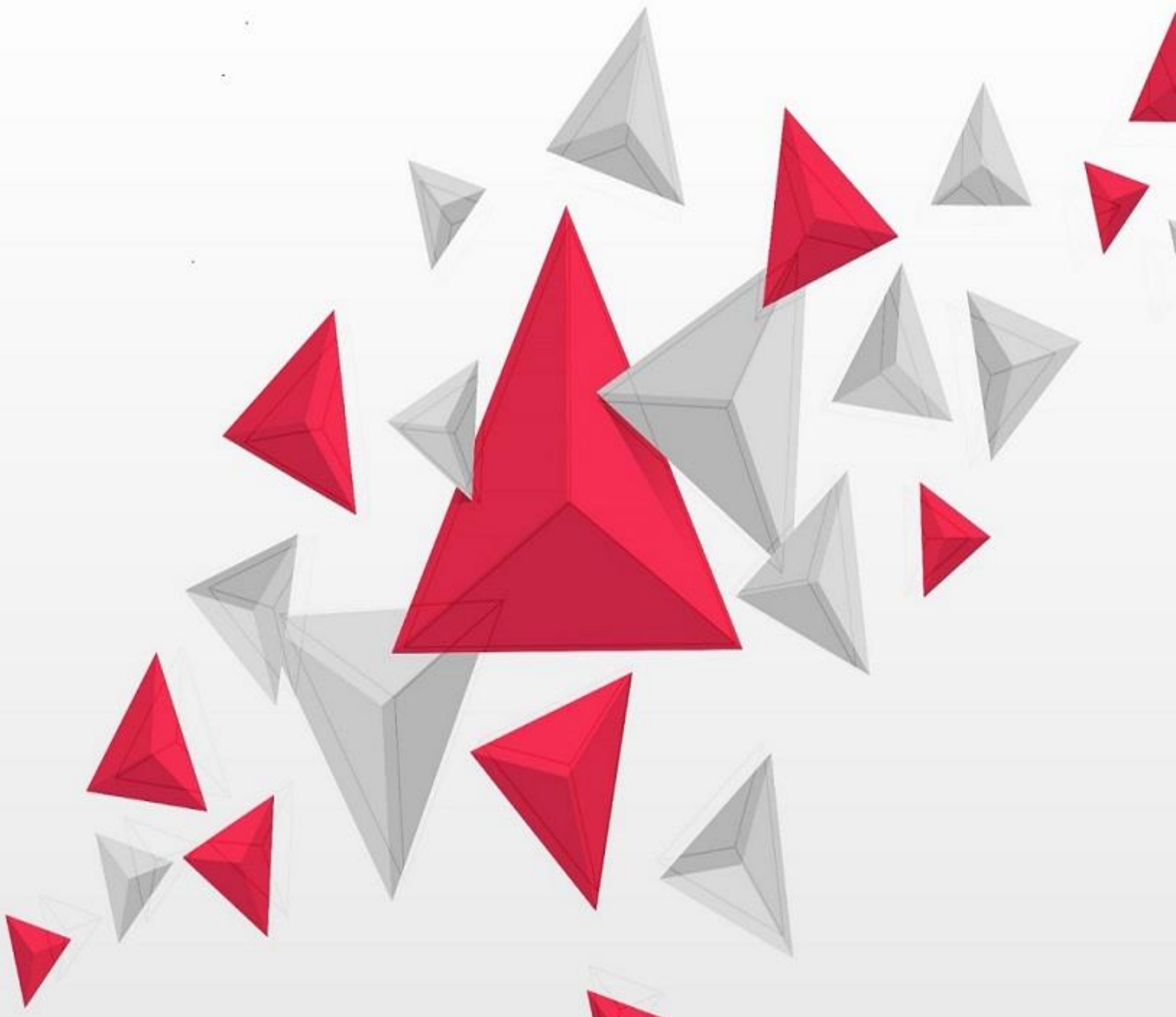


# UI-UX Guide for Business Analysts

## Guide Book





## INTRODUCTION

Wireframes. Prototypes. Mockups. User Interface.

All these words are commonly and loosely used in the Information Technology domain. Today, we will learn what each one of them is, *the situations* that demand them, and *how* to go about creating them.

