Usage Scenarios for Product Requirements

Painting a Picture of User Interactions

The typical product requirements process starts with a list of features or tasks intended for a specific release and then dives into defining Use Cases or User Stories for the user functionality. This can result in a disjointed solution being developed due to lack of an overall picture of what the user expects to be able to do, and context as to what they are trying to achieve. This article discusses a means of identifying the user need and conceptual functional solution in a high level narrative format called a Usage Scenario before diving into more formal detailed requirements methods. Usage Scenarios can significantly improve your requirements process independent of your development methodology (e.g. Waterfall vs. Agile).

What are Usage Scenarios?

The term ***Usage Scenario*** is used in a wide variety of ways, but fundamentally, it is a story of a user (or actor) attempting to achieve some goal. The general application of scenario is a description of a course of action that a situation may take, such as in discussing "the worst case scenario". It can also refer to a specific path through a series of alternate flows, such as through a use case or a test case. In some environments, the terms Use Case and Scenario are used interchangeably, but we'll discuss the differences.

Usage Scenarios are heavily used in Interaction Design of user interfaces and was first popularized by Alan Cooper in *The Inmates are Running the Asylum*. The purpose of the Usage Scenario is to capture the essence of the user interaction with your system towards achieving a goal, without diving into the details of the system actions or the actual design. The focus is on the User, not the system. Their value can be leveraged much earlier in the requirements process than just the UI design, and can facilitate the conceptual definition of the intended system at an early stage. They can be used for communication, analysis and design, prioritization and decision-making.

A simple example is one for an ATM machine. The ATM Customer wishes to withdraw cash as follows:

> The **ATM Customer** realizes that he needs some cash but is nowhere near a bank, so he goes to a convenient ATM nearby to get it. He identifies himself to the ATM and specifies that he wants $100 from his checking account. He indicates that he doesn't need a record of the transaction or his current balance. He takes his cash when delivered.

Note that the description specifically avoids discussion of ATM cards and PINs and abstracts these to a higher level above the technical implementation of the system. This is because the Scenario can be implemented using any number of technical solutions, but the core activities the user performs remain constant even as technology evolves over time.

There are multiple types of Usage Scenarios for realizing different objectives. Two of the most common scenario types are Current and Proposed. The Current form describes the way a process or workflow currently works 'as-is' for informational purposes or to capture an existing problem for analysis. The Proposed form goes the next step and envisions how a solution might look as an alternative. It is this latter form that we'll focus on for the purpose of driving product requirements.

The value Usage Scenarios provide beyond traditional use cases and functional requirements is CONTEXT for the designers for what the user is attempting to accomplish and why. This is often lost in requirements communication and makes it more difficult for designers to fill in the gaps when they encounter questions or missing requirements. They are also very useful for illustrating flow and relationships between activities that are usually broken up into discrete use cases or user stories. They are easy to read and can be reviewed by anybody for validation.

From a process and documentation perspective, Scenarios help bridge the market and user problem to the envisioned solution, and can appear in any number of docs depending on your process. Figure 1 illustrates where Scenarios fit into a requirements process.

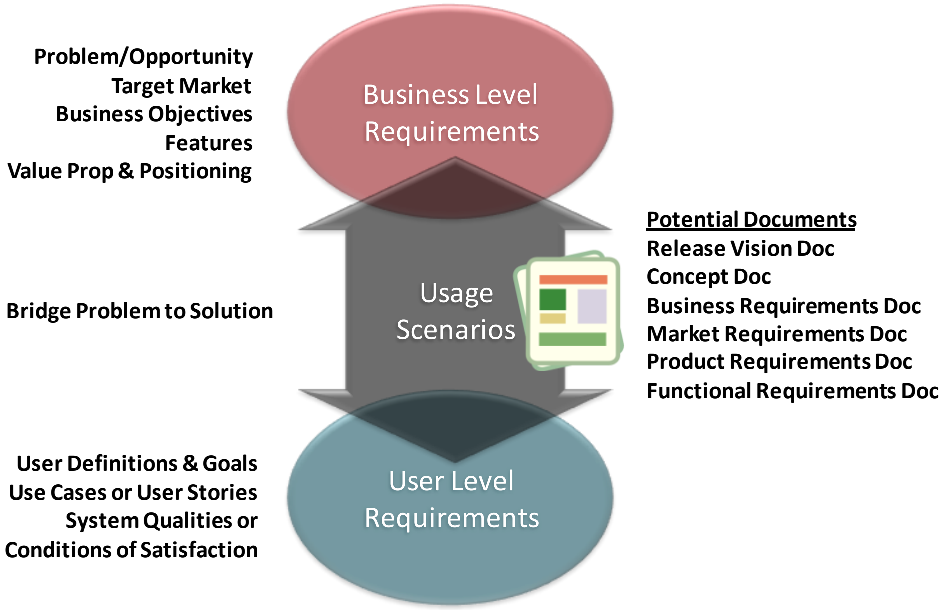

Figure 1 - Usage Scenarios in the Requirements Process

Users Drive Scenarios

As indicated, the focus of a Usage Scenario is on the User, and specifically on the tasks they need to accomplish toward achieving a higher level goal. It's important to identify the primary users of the system who need to interact with it and understand their motivations and expected usage of the product. (Note: it's possible to use Scenarios to describe the interaction between non-human systems or components, but there are other established means of describing these systems, such as sequence diagrams, and the need to capture "motivations" and "context" are not such an issue.)

