



## **Stop the Insanity, Let's Be Agile**



**Presented by:**

**William F. Nazzaro  
Process Synergy, LLC**

**bill@Process-Synergy.com  
(484) 854-1354**

**Version 3.00**

# Premise and Key Learning Points

- ❖ Many companies have employed the “Big Requirements Up-Front (BRUF)” or have found themselves mired in producing piles of paper to only have these documents out-of-date and not well understood by both the customer and the development team.
  
- ❖ In this presentation we will demonstrate:
  - ❶ How user stories are different from traditional approaches to requirements and how they combat “BRUF” and reams of documentation;
  
  - ❷ Leveraging user stories to capture the intent of our customer;
  
  - ❸ The anatomy of a user story and the three major elements of user stories (Card, Conversation, Confirmation).

# Process Synergy Background

- ❖ **A Philadelphia-based consultancy dedicated to the highest-quality delivery of project mentoring, training and organizational transformation**
- ❖ **Full lifecycle consulting, coaching and mentoring**
  - Requirements – Use Cases, User-Stories, IIBA
  - Process – Agile (XP, TDD), Unified Process, CMMi
  - Project Management – Scrum, Iterative, PMI
  - Architecture – Service-Oriented Architecture
  - Organizational Change Management and Improvement
  - Delivered On-Site, Globally
- ❖ **Full lifecycle software training curriculum**
  - Highly differentiated offerings
  - Experiential approach and skills building
  - Virtual and on-site
- ❖ **Multi-tool vendor expertise**
  - IBM Rational, Microsoft, Borland, HP Mercury, BluePrint, VersionOne, RallyDev

# Presenter's Background



- ❖ **Twenty years of success in delivering enterprise software solutions for Fortune 100 companies, he has provided unparalleled services in project mentoring, use case and user story training and modeling, agile SCRUM, XP, TDD, service-oriented architecture, Unified Process and CMMi adoption, Java, C++, C, and Smalltalk programming languages, and technical curriculum development and delivery.**
- ❖ **His pragmatic emphasis on project execution and process improvement has benefited major insurance, health-care, human resources, stock market organizations and Department of Defense where he has enabled the transformation of their corporate development processes and development teams of diverse skill-levels.**
- ❖ **He has provided in-depth talks on Agility, Use Case Modeling, Service-Oriented Architecture, Unified Process, Object Technology and Data Modeling.**
- ❖ **He has extensive background in project management, and has successfully led teams on multi-million dollar projects to provide the highest-quality technical solutions in the most efficient and effective manner.**

# Introduction

**Many companies have employed the Big Requirements Up-Front (BRUF) or have found themselves mired in producing piles of paper to only have these documents out-of-date and not well understood by both the customer and the development team.**

**In this presentation we will discuss an alternative to BRUF, User Stories, and what we should expect when we transition to this technique.**

# Agenda

- ❖ **Some Key Principles of Agile Requirements**
- ❖ **Why Doesn't Big Requirements Up-Front (BRUF) Work?**
- ❖ **What are User Stories?**
  - Anatomy of a User Story
  - The 3 Major Elements
  - User Story Guidelines
  - INVEST Model
- ❖ **How are User Stories Different?**
- ❖ **Benefits of Agile – Some Numbers To Consider**
- ❖ **Wrap-Up**