In any IT project where you create an application or a product, creating some visual layout after you are done with the initial rounds of requirement elicitation is inevitable. The aim of doing so is to create a model or an early sample of the product so that the stakeholders can *feel* and *interact* with it, request and receive feedback and carry out any modifications before the actual source code is written against it.

In some organizations, it's the sole responsibility of an analyst to come up with such representations, whereas, in other, more process matured organizations, the User Experience professionals work hand in hand with the analysts to come up with these visual deliverables. Irrespective of the situation, an analyst *is* expected to know about the basics of design workflows, including coordinating and leading the activities surrounding them – and we will be covering it all under this lesson.

So, let's start with the definition of all those terms that are used quite interchangeably throughout the IT domain.



## DESIGN BASICS AND TERMINOLOGY

### Wireframes

Wireframes are usually the first visual representation that is drawn while designing the blueprint or schematic layout of a website/application and contain the positions of significant application elements depicted through blocks and frames. The **wireframe is static and can be called the bare-bone skeleton of an application.**

Since wireframes concentrate more on functionality and less on design, they can be created relatively quickly through tools like Balsamiq.

*E.g., the wireframe of an application's dashboard will contain rudimentary lines and boxes depicting the position of the application header, top menu bar and its main buttons, the layout of widgets, footer, and the navigations.*

The benefits of developing wireframes for applications involving end-user visual interaction is multifold:

- It helps the customer 'see' all the requirements being taken care of
- Aids the business analyst in ensuring all the flows and functionalities are covered
- Assist the development team 'in understanding' user interactions and navigations
- It gives all the stakeholders an interim 'look and feel' of the final application

### Prototypes

Prototypes are an early version or model that gives an experience of the layout, functionality, and interactions involved with using the final applications or products.

**In terms of visual representation, a prototype contains higher fidelity than a wireframe but a lower fidelity than the mockup.** Since they are *dynamic* in nature, an analogous analogy will be to compare a prototype to the 'muscle' of an application,

which will help you see the movement and interactions amongst various application sections.

*E.g., the prototype of an application's dashboard will contain the application header with a logo, a top menu bar with all buttons and images, the layout of widgets with relevant data, the footer, and clicking the buttons and links take you to the respective screens.*

Prototypes are highly iterative, with the initial versions containing a scaled-down model, and an increased level of functionality and details are added with every successive iteration.

Mockups against a screen are designed only when the customer finalizes the respective prototype.

## Mockup

Especially in the field of software engineering, **a mockup is a realistic representation that will look exactly/almost exactly like the final application** or product but will not do any practical work beyond what the user sees (i.e., a mockup is not clickable). You can call them the 'skin' of the application.

*E.g., the image of an application's dashboard containing all items present in the wireframe along with appropriate graphics, font, colors, margins and appearing exactly like the final dashboard screen.*

Unlike wireframes, creating mockups demands careful analysis of the complete application as a whole and designing the pages with the help of tools like Adobe Photoshop or Illustrator while taking care of the color scheme, spacing, graphics, etc.

Thus, mockups are developed only for complex applications in functionality and representation and are worth the effort and money involved in creating them.

## User Interface (UI)

Simply put, in software development, the **User interface is every single visual aspect that involves interaction between an end-user and the system**. It includes all the commands given, menus accessed, buttons pressed, and icons and links clicked.

A user interface aims to make all user-to-system interactions as accessible, intuitive, and usable as possible while taking care of all the user needs.

With the burgeoning increase in web and mobile-based applications involving interactive, complex interactions between a 'user' and a 'device', the user interface is equally (and sometimes more) important than the very functionality of the application.

