

**MAHENDRA ARTS AND SCIENCE COLLEGE
(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE &
APPLICATIONS**

Paper code : :

Paper Name : Mobile Application Development

Class : III BSc CS

Semester : V

Presented by
P.Sangeetha, AP/CSA,MASC

SYLLABUS

UNIT –I

Introduction: Brief History of Mobile –Evolution of Devices: Brick Era – Candy Bar Era – Feature Phone Era-Smart Phone Era-Touch Era. **The Mobile Ecosystem:** Operators- Networks-Devices-Platforms-Operating Systems- Application Frameworks- Applications-Services. **Why Mobiles?** : Mobile as a Medium.

What is meant by Mobile Application Development?

- Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources.
- A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a Smartphone or tablet computer.
- Mobile applications frequently serve to provide users with similar services to those accessed on PCs

BRIEF HISTORY OF MOBILE

- The evolution of mobile networks, the devices that run on them and the services we use every day have evolved at an amazing rate, from the early phones that looked more like World War II field radios to the ultra-sleek fashion statements of today.
- It is the history, or the context, of the medium that gives mobile designers and developers the patience and passion needed to deal with the frequent issues they face in the mobile ecosystem.
- The mobile industry is a difficult one to jump into without patience and passion.

IN THE BEGINNING

- The telephone is undoubtedly one of the greatest inventions of mankind.
- It revolutionized communications, enabling us to reach across great distances and share thoughts, ideas, and dreams with our fellow man, making the world a much smaller place in the process.
- In fact, the telephone is probably one the most defining technologies of the twentieth century and the most commonly used electronic device in the world today.



Fig: The traditional telephone



Fig: A modern mobile phone

- Although the modern mobile phone is a distant cousin to the telephone, it is a communication and information device.
- It is nearly always connected to the Internet.
- You can send and receive voice and text messages. You can purchase goods and services.
- In fact, the modern mobile phone is capable of doing nearly everything you can do with a desktop computer, but with the potential for more meaningful relevance to our daily activities.
- The mobile phone is not a telephone. In fact, modern mobile devices deliver on the long-overdue promises that technology will make our lives easier.

THE EVOLUTION OF DEVICES

- ❖ Brick Era
- ❖ Candy Bar Era
- ❖ Feature Phone Era
- ❖ Smart Phone Era
- ❖ Touch Era

What is ERA?

A Full form is Electrical Replaceable Assembly.

ERA is an electrical assembly that can be easily removed from a piece of electronic equipment in computer and replaced without having to repair the entire product or system.

1. BRICK ERA

- The first era was called as the Brick Era (1973–1988). It was suitcase phone that basically a corded receiver connected to a portable radio the size (and weight) of a car battery.
- It was more modern and portable mobile phone that powerful enough to get a signal in the remote areas.
- Brick Era phones proved useful only to those who truly needed constant communication, such as stockbrokers or those who worked in the field, such as salespeople or real estate agents.



Fig: The Motorola DynaTAC 8000X was the first mobile phone in 1983; DynaTAC was actually an abbreviation of Dynamic Adaptive Total Area Coverage.

•2. THE CANDY BAR ERA

- The second era, the Candy Bar Era (1988–1998), represented one of the more significant leaps in mobile technology.
-
- “Candy bar” is the actual term used to describe the long, thin, rectangular form factor of the majority of mobile devices used during the Candy Bar Era and even today (see Figure).
- Candy bar phones commonly associated with 2G GSM (Global System for Mobile communications) networks— included SMS (Short Message Service) capabilities.



Fig: A Nokia candy bar phone

3. THE FEATURE PHONE ERA

- The third era, the Feature Phone Era (1998–2008), wasn't nearly as radical a technological leap as the leap from the Brick Era to the Candy Bar Era, but it was an important evolution nonetheless.
- Up to this point, mobile phones had done three things: make voice calls, send text messages, and play the Snake game.
- During this era, GSM network providers added GPRS (General Packet Radio Service), allowing packet-switched data services.
- This network evolution is most often referred to as 2.5G, or halfway between 2G and 3G networks.



Fig: The Motorola RAZR, probably the most iconic device from the Feature Phone Era

- The introduction of the Motorola V3, more commonly known as the RAZR. Although the RAZR was not a technologically advanced phone, became the second-best-selling mobile phone of all time.

•THE SMARTPHONE ERA

- The Smartphone Era occurred at the same time as the third and fifth eras and spans from around 2002 to the present.
- Although Smartphone's have all the same capabilities of a feature phone, like making a phone call, sending an SMS, taking a picture, and accessing the mobile web, most smartphones are distinctive in that they use a common operating system, a larger screen size, a QWERTY keyboard or stylus for input, and Wi-Fi or another form of high-speed wireless connectivity.



Early Smartphone's came from companies like Nokia, Handspring, and Research in Motion (RIM)

- **Nokia 9000**

The Nokia 9000 Communicator was the first product in Nokia's Communicator series, announced at CeBIT 1996 and introduced into the market on 15 August 1996. The phone was large and heavy at 397 grams but powerful at the time.

- **Handspring**

In 2002, it introduced the **Handspring** Treo, one of the first smartphones that integrated a **cellphone** and a PDA into a single device.

- **Research in Motion (RIM)**

Research in Motion applied its background in two-way paging to create the first BlackBerry, which would later be used to “push” email to corporate citizens in a pager-like fashion.

5. TOUCH ERA

- Mobile devices started as simple portable telephones, but they evolved. Messaging was added to mobile capabilities, but mobile devices were still just person-to-person communication tools.
- We saw networks improve and data speeds increase, which allowed for more technology and more features each year, crammed into smaller and smaller packages.
- Mobile devices got smarter by learning from desktop computing, truly becoming personal computers, but people weren't interested.

- January 9, 2007, Steve Jobs went onstage at the MacWorld conference in San Francisco to us her in the fifth and final era and to change the mobile world. He introduced the world to the iPhone.
- iPhone 3G as one of the most significant milestones that the mobile industry has ever seen.
- Within just six months of the launch of the iPhone 3G and the ability to purchase and load applications onto the iPhone, the iTunes App Store had already seen its more than 10,000 applications downloaded over 300 million times, at a rate of 2 million per day.

THE MOBILE ECOSYSTEM

- The mobile ecosystem instead as a system of layers, as shown in [Figure](#). Each layer is reliant on the others to create a seamless, end-to-end experience.

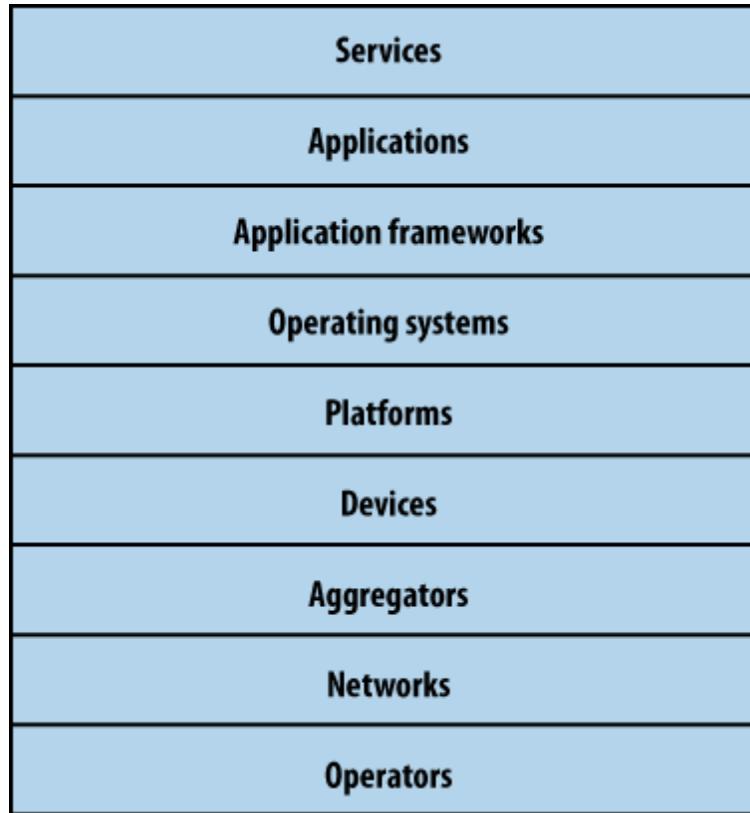


Fig: The layers of the mobile ecosystem

Queries ?

SHORT STORY

➤The Foolish Donkey

- A salt seller used to carry the salt bag on his donkey to the market every day.
- On the way they had to cross a stream. One day the donkey suddenly tumbled down the stream and the salt bag also fell into the water. The salt dissolved in the water and hence the bag became very light to carry. The donkey was happy.
- Then the donkey started to play the same trick every day.

- The salt seller came to understand the trick and decided to teach a lesson to it. The next day he loaded a cotton bag on the donkey.
 - Again it played the same trick hoping that the cotton bag would be still become lighter.
 - But the dampened cotton became very heavy to carry and the donkey suffered. It learnt a lesson. It didn't play the trick anymore after that day, and the seller was happy.
- **Moral of the story:**
- Luck won't favor always.

Previous Topics.....

- ▶ **Introduction**
- ▶ **What is meant by MAD?**
- ▶ **Brief History of Mobile**
- ▶ **Evolution of Devices**

Today Discuss about.....

- The Mobile Eco System
- Operators
- Networks
- Devices
- Platforms
- Operating System
- Application Framework
- Application Services

THE MOBILE ECOSYSTEM

- The mobile ecosystem instead as a system of layers, as shown in Figure. Each layer is reliant on the others to create a seamless, end-to-end experience.

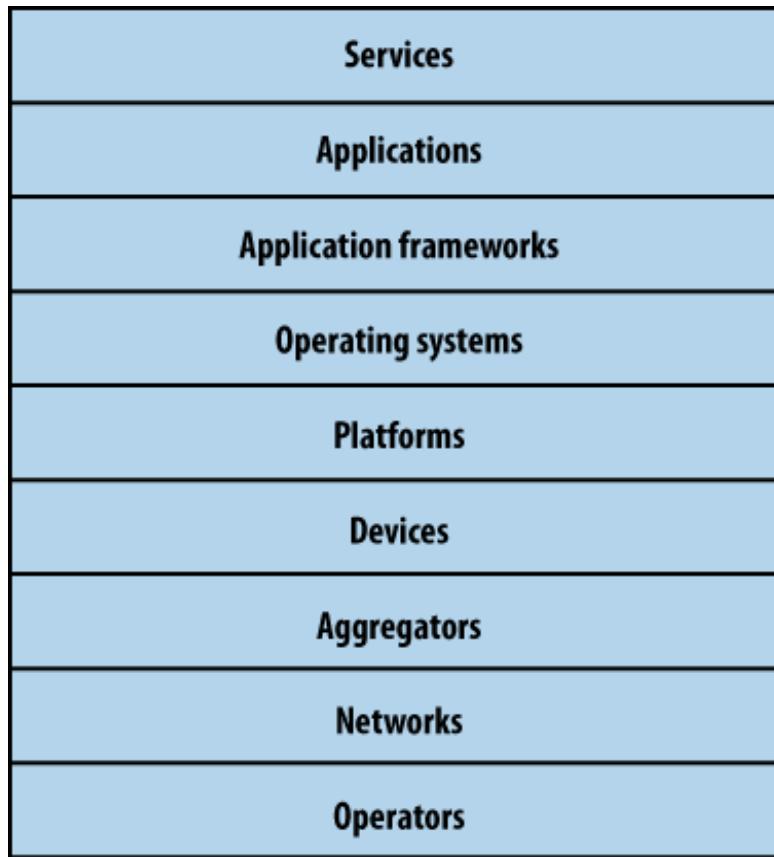


Fig: The layers of the mobile ecosystem

Operators:

- ▶ The base layer in the mobile ecosystem is the operator.
- ▶ Operators can be referred to as Mobile Network Operators (MNOs); mobile service providers, wireless carriers, or simply carriers; mobile phone operators; or cellular companies.
- ▶ Operators are what essentially make the entire mobile ecosystem work. They are the gatekeepers to the kingdom.
- ▶ They install cellular towers, operate the cellular network, make services (such as the Internet) available for mobile subscribers, and they often maintain relationships with the subscribers, handling billing and support, and offering subsidized device sales and a network of retail stores.
- ▶ For example, they have had to establish trust with subscribers to handle the billing relationship and to offer devices, content, and services that often compete with their partners, who are people like us and who want to create content and services for mobile devices.
- ▶ Table lists the markets, technologies used, and subscriber numbers for the world's largest operators.

Operator	Markets	Technology	Subscribers (in millions)
China Mobile	China (including Hong Kong) and Pakistan	GSM, GPRS, EDGE, TD-SCDMA	436.12
Vodafone	United Kingdom, Germany, Italy, France, Spain, Romania, Greece, Portugal, Netherlands, Czech Republic, Hungary, Ireland, Albania, Malta, Northern Cyprus, Faroe Islands, India, United States, South Africa, Australia, New Zealand, Turkey, Egypt, Ghana, Fiji, Lesotho, and Mozambique	GSM, GPRS, EDGE, UMTS, HSOPA	260.5
MTS	Russia, Ukraine, Belarus, Uzbekistan, Turkmenistan, and Armenia	GSM, GPRS, EDGE, UMTS	91.7
Bharti Airtel	India, Seychelles, Jersey, Guernsey, and Sri Lanka	GSM, GPRS, EDGE	72.0

Networks:

- ▶ Operators operate wireless networks. Remember that cellular technology is just a radio that receives a signal from an antenna. The type of radio and antenna determines the capability of the network and the services you can enable on it.
- ▶ The vast majority of networks around the world use the GSM standard (see Table for an explanation of these acronyms), using GPRS or GPRS EDGE for 2G data and UMTS or HSDPA for 3G.
- ▶ We also have CDMA (Code Division Multiple Access) and its 2.5G hybrid CDMA2000 which offers greater coverage than its more widely

2G	Second generation of mobile phone standards and technology	Theoretical max data speed
GSM	Global System for Mobile communications	12.2 KB/sec
GPRS	General Packet Radio Service	Max 60 KB/sec
EDGE	Enhanced Data rates for GSM Evolution	59.2 KB/sec
HSCSD	High-Speed Circuit-Switched Data	57.6 KB/sec

3G	Third generation of mobile phone standards and technology	Theoretical max data speed
W-CDMA	Wideband Code Division Multiple Access	14.4 MB/sec
UMTS	Universal Mobile Telecommunications System	3.6 MB/sec
UMTS-TDD	UMTS + Time Division Duplexing	16 MB/sec
TD-CDMA	Time Divided Code Division Multiple Access	16 MB/sec
HSPA	High-Speed Packet Access	14.4 MB/sec
HSDPA	High-Speed Downlink Packet Access	14.4 MB/sec
HSUPA	High-Speed Uplink Packet Access	5.76 MB/sec

Devices:

- Let's focus on the biggest slice of the device pie—mobile phones. As of 2008, there are about 3.6 billion mobile phones currently in use around the world; just more than half the plan

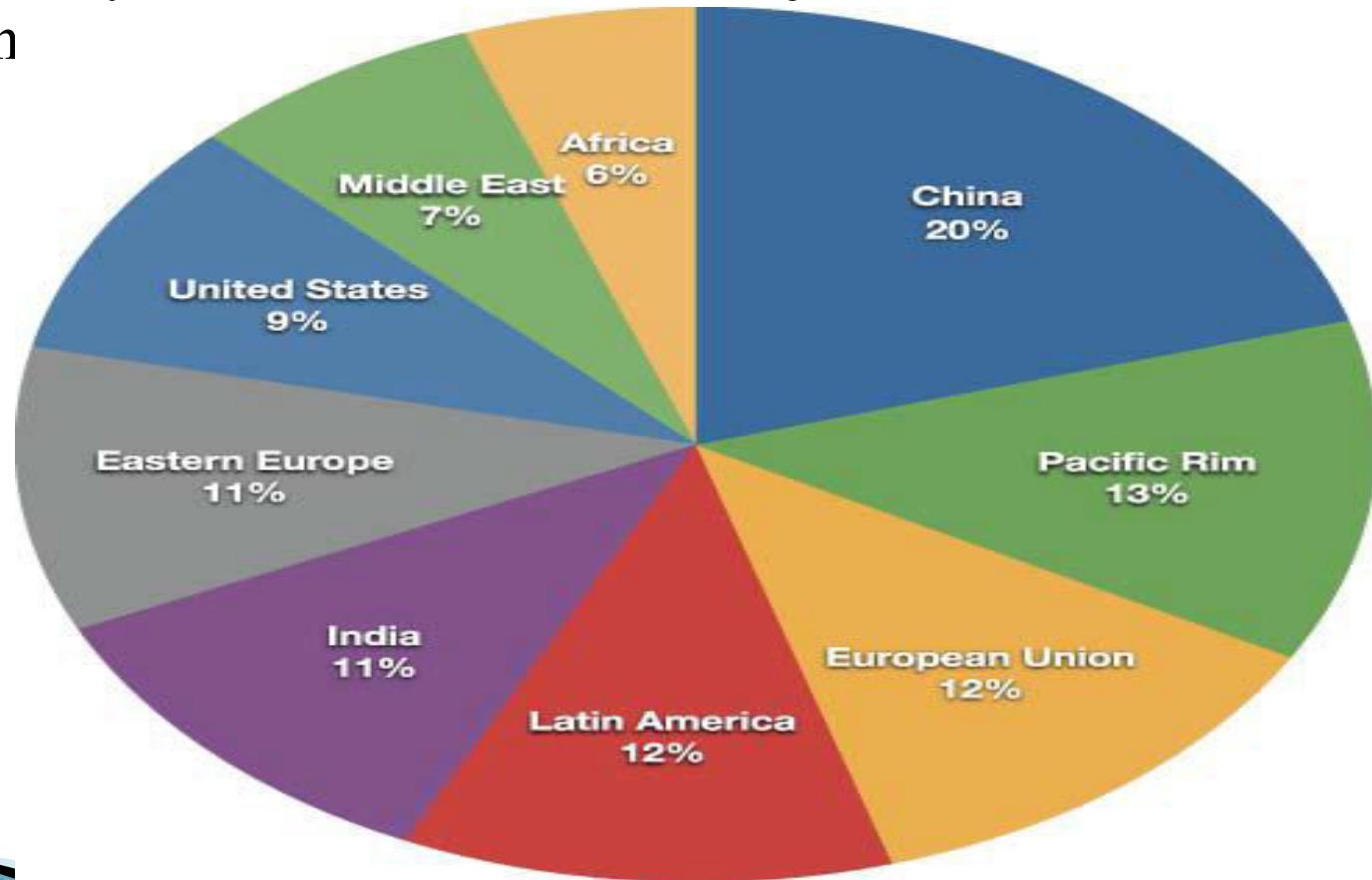


Fig: Mobile devices around the world

- ▶ As next-generation devices become a reality, the distinction between feature phones and smartphones will go away. In the next few years, feature phones will largely be located in emerging and developing markets. Figure shows a breakdown of devices.
- ▶ Subsidization means that devices need to be *provisioned* (or customized) to operators' individual requirements. Provisioning dramatically increases the number of devices released every year, with each device being slightly different from ~~the other~~.

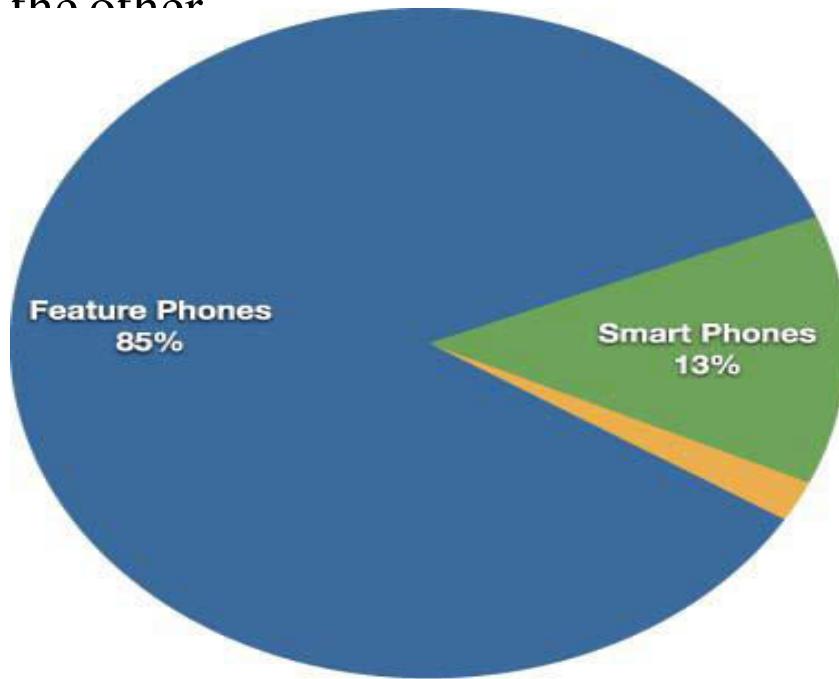


Fig: Breakdown of devices

Platforms:

- ▶ A mobile platform's primary duty is to provide access to the devices.
- ▶ To run software and services on each of these devices, you need a *platform*, or a core programming language in which all of your software is written.
- ▶ Like all software platforms, these are split into three categories:
 - Licensed
 - Proprietary
 - Open source.

