Personas, Requirements, user stories, scenarios and use cases

Personas, scenarios, user stories and use case

4 user research methods to help you create reliable and realistic representations of your target users and design accordingly

<u>USER CENTERED DESIGN!</u>

DANIELE MAZZEI - PROGRAMMAZIONE INTERFACCE 21-22

PERSONAS

PERSONAS

A **user persona** is an **archetype or character** that represents a potential user of your website or app.

In user centered-design, personas help the design team to target their designs around users.

they are an integral part of the user experience research phase of software development.

In user research, UXers will gather data related to the **goals and frustrations** of their potential users. Then, they create personas to put that data into context.

PERSONAS

There is usually more than one type of user who will interact with your website or app, and creating personas **helps to scope out the range of users**.

For UX teams, introducing persona development into the design process helps them learn about the spectrum of goals and needs of their users.

A designer's checklist if you like.

PERSONAS

User persona development helps us **bridge the gap between the company and its users** by allowing us to measure true user behavior and figure out what their end goals might be.

It **drives design decisions** by allowing software teams to get a deeper **understanding of the users** who will be using the systems they are building.

But while they might be useful, creating user personas isn't always fun and games.

PERSONAS

Creating a user persona starts with user research.

By observing users, UXers can **understand their behavior and motivations**, and then design accordingly.

There are plenty of user research techniques that help UXers capture this information, such as:

- task analysis (card sorting, first click testing etc.)

- feedback (contextual interviews and focus groups)

- prototyping (experimenting with ideas prior to developing them)

PERSONAS

How many personas should I define?

user persona's design should ideally be [based around the Pareto Principle](#).

Focus on **that 20% of your user-base that will use/buy 80% of your features/products, or that will account for 80% of your revenue.**

Pareto principle *states that, for many events, roughly 80% of the effects come from 20% of the causes*

PERSONAS

Upon observing users, UX designers split up the test data into possible user archetypes, or user personas.

Then all this information is put into context in a **user persona template**.

Designers can only respond to their users' needs once they know what those needs are.

examples:

https://simplicable.com/new/user-persona

https://www.justinmind.com/blog/user-personas-which-game-of-thrones-character-is-yours/

Personas recap

https://www.smaply.com/blog/personas

REQUIREMENTS

Requirements

a requirement is **a service, function or feature that a user needs.**

Requirements can be functions, constraints, business rules or other elements that **must be present** to meet the need of the intended users.

Requirements

For example:

In a training company with its own training centre:

- **The Course Manager** has a requirement to **schedule training courses** and reserve rooms, in order to make available courses visible and to ensure courses run effectively

- The Training **Centre Manager** has a requirement to **keep track of what training is running**, in order to ensure appropriate allocation of trainers to courses

- The **Financial Accountant** has a requirement to **maximise the amount of time that the training rooms are in use**, in order to maximise revenue from the rooms

Requirements

However, the attempt to define a full and detailed set of requirements too early in a project often proves to be counterproductive, restrictive and wasteful.

It is not possible to define all of the detailed requirements at the outset of a long project.

The business environment changes as time progresses; new requirements and opportunities present themselves. As the project progresses, the team understand more about the business need.

Defining detailed requirements too early means either needing to change the specification later, which wastes the original work, or delivering to the originally-specified requirements and subsequently failing to adequately satisfy the business need.

Requirements

The success of any solution is the product of two aspects:

- **what it does** (functionality, features)

- **how well it performs against defined parameters** (non-functional attributes, acceptance criteria, service levels)

Categories of Requirements

Functional Requirements (FRs) express **function or feature and define what** is required, e.g.

- Visit customer site

- Obtain conference venue

<u>The requirements do not state how</u> a solution will be physically achieved.

- Drive to customer site is one possible solution. However, fly to customer site or travel by train to customer site are potential alternative solutions which may be worth consideration

- Build conference centre is one possible solution. Hire a hotel room is an alternative solution

Categories of Requirements

Non-functional Requirements (NFRs) <u>define how well, or to what level</u> a solution needs to behave.

They describe <u>solution attributes such as security, reliability, maintainability, availability</u> (and many other "...ilities"), performance and response time, e.g.

- responding within 2 seconds

- being available 24 hours per day, every day

Categories of Requirements

NFRs may be:

- Solution-wide or impacting a group of functional requirements: e.g.

    - All customer facing functionality must carry the company logo

    - All customer-facing functionality must respond within 2 seconds to requests

- Related to a particular functional requirement, e.g.

