



UI and UX for Responsive Web Design

UI and UX for Responsive Web Design

© 2015 Aptech Limited

All rights reserved.

No part of this book may be reproduced or copied in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or storing in information retrieval system or sent or transferred without the prior written permission of copyright owner Aptech Limited.

All trademarks acknowledged.

APTECH LIMITED

www.aptech-worldwide.com
E-mail: ov-support@onlinevarsity.com
Edition 1 – November 2015



Preface

This book, **UI and UX for Responsive Web Design**, provides an understanding of how Responsive Design works and highlights the importance of a UI design. This book also provides the learning process of creating a perfect user experience design for all kinds of users and also on how to create a responsive Website that can be viewed on devices with differing screen sizes and characteristics such as retina displays.

The knowledge and information in this book is the result of the concentrated effort of the Design Team, which is continuously striving to bring to you the latest, the best, and the most relevant subject matter in Information Technology. As a part of Aptech's quality drive, this team does intensive research and curriculum enrichment to keep it in line with industry trends and learner requirements. Please feel free to send your feedback.

Design Team

Onlinevaristy
your e-way to learning

TechnoWise

Are you a
TECHIE GEEK
looking for updates?

Logon to

www.onlinevarsity.com

Onlinevaristy
your e-way to learning

TechnoWise

Are you a
TECHIE GEEK
looking for updates?

Logon to

www.onlinevarsity.com

Session 1

Introduction to User Interface Design

Learning Objectives

In this session, you will learn to:

- Define User Interface (UI) and User Interface design
- List and explain the different elements of User Interface Design
- Describe the principles of User Interface Design
- Describe the types of User Interface Design
- Explain the User Interface Design process
- Describe the models in User Interface Design

The user interface is a vital part of any computer system. It determines how easily an end user can interact with the program. One of the achievements of a system depends on how well a user interface is designed and whether it creates a seamless experience for end users.

This session begins with explaining the meaning of User Interface (UI) and UI design. It provides an overview of the fundamental parts of the UI and different types of UI designs. The session also covers the UI design principles that can be utilized to improve the quality of user interface design. Further, the session outlines the user interface design process and the different UI models.

1.1 Definition of UI and UI Design

What is User Interface?

The UI of an application, also known as an 'interface', is the means by which a user and a computer system interact. It can comprise both software and hardware components.

In particular, UI includes:

- The textual, graphical, and auditory information that the program presents to the user.
- The control sequences that a user employs to control the program. For example, mouse movements, keystrokes with the computer keyboard, and selections through the touchscreen.

A simple example of a UI in the real world is an Automatic Teller Machine (ATM). As shown in figure 1.1, it consists of a keypad, a display window, a selection of choice options, and a help screen that displays instructions for completing an ATM transaction.



Figure 1.1: ATM User Interface

What is User Interface Design?

User Interface Design is the process of designing user interfaces for Websites, appliances, computers, and software applications. It focuses on anticipating an end user's requirement, that is, what the users might need to do and then, ensuring that the UI has all elements to facilitate those actions. The goal of UI design is to maximize user's experience and interaction.

UI design brings together the following concepts:

- **Information Architecture** focuses on organizing, structuring, and labelling content in an effective and sustainable way.
- **Interaction Design** concentrates on setting up an engaging interface with logical behaviors.
- **Visual Design** concentrates on the style of a site and its related materials through pictures, hues, text styles, and different components.

The overall goal of UI design is to make the user's experience and interaction as simple and efficient as possible.

1.2 Parts of User Interface Design

The fundamental parts or elements of most UIs are as follows:

- Input Controls
- Navigational Components
- Informational Components
- Containers
- **Input Controls**

An input control defines the way in which the system captures information and it is the most interactive component of the UI. Some common interface input controls consist of elements such as buttons, radio buttons, check boxes, drop-down lists, drop-down buttons, list boxes, and text fields.

Table 1.1 lists the input control elements and their descriptions along with examples.

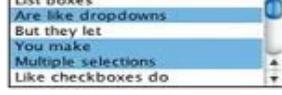
Element	Description	Example
Button	A button can be pressed or clicked by the user to perform an action. It is typically labelled using text, an icon, or both.	  Tweet by you Send an email
Radio button	Radio buttons present a set of choices from which the user can choose one.	Can you tell us why you were visiting? <input checked="" type="radio"/> Just browsing <input type="radio"/> Wanted general information <input type="radio"/> Wanted specific information
Check box	Check boxes also present a set of choices but allow the user to select one or more options from the set.	Shopping Items (Multiple Checkbox): <input checked="" type="checkbox"/> Banana <input checked="" type="checkbox"/> Apple <input type="checkbox"/> Coke <input type="checkbox"/> Pineapple
Drop-down list	Drop-down list comprise a list of items that are displayed upon clicking the arrow and enable one item at a time to be selected. They are preferred when many items in a set need to be presented to the user.	
Drop-down button	Drop-down buttons when clicked display a drop-down list of items that are mutually exclusive.	
List box	List box contains a list of options. It allows the user to select one or more items from a list.	 Image Courtesy: http://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html
Text field	Text field is a basic text control that allows the user to enter a small amount of text. When a user indicates that text entry is complete usually by pressing Enter, the text field fires an action event.	

Table 1.1: Description of Input Control Elements

- **Navigational Components**

Navigational Components provide a way for the users to tell the system what to do. Some of these include search fields, breadcrumbs, pagination, tags, icons, and image carousel.

Table 1.2 lists the navigational components and their descriptions along with examples.

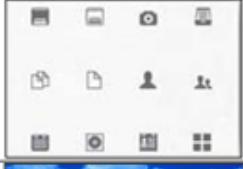
Element	Description	Example
Search Field	Search field enable users to enter a keyword or phrase and submit it with the objective of getting back the most significant results. It is generally a single-line content box that can accept user input to be searched within a database.	
Breadcrumb	Breadcrumb is a navigation aid that allows the user to keep track of user's location within programs. Breadcrumbs show up on a level plane over the highest point of a Web page, below the title bars or headers. They show links for previous page the user navigated through to get to the present page or in hierarchical site structures, the parent pages of the current one.	
Pagination	Pagination enables you to divide content into distinct pages. It allows the user to skip between pages or go in sequential order through the content.	
Tags	Tags allow users to identify content in the same classification. Some tagging framework also allows users to apply their own tags to content by entering them into the framework.	
Icon	An icon acts as a natural symbol to represent some functionality of the system, such as save or open, and so on. Icons are often hyperlinked.	
Image Carousel	An Image carousel allows users to browse through a set of items and choose one. Typically, the images are hyperlinked.	

Table 1.2: Description of Navigational Components

- **Informational Components**

Some common interface information components are tooltips, notifications, progress bars, message boxes, and modal window.

Table 1.3 lists the informational components and their descriptions along with examples.

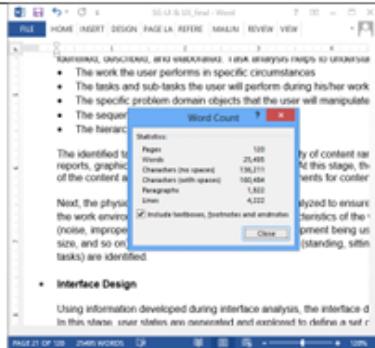
Element	Description	Example
Tooltip	A tooltip is a common UI element used in conjunction with a cursor, usually a pointer. Tooltips allow the user to see clues when they hover over an item demonstrating the name or reason of the item.	
Notification	Notifications are update message that announces something new for the user to see. Notifications are typically used to show items such as, the successful completion of a task, or an error, or warning message.	
Progress Bar	Progress bar shows where a user is as they progress through a series of steps in a process.	
Message Box	Message box is a small window that gives information to users or gets a confirmation from the user. A message box requires users to take an action before they can move forward.	
Modal Window (pop-up)	Modal window is a child window that requires users to associate with it before it can return to operating the parent application, thus, ensuring the user will perform the operation it is intended for.	

Table 1.3: Description of Information Components

- **Containers**

Containers are designed to contain page elements to a reasonable maximum width based on the size of a user's screen. The most common interface container includes Accordion.

Table 1.4 lists the most common type of container and its description along with an example.

Element	Description	Example
Accordion	An accordion is a component that show/hides a vertical list of items and content based on user action. When a label is clicked, it expands the section showing the content within.	<p>Item 1 Content #1 - Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> <p>Item 2</p> <p>Item 3</p> <p>Item 4</p>

Table 1.4: Containers

1.3 Principles and Attributes of User Interface Design

UI design principles focus on improving the quality of user interface design. Some of these include:

- **The Structure Principle**

This principle is concerned with overall UI architecture.

The design of the interface should be visually, theoretically, and linguistically clear. It should provide clear and user-specific paths to useful and relevant information.

Tips/Techniques to support the Structure Principle

The techniques that help to support this principle are as follows:

- You should group the logically connected items to communicate and separate unrelated items to achieve visual organization.
- Ensure good cross-linkages and quick jumps to important sections of the Websites.
- Design page-specific navigation and access to information.
- Keep the screen less cluttered and easier to understand.
- Present the flow of actions, information, responses, and visual preparations in a sensible order that is easy to remember and place in context.

Refer to figure 1.2 to view an example of a clearly structured UI.



Figure 1.2: Example of a Clearly Structured UI

Image Courtesy: <https://www.amazon.com>

Amazon.com is a perfect example of a clearly structured UI. It has a clean interface, less clutter, and a clear hierarchy of the content.

▪ Simplicity Principle

The design should be simple to learn and use. It should include only the elements that are most important for communication. It should also make common tasks easy to perform and provide good shortcuts that are meaningfully related to longer procedures.

Tips/Techniques to support the Simplicity Principle

The techniques that help to support this principle are as follows:

- Ensure that you create natural designs, which can be easily understood by the users based on their past experiences
- Make use of colors and fonts carefully
- Avoid acronyms and terminology with no clear explanation, which are likely to confuse the users
- Ensure that the messages and labels are written in a clear manner
- Use direct icons that are consistent with Web terminology and use less space on screen
- Make use of familiar concepts and use a language that is known to the user
- Use consistent layout, navigation, messages, labels, and representation of the information
- Do not use jarring animations and advertisements
- Use white/blank space effectively
- Avoid elements that may cause compatibility or support issues across different environments

Refer to figure 1.3 to view an example of a simple UI.



Figure 1.3: Example of a Simple UI

Image Courtesy: <http://www.apple.com/ipad/>

The Website of Apple.com is a great example to explain the simplicity principle.

The UI of Apple.com is very simple. To avoid any distraction, only the relevant content is presented in a clean and straightforward way. The clear UI design makes the Website intuitive to use.

▪ **Visibility Principle**

Visibility ensures that the user clearly sees the interface and all the possible actions. The UI design should make all required options for a given task visible without confusing the user with superfluous or redundant information. It needs to be straightforward to let users easily comprehend the interface and navigate through it more efficiently.

Tips/Techniques to support the Visibility Principle

The techniques that help to support this principle are as follows:

- Have limited entry points on the interface
- Make the entry points descriptive and 'task-oriented'
- Avoid creating UIs that will always be busy and unresponsive to users
- Do not overwhelm users with too many alternatives or confuse with unnecessary information
- Do not direct users into pages that have no navigational options
- Create levels of importance. Elements that are mission critical to the Website or application should be designed to be more visible. However, the less important elements can be designed for less visibility
- Use colors appropriately. Do not use too many colors as it may distract users

Refer to figure 1.4 to view an example of a clearly visible UI.

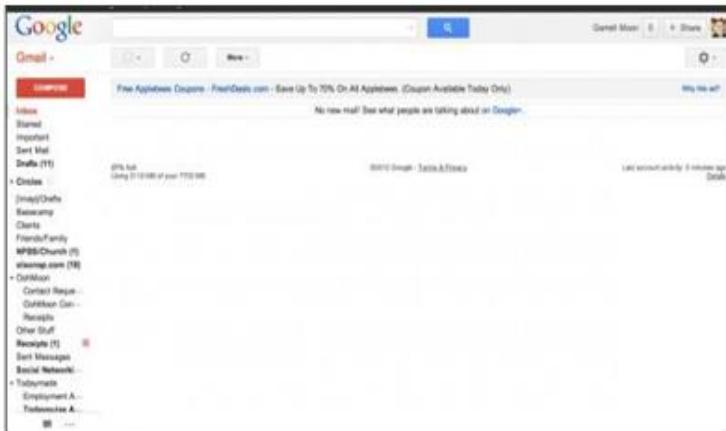


Figure 1.4: Example of a Clearly Visible UI

Image Courtesy: <http://todaymade.com/blog/wp-content/uploads/2012/04/Gmail-Inbox-garrett.t.moon@gmail.com-1-560x283.jpg>

The Website of Gmail is a great case to explain the Visibility principle.

The design of the interface is very straightforward. All the options are clearly visible. Users can quite effortlessly, and perhaps, instinctively distinguish different sections and guess what their functions are, such as Inbox, Sent items, Deleted items, Important mails, and so on. By making the button large, placing it in a noticeable position and giving it a distinct color, it stands out in the design.

▪ **Feedback Principle**

Feedback means to give information to the user based on some action. The UI design should clearly inform users what is going on and show them the result of their actions such as connecting, waiting to download, processing, or downloading. It should also inform users about the actions and changes of state or condition. Further, it should also inform about errors or exceptions that users might face on performing particular actions by using a clear, concise, and understandable language.

Feedback can come in many forms, such as a color change of an interface element, sound alerts, pop-up windows with action buttons, notification bubbles, and so on.

Tips/Techniques to support the Feedback Principle

The techniques that help to support this principle are as follows:

- Write your messages and labels clearly
- Respond to user action as close as possible to the point of interaction (that is, within a reasonable amount of time)
- Use colors appropriately

Refer to figures 1.5 (a) and 1.5 (b) to view an example of a UI displaying feedback.

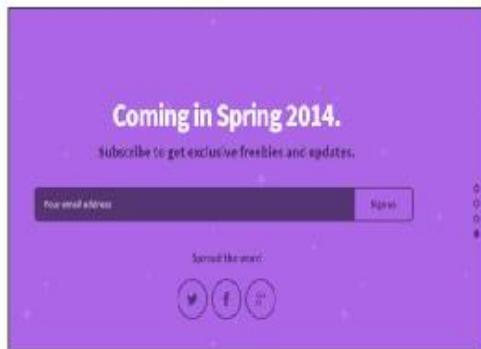


Figure 1.5 (a): UI of 'Coming Soon' Page from Kickdrop.me

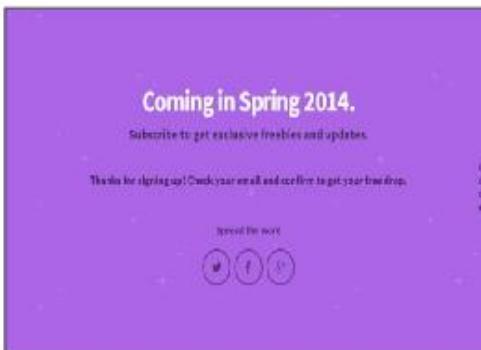


Figure 1.5 (b): Example of UI Displaying Feedback

Images Courtesy: <http://www.howdesign.com/web-design-resources-technology/donald-normans-design-principles-modern-web-design/>

The 'Coming Soon' page with the **Thanks** message from Kickdrop.me is a great example to explain the Feedback principle. To subscribe, the user has to enter the required details (in this case, an email ID) and then click the Sign-up button. When a user clicks the button, the email form slides out and a confirmation message slides into view. This grabs the user's attention and emphasizes that clicking the button has produced a result. In response to the signing up, a Thank you message is displayed.

- **Tolerance Principle**

The Tolerance UI design principle emphasizes the importance of designing the UI to prevent users from making errors. It allows the user to learn how to use the site and informs them of errors.

Tips/Techniques to support the Tolerance Principle

The techniques that help to support this principle are as follows:

- Display error messages that provide the information necessary for recovery
- Use clear and familiar language that can be easily understood by the user
- Use specific and constructive terms to avoid any ambiguity
- If the user requests, provide additional explanation during error connection

- Ensure that users never lose their work as a result of an error on their part or system or communication problems

Refer to figure 1.6 to view an example of a tolerant UI.

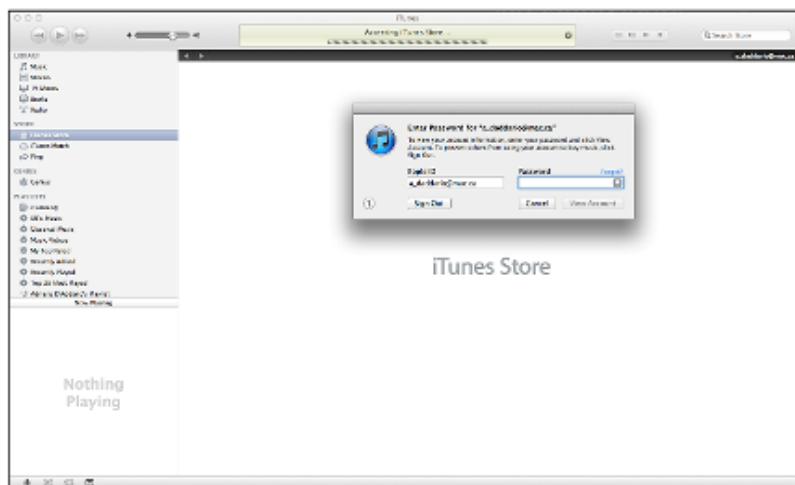


Figure 1.6: Example of a Tolerant UI

Image Courtesy: <https://adaddario16.wordpress.com/2012/06/07/principles-and-techniques-of-user-interface-design/>

The iTunes interface is an example to explain the Tolerance principle.

The iTunes interface allows the user to be notified of certain aspects. For example, when the user is keying in their password in capital letters, it notifies the user with a simple symbol. When the user is missing an attachment, another symbol such as an exclamation point comes up, that explains which file is missing or lost. Each feature tells the user that something needs to be done to fix the issue and the interface assists by offering recommendations.

▪ Reuse Principle

The UI design should reuse internal and external components and behaviors to maintain consistency with purpose. This reduces the need for users to rethink and remember. A simple example of this is the keyboard shortcuts.

The techniques that help to support this principle are as follows:

- Group elements effectively
- Ensure that the same action should always yield the same result
- Do not change the function of the elements
- Do not change the position of standard elements
- Set UI design standards and then stick to them

Refer to figure 1.7 to view an example of a reusable UI.

UI and UX for Responsive Web Design

Figure 1.7: Example of a Reusable UI

Image Courtesy: <http://www.ebay.in/>

An example of the Reuse principle is ebay.in. The process in which a user goes through to purchase a product is very simple and organized. In addition, once the user goes through it, he/ she is able to repeat the actions easily.

Quick Test 1.1

1. Input controls define the way the system captures the information. Input controls are the interactive components of the user interface.
 a. True
 b. False

2. Interaction Design avoid on creating engaging interfaces with well-thought-out behaviors.
 a. True
 b. False

1.4 Types of UIs

UIs can be classified into six categories:

- **Command Language-Based Interface**

A command language-based interface is a means of interacting with a computer program where the user issues commands to the program in the form of successive lines of text. The computer displays a prompt, the user keys in the command, and presses Enter key. After receiving the command, the command language-based interface processes it accordingly and shows the output/result on the same screen. The most common example of the command language-based interface is MS-DOS.

© Aptech Limited

- **Menu-Based Interface**

A menu-driven UI allows the user to access command through the menu. The computer displays a menu, the user makes a choice, and then, the next menu appears. The process continues until the user selects the desired option. The most common examples of a menu-based interface include ATMs, iPods, and mobile phones.

- **Natural Language Interface**

A natural language interface allows the user to speak in normal everyday language in order to interact with the system. The most common example of the natural language interface is the speech recognition software. It accepts spoken words and converts them into the text on the computer.

- **Touch Sensitive Interface**

A touch sensitive interface utilizes a touchscreen display as a combined input and output device. The most common examples of the touch sensitive interface are smartphones and the POS machines.

- **Web-Based Interface**

A Web-based interface accepts the input from the keyboard and mouse and provides output by generating Web pages, which are transmitted via the Internet. These Web pages are viewed by the user using a Web browser program.

- **Graphic User Interface (GUI)**

The most widely used type of UI in use today is the Graphical User Interface (GUI). It accepts input through the keyboard and mouse and displays the output on the user's computer screen. The actions in a GUI are usually performed through direct manipulation of the graphical elements.

The most common elements of a GUI include:

- **Window:** It is an area on the screen that displays information. Contents in a window are displayed in the form of icons or lists. According to the size of the screen, windows can be minimized, resized, or maximized. They can be moved anywhere on the screen. A window may have one or more windows as its children. These child windows are drawn within the client area of the parent window.
- **Icon:** It is a small picture that represents a software application or a hardware device. An application window opens on clicking or double-clicking these icons.
- **Menu:** It is a list of choices presented to the user. A menu is placed at a visible (generally top) inside the application.
- **Pointer/Cursor:** It is an onscreen symbol that represents movement of a device that the user controls to select windows, icons, and menus.

Refer to figure 1.8 to view the most common elements of a GUI.

UI and UX for Responsive Web Design

Figure 1.8: Elements of a Graphical User Interface (GUI)

Image Courtesy: <http://www.intel.com/content/www/us/en/company-overview/company-overview.html>

1.5 Processes in User Interface Design

The User interface development process is repetitive and can be represented using a spiral model.

The process encompasses four distinct framework activities:

- Interface analysis
- Interface design
- Interface construction/implementation
- Interface validation

Refer to figure 1.9.