Licensed:

- ▶ The goal is to create a common platform of development Application Programming Interfaces (APIs) that work similarly across multiple devices with the least possible effort required to adapt for device differences, although this is hardly reality.
 - ▶ Following are the licensed platforms:
 - ***Java Micro Edition (Java ME)***
- Formerly known as J2ME, Java ME is by far the most predominant software platform of any kind in the mobile ecosystem.

Licensed:

- The goal is to create a common platform of development Application Programming Interfaces (APIs) that work similarly across multiple devices with the least possible effort required to adapt for device differences, although this is hardly reality.
- Following are the licensed platforms:
 - ***Java Micro Edition (Java ME)***
- Formerly known as J2ME, Java ME is by far the most predominant software platform of any kind in the mobile ecosystem.

➤ ***Binary Runtime Environment for Wireless (BREW)***

BREW is a licensed platform created by Qualcomm for mobile devices, mostly for the U.S. market.

➤ ***Windows Mobile***

Windows Mobile is a licensable and compact version of the Windows operating

➤ ***LiMo***

- ▶ LiMo is a Linux-based mobile platform created by the LiMo Foundation.
- ▶ Although Linux is open source.

PROPRIETARY

- ▶ Proprietary platforms are designed and developed by device makers for use on their devices.
 - **Palm:**
- ▶ Their first and most recognizable is the Palm OS platform based on the C/C++ programming language; this was initially developed for their Palm Pilot line, but is now used in low-end smartphones such as the Centro line.
- ▶ As Palm moved into higher-end smartphones, they started using the Windows Mobile-based platform for devices like the Treo line.
- ▶ The most recent platform is called WebOS, is based on the WebKit browser framework, and is used in the Pre line.

- ***BlackBerry***
 - ▶ Research in Motion maintains their own proprietary Java-based platform, used exclusively by their BlackBerry devices.
- ***iPhone***
 - ▶ Apple uses a proprietary version of Mac OS X as a platform for their iPhone and iPod touch line of devices, which is based on UNIX.
- ▶ **OPEN SOURCE**
 - ▶ Open source platforms are mobile platforms that are freely available for users to download, alter, and edit.
 - ▶ Open source mobile platforms are newer and slightly controversial, but they are increasingly gaining traction with device makers and developers.
 - ▶ Android is one of these platforms. It is developed by the Open Handset Alliance, which is spearheaded by Google.

Operating Systems:

- ▶ Operating systems often have core services or toolkits that enable applications to talk to each other and share data or services.
 - ▶ Mobile devices without operating systems typically run “walled” applications that do not talk to anything else.
- ***Symbian***
- ▶ Symbian OS is a open source operating system designed for mobile devices, with associated libraries, user interface frameworks, and reference implementations of common tools.
- ***Windows Mobile***
- ▶ Windows Mobile is the mobile operating system that runs on top of the Windows Mobile platform.
- ***Palm OS***
- ▶ Palm OS is the operating system used in Palm's lower-end Centro line of mobile phones.

➤ *Linux*

- ▶ The open source Linux is being increasingly used as an operating system to power smartphones, including Motorola's RAZR2.

➤ *Mac OS X*

- ▶ A specialized version of Mac OS X is the operating system used in Apple's iPhone and iPod touch.

➤ *Android*

- ▶ Android runs its own open source operating system, which can be customized by operators and device manufacturers.

APPLICATION FRAMEWORKS

- ▶ Application frameworks often run on top of operating systems, sharing core services such as communications, messaging, graphics, location, security, authentication, and many others.
- ***Java***
- ▶ Java applications are purchased and distributed through the operator, but they can also be downloaded and installed via cable or over the air.
- **S60**
- ▶ S60 is often associated with Nokia devices—Nokia owns the platform—but it also runs on several non-Nokia devices.
- ▶ S60 is an open source framework.

➤ ***BREW***

- ▶ Applications written in the BREW application framework can be deployed across the majority of BREW-based devices, with slightly less cross-device adaption than other frameworks.
- ▶ However BREW applications must go through a costly and timely certification process and can be distributed only through an operator.

➤ ***Flash Lite***

- ▶ Adobe Flash Lite is an application framework that uses the Flash Lite and Action Script frameworks to create vector-based applications.
- ▶ Flash Lite applications can be run within the Flash Lite Player, which is available in a handful of devices around the world.

- ***Windows Mobile***
 - ▶ Applications written using the Win32 API can be deployed across the majority of Windows Mobile-based devices.
 - ▶ Like Java, Windows Mobile applications can be downloaded and installed over the air or loaded via a cable-connected computer.
- ***Cocoa Touch***
 - ▶ Cocoa Touch is the API used to create native applications for the iPhone and iPod touch.
 - ▶ Cocoa Touch applications must be submitted and certified by Apple before being included in the App Store.

- ***Android SDK***
- ▶ The Android SDK allows developers to create native applications for any device that runs the Android platform.
- ▶ By using the Android SDK, developers can write applications in C/C++ or use a Java virtual machine included in the OS that allows the creation of applications with Java, which is more common in the mobile ecosystem.
- ***Web Runtimes (WRTs)***
- ▶ Nokia, Opera, and Yahoo! provide various Web Runtimes, or WRTs. These are meant to be mini frameworks, based on web standards, to create mobile widgets.
- ▶ Both Opera's and Nokia's WRTs meet the W3C-recommended specifications for mobile widgets.
- ▶ Although WRTs are very interesting and provide access to some device functions using mobile web principles, more complex than just creating a simple mobile web app, as they force the developer to code within an SDK rather than just code a simple web app.

- **WebKit**
- ▶ With Palm's introduction of webOS, a mobile platform based on WebKit, and given its predominance as a mobile browser included in mobile platforms like the iPhone, Android, and S60.
- ▶ WebKit is a browser technology, so applications can be created simply by using web technologies such as HTML, CSS, and JavaScript.

THE WEB

- ▶ The Web is the only application framework that works across virtually all devices and all platforms.
- ▶ Applications Application frameworks are used to create applications, such as a game, a web browser, a camera, or media player. Although the frameworks are well standardized, the devices are not.
- ▶ The largest challenge of deploying applications knows the specific device attributes and capabilities.
- ▶ For example, if you are creating an application using the Java ME application framework, you need to know what version of Java ME the device supports, the screen dimensions, the processor power, the graphics capabilities, the number of buttons it has, and how the buttons are oriented.

- ▶ The mobile web browser is an application that renders content that is device-, platform-, and operating-system-independent.
- ▶ The web browser knows its limitations, enabling content to scale gracefully across multiple screen sizes.
- ▶ However, like all applications, mobile web browsers suffer from many of the same device fragmentation problems.

8.SERVICES

- ▶ Finally, we come to the last layer in the mobile ecosystem: services.
- ▶ Services include tasks such as accessing the Internet, sending a text message, or being able to get a location—basically, anything the user is trying to do.

DAY 3

Previous Topics :

The Mobile Ecosystem:

- ❖ Operators
- ❖ Networks
- ❖ Devices
- ❖ Platforms
- ❖ Operating Systems
- ❖ Application Frameworks
- ❖ Applications-Services

Topics to Discuss Today:

Why Mobiles?

- ❖ Mobile as a Medium

WHY MOBILES?

- To understand how to design and develop mobile products that benefit users, and therefore long-term business, you need to answer the question, “Why Mobile?”
- Mobile is not only a new medium, but also a new business model entirely.
- There is opportunity aplenty, but the trick is learning how to harness the market to the benefit of the business.
- Being a visionary doesn’t have to be rocket science. Taking the leap into the mobile market doesn’t have to be risky; it simply requires an honest look at what exists, what users want, and then taking the next logical step.

MOBILE AS A MEDIUM

- In this sections briefly summarize each of the seven mass media.
- By better understanding what the users are accustomed to and what purpose media plays in people's lives and can better understand how to leverage the mobile medium to suit goals.

The Printing Press

- The first mass medium was the printing press, one of the greatest inventions of mankind.
- The time needed to publish information was dramatically reduced, enabling information to be copied and distributed farther and faster than handwritten predecessors.
- The printing press has continually played a crucial role in history.

- For example, during the American Revolutionary War, the printing press was used to mass-produce the record of civil unrest occurring in Boston, the epicenter of the colonial uprising.
- It is hard to imagine that the Continental Congress would have had the public support needed to form the United States without the aid of the printing press and people like Boston printer Samuel Adams.

Recordings

- The second mass medium was the recorded sound, initially on Edison's phonograph cylinder and later on more durable materials like glass, vinyl, magnetic tape, or today's compact disc.
- Although we normally associate recordings with music, early recordings enabled people to share information by hearing it firsthand and recreating the experience by sharing it over time and over great distances.
- But recorded music also played an important part in influencing society.

Cinema

- The third mass medium was the cinema.
- Like recordings, we tend to think of cinema as entertainment, but cinema enabled a visual experience to be shared over time and distance.
- Suddenly, we were able to witness distant or past events firsthand, enabling the viewer to draw conclusions from what he or she saw and heard.

Radio

- The fourth mass medium was radio—an extension of recordings, but including the live broadcast of material.
- Information could be distributed as it happened and as far as the radio signal would reach.
- Like cinema, radio could give listeners an intensely personal experience.
- And because recording technology was becoming smaller, events could be recorded where film cameras could not go.

Television

- Television is the fifth mass medium.
- The early days of television were more of a visual extension of radio.
- As the price of televisions dropped and they entered more homes, television transformed itself into a more iconic medium—one of the most influential and certainly one of the most disruptive.
- The television became a more practical alternative to previous media like cinema and radio. Suddenly, we could participate with information in more intimate and visceral ways.

The Internet

- Nothing seemed to happen for a long time after the invention of the television—that is, not until we started plugging our computers into the phone jack to hear that weird tone that meant we were connecting to the Internet and the World Wide Web.
- Among the important early Internet developments were the dot-coms(.com), which gave us a reason to have a computer in our home and not just at work.
- Then came Web 2.0, which showed us that the Web could be used in meaningful ways.
- It took a little time to find its path, but eventually the Internet would become a transformative and disruptive mass medium in its own right.

- The Web of today is threatening the printing press and crumbling newspaper empires that have been around for a century.
- The iTunes Music Store, which sells digital content over the Internet, is now the largest purveyor of recordings in the world.
- User can purchase, download, or stream movies through many Internet-connected devices.
- We have laptops and Wi-Fi, but those technologies are local, offering only a small number of clients with a broadcast distance of a few meters.

Mobile

- The seventh mass medium, of course, is mobile technology.
- The mobile industry actually started around the same time as the Web, but it took it years for us to consider it a mass medium.
- The mobile medium is actually quite deceiving; it would be easy to see it as an extension of the previous media, but mobile is actually quite unique, as the only mass medium that can do everything the previous six media can do.
- Understanding how the mobile medium stands apart from other media is an important step in determining how to best leverage mobile for business goals.
- The following sections discuss not just how real people can use mobile technology in place of previous media, but also the unique and competitive benefits.

Read and publish

- Reading text is consistently one of the most frequently performed tasks on a mobile device, second only to making and receiving phone calls.
- From sending an SMS (Short Message Service) or email message to reading some news or even a book, we are increasingly using mobile devices beyond just communication with one another and now for distributing knowledge and for absorbing information.

- New technologies like Amazon's Kindle provide a new method of reading, both on a specific reading device and on devices like the iPhone.
- By using mobile technology, we can synchronize our reading position among contexts; browse, purchase, and download content over the air; and even interact with our friends and our reading groups, giving our thoughts and opinions, regardless of location.

Play recordings

- The iPod is still the most popular portable media device around the world, but that hasn't stopped mobile device makers from including media playback features in just about every new mobile device.
- We're certainly seeing that people don't want to have to carry multiple devices around if they don't have to, which explains part of the rapid popularity of the iPhone.
- The opportunity for sharing recordings over the air with friends, or by proximity, provides a new kind of word-of-mouth advertising, one that could revolutionize the recording industry.

Watch movies

- In watching movies in mobile: imagine watching trailers for movies showing at nearby theaters, or finding out at which nearby theaters your friends are in line getting tickets.
- Even go beyond traditional entertainment: imagine watching clips of events happening around you, or maybe interacting with your social network in real time as you watch videos wherever you are.
- The opportunities are endless.

Listen to radio

- Many mobile devices include a radio tuner to play broadcasts from the local radio waves in the air.
- But we can also stream live broadcasts from the BBC, National Public Radio, and others, getting information as it happens from the source we want, regardless of our locale.

Watch television

- Many operators use the ability to watch live television on a mobile device to highlight the capabilities of their mobile broadband networks, though it has been slow to catch on with consumers.
- In countries like Japan and South Korea, seeing someone watch television on a mobile device is a commonplace occurrence; in other countries, not so much.
- The television networks are only now starting to understand and extend their business to the Internet. Mobile technology won't be far behind.
- Like radio, the opportunity for using the always-on connection and the personal nature of the handset means that a television broadcast like news could become contextual, relevant, personalized, location-based, or maybe based around the interests of an individual's network.

Use the Internet

- Without the Internet, many of these services wouldn't be available on a mobile device.
- Access to the Internet and usage of the Web from a mobile device is increasing exponentially every month around the world.

MOBILE'S UNIQUE BENEFITS

- “The first truly personal mass media”
- For example, we don’t usually share our mobile devices with our spouses.
- Each of the other mass media is or can be shared easily; even your computer can be shared at home or at work.
- Mobile devices offer us, for the first time, a means to interact with information in a personal and intimate way.

- “The first always-on mass media”

- Many don't realize that mobile devices have the capability to send and receive information at all times, even when idle.

- “The first always-carried mass media”

- How many other mass media can you think of that we carry with us at all times?

- Seven out of ten people sleep with their phone within arm's reach.

- “**The only mass media with a built-in payment channel**”
- Every phone sold from an operator has a built-in means of purchasing content, even goods and services known by the funny acronym of BoBo, or Billing on Behalf of.
- You pay for it and it is charged to your phone bill. When you factor in the statistic that twice as many people have phones than credit cards, suddenly the potential seems enormous.

“At the point of creative impulse”

- We are able to create content and distribute it the moment the mood strikes us.
- From taking a picture of something interesting and uploading it to social networks in order to share it with our friends to capturing a video of an important event and sharing an experience, mobile devices enable us to create and publish in near real time.
- Information and experiences can now be shared with audiences around the world as they happen and from multiple points of view. It is simply unlike any other medium we have ever seen before.

Queries?

SYLLABUS

UNIT – II

Types of Mobile Applications: Mobile Application Medium Types. **Mobile Information Architecture:** Mobile Information Architecture. **Mobile Design:** Elements of Mobile Design- Mobile Design Tools.

Today discuss about....

MOBILE APPLICATION MEDIUM TYPES:

- 1. SMS**
- 2. MOBILE WEBSITES**
- 3. MOBILE WEB WIDGETS**
- 4. MOBILE WEB APPLICATIONS**
- 5. NATIVE APPLICATIONS**
- 6. GAMES**

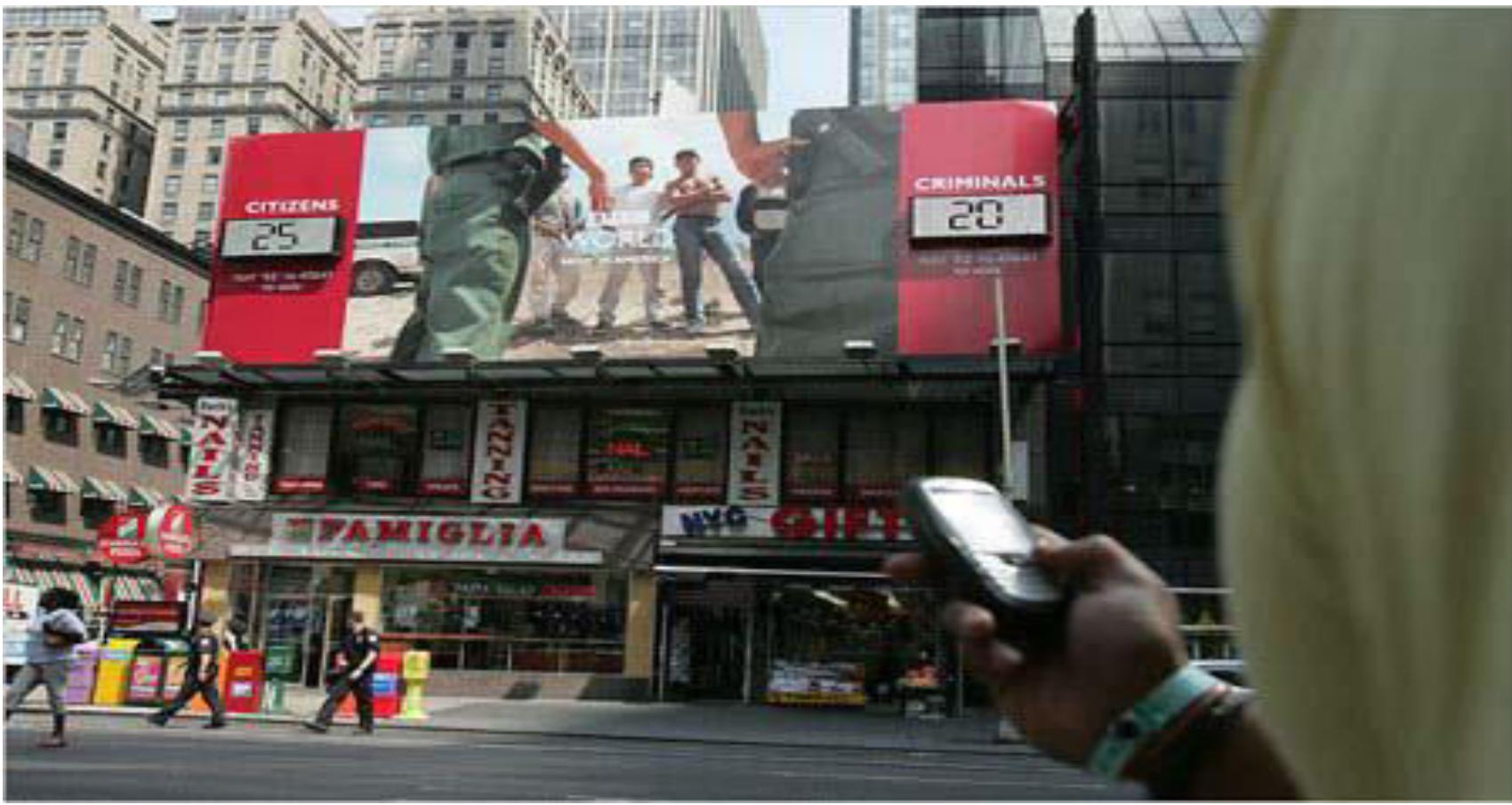
TYPES OF MOBILE APPLICATIONS:

MOBILE APPLICATION MEDIUM TYPES:

- The *mobile medium type* is the type of application framework or mobile technology that presents content. Typically, the user sends a single keyword to a five-digit short code in order to return information or a link to premium content.
- For example, sending the keyword “freebie” to a hypothetical short code “12345” might return a text message with a coupon code that could be redeemed at a retail location, or it could include a link to a free ringtone.
- SMS applications can be either “free,” meaning that there is no additional charge beyond the text message fees an operator charges, or “premium,” meaning that you are charged an additional fee in exchange for access to premium content.
- The most common uses of SMS applications are mobile content, such ringtones and images, and to interact with actual goods and services.or information to the user.
- The technical capabilities and capacity of the publisher also factor into which approach to take.
- Fig, illustrates the spectrum of mobile media; it starts with the basic text-based experiences and moves on to the more immersive experiences.

1. SMS (Short Message Services)

- The most basic mobile application you can create is an SMS application.
- Typically, the user sends a single keyword to a five-digit short code in order to return information or a link to premium content.
- For example, sending the keyword “freebie” to a hypothetical short code “12345” might return a text message with a coupon code that could be redeemed at a retail location, or it could include a link to a free ringtone.
- SMS applications can be either “free,” meaning that there is no additional charge beyond the text message fees an operator charges, or “premium,” meaning that you are charged an additional fee in exchange for access to premium content.
- The most common uses of SMS applications are mobile content, such ringtones and images, and to interact with actual goods and services.
- A great example of how SMS adds incredible value would be Twitter, where users can receive SMS alerts from their friends and post to their timeline from any mobile device, or the SMS-to-Billboard that BBC World News put up in Midtown Manhattan (Figure)



*Fig: An SMS application to interact with a billboard in
Manhattan*

Pros

- The pros of SMS applications include:
- They work on any mobile device nearly instantaneously.
- They're useful for sending timely alerts to the user.
- They can be incorporated into any web or mobile application.
- They can be simple to set up and manage.