    - Hire conference venue might have NFRs of accessibility, security, and availability

USER STORIES

User stories

A user story is a **short statement or abstract that identifies the user and their need/goal.** It determines who the user is, what they need and why they need it.

There is usually one user story per user persona.

there are often multiple user personas – it's a good thing that user stories are brief!

stories are at the center of the user experience. Why? They put things in context and focus on the 'holistic' rather than the 'artifact'.

A User Story is a requirement expressed from the perspective of an end-user goal.

User stories

For example:

"As a UX Manager, John oversees all the design projects, including assets creation and prototyping efforts, at the design consultancy where he works. He needs easy access to a design tool that allows him to centralize UI libraries so that multiple designers to work simultaneously on a prototype."

*Requirements example: **The Course Manager** has a requirement to **schedule training courses** and reserve rooms, in order to make available courses visible and to ensure courses run effectively*

User stories

User stories help to document practical information about users, such as the different needs and motivations for accessing a website or app.

They also help the development team estimate a roadmap needed to deliver the end product.

User stories

How to write a user story:

It's super simple to write a user story.

"As a [role], I want [feature] because [reason]."

For example: "As UX Manager, John wants centralized assets management so that his designers are in sync."

this approach helps you to think about who a certain feature is built for and why

Who writes user stories?

Anyone can write user stories.

It's the **product owner's responsibility** to make sure a product backlog of user stories exists, but **that doesn't mean that the product owner is the one who writes** them.

Over the course of a good agile project, you should expect to have user story examples **written by each team member**.

Also, note that who writes a user story is far less important than who is involved in the discussions of it.

User stories

One of the benefits of user stories is that <u>they can be written at varying levels of detail.</u>

We can write a user story to cover large amounts of functionality. These **large user** stories are generally known as **epics**.

Here is an epic agile user story example from a desktop backup product:

- As a user, I can backup my entire hard drive.

How is detail added to user stories?

Because <u>an epic is generally too large</u> for a design team to complete in one iteration, it is split into multiple smaller user stories before it is worked on.

The epic in the previous slide could be split into dozens (or possibly hundreds), including these two:

- <mark>As a power user,</mark> I can specify files or folders to backup based on file size, date created and date modified.

- <mark>As a user,</mark> I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.

How is detail added to user stories?

Detail can be added to user stories in two ways:

- By splitting a user story into multiple, smaller user stories.
- By adding "conditions of satisfaction."

When a relatively large story is split into multiple, smaller agile user stories, it is natural to assume that detail has been added. After all, more has been written.

The conditions of satisfaction is simply a high-level acceptance test that will be true after the agile user story is complete.

How is detail added to user stories?

Consider the following as another agile user story example:

As a vice president of marketing, I want to select a holiday season to be used when reviewing the performance of past advertising campaigns so that I can identify profitable ones.

Detail could be added to that user story example by adding the following conditions of satisfaction:

- Make sure it works with major retail holidays: Christmas, Easter, President's Day, Mother's Day, Father's Day, Labor Day, New Year's Day.
- Support holidays that span two calendar years (none span three).
- Holiday seasons can be set from one holiday to the next (such as Thanksgiving to Christmas).
- Holiday seasons can be set to be a number of days prior to the holiday.

User Stories recap

??

SCENARIOS

Scenarios

A scenario is a situation that captures how users perform tasks on your site or app.

Scenarios describe the user's motivations for being onsite (their task or goal) and/or a question they need answered, and suggest possible ways to accomplish these objectives.

It is essentially a development of the user story, and can relate to multiple target users.

However, scenarios can also be broken down into use cases that describe the flow of tasks that any one user takes in a given functionality or path.

Scenarios

For example, a scenario could outline how John uses a mobile app to buy a ticket to a design workshop whilst on his way to work.

Scenarios help stakeholders envision the ideas of the design team by providing context to the intended user experience – frequently bridging communication gaps between creative and business thinking.

For the design team, scenarios help them imagine the ideal solution for a user's problem.

"Scenarios are the engine we use to drive our designs." (UX influencer, Kim Goodwin)

What to Consider When Writing Scenarios

Good scenarios are **concise** but answer the following key questions:

- **Who is the user?** Use the personas that have been developed to reflect the real, major user groups coming to your site.

- **Why does the user come to the site?** Note what motivates the user to come to the site and their expectations upon arrival, if any.

- **What goals does he/she have?** Through task analysis, you can better understand the what the user wants on your site and therefore what the site must have for them to leave satisfied.

