Arrows indicate those reviewed today in some detail (in addition to user stories)
- Many other tools and techniques (but 1 hour doesn't quite cover it 😊)

# BAs and the Coupling with the Product Owner



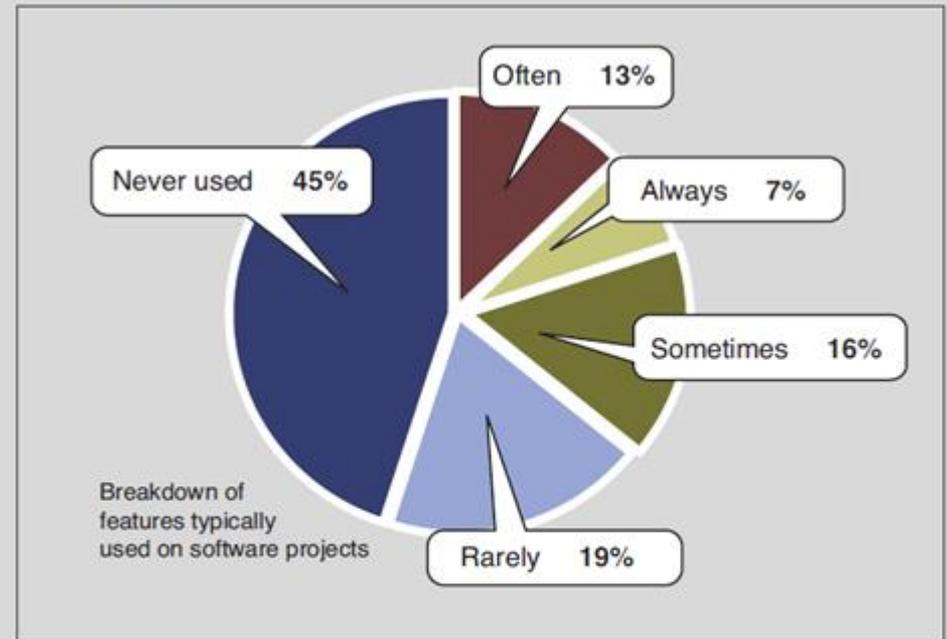
**LEARNING OUTCOME #3**

# Driving Value with our Customers

- BAs are instrumental in anchoring the Product Owner/ Customer's decisions during the project to value-added functionality

## The #1 Source of Waste on Projects

Did you know that 64 percent of features are seldom or never used? It's true!\*



Think about it. How much functionality do you use in Microsoft Word? 5 percent? 10 percent? Maybe 20 percent if you are a real power user?

By asking our customers to focus only on the really important stuff—and parking everything else—we can save them a lot of time and money while putting their software to work for them fast.

# Product Owner Communication Plan

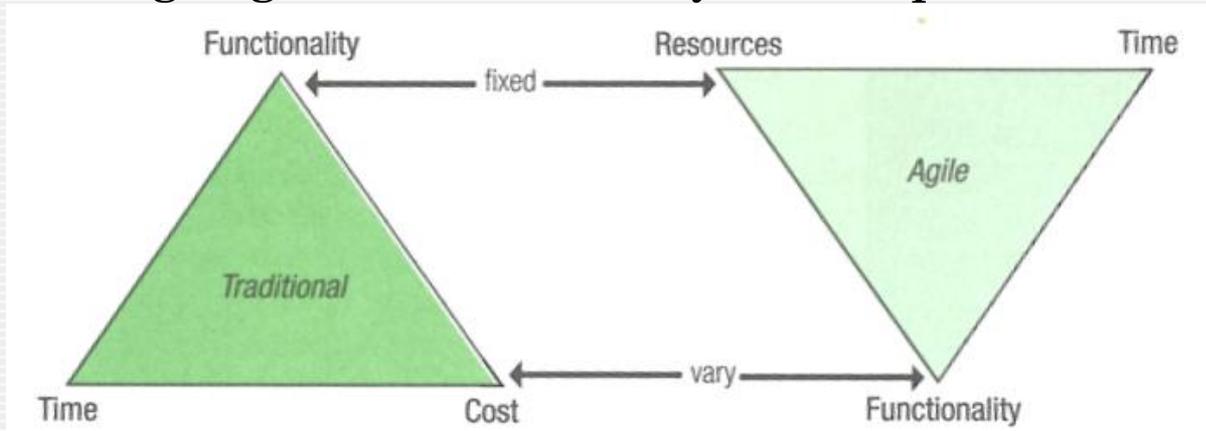


- Two constants on any agile project are setting expectations and getting feedback
  - Continuously setting expectations is necessary because things are always going to be changing
  - The habit of meeting regularly with your customer and reviewing the current state of your project
- Product Owners need to understand their involvement on any project, but especially Agile ones if it is new (assume it is)
  - How will their business needs be captured?
  - What's their time commitments and expectations of them?
  - When will the team involve them for project activities?
  - When can they get updates that they can take to their management?
  - Share understanding of what “done” looks like
  - Candidly discuss estimates and projections: Show true rate of development

# Organizational Readiness



- Culture is very important for Agile projects
- Will you have/is there buy-in the for the following?
  - Frequent product releases
    - ✦ How many, what frequency and what dependencies?
  - Co-location
  - Reduced documentation
  - Customer availability on the project
  - Low accuracy in estimates at the start of a project
- Make sure your customers know how agile planning works going in and where you are going to flex when reality and the plan start to differ



# Personas



- Quick guides or reminders of the key stakeholders on the project and their interests
  - May be based on profiles of real people or composites of multiple users
    - ✦ Provide an archetypal description of users
    - ✦ Be grounded in reality
    - ✦ Be goal-oriented, specific, and relevant
    - ✦ Be tangible and actionable
    - ✦ Generate focus
- They are fictional characters or archetypes that exemplify the way that typical users interact with a product
  - Should not be “generic” as they attempt to focus on the personas of the top customer archetypes
- Used to understand value from the perspective of a particular customer and allow a team that may not have direct access to a customer representative to better understand their needs
  - Personas are not "user roles" (such as sales person, administrator, etc.) primarily defined in terms of tasks or job descriptions
  - Personas put the emphasis on **goals and behaviours**

# Personas



- Should be described as though they are a real person
  - Provide a name, personality, family, work background, skill level, preferences, behavior patterns, and personal attitudes
  - Also a good practice to include a picture and write a short “day in the life”

<b>Name:</b> Bob the Movie Buff	
	<b>Values:</b> Bob would like to be able to order movies from the comfort of his home. He would like to be able to search for movies by title, actor, genre, and director. He would also be interested in knowing how other viewers rated the movie.
<b>Description:</b> Bob loves movies. On average, he rents 5 movies a week from his local rental store.  His two children also like to watch children's TV shows. They often like to watch the same shows more than once, which means that Bob sometimes has to pay late fees.  Bob's wife has different movie tastes than Bob and often spends a lot of time choosing a movie.	He is looking forward to unlimited movies so his children can watch shows multiple times without having to pay additional fees.  He would also appreciate a “recommended” feature to help him and his wife choose movies.