In Search of Excellent Requirements

Neil Potter, Mary Sakry
The Process Group
help@processgroup.com
www.processgroup.com

Licensed from Karl E. Wiegers
Have You Had Any of These Experiences?

- The project’s vision and scope are **never clearly defined**.

- **Customers are too busy** to spend time working with developers on the requirements.

- User surrogates (managers or marketing) **claim to speak for the users**, but they really don’t.

- Users claim that all the requirements are critical and **do not prioritize them**.

- Developers encounter **ambiguities and missing information** and they guess.
Have You Had Any of These Experiences?

- Your customers sign off on the requirements then **change them continuously**.
- The scope increases as **requirements changes** are accepted, but the **schedule slips** because more resources are not provided.
- Requested **requirements changes get lost** and the status of a change request is not known.
- Functionality is requested and built, **but never used**.
- The **specification is satisfied**, but the **customer is not**.
Agenda

- Introduction

- Requirements Development and Management Process
  - ✓ Identify users
  - ✓ Define vision and scope
  - ✓ Understand user needs
  - ✓ Derive functional requirements
  - ✓ Analyze and verifying requirements
  - ✓ Manage requirements changes

Further resources at: [http://www.processgroup.com/downloads.htm](http://www.processgroup.com/downloads.htm)
Introduction
Three Levels of Software Requirements

#1: Business Requirements

Vision and Scope Document

#2: User Requirements

Use Case Document

#3: Functional Requirements

Software Requirements Specification (SRS)

- Quality Attributes
- Business Rules
- Other Nonfunctional Requirements
- External Interfaces
- Constraints
Benefits of Requirements Processes

- Creating and **reviewing multiple views** of requirements helps **satisfy customer expectations**.

- **Controlling** requirements changes **minimizes the adverse impact of changes**.

- Emphasizing **quality requirements reduces rework, maintenance** and **wasted time** implementing unnecessary functions.

- **System testing** can be **based on requirements**.
Relative Cost of Fixing a Defect

Development Phase

[Robert Grady, Applications of Software Measurement Conference, 1999]
The Requirements Job

- Learn about the **business, its language**, and the objectives of the solution.
- **Sort out requirements** information and structure into an SRS.
- Help identify and remove requirements **ambiguities**.
- Help establish requirements **priorities**.
- **Clarify scope** and context of solution with other systems.
- Educate users / customers on how to use / **read the requirements**.
- Assist in decisions regarding requirements changes and **tradeoffs**.
Your Expectations

- Requirements problems / issues?
- Expectations for this class?
Requirements Development & Management Process

Start → Identify Users → Define Vision & Scope → Understand User Needs → Derive Functional Requirements → Analyze & Verify Requirements → Manage Requirements Changes

Identify Users

Start

Identify Users

Define Vision & Scope

Understand User Needs

Derive Functional Requirements

Analyze & Verify Requirements

Manage Requirements Changes

User Classes

- **Distinct groups of users** for a product
  - Could include other systems as *users* of your system

- May differ in:
  - frequency of use of application
  - functions used
  - tasks to be accomplished
  - education and skill level
  - privilege or security level

- **Identify user classes** and their characteristics early

- **Document** user classes in the SRS

- **Not all** user classes are equally important to you
User Class Example

- Scheduling / billing system User Classes and tasks:
  - Office assistant:
    - Daily patient scheduling, patient billing
  - Office manager:
    - Monthly reporting, complaint management
  - System administrator:
    - Database maintenance and recovery
  - Doctor:
    - Schedule summary, scheduling strategy changes (#minutes per patient, blackout days, on-call doctor availability, emergency patient information access, remote data access)

A lack of understanding of your user classes can lead to major omissions in functionality and reduced customer satisfaction.
The Product Champion

- **Primary interface** between development and customer communities
  - Represents a user class or system interface
- **Respected** members of the user community
- **A real user**; not a manager, sponsor, or simulated user
- Has a **vision** of what the product should be and do
- **Reconciles incompatible** customer **requirements**
- Must be **empowered** to make binding decisions

Potential Product Champion Activities

Planning:
- Define **scope and limitations** of system
- Define external **boundaries** and **interfaces**
- Plan **migration pathway** from current system to new one