Figure 1.9: User Interface Design Process

Image Courtesy: <http://csis.pace.edu/~marchese/cs615sp/L6New/Sel6new.html>

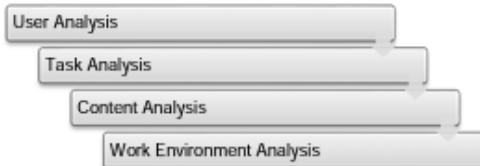
The spiral design shown in figure 1.9 implies that each UI development task will happen more than once. Each task passes around the spiral representing added requirements and the resultant design. The construction phase involves UI prototyping, which is the only practical way to validate what has been designed.

- **Interface Analysis**

The first activity of the UI development is Interface analysis.

It involves understanding the end users who will interact with the system through the interface, the tasks that end users would need to perform to do their work, the content that will be presented as a part of the interface, and the environment in which these tasks will be conducted.

© Aptech Limited



The initial analysis activity centers on the profile of the users who will work with the system. It identifies user's skills and capabilities, and level of business understanding, business process understanding and his/her role in the process. Information from a wide range of sources such as user interviews, sales input, marketing input, and support input can be used to analyze the user profile.

Once general requirements have been defined, a more comprehensive task analysis is conducted. Those tasks that the user performs to accomplish the goals of the system are identified, described, and elaborated. Task analysis helps to understand:

- The work the user performs in specific circumstances
- The tasks and sub-tasks the user will perform during his/her work
- The specific problem domain objects that the user will manipulate during his/her work
- The sequence of work tasks
- The hierarchy of tasks

The identified tasks lead to the presentation of a variety of content ranging from character-based reports, graphical displays to multimedia information. At this stage, the format and the aesthetics of the content are considered to establish the requirements for content presentation.

Next, the physical work environment of the user is analyzed to ensure that the UI products fits into the work environment. In this step, the physical characteristics of the workplace, constraints (noise, improper lightning, and so on), the type of equipment being used (mouse type, keyboard size, and so on), and how the user will execute the UI (standing, sitting, or while doing other tasks) are identified.

▪ **Interface Design**

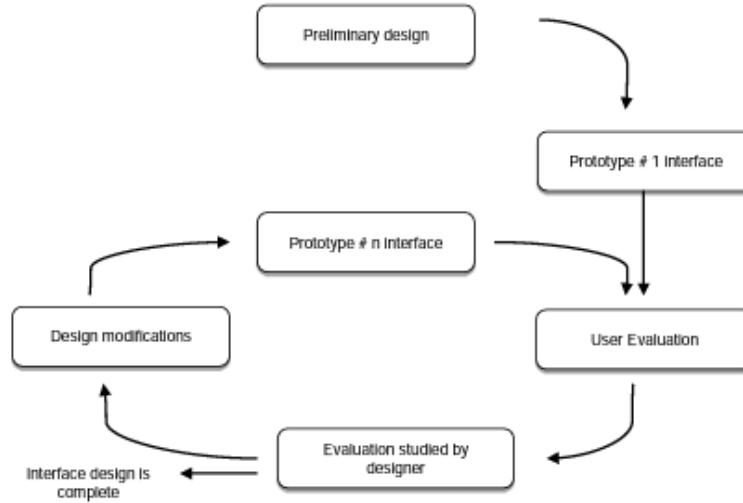
Using information developed during interface analysis, the interface design activity commences. In this stage, user states are generated and explored to define a set of interface objects and actions. These objects then serve as the basis for the creation of screen layout that illustrates graphical design and placement of icons, specification and tilting for menus and specification of menu items, and so on. As the design model is refined, design issues such as error handling, system response time in terms of length and variability, user help facilities and menu and command labelling are also considered at this stage.

▪ **Interface Construction/Implementation**

Once a design model is created, it is implemented as a prototype. The prototype enables usage scenarios to be evaluated and continue with development tools to complete the construction.

▪ **Interface Evaluation**

Once an operational UI prototype is created, it is evaluated to determine whether it meets the needs of the user. It can range from an informal test drive to a formally designed study using statistical methods. The UI evaluation cycle takes the form as shown in figure 1.10. It consists of prototype creation followed by user evaluation. The user directly informs the designer through comments about the effectiveness of the UI. Based on the user inputs, design modifications are made as required, and the next level prototype is created. The evaluation cycle continues until all user issues are resolved.

**Figure 1.10: Design Evaluation Cycle***Image Courtesy: <http://csis.pace.edu/~marchese/cs615sp/L5/Sel5.html>***Quick Test 1.2**

1. The most widely used type of user interface in use today is the Graphical User Interface (GUI).
 - a. True
 - b. False
2. The first activity of the UI development is Interface Design.
 - a. True
 - b. False

1.6 Models in User Interface Design

In designing a UI, the following four model types are especially important:

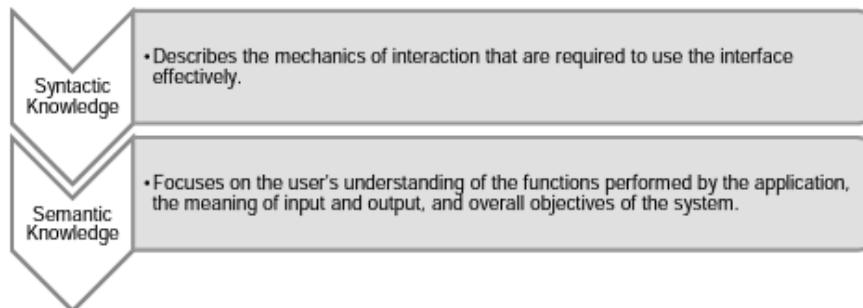
- User model
- Design model
- Mental model
- Implementation model

- **User Model**

The user model is established by a software engineer.

It describes the profile of the end users of the system. The user model is based on the age, gender, physical abilities, education, cultural, or ethnic background, motivation, goals, and personality of the end user.

The user model focuses on the syntactic and semantic knowledge of the end user.



The user model classifies the users as novices/beginners, knowledgeable intermittent users, and knowledgeable frequent users. Refer to figures 1.11 to 1.13 to understand the types of users.

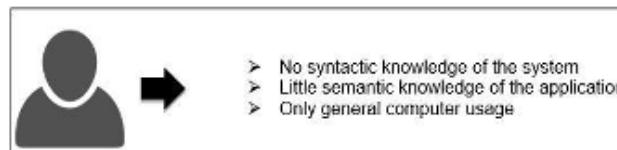


Figure 1.11: Novice

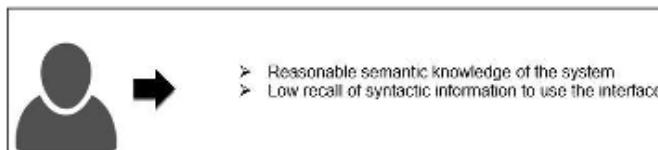


Figure 1.12: Intermittent User

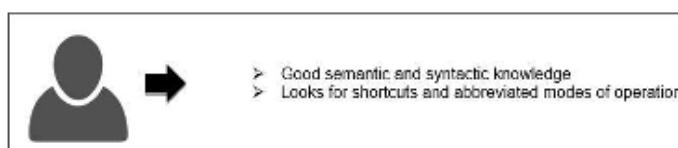


Figure 1.13: Frequent User

▪ Mental Model

The mental model is developed by the user while interacting with the system.

It is a representation of user's system perception - 'How do users think this works?' based on past experiences, knowledge, or common sense. The accuracy of the description depends upon the user's profile and overall familiarity with the system.

▪ Design Model

The Design model is created by a software engineer.

It results from the analysis phase of the requirements and takes into account the data and architectural, interface, and procedural aspects of the system. The design model must be developed to accommodate the information contained in the user's mental model.

- **Implementation Model**

The Implementation model is created by the software implementers.

An implementation model is a representation of how a system actually works. It consists of the look and feel of the interface along with all supporting information such as resources, tutorials, help files, and videos that describe system syntax and semantics.

Basically, it serves as a translation of the design model by providing an insight of the information contained in the user profile model and the user's mental model.

Note

All these models may differ significantly. However, the interface designer resolves these differences and develops a consistent representation of the user interface.

1.7 Summary

In this session, **Introduction to User Interface Design**, you learned that:

- A User Interface is the means by which a user and a computer system interact.
- User Interface Design is the design of Websites, computers, and software applications focusing on maximizing user experience and interaction.
- The fundamental parts of most UIs include Input Controls, Navigational Components, Informational Components, and Containers.
- The UI design principles focus on improving the quality of user interface design.
- The six important UI design principles are the Structure principle, Simplicity principle, Visibility principle, Feedback principle, Tolerance principle, and Reusability principle.
- The UI development process is repetitive and can be represented using a spiral model.
- The UI development process encompasses four distinct framework activities including Interface analysis, Interface design, Interface construction/implementation, and Interface validation.
- The four model types in designing a UI are User model, Design model, Mental model, and Implementation model.
- The most widely used type of UI in use today is the GUI.
- Common elements of a GUI include Window, Menu, Icons, and a Pointer.

1.8 Exercise

1. Which of the following is not a user interface design process?
 - a. User, task, and environment analysis
 - b. Interface design
 - c. Interface collaboration
 - d. Interface evaluation
2. Which of the following UI model incorporates data, architectural, interface, and procedural representations of the system?
 - a. User model
 - b. Design model
 - c. Implementation model
 - d. Testing model
3. Which model depicts the image of a system that an end user creates in his or her head?
 - a. User model
 - b. Design model
 - c. Implementation model
 - d. Mental model
4. Which of the following UI design principles emphasizes the importance of designing the user interface to prevent users from making errors?
 - a. Simplicity
 - b. Visibility
 - c. Tolerance
 - d. Reuse
5. Which user interface allows the user to interact with the computer by typing instructions?
 - a. Command line interface
 - b. Menu interface
 - c. Natural language interface
 - d. GUI
6. A small picture on a computer screen that represents a software application or a hardware device is called _____.
 - a. Menu
 - b. Button
 - c. Icon
 - d. Window

Answers to Exercise

1. Interface collaboration
2. Design model
3. Mental model
4. Tolerance
5. Command line interface
6. Icon

Answers to Quick Test

Quick Test 1.1

- a. True
- b. False

Quick Test 1.2

- a. False
- b. False

Session 2

Introduction to User Experience Design

Learning Objectives

In this session, you will learn to:

- Define User Experience (UX) and User Experience Design
- Describe the difference between UI and UX
- Describe the different elements of UX Design
- Explain the significance of UX
- List and explain the principles of User Experience Design
- Describe the best practices in User Experience Design

The success of a Website or an application depends on how users perceive it. 'Does it give me value? Is it easy to learn and use? Is it a pleasant experience? These are the basic questions that run through the users mind while interacting with any system and form the basis of their decisions on whether or not to become regular users. User experience design strives to make them answer 'Yes' to all of these questions. It focuses on providing rich and interactive experience to the users.

This session begins with defining User Experience (UX) and UX design. It explains the significant difference between UI and UX. The session also provides an overview of the different elements of UX design and explains its significance. Further, the session describes the principles and best practices of UX design.

2.1 Definition of UX and UX Design

What is User Experience?

User Experience, often abbreviated as 'UX', is the overall experience and satisfaction that a user has when interacting with a product such as a Website or computer application. Simply stated, the UX is how the user is emotionally affected by the product. For a satisfying user experience, it is important that the product is easy to use and fits the user expectations. Refer to figure 2.1.

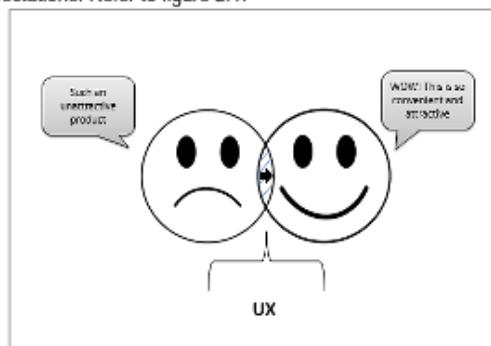


Figure 2.1: User Experience

What is User Experience Design?

User experience design, often abbreviated as 'UXD' or 'UED', is the process of understanding the needs of a user and intuitively addressing those needs by improving the product's (such as, Website or computer application) information architecture, interaction design, and visual design. The core focus of the UXD is to create a simple yet engaging experience for the user.

2.2 Difference Between UI and UX

UI and UX are often used interchangeably but these are two separate entities with completely different meanings.

User Interface (UI): UI is simply a collection of elements such as text fields, buttons, search field, layout, and links by which a user and a computer system interacts. It does not address details such as how the user remembers the system, responds to the system, and re-uses it.

User Experience (UX): It is the overall experience of using the interface - 'how it makes the user feel while interfacing with a system'. UX answers the 'why' questions behind the design layout and user interaction. The focus of the UX is to identify the users who will utilize the system and determine how exactly to meet their needs.

Here are a few examples from the real world to understand the differences between UI and UX:

- A knife is the UI, a cake is the product, and cutting the cake using the knife is the UX.
- A door knob is the UI and turning the knob to open door is the UX.
- iTunes is the UI. It helps create playlists using drag and drop feature. In turn, it makes the user experience better by allowing them to listen to playlists as per their mood.

Now, let us apply the concept to a Website. Refer to figures 2.2 (a), 2.2 (b), and 2.2 (c).

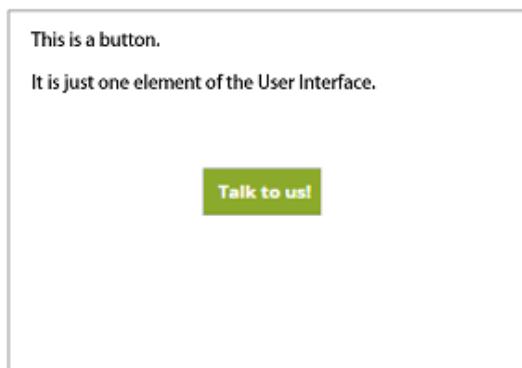


Figure 2.2 (a): Interface Element



Figure 2.2 (b): User Interface

Image Courtesy: <http://infonativesolutions.com/>



Figure 2.2 (c): User Experience

Image Courtesy: <http://infonativesolutions.com/>

Refer to table 2.1 to understand the differences between UI and UX.

User Interface (UI)	User Experience (UX)
It is a means by which a user can interact with the system. It serves as a point of interaction between user and the system.	It is the experience resulting from using a system. It is the interaction itself.
UI is only a component of the overall UX.	UX is the big picture that includes the entire process from concept to completion.
UI is tangible and can be defined easily.	UX is subjective in nature and sometimes difficult to measure.
It focuses on the visual design (look and feel) and the interaction design (how it works) of the system.	It focuses on human behavior and tries to obtain a better understanding of what the user wants to use.
It requires creative and convergent design.	It requires creative and critical design.
It focuses on the presentation the content so that it becomes easier for the user to obtain information.	It focuses on the context and provides users with the solution to their problems.
UI is the determining factor of the UX. It defines the real limits of what a user can do with the system. For example, a user can only interact with the Website through the buttons, forms, and links provided on the screen.	The UX determines the future use of the system. It indicates how helpful it is for the user and what modifications will be required in the future versions of the system.
The UI process involves determining the initial requirements and objectives for the system, organizing the information space, and selecting and applying colors and style of the UI elements. The main purpose of UI work is to clearly inform users what they need to do and how to do it.	UX design is more comprehensive. It begins with a thorough understanding of the users - what drives them and what their limitations are. Typically, the UX process involves pre-planning (data analysis, user stories, user flows), prototyping, designing, testing, and design modification to match the user expectations.

Table 2.1: Difference between UI and UX**Note**

UI and UX signify different things, but they both work hand in hand to build an effective product.

2.3 Elements of UX Design

UX Design is based around the end user's overall experience with the system. The critical elements of a good UX Design are as follows:

- Visual design
- Usability
- Information Architecture
- Interaction Design
- User Research

Refer to figure 2.3.



Figure 2.3: Elements of UX Design

➤ Visual Design

Visual design represents the aesthetics (look-and-feel) of a Website and its associated components. It utilizes the visual elements such as shapes, colors, and images to deliver a clear message to the users and thus, improve their experience with the system.

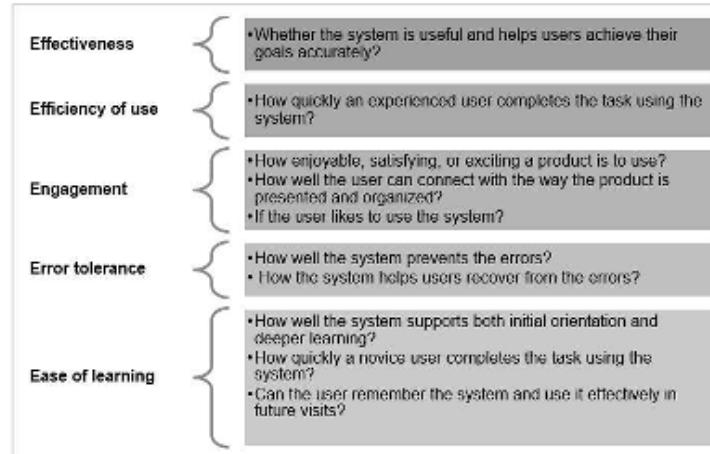
Guidelines for creating a positive visual design:

- Maintain visual consistency through the use of colors, typography, and persona
- Build hierarchies through different colors, font sizes, and placement on the page
- Incorporate white space into the layout design to increase readability
- Emphasize differences in colors, size, and direction to make the important items stand out
- Create continuity throughout the design so that the users can quickly learn the interface
- Optimally size and place the elements so that they are easy to use and indicate the importance and connection between actions
- Use metaphors to support communicating a message

➤ Usability

Usability is about how easily a product can be used by specified users to accomplish their goals.

It is important to understand that 'Usability' is not a single property of a system or a product. It is a combination of various factors including effectiveness, efficiency of use, engagement, error tolerance, and ease of learning. Refer to figure 2.4.

**Figure 2.4: Five Es of Usability****Guidelines for creating a usable design:**

- **To make the UI effective:**
 - Provide response/feedback on all critical activities
 - Remove error opportunities.
 - Provide adequate information for user to make decisions.
- **To make the UI efficient:**
 - Design navigation for ideal and alternate workflows
 - Provide shortcuts
 - Employ design and interaction styles that are faster to execute
 - Minimize unnecessary elements on the screen
- **To make the UI engaging:**
 - Use clear and familiar language that can be easily understood by the user
 - Avoid confusing set of acronyms and terminology with no clear explanation
 - Use consistent layout, navigation, messages, labels, and representation of the information
 - Use colors and fonts carefully. Do not use too many colors as it may distract users
 - Set a suitable conversational tone to help users to understand the content better
 - Make the screen less cluttered and easier to understand
 - Present the flow of actions, information, responses and visual preparations in a sensible order that is easy to recollect and place in context
 - Have good cross-linkages and quick jumps to important sections of the Websites
- **To make the UI error tolerant:**
 - Use controls that help in correct selection
 - Ensure that actions are easily reversible and the user never loses his/her work as a result of an error on their part or system or communication problems
 - Display error messages that provide the information necessary for recovery
 - If the user requests, provide additional explanation during error connection
- **To make the UI easy to learn:**
 - Create a helpful interface. Provide minimalist prompts and instructions wherever required
 - Set UI design standards and then stick to them
 - Construct 'guided/directed' interfaces for challenging or uncommon tasks.
 - Ensure that the same action should always yield the same result.
 - Do not change the function and position of the standard elements.

➤ **Information Architecture**

Information architecture focuses on organizing, structuring, and labelling content in an effective and sustainable way, so that the users can find the relevant information and complete their tasks easily.

Guidelines for creating effective information architecture:

- Design information architecture from end users' perspective - keeping user's requirements and preferences in mind
- Use consistent layouts to help users easily navigate through the site
- Organize the content in a structured manner
- Use concise, clear, consistent, and descriptive labels for navigation items and links
- Maximize findability through proper navigation logic and detail in the descriptions
- Provide navigational hints to let users know their location within the site, how to get back, and how to jump back to the start
- Enable users to navigate to the most important content and functionality in as few clicks as possible
- Present links to the main features and content on the landing page, prioritized according to the user's needs

➤ **Interaction Design**

Interaction design is a user-centric approach of designing an interactive system focused around end users - their goals, experiences, what they need, and how they expect the system to work.

Guidelines for designing interactions:

- Try to match user's previous experiences and expectations by using common conventions or UI patterns
- Use consistent layout, navigation, terminology, labels, and representation of the information to help users learn the system more quickly
- Avoid excessive features and functions. Also, divide difficult tasks into manageable sub-tasks
- Offer visual hints or text messages to users about when to start an interaction
- Provide text messages or visual clues while a request is being processed to let the user know what is happening within the system
- Use clear and easily distinguishable icons to improve perceptibility of interaction
- Group the logically connected items to communicate and separate unrelated items to achieve visual organization
- Design generic interactions so that user can replicate the experience across multiple interactions
- Make the interactions intuitive so that the user can easily predict how to use it

➤ **User Research**

User research is the most critical component in any UXD initiative. It focuses on understanding user needs, behaviors, expectations and motivations through one-on-one interviews, research, surveys, user observation, and other feedback methodologies.

Through user research, you can design a UI that supports and helps user behavior in a way that users will feel it natural and effective.

Guidelines for carrying out user research:

- Allow users to organize the information into different categories. This helps to understand their expectations and thus, enables to create the site structure in the way users think
- Observe the users in their natural work environment and ask questions as they navigate the site. This helps to understand the way they perform their tasks and accomplish their intended goals
- Develop user personas to identify and prioritize their specific goals
- Conduct a moderated discussion to learn about users' attitudes, views, needs, and responses to concepts
- Build prototypes and gather feedback from users while still at design stage
- Carry out online surveys to collect information from a broad audience
- Evaluate a product by testing it with representative users

Quick Test 2.1

1. UI and UX are not interchangeably and these are two separate entities with completely different meanings.
 - a. True
 - b. False
2. UX is the big picture that includes the entire process from concept to completion.
 - a. True
 - b. False

2.4 Significance of UX

A good user experience plays a crucial role in the effectiveness of a Website design. It helps the user to easily navigate the site and understand how to use it. A useful, pleasurable, comprehensive, desirable, and quick UX develops an interaction-rich experience that drives the users back to the site.