## User Experience (UX)

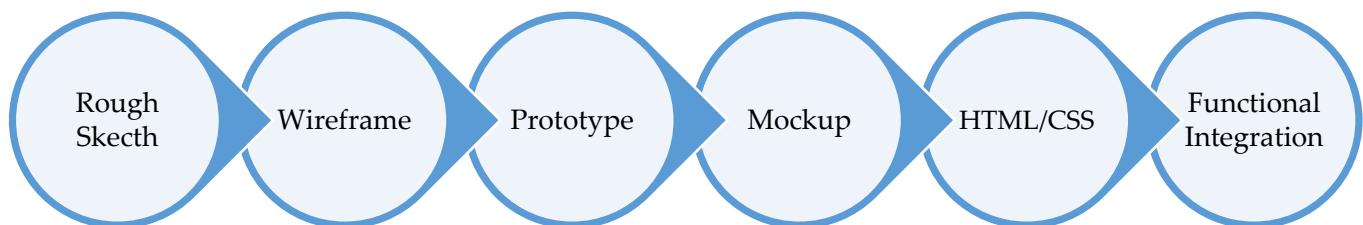
The term user experience is used in conjunction with the user interface, and it covers all aspects of the end user's interaction with an application or a product (that's why the term *experience*).

For instance, while choosing a mobile phone, you consider its features and functionalities and the kind of emotions, feelings, and psychological responses it triggers while using it is a part of the user experience.

Given the large number of options to choose from, perceptions you make about a product or application are now equally important as its functionalities. Even though the UI of a product was ideal, it may still fail if its aesthetics and interaction experience were not satisfactory enough.

## Development Process

The typical processes that are followed, converting sketches and wireframes to working applications are:



It's a pretty streamlined process and kind of an industry-accepted standard.

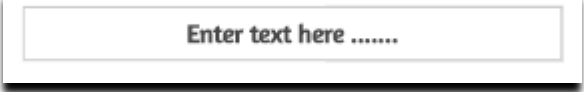

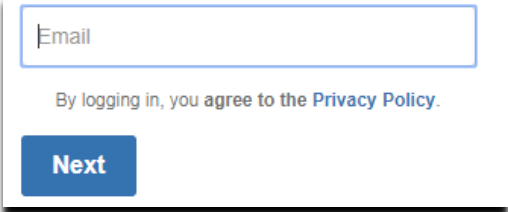
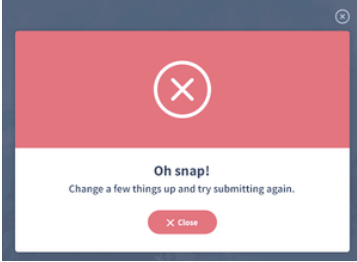
However, there are a few exceptions, like when the prototypes themselves are high fidelity, the mockups are skipped. Or, experienced analysts will jump straight to

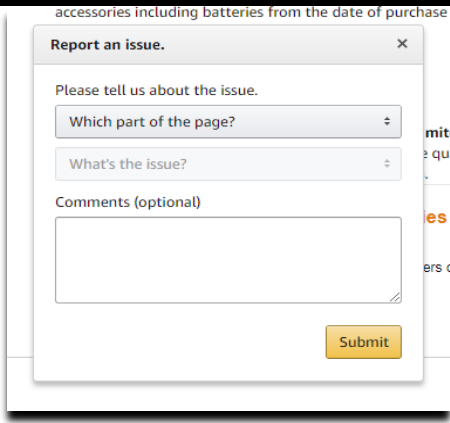

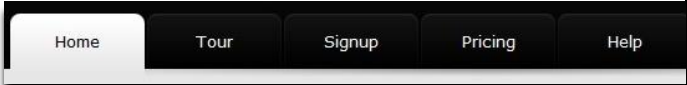


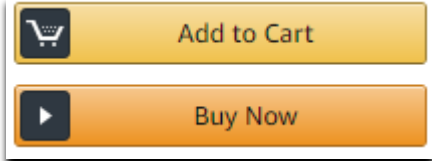
wireframes skipping the sketches. At other times, the organizations will define their own set of processes to carry out the design-related development activities.

## Additional Terms




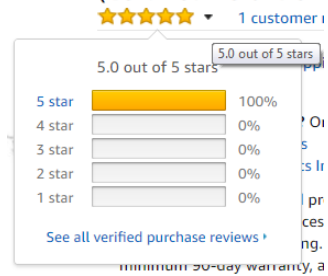

I would also want you to take note of some of the other terms that are used while talking about user interface and design.