Requirements:
- Interview other users they represent
- Resolve conflicting requirements
- Set implementation **priorities**
- Define **quality attributes**
- Participate in requirements **inspections (reviews)**

Change Management:
- Evaluate / prioritize bug fixes and enhancements

Documentation:
- Contribute to user manuals and on-line help
- Help prepare classes and tutorial materials

Advocacy:
- Help “sell” the system to customer communities
Define Vision and Scope

Start

Identify Users

Define Vision & Scope

Understand User Needs

Derive Functional Requirements

Analyze & Verify Requirements

Manage Requirements Changes

Manage Requirements Changes

Three Levels of Software Requirements

1. Business Requirements
   - Vision and Scope Document

2. User Requirements
   - Use Case Document

3. Functional Requirements
   - Software Requirements Specification (SRS)

Define Vision & Scope
- Business Rules
- Other Nonfunctional Requirements
- External Interfaces
- Constraints
1. Business Requirements
   1.1. Background
   1.2. Business Opportunity
   1.3. Business Objectives and Success Criteria
   1.4. Customer or Market Needs
   1.5. Business Risks

2. Vision of the Solution
   2.1. Vision Statement
   2.2. Major Features
   2.3. Assumptions and Dependencies

3. Scope and Limitations
   3.1. Scope of Initial Release
   3.2. Scope of Subsequent Releases
   3.3. Limitations and Exclusions

4. Business Context
   4.1. Customer Profiles
   4.2. Project Priorities
   4.3. Operating Environment
Some Elicitation Questions: Business Reqs.

- What business **problem** are you trying to solve?
- What’s the **motivation** for solving this problem?
- What would a highly successful solution **do** for you?
- How can we judge the **success** of the solution?
- What’s a successful solution **worth**?
- Which **business activities** and **events** should be included in the solution? Which should **not**?
1. Business Requirements

1.1. Background
1.2. Business Opportunity
1.3. Business Objectives and Success Criteria
1.4. Customer or Market Needs
1.5. Business Risks

1. Allow customers world wide to make and track purchases for all consumable products using a web browser.
2. Allow payment to be made electronically using existing or new accounts.
3. Provide customer with Internet access to accounts payable and receivable functions on existing accounts.
4. Capture needs profile of each customer for future marketing use.
5. ...

- Eliminate paper-based orders from world-wide customers by providing a web system for order management.
- Replace all paper transactions by 200X.
- Reduce order management costs 30% by 200Y.
## Another Example - Customer Needs

<table>
<thead>
<tr>
<th>Req #</th>
<th>CURRENT ACCOUNTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA 1.1</td>
<td>A/C to A/C transfer (XXX accounts)</td>
</tr>
<tr>
<td>CA 1.2</td>
<td>A/C to A/C transfer (to other local banks)</td>
</tr>
<tr>
<td>CA 1.3</td>
<td>Transfer from XXX A/C to a Beneficiary through a XXX Branch</td>
</tr>
<tr>
<td>CA 1.4</td>
<td>A/C to A/C transfer (to international banks thru pre-defined transfer)</td>
</tr>
<tr>
<td>CA 1.5</td>
<td>Utility bill payments</td>
</tr>
<tr>
<td>CA 1.6</td>
<td>Request additional XXX ATM card</td>
</tr>
<tr>
<td>CA 1.7</td>
<td>Report lost/stolen XXX ATM card with option to request a replacement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Req #</th>
<th>CREDIT CARD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC 1.1</td>
<td>Apply for a XXX VISA / MasterCard</td>
</tr>
<tr>
<td>CC 1.2</td>
<td>XXX VISA/MasterCard credit card payment with the option to pay for another XXX credit card</td>
</tr>
<tr>
<td>CC 1.3</td>
<td>Transfer from XXX VISA / MasterCard to Current A/C</td>
</tr>
<tr>
<td>CC 1.4</td>
<td>Change of XXX VISA / MasterCard address</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Req #</th>
<th>TIME DEPOSIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD 1.1</td>
<td>Book a Time Deposit from a current account</td>
</tr>
<tr>
<td>TD 1.2</td>
<td>Commission Rate Inquiry/Display Rates</td>
</tr>
<tr>
<td>TD 1.3</td>
<td>Change interest account assignment</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Req #</th>
<th>PERSONAL FINANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF 1.1</td>
<td>Apply for Consumer Loan</td>
</tr>
<tr>
<td>PF 1.2</td>
<td>Inquiry on loan outstanding amount, installment amount, due amount</td>
</tr>
<tr>
<td>PF 1.3</td>
<td>View loan statement</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Req #</th>
<th>FINANCIAL PLANNING</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP 1.1</td>
<td>Plan Details</td>
</tr>
<tr>
<td>FP 1.2</td>
<td>Payment Details</td>
</tr>
<tr>
<td>FP 1.3</td>
<td>Investment Details</td>
</tr>
<tr>
<td>FP 1.4</td>
<td>Benefit Details</td>
</tr>
</tbody>
</table>

