Welcome to the SDIP Virtual Learning Center.

Today we will discuss

**Agile - Business Scenarios and User Stories!**

*Please note, you will hear some silence as we wait for additional participants to join the session.*

---

**Logistics**

- **Today’s Course:**
  - Total time of course is 2 hours
  - If you have problems seeing or hearing the presentation, please alert the training team using the chat function
Course Objectives

When you have completed this course, you will be able to:

• Define “Business Scenario” (or “Epic”) and “User Story,”
• Explain the importance of “Just-In-Time” elaboration of Business Scenarios and User Stories,
• Explain the importance of User Roles in the development of User Stories,
• Describe the attributes of Business Scenarios and User Stories,
• Describe multiple strategies for splitting Business Scenarios into User Stories, and
• Explain the importance of Acceptance Tests.

The Agile Habit

Comic from http://dilbert.com/
Business Scenarios and User Stories

Part I

Players in the Game
Product Owner: Driver of Business Scenarios and User Stories

- Domain Expert
- Study the Market
- Talk to Customers
- Understand the Business
- Run Focus Groups
- Competitive Analysis
- Go to Trade Shows

- .... Also take suggestions from the development team and understand feasibility/limitations

Scrum Master

- Facilitates meetings
- Removes obstacles for the team
- Tracks and reports on team progress
- Produces necessary reports
- Maintains project management tools
- Coordinates work of different teams
- Shares best practices among teams
- Establishes and maintains user groups
The Team

- Made up of:
  - Developers
  - Testers
  - User interface designers
  - Etc.
- Ideal team characteristics
  - Dedicated
  - Focused
  - Cross-functional
  - Self-organizing

Backlog “Iceberg”

Sprint/Iteration:
User story

Release backlog:
Business scenario/epic and user story

Product backlog
Business scenario/epic

Theme:
collection of related user stories and business scenario/epics
Prioritizing the Product Backlog

Business scenario

4  9  16  6  12
22  1  23  21  18
3  11  19  2  13
20  8  5  17  24
15  10  14  7
Release Plan to Iteration Plan

From Business Stories to User Stories
And it All Changes Over Time

The highest priority business scenario is *always* placed into the release plan first, no matter what.

A) True or B) False

What Do You Think?

POLL: Answer A or B
What Do You Think?

The highest priority business scenario is *always* placed into the release plan first, no matter what.

A) True or **B) False**

Agile Principles: Scenarios & Stories Should Always Provide Value

Image from SimaFore.com
Agile Principles: Just-In-Time Elaboration

Card-Conversation-Confirmation

Image from agileinaflash.blogspot.com
Card-Conversation-Confirmation (Cont.)

Card

- Business scenarios and user stories are written on cards.
- The card is a token for further conversation about what is desired.

Card Examples…

- The card also represents a “promise for a conversation” about the intent.

  - Log in to my web energy-monitoring portal
  - See my daily energy usage
  - Check my current electricity billing rate

Card-Conversation-Confirmation (Cont.)

Conversation

- The requirement itself is communicated from customer or product owner to programmers through conversation.
- Best when the “customer” represents as many user types as possible
- This conversation takes place throughout the process.
- The conversation is largely verbal with some documents.

- As a <role> I can <activity> so that <business value>

  - A list of what will make the story acceptable to the product owner.
Card-Conversation-Confirmation (Cont.)

**Confirmation**

- At the beginning of the iteration, the customer communicates to the programmers what she wants, by telling them how she will confirm that they've done what is needed.
- She defines the acceptance tests that will be used to show that the story has been implemented correctly.

Test Cases...
Based on discussions between PO and Team

**User Roles**

- **A user role** represents a population of user types and their intended interactions with the system.
- Stories should not be written from a single perspective or stories will be forgotten.
- **ATM:**
  - Account holder
  - Cash filler
  - Maintenance
  - Bank owner
  - Security officer
  - The bank (representing banking systems)

http://trussty-jasmine.blogspot.com/2012/03/atm-machine-doubtful-at-first.html
Structure of a Business Scenario or User Story

- As an [X], I want to [y] so I can [z].
  - x= user role (who)
  - y= desired functionality (what)
  - z= benefit/value (why)

Business Scenario Example
As a remote power input sensor node [X], I want to send a notification of incoming power fluctuation (outside of the specified operational range) to the power controller module [Y] so that appropriate diagnostic subroutines can be activated [Z].
**User Story Examples**

As a remote power input sensor node [X], I want to create an exception file whenever power coming into the system falls below 210 volts [Y] so that I can alert the power controller module of the problem [Z].