A description of the User provides the background of what they are trying to achieve and how they might use the system. Usage Scenarios are traditionally driven by Personas, which are research-based composites of real target users. However, other forms of user descriptions can be utilized, such as User Profiles or User Roles, which are much less elaborate but still contain valuable information. These user descriptions are missing some data components, such as demographic or biographical information, and the focus is the behavioral information of the user relative to the tasks they are trying to accomplish with your product.

For example, let's continue our analysis of the ATM machine and the user role of the ATM Customer that might be developed through interviews and surveys:

User Role:	ATM Customer
Primary Goal:	To do their banking at times and locations that are convenient
Background:	Banking customer with both checking and savings accounts at a local bank and also has a credit card account. Typically works during normal banking hours during the week and has other activities during typical weekend hours.
Typical Usage:	Needs to make payroll deposits twice a month. Will want to get cash at convenient locations whenever needed, sometimes multiple times per week, and this will be the only transaction needed. Will occasionally want to check balances and transfer funds between accounts.
Motivations & Expectations:	Values convenience, both in being able to do his banking during his free time outside of working hours and at locations local to where he is. Expects the system to be no more difficult to use or slower than working with a teller for the same activities. Wants the system to feel safe and be secure.

Note that opportunities for product features are already becoming apparent in analyzing the behaviors and motivations of a typical user. It also does not take a large amount of survey data or interviews to obtain this information. A wealth of knowledge can be obtained through interviews with only a handful of prospective users, and/or by discussions with user proxies, such as customer support reps or sales personnel. You also could give a more personal name to your role, such as ATM Andy. This helps to make them seem more human and easier to relate to.

Developing Usage Scenarios

The main features of a Usage Scenario include:

- A narrative – it is in a text-based story format
- A user (or persona or actor) – someone is performing activities or actions
- A setting – the environment or context of the user
- A goal – the desired outcome for the user using the system under consideration
- Activities (or tasks) – the actions the user is performing to achieve the goal
- Decisions – choices the user makes from alternative options
- A bounded timeframe – covers a single, typical, continuous interaction within a timeframe

The first part is identifying a typical situation or setting that exhibits the problem the user is facing and their goal. The second part is to identify the proposed tasks or activities the user could do to accomplish their goal using your product, including some decisions they need to make along the way. This is where you are creating the envisioned solution to the problem at a high level. Analogous to the primary success path in a Use Case, the Scenario wants to develop a successful path. Also analogous to a Use Case, the Scenario occurs during a single session using the system.

Let’s expand on our ATM example with a richer set of functionality to show multiple ways of using the ATM system and to highlight the components:

Title: Deposit Paycheck	**Title** **Verb + Noun**
It's payday, and ATM Andy needs to deposit his paycheck into his bank. He worked late, and now the bank is closed, and he won't have time during normal banking hours for the next few days.	**Setting** **What is the problem?**
He decides to use the ATM at the bank branch on his drive home make his deposit.	**Goal** **Desired outcome**
He identifies himself to the ATM, specifies he wants to make a deposit to his checking account and inputs the check. He confirms the amount of the check is $952. He then specifies he wants to transfer $100 from checking to savings, and would also like to withdraw $50 in cash from checking.	**Activities** **(Deposit, Withdraw, Transfer)**
He indicates that he would like to get his current balances and does want a record of the transaction. He takes his $50 cash and transaction record which includes the current balances in his accounts.	**Decisions** **(Account, Amount, Balances, Receipt)**

Note that several different variations of similar activities exist for how ATM Andy could decide to use the ATM. He could just make a deposit, with no other actions. He could just withdraw cash, as illustrated in the first example. He could also just transfer funds or check balances. Or if you were developing a more advanced version of the basics, maybe he could also pay bills or order checks or even buy tickets to a local event.

The goal would be to pick a rich version of a combination of activities as representative of what could be done, and not try to develop a Scenario for every individual task. The goal, remember, is to capture the essence of the interaction and tasks the user could perform.

Where to Start?

You can start developing Scenarios at different points in the requirements process. One common approach is to first identify the high level business requirements for a product or release. These would include the market problem or theme of the release, the business objectives, and the intended list of features. Then identify the primary user(s) of the system for the target feature set and the goals they have in using the system.