2. Vision of the Solution

2.1. Vision Statement

2.2. Major Features

2.3. Assumptions

Context Diagram

- The **system name** goes in the circle
- Outside boxes represent major external entities
- Flows between externals and the system comprehend high-level user needs
Exercise: Business Requirements

- Write 1-2 business objectives for one of your projects
- Write 1-2 business requirements for one of your projects:
  - Focus on Customer or Market Needs
    - Include the needs of a typical customer and the needs that are not being met
    - Include problems that customers are currently having that the product will address
Understand User Needs

- Identify Users
- Define Vision & Scope
- Understand User Needs
- Derive Functional Requirements
- Analyze & Verify Requirements
- Manage Requirements Changes

Start
Three Levels of Software Requirements

#1: Business Requirements

Vision and Scope Document

#2: User Requirements

Use Case Document

System Requirements

#3: Functional Requirements

Software Requirements Specification (SRS)

Define Vision & Scope

Business Rules

Other Nonfunctional Requirements

External Interfaces

Constraints
The Need for Customer Involvement

Customer involvement is a critical factor in software quality.

What the customer wants
What the developer builds

Expectation Gap
Surprise!

Time
Incremental Requirements Definition

Don’t necessarily clarify all user needs for the whole system initially. Consider focusing on areas that are the:

- Highest priority
- Most frequently used
- Highest revenue
- Largest user class
- Core features to support architecture
- Reqs. for regulatory compliance
Sources of Software Requirements - 1

- **Documents**
  - ✓ **descriptions** of current or competing products
  - ✓ **standards** or regulations
  - ✓ help desk **problem reports**

- **Interviews**
  - ✓ **focus groups** of representative customers
  - ✓ **use case workshops** with users + developers
  - ✓ **product champions** as key customer representatives

- **Questionnaires and marketing surveys**
  - ✓ good for **large number** of responses
  - ✓ ask the **right people the right questions**!
  - ✓ run a **pilot** first
Sources of Software Requirements - 2

- Watch users do their jobs
  - create *workflow* diagrams
  - "day in the life" studies
  - look at *what information* the user has *when* performing a particular task

- Prototyping
  - useful for requirements, design, and implementation

- Task analysis
  - apply methods like *use cases or scenarios*
  - *focus on user objectives*, rather than system functions

- Event-response tables
  - identify external *stimuli* and describe system *responses*
Classifying the Voice of the Customer
Use Cases
Gathering Requirements Through Use Cases

- Use cases came from object-oriented world, extended for system analysis and user interface design
- Provides a method to **capture user requirements**
- **Focus on actual**, but abstracted ("essential") **usage scenarios**
- **Ask** users:
  - “Describe a task (or goal) you **need** to perform.”
  - not:
    - “What do you **want** this system to do?”
- **Explore** **sequences of user actions** and system responses
- **Derive functional requirements** and test cases from use cases
Use Case in Context

- Use Case Workshops
- Use Case Descriptions
- Draft SRS
- Draft Test Cases
- Draft Models
- Business Rules
- Verified SRS
- Verified Models
- Shared Vision
Use Case Workshop Approach

Raw Interview Data

<table>
<thead>
<tr>
<th>Interview Data</th>
<th>Category</th>
<th>Interview Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search a document collection for a specific template for a specific type of project</td>
<td>Use Case</td>
<td>Operator</td>
</tr>
<tr>
<td>A purchasing agent can modify an order only with written permission from the customer</td>
<td>Business Rule</td>
<td>Customer Manager</td>
</tr>
</tbody>
</table>

Use Case Name: View an order stored in the database.

