Requirements Management with Use Cases

Theory of Value

May 28, 2006

James Rivera, PMP

james@rivera-associates.com

www.rivera-associates.com

Overview

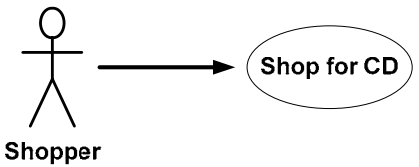
- Perspective
  - Technology Solution
  - Stakeholders
  - System Analyst, Project Manager, Product Manger
  - Eliciting Business Requirements

First Lets Define Use Cases

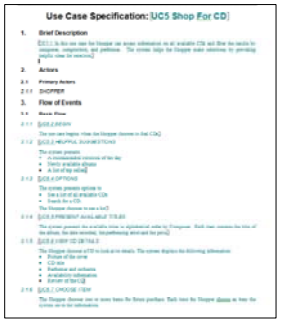
A use case describes a sequence of actions a system performs that yields an observable result of value to a particular actor

Use Cases tie the different UML models together by driving Architecture, Design, and Testing

Use Cases are Shown in UML Diagrams



Use Cases are Described in Text



Use Case Textural Standards

The following are RUP Standards

There is no Standard for writing Use case specifications

The UML does not specify how the text of a use case should be structured, organized, or written

Use Cases are Requirements

Use Cases are about specifying "Functional Requirements"

FURPS

- Functionality
- Usability
- Reliability
- Performance
- Supportability

Design Constraints

- Environments
- Compatibility
- Application Standards
- User Interface

Legal and Regulatory Requirements

- Federal Communication Commission
- Food and Drug Administration
- Department of Defense
- OSHA

Theory of Value

A generic term which encompasses all the theories within economics that explain the worth of goods and services.

What is Value?

- Value is not inherent in goods, or a property of them, but merely the importance that Stakeholders first attribute to the satisfaction of needs (requirements)
- In other words:
  - How important is it to have your needs satisfied? This defines the value

What has Value?

- Products and Services
  - Assigned by Stakeholder
- Resources
  - Required to attain value
  - Managed by project team
  - Observed and sometimes managed by stakeholder

Economizing Stakeholders

- Economizing:
  - “avoid waste or reducing expenditures to satisfy a requirement (need) for a product (good)”
- In other words:
  - Value is the importance that resources (project and product) attain because Stakeholders are conscious of being dependent on the control of them for the satisfaction of their needs.
- Stakeholders desire to control resources to ensure they get the needed value

Use Cases Revisited

A use case describes a sequence of actions a system performs that yields an observable result of value to a particular actor

- Therefore:
  - Use Case style ensures that the stakeholder can understand the sequence of actions performed to attain value
  - Use Case style ensures that the stakeholder can view the attainment of value as the end result

What is Quality?

- Quality is the degree to which a set of Inherent characteristics fulfill requirements
  - "American Society for Quality"
- "Man is the measure of all things"
  - Quality emerges as a relationship between man and his experience
  - "In other words; you must experience or observe something in order to define its quality"

Quality and Value

- Quality is an experience
- Quality is a measure of the fulfillment of needs
- Quality standards and baseline are defined by the stakeholder
- Quality is measured as a variance from baseline
- Quality is assigned to Process and Products

- Value is importance assigned to the satisfaction of needs
- Value is assigned by stakeholder
- Value is based on scarcity of goods and services
- Value is assigned to Process and Products

Wheel of Value

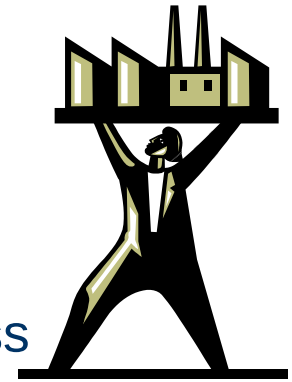
- Everyone has a role in delivering value
- One weak link and the delivery will not meet expectations

Do you know your Value to the Project?
Do you know how your value is aligned with the value of the deliverable?
Each Role has a stakeholder as well

Stakeholder

Represents an interest group whose needs must be satisfied by the project

- May be played by anyone who is (or potentially will be) materially affected by the outcome of the project.