Cons

- The cons of SMS applications include:
- They're limited to 160 characters.
- They provide a limited text-based experience.
- They can be very expensive.

2. MOBILE WEBSITES

- A mobile website is a website designed specifically for mobile devices, not to be confused with viewing a site made for desktop browsers on a mobile browser.
- Mobile websites are characterized by their simple “drill-down” architecture, or the simple presentation of navigation links that take you to a page a level deeper, as shown in Figure.



Fig: An example of a mobile website

- Though mobile websites are fairly easy to create, they fail to display consistently across multiple mobile browsers—a trait common to all mobile web mediums.
- The mobile web has been gradually increasing in usage over the years in most major markets, but the limited experience offered little incentive to the user.
- Many compare the mobile web to a 10-year-old version of the Web: slow, expensive to use, and not much to look at.
- As better mobile browsers started being introduced to device platforms like the iPhone and Android, the quality of mobile websites began to improve dramatically, and with it, usage improved.

Pros

- The pros of mobile websites are:
- They are easy to create, maintain, and publish.
- They can use all the same tools and techniques you might already use for desktop sites.
- Nearly all mobile devices can view mobile websites.

Cons

- The cons of mobile websites are:
- They can be difficult to support across multiple devices.
- They offer users a limited experience.
- Most mobile websites are simply desktop content reformatted for mobile devices.
- They can load pages slowly, due to network latency.

3.MOBILE WEB WIDGETS

- Largely in response to the poor experience provided by the mobile web over the years, there has been a growing movement to establish mobile widget frameworks and platforms.
- For years the mobile web user experience was severely underutilized and failed to gain traction in the market, so several operators, device makers, and publishers began creating widget platforms (Figure) to counter the mobile web's weaknesses.
- Opera Widgets, Nokia Web Run Time (WRT), Yahoo! Blueprint, and Adobe Flash Lite are all examples of widget platforms that work on a number of mobile handsets.



Fig: An example mobile web widget

Pros

- The pros of mobile web widgets are:
- They are easy to create, using basic HTML, CSS, and JavaScript knowledge.
- They can be simple to deploy across multiple handsets.
- They offer an improved user experience and a richer design, tapping into device features and offline use.

Cons

- The cons of mobile web widgets are:
- They typically require a compatible widget platform to be installed on the device.
- They cannot run in any mobile web browser.
- They require learning additional proprietary, non-web-standard techniques.

4.MOBILE WEB APPLICATIONS

- Mobile web applications are mobile applications that do not need to be installed or compiled on the target device.
- Using XHTML, CSS, and JavaScript, they are able to provide an application-like experience to the end user while running in any mobile web browser.
- Web applications allow users to interact with content in real time, where a click or touch performs an action within the current view.
- The history of how mobile web applications came to be so commonplace.
- The explosion of Web 2.0, web applications like Face book, Flickr, and Google Reader hit desktop browsers, and there was discussion of how to bring those same web applications to mobile devices.
- With the introduction of the first iPhone, Using WebKit, the iPhone could render web applications not optimized for mobile devices as perfectly usable, including DHTML- and Ajax-powered content.
- Developers quickly got on board, creating mobile web applications optimized mostly for the iPhone (Figure).

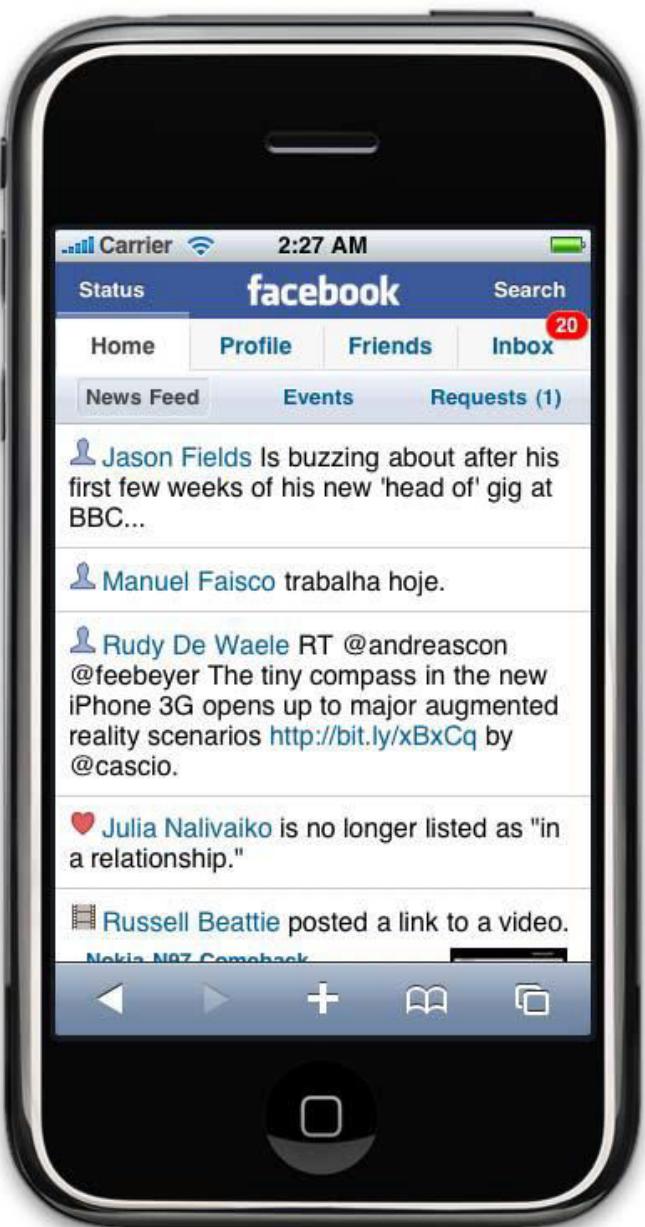


Fig: The Facebook mobile web app

Pros

- The pros of mobile web applications are:
- They are easy to create, using basic HTML, CSS, and JavaScript knowledge.
- They are simple to deploy across multiple handsets.
- They offer a better user experience and a rich design, tapping into device features and offline use.
- Content is accessible on any mobile web browser.

• Cons

- The cons of mobile web applications are:
- The optimal experience might not be available on all handsets.
- They can be challenging (but not impossible) to support across multiple devices.
- They don't always support native application features, like offline mode, location lookup, file system access, camera, and so on.

5.NATIVE APPLICATIONS

- These applications actually should be called “platform applications,” as they have to be developed and compiled for each mobile platform.
- These native or platform applications are built specifically for devices that run the platform in question. The most common of all platforms is Java ME (formerly J2ME).
- Although many smartphones are also powered by Java, an operating system layer and APIs added to allow developers to more easily offload complex tasks to the API instead of writing methods from scratch.
- In addition to Java, other smartphone programming languages include versions of C, C++, and Objective-C (Figure).
- Creating a platform application means deciding which devices to target, having a means of testing and certification, and a method to distribute the application to users.
- The advantage is that a mobile application can be developed faster, will work on more devices, require less testing, and be updated more transparently than a native application, which requires third-party certification and publishing in order to get on users’ devices.



Fig: A native application in the iPhone

Pros

- The pros of native applications include:
- They offer a best-in-class user experience, offering a rich design and tapping into device features and offline use.
- They are relatively simple to develop for a single platform.
- You can charge for applications.

Cons

- The cons of native applications include:
- They cannot be easily ported to other mobile platforms.
- Developing, testing, and supporting multiple device platforms is incredibly costly.
- They require certification and distribution from a third party that you have no control over.
- They require you to share revenue with the one or more third parties.

6.GAMES

- The final mobile medium is games, the most popular of all media available to mobile devices.
- Technically games are really just native applications that use the similar platform SDKs to create immersive experiences (Figure).
- Although can do many things with a powerful mobile web browser, creating an immersive gaming experience is not one of them—at least not yet.
- Adobe's Flash and the SVG (scalable vector graphics) standard are the done on mobile devices, the performance of the device in dealing with vector graphics.



Fig: An example game for the iPhone

Pros

- The pros of game applications are:
- They provide a simple and easy way to create an immersive experience.
- They can be ported to multiple devices relatively easily.

Cons

- The cons of game applications are:
- They can be costly to develop as an original game title.
- They cannot easily be ported to the mobile web.

APPLICATION CONTEXT

- Once your application medium is decided upon, it is time to look at the application context, or the appropriate type of application to present to the user in order for the user to process and understand the information presented and complete her goals.
- Where the application medium refers mostly to the technical approach of creating an application, the application context deals with the user experience.
- Applications can be presented in a variety of ways, ranging from a simple task-based utility to an experience meant to consume the user's focus and attention.

- **Utility Context**
- **Locale Context**
- **Informative Applications**
- **Productivity Application Context**
- **Immersive Full-Screen Applications**

Utility Context

- The most basic application context is the utility, or a simple user experience metaphor that is meant to address short, task-based scenarios. Information is meant to be presented in a minimal fashion, often using the least amount of user input as possible.
- An example of a utility might be a calculator, weather forecast, unit conversion, stocks, world clock, and so on.
- Use utilities for short, simple tasks, at-a-glance information, when there is limited content to display, and when combined with an immersive context to create dual-mode applications.



Fig: An example utility application

Locale Context

- The locale context is a newer application type that is still being defined by the mobile community, but we are certainly seeing some clear patterns of how to create locale applications (Figure).
- As more location information is being published online, and more devices add GPS to pinpoint the user's location, locale is becoming an excellent data point to pivot information around.
- Locale applications almost always have at least one thing in common: a map on which they plot the requested data points visually.
- At the very least, they list items in order of distance, with the nearest item first and the farthest last.
- The user's goal is to find information relative to his present location, and content should always be designed with this in mind.



Informative Applications

- The informative application is an application context in which the one and only goal is to provide information, like a news site, an online directory, a marketing site, or even a mobile commerce site, where the key task of the user is to read and understand and it is not necessary to interact.
- In the case of a mobile news site, provide the user with the option to mark a page or story to be read later.
- With a marketing site, allow users to enter the shortest possible contact information, like their phone number or email. And with a mobile commerce site, allow users to save items to a wishlist to review and purchase later.

Productivity Application Context

- The productivity application context is used for content and services that are heavily task-based and meant to increase the users' sense of efficiency.
- With these types of applications, we can assume that the users are more committed to accomplishing a particular goal, like managing content such as messages, contacts, or media, but we should still assume that they are doing so during idle periods
- Use the productivity application context for information-heavy applications where the user will need to manage content from a mobile device and for heavily structured, hierarchy-based tasks.

Immersive Full-Screen Applications

- The final application context is an immersive full-screen application, like a game, a media player, or possibly even a single-screen utility.
- The most common use of the immersive context is obviously with a game, for which you want the user to focus on how to play the game. But this context can also be used with other contexts, presenting a full-screen view of content when the device orientation changes in many higher-end devices.



Fig: An example of an immersive application

DAY 5

PREVIOUS TOPICS:

MOBILE APPLICATION MEDIUM TYPES:

1. SMS
2. MOBILE WEBSITES
3. MOBILE WEB WIDGETS
4. MOBILE WEB APPLICATIONS
5. NATIVE APPLICATIONS
6. GAMES

- ❖ Today's Topics to Discuss:
- ❖ **Mobile Information Architecture:**
- ❖ Mobile Information Architecture

MOBILE INFORMATION ARCHITECTURE

WHAT IS INFORMATION ARCHITECTURE?

- The structural design of shared information environments
- The combination of organizations, labelling, search, and navigation systems within websites and intranets
- The art and science of shaping information products and experiences to support usability and find ability
- An emerging discipline and community of practice focused on bringing principles of design and architecture to the digital landscape

Information architecture

The organization of data within an informational space. In other words, how the user will get to information or perform tasks within a website or application.

Interaction design

The design of how the user can participate with the information present, either in a direct or indirect way, meaning how the user will interact with the website or application to create a more meaningful experience and accomplish her goals.

Information design

- The visual layout of information or how the user will assess meaning and direction given the information presented to him.

Navigation design

- The words used to describe information spaces; the labels or triggers used to tell the users what something is and to establish the expectation of what they will find.

Interface design

- The design of the visual paradigms used to create action or understanding.

The role of information architecture is played by a variety of people, from product managers to designers and even developers.

To make things more confusing, information architecture can be called many different things throughout the design and development process.

Words like intuitive, simple, findable, usable, or the executive favorite easy to- use—all describe the role that information architects play in creating digital experiences.

MOBILE INFORMATION ARCHITECTURE

Information architecture has become a common discipline in the web industry, unfortunately, the mobile industry like software has only a handful of specialized mobile information architects.

Although mobile information architecture is hardly a discipline in its own right, it certainly ought to be.

This is not because it is so dissimilar from its desktop cousin, but because of context, added technical constraints, and needing to display on a smaller screen as much information as we would on a desktop.

- **Keeping It Simple**

- When thinking about your mobile information architecture, you want to keep it as simple as possible.

- **Support your defined goal**

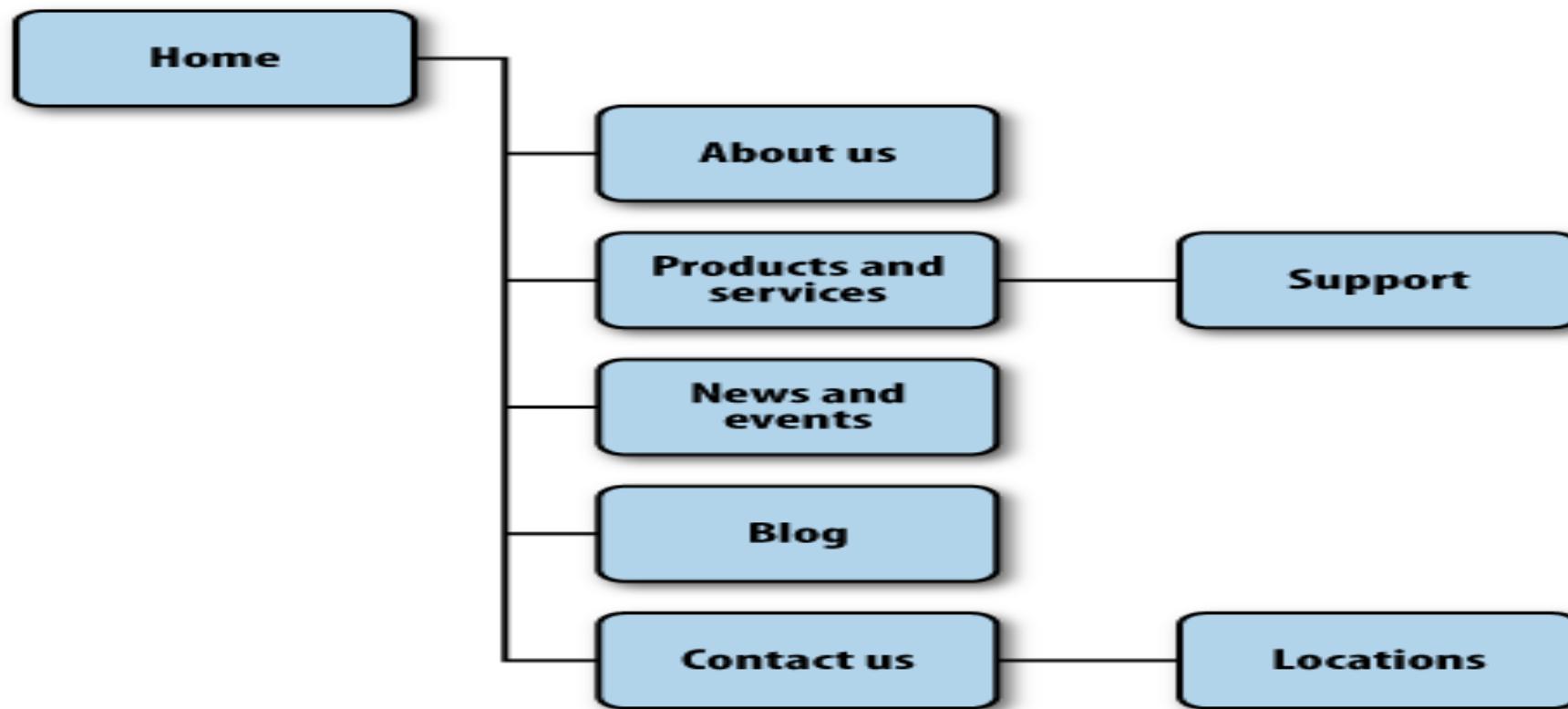
- If something doesn't support the defined goals, lose it. Go back to your user goals and needs, and identify the tasks that map to them. Find those needs and fill them.

- **Clear, simple labels**

- Good trigger labels, the words we use to describe each link or action, are crucial in Mobile. Words like —products|| or —services|| aren't good trigger labels. Users have a much higher threshold of pain when clicking about on a desktop site or application, hunting and pecking for tasty morsels.

- **Site Maps**

- Relationship of content to other content and provide a map for how the user will travel through the informational space



An example mobile site map

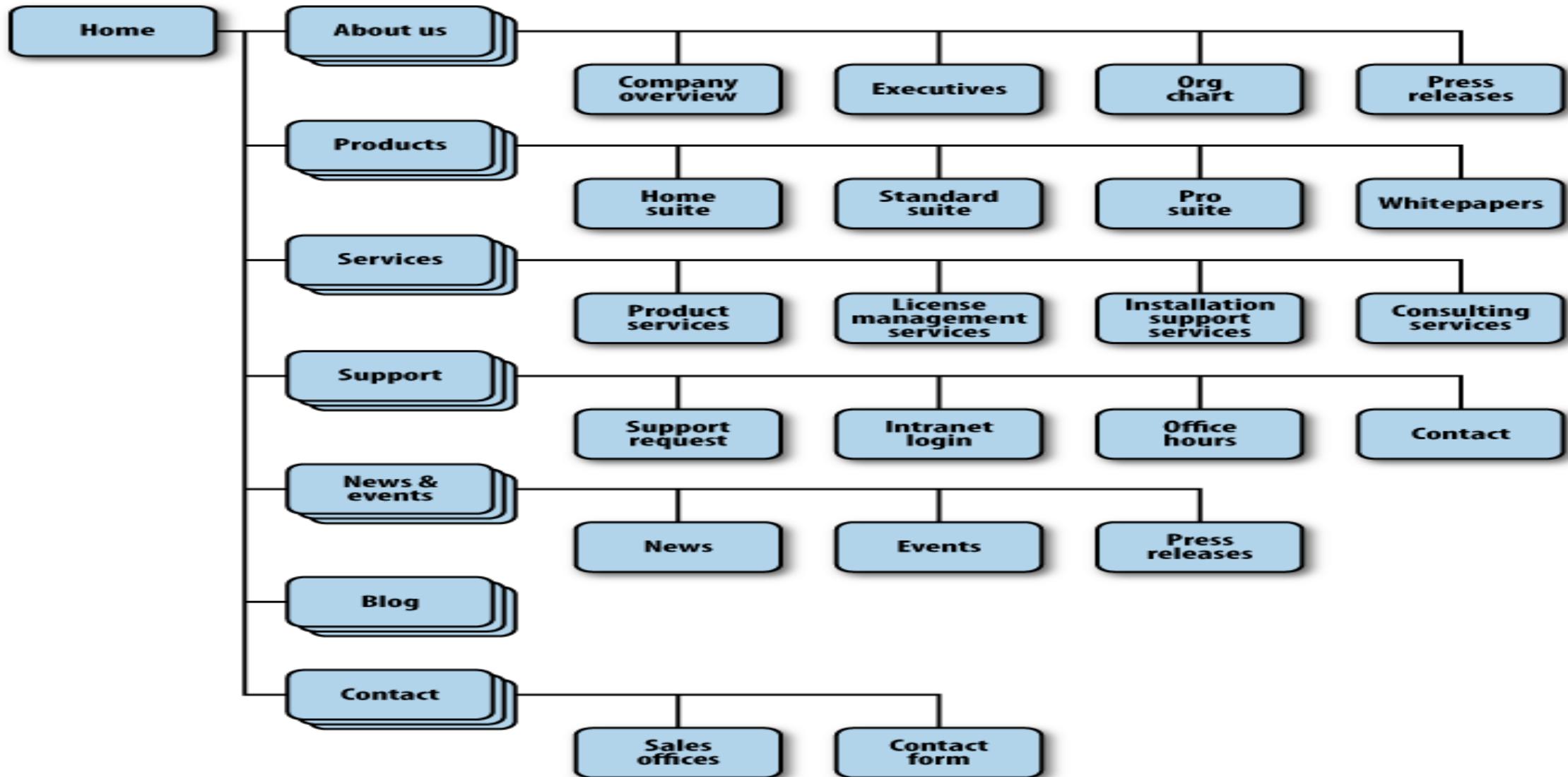


Fig: An example of a bad mobile information architecture that was designed with desktop users in mind rather than mobile users

In the mobile context, tasks are short and users have limited time to perform them.

