

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 in Requirements Gathering Sessions

- What is the business objective behind the functionality requested?
- What basic functionality do you need for this part of the system?
- What is the underlying goal of the requirement?
- Is this particular design critical to the business goal, or would any design be equally effective?
- How does it work today?
- What specific steps do you go through to complete a task in the system?
- What do you not like about the system today?
- If X does not happen, then what should happen?
- What do you want to have happen in the future?
- What known constraints are there from other systems or users?
- What are the non-functional requirements that apply for each of the users, operating environments, and interfacing systems?
- Keep drilling deeper with variations of "why?"

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?

Assumptions

- ▶ What assumptions are we making when analyzing this use case?

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?

Success End 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 Steps

- ▶ How does the actor interact with the system?
- ▶ What does the system do at this step (present options, display data, execute a 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

- ▶ What possible error conditions exist at each step of the main scenario?
- ▶ If X doesn't happen, what should happen?
- ▶ What are the interrupts that can happen at any time?
- ▶ What is the "non-happy" path?
- ▶ What other possible actions can the user take at each step?
- ▶ If the user cancels out at any step, what should happen?

Non-Functional Requirement Types and Examples

Availability - Desired "up time" during which the system and data are available for use. *[The system shall be unavailable between the hours of 3AM and 5AM inclusive every working day for maintenance.]*

Communications Interfaces - Any communication functions the product uses are documented in this section (example – web browsers). *[The system uses HTTP protocol to communicate with the servers. Users use HTTPS (secure site) to login to the system.]*

Design and Implementation Constraints - Issues that will restrict the options available to the development team, including the reason for the constraint (example – technology choice). *[The system must be able to run on Windows 2000 and Windows XP.]*

Documentation - Describe the purpose, desired contents, level of detail, and formatting (User Manuals, Installation Guides, Online Documentation and Help Systems, and Labeling and Packaging). *[The system shall provide context sensitive help that takes the user to the specific help topic related to their present screen.]*

Efficiency - How well the system utilizes resources such as memory, disk space, etc. *[The system shall handle 25% more operations with existing hardware configurations.]*

Flexibility - How much effort is needed to add new capabilities to the product. *[In addition to satisfying all requirements to onboard vendors, the system shall allow for the addition of future vendors and/or future legal entities.]*

Hardware Interfaces - The characteristics of each interface between the software system and the hardware components of the system. *[The system shall have the ability to receive data from the portable barcode scanning devices used by the shipping users.]*

Interoperability - How easily the system can interact with other systems. *[The system shall be able to copy file from the source server to the destination server.]*

Legal - Any system constraints that are required by law. *[The system shall prompt the user to confirm whether or not the system will be used in any export restricted country as defined by the US Department of State in compliance with the USA PATRIOT act.]*

Logical Database Requirements - Requirements for any information that is to be placed into a database, including data models, field types, frequency of use, integrity constraints and accessing capabilities. *[The system shall keep the last five fiscal quarters worth of data.][User accounts consist of user names, passwords, and date of creation.]*

Look and Feel - General requirements that capture how the system will appear. *[The User Interface will utilize web pages and adhere to the standards already in place among the web-based applications.]*

Maintainability - How easy it is to fix a defect or change a requirement in the system after it is deployed. *[The system will follow all IT Change Management procedures when maintenance is required.]*

Memory Constraints - Any constraints on the system based on memory usage. *[The system shall utilize no more than 30% of the available system memory resources (minimum of 256k of RAM) at any time.]*

Operations - Any operations required by the user (example – application monitoring). *[The system shall utilize standard manual error logging as defined by the testing team.]*

Performance - Include requirements placed on the software performance or on performance of the human interaction with the software. *[The system shall respond to search requests within 2 seconds while handling up to 50 transactions per second for eight hours.]*

Portability - How easily the system can be migrated from one platform to another or one machine to another. *[The system shall be designed such that it may be moved from a Windows 2000 operating environment to a Windows XP environment without code modifications.]*

Reliability - Probability of the software executing without failure for a period of time. *[The system shall operate without critical failure for a consecutive 72 hour period under current average loads.]*

Reusability - How easily a component of the system can be used by another system. *[The system shall have separate modules that individually can integrate into existing order management and inventory management systems.]*

Robustness - How well the system needs to function when there is invalid data, defects, or unexpected errors. *[If the primary server is not available, the system shall be routed to the secondary server to perform the functions.]*

Security - Specify the factors that would protect the software from accidental or malicious access, use, modification, destruction, or disclosure. *[Any file uploads using protocols such as FTP shall be secured using 128 bit encryption.]*

Site Adaptation Requirements - Requirements specific to where the product will be installed (example – globalization requirements). *[All timestamps shown by the system shall be translated into the local time where the system is installed.][The system shall support varying names and number of storage sites at each warehouse facility.]*

Software Interfaces - The use of other required software products and interfaces with the system. *[The mainframe must accept orders from the system that contain alphanumeric fields.]*

Testability - Ease with which the system can be tested for defects. *[UAT must be able to be performed from a remote computer outside the UAT lab, as many users are global.]*

Usability - How easy the system is to use for the end users. *[The new user interface design shall increase the speed with which users can enter purchase orders by 30%.]*