- Required Skills:
  - Participate in the requirements capture process
  - Subject-matter expertise in the domain or the interest area
  - Ability to interpret and validate requirements

Requirements Analyst Role

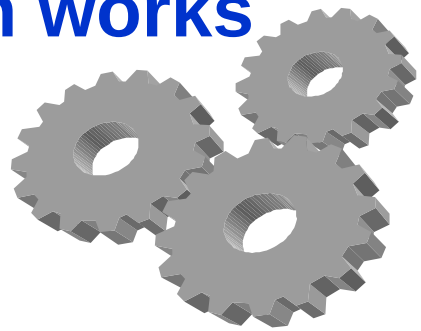
Manages the Requirements Capture Process

- Must analyze and organize informal requirement statements
- Develops requirement specifications in a form that can be verified by a user and used as input to design

- Required Skills
  - Facilitate Requirements Management process
  - Identify user needs and value
  - Create a complete, correct, consistent and unambiguous user requirement specification

Architect/Developer Role

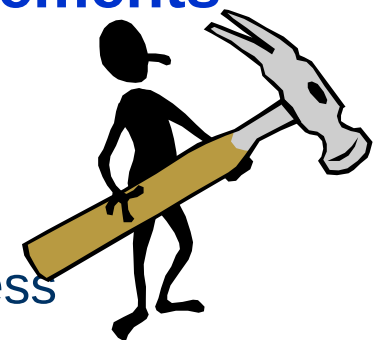
Determine how the system works



- Required Skills
  - Participate in requirements capture process
  - Interpret requirements
  - Define the architecture
  - Create, modify, manage, and test

Tester Role

Ensure that software meets all requirements



- Required Skills
  - Participate in requirements capture process
  - Verify requirements are testable
  - Interpret requirements
  - Create, manage, and run tests
  - Report results and verify fixes
  - Track Defects

Project Manager Role

Ensure that product is delivered to specifications

- Required Skills
  - Participate in requirements capture process
  - Owns schedule and management of deliverables
  - Interpret requirements
  - Reports progress to stakeholders

Theory of Value

Economics

- the theories within economics that explain the worth of goods and services

Intrinsic (objective) theories

- “man is the source of all things”
- the value of goods and services is a property built into the item itself

Subjective theories

- “man is the measure of all things”
- the value of goods and services is a judgment call of the person wanting those goods and services

“meaning-making”

- Items that are not consumption (purchased) and…
- are not production (creating and/or modifying something intended for sale)

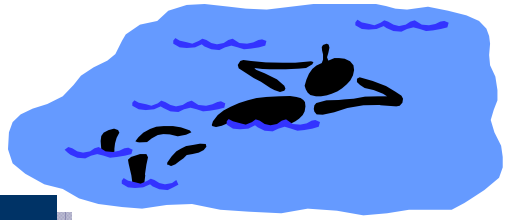
Nature and Origin of Value

- Classical - Labor Theory of Value
  - Inherent property of goods
  - Based on labor required to produce item
- “Theory of Value”
  - Is not inherent in goods
  - Is the importance we first attribute to the satisfaction of our needs

Paradox of Value

- Diamond-water paradox
  - Diamonds have **economic** value
  - Water has **use** value

- Which is of greater value?