On the other hand, poor user experiences create a negative emotional connection to the site for users.

Simply stating, without an effective UX, products such as Websites, apps, or software are likely to fail.

2.5 Principles and Guidelines of UX Design

The key principles for creating engaging user experiences are as follows:

➤ **Familiarity**

A user should be able to recognize the UI components and view the interaction as similar to the ones they have interacted before. Note that users do not spend time in exploring or admiring the UI. They prefer a familiar design where they can easily identify the features based on their past experiences and can directly locate the ones they require. Refer to figure 2.5.



Figure 2.5: Example to Demonstrate Familiarity Principle

Image Courtesy: <http://www.8164.org/familiarity-learnability/>

News/blog sites are a great example to explain Familiarity principle.

These sites share the same layout and navigation. Layout is typically such as a newspaper, and navigation is generally placed on the left or top. To improve the readability, article headers are usually written in bigger font size. The familiarity with the design helps the users to access the Website easily without having to think too much about the functionality.

➤ Clarity

Clarity plays an important role in creating seamless UX design. It focuses on arranging various elements on the Web page to maximize the user's chance of using the site effectively.

A clear design provides answer to three basic questions:

- What is it?
- What can I do here?
- Why should I do it/how it is useful for me?

Refer to figure 2.6.



Figure 2.6: Example to Demonstrate Clarity Principle

Image Courtesy: <http://conversionxl.com/5-principles-of-persuasive-web-design/>

Square reader Website is a great example to demonstrate the clarity principle.

It answers all the three basic questions:

- **What is it?** : It is a credit card that starts accepting credit cards on the very first day of purchase. A clear and relevant image supports the context.
- **What can I do here?** : A user can get a credit card reader here.
- **Why should I do it/how is it useful for me?** : The card reader is free. A user just has to pay 2.75% per swipe for Visa, MasterCard, Discover, and American Express.

➤ **Recoverability**

Recoverability emphasizes that the user actions should be reversible. The design should allow the users to undo the wrong actions without any other negative repercussions. In addition, when a user makes a mistake, the design should guide the users about how to proceed further. The user should never be left at a 'dead-end' in the Website or app, where their next step is not clear and they have to figure out what to do next. Refer to figure 2.7.

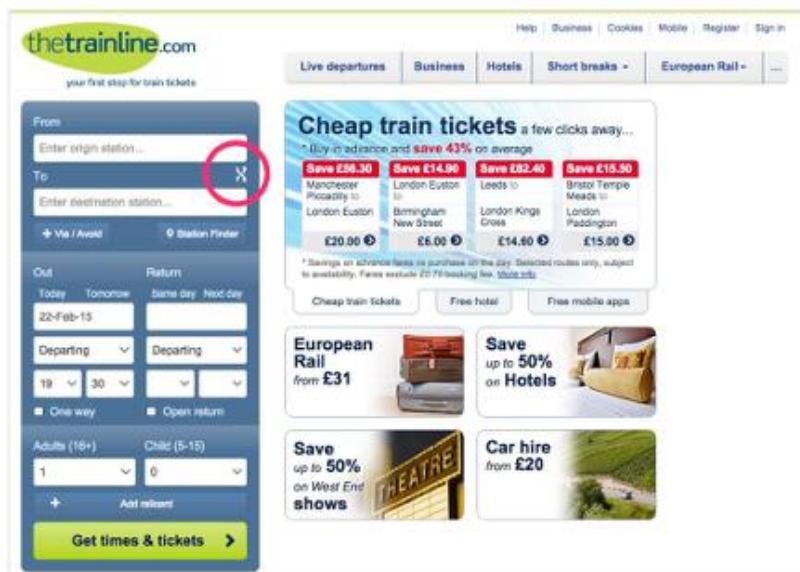


Figure 2.7: Example to Demonstrate Recoverability Principle

Image Courtesy: thetrainline.com

The trainline.com site is a good example to demonstrate recoverability principle. It offers a useful button (highlighted in figure 2.7) to allow users to change the direction of their journey.

➤ Responsiveness and Feedback

The UI design should be quick to respond and there should be no lag time in loading. It should provide helpful information to the user about the task in hand. Also, the UI needs to provide appropriate and timely feedback to specify that the action is happening and whether it is successful or unsuccessful.

Refer to figure 2.8.



Figure 2.8: Example to Demonstrate Responsiveness and Feedback Principle

Image Courtesy: <http://www.makemytrip.com/>

Make my trip (travel site) is a good example to explain responsiveness and feedback principle. A clear and immediate feedback is shown to users in case they select any wrong option in the form. This helps them to easily rectify the problem and move onto the next step.

➤ **Simplicity**

The design should be simple to understand and simple to use irrespective of the user's experience, verbal skills, knowledge, or current concentration level. It incorporates only the components that are most important for communication. It should also make common tasks simple to perform and provide good shortcuts that are meaningfully related to longer procedures.

Refer to figure 2.9.

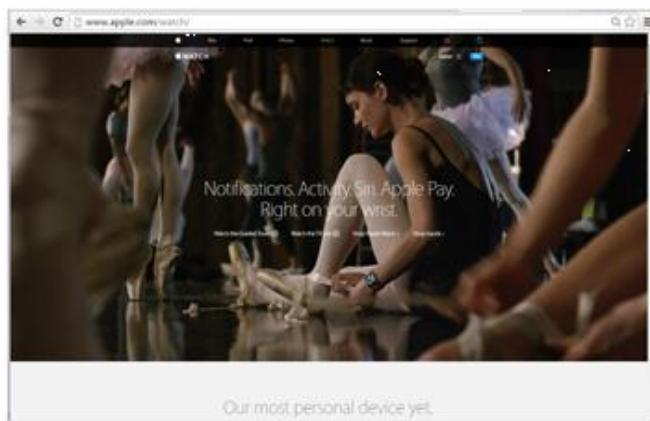


Figure 2.9: Example to Demonstrate Simplicity Principle

Image Courtesy: <http://www.apple.com/watch/>

© Aptech Limited

Apple.com is a great case to explain the Simplicity principle.

The UI of Apple.com is extremely simple. Only the relevant content is presented in a clean and straightforward way to avoid any distractions. The clear UI design makes the Website just as intuitive to use as all other Apple products.

➤ Content Delivery

The UI should provide a well-timed and relevant content to the users. This helps to increase in a tangible manner the user's satisfaction. Refer to figure 2.10.



Figure 2.10: Example to Demonstrate 'Content Delivery' Principle

Image Courtesy: <https://www.amazon.com>

'Customers Who Bought This Item Also Bought' function on Amazon.com is a perfect example of providing immediate and relevant information to the users.

➤ Delight

The UI design should be intuitive, simple, and attractive to delight the user and make an emotional connect with them. It should be thorough and accurate. Also, the fonts, color palette, graphics, and animation should be used wisely to provide a better and more usable experience. Refer to figure 2.11.

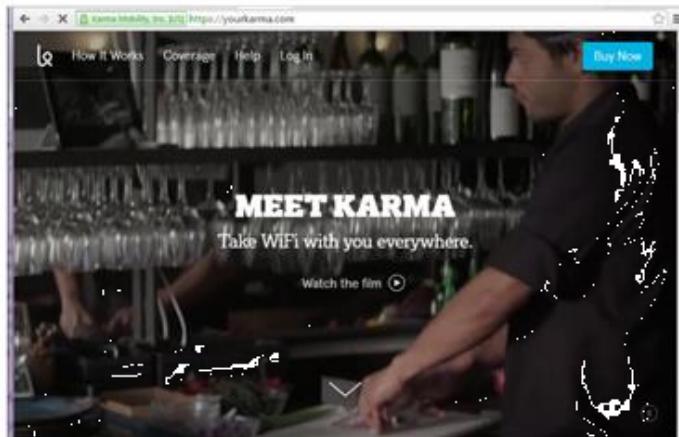


Figure 2.11: Example to Demonstrate 'Delight' Principle

Image Courtesy: <https://yourkarma.com/>

The Website for the Your Karma mobile Wi-Fi device is the perfect example of captivating user experiences. The mixture of great fonts, delicate shading palette, smart illustrations, and profound animation makes for a really nice UI.

Quick Test 2.2

1. Without an effective user experience, products such as Websites, apps, or software are likely to fail.
 - a. True
 - b. False

2. Clarity focuses on arranging various elements on the Web page to maximize the user's chance of using the site effectively.
 - a. True
 - b. False

2.6 Best Practices in UX Design

Some of the best practices for designing and creating amazing user experiences are as follows:

➤ **Connect the Goals:**

Clearly identify business goals and user goals to understand their expectations. Then compare the goals to find any differences and build appropriate design solution.

➤ **Take Inputs:**

Take periodic inputs from the stakeholders to understand their perspective. Involve stakeholders at the kick off stage, wireframe rounds, usability testing and finally at the launch to take their feedback and incorporate suggestions to the design.

➤ **Design Collaboratively:**

Collaborate with other team members/designers to improve the overall design of the system.

➤ **Build Personas:**

Create personas by interviewing various individuals in the same position, and compiling their traits, desires, limitations, and beliefs into an imaginary user. This helps to identify the goals of the target users and prioritize features as per their requirements. Refer to figures 2.12a and 2.12b.



Figure 2.12a: Example of Personas

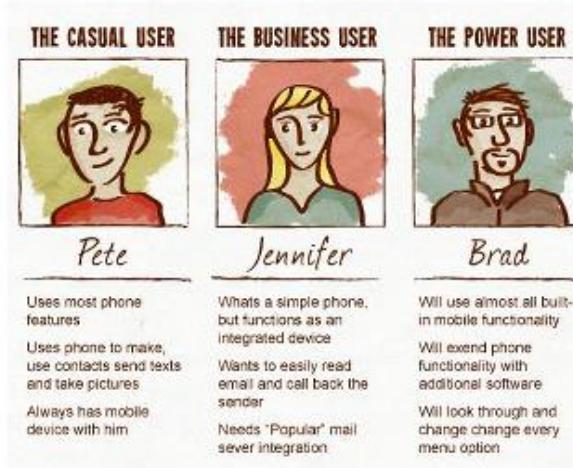


Figure 2.12b: Example of Personas

Image Courtesy: <http://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1/>

<http://frontdigital.com/category/user-experience-ux/>

➤ **Create Paper Prototypes:**

Create a rough prototype showing all interactive pieces on a paper. Now, ask a person (if possible, a person who fits the target end user) to act as the user and simulate actions and interactivities. Carefully notice his/her successes and struggles to get an immediate feedback. Refer to figure 2.13.

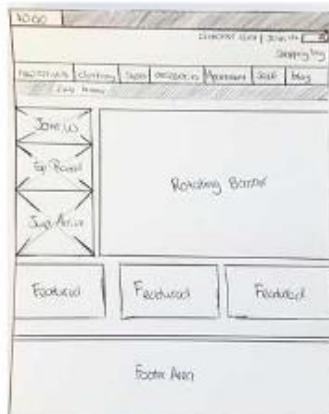


Figure 2.13: Example of a Paper Homepage of a Website

Image Courtesy: <http://www.leanenterprisetoronto.com/blog/paper-prototypes-and-discounting-value-usability-testing>

➤ **Research UI Patterns:**

Research UI design patterns for getting inputs, navigation, dealing with data and other activities and select the preferable pattern from the user's experience perspective. Refer to figure 2.14.

Getting input	Navigation	Dealing with data
Forms WYSIWYG Password Strength Meter Input Feedback Captcha Calendar Picker Inplace Editor Good Defaults Fill In The Blanks Live Preview Forgiving Format Input Prompt Structured Format Explaining the process Wizard Blank State Tour Inline Help Box Steps Left Community driven Lazy Registration Vote To Promote Account Registration Rate Content	Tabs Navigation Tabs Module Tabs Jumping in hierarchy Shortcut Dropdown Home Link Fat Footer Breadcrumbs Menus Horizontal Dropdown Menu Vertical Dropdown Menu Accordion Menu Content Carousel Archive Article List Event Calendar Pagination Continuous Scrolling Tag Tag Cloud Thumbnail	Tables Alternating Row Colors Sort By Column Formatting data Color Box Images Slideshow Gallery Image Zoom Search Autocomplete Live Filter Table Filter

Figure 2.14: Examples of UI Design Patterns*Image Courtesy: <http://ui-patterns.com/patterns>*

➤ **Use Tabs Effectively:**

Use tabs effectively to alternate between views within the same context and not to navigate to different areas. Refer to figure 2.15.

**Figure 2.15: Proper Usage of Tabs***Image Courtesy: <http://userexperiencedesigns.com/>*

➤ **Reduce Clutter:**

Make the screen less cluttered and easier to understand. Consider using tabs, drop-down menu, or another way of displaying only the most helpful information and tools at any one time.

➤ **Create Emotive Design:**

Create an emotive design to engage users in a focused way. Keep the design simple and aesthetically attractive. Also, make it highly functional, easy to comprehend, and intuitive to navigate.

➤ **Take Advantage of Web Typography:**

Consider using Web typography, such as Typekit and Google Web Fonts, to improve the aesthetics of the Website or application. Refer to figure 2.16.



Figure 2.16: Example of Web Typography

Image Courtesy: <http://blogs.adobe.com/creativecloud/creative-cloud-for-desktop-bringing-the-best-fonts-to-your-desktop/>

➤ **Improve Readability:**

To improve the readability of the Website, do not use too much of text. Restrict text to maximum 50-60 characters per line. Use bullet points wherever possible to help users to get the main idea about the content while skimming through it. Also, use a bold interface to make text easily scannable.

➤ **Use Simple and Familiar Language:**

Employ familiar concepts and use a language that is known to the user.

➤ **Anticipate Errors:**

Consider the possible errors a user might encounter while navigating the site and then create a response for that error to help the user get back to normal.

➤ **Keep Menus and Navigation Simple:**

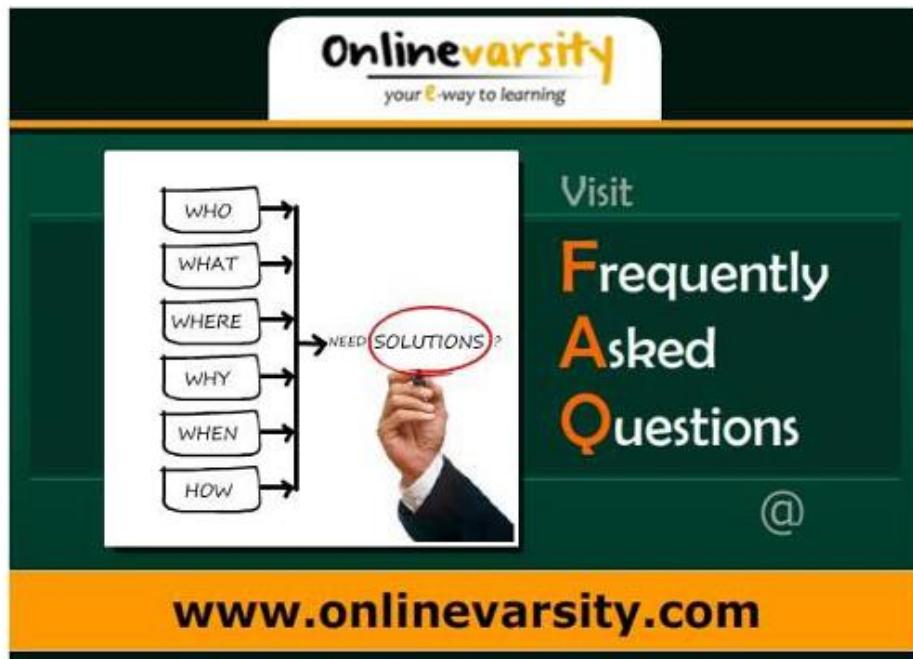
Avoid multi-level menus as much as possible. Use direct icons that are consistent with Web terminology and use less space on screen. Also, avoid elements that cause issues across different environments.

➤ **Highlight Real-time Changes:**

Provide visual cues to draw attention to the updated information. Consider changing the background color around the updated text, changing the font color or font size and/or adding in some animations to highlight the updated information.

➤ **Test and Optimize Usability:**

Continually test and improve UX throughout the design process. Keep the feedback loop open and continuous.



2.7 Summary

In this session, **Introduction to User Experience Design**, you learned that:

- UX is the overall experience and satisfaction a user has when interacting with a product such as a Website or computer application.
- UX Design is the process of understanding the needs of a user and intuitively addressing those needs by improving the product's (such as, Website or computer application) information architecture, interaction design and visual design.
- UI and UX are often used interchangeably but these are two separate entities with completely different meanings.
- The critical elements of a good UX Design are Visual design, Usability, Information architecture, Interaction design, and User research.
- The key principles for creating engaging user experiences include Familiarity, Clarity, Recoverability, Responsiveness and Feedback, Simplicity, Content delivery, and Delight.

2.8 Exercise

1. Which of the following statement is correct?
 - a. UI is only a component of the overall UX.
 - b. UX is the big picture that includes the entire process from concept to completion.
 - c. Both a and b
 - d. None of these
2. A good user experience plays a crucial role in the effectiveness of a Website design.
 - a. True
 - b. False
3. Which of the following statement(s) is correct?
 - a. User Interface serves as a point of interaction between user and the system.
 - b. User Experience is the interaction itself.
 - c. UX is the determining factor of the UI.
 - d. UI is tangible and can be defined easily.
4. What is the core focus of a User Experience Design?
 - a. To create visually appealing interface.
 - b. To create a simple yet engaging experience for the user.
 - c. To provide a means by which a user can interact with the system.
 - d. To improve the speed of the system.
5. The Five E's of Usability does not include:
 - a. Effectiveness
 - b. Efficiency
 - c. Evaluation
 - d. Error tolerance

Answers to Exercise

1. Both a and b
2. True
3. a, b, and d
4. To create a simple yet engaging experience for the user.
5. Evaluation

Answers to Quick Test

Quick Test 2.1

- a. False
- b. True

Quick Test 2.2

- a. True
- b. True

Onlinevarsity
your e-way to learning

GROWTH
RESEARCH
OBSERVATION
UPDATES
PARTICIPATION

www.onlinevarsity.com

Session 3

Understanding Responsive Web Design

Learning Objectives

In this session, you will learn to:

- Define Responsive UI design
- Describe the evolution of RWD
- Describe the importance of Progressive Enhancement
- Outline the differences between Graceful Degradation and Progressive Enhancement
- Explain the RWD Workflow
- Describe the significance of RWD

This session explains the basics of what it means for a Website to be responsive. This session will also take a look at the history of Web design, so that you can comprehend where the idea of responsive design came from. The session will also look at why responsive design is usually the best choice for making Websites that will work well across different devices and screen sizes and how it means less work in the long run.

This session explains the meaning of Responsive UI design and an overview of the evolution of RWD.

The session also explains the importance of Progressive Enhancement and also explains the difference between Graceful Degradation and Progressive Enhancement and guides you through in choosing the right method.

3.1 What is Responsive UI Design?

Responsive Design is a design philosophy where the design of the system responds depending upon the layout of the device. Increasing the reach of your application to a larger user base using an array of devices is the primary reason to keep design responsive.

Responsive Web Design (RWD) is a part of Responsive UI Design and implies creating the Website design in a way that is most optimal for viewing and exploring on a wide range of devices, including traditional PCs, smartphones, and tablet devices.

3.2 Definition of Responsive Web Design (RWD)

RWD is a way of laying-out and coding a Website so that the Website can provide an excellent viewing experience — an ease of reading and navigation with a minimum of resizing, panning, and scrolling across devices.

RWD is about using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen. Refer to figure 3.1.

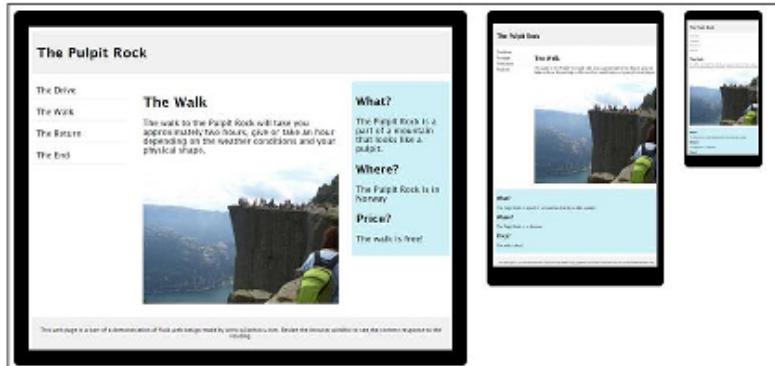


Figure 3.1: Example of RWD

Image Courtesy: <http://www.w3schools.com/>

Responsive Websites offer:

- Smooth navigation
- Easy reading
- Minimum pinching
- Reduces scrolling and zooming
- Excellent user experience

Advantages to the Users

The use of smartphones, tablets, and notebooks for accessing Internet is now quite common. Responsive sites attract more users as they serve the following advantages:

- A responsive Website is flexible to use
- Shuffles content, resizes images, and adjusts font size
- Allows users to read information as per their choice and needs
- Helps in finding fast and intelligent sites
- Saves time for users while browsing the site
- Helps to increase user's experience

Advantages to the Web Designers

Responsive Websites have changed the entire outlook of Web industry. The concept of responsive design offers benefits to the Web designers and the company at large. Following are the advantages to the Web Designers:

- Streamline the designing process
- Saves time and efforts
- Cut down capital employed
- Eliminates the use to maintain multiple Websites
- Reduce maintenance and development cost
- Expand Return on Investment in long run
- Enhance SEO rank
- Better performance means better sales
- Higher conversion rates
- Increasing market share

RWD makes use of media queries, to prepare the page to apply different CSS3 style rules, fluid grids and flexible images, so that the Website can adjust itself for a range of mobile devices and their screens.