From here, the Scenario can be developed in different ways. If the users are already accomplishing the tasks in some form and your goal is to automate it, you could abstract the tasks in a way that removes the current implementation and allows it to map to your new system. For example, in the ATM system, the exact same activities are used daily within a bank between customers and tellers, and in fact, a live teller solution would be one implementation. The activities the customer would expect to be able to do are the same, regardless of how it's implemented. In essence, the implementation is still "magic".

From here, the next step would be to define the Use Cases. A Scenario may map to a single use case, but more likely, a rich Scenario will map to multiple Use Cases that could be used in any combination and sequence, as discussed. Scenarios could also be expanded or evolved with more richness for User Interaction design to capture subtleties in the design of the user experience.

Or maybe you're in an Agile/Scrum software environment. After identifying the users, a list of user stories could be created to capture single tasks. User Stories are a short description of one piece of functionality a user desires to accomplish some a single task. The format for a User Story is:

> As a <user role> I can <do a task> (so that <reason>)

For our ATM, so examples of user stories could be – As an ATM Customer I can:

- Deposit checks to an account
- Deposit cash to an account
- Withdraw cash from an account
- Transfer funds between accounts
- Check my balance of an account

The Scenario can be created from a list of users stories targeted for a specific series of iterations forming a release, and could be used to illustrate the high level objective for the release. This also helps to form a more cohesive end-to-end release if you can tie the bits together in a strong way from a user perspective.

Usage Scenarios vs Use Cases

As mentioned earlier, sometime the terms are used interchangeably, and there are similarities and differences. A Use Case is broken down into numbered steps, and you will often see Usage Scenarios also have number steps, especially if the narrative is long and involves many different activities. This is likely the cause of the confusion.

The major difference, however, is the Use Case also includes statements about what the system will do, whereas the Usage Scenario will avoid this discussion. An example Use Case for our Withdraw Cash scenario could look like:

Title	**Withdraw Cash**
Description	User withdraws cash from their account via the ATM
Actors	(ATM) Customer (Primary), (ATM) Network (Secondary)
Primary Success Path	1. Customer enables operation with ATM. ATM authorizes operation. 2. Customer presents credentials. ATM validates credentials. 3. Customer requests Withdrawal, source account and amount. 4. ATM validates account and availability of funds with the Network and dispenses the proper amount of cash. 5. Customer takes cash and selects receipt and balance options. 6. ATM dispenses receipt and logs the transaction.

Additionally, the Use Case abstracts the selections and data elements to be generic, versus the specifics in the Scenario. The Use Case is at a higher level of abstraction than the Scenario, and encompasses a general flow that handles any number of potential Usage Scenarios. In practice, Usage Scenarios often include generic selection parameters, but in doing so, information can be lost as what is the expected PRIMARY path taken by most users. Thus, important design and prioritization information may be missing to drive a better implementation.

Scenario Tips

Below are a high level list of Do's & Don'ts when working with Scenarios.

Do's	Don'ts
Provide a setting, the problem, and goal to provide a visual for the situation in which a real user could find them self.	Avoid specifying what the system is doing. You haven't yet determined how you're going to implement it.
Provide a rich set of activities or tasks and decisions the user could need to make within a single continuous session.	Avoid specifying any User Interface components, such as "He clicks OK" or "She presses Withdraw Cash". It's too early in the process.
Be specific, such as to the actual selection or decision, e.g. withdrawing $20. You are defining an exact path the user is choosing. It's as if you're watching and recording an actual interaction.	Avoid providing different Scenarios for similar tasks. They don't offer new information and will bore readers. Save this level of detail for Use Cases or User Stories.
Abstract operations to get the technology out. Focus on the user actions and choose verbs and descriptors that are technology independent	Avoid micro-Scenarios that just have one or two activities and few decisions. They don't offer much information. Try to combine into a richer set of activities, or just save them for lower level Use Cases.

Summary

Usage Scenarios are text-based stories that describe user interactions with a system to capture the context and goals of the user. They are very effective as a starting point in the user level requirements process for prioritization purposes and to capture full end-to-end user flows. They can also be applied after some lower level task identification has been accomplished at the Use Case or User Story level to unite the fragments of functionality into a cohesive unit.

For additional reading on the topic, the following books are recommended:

The Inmates are Running the Asylum – Alan Cooper, 1999

About Face 3, the Essentials of Interaction Design – Alan Cooper, 2007

Scenarios, Stories, Use Cases through the Systems Development Lifecycle – Ian Alexander/Neil Maiden, 2004

About Product Arts

Product Arts specializes in Product Management consulting and training. Our consulting consists of product strategy and planning, market performance improvements and processes for creating and delivering products. Our training includes public and private custom training for product management and associated staff. For more information, go to www.product-arts.com or email info@product-arts.com.