



The Agile Business Analyst

Hosted by CEO John Parker

October 16, 2014



John E. Parker



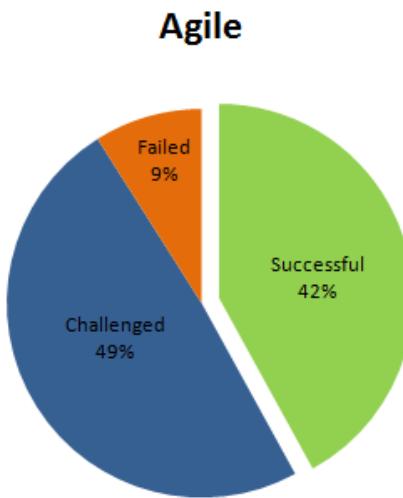
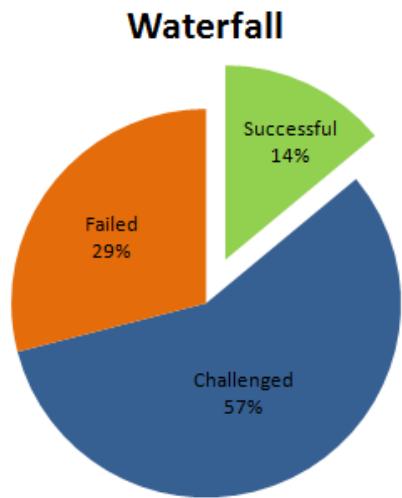
- Chief Executive Officer of Enfocus Solutions Inc.
- Previous Positions
 - Chief Visionary Officer of Enfocus Solutions Inc
 - EVP and CTO, MAXIMUS Inc.
 - Outsourced CIO for HSHS (Large Healthcare System)
 - EVP and Cofounder, Spectrum Consulting Group
 - KPMG Partner
- Expertise
 - IT Strategic Planning
 - Agile Development using Scrum and Kanban
 - Agile Portfolio and Program Management
 - Business Analysis
 - Recovering Troubled and Challenged Projects
 - Enterprise and Business Architecture
 - Business Process Improvement, Reengineering, and Management

Contact:

- <http://enfocussolutions.com>
- info@enfocussolutions.com



Why Agile?



Agile Benefits

- Higher productivity and lower cost
- Improved employee engagement and job satisfaction
- Faster time to market
- Higher quality
- Improved stakeholder satisfaction

- Agile projects are successful three times more often than non-agile projects, according to the *2012 CHAOS Manifesto Report* from the Standish Group.
- The report goes so far as to say, “The agile process is the universal remedy for software development project failure. Software applications developed through the agile process have three times the success rate of the traditional waterfall method and a much lower percentage of time and cost overruns.”

Source 2012 Standish Group CHAOS Manifesto Report



Agile Business Analysis is Much More than Just Writing User Stories

- Defining the problem and the vision
- Defining measures of success (i.e., Outcomes)
- Deciding what needs to change to fix the problem (i.e., Impacts)
- Using models and other visualization methods to understand customer's needs:
 - Who are your customers?
 - How will your customer use your product?
 - What are your customers priorities?
- Engaging Stakeholders
- Breaking the problem into smaller chunks that can be delivered separately (i.e., Features)
- Managing value delivery using roadmaps by mapping Features to Program Increments (PI)
- Participating in Inspect and Adapt (I&A) activities



What do Agile BAs Do?

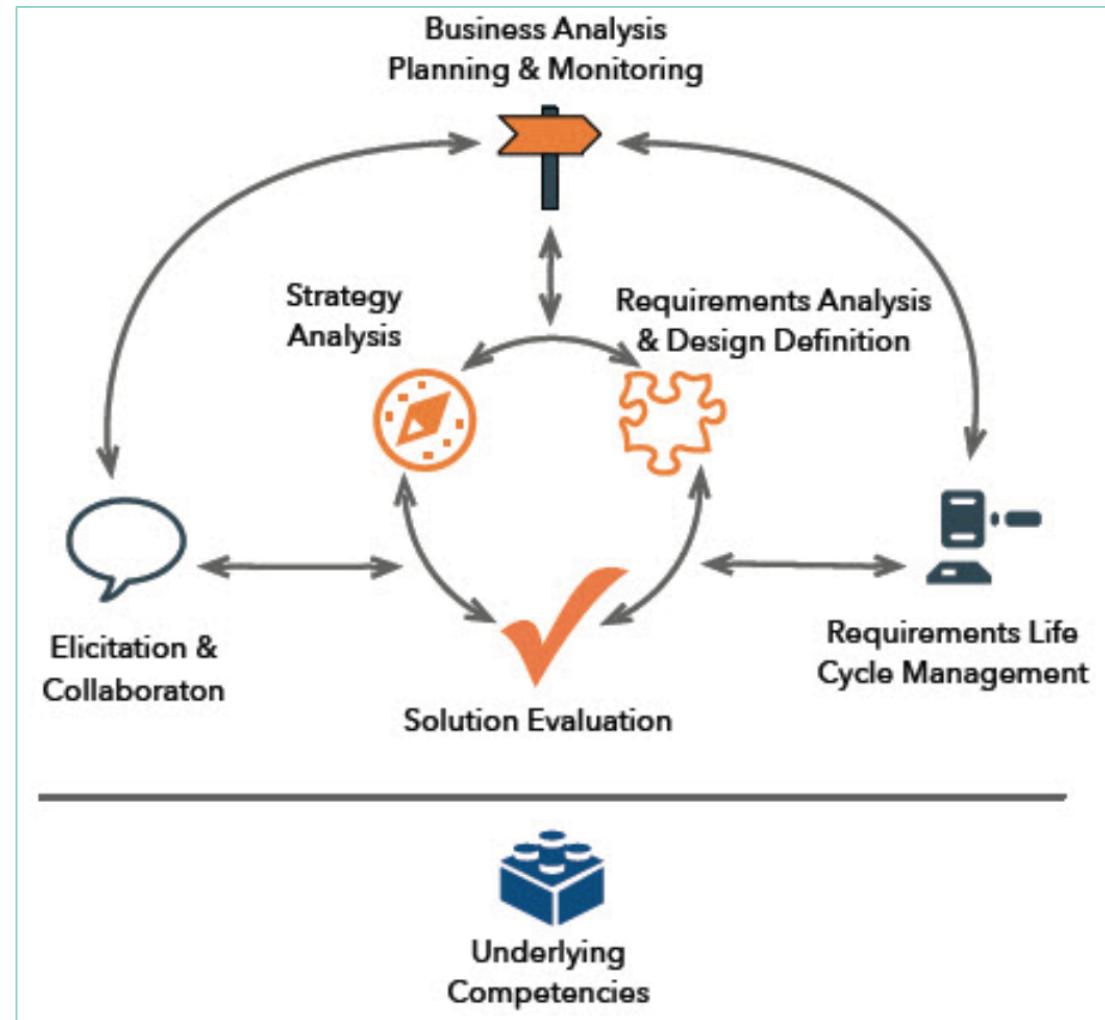
Activity	Description
Strategy Analysis	Document the problem or opportunity (Problem), its Context, and define the desired future state (Vision) using Outcomes.
Business Change Management	Develop a change strategy that will achieve the business outcomes and work with business and technical teams to make the necessary changes.
Stakeholder Engagement	Identify stakeholders and ensure that they actively participate in providing needed input to technical teams and making necessary business changes.
Discovery	Identify and document customer/stakeholder needs. Document assumptions, evaluate options, and validate that the recommended option addresses the customer/stakeholder need.
Solution Requirements	Work with technical teams to develop solutions that meet customer/stakeholder needs.
Business Value Delivery	Participate in Inspect/Adapt activities to ensure that the solution delivers business value and achieves business outcomes.



BABOK V3 Coming Soon

Changes from V2

- Several of the Knowledge Areas were changed.
- The biggest change in V3 is the new **Business Analyst Core Concept Model (BACCM)**, which defines six core concepts related to business analysis:
- Places greater emphasis on the business analysis work that happens prior to project initiation and after project closure.
- BAs do Design.
- Places much more emphasis on business change.
- New Perspectives Added
 - Agile
 - Business Intelligence
 - Information Technology
 - Business Architecture
 - Business Process Management