# 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 items on the left more."

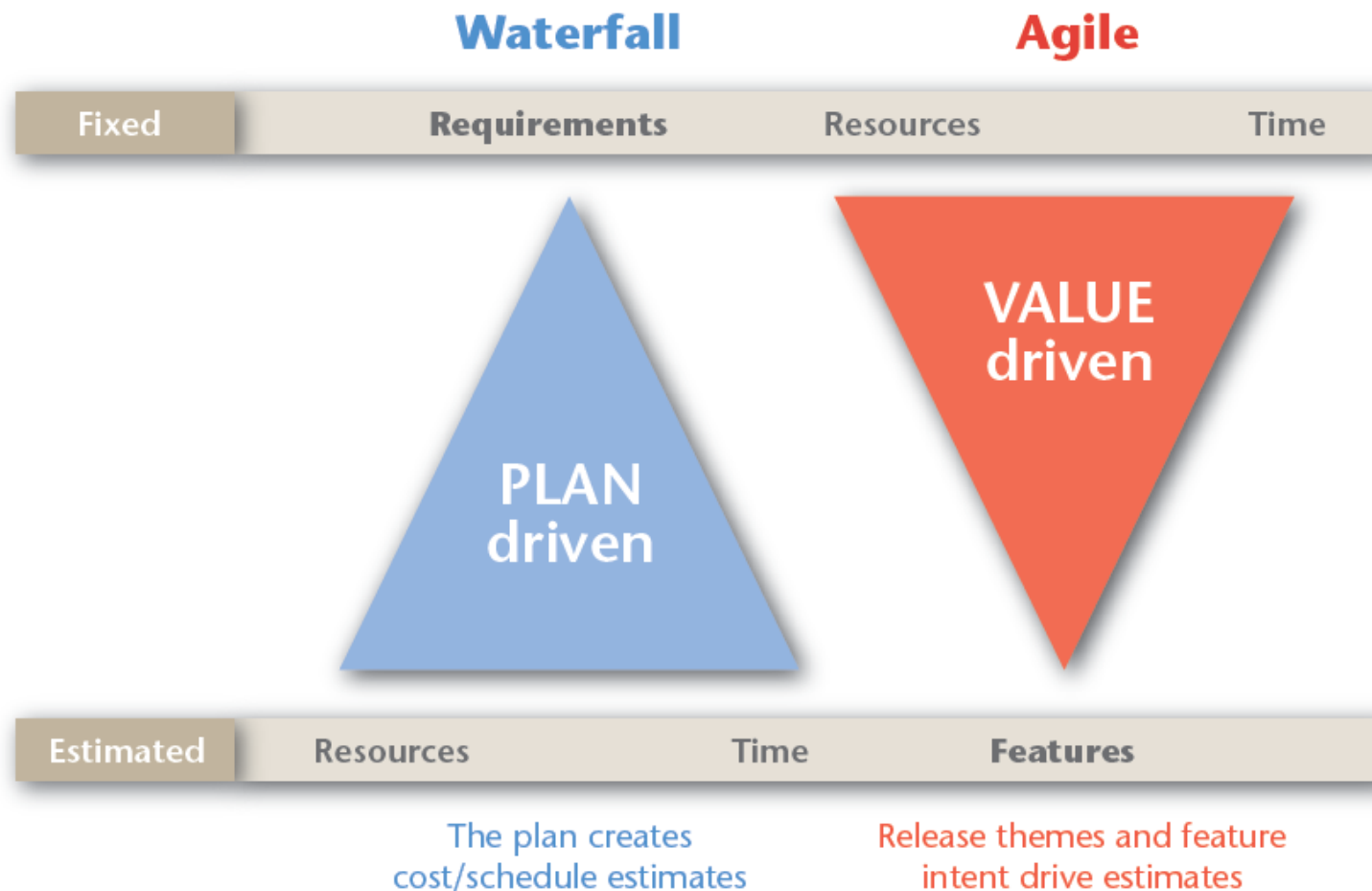
\* [www.agilemanifesto.org](http://www.agilemanifesto.org)

# Agility — An Attitude and Philosophy

- ❖ **Agility is not a process, rather it's an attitude and a philosophy**
  
- ❖ **The defining factors of agility are**
  - How do we continually provide value to our customer?
  - How do we respond to, or react to, change in our projects?
  
- ❖ **When change occurs**
  - Do we adapt to the change?
  - Do we “stick to the plan”?
  - Do we resist change at all costs?
  - Do we embrace change?