The Actor Defines the Value

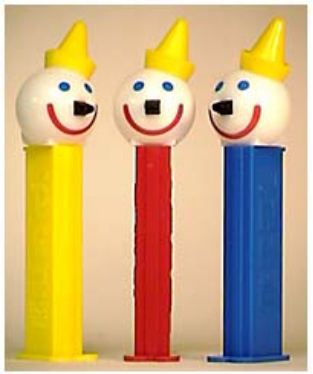
Use Cases and Value

- Use Cases deliver value to Actor
- Use Cases describe user experience
- Use Case addresses
  - Requirements
  - Priorities
  - Desired outcome.
- Stakeholders Define value for Actor
- Actors experience creates value
- Stakeholders value is defined by:
  - Meeting Needs
  - Level of Importance
  - Achieving Outcome

Theory of PEZ

1. Intrinsic $.20
2. Subjective $1.39
   - or more….
3. Meaning making
   - (PEZ collection & trading)
   - PEZ collection value is based on 1 & 2)

- If you sell the PEZ you are giving up 1,2, and 3 because the combination cannot ever be repeated

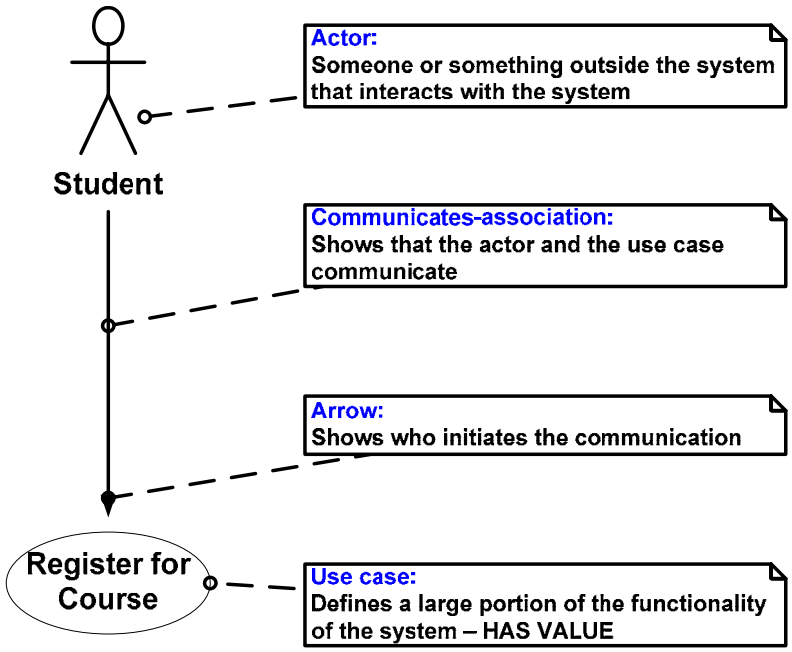


Use Case Diagrams

Visual Representation of Value

UML Symbols: Actors and Associations

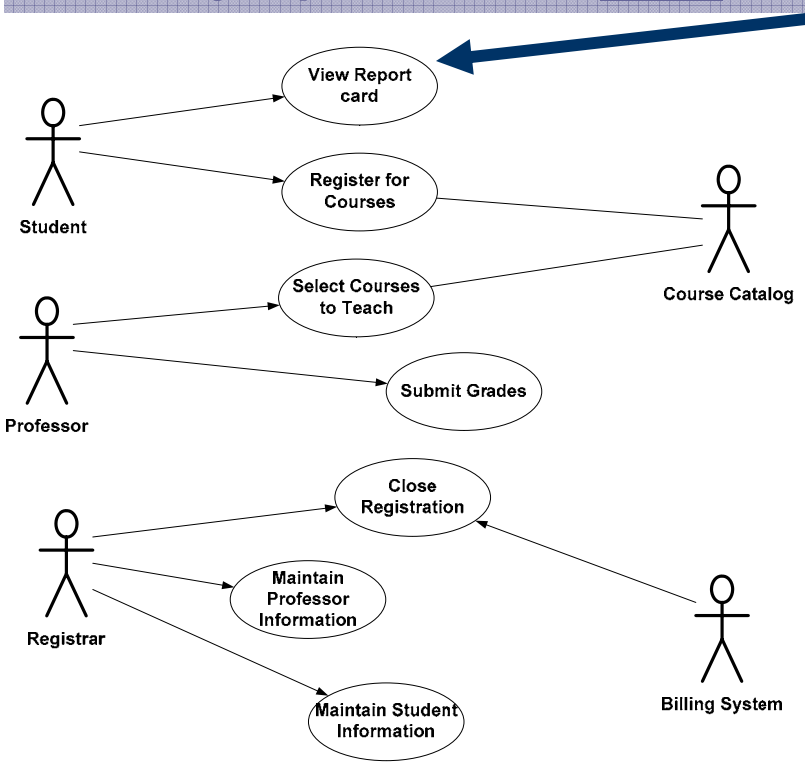
Use Case diagrams are not about data flow or control flow!