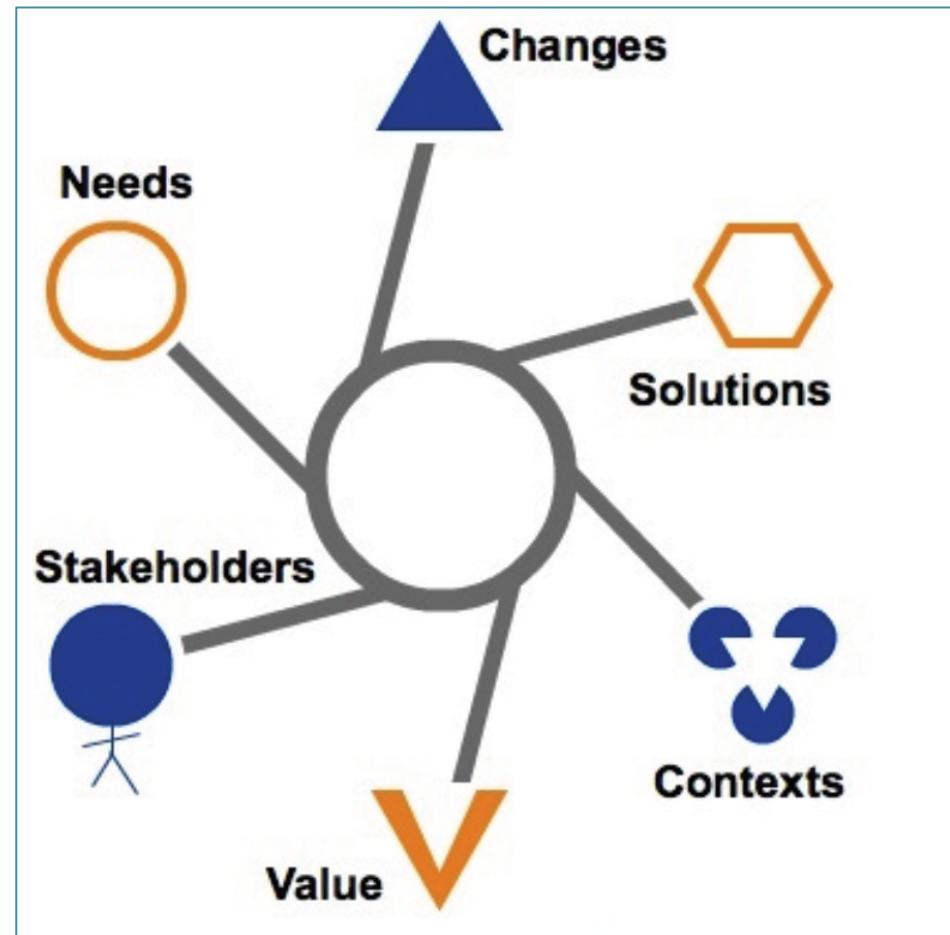
Source: IIBA



IIBA's Business Analysis Core Concept Model (BACCM)

BABOK V3 defines The Business Analysis Core Concept Model™ (BACCM™) as a conceptual framework for business analysis that encompasses what business analysis is and what it means to those performing business analysis tasks, regardless of perspective, industry, methodology, or level in the organization.

- What is the kind of change we're doing?
- What are the needs we're trying to satisfy?
- What are the solutions we're creating or changing?
- Who are the stakeholders involved?
- What do stakeholders consider to be of value?
- What is the context that we and the solution are in?





Agile Overview



Scrum

Great Project Management Method for Teams

3 Roles

- Team
- Product Owner
- Scrum Master

Ceremonies

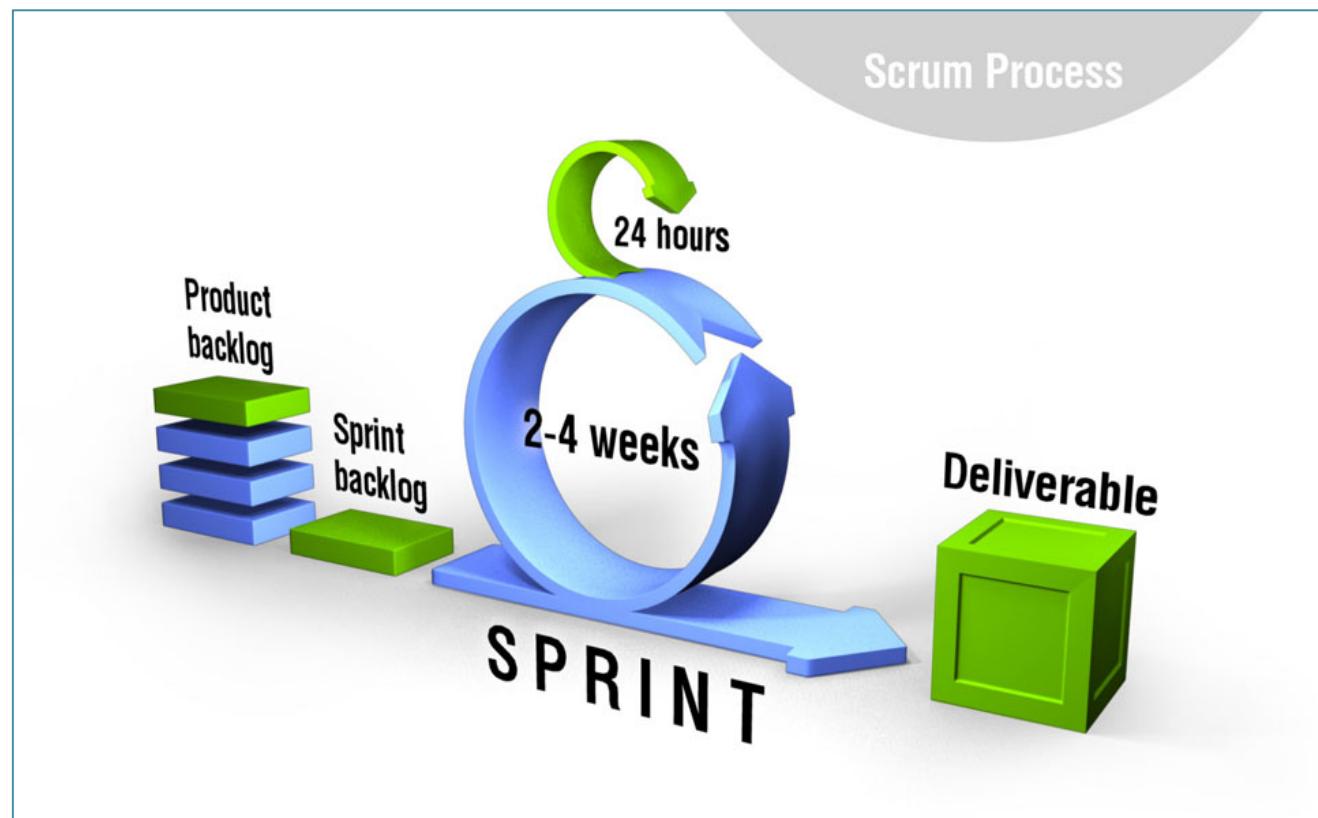
- Sprint Planning Meeting
- Daily Scrum Meeting
- Sprint Review Meeting
- Sprint Retrospective
- Backlog Grooming

Artifacts

- Product Backlog
- Sprint Backlog
- Burndown Chart

Best Practices

- User Stories
- Planning Poker
- Scrum Board





eXtreme Programming (XP) Produces Great Code

XP is a philosophy of software development based on well laid out values, principles, and practices.

Values

- Communications
- Simplicity
- Courage
- Feedback
- Respect

Principles

- Rapid Feedback
- Assume Simplicity
- Incremental Change
- Embracing Change
- Quality Work
- Teach Learning
- Small Initial Investment
- Play to Win
- Concrete Experiments
- Open, honest Communication
- Work with people's instincts - not against them
- Accept Responsibility
- Local Adaptation
- Travel Light
- Honest Measurement

Practices

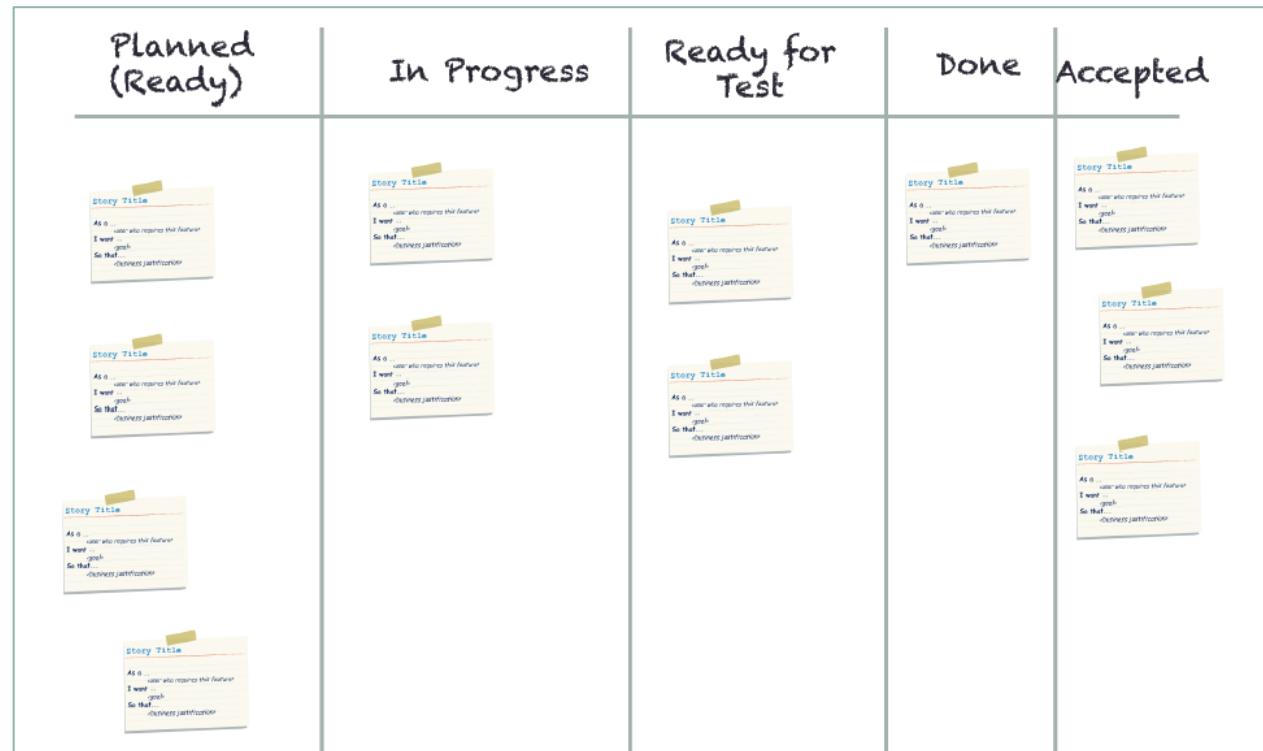
- The Planning Game
- Small Releases
- Simple Models
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40 Hour Work Week
- Onsite Customer
- Coding Standards



Kanban

Manages Demand, Controls Flow and Limits WIP

Using a Kanban approach in software drops time-boxed iterations in favor of focusing on continuous flow.