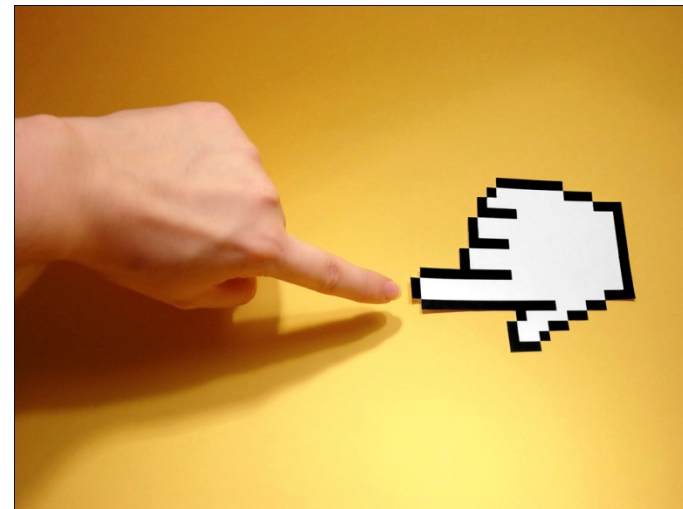
# Agility — An Attitude and Philosophy



From: <http://www.rallydev.com>

# Some Key Principles of Agile Requirements

- ❖ **Active user involvement is essential**
- ❖ **Requirements emerge and evolve as software is developed**
- ❖ **Agile requirements are ‘barely sufficient’**
- ❖ **Requirements are developed in small, bite-sized pieces**
- ❖ **Cooperation, collaboration, and communication between all team members is essential**



# Requirements are a Communication Problem

## ❖ Written Requirements

- Can be well thought through, reviewed and edited
- Provide a permanent record
- Are more easily share with groups of people
- Time consuming to produce
- May be less relevant or superseded over time
- Can be easily misinterpreted

## ❖ Verbal Requirements

- Instantaneous feedback and clarification
- Information-packed exchange
- Easier to clarify and gain common understanding
- More easily adapted to any new information known at the time
- Can spark ideas about problems and opportunities



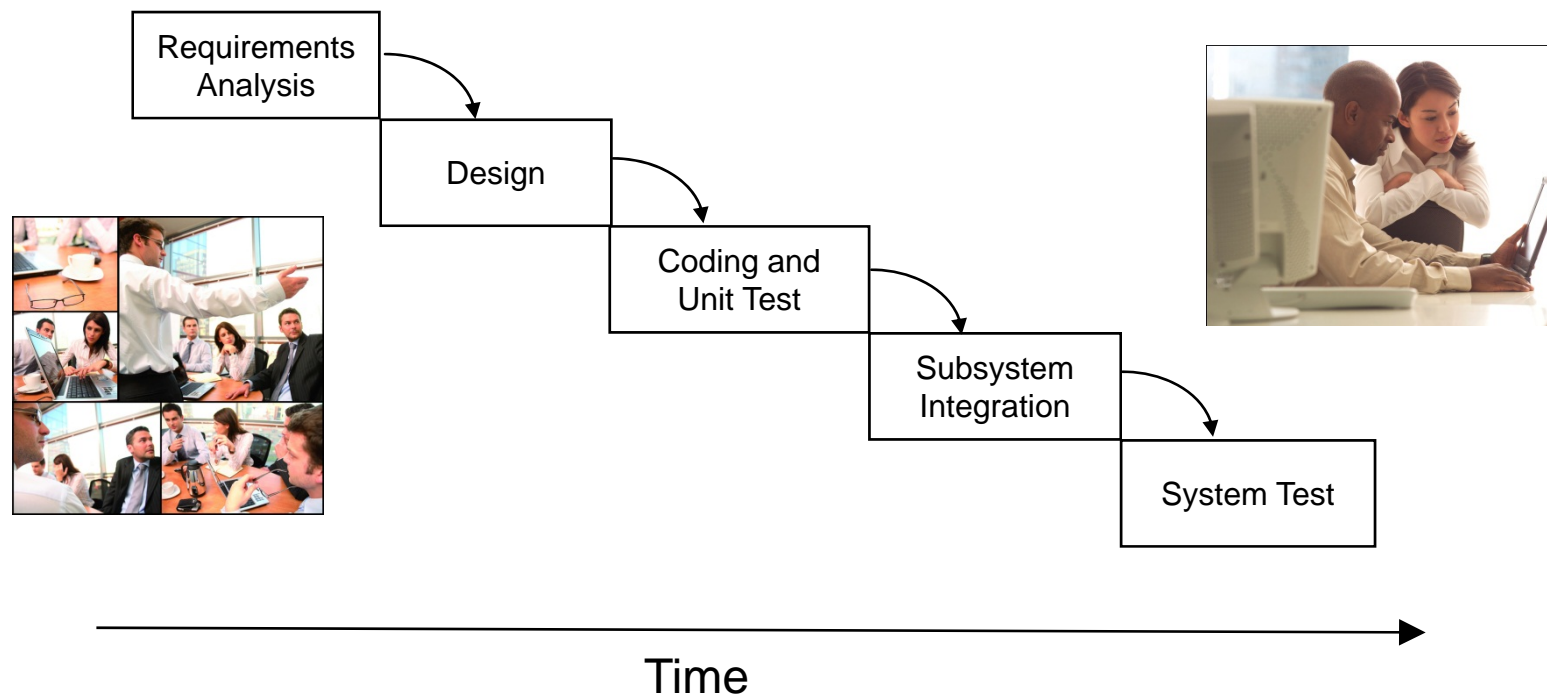
**As we will see, user stories seek to combine the strengths of written and verbal communication**