- **Actor:** Observes and defines the Value
- **Communicates Association**: Represents the use case being observed by the Actor
- **Use Case**: What the system is supposed to do: "The observable result of Value"
- Example of Actors as Human Beings

Use Case Diagrams

Visually represents the value the system provides to the Actor



- Provides a view of the entire system
- Sets boundaries: all actors are “outside the system”
- Represents both human and non-human actors
- Note: Actors do not communicate with Actors, Use Cases do not communicate with Use Cases

The Use Case Textural Specification

Descriptively Defining Value

Identifying Stakeholder Value

- Examining the Scope Statement
- Examine the Project Charter
- Examine the Vision Statement
- Check for:
  - Specifications
  - Design Constraints
  - Ability to identify value

4 Requirements for a Use Case

Improperly scripted Use Cases Make identifying Value difficult

- Must provide value to the stakeholder
  - Goal Oriented
- Must be a complete narrative describing how the value is provided
  - Must have Main and Alternative flows
  - Power of a use case is that it is “natural language”
- Must stand alone
  - No sequencing of use cases
- Must not describe design
  - “What” not “How”
  - How has no value!

Use Cases vs. Declarative Statements

Declarative

- The system shall provide a secure login.
- The system shall provide a list of class offerings for the current semester.
- The system shall only allow registration for courses where the prerequisites are fulfilled.
- The system shall provide a confirmation number when the schedule is submitted.

Use Case

- The student enters a student ID and password and the system validates the student.
- The system presents a list of course offerings. The students chooses up to four…
- The system displays the functions available to the student: create, modify, delete.
- The system validates the courses selected and displays a confirmation number…

The Contents of a Use Case

Use Case Name

1. Brief Description
2. Actors
3. Flow of Events
   i. Main Flow
   ii. Alternative Flows
      i. Alternative flow 1
      ii. Alternative flow 2
4. Special Requirements
   i. Business Rules
   ii. Usability Requirements
   iii. Data Definitions
5. Pre-conditions
6. Post-conditions

Special Requirements

- Non-functional requirements specific to a use case
- Keeps it close to the use case

Pre-condition

- Needs to be "true" for the use case to start
- Usually some "state" of the system

Post-conditions

- The required state of the system once the use case ends

  or…

- Some action the system must take once the use case ends
- Avoid conditions that are about the state of the "world"

The Contents of a Use Case

Use Case Name

1. Brief Description
2. Actors
3. Flow of Events
   i. Main Flow
   ii. Alternative Flows
      i. Alternative flow 1
      ii. Alternative flow 2
4. Special Requirements
   i. Business Rules
   ii. Usability Requirements
   iii. Data Definitions
5. Pre-conditions
6. Post-conditions

Main Flow

- **"Happy Path"**
- **"Path of least resistance"**
- **Should be only one Main Flow**

Alternative Flow

- **Regular "Variance"**
- **Odd cases**
- **Exceptional (Error) flows**

Common Mistakes

- **Multiple "Main Flows"**
  Difficult to see the goal being accomplished
- **No "Alternative Flows"**
  Not really a use case
  Exception may be a draft or early use cases

Use Case Styles

“Style” is how you structure the text of the use case and allows Stakeholder to observe the steps to attaining value

- Use Case Specifications can be considered “structured text”
- There are many different styles to choose from
- For a given project (or organization) it is vital to choose and be consistent with one style
  - For consistency
  - For readability
  - For usability by the development team

Use Case Style Considerations

- Do steps in the “Flows” have numbers or titles or both?
- Do “Alternative Flows” have numbers or titles or both?
- Does the “Main Flow” reference other flows or not?
- How to you reference one part of a use case from another?
- Can “Flows” have embedded flows?
- How do “Alternative flows” tell what happens when they are done?

