

TOOLS FOR REQUIREMENTS GATHERING SESSIONS

Techniques to Trigger Thoughts

Use various tools as a starting point in requirements gathering sessions as opposed to starting from a blank slate.

List of Questions - Prepare a list of questions ahead of time to use as a general guide for the session.

Use Cases - Use cases describe the system from the point of view of the user using the system. They provide an easy format for all people to quickly grasp the system's functionality. Draft a use case to work from.

Existing System - When working with an existing system, use it to trigger ideas quickly. Have the user walk through how they do the task now in the system. Talk about what users do and do not like about the system. Look at screen shots if you do not have the application handy.

Whiteboard - Because visualizing the system or UI improves comprehension for many people, always use a whiteboard to sketch out ideas. Capture use cases, sketch out user interfaces or draw process flows on the whiteboard.

Screen Mockups - For applications with user interfaces, start with mockups of the UI. Wire frames are simple black and white boxes and text, specifically not focused on look and feel. Use paper, PowerPoint, or a whiteboard to draw the UIs.

Questions to Ask When Developing Use Cases

Description

- ▶ What is the actor's goal?
- ▶ What are the high level actions the actor will need to take to reach that goal?

Frequency of Use

- ▶ How many times per minute, hour, or day will this Use Case be executed?

Actor

- ▶ Who uses the system (what is their job)?
- ▶ What other systems will interact with this system?
- ▶ Who or what provides information to the system?
- ▶ Who or what receives information from the system?

Trigger

- ▶ What event causes this Use Case to happen?
- ▶ What actor initiates this Use Case?

Preconditions

- ▶ What conditions must be true before this Use Case can begin?
- ▶ What state is the system in before this Use Case can begin?

Post Conditions

- ▶ What conditions must be true when the Use Case ends?
- ▶ What state will the system be in at the end of the Use Case?

Main Course

- ▶ How does the actor interact with the system?
- ▶ What does the system do at this step (present options, display data, execute process)?
- ▶ What does the system do next?
- ▶ What does the actor do next?
- ▶ Is there part of this Use Case that is another Use Case called by multiple other Use Cases?

Alternative Courses

- ▶ If X doesn't happen, what should happen?
- ▶ What other possible actions can the user take at each step?

Exceptions

- ▶ What possible error conditions exist at each step of the main course?
- ▶ What are the interrupts that can happen at any time?
- ▶ If the user cancels out at any step, what should happen?

Preparing for Facilitated Sessions

Prepare a list of questions

- ▶ Allows you to start out on a strong note to get the group thinking
- ▶ You don't have to follow the questions as a script, but make sure to check the list at the end of the session to make sure you got everything

Determine goals & agenda in advance

- ▶ Define the goals and scope of the meeting based on an overall plan of sessions
- ▶ Prepare an agenda and send it out to everyone to give attendees context ahead of the meeting
- ▶ Example goals for meeting:
 - ▶ Complete the last bits of missing detail
 - ▶ Define a workflow
 - ▶ Determine business rules
- ▶ Invite the appropriate people
 - ▶ Communicate session times well in advance
 - ▶ Sessions are most effective with no more than 6-10 people
 - ▶ Limit sessions to no more than 2 hours in length

Tips for Reviewing Requirements

- ▶ Factor in time differences
 - ▶ Review cycles may take several days
- ▶ Maintain version control of documentation
 - ▶ Ensure that everyone is reviewing the same requirements
 - ▶ Use a requirements management tool or document management tool (like SharePoint)
- ▶ Be aware of cultural differences
 - ▶ Some people may hesitate to disagree
- ▶ Document identified issues
 - ▶ In most cases, the original author is the best person to make changes to the requirements

REQUIREMENT TYPES AND EXAMPLES

Business Objective - A measureable target that specifies when the business problem has been solved. *[Increase revenue from the 20-40 year old professional demographics by 25%]*

Feature - A short-form description of an area of functionality that the solution will ultimately include to meet the business objectives. *[Quick re-generation for one passenger, Updateable profiles]*

Functional Requirement - A behavior or capability that the solution can provide irrespective of any qualifiers. *[The system shall allow the user to search for guest itinerary by Guest Last Name, Guest First Name, and/or Confirmation Number.]*

Availability - Desired "up time" during which the system and data are available for use. *[The system shall be available between the hours of 6AM and 10PM ship time, inclusive, every day of the week.]*

Design and Implementation Constraints - Restrictions on the options available to the development team. *[The system shall use an Oracle database to store data.]*

Documentation - Descriptions of any expected supplemental information, including its purpose, desired contents, level of detail, and formatting. *[The system shall provide context sensitive help that takes the user to a help topic specific to the screen in focus which describes how to use each control and data field on the screen.]*

Emotional - Describe the user's feelings about the experience with the system, including where and when the emotions should be felt, and how they vary over time. *[The system shall elicit a fun feeling for the passenger when they view the physical outputs from the system.]*

Flexibility - How much effort is needed to add or change capabilities within the product. *[The system shall allow for the addition of new fields of interest with no more than 2 hours of effort.]*

Hardware Interfaces - Characteristics of each interface between the software system and the hardware components supported by the system. *[The system shall be able to upload digital photographs directly from a digital camera to attach to the passenger profile.]*

Legal - System constraints that are required by law. *[The system shall not display any information the passenger marked as confidential on the passenger profile.]*

Logical Database - Any information that is required to be placed into a database, including data relationships, field types, frequency of use, integrity constraints and accessing capabilities. *[The system shall maintain a history of passenger suggestions.]*

Memory Constraints - Constraints on the system based on memory usage. *[The client shall run on a system with 500 KB of memory, utilizing no more than 30% of the available system memory resources at any time.]*

Operations - System behaviors necessary for the support and operations of the system. *[The system shall notify the administrator users by email and pager if the database server becomes unavailable.]*

Performance - The speed with which the system accomplishes specific actions under specified conditions. *[The system shall regenerate and display one passenger's updated suggestions within 5 seconds of receiving the update request.]*

Portability - Ability to migrate from one platform to another or one machine to another. *[The system shall be designed such that the client runs in Windows XP now and in Windows Vista without code changes when the cruise ship upgrades its systems.]*

Reliability - Specified period of time and conditions for which the software must execute without a failure. *[The system shall operate without critical failure for a consecutive 72 hour period with 20 users simultaneously performing their common tasks. (See test plan for definitions of "critical failure" and "common tasks.")]*

Reusability - How easily a component of the system can be used by another system. *[The system's components to update the passenger profile shall be usable in the next release of the Passenger Tracking system.]*

Robustness - Expected behavior when there is invalid data, defects, or unexpected errors. *[The system shall inform the user of the issue and allow the user to continue working offline if the primary database server becomes unavailable.]*

Security - Behaviors necessary to protect the software from accidental or malicious access, use, modification, destruction, or disclosure. *[The system shall store all passwords using 128 bit encryption.]*

Site Adaptation - Behaviors necessary to support customization of the product specific to where it will be installed or used, including details of the customizations, globalization, and localization. *[The system shall display all time stamps in the cruise ship's local time.]*

Software Interfaces - Characteristics of each interface between the software system and other software components of the system or other systems. *[The system shall support using the Event Schedule system's data to match passenger interests to available programs.]*

Testability - Behaviors of the system necessary to support testing the system. *[All user interface components must react to scripted input from the testing tool exactly as if the user input the scripted data or command.]*

Usability - Defines the ease with which end user classes can perform specific tasks with the software. *[The user interface shall allow users to regenerate the passenger suggestions from within two clicks after updating their profile information.]*