# Where Are We?

- ❖ **Some Key Principles of Agile Requirements**
- ❖ **Why Doesn't Big Requirements Up-Front (BRUF) Work?**
- ❖ **What are User Stories?**
  - Anatomy of a User Story
  - The 3 Major Elements
  - User Story Guidelines
  - INVEST Model
- ❖ **How are User Stories Different?**
- ❖ **Benefits of Agile – Some Numbers To Consider**
- ❖ **Wrap-Up**

# Why Doesn't "Big Requirements Up-Front" Work?

- ❖ **Generally there's a fallacy and flaw that we can elicit and document our requirements all at once**
  - Customer's don't typically know their requirements
  - Customer's requirements and solution expectation change of time

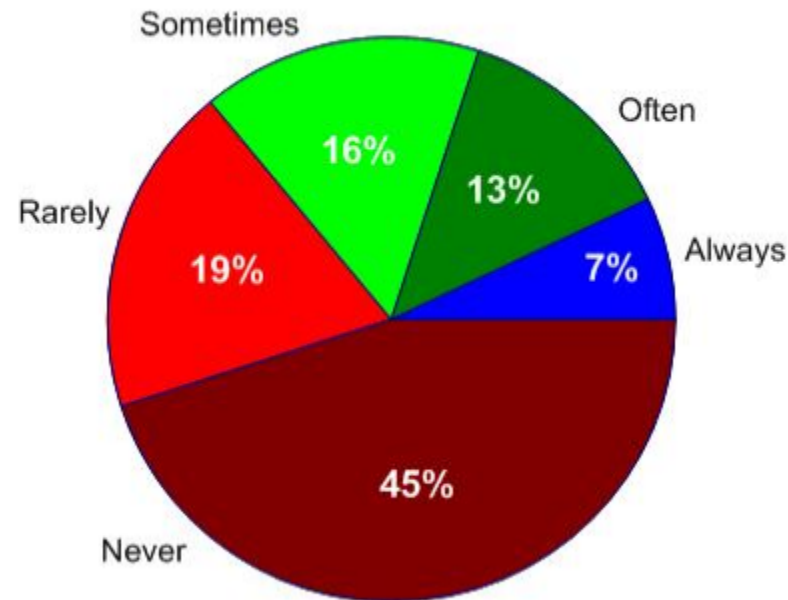


# What's the Goal of Our Software Development Activities?



# Serial Approach to Requirements

Average percentage of delivered functionality actually used when a serial approach to requirements elicitation and documentation is taken on a "successful" information technology project.



Source: Chaos Report v3, Standish Group.

Copyright 2005-2006 Scott W. Ambler

# Wastage Occurs for Several Reasons – 1

## ❖ The requirements change

- The period between the time that the requirements are “finalized” to the time that the system is actually deployed will often span months, if not years
- During this timeframe changes will occur in the marketplace, legislation will change, and your organization’s strategy will change
- All of these changes in turn will motivate true changes to your requirements

## ❖ People’s understanding of the requirements change

- More often than not what we identify as a changed requirement is really just an improved understanding of the actual requirement
- People aren’t very good at predicting what they want
- They are good at indicating what they sort of think that they might want, and then once they see what you’ve built then can then provide some feedback as to how to get it closer to what is actually needed
- It’s natural for people to change their minds, so you need an approach which easily enables this