The style is critical to the understanding of the use case flow

Main (Basic) Flow of Events

Main flow shows the actor succeeding in his/her goal

Use Case Specification: Register for Courses

Brief Description

This use case allows a Student to register for course offerings in the current semester. The student can also modify or delete course selections if changes are made within the add/drop period at the beginning of the semester. The Course Catalog System provides a list of al the course offerings for the current semester.

Actors

1. ***Primary Actor – Student***
2. ***Secondary Actor – Course Catalog System***

Flow of Events

1. Basic Flow

1.1 LOGON

This use case starts when a student accesses the Course Registration System. The student enters a student ID and password and the system validates the student.

1.2 CREATE SCHEDULE

The system displays the functions available to the student. These functions are: Create a Schedule, Modify a Schedule, and Delete a Schedule. The student selects "Create a Schedule."

1.3 SELECT COURSES

The system retrieves a list of available course offerings from the Course Catalog System and displays the list to the student. The Student selects up to 4 primary course offerings and 2 alternative course offerings from the list of available offerings. The student can add and delete courses as desired until choosing to submit the schedule.

1.4 SUBMIT SCHEDULE

The student indicates that the schedule is complete. The system validates the courses selected and displays the schedule to the student. The system displays the confirmation number for the schedule. The system saves the student's schedule information. The uses case ends.

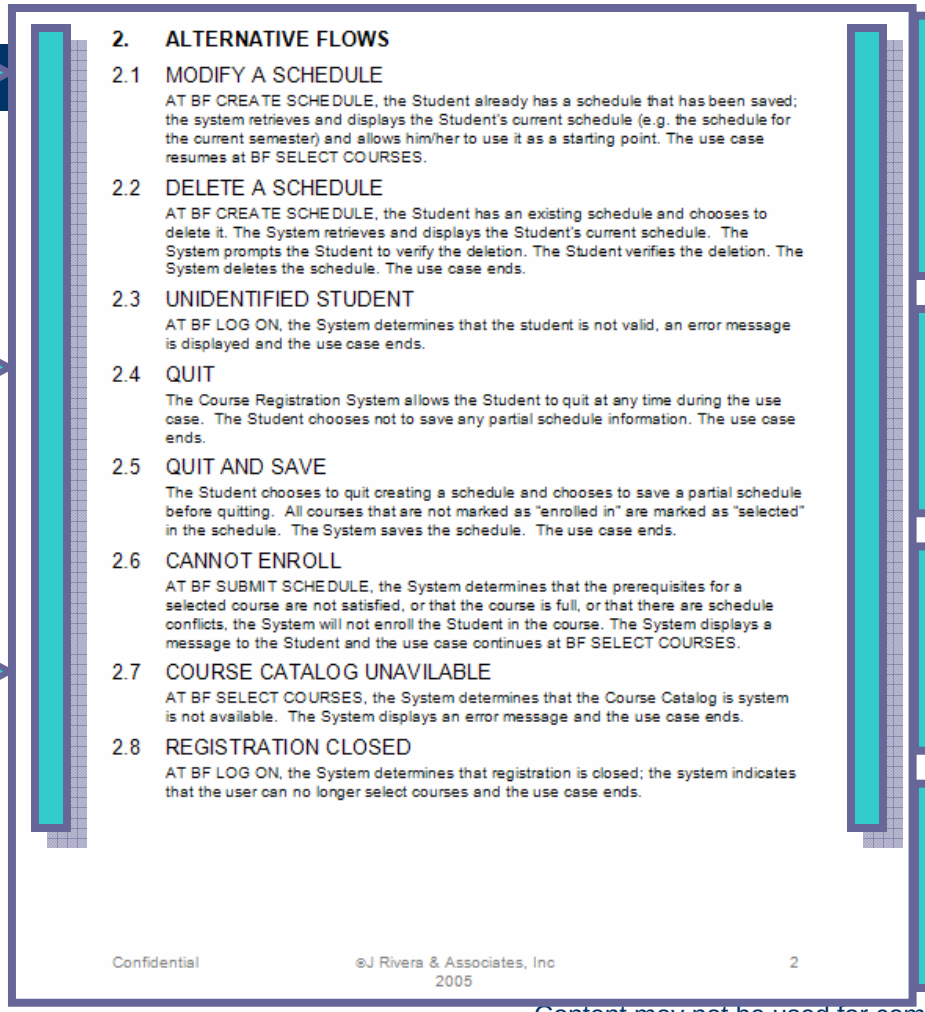
- Structure the flow into steps
- Number and title each step
- Describe steps in 1 to 3 sentences
- Don't refer to alternative flows in the main flow