<u>Some</u> scenarios also answer: **How can the user achieve** their goals on the site? Define how the user can achieve his/ her goal on the site, identifying the various possibilities and any potential barriers.

How to write a scenario

Scenario planning starts with scenario mapping.

The design team, developers and product owner will meet to exchange ideas and create a strategy based on their user personas.

With the **primary user defined through persona** development, they can now consider **the key task that the user hopes to achieve**.

The next step is to perform a scenario analysis, **put the user's goals into context** and walk through the steps that the user would take.

How to write a scenario

Creating Scenarios requires a special mindset.

It is about **focusing on the users' goals**: what will they try to accomplish on a website or inside an app?

Additionally, it is also important to think about their context, their prior knowledge and background.

How to write a scenario

Thanks to Scenarios, we can determine:

● the most important points to focus on during the UX design process

● which steps of the process would require additional help to your users

● the main needs and motivations of your users.

Scenarios are built upon User Story: these short statements describe what a certain User Persona needs, and why. <u>Scenarios take User Stories to the next level by adding the interaction with the product or service to the story.</u>

How to write a scenario

https://uxknowledgebase.com/scenarios-43e05671b07

Type of scenarios

https://www.usability.gov/how-to-and-tools/methods/scenarios.html

Scenarios recap

https://www.interaction-design.org/literature/topics/user-scenarios

USE CASES

Use Cases

A use case is a written description of how users will perform tasks on your app.

It outlines, from a user's point of view, a system's behavior as it responds to a request.

Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

Scenarios vs Use Cases

A Scenario involves a <u>situation</u> that may have single or multiple actors that take a given functionality or path to achieve their goal

A use case involves <u>an actor</u> and **the flow** that a particular actor takes in a given functionality or path. <u>These often get grouped so you have a "set" of use cases to account for each scenario.</u>

The main difference is "perspective".

The use case is more granular than the scenario.

it usually involves coming up with a scenario and then defines all the use cases that fit into that particular scenario.

Use cases

Use cases add value because <u>they help explain how the system should behave and in the process</u>, they also help brainstorm what could go wrong.

They <u>provide a list of goals</u> and this list can be used to establish the cost and complexity of the system.

Project teams can then negotiate which functions become requirements and are built.

Use cases

What Use Cases Include	What Use Cases Do NOT Include
- Who is using the website - What the user want to do - The user's goal - The steps the user takes to accomplish a particular task - How the website should respond to an action	- Implementation-specific language - Details about the user interfaces or screens.

Elements of a Use Case

Depending on how in depth and complex you want or need to get, use cases describe a combination of the following elements:

- **Actor** – anyone or anything that performs a behavior (who is using the system)

- **Stakeholder** – someone or something with vested interests in the behavior of the system under discussion (SUD)

- **Primary Actor** – stakeholder who initiates an interaction with the system to achieve a goal

- **Preconditions** – what must be true or happen before and after the use case runs.

Elements of a Use Case

- **Triggers** – this is the event that causes the use case to be initiated.

- **Main success scenarios [Basic Flow]** – use case in which nothing goes wrong.

- **Alternative paths [Alternative Flow]** – these paths are a variation on the main theme. These exceptions are what happen when things go wrong at the system level.

Use case?

https://www.youtube.com/watch?v=Ct-lOOUqmyY&ab_channel=NowI%27veSeen Everything

How to write a use case

Kenworthy (1997) outlines the following steps:

1. Identify who is going to be using the website.

2. Pick one of those users.

3. Define what that user wants to do on the site. Each thing the use does on the site becomes a use case.

4. For each use case, decide on the normal course of events when that user is using the site.

How to write a use case

5. Describe the basic course in the description for the use case. Describe it in terms of what the user does and what the system does in response that the user should be aware of.

6. When the basic course is described, consider alternate courses of events and add those to "extend" the use case.

7. Look for commonalities among the use cases. Extract these and note them as common course use cases.

8. Repeat the steps 2 through 7 for all other users.

The Takeaway

Engaging in user persona, user story, scenario and/or use case development will help you to identify key information about your users and build products that will delight your users time and time again.

Everything we do to get closer to users is a step in the right direction.

examples

https://uxplanet.org/5-examples-of-brilliant-ux-design-8e847bf0bcc0

https://uxdesign.cc/fitbit-a-usability-case-study-b23e4c539c3c

https://uxplanet.org/foodmix-cooking-app-ux-case-study-d046c1f5896b