

For Audio: Please dial
512-225-3113 – Passcode: 665346#
We will not be using VOIP today.

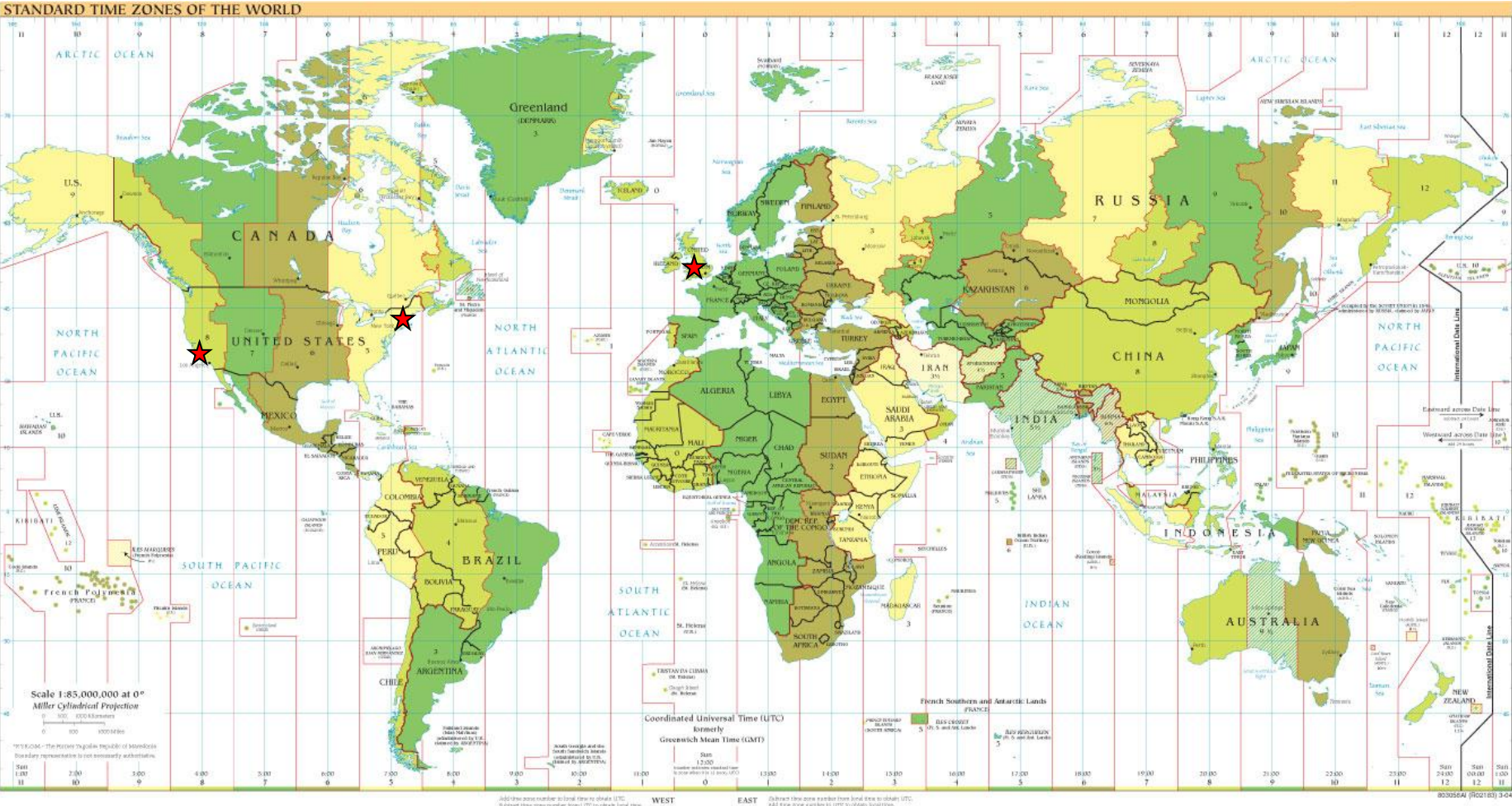
Requirements Made Easy with User Stories

Presented by **William F. Nazzaro**

Hosted by **Dave Bieg**, Executive Vice President



Use the Q&A feature to tell us what city you're in!



About DevelopMentor



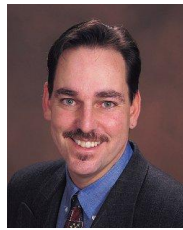
DevelopMentor provides solutions for all professionals involved in the lifecycle of software projects via our consulting, mentoring & learning services.

We're committed to ...

“Developing People Who Develop Software”



Bill Nazzaro



- **Twenty years of success in delivering enterprise software solutions for Fortune 100 companies**
- **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
 - Technical curriculum development and delivery.
- **Enabled the transformation of corporate development processes and development teams of diverse skill-levels.**
- **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Requirements Up-Front (BRUF) or have found themselves mired in producing piles of paper to only have these documents out-of-date and not well understood by both the customer and the development team.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.