And with mobile websites, we can't assume that the users have access to a reliable broadband connection that allows them to quickly go back to the previous page.

In addition, the users more often than not have to pay for each page view in data charges.

So not only do they pay cash for viewing the wrong page by mistake, they pay to again download the page they started from: we can't assume that pages will be cached properly.

Confirm the path by teasing content

Information-heavy sites and applications often employ nested or drill-down architectures, forcing the user to select category after category to get to their target.

To reduce risking the user's time and money, we want to make sure we present enough information for the user to wade through our information architecture successfully.

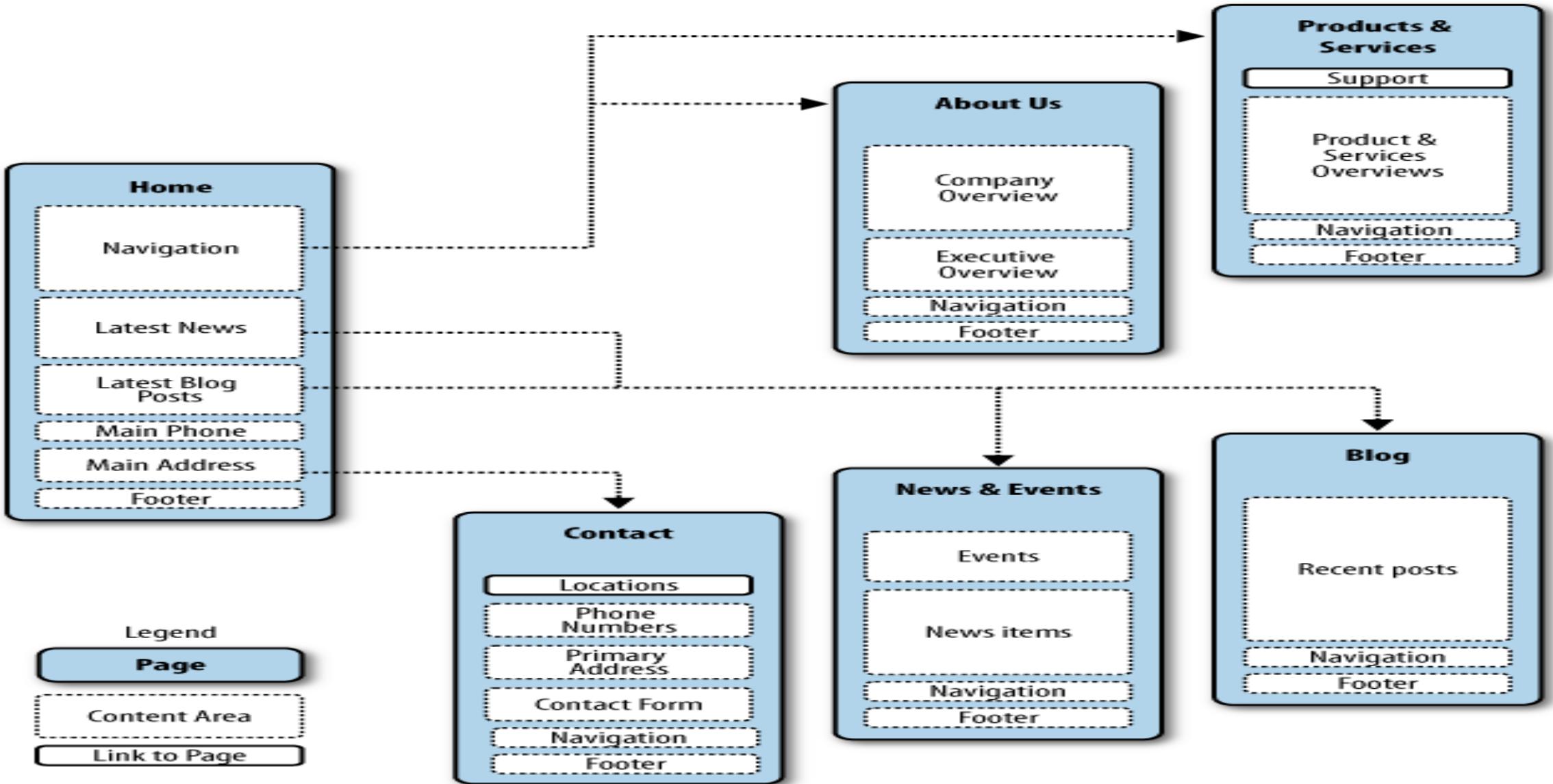
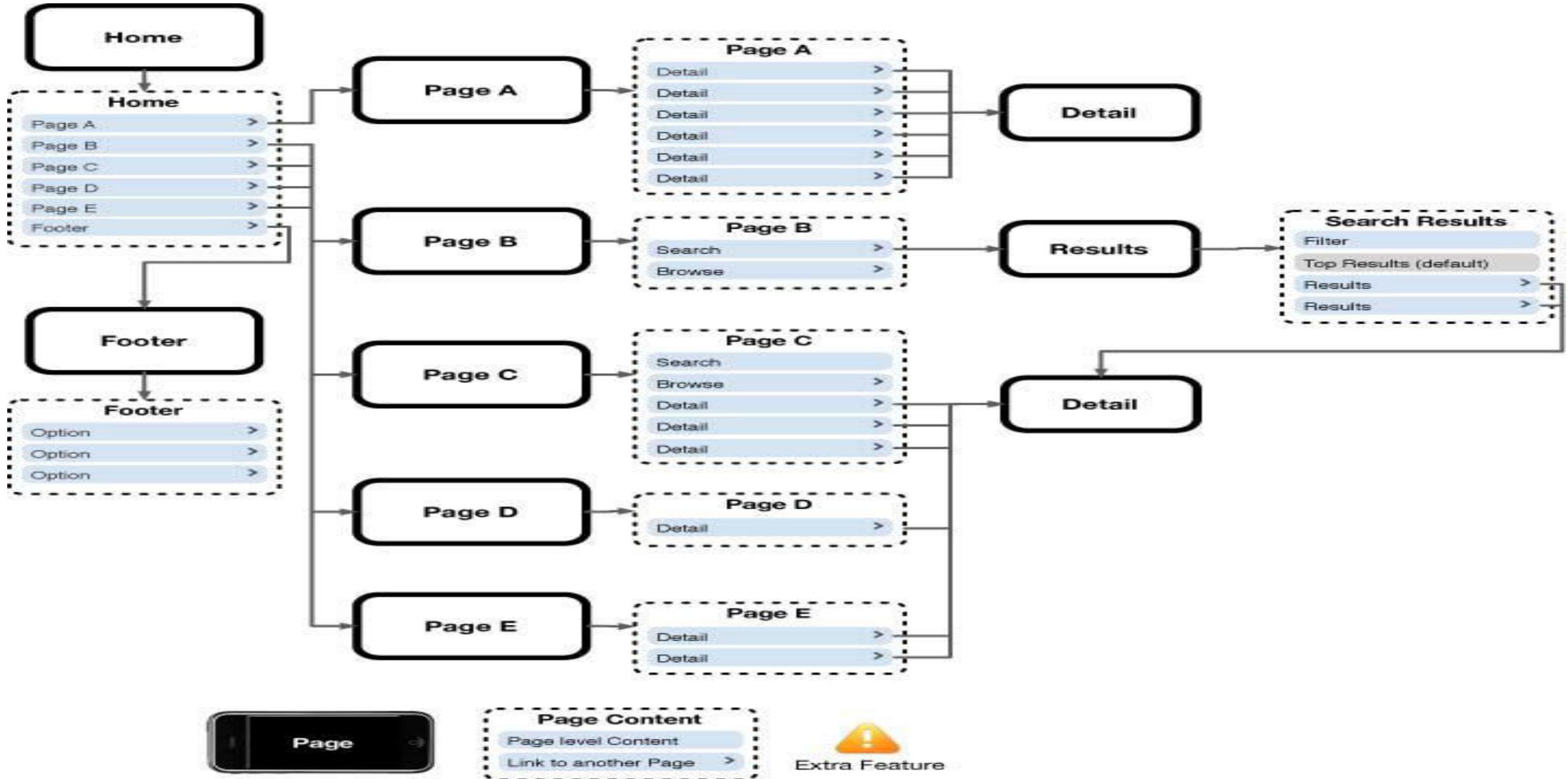


Fig: Teasing content to confirm the user's expectations of the content within

Clickstreams

Clickstream is a term used for showing the behaviour on websites, displaying the order in which users travel through a site's information architecture, usually based on data gathered from server logs.

Clickstreams are usually historical, used to see the flaws in your information architecture, typically using heat-mapping or simple percentages to show where your users are going.



- Remember, in mobile, our job is to keep it as simple as possible. We need to have an unwavering focus on defining an excellent user experience first and foremost. Anything that distracts us from that goal is just a distraction.



Fig: An example of an iPhone web application wireframe, intended to be low fidelity to prevent confusion of visual design concepts with information design concepts

- **Wireframes**

- The next information architecture tool at our disposal is wireframes. Wireframes are a way to lay out information on the page, also referred to as information design.
- Site maps show how our content is organized in our informational space; wireframes show how the user will directly interact with it.
- Wireframes are like the peanut butter to the site map jelly in our information architecture sandwich. It's the stuff that sticks.
- Wireframes like the one in Figure serve to make our information space tangible and useful.

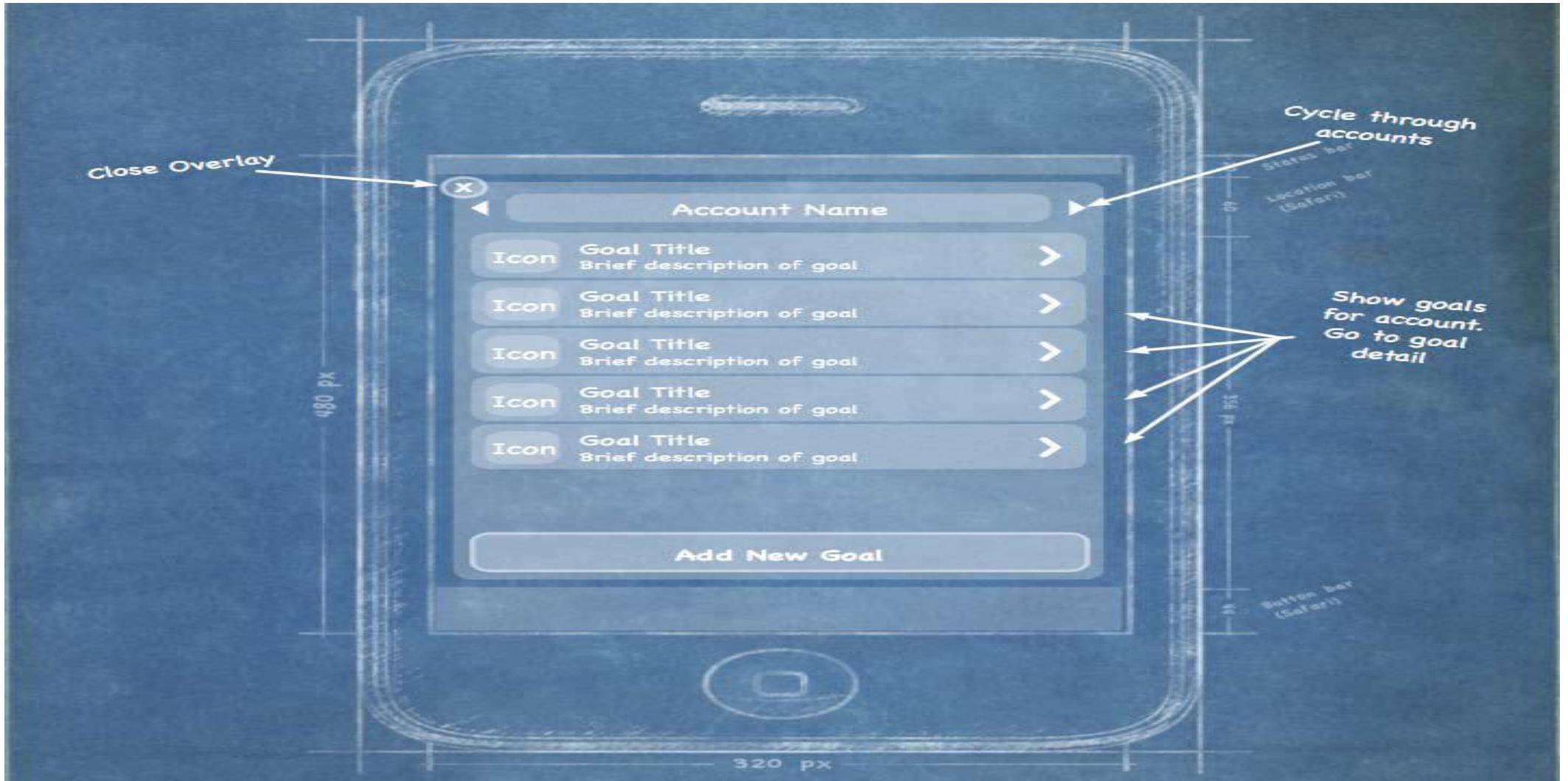


Fig: Using annotations to indicate the desired interactions of the site or application

- **Prototyping**

- Prototypes might sound like a scary (or costly) step in the process. Some view them as redundant or too time-consuming, preferring to jump in and start coding things. But as with wireframes, found that each product we've built out some sort of prototype has saved both time and money.

- **Paper prototypes**

- The most basic level we have is paper prototyping: taking our printed-out wireframes or even drawings of our interface, like the one shown in Figure, and putting them in front of people.

Context prototype

The next step is creating a context prototype (Figure). Take a higher-end device that enables you to load full-screen images on it.

Take your wireframes or sketches and load them onto the device, sized to fill the device screen. Leave the office. Go for a walk down to your nearest café. Or get on a bus or a train.

Pay particular attention to what you are thinking and your physical behavior while you are using your interface and then write it down.



Fig: An example of a context prototype, or taking images loaded onto a device and testing them in the mobile context

- **HTML prototypes**

- The third step is creating a lightweight, semi functional static prototype using XHTML, CSS, and JavaScript, if available.
- This is a prototype that you can actually load onto a device and produce the nearest experience to the final product, but with static dummy content and data (Figure).
- It takes a little extra time, but it is worth the effort.
- With a static XHTML prototype, you use all the device metaphors of navigation, you see how much content will really be displayed on screen (it is always less than you expect), and you have to deal with slow load times and network latency. In short, you will feel the same pains your user will go through.



Fig: An XHTML prototype that you can actually interact with on real mobile devices.

THE DESIGN MYTH

A little secret about interactive design is that people don't respond to the visual aesthetic as much as you might think.

What colours you use, whether you use square or rounded corners, or, gradients or flat backgrounds, helps build first impressions, but it doesn't do too much to improve the user's experience.



Figure: Comparing visual design to information design of the iPhone application Tweetie.

QUERIES

Previous Topic.....

- ❖ **Mobile Information Architecture:**
 - What is Information Architecture
 - Mobile Information Architecture

Today Discuss about.....

THE ELEMENTS OF MOBILE DESIGN

- Context**
- Message**
- Look and Feel**
- Layout**
- Color**
- Typography**

MOBILE DESIGN TOOLS

MOBILE DESIGN:

THE ELEMENTS OF MOBILE DESIGN

- Good design requires three abilities:
- The first is a natural gift for being able to see visually how something should look that produces a desired emotion with the target audience.
- The second is the ability to manifest that vision into something for others to see, use, or participate in.
- The third knows how to utilize the medium to achieve your design goals.
- Six elements of mobile design that you need to consider.

1. Context:

- As the designer, it is your job to make sure that the user can figure out how to address context using your app. Make sure you do your homework to answer the following questions:
- Who are the users? What do you know about them? What type of behaviour can you assume or predict about the users?
- What is happening? What are the circumstances in which the users will best absorb the content you intend to present?
- Why will they use your app? What value will they gain from your content or services in their present situation?
- How are they using their mobile device? Is it held in their hand or in their pocket?
- How are they holding it? Open or closed? Portrait or landscape?
- The answers to these questions will greatly affect the course of your design. Treat these questions as a checklist to your design from start to finish.

2. Message:

- Message is the overall mental impression you create explicitly through visual design. If you take a step back, and look at a design from a distance, what is your impression?
- Branding shouldn't be confused with messaging. Branding is the impression your company name and logo gives—essentially, your reputation. Branding serves to reinforce the message with authority, not deliver it.

Yahoo!

- Yahoo! sort of delivers a message. This app provides a clean interface, putting a focus on search and location, using color to separate it from the news content. But I'm not exactly sure what it is saying. Words you might use to describe the message are crisp, clean, and sharp.

ESPN

- The ESPN site clearly is missing a message.
- It is heavily text-based, trying to put a lot of content above the fold, but doesn't exactly deliver a message of any kind. If you took out the ESPN logo, you likely would have indifferent expectations of this site; it could be about anything, as the design doesn't help set expectations for the user in any way.
- Words you might use to describe the message: bold, cluttered, and content-heavy.

Disney

- Disney creates a message with its design. It gives you a lot to look at—probably too much—but it clearly tries to say that the company is about characters for a younger audience. Words you might use to describe the message: bold, busy, and disorienting.

Wikipedia

- The Wikipedia design clearly establishes a message. With a prominent search and text-heavy layout featuring an article, you know what you are getting with this design. Words you might use to describe the message: clean, minimal, and text-heavy.

Amazon

- Amazon sort of creates a message. Although there are some wasted opportunities above the fold with the odd ad placement, you can see that it is mostly about products (which is improved even more if you scroll down). Words you might use to describe the message: minimal but messy, product-heavy, and disorienting.

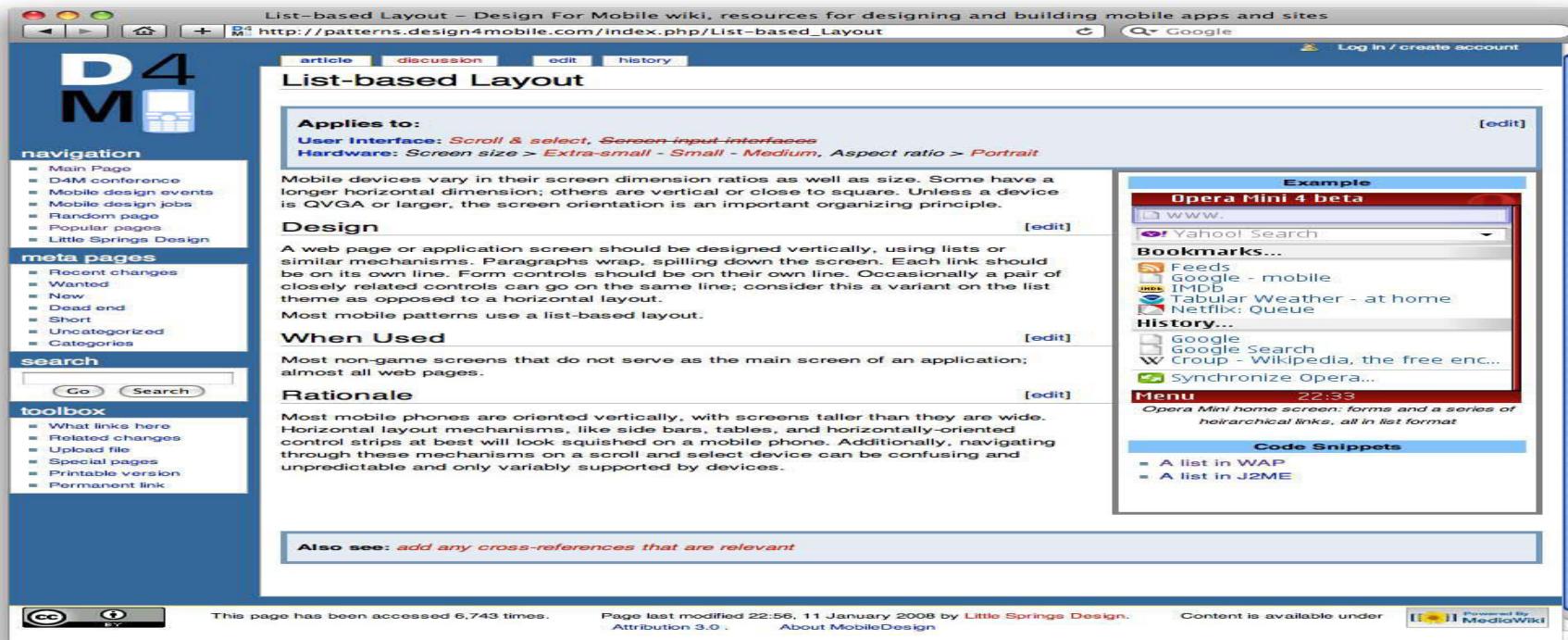
3. Look and Feel :

- Look and feel is used to describe appearance, as in —I want a clean look and feel|| or —I want a usable look and feel.|| The problem is: as a mobile designer, what does it mean? And how is that different than messaging?
- look and feel in a literal sense, as something real and tactile that the users can —look|| at, then —feel||—something they can touch or interact with.
- Look and feel is used to evoke action—how the user will use an interface.
- Establishing a look and feel usually comes from wherever design inspiration comes from. However, your personal inspiration can be a hard thing to justify. Therefore we have —design patterns,|| or documented solutions to design problems, sometimes referred to as style guides.