5 Key Principles

- Visualize the workflow
- Limit WIP
- Manage flow
- Make process policies explicit
- Improve Collaboratively



Scaling Agile

Team level agile practices are different than scaled agile practices

- The practices we put in place at the team level often don't work when we apply them in larger organizations.
- Practices have to be adapted to accommodate more diverse groups of stakeholders and more complicated value streams.

Scaled Agile requires a broader set of tools and techniques

- Use Scrum at the team level
- Use Kanban and Lean at the program and portfolio level
- Enhance discovery methods
- Enhance project management for integration among teams
- Engage stakeholders
- Use agile and lean methods for organizational change

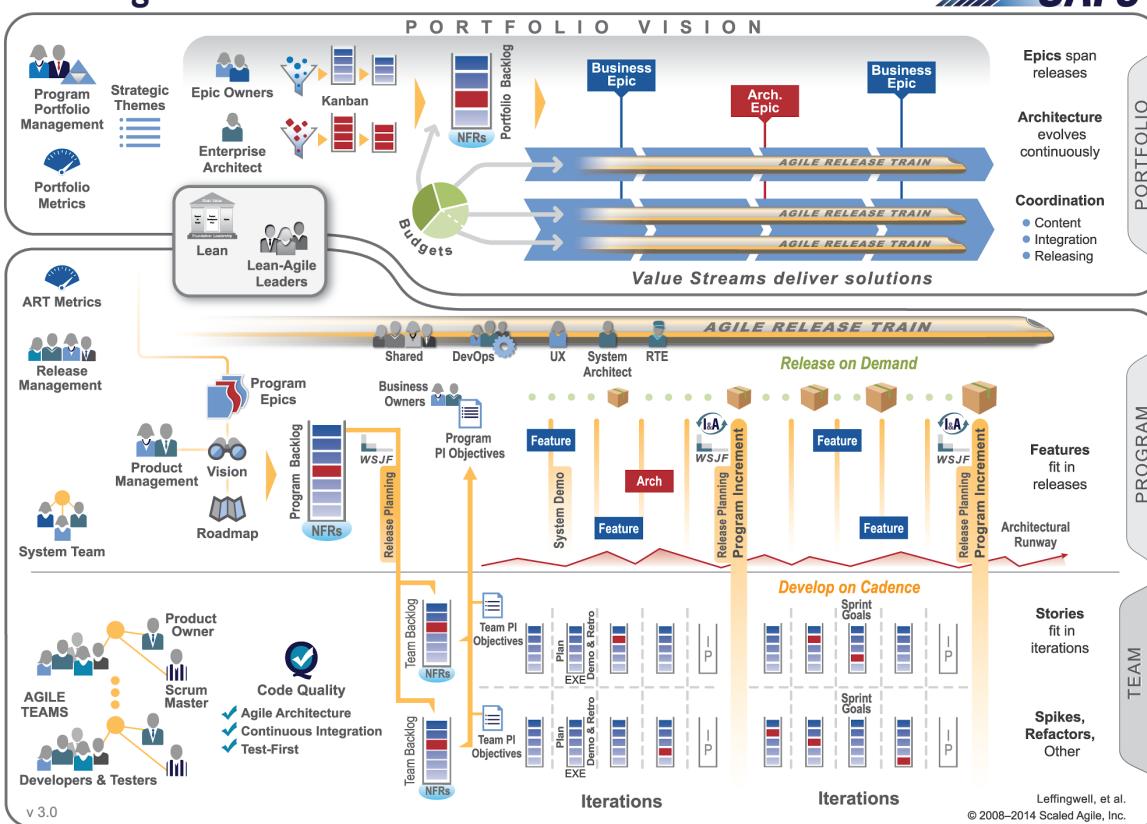


The Scaled Agile Framework®



“The Scaled Agile Framework is a proven, publicly-facing framework for applying Lean and Agile practices at enterprise scale”

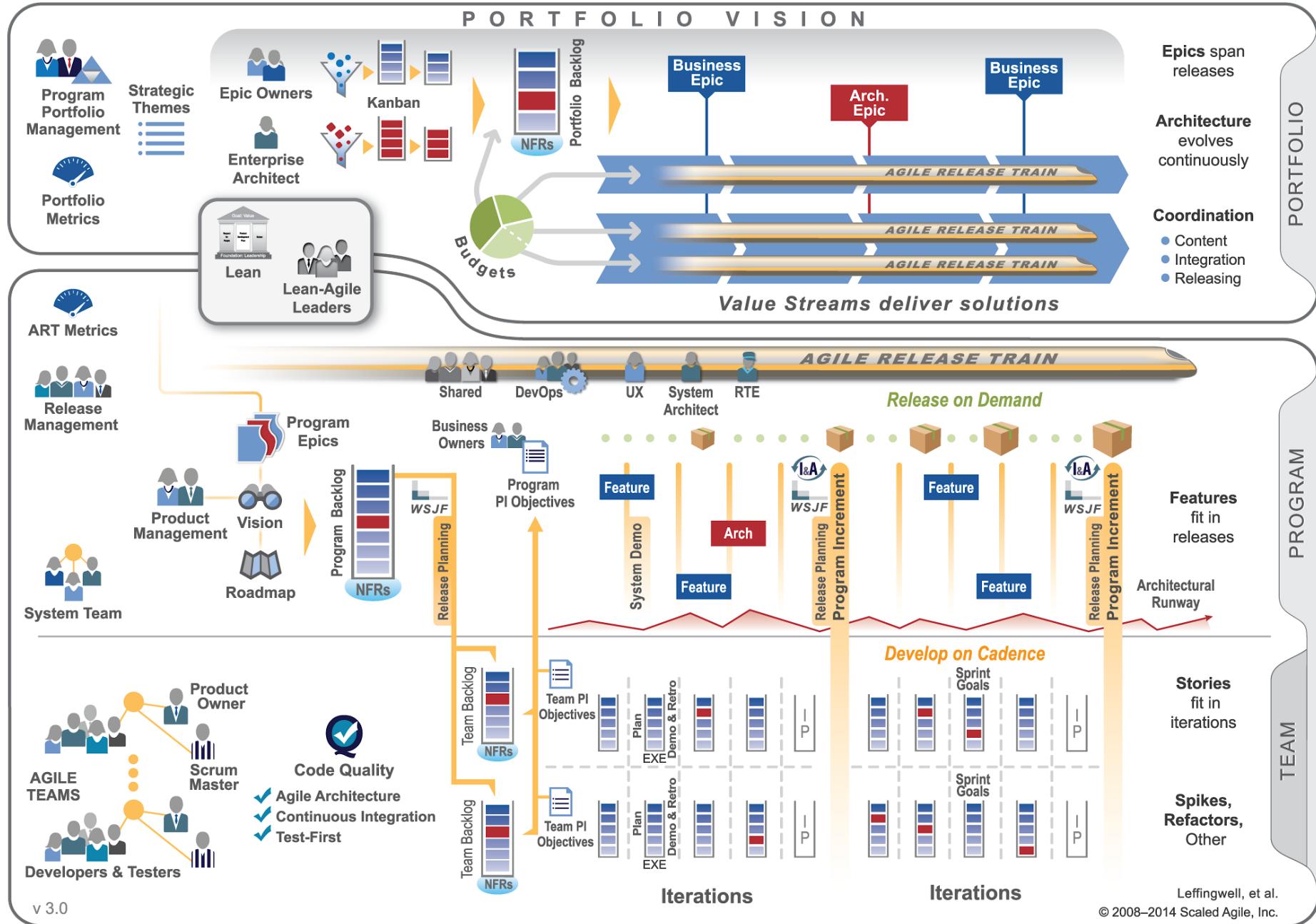
Scaled Agile Framework®



- The Scaled Agile Framework is well defined in books and on the web
- Works well with multiple teams in IT organizations of 50 or more people
- Has been scaled to hundreds of teams and thousands of people
- Synchronizes vision, planning, interdependencies, and delivery of many teams

Reproduced with permission from © 2011-2014 Scaled Agile, Inc.