User Classes: all

Preconditions: database contains orders, user identity is verified

Postconditions: order has been shown

<table>
<thead>
<tr>
<th>Actor Actions</th>
<th>System Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>user enters order number he wants to view</td>
<td>order details are displayed</td>
</tr>
<tr>
<td>user enters order number, but it doesn’t exist</td>
<td>error message: order number not found</td>
</tr>
<tr>
<td>etc. for all normal and exception pathways</td>
<td></td>
</tr>
</tbody>
</table>
Benefits from the Use Case Approach

- User-centric: user’s terminology is applied
- Task-centric: reveals requirements to get work done
- Helps analysts understand application domain
- Avoids building unnecessary functionality
- Permits early drafting of functional test cases
- Helps set implementation priorities on functional requirements (i.e., which is the most important user task?)
- Permits tracing requirements back to voice of the customer
Examples of Use Case Statements

- Order a bottle of a specific chemical from a specific vendor.
- Search a document collection for a specific template.
- See which flight has the lowest fare from city A to city B on a specified date.
  - ✓ with an advance ticket purchase
  - ✓ without an advance ticket purchase
- Pay for a purchase made at a commercial website.
- Output data every 3 milliseconds in JJU format.
- Update interest rate based on loan database update.
Name: Withdraw Cash
Actor: Account Owner
Description:
The user withdraws a specific amount of cash from a specified account.
Trigger: Account Owner selects Withdrawal action.
Preconditions:
1. The Account Owner is logged in to the ATM.
2. The Account Owner has at least 1 account with a positive balance.
3. The ATM contains cash.
Postconditions:
1. The requested amount of cash has been dispensed.
2. The account balance is reduced by the amount withdrawn plus any fees.
3. The ATM cash balance is reduced by the amount withdrawn.
Priority: High
Sample Use Case for an ATM - 2

■ Normal flow:

<table>
<thead>
<tr>
<th>Actor Actions</th>
<th>System Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Select Withdrawal action.</td>
<td>2. Display user’s accounts.</td>
</tr>
<tr>
<td>3. Select desired account.</td>
<td>4. Ask user to enter amount.</td>
</tr>
<tr>
<td>5. Enter desired amount.</td>
<td>6. If amount is okay, dispense cash and debit account.</td>
</tr>
<tr>
<td>7. Remove cash from dispenser.</td>
<td></td>
</tr>
</tbody>
</table>

■ Alternative Flow:

✓ Step 4: display list of common amounts, let user select one

■ Exceptions:

✓ Step 6: amount is not a multiple of $20.00
✓ Step 6: amount exceeds $400
✓ Step 6: amount exceeds account balance
✓ Step 6: amount exceeds cash available in ATM
Use Case Template

<table>
<thead>
<tr>
<th>Use Case ID:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name:</td>
</tr>
<tr>
<td>Created By:</td>
</tr>
<tr>
<td>Date Created:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actors:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
</tr>
<tr>
<td>Trigger:</td>
</tr>
<tr>
<td>Preconditions:</td>
</tr>
<tr>
<td>Postconditions:</td>
</tr>
<tr>
<td>Normal Flow:</td>
</tr>
<tr>
<td>Alternative Flows:</td>
</tr>
<tr>
<td>Exceptions:</td>
</tr>
<tr>
<td>Includes:</td>
</tr>
<tr>
<td>Priority:</td>
</tr>
<tr>
<td>Frequency of Use:</td>
</tr>
<tr>
<td>Business Rules:</td>
</tr>
<tr>
<td>Special Requirements:</td>
</tr>
<tr>
<td>Assumptions:</td>
</tr>
<tr>
<td>Notes and Issues:</td>
</tr>
</tbody>
</table>

- **Actor:** a human user or other system that provides the trigger
- **Trigger:** event that causes your system to respond
- **Pre/post conditions:** bounds the use case to avoid major design flaws
- **Alternative flow:** alternative route to accomplish Use Case
- **Exceptions:** describes how the system reacts to bad input
Business Requirements
Eliminate paper-based orders from world-wide customers by providing a web system for order management. Replace all paper transactions by 200X. Reduce order management costs 30% by 200Y.