4. Layout

- Layout is an important design element, because it is how the user will visually process the page, but the structural and visual components of layout often get merged together, creating confusion and making your design more difficult to produce.
- The first-time layout should rear its head is during information architecture. In fact, to make about 90 percent of my layout decisions during the information architecture period.
- Design is just too subjective of an issue. If you are creating a design for anyone but yourself, chances are good that there will be multiple loosely-based-on-experience opinions that will be offered and debated.
- Where the design opinions of the CEO or Chief Marketing Officer (CMO) might influence a design direction more than, say, the Creative Director or Design Director.
- By defining design elements like layout prior to actually applying the look and feel, you can separate the discussion.

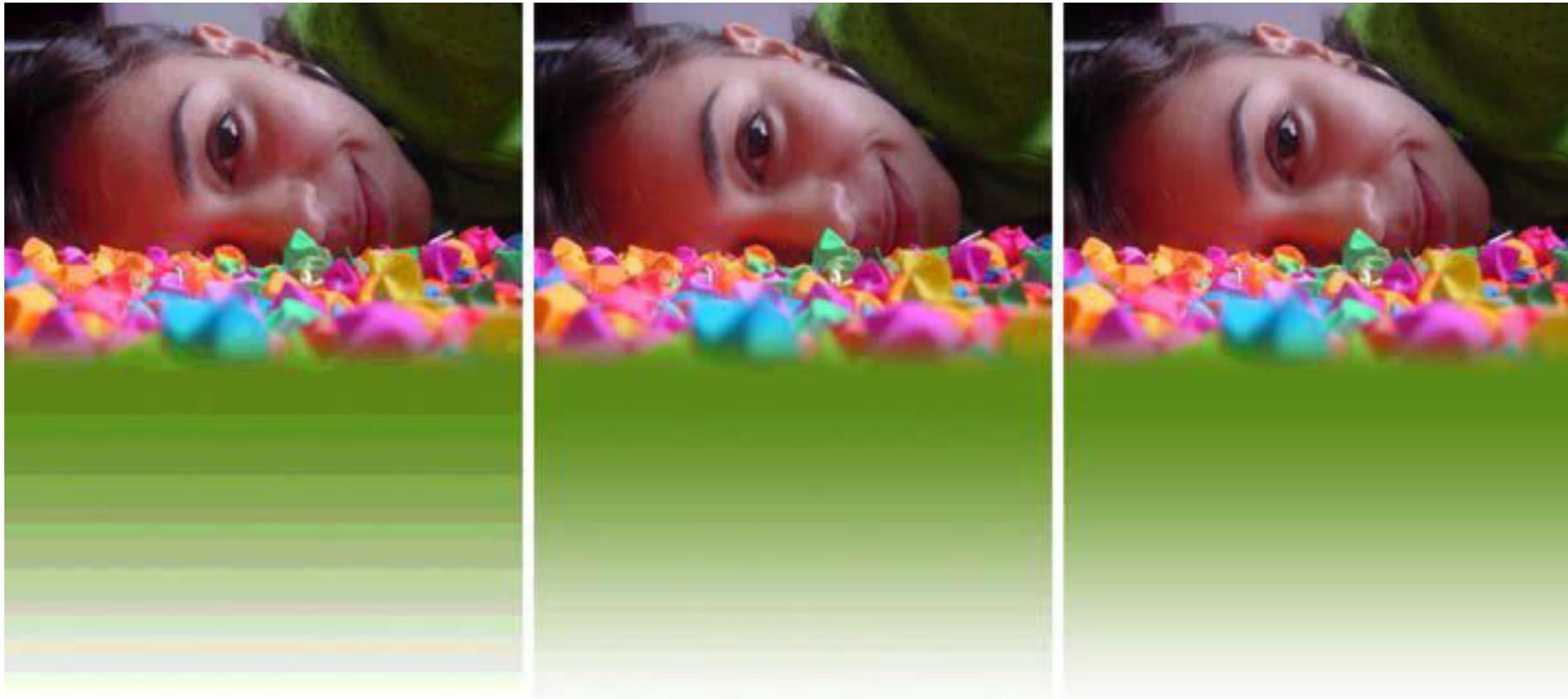
- First, there is confirmation that the navigation and layout are correct. Second, there is a question about the —look and feel.|| Because designers hear —Do it again,|| they typically redo the layout, even though it was actually fine.
- For example, when I show my mobile design layouts as wireframes during the information architecture phase, I intentionally present them on blueprint paper, using handwriting fonts for my annotations (Figure below). It also helps to say that this is not a design, it is a layout, so please give me feedback on the layout.



5. Color :

The fifth design element, color, is hard to talk about in a black-and-white book. Maybe it is fitting, because it wasn't that long ago that mobile screens were available only in black and white well, technically, it was black on a green screen).

- The most common obstacle you encounter when dealing with color is mobile screens, which come in a number of different color or bit depths, meaning the number of bits (binary digits) used to represent the color of a single pixel in a bitmapped image.
- When complex designs are displayed on different mobile devices, the limited color depth on one device can cause banding, or unwanted posterization in the image.



12-bit color

16-bit color

24-bit color

Example produced by Dave Patteridge. Photo Credit: szondyek_ma

Fig: An example of different levels of posterization that can occur across multiple device color depths

Bit depth	Supported colors	Description	Example devices
12-bit	4,096 colors	Used with older phones; dithering artifacts in photos can easily be seen.	Nokia 6800
16-bit	65,536 colors	Also known as HighColor; very common in today's mobile devices. Can cause some banding and dithering artifacts in some designs.	HTC G1, BlackBerry Bold 9000, Nokia 6620

Color	Represents
Red	Passion, strength, energy, fire, sex, love, romance, excitement, speed, heat, arrogance, ambition, leadership, masculinity, power, danger, gaudiness, blood, war, anger, revolution, radicalism, aggression, respect, martyrs, conservatism (U.S. politics), Liberalism (Canadian politics), wealth (China), and marriage (India)
Orange	Energy, enthusiasm, balance, happiness, heat, fire, flamboyance, playfulness, aggression, arrogance, gaudiness, overemotion, warning, danger, autumn, desire
Pink	Spring, gratitude, appreciation, admiration, sympathy, socialism, femininity, health, love, romance, marriage, joy, flirtatiousness, innocence and child-like qualities
Brown	Calm, boldness, depth, nature, richness, rustic things, stability, tradition, anachronism, boorishness, dirt, dullness, heaviness, poverty, roughness, earth

Colour palettes

- Defining color palettes can be useful for maintaining a consistent use of color in your mobile design. Color palettes typically consist of a predefined number of colors to use throughout the design.
- Selecting what colors to use varies from designer to designer, each having different techniques and strategies for deciding on the colors.
- Three basic ways to define a color palette:

Sequential

- In this case, there are primary, secondary, and tertiary colors. Often the primary color is reserved as the —brand|| color or the color that most closely resembles the brand's meaning. The secondary and tertiary colors are often complementary colors that select using a color wheel.

Adaptive

- An adaptive palette is one in which you leverage the most common colors present in a supporting graphic or image. When creating a design that is meant to look native on the device, use an adaptive palette to make sure that my colors are consistent with the target mobile platform.

Typography

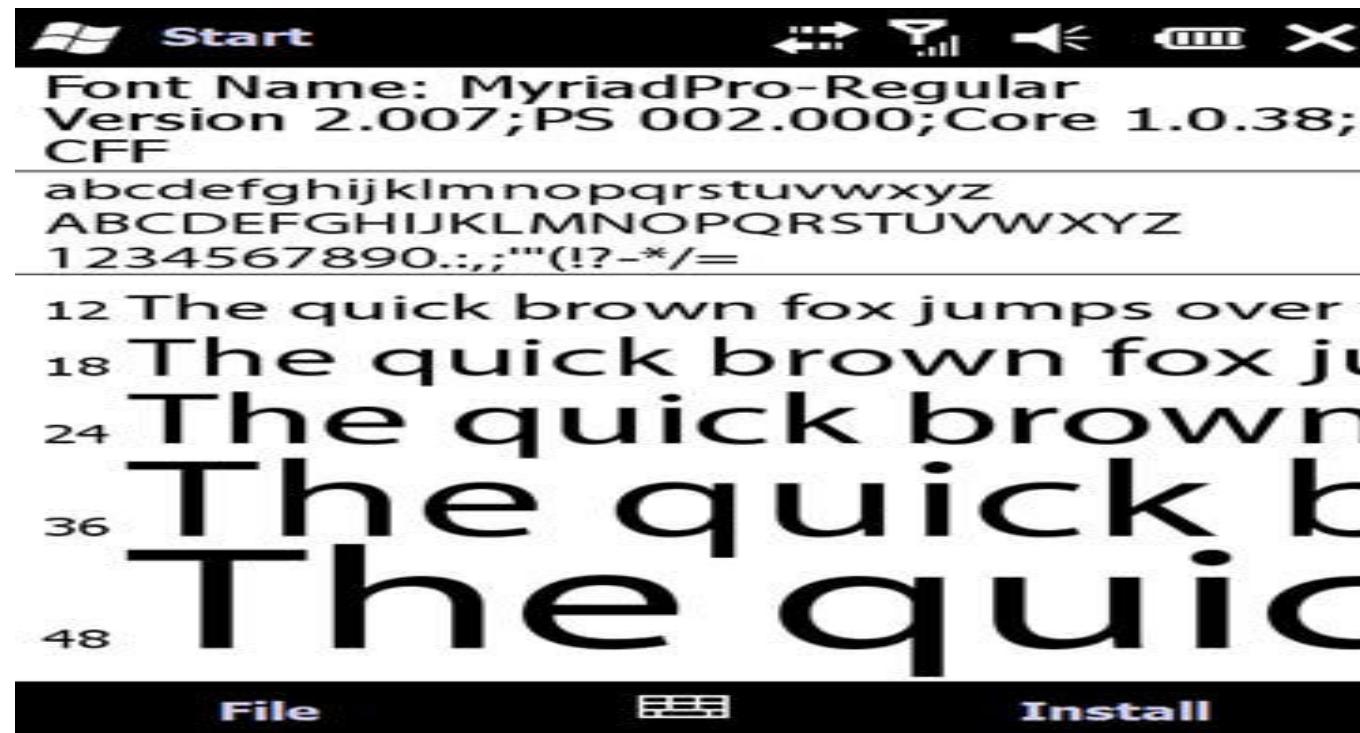
- The sixth element of mobile design is typography, which in the past would bring to mind the famous statement by Henry Ford:
- “Any customer can have a car painted any color that he wants so long as it is black”.
- Traditionally in mobile design, you had only one typeface that you could use (Figure), and that was the device font. The only control over the presentation was the size.

The quick brown
fox jumped over
the lazy dog..

- As devices improved, so did their fonts. Higher-resolution screens allowed for a more robust catalog of fonts than just the device font.

Subpixels and pixel density

- There seem to be two basic approaches to how type is rendered on mobile screens:
- Using subpixel-based screens or having a greater pixel density or pixels per inch (PPI). A subpixel is the division of each pixel into a red, green, and blue (or RGB) unit at a microscopic level, enabling a greater level of antialiasing for each font character



- The second approach is to use a great pixel density, or pixels per inch. We often refer to screens by either their actual physical dimensions or their pixel dimensions, or resolution (“The resolution of my laptop is 1440×900 pixels”).
- The pixel density is determined by dividing the width of the display area in pixels by the width of the display area in inches. So the pixel density for my 15.4-inch laptop would be 110 PPI. In comparison, a 1080p HD television has a PPI of 52.

Mobile device	Diagonal	Pixels	PPI
Nokia N95	2.6"	240×320	153
Apple iPhone 3G	3.5"	320×480	163
Amazon Kindle	6.0"	600×800	167
HTC Dream	3.2"	320×480	181
Sony Ericsson W880i	1.8"	240×320	222
Nokia N80	2.1"	352×416	256

Type options

- Fortunately, today's mobile devices have a few more options than a single typeface, but the options are still fairly limited.
- Coming from web design, where we have a dozen or so type options, the limited choices available in mobile design won't come as a big surprise.
- Essentially, you have a few variations of serif, sans-serif, and monospace fonts, and depending on the platform, maybe a few custom fonts.

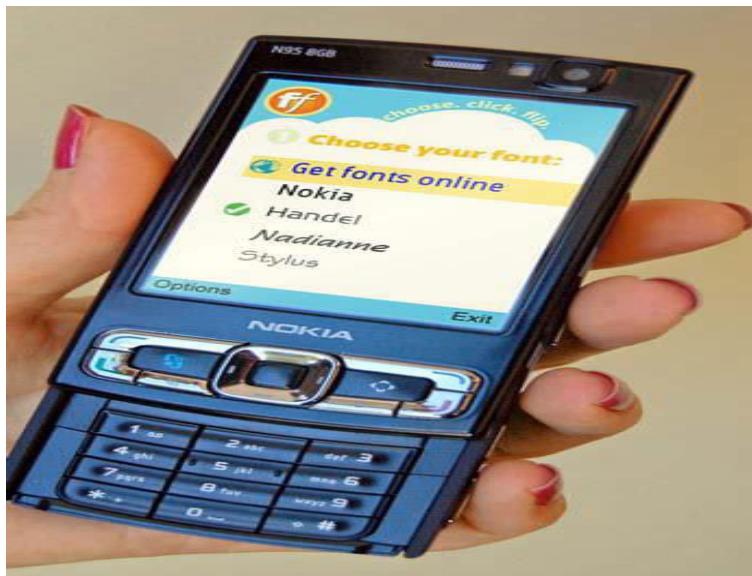


Fig: Options in typography increase as the devices become more sophisticated

Font replacement

- The ability to use typefaces that are not already loaded on the device varies from model to model and your chosen platform.
- Some mobile web browsers support various forms of font replacement; the two most common are sIFR and Cufon.
- sIFR uses Flash to replace HTML text with a Flash representation of the text, but the device of course has to support Flash.
- Cufon uses JavaScript and the canvas element draws the glyphs in the browser, but the device of course needs to support both JavaScript and the canvas element.

Readability

- The most important role of typography in mobile design is to provide the user with excellent readability, or the ability to clearly follow lines of text with the eye and not lose one's place or become disoriented, as shown in Figure.

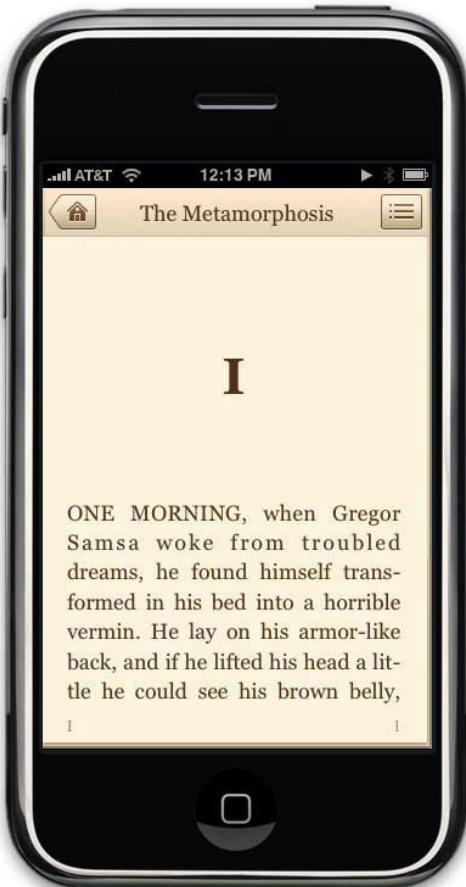


Fig: Classics, an iPhone application designed with readability and typography in mind

This can be done by following these six simple rules:

1. *Use a high-contrast typeface*
2. *Use the right typeface*
3. *Provide decent leading (rhymes with “heading”) or line spacing*
4. *Leave space on the right and left of each line; don’t crowd the screen*
5. *Generously utilize headings*
6. *Use short paragraphs*

MOBILE DESIGN TOOLS

- Mobile design requires understanding the design elements and specific tools. The closest thing to a common design tool is Adobe Photoshop, though each framework has a different method of implementing the design into the application.
- Some frameworks provide a complete interface toolkit, allowing designers or developers to simply piece together the interface, while others leave it to the designer to define from scratch.

Mobile framework	Design tool	Interface toolkits
Java ME	Photoshop, NetBeans	JavaFX, Capuchin
BREW	Photoshop, Flash	BREW UI Toolkit, uiOne, Flash
Flash Lite	Flash	Flash Lite
iPhone	Photoshop, Interface Builder	iPhone SDK

Designing for Different Screen Sizes

- Mobile devices come in all shapes and sizes. Choice is great for consumers, but bad for design.
- It can be incredibly difficult to create that best possible experience for a plethora of different screen sizes.
- The bad news is that there is no simple answer. How you design each screen of content depends on the scope of devices you look to support, your content, and what type of experience you are looking to provide.
- The good news is that the vast majority of mobile device screens share the same vertical or portrait orientation, even though they vary greatly in dimension as shown in Figure.

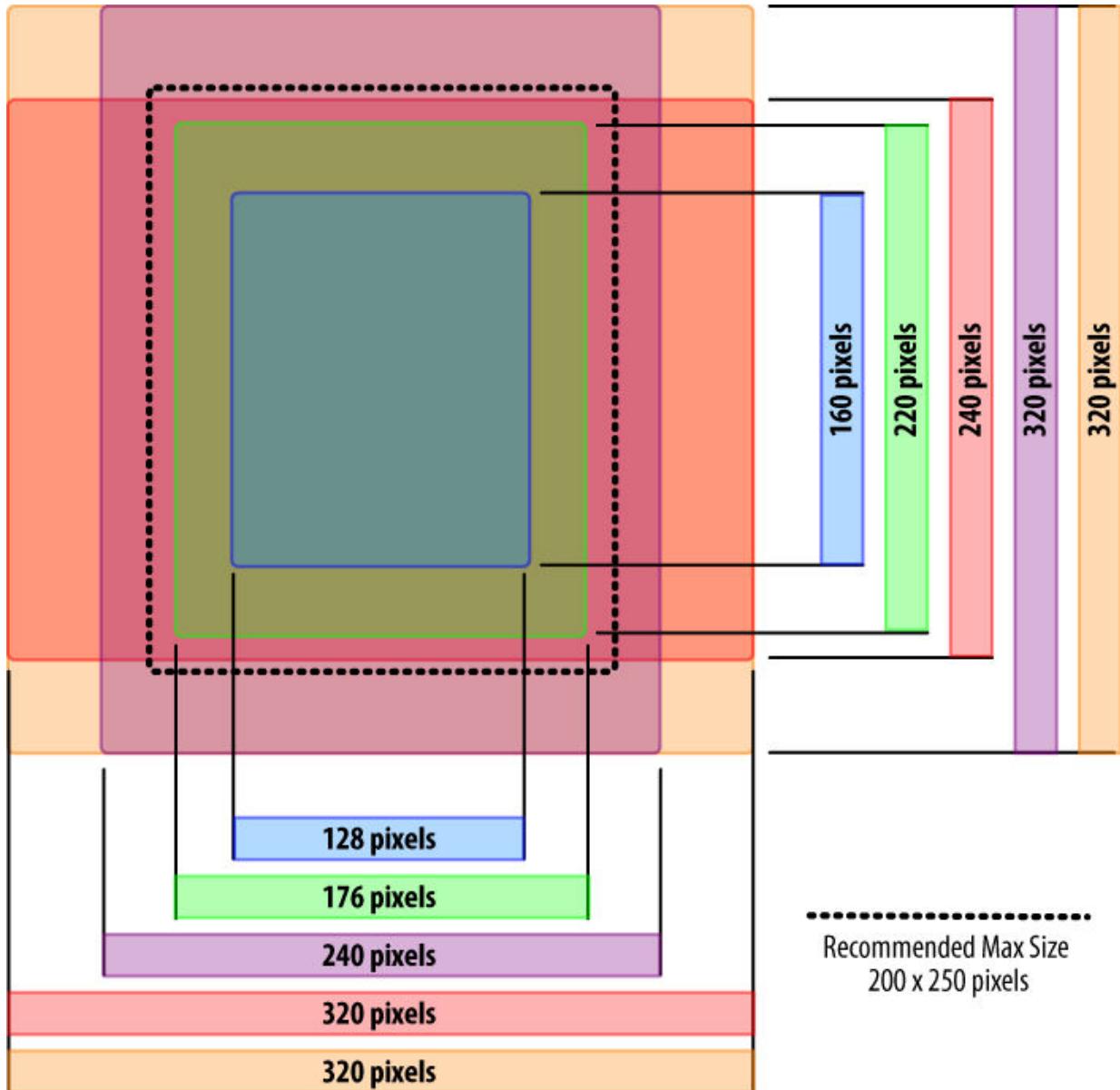


Figure: Comparing the various screen sizes



Your Queries Please!!!