This is a carefully curated list that contains only those essential terms and their examples that are used in the daily conversations of an analyst:

<b>Text box</b>	A rectangular box on a page where you can enter text	
<b>Text area</b>	An area (more prominent than the text box) on a page where you can enter text	
<b>Field</b>	A specific area on the page where you need to enter information.	
<b>Dialog</b>	A popup window that appears after executing an action like an alert or notification	

<b>Modal window</b>	A popup window that comes up as a result of action allows the user to interact (like fill info) and then returns the flow to the original screen	
<b>Icons</b>	Graphical or web element that represents a shortcut to an action	
<b>Tab</b>	A graphical or web element that groups a set of actions and functionalities	
<b>Wizard</b>	A series of dialogue boxes that walks a user through the sequence of steps to perform a particular task.	
<b>Legend</b>	The wordings on a map or a diagram explaining the colors or symbols used	
<b>Call To Action (CTA)</b>	A button or a link that excites or provokes the user to act. Like 'Sign Up', 'Call Now', 'Find Out More'.	

<b>Navigation</b>	Set of actions and movements that guides the users along with a website or application	
<b>Menu</b>	Categories along which the content or functionalities are organized in an interface. Like side menus, header menus.	
<b>Landing page</b>	Any page on which a user can land (usually after clicking a CTA). They are specially designed to make the user perform a specific action.	
<b>Picker</b>	An element that allows the user to pick from a row of options. Like date picker, time picker.	
<b>Checkbox</b>	Squared UI element used to make a selection of a particular option on the page	
<b>Radio Button</b>	A graphical control element allows a user to select only one of the predefined option	
<b>Bread-crumb</b>	A clickable trail of navigation allowing a user to identify their	

	current location within an application	
<b>Pagination</b>	Clickable page numbers dividing large content into pages and allowing a user to jump between pages	
<b>Slider/Track Bar</b>	A bar with a slider that allows a user to define and adjust values against a functionality	
<b>Carousel</b>	A set of images represented in a format where the user can see a couple of them at the same time and make a selection	
<b>Tool Tip</b>	The text that appears when a user hovers its mouse over a field and explains the use/intent/details of the field	
<b>Responsive design</b>	The design of an application that adapts itself (or 'responds') based on the screen size and orientation it's being viewed on	

We've covered the basics and are now well versed in the lingo; it's time to learn how to create these UI models.



## HOW TO CREATE WIREFRAMES

### Part A

At the onset, I would like you to understand that most business analysts only create the wireframes and sometimes the prototypes. At the same time, the other design-heavy intricacies like mockups and user interface are left to the respective experts, i.e., the UI-UX Designer and Graphics Designer.

However, if the analysts are not creating even the wireframes and the complete work is being delegated to a UI-UX designer, then the BA creates a document that assists a designer in understanding what exactly the application contains, the main menus, the functionalities, and their placement. This document is called a *User Interface Specification* document.

Now, if you are wondering when I am going to teach you to create the user interface specification document, then the answer is 'I won't'!

Don't worry. It's not because I don't want to but because there is no specific format for it, and there are reasons for it.

The UI specs is an informal diagram and are not deliverable to anybody. It's more of a document that is created to facilitate the conversation and information exchange between you and the designer.

Couple of important things before you create a specification document:

- It's a document that is made with the intent of describing the visual layout of the application to somebody who is designing it (usually the UI-UX designer)
- The specifications document should contain the menus, label names, display rules, message wordings, and the flow navigation in precise wording as the designer might not know the more advanced functions of your application.

- It's not necessary to create specifications for every page as some pages are self-explanatory
- While creating the UI specifications document, if you find yourself copying details from a use case document, then you need to stop.  
It's better to give the complete use case to the designer, who will better understand the product in general and the functionality in particular. Also, if you have created your use case carefully, it should contain all the design notes and business rules under the 'Special Requirements' section.
- Ensure that the designer has a fair understanding of the overall application/product – the more, the merrier. It will help her figure out some of the things on her own when you are not around, and what's more, she could even suggest some creative ways to represent the application's sections and functions in a better fashion.

## Part B

This part covers the guidelines that the analyst should take care of when they are creating the wireframes, all on his own.