Source: <http://www.agilemodeling.com/essays/examiningBRUF.htm>



## Wastage Occurs for Several Reasons – 2

### ❖ People make up requirements

- An unfortunate side effect of the traditional approach to development is that we've taught several generations of stakeholders that it's going to be relatively painless to define requirements early, but relatively painful to change their minds later
- In other words, they're motivated to brainstorm requirements which they think they might need at some point, guaranteeing that a large percentage of the functionality that they specify really isn't needed in actual practice

### ❖ You effectively put a cap on what you will deliver

- Very often with a BRUF approach you in effect define the best that will be delivered by the team and then throughout the project you negotiate downwards from there as you begin to run out of resources or time

Source: <http://www.agilemodeling.com/essays/examiningBRUF.htm>

# Where Are We?

- ❖ Some Key Principles of Agile Requirements
- ❖ Why Doesn't Big Requirements Up-Front (BRUF) Work?
- ❖ **What are User Stories?**
  - Anatomy of a User Story
  - The 3 Major Elements
  - User Story Guidelines
  - INVEST Model
- ❖ How are User Stories Different?
- ❖ Benefits of Agile – Some Numbers To Consider
- ❖ Wrap-Up

# What are User Stories

- ❖ **Describes desired functionality from the customer's, purchaser's or user's perspective**
- ❖ **A simple, written description of a piece of functionality that will be valuable for the user or owner of the software**
- ❖ **A "good" user story describes the desired functionality, who wants it, how and why the functionality will be used, and is not technical**

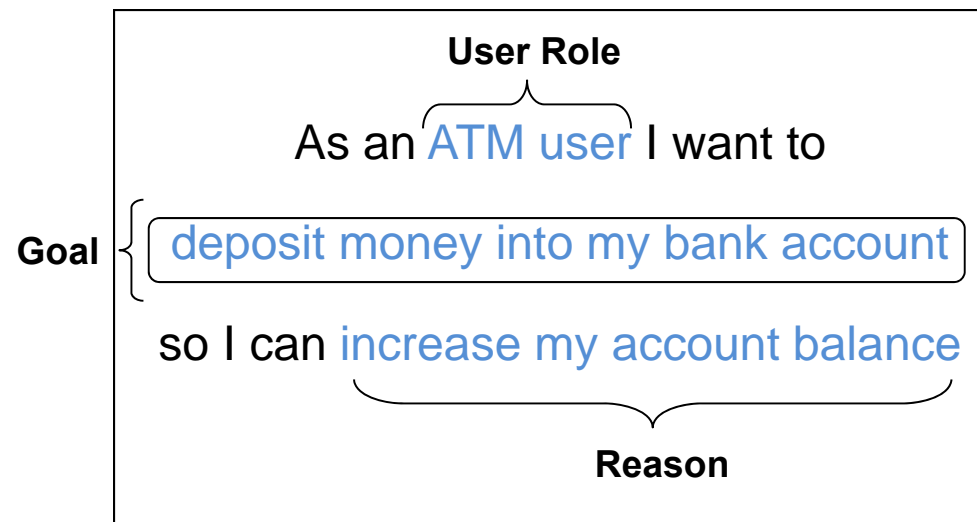
As an **ATM user** I want to **withdraw funds from my bank account** so I can **increase my cash on hand**.

As an **ATM user** I want to **transfer money from one bank account to another bank account** so I can **increase my account balance**.

As a **Bank Business Owner** I want to **set the ATM's withdrawal parameters** so the **ATM will provide funds to customers but protect against fraudulent activities**.

# Anatomy of a User Story

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



3x5 Index Card

## The 3 Major Elements

- ❖ **Card**: Contains the written description of the user story for planning purposes and as a reminder
- ★ **Conversation**: This is the meat of the story; the dialogue that is carried out with the users; recorded notes; mockups; documents exchanged
- ❖ **Confirmation**: The acceptance test criteria that the user will utilize to confirm the story is completed



The card is the most visible, but it is important to think of the card as representing the story, not as the whole story itself. The whole story includes the conversations about it and tests to confirm it.

# ATM User Story Examples

As an **ATM user** I want to **withdraw funds from my bank account** so I can **increase my cash on hand**.

As an **ATM user** I want to **transfer money from one bank account to another bank account** so I can **increase my account balance**.

As an **ATM operator** I want to **restock the ATM with money** so the **ATM will be available for customers to withdraw funds**.