Media queries, a mark of Cascading Style Sheets (CSS), help the developer to point out when a certain style takes effect. With CSS2, a media query will provide printer-friendly style sheets if needed. CSS3 has developed query capabilities that allow style sheets to be directed to a device's display and serve a desktop, tablet, or smartphone template depending on the query answer.

This facility implies that as an option of having to frame an exclusive mobile version of a Website, developers can simply build different style sheets for the same Web page and perhaps even associate multiple images with each of the style sheets. HTML code can be repurposed rather than rewriting, which saves extensive development time.

Refer to figure 3.2 to view an example of a CSS.

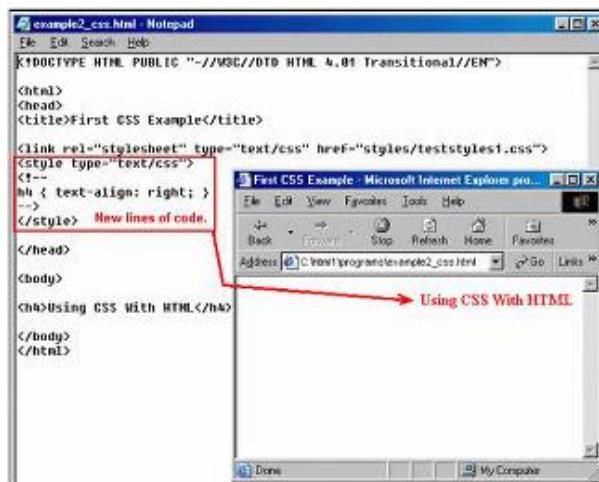


Figure 3.2: Example of a CSS document

Create Your Own Responsive Design

One of the ways to build a responsive design is to create it yourself:

Example

```
<html lang="example">
<head>
<style>
.city {
  float: left;
  margin: 5px;
  padding: 15px;
  width: 300px;
  height: 300px;
  border: 1px solid black;
}
</style>
</head>
<body>

<h1>Responsive Web Design Demo</h1>

<div class="city">
<h2>London</h2>
<p>London is the capital city of England.</p>
<p>It is the most crowded city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital city of France.</p>
<p>Paris is one of the largest population centers in Europe,
with more than 12 million inhabitants.</p>
</div>

<div class="city">
<h2>Tokyo</h2>
<p>Tokyo is the capital city of Japan.</p>
<p>Tokyo is one of the most populous metropolitan areas in the world.</p>
</div>

</body>
</html>
```

Using CSS

Another way to build a responsive design is to use a responsive style sheet. For example, the responsive style sheet used in the following example makes it simple to develop sites that can look good at any size: desktop, laptop, tablet, or phone.

CSS DEMO**Resize this responsive page!****London**

London is the capital city of England.

It is the most crowded city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Paris

Paris is the capital city of France.

Paris is one of the largest population centers in Europe, with more than 12 million inhabitants.

Tokyo

Tokyo is the capital city of Japan.

Tokyo is one of the most populous metropolitan areas in the world.

On resizing to smaller size, the page will look as follows, thus implementing responsiveness:

CSS DEMO**Resize this responsive page!****London**

London is the capital city of England.

It is the most crowded city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Paris

Paris is the capital city of France.

Paris is one of the largest population centers in Europe, with more than 12 million inhabitants.

Tokyo

Tokyo is the capital city of Japan.

Tokyo is one of the most populous metropolitan areas in the world.

The HTML code to implement this is as follows:

```
<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="./css/style.css">
<body>

<div class="container orange">
  <h1>W3.CSS Demo</h1>
  <p>Resize this responsive page!</p>
</div>
<div class="w3-row-padding">
<div class="city">
  <h2>London</h2>
  <p>London is the capital city of England.</p>
  <p>It is the most populous city in the United Kingdom,
  with a metropolitan area of over 13 million inhabitants.</p>

```

```
</div>
<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
  <p>The Paris area is one of the largest population centers in Europe,
  with more than 12 million inhabitants.</p>
</div>
<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
  <p>It is the center of the Greater Tokyo Area,
  and the most populous metropolitan area in the world.</p>
</div>
</div>
</body>
</html>
```

The responsive stylesheet, style.css, is defined as follows:

```
*{-webkit-box-sizing:border-box;-moz-box-sizing:border-box;box-sizing:border-box}
html{-ms-text-size-adjust:100%;-webkit-text-size-adjust:100%}
body{margin:0}

/* End extract from normalize.css */
html,body{font-family:Verdana,sans-serif;font-size:15px;line-height:1.5}
h1,h2,h3,h4,h5,h6{font-family:"SegoeUI",Arial,sans-serif;font-weight:400;margin:10px 0}
h1 a,h2 a,h3 a,h4 a,h5 a,h6 a{font-weight:inherit}
h1{font-size:36px}h2{font-size:30px}h3{font-size:24px}h4{font-size:20px}h5{font-
size:18px}h6{font-size:16px}
hr{height:0;border:0;border-top:1px solid #eee;margin:20px 0}

table{border-collapse:collapse; border-spacing:0; width:100%; display:table}
table,th,td{border:none}

.container:after,.w3-row-padding:after{content:"";display:table;clear:both}

.city{float:left; width:100%}

.city{width:99.99999%}

@media only screen and (min-width:601px){
.city{width:33.33333%}
}

@media only screen and (min-width:993px){
.city{width:33.33333%}
}

.container{padding:1px 16px}
.w3-row-padding{padding:0 8px}

.orange{color:#000 !important;background-color:#fffcf89 !important}
```

3.3 Evolution of RWD

Cameron Adams in 2004 was the first to demonstrate a site layout example that adapts to browser viewport width. In a May 2010 article in *A List Apart*, Ethan Marcotte coined the term Responsive Web Design (RWD) and defined it to mean fluid grid/flexible images/media queries. In his brief 2011 book titled *Responsive Web Design*, he described the theory and practice of Responsive Web Design. In Ethan Marcotte's article, he created a sample Web design that features this better flexible layout.

Refer to figure 3.3 for an example of Responsive Web Design.



Figure 3.3: Example of Responsive Web Design

Image Courtesy: <http://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>

Over the last two years, the number of mobile devices has grown at rapid pace. Due to this growth, providing a first class mobile Web experience has become very important. Union Square Ventures' Fred Wilson has gone on to point out that new Web apps are commonly designed as 'mobile first, Web second'.

'Responsive Design' has become an increasingly popular way to resolve 'mobile with desktop' Web design problems.

3.4 Fundamental Techniques for RWD

Responsive Web Design implementation is based on some key features. The features, which need to be addressed to make a design responsive, are as follows:

- **Fluid, Proportion-based Grids**

The concept of fluid grid is to arrange the grid columns in proportion based on page element sizing. It should be in proportionate units such as percentage, instead of absolute units such as pixels or points. Proportion-based grids are otherwise called as flexible layouts. To declare typical grid, the relative lengths and units are used.

➤ Flexible Images

Define image dimensions in relative units so that it always displays within the size available. Images displaying outside their containing screen element area can be prevented and can set a max-width of 100% on all images on Website.

➤ CSS3 Media Queries and Screen Resolutions

CSS3 provides easy media query support and media queries and enables the Websites pages to utilize diverse CSS styles sheet based on media rule. The width of the viewport or device orientation such as landscape or portrait are some of the key parameters used while applying media queries.

Following images are examples of a Responsive Web Design approach that uses media queries. Figures 3.4 and 3.5 shows a desktop using Internet Explorer 9 with different resolutions. Figure 3.6 shows the same site on a Windows Phone, also with Internet Explorer 9.

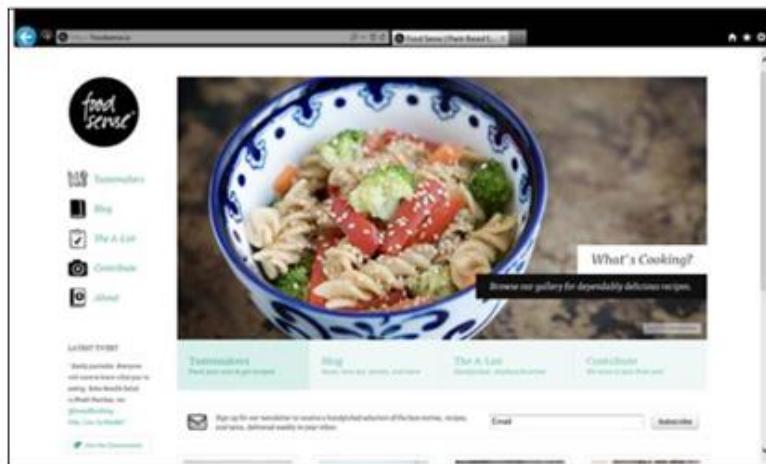


Figure 3.4: Navigation Appears on the Left

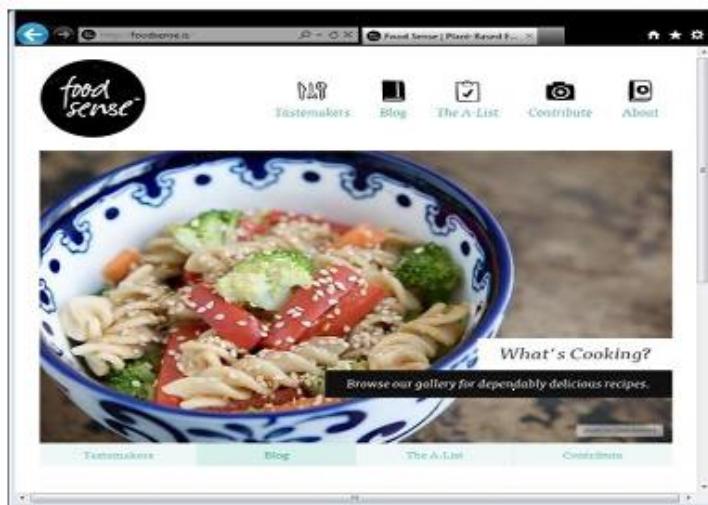


Figure 3.5: In an 800x600 Resized Window, Navigation Switches to the Top



Figure 3.6: Same Site on a Windows Phone

Image Courtesy: <https://msdn.microsoft.com/en-us/magazine/hh653584.aspx>

Quick Test 3.1

1. HTML code cannot be repurposed rather than rewriting, which saves extensive development time.
 - a. True
 - b. False
2. CSS3 provides easy media query support and media queries enables the Websites pages to use different CSS style sheets based on media rule.
 - a. True
 - b. False

3.5 Progressive Enhancement

Progressive Enhancement (PE) is a powerful technique that enables to build the best possible Websites while adjusting issues that may occur while being accessed by multiple unknown user-agents. The term, Progressive Enhancement, was coined in 2003, several years before RWD emerged.

A Web design strategy wherein designers can create Web pages for latest browsers that can also work well in earlier versions of browsers is called Graceful Degradation.

Graceful Degradation is the journey from complexity to simplicity, while Progressive Enhancement is the journey from simplicity to complexity. Progressive Enhancement is considered a better technique than Graceful Degradation because it covers a greater range of potential issues as a baseline.

Concept of Progressive Enhancement

In practical terms, it is easiest to break the concept of Progressive Enhancement into different layers, each one building on the previous to enhance the experience of interacting with the Website.

Layers

The first layer is semantic HTML. It allows text and speech based, outmoded and robotic user-agents to properly navigate the content of your Website.

The second layer is CSS. It allows visual-based user-agents to show or modify the visual representation of Website's content.

The third layer is JavaScript. It allows user-agents that are capable of using it to provide users with improved usability.

Practical Example

The ultimate goal for users is to have a drag-and-drop experience that saves the menu order via AJAX. All user-agents should enable user to interact with a list in the way most appropriate to them.

Following example demonstrates this.

- **First layer**

In the first layer, there is the semantic markup of the navigation. Refer to figure 3.7: *navigation-1.html*.

```

<form action="record.php" method="post">

    <fieldset>Record of Navigation
    <ol>
        <li id="homepage-12">Homepage <label for="menu-id-12">Change the
            order for Homepage</label><input type="text" name="homepage-12" id="menu-
            id-12" value="1" /></li>
        <li id="contact-23">Contact Us <label for="menu-id-23">Change the
            order for Contact Us</label><input type="text" name="contact-23" id="menu-
            id-23" value="2" /></li>
        <li id="about-16">About Us <label for="menu-id-16">Change the order
            for About Us</label><input type="text" name="about-16" id="menu-id-16"
            value="3" /></li>
        <li id="latest-14">Latest News <label for="menu-id-14">Change the
            order for Latest News</label><input type="text" name="latest-14" id="menu-
            id-14" value="4" /></li>
    </ol>
    </fieldset>
    <p><input type="submit" value="Record New Order" /></p>
</form>

```

The screenshot shows a web form titled 'Record of Navigation'. It contains an ordered list of four menu items with corresponding input fields for changing their order:

- 1. Homepage Change the order for Homepage
- 2. Contact Us Change the order for Contact Us
- 3. About Us Change the order for About Us
- 4. Latest News Change the order for Latest News

Below the list is a single button labeled 'Record New Order'.

Figure 3.7: navigation-1.html

The navigation is arranged in ordered list within a form so that the user can modify the order using the input boxes provided. The individual list items are provided with an ID that corresponds to the name of the input box. The labels for the input fields help the user to figure out how to use the form.

Upon clicking 'Record New Order', the form will post to the PHP page, which will implement the new ordering and print the results. The page might not be attractive to look at, but the important thing is that every element is in place to allow any user-agent to read and comprehend the structure. It is designed such that screen readers should be able to process this form easily, and keyboard navigation is functional and implemented by default.

record.php page is very simple and just echoes the order of the menu items sent through the POST:

```

<?php
    foreach($_POST as $menu_item => $order){
        echo "The order for $menu_item is $order";
    }
?>

```

Note that this is not a production script. It simply explains that the procedure for converting a form should be one of the first interactions in Progressive Enhancement.

- **Second layer**

The second layer is added to give the form a bit of visual elegance.

Take the labels out of the document flow with `position: absolute` and then give them negative margins to remove them visually for screen readers.

Erase the numbers from the ordered list to avoid the confusion of having two sets of numbers for the list items.

Manage the font size equal to whatever the user-agent has set as preference and enable the user-agent to utilize its own default font.

Provide the individual list of elements an outline and some padding so as to view them more easily. Also, give a hover state, which can be used by any user-agent that can handle it. This will give the user a visual hint that the list items are movable. Also, input boxes are arranged, for browsers that can handle attribute selectors. The form will look more neat and remains just as functional as the first layer.

Refer to figure 3.8 for `navigation-2.html` file.

```
<style type="text/css">
form {width: 50%;margin: 0 auto;}
fieldset {background: #ccffff;padding: 1em;}
legend {border:1px #513939 solid;background: #FAFAFA;}
label {position: absolute; margin-left: -999em;}
ol {list-style: none;position: relative;}
body {font: 100% serif;}
ol li {border: 1px #FFF solid;background: #00caca;padding: 0.7em;}
ol li:hover {border: 1px #513939 solid;}
input[type='text'] {width: 2em;text-align: center;position: absolute;left: 40%;}
</style>
```

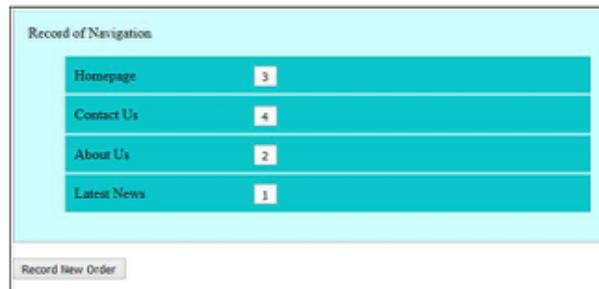


Figure 3.8: navigation-2.html

- **Third layer**

Finally, add the JavaScript layer, which allows the user to simply drag and drop the navigation items according to the way they need. Use jQuery to make the process as painless as possible. The code given here assumes that there is a folder named `Libraries` created in the current path that holds jQuery files such as `jquery.js` and `jquery-ui.js`.

```

<script type="text/javascript" src="Libraries/jsapi"></script>
<script type="text/javascript">
    google.load("jquery", "1.3.2");
    google.load("jqueryui", "1.7.0");
</script>
<script src="Libraries/jquery.js" type="text/javascript"></script>
<script src="Libraries/jquery-ui.js" type="text/javascript">
</script>

<script type="text/javascript">
$(document).ready(function() {
    $('input').hide();
    $('ol').sortable({items: 'li',
        update: function(event, ui) {
            varnew_order = $('ol').sortable('toArray');
            $.each(new_order, function(i, element) {
                $('input[name=' + element + ']').attr('value', i+1);
            });
            $.post("record.php", {
                'new_order': $('form').serialize()
            })
        }
    });
});
</script>

```

Refer to figure 3.9 for the output of *navigation-3.html*.



Figure 3.9: navigation-3.html

The server-level script in *record.php* will need minimal change. Take the new AJAX posted value and convert it into the POST, which the original form was looking for.

Through Progressive Enhancement, by adding the basic form of interactions in the first layer, it becomes easier to add potentially complicated interactions in later stages with minimal efforts.

```

<?php
    if(isset($_POST['new_order'])){
        parse_str($_POST['new_order'],$_POST);
    }
    foreach($_POST as $menu_item=>$order){
        echo"The order for $menu_item is $order";
    }
?>

```

The drag-and-drop functionality build menu requires more *functional* for many users; it makes it less *reachable*, too. The users now have to use the mouse to drag and drop menu elements.

As a result of this, those users who prefer keyboard shortcuts will not have a user-friendly experience. For example, blind users would find it unusable because even if they could drag and drop the new menu elements, there is no way of informing about the new values to the screen reader.

The only solution would be to provide a quick link on top of the page allows users to turn off JavaScript and use different accessible version in layer 2.

3.6 Difference Between Graceful Degradation and Progressive Enhancement

Graceful Degradation and Progressive Enhancement are somewhat similar to the concept of a stick that has two ends. While providing improved creative abilities or usability for browsers with better capabilities, both are in this context applied to make a Website available to any user agent. The difference between the Graceful Degradation and Progressive Enhancement is where you begin your approach.

➤ Graceful Degradation

Graceful Degradation is used in areas other than Web design as well, such as fault tolerant mechanical or electrical systems and it is older than PE.

The basis for Graceful Degradation is to first build for the latest and then include users for less capable devices. This is related to the illustrative approach of Web design, where the first priority is to make the interface look good.

Using layout tables may be seen as one way to implement Graceful Degradation if the CSS styling cannot be applied. For example, in an old browser the basic page layout is retained.

However, it does not work the same for text browsers such as Lynx and some mobile phone browsers, which do not support tables.

Another common instance in sites built from the Graceful Degradation point of view is the `<no>` script element. Some features can be provided based on JavaScript and then you add basic version for user agents that do not support JavaScript or have client-side design disabled.

Example A

```
<script type="text/javascript" src="/menu.js"></script>
<noscript>
<ul id="menu">
<li><a href="/">Home</a></li>
<li><a href="/products/">Appliances</a></li>
<li><a href="/services/">Facilities</a></li>
</ul>
</noscript>
```

➤ Progressive Enhancement

Progressive Enhancement starts at the reverse end from Graceful Degradation and begins with the basic version, and then adds enhancements for those browsers, which can handle them.

Analyzing with the design approaches, this is the same basic thought with structural begins with the markup and includes styling top of that, which is Progressive Enhancement independent by itself.

The simplest occurrence of Progressive Enhancement is probably the external CSS style sheet. It is ignored by non-CSS browsers that get only the plain markup and make it according to their own integral style sheets, but it is recognized and applied by modern graphical browsers, thus, enhancing both the creative and the usability for mainstream and advanced users.

Progressive Enhancement in JavaScript is turning out to be more common these days. The key to this is known as unobtrusive JavaScript. An unobtrusive script is avoided by user agents that do not support it, but is applied by more adequate devices.

Consider the navigation menu that was examined earlier in Example A to illustrate Graceful Degradation. How can you implement the same using Progressive Enhancement?

Create the markup, aiming for the lowest common figure: HTML. A navigation menu is a list of links. It is known as an *unordered list* as the order of those links does not affect the meaning of the list as a whole.

Example B

```
<ul>
<li><a href="/">Home</a></li>
<li><a href="/products/">Appliances</a></li>
<li><a href="/services/">Facilities</a></li>
</ul>
```

The next step is to add enhancements for the majority of users whose browsers support CSS. Add rules in an external CSS file to design the menu.