Previous Topic.....

THE ELEMENTS OF MOBILE DESIGN

- Context
- Message
- Look and Feel
- Layout
- Color
- Typography

MOBILE DESIGN TOOLS

UNIT – III

Mobile Web Apps Vs Native Applications: Ubiquity Principle- When to Make a Native Application – When to Make a Mobile Web Application. **Mobile Web Development:** Web Standards- Designing for Multiple Mobile Browsers- Device Plans- Markup- CSS- JavaScript.

Today Discuss about.....

Mobile Web Apps Vs Native Applications:

- **The Ubiquity Principle**
- **When To Make a Native Application**
- **When To Make a Mobile Web Application**

MOBILE WEB APPS VS NATIVE APPLICATIONS:

THE UBIQUITY PRINCIPLE:

The mobile web is the only long-term commercially viable content platform for mobile devices.

The four key reasons to support this belief:

- 1. Fragmentation**
- 2. The Web**
- 3. Control**
- 4. Consumer Expectations**

Fragmentation

- First of all, we already know that mobile is a much larger playing field than desktop computing, but there is currently no economically feasible means to create native applications that can support the majority of the market.
- **Device fragmentation** refers to the wide variety of **devices** that are in use at any given time. This is important because websites need to ensure a good user experience on as many **devices** as possible. ... For example, there are thousands of **device** models running the **Android** operating system today.
- Mobile device fragmentation is a phenomenon that occurs when some mobile users are running older versions of an operating system, while other users are running newer versions.

The Web

- The overall technology market is going to the Web.
- It is a highly vetted consumer medium that offers many pros and few cons.
- It is the only medium for information, applications, and services that has gone the distance for the last 15 years.
- The **Web**, or World Wide **Web** (W3), is basically a system of **Internet** servers that support specially formatted documents. The documents are formatted in a markup language called HTML (HyperText Markup Language) that supports links to other documents, as well as graphics, audio, and video files.
- allows users to click through pages on the **Web** via links. The **Web uses** HTTP protocol to transmit data and share information. Browsers such as **Internet Explorer**, Google Chrome or Mozilla Firefox are used to access **Web** documents, or **Web** pages, which are connected via links

Control

- Mobile application distribution cannot and will likely never be under the control of the developer.
- In other words, mobile application vendors always have to rely on middlemen to get their products to market and take a slice of their profits.
- This has been the case since the beginning of downloadable mobile applications, when they were under the tight control of the operator.

- These days, we see that control shifting to the device and platform makers, making getting applications to market easier.
- Without control, this reason alone means that the funding of creating mobile applications will always remain a small, high-risk investment.

Consumer Expectations

- The challenge with native mobile applications is that the consumer may see an application that might look appealing to him, but if it isn't supported for his particular device.
- This is one of the reasons that operators usually require applications sold on their marketplace to support their top 10 to 15 devices.
- From the consumer's perspective, he spends good money on a device and wants content to support it.

- The lack of available content lowers the perceived value of the device.
- Consumers don't care what device or platform they have; they just want to participate in the same content and services that their friends are using.
- Because cross-platform support is so challenging, that is hardly ever the possible.
- Numerous visits occur when a consumer has purchased a brand-new device, but then visits drop off precipitately in just days.

Ubiquity in the Mobile Web

- The mobile web is the only platform that is available and works across all mobile devices, sharing the same set of standards and protocols with each other as well as the desktop web.
- The mobile web is also the only mobile distribution channel available to developers that they can control.

Ubiquity Principle: easier-to-produce quality content and services for the largest available market will always win.

- The key word is “quality,” which the mobile web hasn’t had a lot of over the years.

WHEN TO MAKE A NATIVE APPLICATION:

The following sections discuss some of the key features you may be considering that almost immediately point toward creating a native application.

Native apps are applications that are written in languages specific to the operating systems that they are developed for - primarily, this has been Java for [Android](#) and Objective-C for [iOS](#) for iphone.

If you want to provide the best user experience possible in terms of the look and feel of your application, then [native app development](#) may be a good option for you.

- 1. Performance**
- 2. Security**
- 3. Access to developers**
- 4. Scalability(capacity)**
- 5. Access to hardware features**
- 6. Stability**

Operators

- A native mobile application directory, either through the operator or through a device maker, always includes a means to collect payment for it.
- Since those days, device maker markets have made it much easier to get products to market, but are plagued(trouble) by most, if not all, of the same problems.
- The operator gives a SIM card to the customer who inserts it into the mobile device to gain access to the service.
- There are two types of mobile operators: a **mobile network operator** (MNO) which owns the underlying network and spectrum assets required to run the service.

Creating a Game

- If your goal is to create a mobile game—one of the biggest mobile content markets— then you need to create a native application.
- Games are resource-intensive and almost always require the use of a device or platform API.
- When users launch a game, they have some expectations of what it is going to look and act like.
- When making mobile games, you need to carefully consider which platforms to support.
- Luckily, there are a variety of game porting houses that can help get your game onto multiple platforms, but as you might imagine, it can be a costly endeavor in terms of money and time.

Using Specific Locations

- The next feature is location, or being able to detect the users' locations by GPS or cell tower triangulation for the purpose of presenting users with information based on their current location.
- Devices that run WebKit, like the iPhone or Android devices, as well as devices that run the Opera or Mozilla browsers, will all have the ability to detect user's locations.
- Adding location means sharing sensitive information with websites, which could actually be quite dangerous.

Using Cameras

- The camera is another device function that can come in handy in your applications.
- Traditionally, mobile MMS (Multimedia Messaging Service) is used to handle mobile photo interactions.
- In other words, you take a photo, send it via MMS to a shortcode, then a server somewhere does something with the photo and sends an alert to you when it is done.
- With access to the camera, native application developers can simplify the task to just taking a photo from within an application.
- The camera is useful in several types of mobile applications, from sharing snapshots or videos of friends to capturing important events as they happen and recording visual information, such as an important advertisement or sign found while out and about.

Using Accelerometers

- **Accelerometers** are **devices** that can measure acceleration (the rate of change in velocity), but in smartphones, they're able to detect changes in orientation and tell the screen to rotate.
- The most common use of an accelerometer is to detect when the device is physically rotated, adjusting the display from vertical to horizontal orientation (or vice versa).
- Using the accelerometer can be a benefit to mobile users, enabling them to interact with a device in a more natural way; being that the device is likely held in the hand, it can adjust content to suit its physical orientation, like rotating the screen, or detecting physical movement, and can therefore have limited prediction of the users' context.
- Every mobile interaction should pass the “transit test.” You should always assume that the user will interact with a mobile device the same way she would if she were sitting on a crowded bus or train.
- Ask yourself what the likelihood is that the user will shake the phone while standing in a packed subway or riding in a car.

Accessing the File systems

- Another reason you would want to create a native application is if you want to use the data stored on the device itself.
- This might be the user's address book, photos, an email message, or even data from another application.
- Such file system access is obviously a big security and privacy issue.
- An infected application could use your contacts and constant connection to spread a virus to multiple phones, something that occurred quite often before widespread adoption of mobile application certification.
- On the other hand, mobile devices are becoming highly personal, mobile computers that store an increasing amount of content and information about their owners and the owners' friends and business contacts.
- The idea of leveraging this information across applications is appealing.
- Though not without risk, using stored data is a powerful way to present contextually relevant information to the user.

Offline Users

- The final reason to make a native application is because you know the user is likely to be offline or out of range of a mobile network.
- For those of us who live in the city, that may seem like a rare occasion.
- Even for those who live in more rural areas, network dead spots are becoming increasingly rare. But going periods without a connection does of course happen frequently and your application should be designed to take this into account.
- A trail map application will likely be used in more remote areas, with less coverage.
- A mobile travel guide will probably be used in a foreign network, where roaming and international fees could occur.
- Each of these applications should have an offline mode in which the user can still perform the most common tasks of the application without the need of a wireless connection.

WHEN TO MAKE A MOBILE WEB APPLICATION:

- Native applications don't service the user better in any significant way; they only add cost to your project, decrease your distribution channels, plus cause you to lose the ability to incrementally improve your application, lose control and profit, and add to the device fragmentation problem.
- The most interesting case for mobile web apps is actually composed of the reasons stated earlier.
- Palm's webOS have created an entire mobile operating system built on WebKit, turning the phone into a web browser; “native applications” are just web applications.

- Another innovation is the PhoneGap project, an open source effort that allows you to create native applications for iPhone, Android, and BlackBerry devices, exposing many device features like location and file system access to your web app.
- These applications can be distributed and sold in device marketplaces, but they share the same code and design.
- And because it is a web app, we can make a less capable version of our app available for free to lower-end mobile browsers.
- Build once, and deploy everywhere.
- The rate of innovation for creating mobile web apps across every mobile device maker is at its highest level in years, but more important than that, for the first time device makers are all working toward achieving the exact same standards, which just happen to be the same standards as the desktop web.



Your Queries Please!!!

Previous Topic.....

Mobile Web Apps Vs Native Applications:

- **The Ubiquity Principle**
- **When To Make a Native Application**
- **When To Make a Mobile Web Application**

Today Discuss about.....

- MOBILE WEB DEVELOPMENT
- DESIGNING FOR MULTIPLE MOBILE BROWSERS
- DEVICE PLANS
- MARKUP
- DOCUMENT STRUCTURE

MOBILE WEB DEVELOPMENT

WEB STANDARDS

- Web Standards is defined as a formal set of standards and technical specifications used to define aspects of the world wide web. These are best-practice standards used by organizations to build web sites and web applications.
- Web standards is just an easy way to say “a web page based on the XHTML 1.0 and CSS 2.1 specification, coded in such a way that the majority of presentation elements are omitted from the XHTML markup and defined in the CSS instead.”

DESIGNING FOR MULTIPLE MOBILE BROWSERS

- Designing and developing for multiple mobile browsers simultaneously is a challenge, but not impossibility.
- For example, you are creating the markup and styles for a desktop site that has to support Internet Explorer 6, which has very quirky support for web standards techniques.
- You would employ a different technique to express your design; one that you know is proven to be compatible on most browsers, than if you were just going to support the latest browsers.

Progressive Enhancement

- *Progressive enhancement* is the practice of using web techniques in a layered fashion to allow anyone with any web browser to access your content, regardless of its capabilities.
- This means that you are creating not just one ideal experience, but multiple less-ideal experiences, depending on who views the content, also called *graceful degradation*. To illustrate this, take a look at Figure.
- This can be done with or without content adaptation, although my advice is to always start your project without.
- It's far faster and easier to start with one code base that gracefully degrades than immediately jumping into creating multiple versions of your site.
- It should be noted that you can add a desktop layer to your site using progressive enhancement.

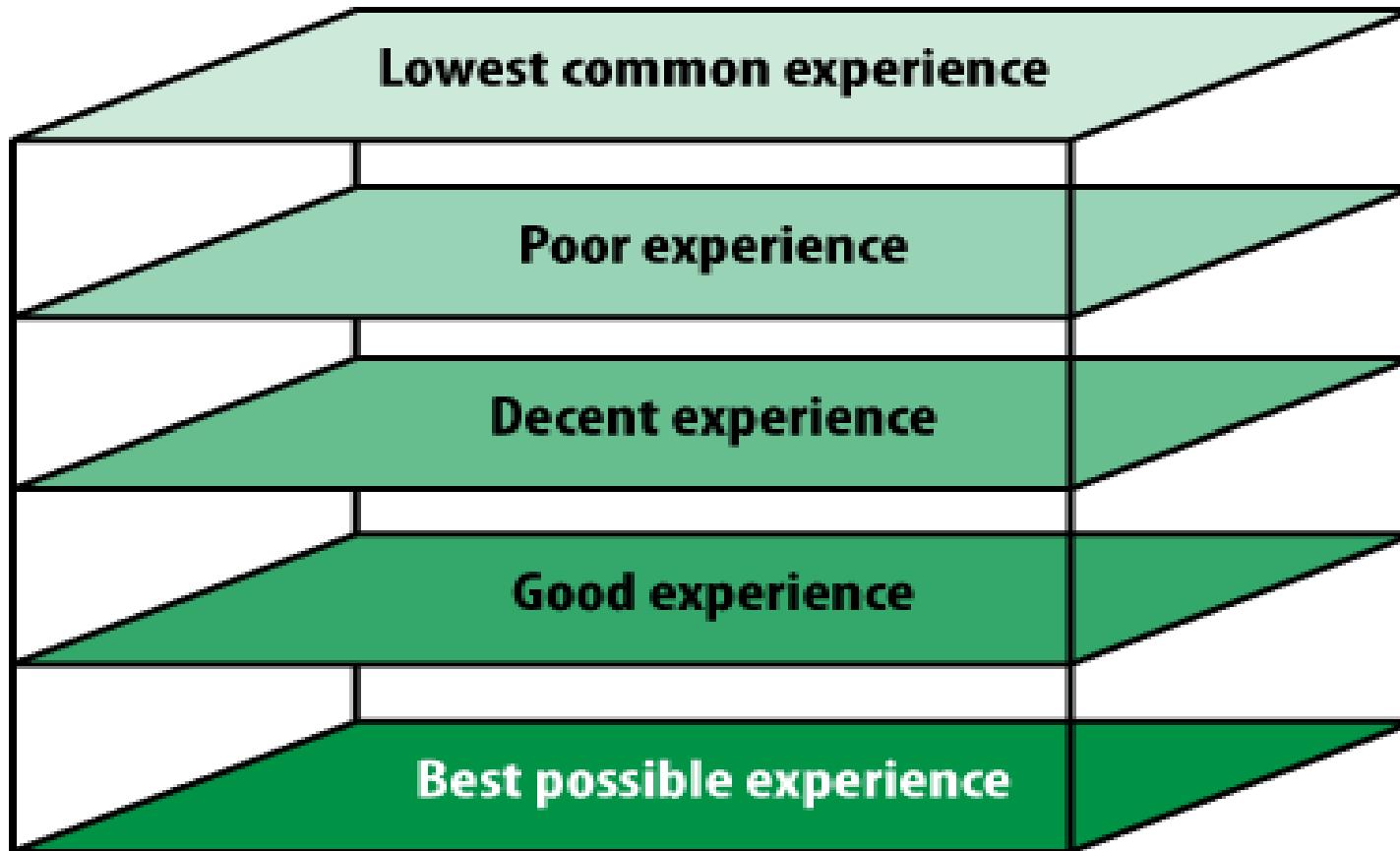


Fig: Using progressive enhancement to layer support

DEVICE PLANS

- Developing a mobile product means having a device plan at the very start.
- The goal is to get to the test stage without any surprises.

The Device Matrix

- Table provides a listing of popular browsers and their assigned classes, starting with A, the highest grade, to be considered on par with desktop browsers, and ending with F, the lowest possible grade.

Class	Markup	CSS	JavaScript
Class A	XHTML, XHTML-MP, HTML5	CSS2, CSS3	Great, includes DHTML, Ajax
Class B	XHTML, XHTML-MP	CSS2 (Decent)	Limited, some DHTML
Class C	XHTML, XHTML-MP	CSS2 (Limited)	Limited
Class D	XHTML-MP	CSS2 (Basic)	None
Class F	XHTML-MP, WML	None	None

Class A mobile browsers

- Some of the characteristics of a Class A mobile browser are:
- Excellent XHTML 1.0 support
- Good HTML5 support; specifically, the canvas element and offline storage

Class B mobile browsers

- Some of the characteristics of a Class B mobile browser are:
- Excellent XHTML 1.0 support
- Padding, border, and margin properties are correctly applied
- Can reliably apply colors to links, text, and background
- Supports image replacement techniques
- Minimum screen width: 164 pixels

Class C mobile browsers

- Some of the characteristics of a Class C mobile browser are:
- Good XHTML 1.0 support
- Limited CSS Level 2.1 support (scores 50 percent or higher on the ACID2 test)
- Limited or no JavaScript support

Class D mobile browsers

- Some of the characteristics of a Class D mobile browser are:
- Basic XHTML
- Limited CSS support (CSS Level 1, or does not recognize cascading)
- Minimum screen width: 120 pixels
- Hyperlinks may not be colorable by CSS
- Basic table support: 2×2 or more
- colspan and rowspan may not be supported

Class F mobile browsers

- Some of the characteristics of a Class F mobile browser are:
- Poor table support or none at all
- Basic forms: text field, select option, submit button

MARKUP

- Markup is used to make content readable by mobile browsers.
- Normally when we think of markup, we think of HTML, or Hypertext Markup Language, the language of the Web.
- It was quickly followed by Nokia's TTML (Tagged Text Markup Language) and other proprietary markup languages

- HyperText Markup Language (HTML) – the original markup language that was defined as a part of implementing World Wide Web.
- Extensible HyperText Markup Language (XHTML): HTML reformulated in XML syntax.
 - XHTML Basic – a subset of XHTML for simple (typically mobile, handheld) devices. It is meant to replace WML, and C-HTML.
 - XHTML Mobile Profile (XHTML MP) – a standard designed for mobile phones and other resource-constrained devices.
- Mathematical Markup Language (MathML)
- Scalable Vector Graphics (SVG)
- TeX, LaTeX – a format for describing complex type and page layout often used for mathematics, technical, and academic publications.

- XML stands for EXtensible Markup Language.
- XML is a markup language much like HTML
- XML was designed to carry data, not to display data.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language.

DOCUMENT STRUCTURE

Doctypes

1. Describes the format of an electronic document. Some examples include .doc, .xls, .mdb, .pdf, and .jpeg.
 2. Document that accompanies files coded in Standard Generalised Markup Language (SGML) to identify what each markup means and control the structure of the coded files.
- The purpose of a DTD (Document Type Definition) is to define the legal building blocks of an XML document.
 - A DTD defines the document structure with a list of legal elements and attributes.
 - <!DOCTYPE NEWSPAPER [
 - <!ELEMENT NEWSPAPER (ARTICLE+)>
 - <!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
 - <!ELEMENT HEADLINE (#PCDATA)>

Character Encoding:

- Many **character encoding standards**, such as those in the ISO 8859 series, use a single byte for a given **character** and the **encoding** is a straightforward mapping to the scalar position of the **characters** in the coded **character** set. ... The **encoding** forms that can be used with Unicode are called UTF-8, UTF-16, and UTF-32
- **Encoding** is the process of converting data from one form to another. While "**encoding**" can be used as a verb, it is often used as a noun, and refers to a specific type of **encoded** data. There are several types of **encoding**, including image **encoding**, audio and video **encoding**, and character **encoding**.
- **UTF-8** can represent any character in the Unicode standard. **UTF-8** is backwards compatible with ASCII. **UTF-8** is the preferred **encoding** for e-mail and web pages. **UTF-16**. 16-bit Unicode Transformation Format is a variable-length character **encoding** for Unicode, capable of **encoding** the entire Unicode repertoire.

MIME types(Multipurpose Internet Mail Extensions):

- It's a way of identifying files on the Internet according to their nature and format. For example, using the "Content-type" header value defined in a HTTP response, the browser can open the file with the proper extension/plugin.
- Servers sending MIME types provide important information to browsers on how to treat a document.
- Sending incorrect MIME types with a document may cause the browser to incorrectly interpret and fail to render the document.
- Administrators often set up web servers correctly for common document types such as HTML and CSS, but not for XHTML-MP.

Page titles

- Page titles surrounded by the <title> element are an important and frequently overlooked page element.
- Good titles increase the findability and usability of web pages.
- Add a short descriptive page title for easy identification, but remember that the mobile device may truncate the title.

Use of stylesheets

- Many mobile browsers prioritize markup before presentation, loading stylesheets and images last.
- This sometimes causes markup to appear with styles briefly while the external stylesheet loads, known as the “screen flash.”
- You can avoid this by adding styles to the document head instead of using an external stylesheet.

Objects and scripts

- Most mobile devices don't support embedded objects or scripts, and it's not possible for users to install plugins to provide support.
- Even where a device does support scripting, avoid using it unless you can't find another means to express your design.
- Though many modern browsers support scripting, you may want to skip it altogether so that you limit your power consumption and have fewer rendering inconsistencies to contend with.

Redirects

- Using markup to redirect pages increases the load time and cost as a result of downloading and processing another page.
- If you need to use redirects, configure the server to perform redirects using HTTP 3xx codes.

Caching

- Using cached information sometimes reduces the need to reload resources such as images and stylesheets, thereby lowering download times and costs.
- By specifying cache information on your mobile pages, you lower the number of times devices download common resources.
- This especially helps resources like a stylesheet or logo, as shown in the following example:
- `<meta http-equiv="Cache-Control" content="max-age=300"/>`

Minimal document structure

It is good practice for documents to indicate structure with headings and subheadings.