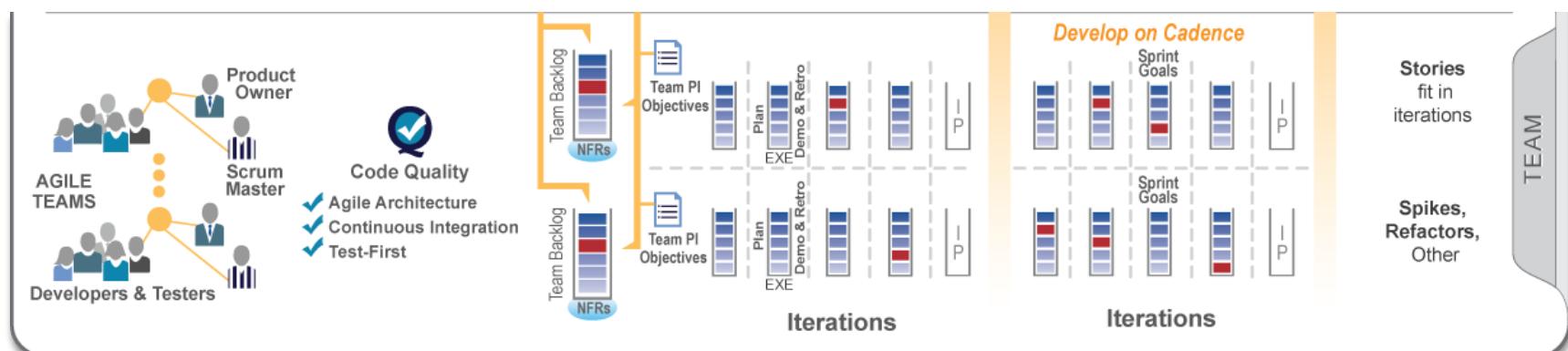
Scaled Agile Framework®





Nothing Beats an Agile Team

- ▶ Valuable, fully-tested software increments every two weeks
- ▶ Empowered, self-organizing, self-managing cross-functional teams
- ▶ Teams operate under program vision, architecture, and user experience guidance
- ▶ Scrum project management and XP-inspired technical practices
- ▶ Value delivery via **User Stories**

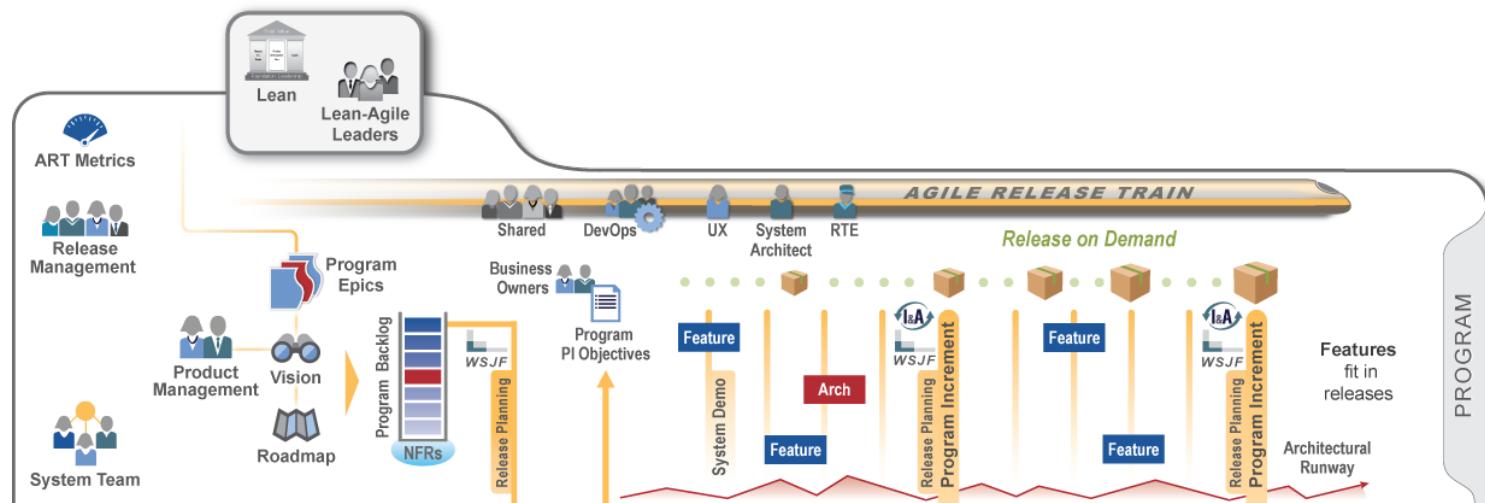


Reproduced with permission from © 2011-2014 Scaled Agile, Inc.



Scale to the Program Level

- ▶ Self-organizing, self-managing team-of-agile-teams
- ▶ Working, system increments every two weeks
- ▶ Aligned to a common mission via a single backlog

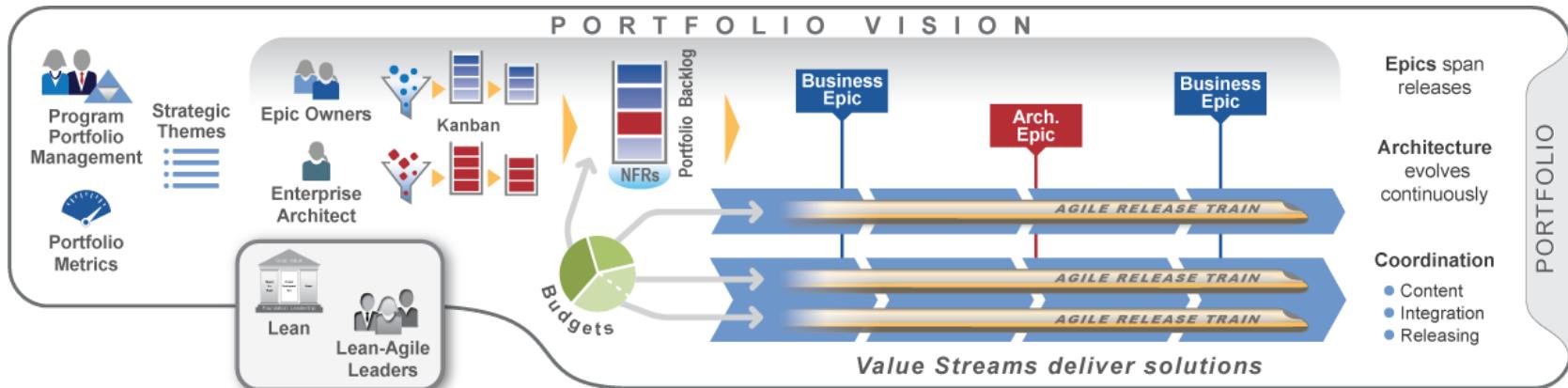


- ▶ Common sprint lengths and estimating
- ▶ Face-to-face release planning cadence for collaboration, alignment, synchronization, and assessment
- ▶ Value Delivery via **Features and Benefits**

Reproduced with permission from © 2011-2014 Scaled Agile, Inc.



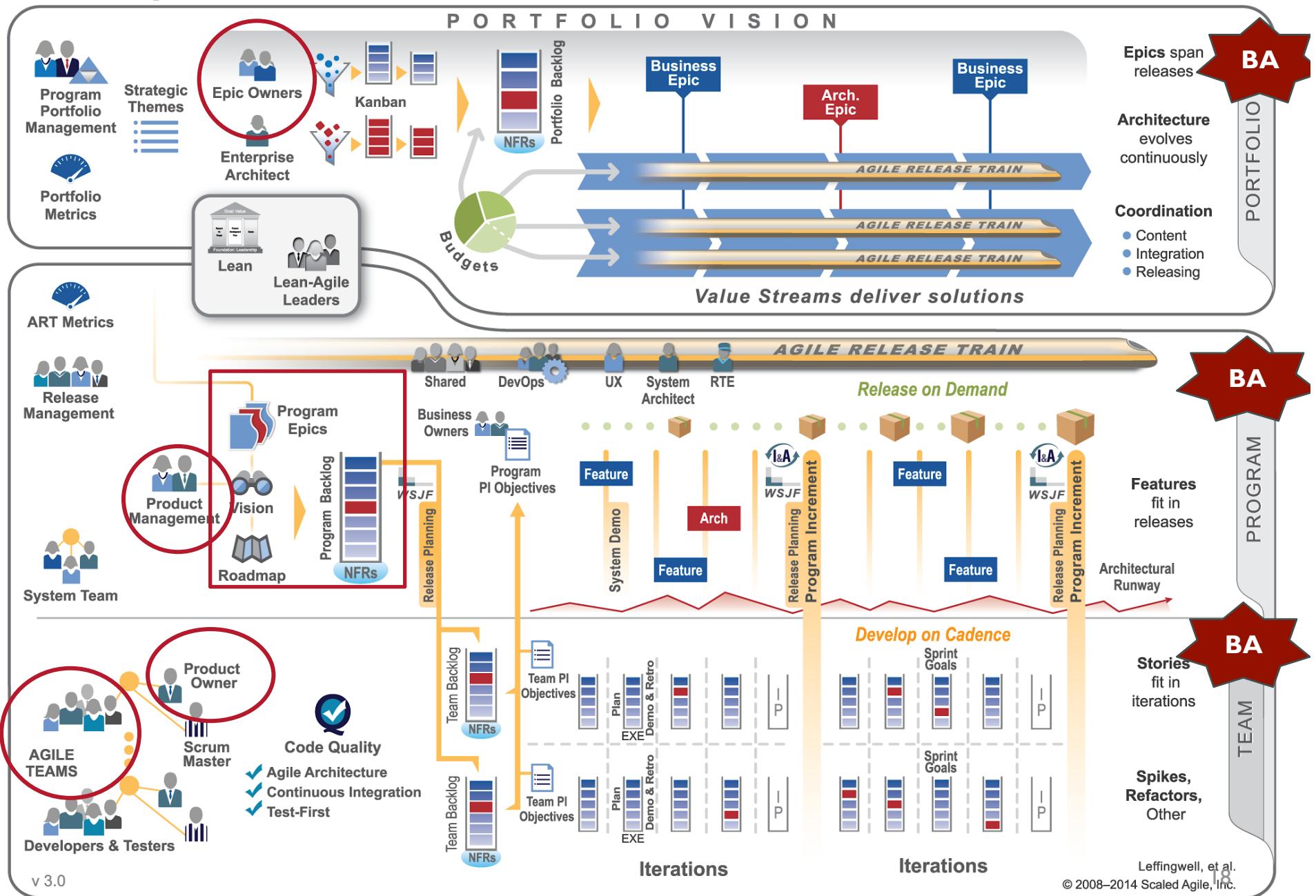
Scale to the Portfolio Level



- ▶ Centralized strategy, decentralized execution
- ▶ Lean-agile budgeting empowers decision makers
- ▶ Kanban systems provide portfolio visibility and WIP limits
- ▶ Enterprise architecture is a first class citizen
- ▶ Objective metrics support governance and kaizen
- ▶ Value description via ***Business*** and ***Architectural Epics***

Reproduced with permission from © 2011-2014 Scaled Agile, Inc.