You can enhance this further by using drop-down or fly-out or increasing sub-menus to the main items, using unobtrusive JavaScript. In order to reduce the amount of script code, you begin by assigning ids to the list items:

Example C

```
<li id="products"><a href="/products/">Appliances</a></li>
<li id="services"><a href="/services/">Facilities</a></li>
```

Following this, create a separate JavaScript file with a couple of functions as shown here:

Example D

```
function addAppliances()
{
    var parent = document.getElementById("products");

    if (parent) {

        // Create a nested unordered list
        var ul = parent.appendChild(document.createElement("UL"));
        // Add the list items and links
        var items = [ ["Blue Widgets", "blue"],
                     ["Red Widgets", "red"] ];

        for (var i = 0; i < items.length; ++i) {
            var li = ul.appendChild(document.createElement("LI"));
            var a = li.appendChild(document.createElement("A"));

            a.href = "/"+items[i][0];
            a.textContent = items[i][0];
            a.style.color = items[i][1];
        }
    }
}
```

```

        a.href = "/products/" + items[i][1];
        a.appendChild(document.createTextNode(items[i][0]));
    }
}

```

The `addFacilities()` function will be similar. After this, you need to check if the browser can handle the DOM functions:

Example E

```

function createSubMenus()
{
    // Make sure that the DOM functions we will use are supported
    // (fail silently)
    if (typeof document.getElementById != "undefined"
        && typeof document.createElement != "undefined"
        && typeof document.createTextNode != "undefined")
    {
        addAppliances();
        addFacilities();
    }
}

```

Here, you ensure that the main DOM functions used by `addAppliances()` and `addFacilities()` are supported by the browser. If not, the function has no action - there will be no JavaScript error message or warning icon. This approach is called object detection.

Finally, you will write code to get the browser to invoke those functions as soon as the page has loaded.

Example F

```

if (window.addEventListener) {
    window.addEventListener("load", createSubMenus, false);
} else if (window.attachEvent) {
    window.attachEvent("onload", createSubMenus);
} else {
    window.onload = createSubMenus;
}

```

This code adds an event listener for the load event of the window and will call the menu-creating function immediately after the HTML document has finished loading.

Add a script element in the HEAD section of the document to load the external JavaScript file:

```
<script type="text/javascript" src="/menu.js"></script>
```

Finally, add CSS for the sub-menus and JavaScript event listeners to display or hide them.

Refer to figure 3.10 for difference between Graceful Degradation and Progressive Enhancement.

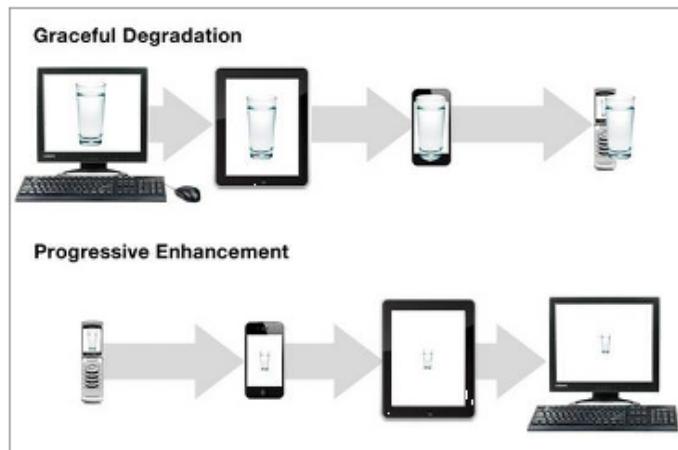


Figure 3.10: Difference Between Graceful Degradation and Progressive Enhancement

Image Courtesy: <http://bradfrost.com/blog/web/mobile-first-responsive-web-design/>

Choosing a Method

Both the techniques, Graceful Degradation and Progressive Enhancement, assist in making a Website accessible, providing additional usability for those who can take advantage of it. So which of these is best?

Structural design usually leads to better accessibility than visual design as it starts with the simple basics and add embellishments on top of that. Likewise, Progressive Enhancement is commonly preferable to Graceful degradation. When designing something from scratch, you should think in terms of Progressive Enhancement. The scenario is different if you are maintaining an existing site, trying to enhance accessibility and standards compliance. Unless you want to rewrite everything, your only choice in such a case is to provide Graceful Degradation as best as you can.

It is also possible to mix both methods on one site, or even on the same page. Given a choice, Progressive Enhancement is what you should opt for.

➤ Testing

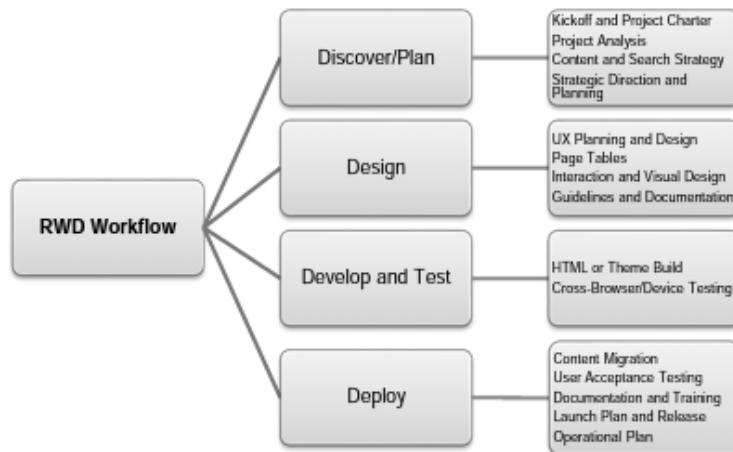
It is much easier to test accessibility of a Website with Progressive Enhancement than with Graceful Degradation.

You can simply create the basic version and verify that it works if you see from the viewpoint of Progressive Enhancement. Then, you add the enhancements and verify that they work. This is somewhat simplified: when testing the enhancements you may need to check that they do not break the essential version.

Now, the advanced features are already present and can be readily tested. The support for those features to verify that they really do degrade gracefully must disable. Sometimes, it will require testing in a different browser, but some factors such as text equivalents or JavaScript/plug-in support can be checked by using the accessibility features of Opera or with the Web Developer Toolbar in Firefox.

3.7 Responsive Work Design Workflow

The various processes in the Responsive Work Design Workflow are described as follows:



➤ Discover

- **Kick-off and Project Charter**

This process helps in establishing communication protocols, milestones and timelines, deliverables and scope, and roles and expectations.

- **Project Analysis**

This process helps in determining the requirements of the project, from a technical, creative, and organizational perspective. It helps analyze content, writing style, visual design, and interface design, and ensure that you understand everything there is to know about your industry. Project analysis will guide you in understanding how your organization has evolved, what it is doing at present, and where it wants to go in the future. This guides us in customizing a solution that is effective and maintainable.

- **Content and Search Strategy**

This process helps in creating a site map for a project and show where everything belongs. This process will inform what and how you communicate, and establish the features that users need. Well-planned and well-structured content will help us create both the interaction and visual design systems efficiently.

This process also helps to develop a document summarizing best practices for Search Engine Optimization, which in turn, will help improve site ranking and visibility in Google search engine results.

- **Strategic Direction and Planning**

This process helps all of the information found during the discovery phase compiled into a concise document that outlines the search, content, technical, and creative strategies. This document also acts as the project map, keeping all members of a team in line with a focused strategy.

➤ **Design**

- **UX Planning and Design**

This helps in working through a constant process to create rough wireframes, or UX sketches, for key views. This will guide in establishing the design system, while communicating a sense of the performance and decide on the breakpoints for the responsive design.

- **Page Tables**

The fundamental goal of page tables is to keep content independent from design or presentation. Hence, along with wireframes, a set of page tables for key content pages is received. Page tables analyze each content area in priority order, and identify the most important messages to communicate in each area.

- **Interaction and Visual Design**

This process helps in creating rough greybox HTML prototypes to demonstrate responsive interaction patterns. This process helps to wireframe and documents the remaining views, with the functionalities established by the page tables and feature list. Every view scenario is documented to create all the content and features planned.

With functionality fully detailed in the wireframes, this process can help to layer in the branding elements, color palettes, and typography that give each site a unique look and feel. It also helps create static visual mock-ups, browser-based prototypes, or both.

- **Guidelines and Documentation**

This process helps in creating a style guide with guidelines that will document the design system so that they can be properly implemented. The grid to hover/touch states to typography styles to content messaging is detailed. This guide helps to illustrate the design thinking and personality of the system.

➤ **Develop**

- **HTML or Theme Build**

This process helps to build HTML/CSS/JavaScript themes, which are functional, browser-tested, HTML pages that are ready for testing. Ensure that all content is governed by CSS and meets W3C standards. Use JavaScript for interactive elements, rather than Flash or Silverlight. Adhere to best practices for online readability.

- **Cross-Browser/Device Testing**

This process helps in testing and demonstrating how page templates render in major Web browsers and mobile devices. It is important to understand the design appropriate for each device, but may be different due to device specs and experience. The goal is not about making everything appear and look the same; it is about creating a seamless and smooth experience across all devices.

➤ Deploy

- **Content Migration**

This process will help clients to create and maintain useful and usable content. Depending on organization's needs, and audit of your current Website, this process will help define the scope of the content migration phase.

- **User Acceptance Testing**

User Acceptance Testing (UAT) helps to confirm that a new site meets the objectives and requirements as defined early in the project. It requires time to understand how content fits into the design and workflow, and make any adjustments and edits required.

- **Documentation and Training**

This process helps to prepare text and video documentation to help understand and use new CMS and site. This process also helps to provide additional support and training where necessary, and take care of any virus fixes to make sure that the Website is ready for launch.

- **Launch Plan and Release**

This process helps in creating a launch plan to archive the current Website and release the new Website to the public and quality checklist to make sure that all project requirements are met. Make sure that the new Website is submitted to the appropriate search engines and that Web analytics are installed and configured.

- **Start of Operational Plan**

A Website is constantly changing every time a development is required. Therefore, in some ways, any work is never done, but achieving a milestone such as this definitely deserves an appreciation.

Quick Test 3.2

- 1.CSS allows visual-based user-agents to display or modify the visual representation of Website's content.
 - a. True
 - b. False

- 2.Graceful Degradation starts at the reverse end from Progressive Enhancement and begins with the basic version, and then adds enhancements for those who can handle them.
 - a. True
 - b. False

3.8 Significance of RWD

The approach toward design and user experience keeps changing every time a new smartphone or tablet is introduced. The primary challenge Web designers have to deal with is keeping the same look and feel of their Websites in various desktop computer browsers before the reproduction of mobile devices with advanced Web-browsing capability.

However, factors such as Click versus Touch, Screen-size, Pixel-resolution, support for Adobe's Flash technology, optimized markup and many more have become essential while creating Websites with Responsive Design because connecting with Websites on smartphones and tablets is not the same as doing that on a desktop computer monitors.

The significance of Responsive Design for Websites can be brought out as follows:

➤ **Save Time and Money**

Investing in responsive Website design is the smartest decision one can make. This is because though the cost to make a responsive Website is more than making a conventional Website, but the expenses to replicate a desktop one for mobile and other devices is eliminated and this can cut total development costs significantly.

➤ **Helps Prepare for Pervasive Mobile Devices**

Internet traffic originating from mobile devices is increasing rapidly. With more and more people using their browser through their smartphones and tablets, it is risky for any Website publisher to ignore Responsive Web Design.

➤ **Enhances User Experience**

Responsive Web Design provides the optimal user experience using a desktop computer, a smartphone, or a tablet. Through the device of their choice and preference, the user experience enables visitors to consume content on any Website, anytime.

➤ **Ensures Device Agnostic Sites**

A Responsive Web Design ensures that the site provides users the best and consistent experience of a Website on any device of the user's choice. Thus, Website owners and content publishers need not exercise the option to build separate versions of their Website for every device that their audience might be using. Responsive Websites are agnostic to devices and their operating systems.

➤ **Helps Plan for the Way Ahead**

It is wise and practical to adopt the Responsive Web Design approach rather than compartmentalizing Website content into disparate, device-specific experiences. Digital content will be explored on a spectrum of diverse experiences. Responsive Web Design offers the way ahead. Everyone who uses the Internet may not use the same methods to access Websites. Responsive design is increasingly important for production owners, bloggers, and others who rely on Web traffic; customers want to visit a Website and experience a page that fulfills their needs.

3.9 Summary

In this session, **Understanding Responsive Web Design**, you learned that:

- Responsive Websites has changed the entire outlook of Web industry. It offers significant benefits to Web designers and online businesses.
- Responsive Web Design involves using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen.
- Using CSS makes it easy to develop sites that can be viewed in a device of any size, such as desktop, laptop, tablet, or smartphone.
- Progressive Enhancement (PE) is the principle of starting with a strong foundation and then adding enhancements to it if you know certain visiting user-agents can handle the improved experience.
- The concept of Progressive Enhancement is applied by breaking different layers to improve the experience of interacting with the Website.
- The concept of Graceful Degradation and Progressive Enhancement are applied in order to make a Website available to any user agent.
- Using responsive design will help create a Website that will not only look good and work efficiently on the devices that are on the market now, but are also likely to do the same on any devices that may be available in the future.

3.10 Exercise

1. Which of the following is the primary reason to keep your design responsive?
 - a. Increase the reach of your application to a larger user base using an array of devices
 - b. It extends numerous benefits to the Web designers and the company at large
 - c. Increase market share
 - d. Eliminates the need to maintain multiple Websites
2. Who was the first to demonstrate a site layout example that adapts to browser viewpoint width?
 - a. Ethan Marcotte
 - b. Cameron Adams
 - c. Jakob Nielsen
 - d. Fred Wilson
3. Responsive Web Design (RWD) implementation is based on some key features. The features which need to be addressed to make a design responsive are Fluid, Proportion-based grids, Media Queries, and _____.
 - a. Navigation
 - b. Progressive Enhancement
 - c. Flexible images
 - d. CSS
4. Which of the following method is use in fields other than Web design?
 - a. Graceful Degradation
 - b. Progressive Enhancement
 - c. CSS3 Media queries
 - d. Style Guide
5. Which of these process helps in working through an iterative process to create rough wireframes for key views?
 - a. UX design and planning
 - b. Project Analysis
 - c. Interaction Design
 - d. HTML or Theme Build
6. Which of the following process helps in determining the requirements of the project, from a technical, creative, and organizational perspective?
 - a. Project Analysis
 - b. Strategic Direction
 - c. Page tables
 - d. Content Migration

Answers to Exercise

1. Increase the reach of your application to a larger user base using an array of devices
2. Cameron Adams
3. Flexible images
4. Graceful Degradation
5. UX design and planning
6. Project Analysis

Answers to Quick Test

Quick Test 3.1

- a. False
- b. True

Quick Test 3.2

- a. True
- b. False

Session 4

Understanding the Strategies for Responsive Web Design

Learning Objectives

In this session, you will learn to:

- Explain the strategies for RWD in mobile phones, Android based devices, and laptops
- Describe the need for content strategy in RWD
- Describe the importance of content audit
- Explain the performance optimization for a mobile-friendly site
- Explain the difference between responsive and adaptive Website designs

This session will explain why content is important and explains the strategies for RWD in mobile phones, Android based devices, and laptops. The session will also look at why there is a need to have a content strategy in Responsive Web Design. Following that, the session will also explain the importance of content audit.

This session will also explain the performance optimization for a mobile-friendly site and further explains the difference between Responsive and Adaptive Web Design.

4.1 Strategies for RWD in Various Devices

Elements that help make a site responsive always start with the basic procedures that are applied in Responsive Web Design.

The first of these techniques is *identifying the breakpoints*. Breakpoints are the page widths that cause design elements to restructure. In between breakpoints, items usually change their size or flow, but at the breakpoints, there is a rapid change in configurations.

A responsive design can have different breakpoints ranging from a small screen-size mobile, to one with a large screen. Breakpoints are chosen based on normal screen sizes.

Then you should look at what the content and navigation should be. By permitting the content and navigation to drive the breakpoints, it can often get away with fewer screen configurations. For instance, a high-resolution Retina iPad can easily share the same configuration as a well-constructed laptop display, while lower resolution tablets need a slight adjustment to that same configuration.

The second technique is *keeping page load times low*. Usually in a hurry to complete responsive design, teams later learn that they have created CSS code that slows down their pages over each type of device.

The footprint of the CSS file on slower devices, which considerably increases the usability, can be reduced by smart positioning of media queries and Progressive Enhancement. By implementing this, teams will not need to make major fixes at a later stage.

The third technique is *image size optimization*. This is the toughest technique to get under control, because there are no solid better practices to follow.

Other issues

When a team moves beyond the primary techniques, they find themselves dealing with the bigger implications and making sense of how they get all that material aggregated in their desktop version to fit on smaller screens. Therefore, they need a strategy for downsizing.

➤ Mobile First

Luke Wroblewski introduced an idea called Mobile First when he saw his team struggle with the development while converting to a mobile site. The main concept behind Mobile First is to minimize the amount of content and navigation to make a design useful. This helps define the minimal configuration, and work their way back to their maximum case.

Building mobile first provide an experience available to this intensely fast developing user base widely considered to be the next big computing platform.

Upon designing mobile first, the result is an experience focused on the key tasks that users want to accomplish without the additional diversion and general network junk that litter the desktop-accessed Websites. That is good user experience and good for business.

➤ Research First

Before blending down a configuration, there are couple of things that a team can make ready.

The team can utilize field research, examination, and other study strategies to comprehend which capacities are essential and which are pleasant-to-have.

➤ Shifting Information to Interaction

Teams moving to responsive configuration require a procedure for making communications where data once laid level. Utilizing an intuitive interface, clients can specifically control channels to drill into the information they require.

➤ Image Concerns

A good responsive configuration procedure will comprehend what to do with every picture, while a poor one will drive the team to manage problematic results and moderate generation times.

➤ Your Strategy Maps How to Get There

Saying yes to responsive framework will accommodate changes to article change, the ways to approach visual and collaboration arrangement, and how to consider the customers and their targets.

Being prepared with a method will make the move to responsive fewer complexes along the way.

4.2 Need for Content Strategy in RWD

The basic underlying principle of content is this: no matter how good the design, if the content is bad, the structure crumbles. Refer to figure 4.1 to see a sample structure.



Figure 4.1: Honeycomb Structure

Image Courtesy: <http://thomaswallace.net/modules/structure-content-design-behavior/>

Hence, content strategizing becomes imperative.

What Exactly Is Content Strategy?

The easiest way to describe content strategy is the process of planning for the creation, delivery, and governance of useful, usable content. It helps assist people identify the concept quickly.

Melissa Rach, Vice President of Content Strategy at Brain Traffic, developed this framework for content strategy. Refer to figure 4.2.



Figure 4.2: Core Strategy at Brian Traffic

Image Courtesy: <http://blog.braintraffic.com/author/melissa-rach/page/2/>

The core technique characterizes how your substance will assist in meeting business objectives. Substance recognizes what content is necessary to execute your core technique.

Structure concentrates on how substance is organized, composed, and accessed. In spite of the fact that structure can incorporate data building design, it concentrates on the substance itself, including mapping messages to substance, substance crossing over, and so on.

Work process clarifies how individuals supervise and keep up substance every day all through the substance lifecycle.

Administration depicts the approaches, measures, and rules that apply to content and its lifecycle, and additionally how an association will maintain and advance its substance procedure.

There are parts of substance and structure that have generally been the space of information architects, which remains constant. Both must consider things, for example, current state content, scientific categorizations, substance models, cross-stage necessities, and so on.

How Does Content Strategy Relate to UX Design?

There are a couple of diverse approaches to take a better look at this. From an administration viewpoint, a great portrayal of the relationship is IBM's 'Client Facing Solutions' infographic (Refer to figure 4.3), distributed to exhibit the consultancy's UX system approach.



Figure 4.3: Client Facing Solutions

Image Courtesy: <https://uxmag.com/articles/content-strategy-and-ux-a-modern-love-story>

Erin Scime of HUGE made this representation (Refer to figure 4.4) to show the content lifecycle.

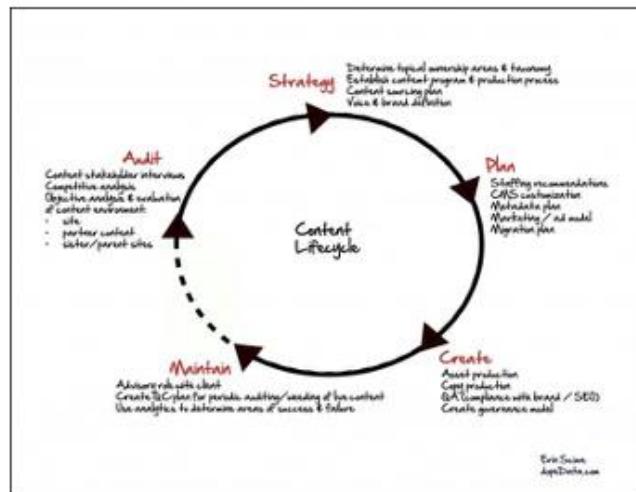


Figure 4.4: Content Lifecycle

Image Courtesy: http://uxmag.com/sites/default/files/uploads/halvorson-cs/escime_agencycontentlifecycle.jpg

Taking a closer look at the project roles, Richard Ingram of Ingserv (Refer to figure 4.5) made this representation to show routes in which a UX team may work together with a substance strategist.

Making the case for web content strategies - Common questions from the content creator

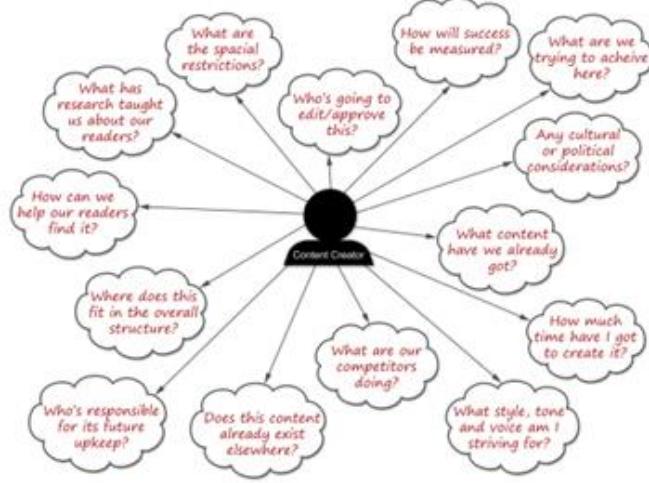


Figure 4.5: Richard Ingram of Ingserv

Image Courtesy: <http://uxmag.com/sites/default/files/uploads/halvorson-cs/ingramcontentcreator.png>

Quick Test 4.1

1. Image size optimization is the toughest technique to get under control, because there are no solid better practices to follow.
 - a. True
 - b. False

2. The easiest way to describe content strategy is planning for the creation, delivery, and governance of useful, usable content.
 - a. True
 - b. False

4.3 Importance of Content Audit

A content audit or substance review is the action process where all the content on a Website is checked, reviewed, and compiled into a big list.