- The following is an example of coding semantically:
- <h1>Top Level Heading</h1>
- <h2>Second Level Heading</h2>
- <p>Paragraph Body</p>
- <h3>Third Level Heading</h3>
- <p>Paragraph Body</p>

TEXT ELEMENTS

Headings

- HTML defines **six levels of headings**. A **heading** element implies all the font changes, paragraph breaks before and after, and any white space necessary to render the **heading**. The **heading** elements are **H1**, **H2**, **H3**, **H4**, **H5**, and **H6** with **H1** being the highest (or most important) **level** and **H6** the least.
- `<h1>...</h6>`

Paragraphs

- <p>...</p>
- The paragraph is the tag you will probably use the most.
- Each paragraph of text should be wrapped in the paragraph tag.
- The paragraph will apply default margins to the top and bottom on the element, which can be modified in the CSS.
- Historically, paragraph tags would not be used due to the poor CSS support of devices; break tags would instead be used to create line breaks between paragraphs

Quotations

- <bq>...</bq>
- The blockquote is used for quotations or comments, and is often used as a wrapper tag for one or more paragraph tags.
- Blockquotes often inherit margin around the entire element to give the appearance of being indented from the primary text, which may not be rendered consistently on all devices.

Unordered lists

```
<ul><li>...</li></ul>
```

- HTML lists are used to present list of information in well formed and semantic way. There are three different types of list in HTML and each one has a specific purpose and meaning.
- **Unordered list** — Used to create a list of related items, in no particular order.

```
<ul> <li>Chocolate Cake</li>
<li>Black Forest Cake</li>
<li>Pineapple Cake</li> </ul>
```

Output:

- Chocolate Cake
- Black Forest Cake
- Pineapple Cake

Ordered list — Used to create a list of related items, in a specific order.

...

- Ordered lists are not used as often as unordered lists in desktop sites.

```
<ol> <li>Fasten your seatbelt</li>
<li>Starts the car's engine</li>
<li>Look around and go</li> </ol>
```

Output:

1. Fasten your seatbelt
2. Starts the car's engine
3. Look around and go



Your Queries Please!!!

TEXT ELEMENTS

Definition lists

```
<dl><dt>...</dt><dd>...</dd></dl>
```

- A definition list is for lists that contain term and definition pairs, and is useful for creating repetitive lists where you simply need a title and do not wish to use a header; this is often used to denote a section.
- For Class A browsers, I use definition lists to structure forms, allowing me to place the form label either above or to the left of the form input or control.

Structural elements

```
<div>...</div><span>...</span>
```

- The div and span elements are just as critical to mobile web development as they are to desktop web development.
- The div is used to identify and label any block-level division of text or content; this could be a line of text, or it could be an entire page, whereas the span tag is used to identify a grouping of inline elements and is often used within a block-level element.
- The div and span are essential for adopting a progressive enhancement strategy, allowing you to conditionally define what content is seen for what devices.

Line breaks

, <hr />

- Line breaks and horizontal rules work as expected on virtually all mobile devices.
- However, breaks should be used sparingly and not as a replacement for the paragraph tag.

Character entity references

nonbreaking space, &, <, >, ', ", TM, ©

- Common characters not found in the normal alphanumeric character set, such as & or ©, must be specified in XHTML as character references starting with an & and ending with a ;.
- Entities are often expressed in entity, decimal, or hex syntax; however, for mobile devices, the entity syntax should be used.
- For example, a nonbreaking space would be coded as and an ampersand is &.

CREATING LINKS

- Links are the foundation of how hypertext works.
- They can take you to new pages or be used as an anchor to content further down the page.
- With XHTML-MP, links can also initiate a telephone call and perform other device actions in certain phones.

Number of links

Too many links on a page makes it difficult for the user to navigate and read content.

Most mobile browsers stop the vertical scroll when a link appears, meaning that for each press down on the D-pad, the user is taken only as far as the next link.

Try to limit links to 10 links per page and add access keys to links whenever possible so that users can navigate with the keypad as well.

Access keys

- Navigating a mobile site can be difficult and cumbersome, but you can simplify navigation and limit scrolling by providing keyboard shortcuts for common links for devices with number pads (this obviously doesn't apply to touch-only devices).
- Associating an access key attribute with each link gives the user an easy way to access the link using the device's keypad.
- Access keys come in handy when used consistently across a site, letting users jump quickly to their chosen sections without scrolling to find a link.
- You may have more than 10 links per page. Try to create access keys for all navigational links.
- You can save some access keys, because it isn't necessary to create access keys to links appearing within content blocks.

Initiating telephone calls

- These information devices are, after all, phones, so XHTML-MP includes a means to initiate a telephone call within the `<a>` element, by prefacing the full phone number, including country code, with `tel:` within the `href` attribute.
- This will prompt the user to initiate a telephone call:
`+1 (555) 555-1212`

IMAGES AND OBJECTS

- The desktop web is rich with a variety of embedded content; however, due to the hardware limitation of many devices, you cannot assume that all mobile devices have the same capabilities.

Image types

- Nearly all mobile devices support the JPEG, GIF, and PNG formats.
- Both the 8-bit PNG and the 24-bit PNG with alpha transparency are supposed to be supported as of WAP 2.0, but some older devices may not support them, due to hardware limitations.
- Whenever possible, use PNGs, as they are the recommended image format for the mobile web.

Image sizes

- Adding images to your content can be tricky.
- The safe approach is to edit images so that they're as small as possible in terms of pixel dimensions.
- With most mobile device screens about 120 pixels wide, it is recommended that you not use images any wider than that. However, there are several devices with screens much larger.
- If you are using a content adaptation system, you can dynamically insert different images based on the requesting device.
- Conversely, you can load larger images meant for larger devices, and use CSS to reduce the image height and width for smaller devices, but this means that lower-end devices are downloading larger images.

Image dimensions

- Not specifying the pixel height and width of an image forces the mobile device to calculate the values, increasing render times and degrading performance in lower-end devices.
- Bitmap images have an intrinsic pixel size, so telling the browser their size in advance avoids having the browser recreate the page when it receives them.
- Letting the server resize the image cuts down the amount of data transferred and the amount of time it takes for the client to process and scale the image.
- If the specified width and height attributes match the intrinsic size, then the client doesn't resize the image.

- This can, of course, be redefined using CSS, though some mobile devices might not override sizes defined in CSS correctly. If you plan to use progressive enhancement techniques, it may be easier to wait to define image sizes until you have tested them on your target devices.

Image maps

Most devices lack a pointing device, making it difficult for users to use image maps. If you know that the device supports touch, you can certainly use image maps, but recommend avoiding them entirely.

Alt text

Always provide alt text values for all images. Downloading images can take a considerable amount of time to load over a mobile network. Some mobile browsers allow you to disable downloading images, opting for text-only mode in order to increase rendering speeds.

Having alt text will ensure that any important images can be seen, regardless of the users' preferences.

Flash and SVG

- Though many devices support vector objects like Adobe Flash, SVG (Scalable Vector Graphics), and SVG-Tiny, we are still a few years out from seeing these formats ubiquitously supported, due to hardware constraints.
- Avoid using any vector graphics unless you specifically know that the targeted devices support it.

Embedded audio and video

All WAP 2.0 devices should support the 3GPP video format and the MP4 audio format, meaning that if you link to one of these resources, it should be able to be played in the device player.

However, due to hardware constraints, not all mobile web browsers support the ability to embed this content into web pages.

TABLES

Layout tables

- The web design industry considers using tables for layout as bad practice, particularly for mobile devices.
- Table-based layout combines presentation and markup, which makes development more difficult and essentially eliminates the ability to adapt to other media.
- Though tables provide a more consistent mobile web experience, they're cumbersome and difficult to support.
- Table-based layouts restrict your ability to adapt for various devices and to increase page size.

Using data tables

- On smaller screens, data tables—or content considered to be tabular data—often doesn't fit or can appear erratically.
- Unless you know that a device supports tables, avoid using them. Smaller tables with two or three columns work on most devices, but even then they're not recommended.
- Try using a definition list (`<dl>`) instead of a table to vertically display data.

Nested tables

- Nested tables, like layout tables, don't really belong in mobile design—especially because they have a tendency to render inconsistently and add to the page size.
- In part because of the conditions mentioned earlier, stay away from nested tables for controlling presentation.
- Instead, focus on creating well-formed XHTML-MP and control the presentation with stylesheets.

FRAMES

Frames just don't work in mobile design. In most cases, either devices don't support them or they cause a variety of usability problems. Instead, try applying server-side includes for loading local content.

FORMS

- Designing and developing great forms can be a challenge in the mobile context.
- In the Class A browsers, forms can resemble their desktop cousins, but for all other browsers, forms don't always render like you might expect.
- However, the larger challenge is actually for the user, as forms are difficult to control and add content to.
- The rule of thumb is to limit the use of forms in the mobile context.

Free text input controls

Though unavoidable in forms that need information from the user, avoid using text boxes and text areas as much as possible. It's difficult for the user to enter content into free text input controls such as text boxes and text areas. Instead, rely on radio buttons, select boxes, and even lists of links to reduce the need for text entry.

Default input mode

- It's possible to limit the type of data entered into an input field by defining the input mask or input mode using Wireless CSS or CSS-MP, thereby making it easier for users to enter information into a free text field.
- The input mode (alphanumeric or numeric) of the mobile device's keypad is automatically set according to the input mask value.
- The following example limits the input to only numeric values:

```
<input type="text" style=' -wap-input-format: "*N"' />
```

- This example limits the input to alpha characters by capitalizing the first letter:

```
<input type="text" style=' -wap-input-format: "A*a"' />
```

OTHER RECOMMENDATIONS

Validate markup

- No validating markup may not display correctly or efficiently on mobile devices.
- In some cases, especially on older phones, nonvalidating XHTML-MP won't render at all, leaving users with an error message in their browser.
- Using the W3C validator can be helpful for finding rendering errors.
- To check markup against the W3C mobile web best practices, you can validate your code at <http://validator.w3.org/mobile/> or <http://ready.mobi>.

Pop-up windows

- Most mobile devices don't support pop-up windows. Even when they do, try not to rely on them, because changing the current view can be disorienting to the user.

External resources

Most mobile browsers download each resource as a separate element, beginning with downloading and rendering markup, followed by stylesheets and images.

Depending on network speed, the user may see the basic markup while external resources download.

When the download finishes, the browser renders the page again with the included elements. Carefully consider the number of external resources you use, limit them, and keep each resource's file size as small as possible.

Total page download size

- Page sizes (including images and stylesheets) should remain as small as possible, because large pages take longer to load and cost the user money.
- Try to target your combined page weight to be between 10–25 KB whenever possible.
- Avoid exceeding 50 KB per page, as download times and approximate cost become increasingly prohibitive.

CSS: CASCADING STYLE SHEETS

Designing your CSS to work across multiple mobile browsers isn't easy and can be quite a painful process.

There is no one perfect way to create CSS that renders consistently on more than a handful of devices. I have three techniques I use, depending on the devices to support:

Keep it simple

Keeping your styles very basic, using no complex styling techniques whatsoever, can be the ideal method for simple sites. Though it may not be pretty to look at, it works.

Code and reload

In this approach, you constantly test how your styles render on devices.

For each code change, you reload the browser on each device you plan to support. This approach is slow and tedious, but it means fewer issues toward the end of the project.

Progressive enhancement

- This approach requires you to create multiple layers of support, so that your style gracefully degrades depending on the device.
- This technique takes some practice to get right, but if you can master it, it can be a powerful approach.

CSS: CASCADING STYLE SHEETS

WIRELESS CSS AND CSS-MP

For markup, we have XHTML-MP, a descendant of XHTML; it only makes sense that we have a mobile equivalent for CSS.

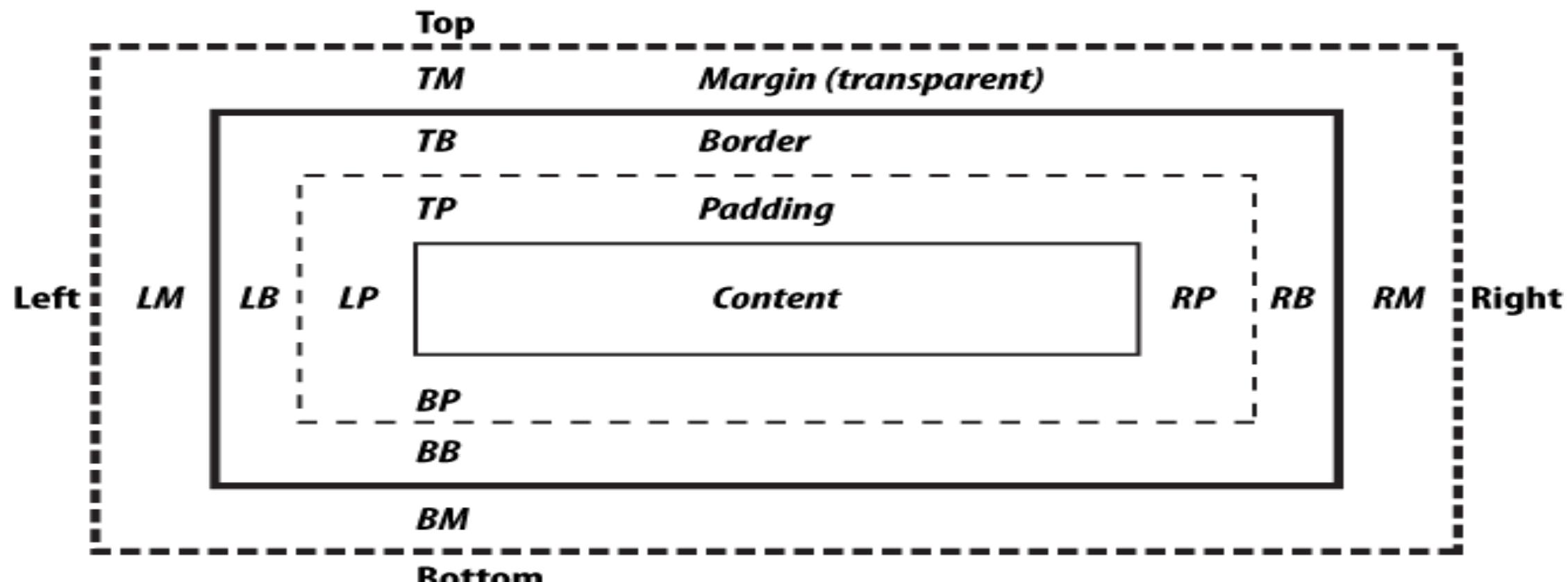
In fact, we have two: Wireless CSS (sometimes referred to as W-CSS or WAP CSS) managed by the OMA and part of WAP 2.0 along with XHTML-MP.

And then we have CSS-MP, or CSS Mobile Profile, managed by the W3C.

So, in case you are keeping score: the OMA owns XHTML-MP and W-CSS, and the W3C owns XHTML Basic and CSS-MP.

BOX MODEL

- The box model is one of the key concepts of CSS design, and therefore the first thing that tends to go wrong in mobile devices.
- The box model is the imaginary box that is around every element in your markup.
- It consists of five areas: the content, the padding, the border, the margin, and the outer edge, as shown in [Figure](#)



- Margin edge
- Border edge
- - - Padding edge
- Content edge

SELECTORS

- The selector is used to tell which markup elements it should apply rules to—basically, what makes CSS work to control the presentation.
- There are a number of different types of selectors:
 - ⊕ Universal
 - ⊕ Type or element selectors
 - ⊕ Contextual selectors (descendant, child, and adjacent sibling)
 - ⊕ Class and ID selectors
 - ⊕ Pseudoclasses
 - ⊕ Pseudoelements

Universal selector

- The universal selector selects all elements useful for defining the default typeface or font size. body or html can be used as well:

```
{font-family:serif;}
```

Type selector

- The type or element selector targets the element by name.
- Type selectors are fairly safe to use across all mobile devices:
`h1 {color: red;}`

Descendant selector

- The descendant selector targets elements that are descendants of another element, indicated by a single space separation.
- Despite being a fairly standard technique for styling desktop sites, the descendant selector is not consistently supported on Class C or lower browsers:
`li a {color: black;}`

Child selector

The child selector is similar to the descendant selector, but targets only the child of the defined element, not all instances like the descendant selector.

Like the descendant selector, Class C browsers and lower can render child selectors inconsistently:
`p > em {background-color: yellow;}`

Adjacent sibling selector

- The adjacent sibling selector targets an element that comes immediately after another element with the same parent.
- Again, this is not a widely supported selector for Class C or lower browsers, but you can find some incompatibility among Class B browsers as well:

```
h1 + p { margin-bottom: 1em; }
```

Class selector

- The class selector targets any element with the matching class.
- Class selectors are one of the more common techniques used for cross-platform mobile designs, given their wide support in nearly all mobile browsers:

```
.error {background-color: red; }
```

ID selector

- The ID selector targets any element with the matching ID (remember that all IDs need to start with at least one letter).
- ID selectors, like class selectors, are one of the more common techniques used for cross-platform mobile designs, thanks to their wide support in nearly all mobile browsers:
`#alert {background-color: yellow;}`

Simple attribute selector

- A simple attribute selector targets all elements with the matching value. These attribute selectors tend to work only in newer mobile browsers:
`p[class] {color: blue;}`

Pseudoselectors

- You can use pseudoselectors to target elements that might not exist in the markup, like a visited link. Unfortunately, pseudoselectors do not have strong support in Class B or lower devices:
:link, :visited, :hover, :active, :before, :after, :first-child, :last-child
- In [Table](#), you can see how different CSS selectors fare in each browser class.

Table: CSS selector compatibility table

Selector type	Class A	Class B	Class C	Class D	Class F
Universal selector	Yes	Yes	Yes	Yes	Yes
Type selector	Yes	Yes	Yes	Yes	Flaky
Descendant selector	Yes	Yes	Flaky	Flaky	No
Child selector	Yes	Yes	Flaky	Flaky	No
Adjacent sibling selector	Yes	Yes	Flaky	No	No
Class selector	Yes	Yes	Yes	Yes	Flaky
ID selector	Yes	Yes	Yes	Yes	Flaky
Simple attribute selector	Yes	Flaky	No	No	No
Advanced attribute selector	Yes	No	No	No	No
Pseudoselector	Yes	Flaky	No	No	No

FONT AND TEXT PROPERTIES

Available fonts

- With the desktop web, we have at least 10 different fonts we can use in our designs. In mobile development, we can count on only two options: serif and sans-serif.
- In lowend devices, we might have only one option—almost always a sans-serif variant:
`p {font: sans-serif;}`

Absolute size keywords

It's not very common in desktop web design, but the typical method to determine the size of text is using one of seven absolute size keywords: xx-small, x-small, small, medium, large, x-large, and xx-large:
`p {font-size: xx-small;}`

Percentage measurements

- Percentage size values will work on Class B and higher browsers, but I wouldn't recommend them:
`p {font-size: 80%;}`
- Percentage size values rely on inherited parent values, which are not consistent across all browsers. Use absolute size keywords instead.

Length measurements

- One of the more common means to style text on the desktop web is by using a specific number of units—for example the pixel, which is relative to the display resolution or ems, or the distance from baseline to baseline:

```
p {font-size: 10px;}  
p {font-size: .9em;}
```

Additional styling

- There are a great number of ways to style text.
- The majority of them will work across multiple devices, given that if there is one thing a mobile browser should do fairly well, it is display text:

- ❖ font-weight: bold;
- ❖ font-style: italic;
- ❖ text-transform: capitalize;
- ❖ text-decoration: underline;
- ❖ line-height: 2em;
- ❖ text-align: right;
- ❖ white-space: nowrap;

Text shadow

- Creation of a text shadow, though supported only by Class A browsers, is a useful technique for mobile devices, reducing the need for images to create a desired visual effect:

```
text-shadow: 2px -1px 2px #ccc;
```

Font replacement

Given the limited typography options available in the desktop web, many designers and developers are starting to use text replacement techniques.

There are three techniques used—some that work on Class A mobile browsers only:

@font-face

This easiest method is to define a font using the CSS3 at-rule @font-face, which references the typeface file to be used. But due to the heavy processing required to render custom fonts, this technique is not yet widely supported.

sIFR

The technique uses Flash to replace the text with a small vector representation of the desired glyphs. However, this technique requires a browser with the Adobe Flash Lite plugin.

canvas

The third technique is to use JavaScript and the canvas element to render text to the device. I've found this to be the only reliable way to replace text on mobile devices, but only for browsers that have good JavaScript support and that support the canvas element.