Scaled Agile Framework®



Reproduced with permission from © 2011-2014 Scaled Agile, Inc.



BA Role on Agile Projects



Contrast in Styles

	Traditional BA	Agile BA
Focus	Requirements Documentation	Value Delivery in the form of Business Outcomes
Artifacts	Requirement Documents	Backlogs
Requirements Definition	All requirements defined upfront	Defined continuously in small batches (i.e., Features)
Requirements Writing	Written by BA	Written by anyone and collaboratively elaborated
Requirements Format	“Shall” Statements	User Stories
Negotiability	Non-negotiable	Negotiable
Sign-Offs	Formal Sign-Off obtained	Quick feedback
Requirement Changes	Strictly controlled after baseline	Changes are anticipated and welcome at any time
Location	Usually separate from the development team	Often collocated and part of the development team



Product Manager vs. Product Owner

Discovery

Product Manager

- External (Customer Facing)
- Tracks the Market and Competition
- Manages the long-term product roadmap and Program Roadmap
- Determines which Features are needed to meet customer and market demand
- Attends iteration demos and some daily scrum meetings
- Writes conditions of satisfaction for Features
- Demonstrates latest iterations to customers and gathers feedback

Delivery

Product Owner

- Internal (Team Facing)
- Prioritizes and maintains the product backlog
- Answers questions from developers and clarifies requirements
- Writes acceptance tests and criteria



BA Roles on Agile Project

Portfolio



Epic Analyst

Value Delivery

Program



Product Manager



Business
Change Manager

Customers' Needs
Business Change

Team



Product Owner



Team Member

Working Software



Business Analysis Responsibilities

Portfolio	Program	Team
Epics	Features	Stories
<i>Value Delivery</i>	<i>Customer Needs and Business Change</i>	<i>Working Software</i>
<ul style="list-style-type: none">• Prepare lightweight Business Cases• Document Business Objectives and Outcomes• Define Problem• Define Vision	<ul style="list-style-type: none">• Define Features• Document user personas• Discover user needs and activities• Design underlying business process• Manage business change activities• Define conditions of satisfaction	<ul style="list-style-type: none">• Develop user stories• Develop test cases and scenarios



Business Analyst as Epic Analyst



Epic Analyst

- Provides assistance to the **Epic Owner**
- Creates **Architectural or Business Epics**
- Work with Product Management to split into **Program Epics and Features**
- Coordinates epics related activities in sales, marketing, distribution
- Identifies what needs to change (*i.e.*, **Context**)
- Defines **Problem or Opportunity**
- Defines **Vision**
- Work with key stakeholders to document **Business Benefits** and rationale for change
- Prepares lightweight **Business Case**
- Defines **Objectives** and **Business Outcomes**
- Documents **Success Criteria**
- Monitors **Value Delivery**



Business Analyst as Product Manager

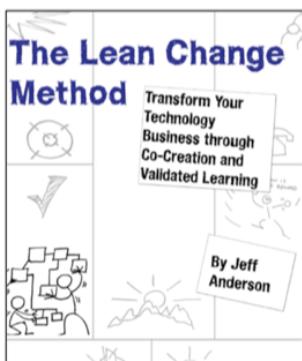


Product Manager

- Slices Epics into Features that can be developed and delivered in Program Increments
- Documents customer, market, and business needs
- Formulates hypothesis, assumptions and concepts focusing on what customers need
- Validates Features in the least costly way necessary viewing coding as the last resort
- Engages stakeholders in the discovery, delivery, and deployment processes
- Maintains roadmap showing when features are planned for delivery
- Works with the Product Owner to connect the development team to the ultimate sources of business needs (often the business executives, the customers, or the system users)
- Ensures that delivered Features address customer needs and achieve expected business outcomes



Business Analyst as Business Change Manager



- Identifies required business changes:
 - Stakeholders
 - Business Processes
 - Business Rules
 - Data
 - IT Services
- Develops overall change strategy using Lean Change canvas or similar method
- Negotiates planned changes in the form of Minimum Viable Changes (MVC) with business stakeholders
- Documents hypothesis and assumptions for planned changes
- Works with stakeholders, identify and conduct change experiments
- Engages stakeholders/customers in business change activities
- Maintains and tracks change activities using Kanban Board
- Coordinates business changes with software changes



Business Analyst as Product Owner



Product Owner

- Negotiates with the team what will be done in the next sprint in sprint planning meetings
- Provides guidance to team members on user story functionality
- Attends daily scrum meetings
- Works with Product Manager to realize the product vision
- Constantly re-prioritizes the Product Backlog, adjusting any long-term expectations such as release plans
- Acts as final arbiter of requirements questions
- Accepts or rejects each product increment
- Decides whether to ship
- Decides whether to continue development
- Considers stakeholder interests
- May contribute as a team member
- Has leadership role



Business Analyst as a Team Member



Team Member

- BAs works with the Product Owner and users to write **user stories**.
- BAs develop **acceptance tests** for user stories.
- The Business Analyst help ensure that there is **ongoing and meaningful two-way communication** between the Team and Users to help ensure the team is meeting the customer's needs.
- The Business Analyst **supports the ScrumMaster** as needed and help identify potential issues as they arise to ensure the team is working as efficiently as possible.



Agile Requirements



Five Types of Requirements

Requirement Type	Description
Business Requirements (Epics)	Describe the reasons why a project has been initiated, the objectives that the project will achieve, and the metrics that will be used to measure its success.
Stakeholder Requirements (Stories)	Describe how various stakeholders will interact with the solution and the needs they have in performing their assigned tasks and activities.
Functional Requirements (Conversations)	Capture and specify expected behavior of the system being developed. They define things such as system calculations, data manipulation and processing, user interface and interaction with the application, and other specific functionality that shows how user requirements are satisfied.
Non-Functional Requirements	Associated with the state of the system and not with the functionality that the system provides. General 'ilities' of the system, such as scalability, interoperability, maintainability, portability, performance, and security are different types of non-functional requirements.
Transition Requirements	Describe capabilities that the solution must have in order to facilitate transition from the current state to the desired future state. They typically address data conversion, skill gaps, and other related changes to reach the desired future state.



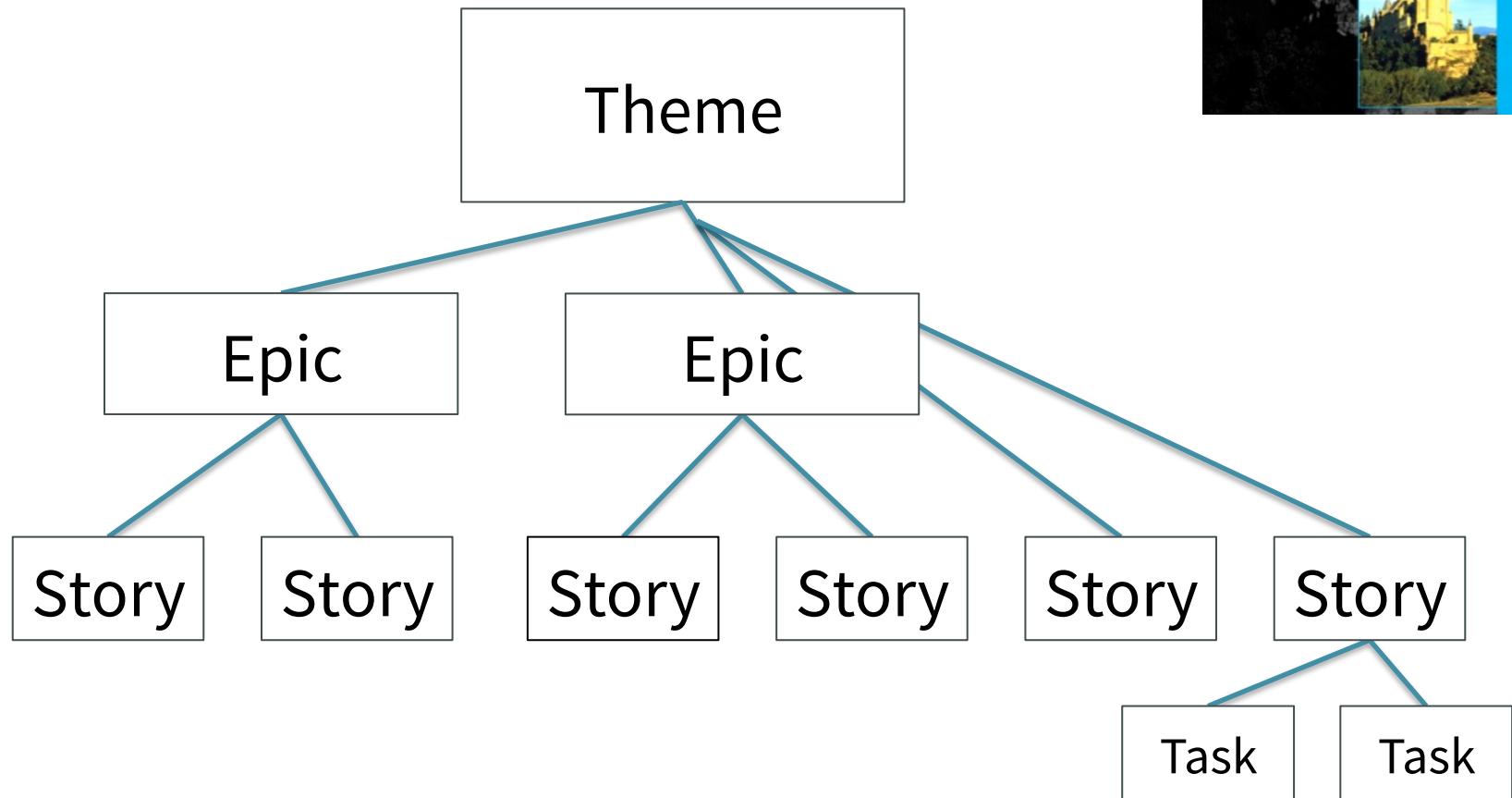
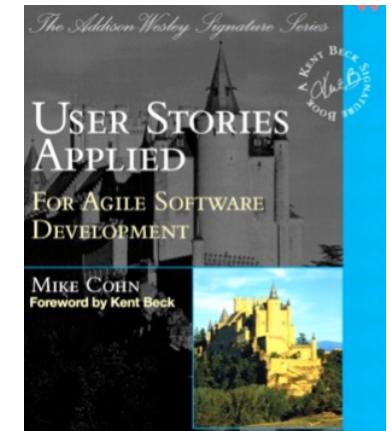
Difference in Agile Requirements