Alternative Flows of Events

Alternative flows are flat

The can have steps

They have names



2. ALTERNATIVE FLOWS

2.1 MODIFY A SCHEDULE

AT BF CREATE SCHEDULE, the Student already has a schedule that has been saved; the system retrieves and displays the Student's current schedule (e.g. the schedule for the current semester) and allows him/her to use it as a starting point. The use case resumes at BF SELECT COURSES.

2.2 DELETE A SCHEDULE

AT BF CREATE SCHEDULE, the Student has an existing schedule and chooses to delete it. The System retrieves and displays the Student's current schedule. The System prompts the Student to verify the deletion. The Student verifies the deletion. The System deletes the schedule. The use case ends.

2.3 UNIDENTIFIED STUDENT

AT BF LOG ON, the System determines that the student is not valid, an error message is displayed and the use case ends.

2.4 QUIT

The Course Registration System allows the Student to quit at any time during the use case. The Student chooses not to save any partial schedule information. The use case ends.

2.5 QUIT AND SAVE

The Student chooses to quit creating a schedule and chooses to save a partial schedule before quitting. All courses that are not marked as "enrolled in" are marked as "selected" in the schedule. The System saves the schedule. The use case ends.

2.6 CANNOT ENROLL

AT BF SUBMIT SCHEDULE, the System determines that the prerequisites for a selected course are not satisfied, or that the course is full, or that there are schedule conflicts, the System will not enroll the Student in the course. The System displays a message to the Student and the use case continues at BF SELECT COURSES.

2.7 COURSE CATALOG UNAVILABLE

AT BF SELECT COURSES, the System determines that the Course Catalog is system is not available. The System displays an error message and the use case ends.

2.8 REGISTRATION CLOSED

AT BF LOG ON, the System determines that registration is closed; the system indicates that the user can no longer select courses and the use case ends.

1) Start:

Indicate the starting point. May be basic flow or alternative flow

2) Cause:

Say what causes the flow to start

3) Effect:

Say what happens as a result of the cause

4) End:

Say where the flow resumes or if it ends

Grammar and Value

- How something is written speaks to value
- Being direct, wordiness, tense (past, future, present), perspective
- How to read a scope or vision statement and detect well understood value and where there is lack of alignment or misalignment.

Use Case Writing Techniques

- Using if-statements (or not)
- Making choices
- Showing iteration
- Sequence of events
- Correct level of detail
- Creativity
- Supplementary Specifications

Summary Slide

- Use Cases deliver Value
- Use Cases demonstrate the steps to deliver value
- Actors/Stakeholders Define Value
- Every project member has a role in delivering the value
- Use Case Styles help to make the value easy to understand
- Understand the project value from all perspectives
- Understand your value to the project and organization
- Focus on defining value and not on developing documentation