---

**Attributes of a business scenario or user story (INVEST)**

- Independent
- Negotiable
- Valuable to purchasers or users
- Estimate-able
- Small*
- Testable

* Only applies to user stories
**INVEST**

- **Independent**: Avoid dependencies between stories in the same iteration. Dependencies lead to prioritization and planning problems and cascading failure risks.
- **Negotiable**: Details negotiated via conversation between the product owner and the development team.
- **Valuable to purchasers or users**: Make someone other than the development team happy by showing interim progress the team can get feedback on.

**Estimate-able**

- What if the team cannot begin to estimate business scenario or user story because the technology or functionality is too new to them?
  - Run a **spike**: a brief “end-to-end executable experiment”
  - Preform a research task
- **Purpose**: To learn, to “buy information” so that you can estimate
- Spikes and research tasks become user stories
  - Timeboxed/bounded resource allocation
  - Accountability
INVEST

- **Small**: A user story must be small enough to meet all done criteria in one iteration (i.e. finish a “potentially shippable” feature)
- **Testable**: Demonstrating acceptance tests is a strong indication the functionality has been developed. Watch for “untestable” requirements being non-functional requirements or not specified well enough for the team to know what to do.

What Do You Think?
Why do Agile approaches emphasize the importance of user stories being valuable to purchasers or users?

A) To enable interim feedback
B) To prevent needless
C) Both A and B
What Do You Think?

Why do Agile approaches emphasize the importance of user stories being valuable to purchasers or users?

A) To enable interim feedback
B) To prevent needless infrastructure from being developed
C) Both A and B

Please Be Back in 5 Minutes!
Business Scenario Splitting-1

• Split business scenarios around the boundaries of the data.

• “As a business owner, I want to be able to enter my balance sheet information so I have a detailed view of my finances.”
  • “As a business owner, I want to be able to enter assets and liabilities ...”
  • “As a business owner, I want to be able to enter cash, securities, and real estate ...”
  • “As a business owner, I want to be able to enter detailed loan information ...”
  • “As a business owner, I want to be able to enter checking and savings account information ...”
  • “As a business owner, I want the values I enter to be validated so I know if I make any mistakes in my input.”

Business Scenario Splitting-2

• Split business scenario based upon the operations that are performed within the story.

  A common approach is to split along the boundaries of the common CRUD database operations – Create, Read, Update, Delete.

  – “As a business owner, I want to be able to enter the checks I write ....”
  – “As a business owner, I want to be able to get a listing of the checks I write ....”
  – “As a business owner, I want to be able to edit the information on the checks I previously entered ....”
  – “As a business owner, I want to be able delete previously-entered checks ....”
Business Scenario Splitting-3

• Initially, removing non-functional requirements (such as security, logging, error handling) and create two versions of the story: one with and one without the support for the non-functional requirement.

  – “Make it work, then make it faster.”
    • Kernighan and Plaugher (The Elements of Programming Style, 1974)
  – But the second story cannot be forgotten!
  – Such an approach can also serve to emphasize important non-functional requirements.

What Do You Think?

Example business scenario: As a bank [X], I want to receive a file showing all checks deposited in our ATMs [Y] so that I can clear the checks and credit the appropriate accounts [Z].

Use the CRUD (Create, Read, Update, and Delete) method to break out a smaller user story from this business scenario.
What Do You Think?

Example business scenario: As an owner of a music store and musician participating in an online musician’s community, I can post various items for musicians to consider.

Use the Data Boundaries method to break out a smaller user story from this business scenario.

Non-functional Requirements and Constraints

• Non-functional requirements are requirements which are not specifically concerned with the functionality of a system but place restrictions on the product being developed.
  – Response time must be less than one second with up to 100 concurrent users.
  – Role-based access will be used to restrict the functionality available for each user role.

• Constraints are a type of non-functional requirement that restricts the implementation of the system or the development process. Constraints have no direct effect on the users’ view of the system.
  – All graphing and charting will be done using a third-party library.
  – The software will be written in Java.
### Non-functional User Story Examples

**As a customer,** I want to be able to run your product on all versions of Windows from Windows 95 on so that I don’t have to upgrade all my systems.

**As the CTO,** I want the system to use our existing orders database rather than create a new one so that we don’t have one more database to maintain.

**As a user,** I want the site to be available 99.999% of the time I try to access it so that I don’t get frustrated and find another site to use.

**As a user,** I want the driving directions to be the best 90% of the time and reasonable 99% of the time so I don’t get lost.

