Business Requirements Using
Unified Modeling Language

Eric H. Castain, SVP
Internet Services Group, Architecture
Wells Fargo
March 2005
Agenda

- UML Use Cases
- Storyboards as an effective way to model requirements – Visual Use Cases
- Easel: An Implementation of Visual Use Cases Using UML
- Benefits of the Storyboard Solution
- Status
- Questions
UML Use Cases

- A diagram that shows the relationships among actors and the system
- Allows definition of associations such as
  - Uses
  - Includes
  - Extends
  - Dependency, etc.
- Does not define the details of the Use Case
  - Current recommended solutions consist of
    - Narrative
    - Outline
    - Table-oriented
  - Resultant products are outside of the model and are not very robust
UML Example

- Defines
  - Stakeholder
  - System activities
  - Associations
  - Pre and Post conditions

- Does not define the details of the Use Cases
  - May add comments within the “notes” section of the use case
Narrative Example

**Use-case name:** Add a non-Wells Fargo Account

**Description:** This use case allows the actor, WF Customer, to define an account at another institution for ACH transfers.

**Narration:** A Wells Fargo customer transitions into the Add Non-Wells Fargo Account screen and specifies the bank name, account type, account number, etc. This information is validated by the system using the business rules to validate that we recognize the RTN entered by the customer, that the financial institution associated with the entered RTN participates in ACH …
Outline Example

1. This use case starts when the a Wells Fargo customer transitions into the Add Non-Wells Fargo Account screen
   1. The system prompts for the bank name, account type, account number, etc.
2. The actor enters their bank name, account type, account number, etc. of the account at the desired financial institution.
   1. The system validates inputs
   2. The system validates the business rules
      1. Does RTN exist in the Bank Directory?
      2. Does the financial institution represented by the RTN participate in ACH?
      3. …
# Table-oriented Example

<table>
<thead>
<tr>
<th>Wells Fargo Customer</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Selects the link that transitions into the Add Non-Wells Fargo Account screen</td>
<td>1.1. Prompts for the bank name, account type, account number, etc.</td>
</tr>
<tr>
<td>2. Enters their bank name, account type, account number, etc. of the account at the desired financial institution.</td>
<td>2.1. Validates inputs using standard data validation Rules</td>
</tr>
<tr>
<td>2.2. The system validates the business rules:</td>
<td></td>
</tr>
<tr>
<td>2.2.1 Does RTN exist in the Bank Directory?</td>
<td></td>
</tr>
<tr>
<td>2.2.2 Does the financial institution represented by the RTN participate in ACH?</td>
<td></td>
</tr>
<tr>
<td>2.2.3 …</td>
<td></td>
</tr>
</tbody>
</table>
UML Based Requirements

- Based on Standards
  - UML 2.0 notation with stereotyped extensions e.g. screens, UI controls, etc.
- Support multiple stakeholders
- Allow analysts, architects, etc. to collaborate in iterative workflows
- Use a model-based repository of shared, structured information
Visual Use Cases

- Storyboards *realize* the Use Case using a visual metaphor.
- Storyboards are defined using UML notations in the model.
- Depicts the *graphical user interface controls* (screens, buttons, etc.) and *system activities* that customers experience when they access an application function.
- Controls have associated metadata that refine its requirements (format, minimum input length, required/optional input field, etc.).
- Allows an analyst familiar with Use Cases and Visio to be productive with minimal training.
Collaborative Process

- Architects
- Create & Refine

- Business Analysts
- Create & Refine

- Use Cases
- Activity Model
- Component Model

- Storyboard Scenarios

- Test Cases
- Development Model
- Deployment Model

- Quality Assurance
- Platform Engineering
Storyboard Example
Support of Included Use Cases
Drill Down into Logic Detail
Easel - Extension to a UML Modeling Tool

- **Modeling palette**
  - Screen elements and controls implemented as a UML profile
  - A set of extensions to UML components used by the business analyst to define their requirements

- **Screenplay**
  - Clickable prototype of the Use Case based on the storyboard model
  - Quickly answers the question “Is this what you wanted?”

- **Scriptwriter**
  - Generates documentation automatically from the model
  - Document reflects all inputs and changes from all stakeholders

- **Test-drive**
  - Decision/activity branches in a storyboard yield test paths.
  - A test path is a distinct route through a storyboard.
  - Test paths are traceable back to use cases.
Wells Fargo UML Profile

- Contains assets needed to create the details of a use case
- Each component is based on an existing UML component
  - “Button click” is an extension of a “control flow”
- Additional data requirements have been defined
  - “Textbox” has tagged fields such as min/max length, format, optional/required, etc.
Easel Models are Organized into Views

- Packages contain
  - Use cases
  - Screens and associated GUI controls
  - Activities
  - Components
  - Test cases
  - Deployment nodes
  - Diagrams visualizing the relationships between elements

- Project artifacts are grouped by organization
  - Minimizes merge conflicts
  - Allows iterative development
  - Each team can leverage content from the others
Views Reflect Stakeholder Goals and Responsibilities

- **Business Analyst**
  - What is the business use case?
  - What capabilities should the customer have?
  - What must the system do?

- **Architect:**
  - What applications must collaborate to meet the requirements?
  - How do the applications need to be extended?
  - What patterns can be applied in the software design?
  - Are there reusable components?

- **Developer:**
  - What classes implement the patterns?

- **Quality Assurance:**
  - What functional paths should we test?

- **Platform Engineering:**
  - Where are the resource/performance constraints?
  - Other non-functional requirements?
Screenplay Example
Test Paths

- In complex storyboards, distinct paths often overlap before converging on a single end-user visible screen.
  - Example: A process with multiple system tests, any of which can result in an error screen.

- Test-drive groups these paths using **packages** to represent nodes whenever those nodes are part of a common end-point in 2+ paths
  - Packages represent nodes that appear in 2+ paths that are identical from that point to the terminal screen
  - **<<test path>>** use cases represent distinct paths or path segments (the unique prefix of an overlapping suffix)
Benefits of UML Requirements

- **Visual modeling**
  - “A picture is worth a 1000 words”
  - Easy identification of logic errors
- **Easy to relate flow sequencing to business logic and requirements**
- **Easy to collaborate with multiple disciplines; analyst, UI designer, architect and engineering**
- **Centralized model leveraged by all disciplines reducing rework throughout the lifecycle**
Status

- **Slow adoption**
  - Biggest barrier for the business unit analysts is learning how to write effective use cases

- **Screenplay well received**
  - Business unit analysts, product groups and the user experience teams
  - Easily answers the question “Is this what your really wanted?”

- **Scriptwriter essential to adoption**
  - Needed to provide the familiar documentation to ease the transition
  - Provides historical record of the project

- **Test-drive still in a learning mode**
  - Large amount of output for moderately complex projects
  - Points out how many alternative paths there are through the requirements that where not recognized before
Questions?