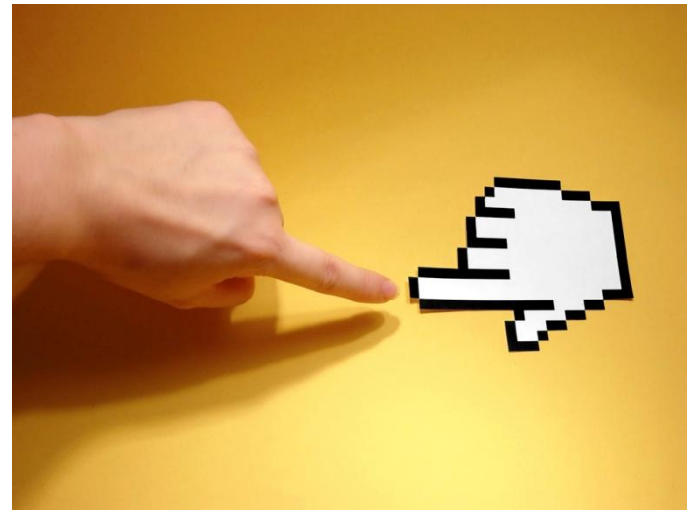
Agenda

- **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**



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**



Grade your organization on how well it currently meets these principles? [Use Q&A feature]

My organization:

- **A = excels at all**
- **B = is good at most**
- **C = is trying to improve**
- **D = doesn't support these principles, but they happen occasionally through efforts of individuals**
- **F = doesn't support them, never will**

Principles:

- 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



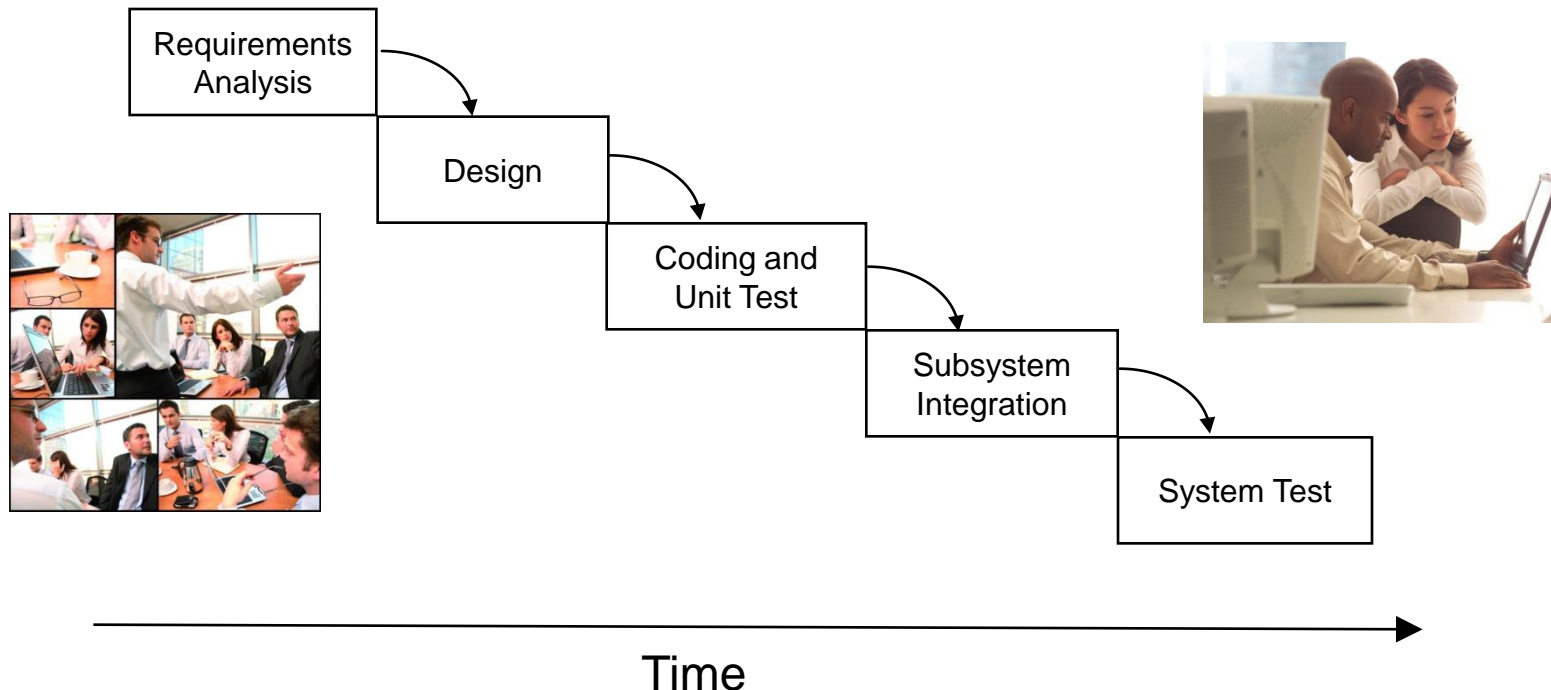
Where Are We?