- Requirements are defined iteratively and incrementally
- Requirements are maintained in backlogs, not in paper requirement documents
- Requirements do not go through a formal sign-off process
- Solution requirements are defined via conversations between the team and users
- Business analysis work is done in increments rather than all up-front
- Business Analyst working on a team or serving as Product Owner are generally co-located with the team they support
- More emphasis is placed on verbal and visual communications versus paper documents
- Just enough, just-in-time
- Requirements are collaborative

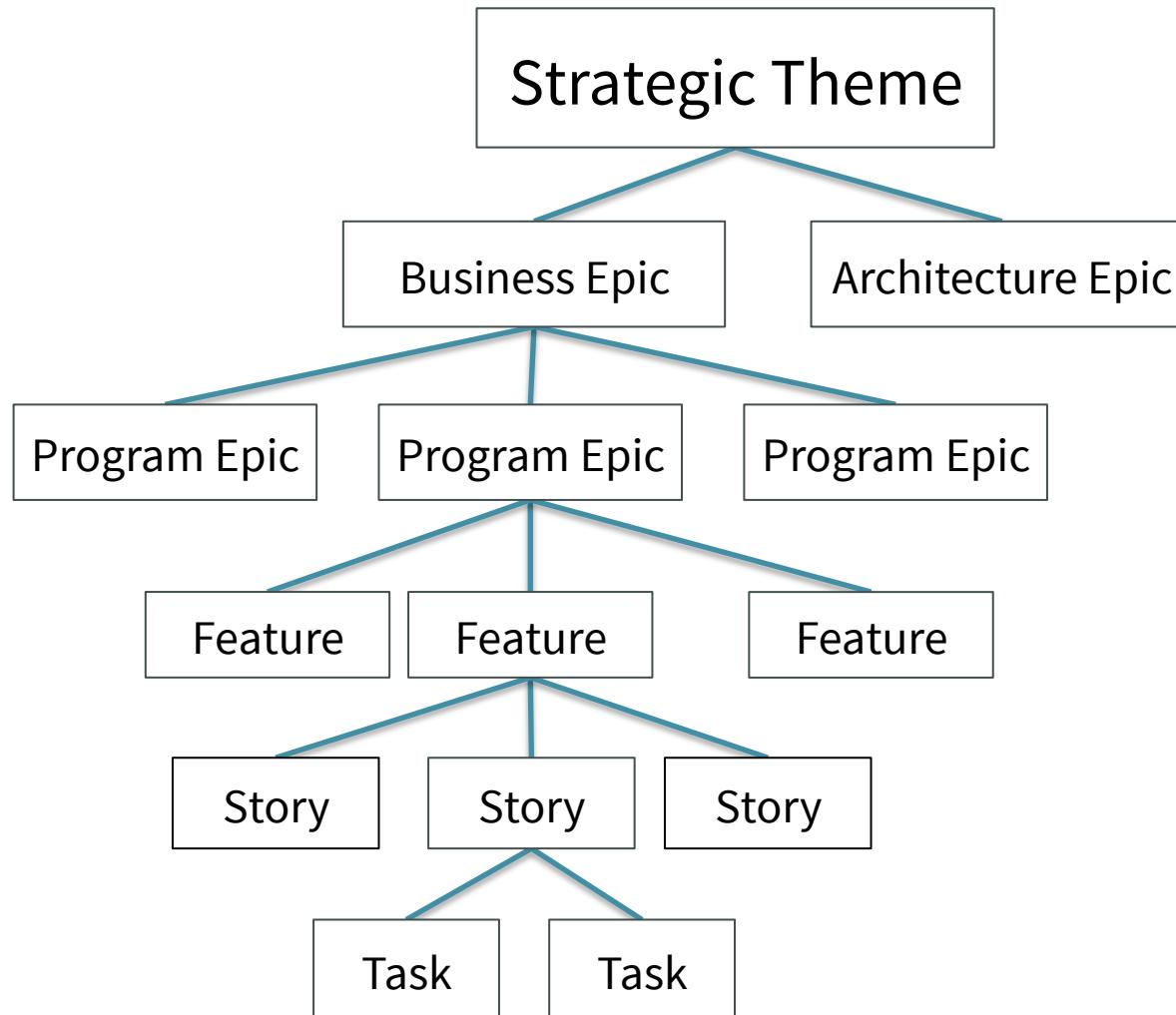


Agile Requirement Hierarchy, Mike Cohn





SAFe® Requirement Hierarchy





Agile Requirements



Epic

- Large initiatives delivering new products, solutions, or services to customers
- Comprised of a large collection of features

Feature

- Capabilities that the Product Owner is interested in
- Provides value to users
- Realized by some number of user stories

User Story

- Represents a user's need
- Planning item
- Causes a conversation to occur



Business and Architecture Epics

Epics document **Business Requirements** by describing the problem, vision, and change impacts.

Problem/ Vision

What is the
problem?

Objectives/ Outcomes

What are the
outcomes we expect
to achieve?

Change Impacts

What needs to
change to correct
the problem?

Lightweight Business Case

What is the
justification for
proceeding?

Strategy Analysis, Business Requirements, and Portfolio Management



Features

Features document **Stakeholder Requirements** and needed business changes.

Customer Needs

- Personas
- Scenarios
- Use Cases
- Initial Set of User Stories

Business Change Impacts

- People
- Technology
- Business Processes
- Business Rules
- Data

Conditions of Satisfaction

What needs to be completed to be done?

Test Scenarios

How do we know that the solution is working?

Discovery, Roadmapping, Business Change, and Release Planning



Stories

Stories are used to document **Functional Requirements**.

Card

A placeholder for future conversations between the developer and the user

Conversation

Discussions between the developer and the user concerning the functionality

Confirmation

Documented tests to confirm the user story is complete and working as expected

Functional Requirements and Acceptance Testing



What Problem Do User Stories Resolve

The Problem

- Requirements are a communication problem – the business must communicate to developers what they need
- Often, users have difficulty expressing what they need
- Words are imprecise
- Requirements change over time

So, Why User Stories?

- User Stories are written in the user's language but are understandable by both developers and the business
- Shifts focus from writing to talking

Remember

- Don't forget: the story text is less important than the conversation



User Stories

**As a [user role], I want to [goal]
so I can [reason]**

For example:

- As a registered user, I want to log in
so I can access subscriber-only content

User Story ≠ Requirement



Guidelines for Writing Good User Stories

1. Always start with a hierarchy such as: Epics – Features – Stories
2. Maintain separate backlogs for Epics, Features and Stories – Do not place in single backlog or you will experience what is known as “User Story Hell”
3. Remember user stories are not the requirements – they are placeholders for discussion – don’t forget conversations and confirmations
4. The success of a user story is dependent on regular face-to-face communication between developers and users or user representatives.
5. Start with User Personas – Don’t write users stories as “As a User, I want to...”
6. Remember **size** when writing user stories – Try to write good user stories that are Sized right, are **Estimable** and do not have to be split later
7. Keep your stories simple and concise!
8. Don’t solely rely on user stories – add visualizations, examples, etc.
9. Make sure the business value is explicitly stated in the user story
10. Try to get users involved in writing user stories



Acceptance/Validation Criteria

- Written in **simple language**
- Often written in the form of **Given-When-Then**
- Define the **conditions of success/satisfaction**
- Provide clear user story **boundaries**
- **Remove ambiguity** by forcing the team *to think through how functionality will work from the user's perspective*
- Establish the basis for **acceptance testing**
- Function best when written by actual users



User Story: The Complete Picture

Story

As a social networker,
I want to upload my profile picture into Facebook
So that my online friends can see what I look like.