As a **Bank Business Owner** I want to **set the ATM's withdrawal parameters** so the **ATM will provide funds to customers but protect against fraudulent activities**.

# User Story Guidelines

## ❖ Good stories achieve a goal

- For each user role, identify the goals the role has for interacting with the system
- Write stories that achieve a meaningful goal for the user
- Write stories that provide some level of end-to-end functionality

## ❖ Use the user role in the story

## ❖ Focus on one user in the story, and use the active voice

## ❖ Avoid user interface design details

## ❖ Focus your attention on the high priority stories that will be implemented in the next iteration

# How Detailed Should a User Story Be?

**Detailed enough**

**Detailed enough for the team to work from,  
and further details to be established and clarified  
at the time of development**

**A well-written user story follows the [INVEST model](#)**



# INVEST Model

❖ Independent

❖ Negotiable

❖ Valuable

❖ Estimable

❖ Small

❖ Testable

# Where Are We?

- ❖ **Some Key Principles of Agile Requirements**
- ❖ **Why Doesn't Big Requirements Up-Front (BRUF) Work?**
- ❖ **What are User Stories?**
  - Anatomy of a User Story
  - The 3 Major Elements
  - User Story Guidelines
  - INVEST Model
- ❖ **How are User Stories Different?**
- ❖ **Benefits of Agile – Some Numbers To Consider**
- ❖ **Wrap-Up**

# How are User Stories Different? – 1

**User Stories are not meant to be complete!**

**When working with one client I told them a user story could be thought of as a “to-do” list.**

**The “to-do” was we had to have a further detailed conversation to understand the client’s intent.**

## How are User Stories Different? – 2

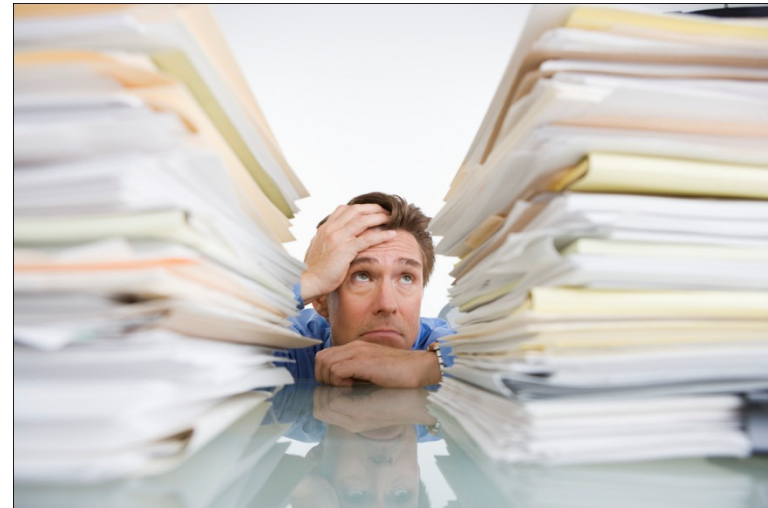
**A user story is a high-level description  
of the requirements to be built.**

**It usually fits on a 3x5 index card and is a  
“promise for a conversation”  
later between the team and the customer.**

Elssamadisy: Agile Adoption Patterns, 2009

# User Stories and Acceptance Testing – 1

- ❖ **The challenge for many companies is they adopt an agile practice for requirements, but they fail to adopt its counterpart which is Test-Driven Development**
- ❖ **Reality is we have to uncover and capture the details of what's needed somewhere, this is done through Acceptance Testing**
- ❖ **A principle of agility is maximizing the art of what's not done**
  - Why elicit and capture requirements into a requirements document to then translate those requirements into a test document?
  - Why not capture our requirements in our test document?



## User Stories and Acceptance Testing – 2

- ❖ **Acceptance tests provide basic criteria that can be used to determine if a user story is fully implemented \***
- ❖ **Having criteria that tells us when something is done is the best way to avoid putting too much, or too little, time and effort into it \***
- ❖ **Acceptance tests provide a great deal of information that the developers can use in advance of coding their story \***

As an **ATM user** I want to **withdraw funds from my bank account** so I can **increase my cash on hand**.