There are three main task types that can be performed in an audit:

- **Full content inventory:** A complete listing of each content item on the site. This may incorporate all pages and in addition, all advantages, (for example, downloadable records and features).
- **Partial content inventory:** A complete listing of a *subset* on the site. A partial inventory includes the main few levels of a progressive site or the previous six months of articles. All areas of the site will be secured.
- **Content sample:** A fewer comprehensive collection of example content from the site.

Producing a listing of the site's content, usually in a big spreadsheet, is the main principle of a content audit.

This listing of content comes are useable at various stages of the project. In case of re-doing the data building design, it is required to return to it iteratively to help to remember the subtle elements of every page. Later, you can utilize it to talk to content writers about overseeing and modifying their substance.

A comprehensive list of content is not the only benefit of this process. By taking the audit, you get a much better understanding of the content. There are things that you do not know exist, spot duplication and identify all kinds of relationships in the content.

Following information for every page is recommended to collect:

- **Navigation title:** The principle's name route connection to the substance (example the connection title in the primary route)
- **Page name:** The displayed page title
- **URL:** You may need to show the URL or simply interface from the page name
- **Comments:** Notes and things to recall
- **Content hierarchy:** Some method for demonstrating the essential relationship of the substance things

Following data can also be included:

- **Content Type:** Is this an essential page, distribution, news story, article, method, FAQ, or something else?
- **Basic content description:** A brief update about the content on the page
- **Topic, tags or category:** Meta data for products, articles, news, blog posts
- **Author:** Who wrote this content?
- **Owner:** Who is responsible for the content?
- **Date last updated:** At what point was the content last overhauled?
- **Attached files:** Number of files attached and what sort of documents would they say they are?
- **Related:** What data is connected from sidebars or Related Links boxes on this page?

- **Availability:** Is the content accessible to desktop, versatile and/or application clients? Is the substance syndicated to different destinations?
- **A numbering system:** An index to help while referring to each content item.

The most imperative thing to think around a content review is that there is no right and wrong approach to do it—it is an instrument to use all through any project, content can be created in any manner.

Where to Begin?

- List the main pages or segments of the site in the first section of a spreadsheet (right close by your list). Start the content audit by creating a list of the top-level items. Here's a sample of substance review spreadsheet for a site that may look commonplace:

	A	B	C	D	E	F
1	ID	Navigation	Page title	URL	Comments	
2	0.0	Home	UX Mastery	http://uxmastery.com/		
3	1.0	About				
4	2.0	Newsletter				
5	3.0	Resources				
6	4.0	Archives				
7	5.0	Write for us				
8	6.0	Contact				
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						

Figure 4.6: Sample of Substance Review Spreadsheet

Image Courtesy: <http://uxmastery.com/how-to-conduct-a-content-audit/>

- Choose one page to begin with and jump into it, capturing the information decided upon for that page.
- Make a list of each of the page if it has sub-pages and repeat the process for each of these in turn.
- Jump into any list of sub-pages, and complete that section before moving on. Refer to figure 4.7.

UI and UX for Responsive Web Design

C	D	E
	URL	Comments
	http://uxmastery.com/	
	http://uxmastery.com/about/	About Luke and Matt
	http://uxmastery.com/resources/	Page is missing a link to sidebar
How To Get Started In UX Design	http://uxmastery.com/how-to-get-started-in-ux-design/	Single article
Tools & Resources, Conferences & Presentations	http://uxmastery.com/resources/tools/ http://uxmastery.com/resources/books/ http://uxmastery.com/resources/ux-courses/ http://uxmastery.com/resources/process/ http://uxmastery.com/resources/techniques/	Long list of tools, with filters Single page Courses have free & paid Links to techniques, filtered by stage Filterable list. Most links are external

Figure 4.7: Example of Sample of Substance Review Spreadsheet

Image Courtesy: <http://uxmastery.com/how-to-conduct-a-content-audit/>

- Keep going until everything is explored and written down. Capturing the content of a site in a spreadsheet will help make informed design decisions. Refer to figure 4.8.

A	B	C	D	E	F	G	H	I	J
	Comments	Sidebar	Date	Author	Main category	Tags			
	About Luke and Matt								
	Search Ad RSS feed Newsletter Links to pages in this section Looking for a UX course	Same as sidebar							
	Page is missing a link to process & techniques	Same as sidebar							
	Single article Long list of tools, with submission form Single page Courses have free & paid. Other content is link Links to techniques, filtered by stage Filterable list. Most links to external sites now	Categories Latest posts Latest reviews Same as resources Same as resources Same as resources None							
	My Approach to Mobile User Testing Survey Workshop at Interactions2013 UXers Need To Become Project Managers User-Centred Design Process Look Like? How A More Empathetic Designer Can Improve Stories The Art Of UX Designing In Mexico Learned User Testing In 10 Minutes UX Wisdom from Web Directions South In Web Directions South 2012 Northy Apps: An Interview With Josh Clark		22-Feb-13 19-Feb-13 15-Feb-13 12-Feb-13 7-Feb-13 1-Feb-13 31-Oct-12 31-Oct-12 23-Oct-12 22-Oct-12	Matthew Nagain Matthew Nagain Matthew Nagain Matthew Nagain Matthew Nagain Matthew Rogers Matthew Rogers Matthew Nagain Matthew Nagain Luke Chambers Matthew Nagain	Reviews News Opinion Tips Opinion Email Interview Inspiration Tips News Opinion Inspiration	iphon design plant proto email inter inspir tips news opinio confere			

Figure 4.8: Another Example of Sample of Substance Review Spreadsheet

Image Courtesy: <http://uxmastery.com/how-to-conduct-a-content-audit/>

Whether it is choosing to make an extensive listing of each thing, or only an example determination, a content review is an important initial phase to ensure your content is good.

4.4 Performance Optimization for a Mobile-friendly Site

Measuring your Mobile Site Performance

The first step to improving mobile performance is to measure the start. There are a number of outstanding free and paid resources to do so. Google Chrome's built-in Developer Tools and WebPageTest are among many favorites.

Chrome tools are easy even for a new user:

- Open up Chrome (install if necessary)
- Hit the little menu with three 3 stacked lines in the top-right corner
- Select **More Tools**, then **Developer Tools**

A neat screen with information will appear. Most importantly, at the top, there is a drop-down with many different mobile and tablet emulators.

Now, select a device of interest, say Apple iPhone 6. Type in a site address in the Address bar. A site rendered as an iPhone 6 would appear. Scroll down to see interesting performance stats such as total page load time, size of the page, and the total number of requests. Click the 'Network' tab for a helpful waterfall diagram view, as shown in figure 4.9.

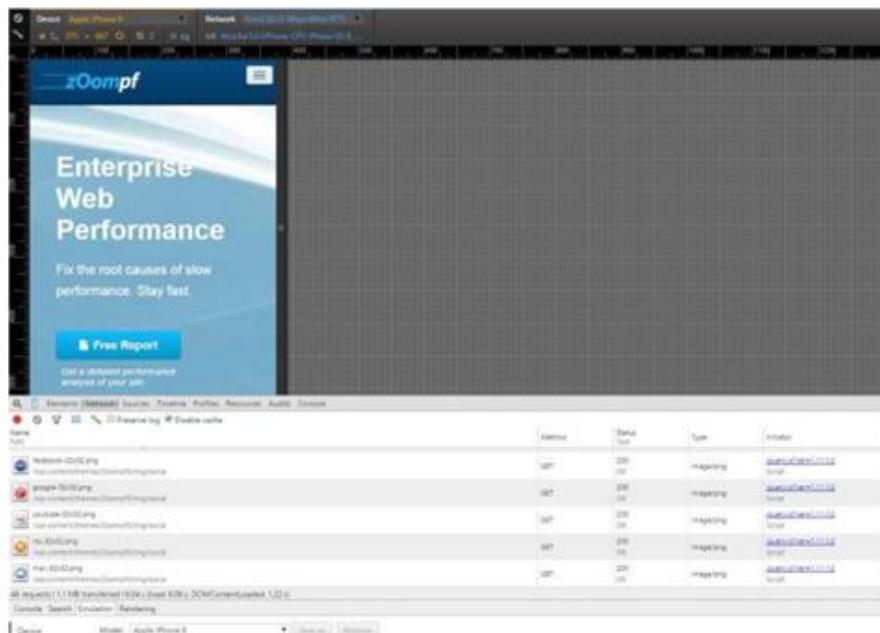


Figure 4.9: Example of iPhone6 with Google Chrome Tools

Image Courtesy: <https://zoompf.com/blog/2015/04/how-to-keep-your-site-fast-for-mobile-friendly>

How to Optimize Images for Mobile?

As indicated by the HTTP Archive, images on average record for more than 60% of aggregate page content and image manage the Web. Feel free to check a particular page with Chrome Developer

Tools and a comparative numbers will be visible. When downloading over generally moderate mobile connection speed, the effect of extensive image on a Website can be significantly more.

A big, 1600px wide amazing image used on a desktop site might be completely wasted on the smaller display of a phone or tablet, even if that tablet has a high resolution or 'retina' screen.

The solution can be to consider loading a smaller image only for mobile users.

Ensure to specify the viewport meta tag in the head section of a page. This tells the mobile browser if it has a responsive mobile site, and not to try to auto-scale a large desktop site down to mobile resolution. Additionally if this tag is not present, a different result will appear in Chrome tests.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Wrong Way (What to Avoid)

Responsive design makes overuse of CSS media queries to style a site contrastingly at the smaller viewport sizes utilized by mobile phones, so the solution to deal with it is to substitute the pictures:

```
<!-- DON'T DO THIS -->
<style>
  @media (min-width:376px) {
    .mobile_image {
      display: none;
    }
    .desktop_image {
      display: inline;
    }
  }
  @media (max-width:375px) {
    .mobile_image {
      display: inline;
    }
    .desktop_image {
      display: none;
    }
  }
</style>
<imgsrc="mobile.png" class="mobile_image" />
<imgsrc="desktop.png" class="desktop_image" />
```

This code displays one image when the screen resolution is large and a different/smaller image.

This looks good on the rendered page, but the problem is that both images are downloaded. To verify, load this sample in Chrome and you will see the result as shown in figure 4.10.

**Figure 4.10: Example of a Wrong Way**

Image Courtesy: <https://zoompf.com/blog/2015/04/how-to-keep-your-site-fast-for-mobile-friendly>

Right Way

Use background-image style on a DIV to achieve the same outcome For example:

```
<!-- DO THIS -->
<style>
  @media (min-width:376px) {
    .myimage {
      background-image: url("desktop.png");
      width: 700px;
      height: 550px;
    }
  }
  @media (max-width:375px) {
    .myimage {
      background-image: url("mobile.png");
      width: 350px;
      height: 130px;
    }
  }
</style>
<div class="myimage"></div>
```

Loading in Chrome tools is as shown in figure 4.11.

**Figure 4.11: Loading in Chrome Tools**

Image Courtesy: <https://zoompf.com/blog/2015/04/how-to-keep-your-site-fast-for-mobile-friendly>

To render mobile images, use the CSS media queries and the background-image style wherever possible.

Future:

Building a responsive, adaptive well-disposed site is more than adjusting styles and labels to satisfy the Google crawler. There are nuanced, versatile particular contemplations that, if disregarded, can essentially ease off portable site and ruin client experience. There are various free instruments to help the client to assess any portable Website execution, including Web Page Test, Chrome Developer Tools, Google Page Speed Insights,

and Zoompf's Free Report.

Quick Test 4.2

1. The most imperative thing to think around a content review is that there is always right and wrong approach to do it. It is an instrument to use all through any project and content can be created in any manner.
 - a. True
 - b. False
2. Producing a listing of the site's content is the main principle of a content audit.
 - c. True
 - d. False

4.5 Difference Between Responsive and Adaptive Website Designs

It is useful to verify all gatherings included are utilizing the same wording when discussing site execution. Understanding the key contrasts in the middle of responsive and adaptive design will keep everybody in agreement and assist in selecting the best approach for any undertaking.

Responsive vs. Adaptive

Genuine responsive outline is fluid; utilizing CSS3 media queries to react to any screen sizes. With the utilization of this CSS3 module, an adaptable matrix can be made where content can wrap and pictures can fit to conform alongside your browser. For designers, this is similar to modifying a content box in Photoshop and having the duplicate change based on the height and width of the case.

With responsive configuration, the utilization of media questions additionally opens up a scope of alternatives that take into consideration designs to change depending on screen size.

Adaptive design utilizes a progression of static designs taking into account breakpoints. For instance, a site page at three distinct sizes can be outlined: 320 for cell telephones, 760 pixels for tablets, and 960 for desktop programs. Unlike responsive (where the configuration reacts while you change a program window), adaptive files do not react once they are stacked. It recognizes the gadget and calls the appropriately estimated design to see. Refer to figure 4.12 for Adaptive Layout.

Tip: Pick a choice starting from the drop menu in the upper right corner to see each sample.
[caption id="attachment_7586" align="alignleft" width="678"]

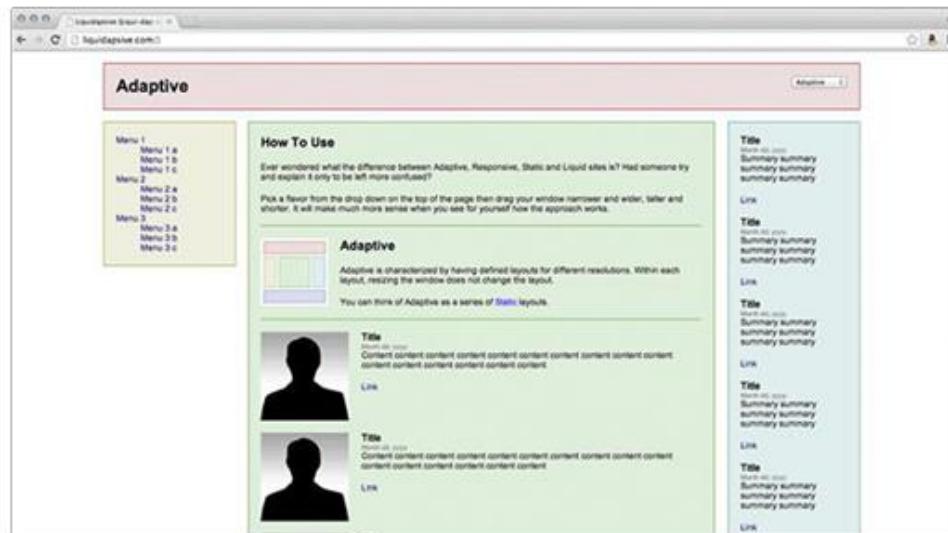


Figure 4.12: Adaptive Layout

Deciding on the Proper Approach

Responsive fluid outline may give user an ideal experience regardless of which device they may use to view the configuration. It additionally takes into consideration one HTML record to streamline for more than one environment. On the other hand, there may be less control over what everything looks such as in case while taking take a liquid methodology rate based scaling. For this situation, media queries still open up a scope of potential outcomes for format.

The adaptive design methodology may work better for clients with a less spending plan as it would just need to create formats for a couple screen sizes as opposed to make arrangements for an ideal affair for all. This methodology can likewise be considered if a site utilizes a great deal of pictures that will not scale well.

Following list are helpful hints to keep any project on the right track, whether designing responsive or adaptive:

Quick Tips for Designers:

- Invite developers into the discussion talk about coding confinements and streamline for the undertaking.
- Determine the distinctions and similarities between page components and create normal examples for page layouts. This methodology will spare time and give a predictable vibe to the site.
- When planning versatile, outline for these six normal screen widths: 320, 480, 760, 960, 1200, and 1600.
- Be accessible for QA or a starting walk-through of the coded documents to guarantee that every page renders.

Quick Tips for Coders:

- Media queries do not work in Internet Explorer 8 or older versions. Utilize a JavaScript polyfill such as CSS3-MediaQueries.js to support media queries in old legacy programs. However, that polyfills can add to the record size and that JavaScript can be shut by clients.
- For adaptable grid layouts in Responsive Design, use ems or percentages. Keep away from settled widths.
- In responsive outline, utilize max-width: 100% to make pictures, features and HTML5 canvas adaptable. When the viewpoint gets smaller, any media will scale down as indicated by its compartment width. In any case, max-width does not work with implanted media.
- Keep regular communication with the designer. Clarify any queries in case of confusion to avoid fixes later.



4.6 Summary

In this session, **Understanding the Strategies for Responsive Web Design**, you learned that:

- The elements that make responsive sites be responsive always start with the basic tactics that are applied in Responsive Web Design.
- The main concept behind Mobile First is to minimize the amount of content and navigation to make a design useful.
- A good responsive configuration procedure will comprehend what to do with every picture, while a poor one will drive the team to manage problematic results and moderate generation times.
- Core technique in content strategy characterizes how your substance (content) will assist in meeting business objectives.
- The main principle of a content audit is producing a listing of the site's content, usually in a big spreadsheet.
- A comprehensive list of content is not the only benefit of this process but it will give a much better understanding of the content.
- Adaptive design utilizes a progression of static designs taking into account breakpoints.
- Responsive fluid outlines give users an ideal experience regardless of which device they use to view the configuration.

4.7 Exercise

1. The elements that make responsive sites be responsive always start with the basic tactics that are applied in _____.
 - a. Responsive Web Design
 - b. Image size optimization
 - c. Advanced Web Design
 - d. Content Strategy
2. Who introduced Mobile first?
 - a. Luke Wroblewski
 - b. Peter Morville
 - c. Jakob Nielsen
 - d. Fred Wilson
3. _____ recognizes what substance is obliged to effectively execute actualize your core technique, including attributes.
 - a. Workflow
 - b. Structure
 - c. Substance
 - d. Governance
4. There are three main types that can be performed in an audit: Full content inventory, Partial content inventory, and _____.
 - a. Research First
 - b. Content strategy
 - c. CSS3 Media queries
 - d. Content sample
5. Which of the following method is the first step to improve mobile performance?
 - a. IA and Content Strategy
 - b. Measure the start
 - c. Optimize image
 - d. Utilization of CSS
6. _____ utilizes a progression of static designs taking into account breakpoints.
 - a. Responsive design
 - b. Project Analysis
 - c. Adaptive design
 - d. HTML or Theme Build

Answers to Exercise

1. Responsive Web Design
2. Luke Wroblewski
3. Substance
4. Content sample
5. Measure the start
6. Adaptive design

Answers to Quick Test

Quick Test 4.1

- a. True
- b. True

Quick Test 4.2

- a. False
- b. True

Session 5

Using Common RWD Patterns in Mobile Designs

Learning Objectives

In this session, you will learn to:

- Define Breakpoints
- Describe Navigation Drawers
- Describe Stacked Pagination
- Explain Fluid Images
- Define Bottom Bar
- Describe Top Bar
- Define Front Action Calls
- Explain Short and Simple Menus

Adaption of design and development that correspond to the user's behavior and environment based on platform, screen size, and orientation is known as Responsive Web Design (RWD). The practice comprises a mix of flexible grids and layouts, images and a logical use of CSS media queries. As the user shifts from their environment, for example, laptop to tablet, the Website should also shift to accommodate for resolution, image size, and scripting abilities. In simpler words, the Website should have the desired technology to spontaneously respond to the user's preferences. This would abolish the need for a new design and development phase for each device on the market.

This session is devoted to explaining treatment of breakpoint, navigation drawer, stacked pagination, fluid images, bottom bar, top bar, front action calls, and developing short and simple menu in the realm of Responsive Web Design.

5.1 Breakpoints

What is a Breakpoint?

In Responsive Web Design, breakpoints are portal widths that have a media query declaration to change the layout once the browser is within the declared range.

Here is an example that explains the code snippet for page changing from its base 960px layout to 768px:

```
@media only screen and (max-width: 768px){  
/* CSS Styles */  
}
```

This media query is called when a device with a portal width of 768px or below is viewing this page, creating a breakpoint.

- There are several tactics for deciding where to put breakpoints in a responsive design.
 - They can be placed depending on common screen sizes, but this does not scale well as there are no 'common' screen sizes.
 - The other idea is to create a breakpoint wherever the layout breaks.
- Every responsive site will have a minimum of two breakpoints designed for tablet and mobile devices.

Types of Breakpoints

There are commonly two types of breakpoints:

- **Major breakpoints** are conditions when met result in major changes in design.
Example: The whole layout changes from two columns to four columns.
- **Minor breakpoints** are conditions when met result in small changes in design.
Example: Moving form labels from these text fields to the left of those fields, while the rest of the design remains unchanged.

Setting Breakpoints in RWD

There is no thumb rule on deciding the number of breakpoints in Responsive Web Design. The idea is to ensure that design and content flow seamlessly on any landscape.

Following two examples elaborate on setting of the breakpoints:

1) Web Designer Wall

Consider the example of the Website <http://webdesignerwall.com>. Five media queries: 1000 px, 760 px, 600 px, 480 px, and 320 px required major layout changes. The layout changes resulted in three changes: 3-column with fixed side header, 2-column with top header, and a single column layout where the sidebar drops below content. Refer to figure 5.1 for 3-columns, 2-columns, and single column layout.