1) Allow customers world wide to make and track purchases for all consumable products using a web browser.
2) Allow payment to be made electronically using existing or new accounts.
3) Provide customer with Internet access to accounts payable and receivable functions on existing accounts.
4) Capture needs profile of each customer for future marketing use.
5) Provide customizable reporting of system use to executive management (reporting to be defined).
6) Provide simple transaction reporting summary to customers (reporting to be defined).
7) Support sales force in creating and managing Internet accounts for customers.
8) Eliminate all lost sales due to incorrectly routed paper orders.
9) Provide audit / verification mechanism to verify orders received and shipped.
Example Use Cases for Business Requirement #7.

1. Register a site admin – by site admin.
2. Create a new site admin profile – by site admin.
3. Modify site admin profile – by site admin.
4. Remove the site admin profile – by site admin.
5. Register a new user - by site admin.
6. Remove an existing user - by site admin.
7. View own profile - by user.
8. Modify own profile - by user.

Obtain a list of use case names (actor tasks) before expanding each use case.
## Example Use Case #2

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>US2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Create a new site admin profile.</td>
</tr>
<tr>
<td>Created By</td>
<td></td>
</tr>
<tr>
<td>Date Created</td>
<td></td>
</tr>
<tr>
<td>Last Updated By</td>
<td></td>
</tr>
<tr>
<td>Date Last Updated</td>
<td></td>
</tr>
<tr>
<td>Actor</td>
<td>Sales Administrator.</td>
</tr>
<tr>
<td>Description</td>
<td>Create a site admin profile for a customer. The site admin can then create users.</td>
</tr>
<tr>
<td>Trigger</td>
<td>Select &lt;new site admin profile&gt;</td>
</tr>
<tr>
<td>Preconditions</td>
<td>Site admin has a valid license ID to enter. Site admin will have a list of user membership names &amp; email addresses to enter for this new profile.</td>
</tr>
<tr>
<td>Postconditions</td>
<td>A new site admin ID will be created for the customer. The site admin can then access this profile and delegate management of the profile to users under him/her.</td>
</tr>
</tbody>
</table>
## Example Use Case #2  - slide 2/3

<table>
<thead>
<tr>
<th>Normal Flow—User Actions</th>
<th>Normal Flow—System Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The site admin will log in with new license ID.</td>
<td>2. Display screen to create the new profile.</td>
</tr>
<tr>
<td>3. Enter the user ID.</td>
<td>4. Check if member profile exists.</td>
</tr>
<tr>
<td>5. The admin can set a flag to specify an “admin only” user, where he/she can administer other users – but cannot do any transactions.</td>
<td>• (If the member profile does not exist then the user must first register at the <a href="http://www.xxx.com">www.xxx.com</a> site (to create a member profile), and then only the Profile can be created.)</td>
</tr>
<tr>
<td>6. Enter user membership names and email addresses for new profile.</td>
<td>7. Insert the site admin profile in the database.</td>
</tr>
<tr>
<td>9. Log off.</td>
<td>8. Display confirmation that new profile has been created. Email copy of profile information to site admin.</td>
</tr>
</tbody>
</table>
### Example Use Case #2