**Note: Funds dispensed in multiples of \$20**

**(Index Card Flipped Over)**

- 1. Sufficient funds available for requested amount (pass)**
- 2. Requested amount exceeds account balance (fail)**
- 3. Requested amount exceeds daily withdrawal amount maximum (fail)**
- 4. Invalid ATM card inserted (fail)**

\* Cohn: User Stories Applied, 2004

# User Stories and Acceptance Testing – 3

## ❖ Eventually we will then take these high-level acceptance tests and then start to codify them

- Unit Level Testing using XUnit
  - » JUnit, NUnit, RUnit, ...
- Acceptance Testing
  - » Framework for Integrated Test (FIT) available at <http://fit.c2.com>
  - » Fitnesse available at <http://fitnesse.org>
- GUI Testing
  - » Selenium available at <http://seleniumhq.org>

As an ATM user I want to transfer money from one bank account to another bank account so I can increase my account balance.

**Note: Funds transferred in multiples of \$10**

### (Index Card Flipped Over)

1. Sufficient funds available for transfer amount (pass)
2. Specified bank account invalid routing # (fail)
3. Amount entered not in multiples of \$10 (fail)
4. Invalid ATM card inserted (fail)

# Where Are We?

- ❖ **Some Key Principles of Agile Requirements**
- ❖ **Why Doesn't Big Requirements Up-Front (BRUF) Work?**
- ❖ **What are User Stories?**
  - Anatomy of a User Story
  - The 3 Major Elements
  - User Story Guidelines
  - INVEST Model
- ❖ **How are User Stories Different?**
- ❖ **Benefits of Agile – Some Numbers To Consider**
- ❖ **Wrap-Up**



# Benefits of Agile – Some Numbers To Consider \*

- ❖ **93% increased productivity** <sup>1</sup>
- ❖ **88% increased quality** <sup>1</sup>
- ❖ **83% improved stakeholder satisfaction** <sup>1</sup>
- ❖ **49% reduced costs** <sup>1</sup>
- ❖ **66% three-year, risk-adjusted return on investment** <sup>2</sup>
  
- ❖ **Reasons for Agile adoption include:**
  - 47% to better manage project scope <sup>3</sup>
  - 45% to create clear business requirements <sup>3</sup>
  - 40% to speed or better predict time to market <sup>3</sup>



<sup>1</sup> "Agile Methodologies: Survey Results", Shine Technologies, 2003

<sup>2</sup> Forrester Research, 2004

<sup>3</sup> "Agile 2006 Survey Results and Analysis", Digital Focus, 2005

\* Source: <http://www.rallydev.com>

# Wrap-Up

- ❖ **Some Key Principles of Agile Requirements**
- ❖ **Why Doesn't Big Requirements Up-Front (BRUF) Work?**
- ❖ **What are User Stories?**
  - Anatomy of a User Story
  - The 3 Major Elements
  - User Story Guidelines
  - INVEST Model
- ❖ **How are User Stories Different?**
  - User Stories Focus on Intent
- ❖ **Benefits of Agile – Some Numbers To Consider**
- ❖ **Wrap-Up**

**“If you really want to do something, you'll find a way...  
If you don't, you'll find an excuse.”**

**– Unknown**

# What We've Learned

## ❖ In this presentation we've demonstrated:

- ❶ How user stories are different from traditional approaches to requirements and how they combat “BRUF” and reams of documentation;
- ❷ Leveraging user stories to capture the intent of our customer;
- ❸ The anatomy of a user story and the three major elements of user stories (Card, Conversation, Confirmation).

## For More Information

- ❖ **To contact us our consulting and training services or how we work with organizations to build an Agile Delivery Model and Business Analysis Center of Excellence**
  - Please call +1.484.854.1354
  - E-mail: [bill@Process-Synergy.com](mailto:bill@Process-Synergy.com)
  
- ❖ **Ask about some of our popular courses like:**
  - Developing Requirements with User Stories
  - Applying Scrum with User Stories
  - Applying Scrum with Use Cases
  - Service-Oriented Analysis
  - Effective Use Case Development
  - Business Analysis – Requirements Discovery and Facilitation
  - Agile Project Management Primer



## **Stop the Insanity, Let's Be Agile**



**Presented by:**

**William F. Nazzaro  
Process Synergy, LLC**

**bill@Process-Synergy.com  
(484) 854-1354**

**Version 3.00**