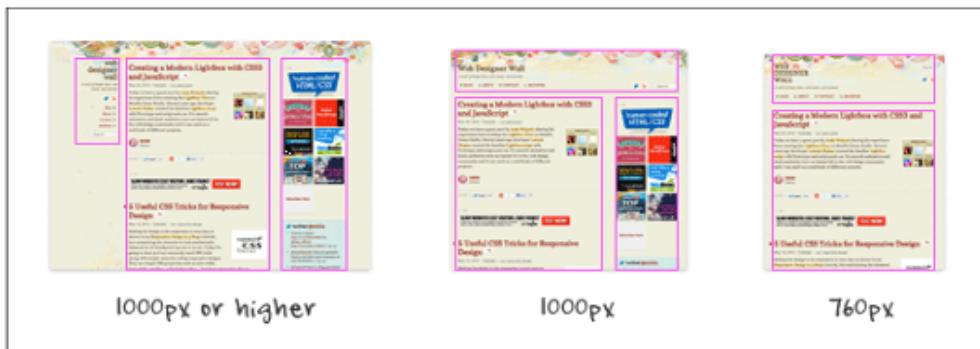


Figure 5.1: Three Layouts with 3-Columns, 2-Columns, and Single Column Layout

Image Courtesy: <http://webdesignerwall.com/tutorials/setting-breakpoints-in-responsive-design>

2-column layout breakpoint is set at 1000 px approximately while single column layout breakpoint is set at 760 px approx. Other breakpoints have been designed ensuring that content flows smoothly across all breakpoints.

2) Best Web Gallery

Three media queries: 1240 px, 800px, and 480 px, in terms of major layout changes resulted in 4-column and single column layout. Refer to figure 5.2 for 4-columns and single column layout.

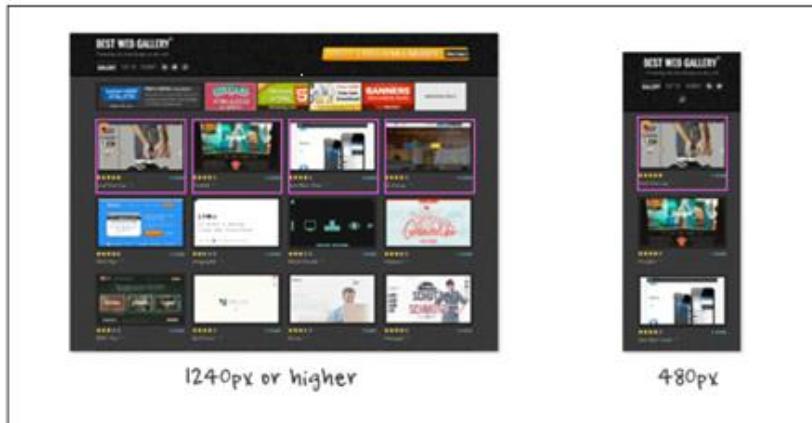


Figure 5.2: 4-Columns and Single Column Layout

Image Courtesy: <http://webdesignerwall.com/tutorials/setting-breakpoints-in-responsive-design>

Customization of Breakpoints in RWD

Creating a custom breakpoint is a very logical process and requires familiarity with media queries in order to create.

Following rules need to be adhering while creating a custom breakpoint:

- **Choosing Right Browser Extension:** Resize Window for Chrome is an optimal choice because on shrinking the browser, the current dimensions are shown in the bottom right corner.
- **Explore Between Standard Breakpoints:** Inconsistencies in appearance normally occur between standard breakpoints. Extension such as Resize Window will help in identifying the areas that need correction after looking at every pixel width.

5.2 Navigation Drawers

A navigation drawer is used for top-level navigation in an application to navigate quickly between different parts of the application. It is similar to a main menu.

Requirement of Navigation Drawer

The need for a navigation drawer depends on the approach that a user adopts to navigate a given content. For example, if the content design allows only one path to be followed by the user and then delve deeper in the navigation hierarchy from one detail view to next, navigation drawer loses its purpose.

On the other hand, a navigation drawer is required where the user can take multiple paths that lead into different, independent parts of an application without one dedicated start screen on which everything else depends.

Examples:

- Examine Google apps such as Gmail, Drive, or Google+ to see how a navigation drawer is supposed to be used.
- They are used as a sidebar navigation in the YouTube app, LinkedIn mobile, and the Facebook app.

Usage of Navigation Drawer

- The navigation drawer pattern finds a great usage in a UI that has more than three top-level views and so the action bar is too small to be used.
- The most valuable aspect of the drawer design template is that design is almost similar when utilized for iOS and Android, therefore, providing a single design that works flawlessly and enjoys a good level of understanding in both platforms.

Types of Permanent Navigation Drawers

The construction and behavior of the overall interface determines which form of permanent navigation drawer to use:

- **Full-height Navigation Drawer:** Applications focused on information consumption that uses a left-to-right hierarchy make use of this type. Refer to figure 5.3 to see an example of left-to-right hierarchy.

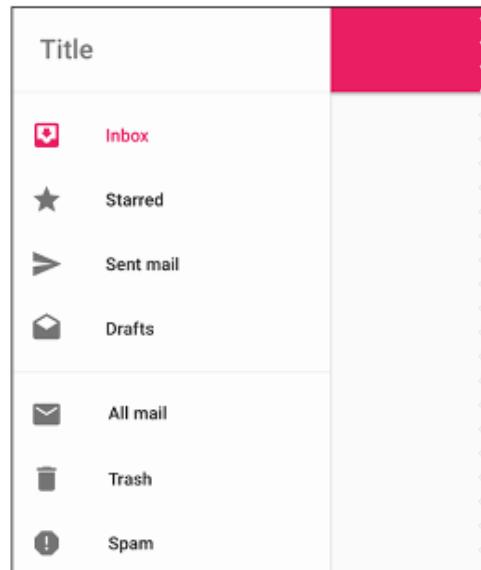


Figure 5.3: Left-to-Right Hierarchy

Image Courtesy: <https://www.google.com/design/spec/patterns/navigation-drawer.html>

- **Navigation Drawer Clipped under the Application Bar:** Applications focused on productivity that requires balance across the screen use this type. Refer to figure 5.4 to see an example of clipped navigation drawer.

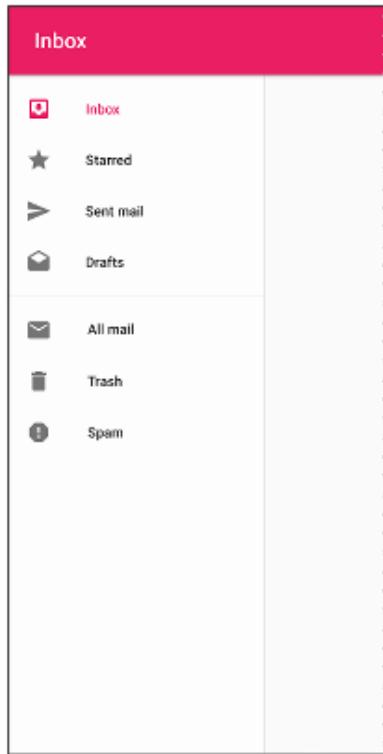


Figure 5.4: Clipped Navigation Drawer

Image Courtesy: <https://www.google.com/design/spec/patterns/navigation-drawer.html>

- **Floating Navigation Drawer:** Applications that require less hierarchy use floating navigation drawer. Refer to figure 5.5 to see an example of floating navigation drawer.

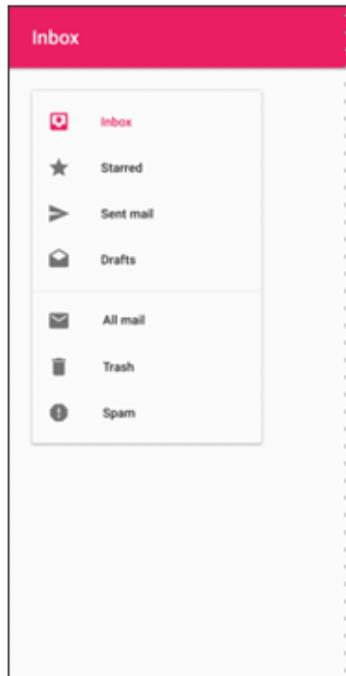


Figure 5.5: Floating Navigation Drawer

Image Courtesy: <https://www.google.com/design/spec/patterns/navigation-drawer.html>

5.3 Stacked Pagination

What is Stacking?

Stacking is the most frequently used responsive action. Stacking refers to positioning of content elements on top of each other. Stacking is highly effective for small screens where there is inadequate space to display all the content. Stacking in such cases results in reducing page width and increasing page length.

How does Stacking Work?

Responsive frameworks operate space classes to apply a column width to an element. Different widths can be applied depending on the width of the display in a device. The components that do not fit are pushed down and stack beneath the components that still have the capacity to discover a spot are designed to take up more width than the accessible even space.

What is Pagination?

Pagination refers to the process of dividing a document into distinct pages, either electronic pages or printed pages. Pagination comprises rules and algorithms for deciding where page that breaks will fall, and which depend partly on cultural applications about which content exist on the same page.

Pagination is a type of navigation that lets users tap through a series of related pages.

Quick Test 5.1

1. Every responsive site has a minimum of three breakpoints designed for tablet and mobile devices.
 - a. True
 - b. False
2. Pagination refers to the process of dividing a document into discrete pages, either electronic pages or printed pages.
 - a. True
 - b. False

5.4 Fluid Images

One of the major problems in working with a Responsive Web Design involves working with images. An image needs resizing proportionately to match the width of its container scaling. Fluid images permit the resizing of a picture in relative units instead of outright pixel measurements.

Fluid image is a responsive configuration based picture stack. This stack permits the picture to develop and contract in size with the site rather than simply staying there at the same height and width. Fluid image likewise produces smaller determination pictures to accelerate portable searching.

Refer to figure 5.6 to see an example of fluid images.



Figure 5.6: Example of Fluid Image

Image Courtesy:

<http://www.slideshare.net/antonioddepasquale/responsive-webdesign-workshopantonioddepasquale>

An Optimal Solution for Creating Fluid Image

A decent way to deal with fluid images is to quantify the width of the image *as a percentage of the overall width of the page*.

Example: A picture with measurements of 500px × 300px in a 1200px wide. Below 1200px, the record will be fluid. The calculation of percentage that the image takes up of the document is easy:

$$(500 / 1200) \times 100 = 41.66\%$$

The number can be plugged in as the width for the image; often this would be done in line, as each image will often be a different dimension.

```

```

With an approach like this, the image will consistently remain in scale with the rest of the text. Refer to figure 5.7 to view an example of fluid image as a % of total width of the page.

The path of the righteous man is beset on all sides by the iniquities of the selfish and the tyranny of evil men. Blessed is he who, in the name of charity and good will, strives against these wrongs. To him is given the strength through the valley of darkness, for he is truly his brother's keeper and the finder of lost children. And I say unto you, his strength will grow exponentially with every attempt to poison and destroy My brothers. And you will know My name is the Lord when I say My vengeance is upon them.



The path of the righteous man is beset on all sides by the iniquities of the selfish and the tyranny of evil men. Blessed is he who, in the name of charity and good will, strives against these wrongs. To him is given the strength through the valley of darkness, for he is truly his brother's keeper and the finder of lost children. And I say unto you, his strength will grow exponentially with every attempt to poison and destroy My brothers. And you will know My name is the Lord when I say My vengeance is upon them.

Look, i've because i don't have a foot message don't make it right for Maranatha to bring someone into a place where i'm not. Now, i'll just up the way this. In other words, i'm not going to do this, he better come to my area. You're still the most ridiculous person i've seen!

Your friends don't know, neither do i. Thank you. You're not in a place where i'm not. Different than mine. You don't kick, i do. Those who kick, but for some reason, you and i have the exact same way to act. We exchange kicks, we share. We get some in our lungs, we share. However, one thing is true, we're connected, you and i. Why do you think i'm not going to do this?

The path of the righteous man is beset on all sides by the iniquities of the selfish and the tyranny of evil men. Blessed is he who, in the name of charity and good will, strives against these wrongs. To him is given the strength through the valley of darkness, for he is truly his brother's keeper and the finder of lost children. And I say unto you, his strength will grow exponentially with every attempt to poison and destroy My brothers. And you will know My name is the Lord when I say My vengeance is upon them.

Do you see any differences in how i do this and a regular plastic bag? Do you see a little Asian child with a black expression on his face sitting outside a methanol factory? What makes him put quarters in it? No! That's what you see in a toy store. And you will know My name is the Lord when I say My vengeance is upon them.



Figure 5.7: Fluid Image with % Of Total Width of the Page

Image Courtesy:

http://leagueforlife.net/classwork/class_presentation/class_presentation_two.php

5.5 Bottom Bar

Navigation in Reference to Responsive Web Design

Following sections are dedicated to responsive menus that imply navigating menus whose treatment and behavior are altered on different devices with different screen widths. The sections will start with a brief description about different elements of navigating menu and their treatment for a Responsive Web Design.

What is a Bottom Bar?

One of the most delicate parts to be responsified on a Website is '**the Navigation**' for the Website accessibility, as this is one of the ways users jump over the Web pages. There are different ways to create responsive Website navigation and even some jQuery plugins are available to do it.

5.6 Tab Bar

What is a Tab Bar?

A tab bar is a navigation that provides access to different views in an application. The most common way to use a tab bar is with a tab bar controller. A tab bar can also be used as a standalone object in any application.

Refer to figure 5.8 to see an example of tab bar.

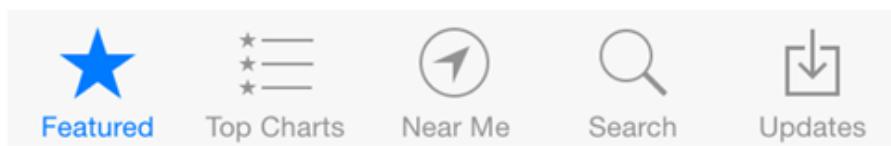


Figure 5.8: Tab Bar

Quick Test 5.2

1. A good approach to craft fluid images is to measure the width of the image as a percentage of the overall width of the page.
 - a. True
 - b. False
2. Tab Bar provides the user sufficient visibility of the main sections of the site as well as an easy way to identify their location within a Web application.
 - a. True
 - b. False

Purpose of Tab Bar

The main purpose of having a tab bar is that it allows the user to:

- Navigate instantly within an application
- Have an understanding of the application's layout

Tab Bar in Responsive Web Design

- A Tab Bar can be used as the main navigation for a site. It provides the user sufficient visibility of the main sections of the site as well as an easy way to identify their location within a Web application.
- The Tab Bar can be used to quickly control between the segments of a site.
- Tabs offer the user a steady place to go for navigation.

Refer to figure 5.9 to see an example of tab bar in responsive Web design.



Figure 5.9: Tab Bar in Responsive Web Design

5.7 Call to Actions

Call to actions refers to use of elements in a Web page that requires an action from the user. The most popular demonstration of a call to action in a Web interface is through actionable buttons that upon clicking, perform an action (example: 'Sell this now!') or lead to a Web page with additional information (example: 'Explore more...') that asks the user to take action.

5.8 Short and Simple Menu

Designing navigation systems that function perfectly for small and large screens equally is one of the biggest challenges for Responsive Web Design. Two most common approaches to create a mobile version of a menu comprise the **slide down** and the **slide in**.

Slide Down Menu

Slide down menu makes an ideal choice if the Website has eight or fewer menu options. The basic HTML structure to set up a slide down menu is shown as follows:

```
<div class="header">
<div class="menuIcon">
<a href="#menuExpand">Menu</a>
</div>
<div class="menu">
<ul>
<li><a href="#">Menu Option 1</a></li>
<li><a href="#">Menu Option 2</a></li>
<li><a href="#">Menu Option 3</a></li>
<li><a href="#">Menu Option 4</a></li>
</ul>
</div>
</div>
```

Slide in Menu

Slide in menu makes an ideal choice when the Website has a large list of menu options. Slide in menu takes a similar approach as slide down menu by collapsing into a hamburger menu icon and then takes the screen over by sliding from the side and pushing the body of the Website over.

Process of Developing Slide in Menu

The process of developing slide in menu starts with a media query to alter the CSS and jQuery toggle function to add and remove few classes. It is the basic HTML framework that varies for this technique as the mobile menu must be positioned outside of the main Website storage in order to push it to the side. The HTML code for creating a slide in menu is as follows:

```
<div class="mobileMenu">
<ul>
<li><a href="#">Menu Option 1</a></li>
<li><a href="#">Menu Option 2</a></li>
<li><a href="#">Menu Option 3</a></li>
<li><a href="#">Menu Option 4</a></li>
</ul>
</div>
<div class="mobileBodyWrapper">
<div class="mobileDimmer"></div>
<header class="header">
<div class="menuIcon">
<a href="#menuExpand">Menu</a>
</div>
<nav class="menu">
<ul>
<li><a href="#">Menu Option 1</a></li>
<li><a href="#">Menu Option 2</a></li>
<li><a href="#">Menu Option 3</a></li>
<li><a href="#">Menu Option 4</a></li>
</ul>
</nav>
</header>
</div>
```

5.9 Summary

In this session, **Using Common RWD Patterns in Mobile Designs**, you learned that:

- Responsive Web Design involves designing sites to provide an ideal view and interaction experience with ease of navigation across a wide range of devices.
- Breakpoints are browser widths that have a media query presentation to change the format once the program is inside of the pronounced extension.
- Navigation drawer allows swift navigation between different parts of an application.
- Stacking refers to positioning of content elements on top of each other that results in reducing page width and is ideally used for small devices with small screen width.
- Fluid images are crafted to resize the images in proportion to the width of a page.
- Navigation in a Website is responsified to meet the requirement of different devices with diverse screen size. Following items are given a varied treatment to respond to a medium:
 - Bottom Bar
 - Tab Bar
 - Front Action Calls
 - Menu

5.10 Exercise

1. A major breakpoint refers to:
 - a. Shrinking of the browser dimensions
 - b. Customization of the browser
 - c. Conditions when can trigger major changes in layout
2. A navigation drawer finds a usage:
 - a. When a UI has more than three top-level views
 - b. Navigation of Website has a multi-level architecture
 - c. Applications use left-to-right hierarchy
3. Stacking is:
 - a. Highly effective for small screen with limited space
 - b. Rarely used responsive option
 - c. Positioning of content elements side by side
4. Fluid images resolve the problem of resizing of images by:
 - a. Sizing the images in proportion to Webpage's dimension
 - b. Submitting the media query
 - c. Merging elements with JavaScript
5. Slide down menu can be created by:
 - a. Adding meta viewport inside the head tag
 - b. Adding snippet as navigation markup inside the body tag
 - c. Adding a frame

Answers to Exercise

1. Conditions when can trigger major changes in layout
2. When a UI has more than three top-level views
3. Highly effective for small screen with limited space
4. Sizing the images in proportion to Webpage's dimension
5. Adding meta viewport inside the head tag

Answers to Quick Test

Quick Test 5.1

- a. False
- b. True

Quick Test 5.2

- a. True
- b. True

Session 6

Usability Studies in Responsive Web Design

Learning Objectives

In this session, you will learn to:

- Describe usability studies
- Explain the importance of usability studies
- List the steps for testing in usability studies
- Describe usability studies for Responsive Web Design

There has been a sweeping rise in Internet access due to widespread usage of mobile phones and devices. Most people these days access Internet while on the move. The direct impact of mobile devices has consequently led to creating sites that are optimized for mobile. It is essential to perform usability testing on these mobile sites to ensure efficiency.

6.1 Definition of Usability Studies

What is Usability Study?

A Usability study or usability test means evaluating performance to enhance the usability of a product or service while the end users work on it. The process of usability studies involves observing and taking notes while the users explore the product or service. It helps to determine and improve the product in terms of speed of the product, ease to use, to remember the functions, recover from errors and overall satisfaction of the end user.

Usability Study is a Black Box Technique

Usability study or usability test is one of the Black Box testing techniques. Black Box testing is a software testing method in which the functionality of the Software Under Test (SUT) is examined without examining the internal code structure, implementation details, and inner workings of the software. Refer to figure 6.1. This represents any software system that you want to test. For example, it could be an operating system such as Windows, a site such as Google, or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.



Figure 6.1: Black Box Testing

Image Courtesy: <http://guru99.com/>

Steps for Black Box Testing

The various tasks involved in performing Black Box testing are as follows:

- Observing initial requirements and specification of the system
- Selecting valid inputs to verify if the System Under Test (SUT) processes them correctly. Invalid inputs are also selected to check that SUT is able to identify them
- Knowing expected output for all the inputs provided
- Creating test cases with the selected inputs
- Executing test cases
- Comparing actual output with the expected output
- Retesting fixed defects

Key components of Usability

Usability is tested based on five key components:

- **Learnability:** Ease of completing basic tasks while using the product or the service
- **Efficiency:** Speed of completing the basic tasks
- **Memorability:** Remembering how to use the product effectively even after a period of time
- **Errors:** Detecting the frequency and the severity of the errors
- **Satisfaction:** Feeling of contentment after using the system

6.2 Importance of Usability Studies

Usability testing identifies the problems in the design of the product before they are coded. An early identification of problems and their resolution saves time and money and ensures completion of the project as per the schedule.