- **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

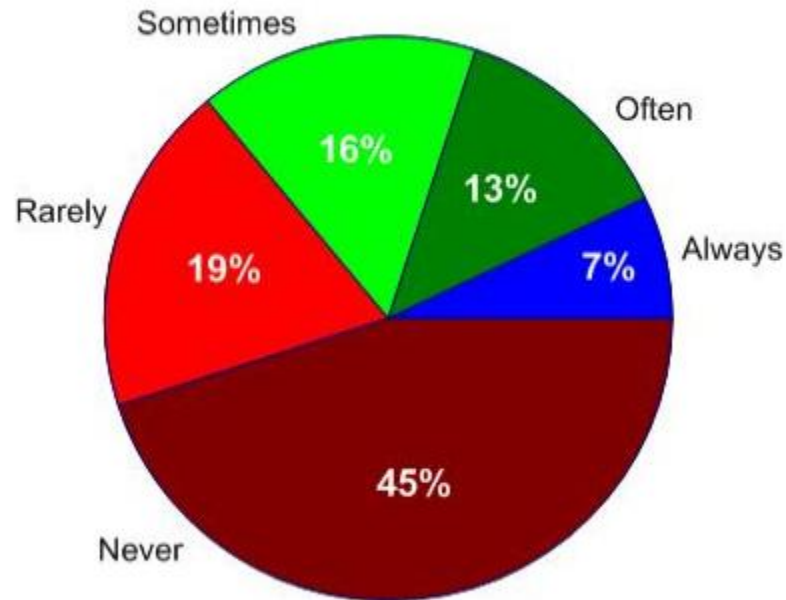


What's the Goal of Our Software Development Activities?



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

- **The requirements change**
- **People's understanding of the requirements change**
- **People make up requirements**
- **You effectively put a cap on what you will deliver**

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**



Polling Question

- **How experienced is our audience with User Stories?**



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**.

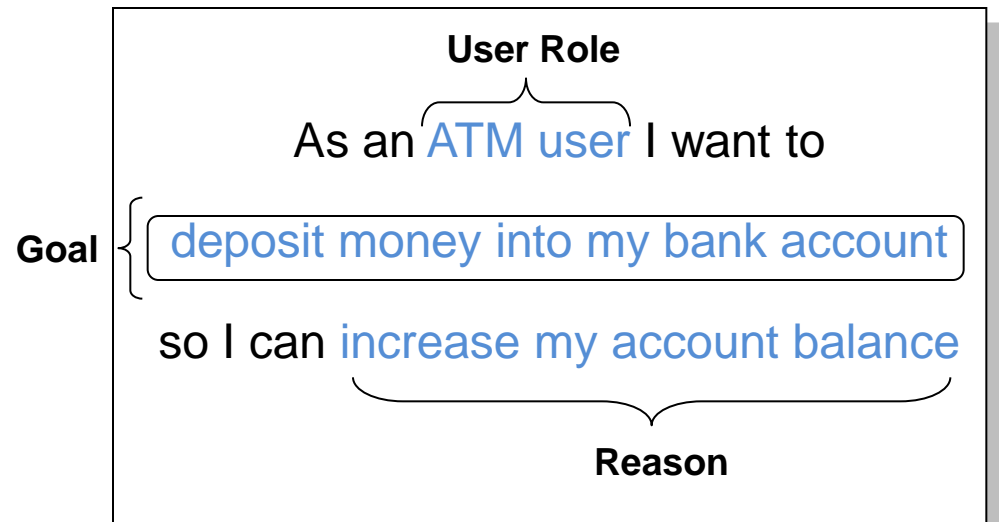
As an **ATM user** I want to **transfer money from one bank account to another bank account** so I can **increase my account balance**.

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.**



Can our audience share any other differences that they have discovered? [Use Q&A feature]



User Stories and Acceptance Testing

- **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**



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** ¹
- **88% increased quality** ¹
- **83% improved stakeholder satisfaction** ¹
- **49% reduced costs** ¹
- **66% three-year, risk-adjusted return on investment** ²

- **Reasons for Agile adoption include:**
 - 47% to better manage project scope ³
 - 45% to create clear business requirements ³
 - 40% to speed or better predict time to market ³



1 "Agile Methodologies: Survey Results", Shine Technologies, 2003

2 Forrester Research, 2004

3 "Agile 2006 Survey Results and Analysis", Digital Focus, 2005



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**

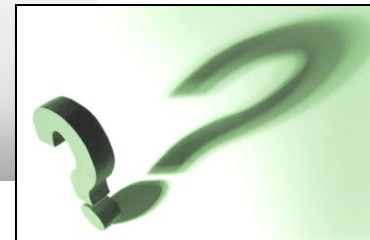


**“If you really want to do something, you'll find a way...
If you don't, you'll find an excuse.”**

– Unknown



Q & A



DevelopMentor

Agile Products & Services

- Curriculum
 - Foundations of Agile Development Using Scrum
 - Developing Agile Requirements with User Stories
 - Dec 8-9 in LA
 - Test Driven Development for C# Developers
 - Dec 10-11 in LA
- Consulting
- Mentoring



Contact me

Dave Bieg
Executive Vice President
DevelopMentor

678.296.8389
DBieg@Develop.com



Final Polling question



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)**