<table>
<thead>
<tr>
<th>Alternative Flow - User Actions</th>
<th>Alternative Flow - System Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>The admin may assign user membership names to the newly created site admin profile using the ABC toolkit.</td>
<td>None.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exceptions - User Actions</th>
<th>Exceptions - System Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1. If the site admin profile already exists.</td>
<td>A2. Site admin not added. The site admin is prompted to use the modify option.</td>
</tr>
<tr>
<td>B1. If the user or site admin that is to be added is not in the <a href="http://www.xxx.com">www.xxx.com</a> membership accounts list.</td>
<td>B2. Site admin not added. An error will be displayed and the user will have to first register at the <a href="http://www.xxx.com">www.xxx.com</a> site, before his/her profile can be created.</td>
</tr>
<tr>
<td></td>
<td>B3. Open new browser window for <a href="http://www.xxx.com">www.xxx.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Business Rule:</th>
<th>BR1: Only allow membership names from approved counties (see approved-list.doc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Requirements</td>
<td>The site admin should be able to maintain his/her own profile and should be able to add more users under him/her.</td>
</tr>
<tr>
<td>Assumptions</td>
<td>Site admin must use Internet Explorer 5.X or greater. All existing site admin profiles remain unaltered.</td>
</tr>
<tr>
<td>Notes and Issues</td>
<td>Resolutions</td>
</tr>
<tr>
<td>Can an admin add another admin? Will there be more than one admin who will be adding/modifying the same profile?</td>
<td>Yes, an admin can create another admin and more than one admin can modify the same profile.</td>
</tr>
</tbody>
</table>
Exercise: Use Cases

- Elicit several proposed use case names for your system.

**Example Use Cases for Business Requirement #7.**

1. Register a site admin – by site admin.
2. Create a new site admin profile – by site admin.
3. Modify site admin profile – by site admin.
4. Remove the site admin profile – by site admin.
5. Register a new user - by site admin.
6. Remove an existing user - by site admin.
7. View own profile - by user.
8. Modify own profile - by user.
Derive Functional Requirements

Start → Identify Users

- Manage Requirements Changes
- Define Vision & Scope
- Understand User Needs
- Analyze & Verify Requirements
- Derive Functional Requirements

Three Levels of Software Requirements

1: Business Requirements

Vision and Scope Document

2: User Requirements

Use Case Document

3: Functional Requirements

Software Requirements Specification (SRS)

Derive Functional Requirements

Business Rules

Other Nonfunctional Requirements

External Interfaces

Constraints

Two common styles of defining Functional Requirements:

1: Functional requirements = complete rewrite of the user requirements from a system / developer perspective, e.g.,
- Feature A: <description, use case(s) it implements, characteristics>
- Feature B: <description, use case(s) it implements, characteristics>

2: Functional requirements = ADDITIONAL software characteristics that more completely define system behavior, at the level that designers can design without causing significant rework and defects, e.g.,
- Ensure record locking for database X
- Don’t allow control characters in the password
- Process empty records by flagging them as null

You might decide that some of the functional requirements are more appropriately recorded as user requirements.
Business Requirement 7:
Support sales force in creating and managing Internet accounts for customers.

Use Case 2: Create a new site admin profile.

Functional Requirements for Use Case 2:

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>No system use will be possible if the license is invalid. The site admin will be asked for a</td>
</tr>
<tr>
<td>license renewal number annually.</td>
</tr>
<tr>
<td>The system will remember partial profile creations, should the site admin not complete the</td>
</tr>
<tr>
<td>profile creation process at one time. The site admin will be able to continue where he/she</td>
</tr>
<tr>
<td>left off.</td>
</tr>
<tr>
<td>The admin tool will not provide any record locking and will depend on the default record</td>
</tr>
<tr>
<td>locking provided by the database.</td>
</tr>
<tr>
<td>If a user profile is being modified simultaneously by 2 site admins, then the profile in the</td>
</tr>
<tr>
<td>database will be the profile that was last updated.</td>
</tr>
<tr>
<td>There will be only one site admin per user.</td>
</tr>
<tr>
<td>User membership names and email addresses shall be typed in or imported from a delimited</td>
</tr>
<tr>
<td>file. The site admin can specify the delimiters used in the ascii file.</td>
</tr>
</tbody>
</table>
Exercise: Writing Functional Requirements, System Requirements, External Interfaces and Constraints

- Select one use case. Write several functional requirements that, if implemented, would allow the user to perform that use case.

  ✓ Assume: Functional requirements = ADDITIONAL software characteristics that more completely define system behavior.

- Identify System Requirements, External Interfaces and Constraints for a section of your project.
Q&A