Acceptance Criteria

Given the user has a valid Facebook account and a digital picture on her computer,
When she uploads a picture in Facebook,
Then her the picture should be visible to all her friends in her network.

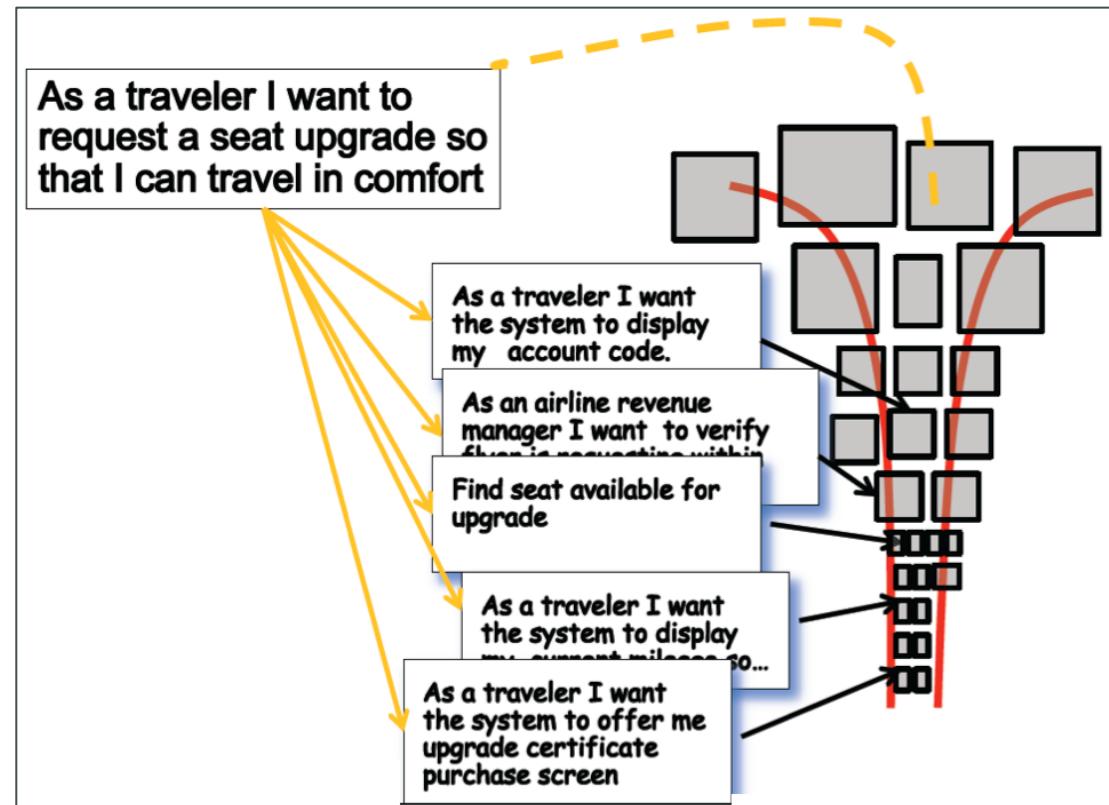
Conversations

- Must be able to accept any size picture and scale to fit
- Only the owner of the account can upload a picture
- Must accept gif, png, pdf, and jpg



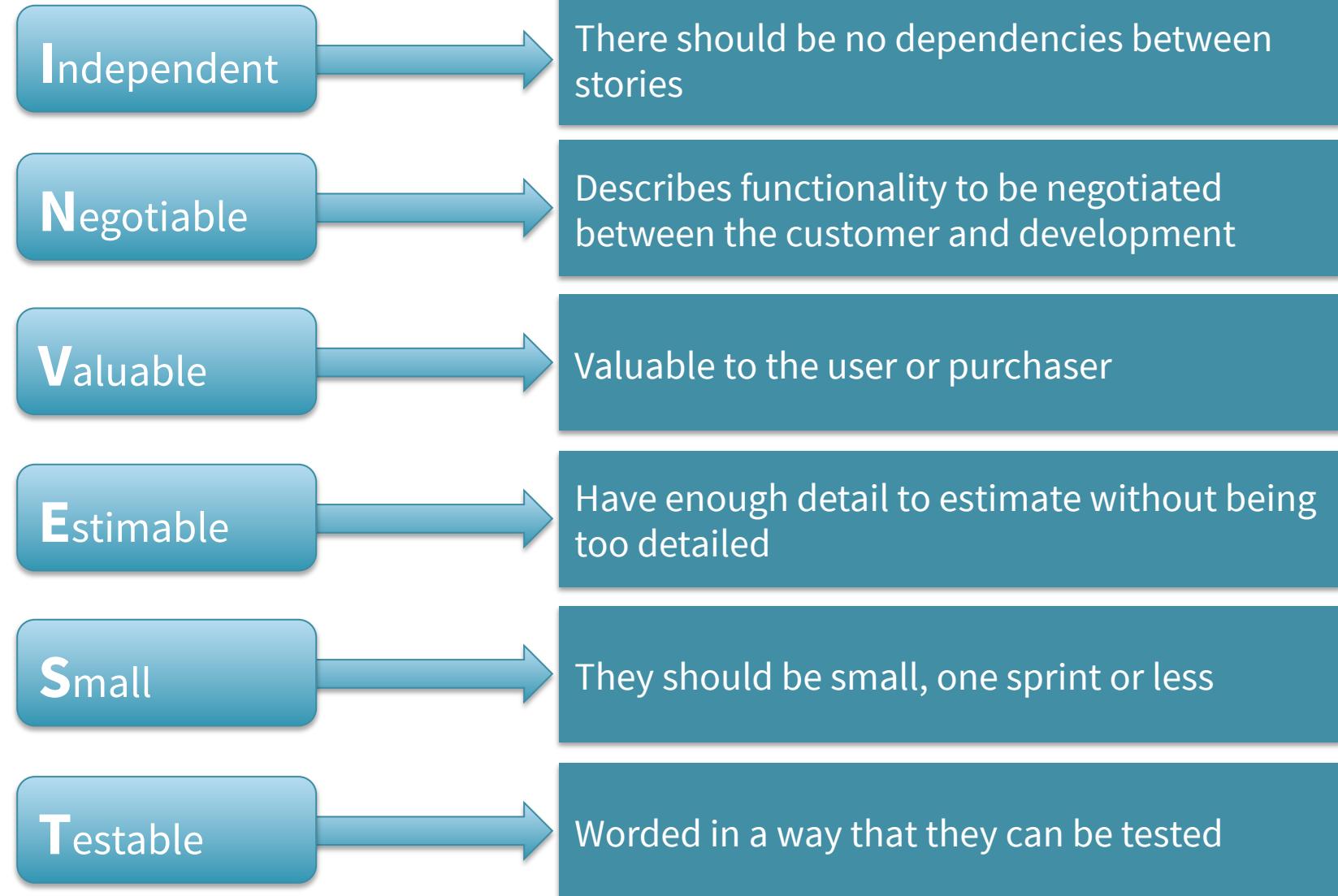
Splitting User Stories

- Often user stories have to be split because they do not fit into a Sprint.
- Below is a list of patterns that may be used to split user stories
 - By Variations of Data
 - By Workflow steps
 - By Data Entry Methods
 - By Business Rule Variations
 - By Operations (CRUD)
 - By Simple or Complex cases





INVEST Model for Stories





Non-Functional Requirements

- Non-Functional Requirements relate to qualities of the system that cut across user facing Features, such as security, reliability, and performance.
- User stories do not work well for Non-Functional Requirements. Employing user stories is like forcing a square peg in a round pole.
- The difference from Functional Requirements is that these qualities must be present throughout the system rather than delivered in one-shot like a user facing feature.
- Make sure that you engage with technical stakeholders in your organization, such as architects, user experience designers, and operations teams. These people can help an agile team spot NFR that are not captured in your user stories.

Consider Non-Functional Requirements from the start!



Putting it All Together

Epic

Increase Up-Sells on Sales Orders by 20% by January 2014.

Features

- Build up-sell business rules and data relationships considering customer profile and products being ordered.
- Add up-sell suggestions to Web-based order processing system.
- Add up-sell suggestions to Call-Center order processing system.
- Produce reports and queries to measure performance.