---

### What Do You Think?

It’s OK to temporarily build functionality without considering non-functional requirements or error conditions.

A) True or B) False
What Do You Think?

It is OK to temporarily build functionality without considering non-functional requirements or error conditions.

A) True or B) False

Acceptance Test-Driven Development/ Behavior-Driven Development

Part II
Acceptance Tests

- Creation of test cases is a collaborative effort of:
  - Customer/product owner (talks, clarifies, thinks through);
  - tester (guides, structures, questions);
  - developer (listens, questions, learns expectations).
- Writing the acceptance tests is a means of structuring the “conversation” around expectations.
- When automated, tests become executable documentation.
- Provided to developer as soon as possible at start of iteration.
Characteristics of a Good Acceptance Test Case

• **Specific**
  - Example of a good test:
    • Set up: John Smith has $500 in his checking account.
    • Operation: John Smith authenticates his ATM card by providing a correct PIN after inserting in the machine. He then withdraws $100 from his checking account.
    • Verify: John Smith gets his $100. His checking account balance is now $400.
  - Example of a bad test:
    • Operation: Account holder withdraws cash from an ATM.
    • Verify: Account holder gets cash.

• **Repeatable** by any tester with the same results every time (assuming the software remains the same)
  - Specific steps are documented including pre-conditions/set-up
  - Specific data is provided for input and output

• **Objective**, based upon requirement, not implementation.
• Binary accept/reject (or pass/fail) criteria is provided based upon pre-determined expected results

**Expected Results May be Time Consuming to Produce . . . Do It Anyway**
What Do You Think?

Acceptance test case creation provides a structure for some of the conversation needed to elaborate requirements?

A) True or B) False

What Do You Think?

Acceptance test case creation provides a structure for some of the conversation needed to elaborate requirements?

A) True or B) False
Test Automation

- Automated acceptance test cases:
  - Become “executable documentation”
  - Can be used for

- Behavior-Driven Development Automation
  - JBehave
  - Cucumber
  - SpecFlow
  - ABB SafetyInsight
  - ABB AlarmInsight

Behavior-Driven Development: Given-When-Then

- Given <precondition>
- When <actor + action>
- Then <observable result>

- Example:
  Given the account balance is $100
  And the card is valid
  And the machine has enough money
  When the account holder requests $20
  Then the ATM should dispense $20
  And the account balance should be $80
  And the card should be returned
More Examples

• **Given** the account balance is $10  
  *And* the card is valid  
  *And* the machine has enough money  
  **When** the account holder requests $20  
  **Then** the ATM should not dispense any money  
  *And* the ATM should say there are insufficient funds  
  *And* the account balance should be $20  
  *And* the card should be returned

• **Given** the card is disabled  
  **When** the account holder requests $20  
  **Then** the ATM should retain the card  
  **And** the ATM should say the card has been retained

Resources

**Business Scenarios & User Stories Resources**

- **SDIP Portal:**  
  [https://abb.sharepoint.com/sites/SDIP/Portal/_layouts/15/start.aspx#/SitePages/Home.aspx](https://abb.sharepoint.com/sites/SDIP/Portal/_layouts/15/start.aspx#/SitePages/Home.aspx)

- **Agile Software Requirements**  
  by Dean Leffingwell

- **SDIP Course Catalog:**  
  [https://abb.sharepoint.com/sites/SDIP/Portal/_layouts/15/start.aspx#/SitePages/Course%20Catalog.aspx](https://abb.sharepoint.com/sites/SDIP/Portal/_layouts/15/start.aspx#/SitePages/Course%20Catalog.aspx)
Let’s Review

We’ve accomplished a lot today. We...

• Defined “Business Scenario” (or “Epic”) and “User Story,”
• Described how Business Scenarios evolve into User Stories in Agile development processes,
• Explained the importance of “Just-In-Time” elaboration of Business Scenarios and User Stories,
• Described the Card-Conversation-Confirmation method of elaborating product requirements, and
• Explained the importance of User Roles in the development of User Stories.

Let’s Review

We also...

• Described the attributes of Business Scenarios and User Stories,
• Described multiple strategies for splitting Business Scenarios into User Stories,
• Explained the importance of Acceptance Tests, and
• Described the characteristics of good Acceptance Tests.
Course Evaluation and Follow Up

After today’s session, each participant will be connected to the course evaluation.

Also, please remember to send an e-mail to sdip.training@fr.abb.com if you are not logged in to your own computer for this session.

Thank you in advance for your cooperation!

Questions/Comments