If you think that wireframes require you to master a graphic designing tool, you need a mindset shift!

**Wireframing is 80% contemplation & visualization and just 20% drawing.**

If you have ever scribbled a layout or explained a concept/information flow to somebody on paper, you have the skills required to create wireframes and prototypes.

Talking about drawing tools, there are a plethora of choices to make from Microsoft Visio, Balsamiq, UXPin, Moqups, Gliffy, Mockflow, and your pen and paper (that's right – an image of your rough sketch will qualify as a wireframe).

And although Balsamiq is regarded as one of the best low-fidelity wireframing tools out there, the choices vary based on the analyst's preferences.



## WIREFRAMING BEST PRACTICES

To draw the wireframes, follow the guidelines below:

### 1. Don the persona of the application user

Any application is created from the perspective of a user or a group of individuals using it. It could be the buyer, the seller, the admin, or anybody else, and the analyst should put herself and her thought process into the shoes of the person for whom the application is being created.

Since an analyst understands the user types and their personas the best, she is the perfect and the ideal candidate to create the wireframes and the prototypes of an application.

### 2. Think in your mind before you trace it on paper

The analyst should create a mental picture of the visual layout, the placement of the various sections, chalk out the application hierarchy, and picture a user interacting with the application.

Again, if you are worrying that you are not creative enough, believe me, your natural thought process and analytical thinking will automatically get to work the moment you start pondering over. You will find yourself arranging the information in your head into logical modules and sections of the application.

Additionally, answering some of the questions below will be a great help along the way:

- a) What is the background of the prime user base?

- b) What are the different roles and the demographics of the users belonging to those roles?
- c) What is the application's primary purpose (this should be kept in mind throughout the design process)?
- d) What is the goal of this page (ask this question for every page)?
- e) Are all user flows being taken care of?
- f) Is there a way the current design, layout, or representation can be made more usable?
- g) Is the layout clean and intuitive?
- h) Are there too many elements fighting for a user's attention? Is the user susceptible to confusion or distraction?
- i) Will somebody be able to use the application without any help or wizard?

Don't just treat the above points as yet another piece of advice. Spend some time answering them, and you will be amazed by the insights they give you.

### **3. Draw the skeleton by defining broad outlines**

Based on your comprehension of the application and imagination, start arranging the layout in the form of plain square or rectangular boxes acting as mere placeholders of information. Take one screen at a time, draw the placeholders, and move on.

This is an iterative process, and you might find yourself changing those placeholders here and there. Once you feel you are done, walk yourself through all the elements a couple of times and make sure the overall flow looks coherent.

A piece of advice here - There is no right or wrong way to create a wireframe, but a wireframe that is structured, straightforward, yet covers all the functionalities will be preferred. Keep that in mind, and you may also look out for similar applications or products in the same space and get some ideas or inspiration (and don't be a copycat!).

Another point, since you are designing one screen at a time, you might be tempted to put all the screen details in one go, finish it in all respects and then move on to the next screen - Resist that temptation.

#### 4. Add details under the placeholders

Now it's time to add another layer of information to your placeholders to make them look meaningful. But be careful with how many details you put in here. Some quick pointers here:

- a) Give proper names and labels to the sections
- b) Include actual text (no lorem ipsum, please)
- c) Define hierarchy – what comes first and what next
- d) Add size and relative positioning to the elements
- e) Functionality and flow should be clear
- f) Adding colors is a strict no-no and shows you are adding more details than required
- g) They need not look polished but just get the information across

Refrain from putting any more information; the wireframes are expected to look bare and bland!

#### 5. Get feedback before you finalize

User's behavior and feelings while using the application should be given special consideration as they are crucial in defining its success.

Try to find out if the users will be '*Satisfied*' that the application is intuitive and behaves as expected, will they be '*Puzzled*' as the flow is confusing, or will they be '*Frustrated*' that it's challenging to figure things out.

Rather than trying to answer it yourself, request somebody either from your team or outside to spend some time on your wireframe and get some feedback. Remember, working or thinking on the same lines for a long time makes your thought process skewed, and having a fresh pair of eyes look at your wireframes will help you appreciate things from a different perspective.