Impacts:

Personas

- On-line Customer
- Call Center Sales Agent
- Sales Data Manager
- Director of Sales

Teams (Systems)

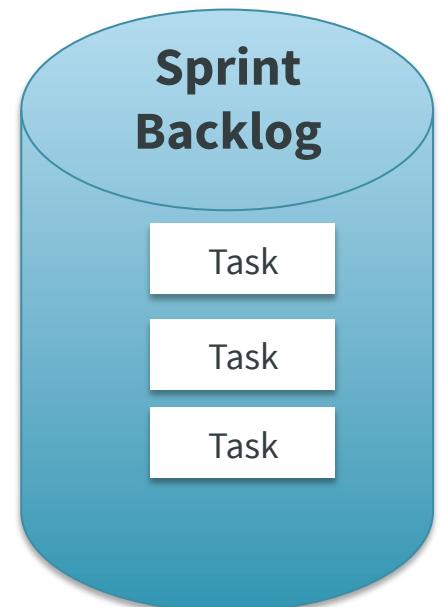
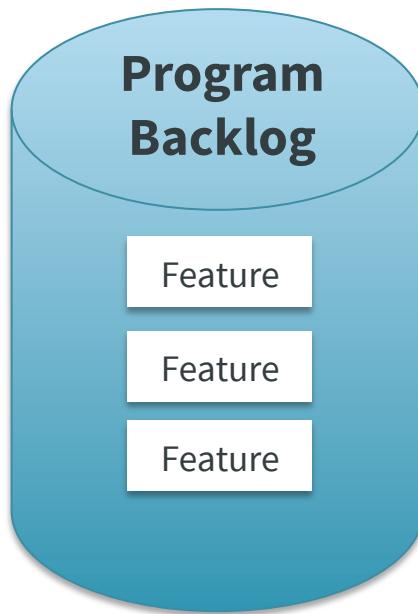
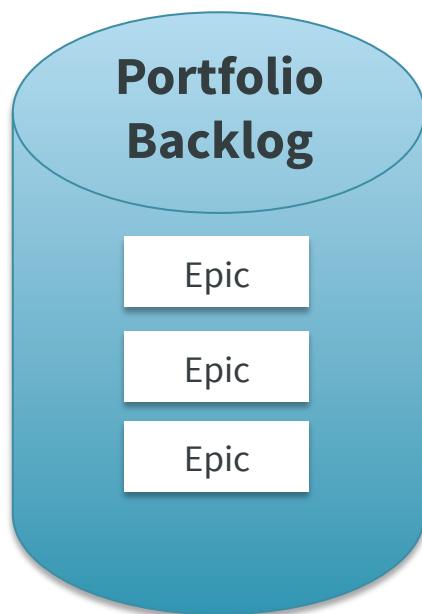
- Web Order Entry System
- Call Center Order Processing
- Product Master Data
- Upsell Business Rules

Stories

- As the Director of Sales, I want to know the amount of sales that were generated from up-sell recommendations provided to the customer so I can optimize the data to produce more up-sells.
- As the Director of Sales, I want to know the percentage of sales orders that had up-sell items so that I can optimize sales opportunities.



Backlogs



Portfolio
Management
Office

Product
Manager

Product
Owner

Team



Become Expert at Engaging Stakeholders

Three Keys:

1. Understand their perspective
2. Demonstrate benefits to them
3. Create opportunities to get them involved



Business Analysis Transformation

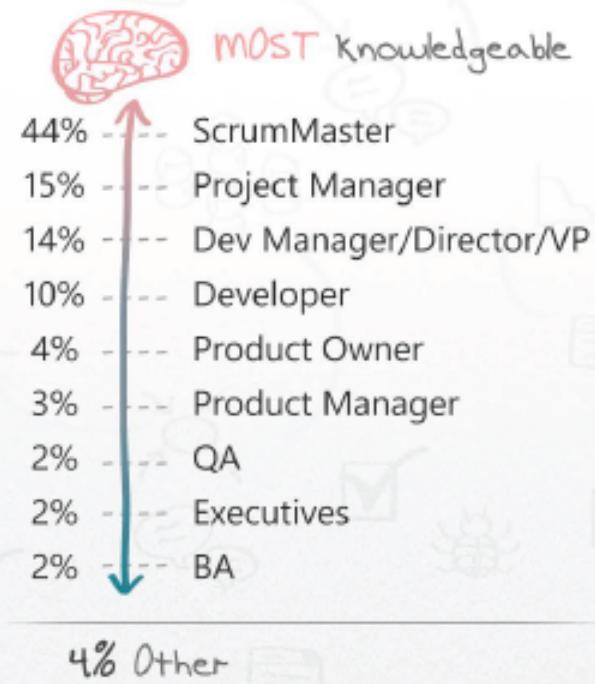


A Disturbing Fact



WHO KNOWS AGILE?

Product Owners are getting more knowledgeable; Business Analysts have taken their spot at the bottom of the pack.





Agile BA Transformation Challenges

- Working more collaboratively – everyone is involved in the requirements
- Moving away from paper requirement documents to managing backlogs
- Embracing requirement changes as a good thing
- Continuous discovery of requirements instead of defining all requirements upfront
- Changing approval processes
- Being collocated with the Team
- Defining Epics and Features
- Focusing on outcomes instead of outputs
- Learning new methods for engaging stakeholders
- Learning how to Inspect and Adapt
- Learning Lean concepts to eliminate waste
- Managing and coordinating business change activities
- Learning new techniques – Personas, Story Maps, User Stories, etc.
- Facilitating the big picture perspective

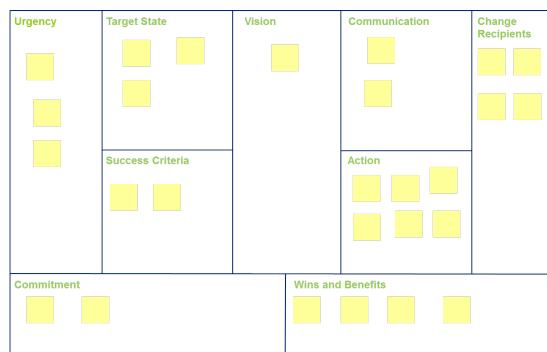


Lean Change is a Framework that takes an Experiment Based Approach to Change to de-risk Transformations

1

Brainstorm your initial change model.

The Lean Change Canvas

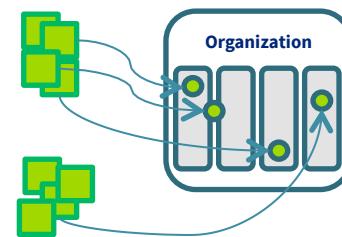


Is our change plan.

2

Build an iterative change plan with small change experiments.

Build Minimum Viable Changes (MVC)

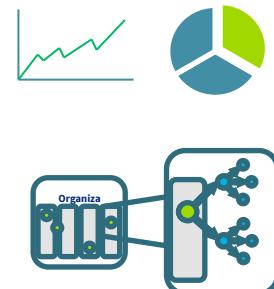


As incremental change actions applied throughout the organization

3

Systematically test your change.

Measure Outcomes



To inform future change actions and our plan.

Accelerate Adoption



MVCs for Business Analysis

Level	Minimum Viable Changes
Portfolio	<ul style="list-style-type: none">• Epics• Business Objectives and Outcomes• Lightweight Business Cases• Product Vision• Business Change Impacts
Program	<ul style="list-style-type: none">• Features• Program Increments• Customer Discovery• Inspect and Adapt Workshops• Stakeholder Engagement• Features• MVCs• Transition Requirements (Releases)• Personas• Scenarios• Lean Change Canvas• Customer Journeys and Touchpoints
Team	<ul style="list-style-type: none">• User Stories• Acceptance Tests• Product Backlog



How Agile Business Analysis Delivers More Value

	BA Activity/Artifact	Outcome
1.	Definition of Clear Business Objectives with Measurable Outcomes Linked to Features and Stories	<ul style="list-style-type: none">• Higher Revenue• Lower Cost
2.	Elimination of Unnecessary Features	<ul style="list-style-type: none">• Lower Cost• Faster Time to Market
3.	Validating Needs using MVPs, Prototyping, and Experiments versus using code	<ul style="list-style-type: none">• Lower Cost• More Rapid Product to Market Fit
4.	Using Better Discovery Methods (i.e., Personas, Models, Scenarios, Customer Journeys, Touchpoints)	<ul style="list-style-type: none">• Lower Cost• Less Developer Frustration• Higher Quality
5.	Writing Clear Acceptance Criteria used for Defining When Done and for Testing	<ul style="list-style-type: none">• Higher Quality• Lower Cost• Less Developer Frustration
6.	Implementing Enhanced Methods for Business Change	<ul style="list-style-type: none">• Faster and Wider User Adoption• Higher Stakeholder Satisfaction• Better Business/IT Alignment
7.	Developing Better Requirements	<ul style="list-style-type: none">• Less Rework• Lower Costs (Fewer Iterations)• Higher Quality
8.	Implementing Enhanced Inspect and Adapt Methods for BA	<ul style="list-style-type: none">• Higher Quality /Fewer Defects
9.	Increasing Stakeholder Engagement in discovery, development, and transition activities	<ul style="list-style-type: none">• Higher Quality and Satisfaction• Delivering Products that Customers want
10.	Operating a Lean Value Stream with Integrated Discovery, Development, and Delivery Process	<ul style="list-style-type: none">• Less Waste• Lower Costs

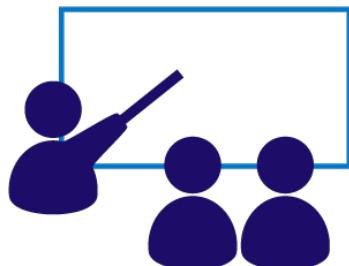


Agile Business Analysis Transformation Is Not Easy!!!

Doing It Right Saves Money and Time.



Strategy



Training



Coaching



Software

How We Can Help

- Develop strategy and roadmap for agile business analysis transformation
- Plan and conduct change experiments
- Design efficient agile business analysis processes
- Provide training on agile methods
- Provide mentoring on projects
- Provide software for defining epics, features, and stories and collaborating with developers and business stakeholders



Let's talk about your organization's needs.

We'll help develop a strategy for scaling to agile
with no obligation to purchase.

**Sign up for a free
needs assessment today!**

bit.ly/assess_needs