Conducting a Usability Test

While conducting a usability test, the following things are considered:

- Verify if users can finish specified tasks successfully
- Ascertain the time to finish the specified task
- Validate the approval level of users with the Web function, Website, or other product
- Categorize the changes necessary to progress the user performance and satisfaction
- Evaluate the performance if it matches the usability goals

Advantages of Conducting Usability Studies

Following are the advantages of conducting usability test:

- It can be modified to perform other types of testing such as practical testing, structure integration testing, unit testing, and so on
- It can help in fixing the problems that user may face before the product is finally released. This may help to enhance the performance of the product
- It can help in determining possible mistakes, bugs, and loopholes in the structure which generally are not evident to developers and even escape the other types of testing
- It can be economical and highly effective

Benefits of Conducting Usability Study to the End User

The benefits of usability test are as follows:

- Improves the quality of software
- Easy to use software
- More readily accepted by the users

- Shortens the learning curve for new users

Methods of Usability Studies

There are two main methods to perform usability testing:

- **Laboratory Experiment:** Here, usability testing is conducted in an organized environment that is established according to pre-set scenarios. Usability tests can be performed on a real system by giving demonstrations and showing a paper prototype. User activity is documented based on their response, expressions, and screen activity. This method is observed by usability experts by using the quantitative data. This method requires more time and is more expensive than other methods of usability testing.
- **On-site Observation:** In this method, the testing is conducted on-site which enables the study of user's actual working environment. This type of testing involves long observation periods that helps to collect the real environment information. It takes less time and is less expensive than other usability testing methods.

Checklist of Usability Studies

Usability testing checklist is categorized into three parts: Accessibility, Navigation, and Content.

Section I: Accessibility

- Verify the loading time of the Website
- Check for suitable Text-to-Background contrast
- Ensure formatting of the text including font and spacing of the text
- Verify if the Website has its 404 page or any custom designed Not Found page
- Ensure addition of appropriate Alt tags for images

Section II: Navigation

Verify if:

- user easily identifies the Website navigation
- navigation options are short and easy to understand
- number of buttons/links are realistic
- the logo of the company is linked to the home page
- links are consistent on all pages and are easy to understand
- site search is present on page and are easy to access

Section III: Content

Verify if:

- URLs have meanings and are user-friendly
- HTML pages are meaningful
- Highlighting of the content is used cautiously
- Original copy is concise and illustrative
- Major headings are clear and meaningful
- Styles and colors are consistent

Quick Test 6.1

1. Usability study is a type of Black box testing
 - a. True
 - b. False
2. Usability testing identifies the problems in the design of the product before they are coded.
 - a. True
 - b. False

6.3 Testing in Usability Studies

The steps used for performing tests in usability studies are as follows:

- Create a test plan
 - Select a test environment
 - Search and assign users
 - Create test checklist or material
 - Schedule the test session
 - Interrogate with users
 - Analyze data and observations
- **Create a Test Plan**
The first step to perform in testing usability study is to create a test plan. It involves identifying the objectives and elements that need to be tested. It also specifies the features of the test group and which methods and measures should be created to perform the test.
 - **Select a Test Environment**
The next step to perform in testing usability study is to select a testing environment. In this step, an environment consisting of required software and hardware is finalized where the testing team performs their tasks.
 - **Search and Assign Users**
In this step, users on which testing is to be performed are identified depending on their usage. For example, if a testing involves ticketing process on a Website, then the users will be individuals who book tickets online.
 - **Create Test Checklist or Materials**
In this step, you need to create an interview questionnaire for the users so that all the possible conditions are addressed for the testing scenario.
 - **Schedule the Test Session**
This is the most important step of the usability test. In this step, the actual testing is performed. A moderator is a person who performs the usability testing. He/she is responsible for security and ease for users and manages the testing team.
 - **Interrogate with Users**
In this step, a moderator should take the feedback from the users. The other team members who act as observers should cautiously attend the interrogation session with users so that they can provide their valuable suggestion to the moderator.

- **Analyze Data and Observations**

The result of the usability testing depends on the result of the earlier step, for example, Interrogate users. By analyzing data and observations, the moderator can easily find the reasons of the problem. With the expertise of the testing team, the design problem can be easily fixed.

6.4 Usability Test Plan Template

You can use the following sample template (based on <http://www.usability.gov/how-to-and-tools/resources/templates/usability-test-plan-template.html>) for your usability test plan. It is used to manage and organize a usability test depicting every stage of the project. This template describes a high-level picture of the Web application or Website under the usability test and explains the important user objectives.

[Project Name]

Usability Test Plan

[Version]

[Author Name]
[Distribution Date]

Test Plan Overview
Administrative Summary
Approach
 Users/Participants
 Training
 Procedure
Roles/Responsibilities
 Instructors
 Organizer
 Data Logger
 Test Observers
Principles/Ethics
Usability Functions
Usability Metrics
 Scenario Completion
 Critical Issues
 Non-Critical Issues
 Subjective Assessments
 Scenario Conclusion Time (time on task)
Usability Objectives
 Completion Rate
 Error-free rate
 Time on Task (TOT)
 Subjective Measures
Problem Severity
 Impact
 Frequency
 Problem Severity Classification
Reporting Outcomes

Test Plan Overview

This document specifies a test plan for performing a usability test during the development of [application or Website name]. The objectives of usability testing include creating a starting point for user performance, creating and verifying user performance methods, and classifying prospective design issues to be fixed for increasing the productivity and end user satisfaction [add or delete objectives].

The goals of usability test are as follows:

- Define design variations and usability issues within the UI of the [application or Website name]. The prospective design issues may include:
 - Navigation issues – problem to find functions, excessive keystrokes to perform a function, problem to track the required screen flow.
 - Presentation issues – problem to find and properly act upon preferred data on screens, problems because of improper labeling.
 - Control usage issues – problem to use of incorrect toolbar or entry fields.
- Use the application or Website under organized test environment with representative users. Data will be used to access whether usability goals of the UI have been achieved efficiently.
- Create baseline for the performance and satisfaction levels of the UI for the forthcoming usability tests.

[Provide more information that summarizes the user groups that will contribute in the usability test and the number of user/participants from each user group that are expected to contribute. Specify whether the testing will take place in a usability lab or remotely with the expected time schedule.]

Administrative Summary

[Provide information that summarizes the specific details of the usability test for the specified

application or Website; explain a specific task that needs to be evaluated. Summarize the usability goals.]

Upon examination of this usability test plan, containing the initial drafts of the task scenarios and usability objectives for [application or Website name], with documented approval of the test plan are expected.

Approach

[Specify the number of users, different settings required for the usability test sessions, and the tools required helping users to interact with the application. Describe briefly the measures need to be collected, such as demographic information, satisfaction assessment, and suggestions for improvement.]

Users/Participants

[Explain properly the number of users expected, how they will be employed, their eligibility criteria, and expected skills or knowledge.]

The duties of users will be to complete a set of tasks efficiently and in timely manner, and to provide feedback regarding the usability and acceptability of the UI. The users will be suggested to provide authentic opinions regarding the usability of the application, and to participate in post-session questionnaires and debriefing.

[Describe how the team will select test participants to meet stated requirements. Describe if participants will have definite skills and/or setting requirements, if they will be familiar with the assessment tasks, or have experience with performing definite tasks.]

Training

[Mention any training delivered as an overview of the Web application or Website.] The users will receive and overview of the usability test practice, equipment, and software. [Specify any non-functional portion of the test environment or testing situation.]

Procedure

[Usability Lab Testing]

Users will participate in the usability test at [give the name of the testing lab here] in [location here]. A [kind of computer] with the Web application/Website and supporting software will be used in a distinctive office setting. The user's interaction with the Web application or Website will be tracked by the moderator seated in the same office. Note takers and data logger(s) will monitor the sessions in observation room by the surveillance of video camera [Mention if the testing lab has one-way mirror or video feed]. The test sessions will be recorded.

[If the moderator is seated in a control room – specify the environment and the equipment used and how communication is supported.]

The moderator will provide overview to the users on the Web application or Website and instruct the users that they will be assessing the application, rather than the moderator evaluating the users. Users will sign an informed agreement that recognizes: the input is voluntary, that input can cease at any time, and that the session will be recorded but their privacy of recognition will be private. The moderator will ask the users if they have any questions.

Users will complete a pretest demographic and setting information survey. The moderator will clarify that the time taken to finish the test task will be calculated and that examining behavior outside the task flow should not happen until the task completion. At the beginning of each task, the user will need to read the task details loudly from the hard copy and then start the task. The measurement of Time-on-task starts when the participant initiates the task.

The moderator will provide the instructions to the users to 'think aloud' so that a verbal record will be stored of their participation with the Web application or the Website. The moderator will notice and keep details of the user behavior and their comments, and system activities in the data logging application [explain how these metrics will be stored if a data logging application is not used.]

After completion of each task, the next task of users is to complete the post-task questionnaire and then discuss about the task session with the moderator. Once done, the users will complete the post-test satisfaction questionnaire.

[For Remote Testing]

Users will participate in the usability test through remote screen-sharing tools by sitting on their workstations. They will use the telephone service for verbal communication.

Users will complete a pretest demographic and background information questionnaire. The moderator will clarify that the time taken to finish the test task will be calculated and that examining behavior outside the task flow should not happen until the task completion. At the beginning of each task, the user will need to read the task details loudly from the hard copy and then start the task. The measurement of Time-on-task starts when the participant initiates the task.

The moderator will provide the instructions to the users to broadly so that an unwritten record will be stored of their contribution with the Web application or the Website. The moderator will notice and keep details of the user behavior and their comments, and system activities in the data logging application.

After completion of each task, the next task of users is to complete the post-task questionnaire and then discuss about the task session with the moderator. Once done, the users will complete the post-test satisfaction questionnaire.

Roles/Responsibilities

The roles and responsibilities involved in a usability test are mentioned here. An individual may support multiple roles and responsibilities but tests may not require all roles and responsibilities.

Instructors

- Impart training on overview of the Web application/Website prior to the usability testing.

Organizer

- Provides a brief of study to users.
- Explains usability and objective of usability testing to users.
- Helps to maintain proper conduct of users and observer during the debriefing sessions.
- Provides assistance to users by responding to their queries.

Data Logger

- Keeps track of the actions and comments of the users.

Test Observers

- Quiet observer
- Help the data logger in finding problems, concerns, coding bugs, and procedural errors.
- Act as note takers.

Principles/Ethics

Every person involved with the usability test are required to follow to the following principles or ethical guidelines:

- The performance of any test user must not be individually attributable. Individual user's name should not be used in reference outside the testing session.
- A description of the user's performance should not be reported to his/her manager.

Usability Functions

[The usability tasks performed by users were designed from the test scenarios created from use cases and/or with the help of a subject-matter expert. These tasks are identical for all test users of a given role in the study.]

The task descriptions should be verified by the application owner, business-process owner, development owner, or operation executive to be certain that the content, layout, and presentation are of real use and assess the validity of the Web application or Website. Their acceptance is to be documented prior to usability test.

[Explain the test setup of Web application or Website consisting of special development environments, test databases. It should also explain concurrent development activities that may control the availability or performance of the test Web application or Website.]

Usability Metrics

Usability metrics refers to measurement of user performance depending on specific performance goals required to fulfill usability requirements. It consists of various factors, such as scenario completion success rates, adherence to dialog scripts, error rates, subjective evaluations, and time-to-completion.

[Add or remove any metrics not used in the planned test.]

Scenario Conclusion

Each scenario requires that the users get specific data that would be used to complete a task. The scenario assumes to be completed when the user indicates the scenario's goal has been obtained whether successfully or unsuccessfully.

Critical Issues

Critical issues are deviations at conclusion from the targets of the scenario. Finding or reporting of the incorrect data value because of user workflow is a critical issue. Users may or may not be aware that the task goal is incorrect or incomplete.

Scenario completion is the essential thing that should be completed in the usability testing. If the scenario becomes unobtainable then the action takes the form of critical issues. In general, critical issues are unresolved issues at the time of completing the task or errors that result in an incorrect consequence.

Non-Critical Issues

Non-critical issues recover from the user or, if not discovered, do not result in processing problems or unexpected results. Although non-critical issues can be unnoticed by the user, when they are noticed they are generally frustrating to the users.

These issues may be practical, in which the user does not complete a scenario in the most ideal situations, such as excessive steps and keystrokes. These non-critical issues may also be errors of confusion because of choosing the incorrect function initially, using a user-interface control incorrectly and trying to modify an un-editable field.

Subjective Assessments

Subjective assessments and satisfaction is evaluated using questionnaires, and during debriefing at the time of concluding the test session.

Scenario Conclusion Time (time on task)

The time to complete each scenario, excluding subjective assessments durations, will be recorded.

Usability Objectives

The next section explains the usability objectives for [Web application name or Website name].

Completion Rate

Completion rate is the percentage of test users who successfully complete the task without critical issues. A critical issue is an incorrect or incomplete outcome. In other words, the completion rate represents the percentage of users who have successfully finished their tasks. Note: If a user needs assistance for getting a correct output then the task is counted as a critical issue and the overall completion rate for the task is affected.

A conclusion rate of [100%/enter conclusion rate] is the objective for each task in this usability test.

Error-free rate

Error-free rate is the percentage of test users who complete the task without any issues (critical or non-critical issues). A non-critical issue does not have an impact on the final outcome of the task but would result in the task being completed less efficiently.

An error-free rate of [80%/enter error-free rate] is the goal for each task in this usability test.

Time on Task (TOT)

The time to complete a scenario is also known as 'time on task'. It is calculated from the time when the user starts the scenario to the time the user notifies the completion.

Subjective Measures

Subjective measures are calculated by considering time to perform each task, features, and functionality of these tasks. At the end of the usability test, users will scale their satisfaction with the overall system. With the help of interview or debriefing session, approach of the users is measured in respect to the usability test.

Problem Severity

To provide a rank to recommendations, a process of difficulty severity arrangement will be used in the examination of the data collected throughout review. The approach treats problem severity as a mixture of two features - the impact of the problem and the frequency of users experiencing the

problem during assessment.

Impact

Impact is the grading of the problem by defining the level of impact that the setback has on successful task end. There are three levels of impact:

- High: Avoids the user from completing the task. It is known as critical issue.
- Medium: Affects users but the task can be completed. It is known as non-critical issue.
- Low: Affects the task completion of users minimally. It is known as non-critical issue.

Frequency

Frequency is the percentage of users who experience problem while working on a task.

- High: 30% or more.
- Moderate: 11% - 29%.
- Low: 10% or fewer.

[The percentages may need to be adjusted if there are less than ten participants in a group. For example, for a study with eight participants, the low frequency should be 12.5% (1/8 = .1250).]

Problem Severity Classification

Problem severity classification is used to categorize the problems according to the level of severity of problem. If the problem falls in higher severity category then it needs immediate attention to resolve the issue and the product is not released. If the severity of the problem is not high then the product can be released but would need immediate attention to resolve the issue.

- **Severity 1:** Problems that falls in severity1 category highly impact the functioning of the product and prevent users from correctly completing a task. The occurrence of these problems varies and user is required to call the Help Desk. Resolving this type of problems results in reduced redevelopment cost and fewer Help Desk calls.
- **Severity 2:** Problems are moderate to high occurrence problems, which have low impact on the usage of product. Resolving these types of problems result in reducing the time of completing the task and decrease in cost of training.
- **Severity 3:** Problems are either moderate problems with less occurrence or low problems with moderate occurrence. These are minor problems and are faced by number of participants. Resolving these kind of problems results in increasing the data integration and reduction in completing the task.
- **Severity 4:** Problems are low risk problems with lower impact on the product. There is low risk if the problem occurs and may not need immediate attention. Resolving these problems result in increased user satisfaction.

Reporting Outcomes

The Usability Test Report is generated at the end of the usability test. It includes presentation of results, evaluation of usability metrics against the pre-approved goals, subjective evaluations, specific usability problems and recommendations for resolution. The Usability Test Report is expected to be sent to the Project UCD Contact by [enter date].

Limitations of Usability Testing

Process of Usability testing includes planning and data collection that is tedious and time consuming.

The conclusion derived from usability testing is not 100% accurate as the scenarios for testing are different from the actual environment. People tend to give different reactions to particular situations when they are aware that they are being observed for the testing. Usability testing verifies experience of users using products for a very short time but it is not possible to know the response of users in the long term.

6.5 Usability Testing for RWD

Responsive Web Design is a method of Web page creation that makes use of flexible layouts, flexible images, and cascading style sheet. The goal of responsive design is to create Web pages that change the layout according to the screen size of the device. Usability testing for responsive design verifies that Responsive Web Design provides the same user experience across all the devices.

Testing for mobile usability is considered as complex, expensive, and time-consuming process.

Mobile Website Testing

Mobile Website testing should be conducted on mobile devices such as mobile phones and tablets. It should not be conducted on static devices such as laptops. Testing a mobile interface on a static device defeats the purpose of testing due to following reasons:

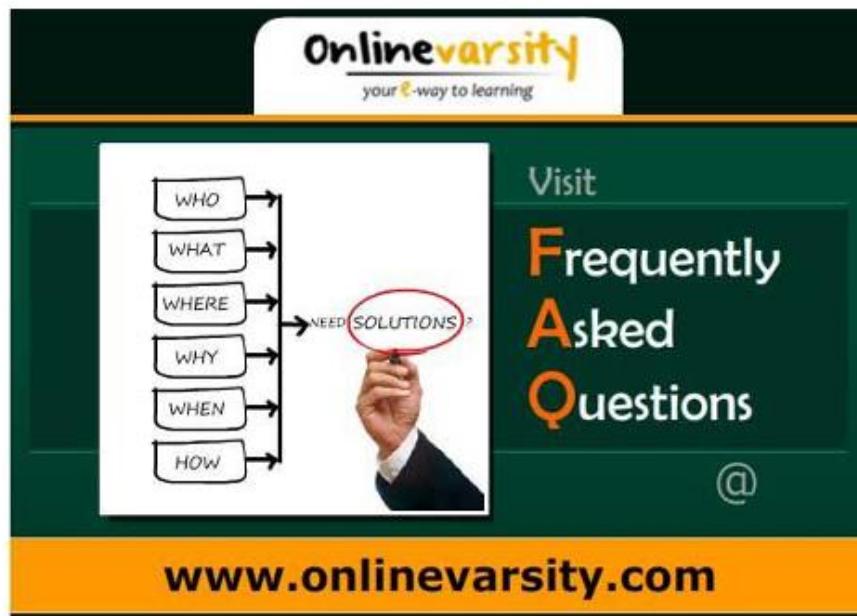
- **Roll-overs:** Roll-overs used for navigating or accessing content on static devices do not translate to mobile devices automatically.
- **Re-sizing:** Display in static and mobile devices differ due to variation of screen size.
- **Overlays and Pop-ups:** Some of the features such as light boxes and slide shows use of overlays or pop-ups that may work accurately on the static devices whereas, light boxes and slide shows may lead users into navigational dead ends on mobile devices.
- **Environmental Factors:** Websites created for mobile devices need verification for test designs, color choices, and textures in a variety of environmental light. Mobile devices are used in different environments, whereas static devices are used in a particular environment.

Tools for Responsive Design Testing

Internet access through mobile devices is gradually increasing so it is important that Website should be mobile friendly. Following free tools that can be used for Responsive Design Test are as follows:

- **Responsive Test:** This tool helps to know how a Web page will look on devices with different sizes of screen. Some devices supported with this tool are as follows: iPhones, Blackberry and Samsung phones, and Dell laptops.
- **Responsinator:** This tool accepts a URL and gives output of sequences of device mock-ups by rendering the page. This gives an idea of how the users will experience the layout of Web page on different devices.
- **Responsive:** This tool has a very good feature that can provide you some keyboard shortcuts. For example, when you press the 'T' key, you will get a tablet preview. This tool helps users to often switch between different device previews.
- **Am I Responsive?:** This tool renders preview of images and is helpful for presentational design meetings.
- **Viewport Resizer:** It allows resizing the browser into a specific dimension. For example, the dimension of an iPhone and Amazon Kindle Fire.

- **ResizeMyBrowser:** It allows resizing the browser into 15 default sizes that matches almost all the popular devices such as the MacBook's or iPads. It also allows creating custom dimensions and lets you know the size of current window.
- **Screenfly:** It takes a URL and provides a preview of Web pages according to various screen dimensions.
- **Responsive Web Design Tool by Designmodo:** It helps in designing and debugging of responsive breakpoints.
- **Responsive Web Design Tool by pixeltuner.de:** It renders a URL in the several device mock-ups.



6.6 Summary

In this session, **Usability Studies in Responsive Web Design**, you learned that:

- Usability studies mean evaluating performance to enhance the usability of a product or service while the end users work on that particular product or service.
- Usability study is a type of Black Box technique.
- Black Box testing is a software testing method in which the functionality of the Software Under Test (SUT) is examined without peering at the interior system configuration, operation details, and internal paths of the software.
- Usability testing identifies the problems in the design of the product before they are coded.
- There are two methods to perform the usability studies are: Laboratory experiment and on-site observation.
- Usability testing checklist is categorized into three parts: Accessibility, Navigation, and Content.
- The most common elements of a GUI include Window, Menu, Icons, and a Pointer.
- Mobile Website testing should be conducted on mobile devices such as mobile phones and tablets.
- Various free tools used for Responsive Design Test, such as Responsive Test, Responsive, and Screenfly.

6.7 Exercise

1. Which of the software testing method is used to examine the Software Under Test (SUT) without examining the internal code structure, implementation details and internal workings of the software?
 - a. Black Box testing
 - b. White Box testing
 - c. Stroke testing
 - d. Blue Box testing
2. Which of the following is not a key component of usability testing?
 - a. Learnability
 - b. Efficiency
 - c. Memorability
 - d. Navigating
3. Which of the following is not categorized as a part of the Usability testing checklist?
 - a. Learnability
 - b. Accessibility
 - c. Navigation
 - d. Content
4. The time to complete a scenario is also known as _____.
 - a. Time on Task
 - b. Timing
 - c. Schedule
 - d. Time Control

Answers to Exercise

1. Black Box testing
2. Navigating
3. Learnability
4. Time on Task

Answers to Quick Test

Quick Test 6.1

- a. True
- b. True