Who are the primary actors?
What are their roles?
What tasks must they perform?

WHAT DO THESE QUESTIONS REMIND YOU OF?

The value of task analysis has been apparent to technical writers since the 1980s. Various approaches and methodologies have helped to build companies and to produce libraries of books and articles. However, a specialized methodology based on object-oriented programming has emerged that could involve technical writers and build on the skills we have doing task analyses. This methodology is known as: USE CASES.
One of the best known advocates of use cases Alistair Cockburn encourages professionals involved with software systems and business processes to practice understanding users in a structural way.


Technical writers know how important it is to understand the users. How can we work with our development and design organization to apply understanding of tasks and users? One way to get user and task analysis expertise integrated into the development process is to help develop use cases, which our allies in testing organizations can also use.

Understanding Terms and Concepts

Use Cases can be used to define requirements so everyone involved in product development understand the product. Following object-oriented models, Use Cases have attributes that must be defined for a complete Use Case.

- **Use Case** – a collection of related scenarios useful for defining requirements.
- **Actor** - a human (or a system) who instigates a Use Case for the purpose of achieving its goal. Actors can be identified as Primary Actors or Secondary Actors.
- **Scope** —the System boundary of the Use Case.
- **Pre-Condition** and **Post-Condition** — the preliminary condition of the system and the results after the scenario ends.
- **Scenario** — the series of events that result in the Primary Actor successfully realizing a Use Case goal. (Assumes “sunny-day” or success scenario.) Documented as a series of numbered steps written in text and/or supplemented with figures.
- **Extension** — possible “branch” from main scenario.
- **Variation** — alternative step within scenario.
- **UML (Unified Modeling Language)** – a standard object-modeling language; graphical symbols used to diagram Use Cases. A framework for Object-Oriented technologies and tools.
EXAMPLES

Cockburn quote for writers to note:
“Spend your energy learning to write clear text instead [of diagramming in UML].”

EXERCISE: Given the environment and background summarized at the top, fill in the template at the bottom to begin developing a use case.

Tips:
• Use Action Statement Format
  (time factor or sequence -> actor-> action ->constraints)
• Number each action sequentially
• Follow the primary course (separate extensions and variations)

Use Cases and Test Cases as your “friends”

Testers in software organizations support Use Cases because certain tools and methodologies support the automatic generation of test cases.

User Interface Designers support Use Cases for providing flexibility but structure for changing requirements.

Technical writers support Use Cases because…

Hint: Can user and task analysis expertise be truly integrated into the development process?

Discussion

INTRODUCING USE CASE METHODOLOGIES TO YOUR BUSINESS
Other Terms and Tools

Terms:
Systems Engineering
Class Diagrams (Architecture)
State Charts
Sequence Diagrams
Process Diagrams (Block)
Specification and Description Language (SDL) Transitions
Message Sequence Charts
Bubble Diagrams
Spreadsheets

Tools: Telelogic (telelogic.com)

References

- Agile Software Development: The Cooperative Game by Alistair Cockburn, 2001 (on the web: http://members.aol.com/acockburn/papers/AltIntro.htm)
- “Use Case Alternatives Introduction” by Alistar Cockburn, (on the web: http://members.aol.com/acockburn/papers/AltIntro.html)
- Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design by Craig Larman, Prentice-Hall.