BASIC BOX PROPERTIES

Height and width

- Height and width can be specified on the majority of mobile browsers, but issues can occur when trying to define percentage-based height values:

```
width: 100%;  
height: 50px;
```

Minimum and maximum dimensions

- Minimum and maximum dimension values are not a reliable means to style an element across multiple mobile browsers.
- Though it is a very useful technique in the mobile context, it is largely used for Class A browsers only:

```
min-width: 120px;  
max-height: 100px;
```

Margins

- The margin is the area applied to the outside of the element, including the border area. Luckily, margins render fairly consistently and can be relied on across multiple mobile browsers:

`margin: 10px;`

Padding

- Padding is applied to the area within an element, inside of the border area. Like margins, padding works across multiple mobile browsers, however, not as reliably in lower-end browsers that have flawed box models. However, this can be worked around by applying margins to a child element:

`padding: 10px;`

Borders

- Simple borders, like this one, will work in many mobile browsers, but can sometimes cause some inconsistencies in low-end browsers that have flawed box models:

`border: 1px solid black;`

Box shadow

Like text shadowing, the box shadow is a useful way to create a desired visual effect without relying on downloading images to the device, though it is supported only by Class A browsers:

box-shadow: 10px 10px 5px black;

	Class A	Class B	Class C	Class D	Class F
Height and width	Yes	Yes	Limited	Flaky	Flaky
Min and max dimensions	Yes	Flaky	No	No	No
Margins	Yes	Yes	Yes	Yes	Limited
Padding	Yes	Yes	Yes	Flaky	Flaky
Borders	Advanced	Limited	Limited	Flaky	Flaky
Box shadow	Yes	No	No	No	No

COLOR AND BACKGROUNDS

- Styling an element means defining colors and background images. Relying on CSS instead of images to create desired visual effects reduces time to download as well as cost.

Background color

- The background color allows you to add a color value to the content area of an element. This technique works fairly reliably across all mobile devices:

```
background-color: red;
```

Background image

- The background-image property is incredibly useful for creating mobile designs, allowing you to send small images to the device and tile them in a number of ways to create bandwidth-friendly designs.

Multiple background images

Though supported by Class A browsers only, the multiple background support of CSS3 comes in quite handy when creating mobile designs.

POSITIONING AND PAGE FLOW

- CSS goes beyond just being able to add design content within the page; it can also be used to define the design layout of the page.
- Using positioning and page flow attributes, we can add style to the page and help make it easier to read or interact with on small screens.

Display

- The display property allows you to define how an element is to appear in the page flow.
`display: block;`

Floats

Floats are usually applied to images and other elements to position them either to the right or left of the page flow, causing nearby elements to wrap around them. Due to the smaller screens of mobile devices, floats are not encouraged for layout use.

Clearing

- Clearing goes hand in hand with floats, allowing you to prevent how block-level elements wrap in the use of a float.
- Usually, support for clearing in mobile devices is added to the browser whenever it adds support for floats:

`clear: left;`

Positioning

- Both relative and absolute positioning might seem to be odd tools for mobile design given the small screen size, but they actually come in handy for creating designs for Class B and higher browsers:

`position: absolute;`

`top: 10px;`

`left: 50px;`

Overflow

- There are several types of overflow: visible, hidden, scroll, and auto.
- This property is designed to control how content is displayed when it exceeds the defined value of an element:

`overflow: hidden;`

- Both the visible and hidden types are fairly well supported in Class B and higher browsers, allowing you to save space in lists and menus.

Stacking order

The z-index element is used to determine the stacking order of positioned elements:

`z-index: 10;`

This property is used when overlaying elements on top of each other to create a specific visual effect. Though used mostly for designs targeting Class A browsers, it does work on other classes as well.

Table shows how positioning and page flow attributes are supported in each of our mobile browser classes.

JAVASCRIPT

- Last, but not least, we come to JavaScript: the last pillar of mobile web development.
- Unfortunately, JavaScript simply hasn't been a priority in mobile browsers for many years, due to the hardware limitations of devices.
- The bad news is that unless you are targeting Class A browsers, JavaScript just can't be assumed in your mobile project.
- Just go back to the concepts of progressive enhancement that at the beginning of this chapter, and start to map out how your experience will degrade to each of the device classes you plan to support.

- The good news is that we are starting to see widespread JavaScript support in multiple mobile browsers with Class A browsers and a few Class B browsers.
- Some devices, like the iPhone, are taking JavaScript to a new art form
- As devices get more powerful and browser engines more adept at processing JavaScript.

[Table](#) shows that JavaScript is available only in a few devices.

	Class A	Class B	Class C	Class D	Class F
JavaScript support	Yes	Some	No	No	No
DHTML	Yes	Limited	No	No	No
Ajax	Yes	Limited	No	No	No

QUERIES?

Previous Topic.....

- **CSS-WIRELESS CSS AND CSS-MP**
- **BOX MODEL**
- **SELECTORS**
- **FONT AND TEXT PROPERTIES**
- **BASIC BOX PROPERTIES**
- **POSITIONING AND PAGE FLOW**
- **JAVASCRIPT**

UNIT – IV

iPhone Web Apps: Why Web kit? – Markup – CSS - JavaScript- Creating a Mobile Web App- Web Apps as Native Apps- PhoneGap- Tools and Libraries.

Today discuss about.....

- iPhone Web Apps
- Why WebKit?
- Why WebKit?
- CSS

iPhone Web Apps:

iPhone Web Apps

- The iPhone was a game changer in the mobile ecosystem.
- The mobile web browser behind the iPhone and iPod touch,a new category of mobile content: the mobile web app, which existed before the iPhone.

Why WebKit?

- Here are a few reasons why I believe the iPhone is worth prioritizing:
- The iPhone has proven to be the market leader in terms of innovation and market share of the devices that access the mobile web the most.
- The iPhone is marketed, sold, and supported by Apple, not the operator. operators tend to promote phones for only short periods of time. Apple's strong reputation and brand appeal make it unique.
- The lowest development cost for the highest number of supported devices. Only a text editor and a web browser. No devices, emulators, or special software are required for iPhone web app.
- It requires little to no additional knowledge apart from HTML, CSS, and JavaScript.
- It has a simple and cost-effective testing environment. —device testing on an inexpensive iPod touch on a Wi-Fi network and render exactly the same on an iPhone.
- The highest consumer of mobile data.

A Brief History of WebKit

- WebKit is an open source web browser engine derived by Apple from the Konqueror HTML layout engine called KHTML and its KJS JavaScript engine.
- Apple ported KHTML and KJS to Mac OS X to create the first Safari desktop browser in 2003 and has since been used in a number of Mac OS X applications.
- At Netscape, Hyatt created a Mac OS X application called Chimera that was based on Mozilla's Gecko rendering engine.
- Chimera, later to be renamed Camino, was at the time the fastest web browser on any Mac or PC, showing that the Gecko could be turned into a fast, lightweight browser.
- Camino also introduced or popularized many features commonplace in the web browser, like tabbed browsing and pop-up blocking.

Background As a Mobile Browser

- WebKit's life as a mobile browser engine did not start at Apple.
- It started at Nokia, the number-one device maker in the world, long before anyone outside of Apple knew
- about the iPhone.

1)Web Browser for S60

- The Web Browser for S60, or the Nokia Mini Map Browser , is the defaultmobile browser for the S60 platform, common in most N and E series Nokia smartphones.
- The browser uses a Mini Map, for which it is nicknamed, to allow the user topan around and zoom into a web page not optimized for mobile devices.

2)iPhone and iPod touch

- WebKit made its big splash in the mobile market as the device browser for the iPhoneand iPod touch, usually referred to as Safari or Mobile Safari.
- Apple'sversion of WebKit includes the ability for users to double-tap on an area of a nonoptimized
- web page.
- Users can pan around the pagewith their finger, rotate the orientation, or use the multitouch gestural features, like
- pinching to zoom in and outthe iPhone initially supported only web apps, not native applications



3)Android

- The open source mobile framework Android is a WebKit-based device browser.
- Android is open source and can be customized by the device maker or the operator, the WebKit browser can be replaced by another browser written for the Android platform,

4)WebOS

- Palm has created an entire operating system based on WebKit, called webOS. The Mojo SDK, developers can create “native” applications using HTML, CSS, and JavaScript. All applications render using the WebKit browser engine.



5)Myriad Browser

- The Myriad Browser v9 is a browser from the Myriad Group that is also based on WebKit.
- Myriad, like most mobile browser makers, works with device makers and operators to preload devices with their product, which is often referred to as the device browser.

6)Iris Browser

- The Iris Browser from Torch Mobile is one of the many WebKit browsers designed for Windows Mobile devices.
- Torch Mobile works with operators to preload the IrisBrowser on Windows Mobile devices, and is not meant to be a direct-to-consumer product, though Windows users can download a free copy from the Torch Mobile website.

Markup

- The specifications for HTML5 or XHTML2 (as well as CSS3) will have been finalized browser makers are moving forward implementing what and then reporting their findings back to the working groups.

1) XHTML

- The HTML standard was published in 1990, it was designed for formatting and laying out elements on the screen contextually.
- The contextual markup of HTML with the semantically structured data of XML.
- On a desktop browser, a mobile browser, a gaming console, and so on, each designed for the context. This is known
- as *web standards*,

a simple example for “web-standard code”

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">  
<head>  
<title>An XHTML 1.0 Compliant Document</title>  
</head>  
<body>  
<p>Here is some text</p>  
</body>  
</html>
```

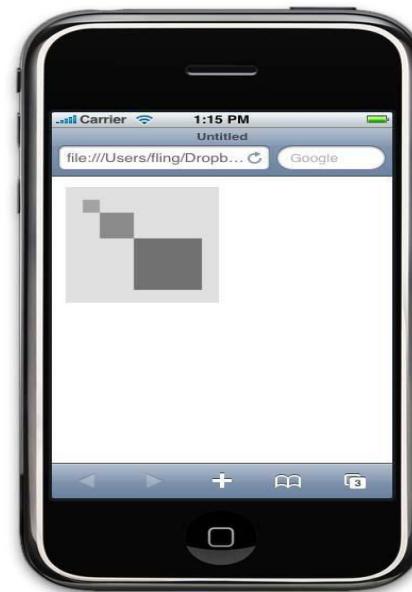
Apple recommends that the developer allow the user to browse the desktop version of the site.

2) XHTML-MP

- The iPhone will render XHTML Basic and XHTML-MP pages.

Testing XHTML-MP code in WebKit *Testing HTML5 code in WebKit*

Using the canvas element on the iPhone



- XHTML 2.0 is the successor to XHTML 1.0, the primary markup language of webstandards ,includingmost iPhone web apps.
- XHTML 2.0 is based on the principles of XML and is meant to be a minimal, all-purpose language used for marking up content to be machine-readable, regardless of context.
- XHTML 2.0 would be a massive change to how we create sites and web applications.

3)HTML5

- HTML5 is designed to be the successor to HTML4, as a transitional standard toXHTML 2.0. It stays fairly close in syntax to the HTML4 and XHTML 1.0 standards.
- HTML5 creates some interesting new opportunities for mobile web applications, likethe canvas element, offline storage, document editing, and media playback.
- HTML5 incorporates the Web Forms 2.0 standard from the W3C's Web Hypertext Application Technology Working Group (WHATWG).
- HTML5 also allows developers to create cross-platform designs through expressing the content more semantically.
- For example, the addition of HTML5 elements like header, nav, article, section, aside, and footer make our
- content more machine-readable.

4)The canvas element

- The canvas element is part of HTML5; it allows designers and developers to essentially draw content within your HTML page.
- The canvas HTML tag defines a custom drawing area within your content that you can then access as a JavaScript object.
- canvas was created by Apple, included in the WebKit source and the iPhone.

5)Offline data storage

- Part of HTML5 and supported by WebKit and the iPhone is the ability to create client-side data storage systems, to create web applications that work when offline.
- Client-side storage has a variety of names: DOM storage, offline storage, and others.
- Cookies can be written to the client by the server and retrieved by the server.
- The type of data best suited for client-side storage is information that does not change often, like contact information or map locations.



Use of offline storage in the Gmail iPhone webapp

CSS

- The iPhone stands apart in its excellent support of CSS and JavaScript.
- CSS support means that you can use the same techniques to create mobile experiences.
- A mobile browser supported some CSS2.
- Most devices supported CSS-MP, the subset of CSS2 meant for mobile devices.
- Browsers and their ability to bring CSS-based, web standard designs to mobile devices on par with what we are accustomed to on the desktop web.

1)CSS2

The iPhone has excellent CSS2 support for a mobile browser.

- WebKit and Safari for the desktop support the full CSS2 specification, passing the CSS2 Acid2 test with a 100 percent score.
- The vast majority of styling techniques we employ to create desktop designs can also be used to create our mobile designs.



How the iPhone fares with the Web Standards Project's Acid2 test for CSS2 support

i)Positioning and page flow

- iPhone supports the majority of positioning techniques, including relative, absolute and fixed positioning.
- iPhone currently does not support fixed positioning to the bottom of the perceived viewport, which is useful for creating bottom tabbed navigation.
- In addition to positioning, page flow styling tools like display: and float: all work as expected and can be reliably used to create complex styled sites and web apps.

ii)Image replacement

- One of the most common techniques in standards-based designs that is still largely missing from mobile design is the ability to replace text with an image defined by your stylesheet.
- One of the more common examples is replacing the h1 element, often indicating the title of the page, with your logo.

For example, you can put your company name as text within the h1 element in your HTML:

```
<h1 id="logo">My Company Name</h1>
```

In CSS, you use the background element to set the logo as a background image, while pushing the actual text beneath the visible area that you define:

```
#logo {  
background: url(logo.png) no-repeat;  
width: 200px;  
height: 75px;  
overflow: hidden;  
line-height: 10em;  
}
```

2)CSS3

- Lots of images and even tables to create simple visual elements like rounded corners, shadows, and semi-opaque areas.
- CSS3 can come in as a wonderful tool for creating complex designs using the minimum of images, making it ideal
- for mobile design.
- The iPhone supports the majority of the CSS3 specification, allowing us to create visually stunning and bandwidth-friendly designs using minimal amounts of code.
- For CSS3, Mobile Safari 3.0 scores 97 percent in the Acid3 test nearly a perfect score, and one that certainly makes it a leader among the Class A browsers.

i)The –web kit prefix

- There are a number of CSS3 modules that are being proposed by the W3C, each in various states of recommendation status
- the browser makers do is add a prefix before the proposed style is ratified.
- Web Kit, you add a –web kit in front, though other browsers use their own prefixes; in the case of Mozilla, you use -moz and Opera, -o.

ii)Selectors

- The iPhone, as a Class A browser, supports the majority of CSS attribute selectors.
- The site CSS3.info has a handy CSS3 selector test that can be performed on mobile browsers, with Mobile Safari 3.0 passing all 43 tests.
- CSS3 attribute selectors come in very handy when creating iPhone web apps.
- selectors means less markup and CSS code, reducing your overall page weight.

iii) Multiple background images

- Multiple background images, when combined with CSS3, are incredibly useful for mobileweb apps.
- There is a 24-bit transparent PNG overlay added to create a glare effect.
- The final result is an icon that is a fraction of the overall weight of usinga single 24-bit transparent PNG file.

- + 
 - +  3 Kilobytes
 - +  320 bytes
 - + `-webkit-border-radius: 10px;`
-



Example of using multiple backgrounds for bandwidth-friendly design

vi) Box model sizing

- The box model is made up of five areas: the content area, the padding, the border, the margin, and the outer edge.
- Creating multiple columns of content with percentage values and adding borders or padding to them is very difficult.
- Box sizing—a CSS3 property supported by the iPhone using the `-webkit-prefix`—has been introduced.

```
<div style="width: 200px; border: 5px black solid; padding: 10px; -webkit-box-sizing:  
border-box;">
```

Box

```
</div>
```

The available options are:

`box-sizing: border-box;`

The specified width is from border edge to border edge

`box-sizing: padding-box;`

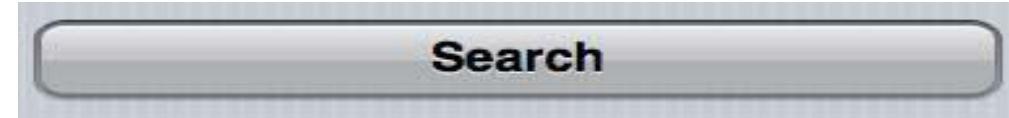
The specified width is from padding edge to padding edge

`box-sizing: content-box;`

The default method

v)Box shadow

- The box-shadow property applies a shadow to the desired element. Like most CSS3 properties, it is useful for reducing the need to use weighty images to achieve desired visual effects:
- -webkit-box-shadow: hoffvoff blurcolor;



vi)Rounded corners

- Rounded box corners are a common visual effect used on the Web,
- Creating the effect can add additional markup and images to your code, increasing page weight and reducing compatibility across multiple devices.
- The CSS3 border-radius property allows us to easily define rounded borders using nothing more than CSS:
- Border imagesThe ability to define border images can come in handy to define visually unique controls
- and elements using a minimal amount of resources:
- -webkit-border-image: url("border.png") 20 14 round stretch;
- In the iPhone user interface library, iUI, the border image is used to create buttons using small reusable images, as shown in the Search button

vii) Text effects

- Text effects can be applied in order to reduce the dependency on images to achieve a desired visual effect.
- The most common is text-shadow, which does not require the -webkit-prefix, and which can be used to create a small drop shadow, or a bevel effect, common in iPhone designs:
- text-shadow: 3px 3px 2px #333333;
- -webkit-text-fill-color: #0000ff;
- -webkit-text-stroke: 1px #000
- Adding the following code to the body will prevent the text size from being changed when the orientation changes:
- -webkit-text-size-adjust: none;

viii) Text overflow

- Designing mobile interfaces for content that isn't entirely within your control, like a dynamic application, content management system, or imported feeds, it is incredibly useful to present a line or two of text

QUERIES?

PREVIOUS TOPICS:

IPhone Web Apps

Why Web Kit?

CSS

JavaScript

Creating a Mobile Web App

Web Apps as Native Apps

PhoneGap

Tools and Libraries.

Today's Topics to
discuss:

CSS

Visual Effects

- There are a number of visual effects that you can perform specifically with the iPhone and iPod touch.
- These effects are part of the Web Kit distribution, but not all devices that use Web Kit support them

i) Gradients

- Web Kit supports two extensions to CSS for producing gradients.
- Added to the background element, the linear and radial gradients can be used to add sophisticated shading and simulate 3-D effects on buttons, and so on, without the use of images.

Prototype Extension

`-webkit-gradient(type, start_point, end_point, / stop...)`

or:

`-webkit-gradient(type, inner_center, inner_radius, outer_center, outer_radius, / stop...)`

start_and end_point

Are the positions for the beginning and ending of the gradient, such as left top or right bottom but can be any combination of left or right and top or bottom.

inner_and outer_radius

Specify the size of a radial gradient's beginning and ending circles.

inner_and outer_center

Specify the start and end positions of the radial gradient, similar to *start_* and *end_point* for linear gradients *stop*.

ii) Masks

Masks are images; **Clips** are paths. Imagine a square image that is a left-to-right, black-to-white gradient. That can be a **mask**.

- Masks are a powerful extension to CSS that can produce effects such as vignetting, soft edges, glare, and images with rounded corners or irregular boundaries.
- The general idea is that multiple images can be layered.
- Each layer can use white, black, or its alpha channel to specify what parts of the back image show through the foreground image.

iii) Transitions

A transition or linking word is a word or phrase that shows the relationship between paragraphs or sections of a text or speech.

- Transitions are an extension to WebKit that can be used to modify a CSS property,
- Such as height or width, over time. Not all properties can be animated with a transition, but the list is fairly large.
- The prototype is:

• -webkit-transition:*property duration timing_function delay*

property

• Is the name of the property in the same style that is to be modified.
duration

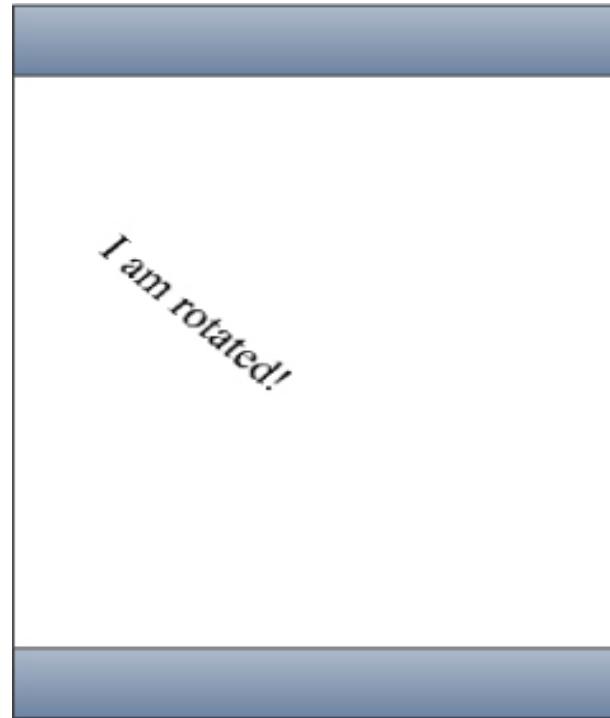
• Specifies how long the transition will take, specified in seconds or milliseconds.
timing_function

Delay

• Specifies how long of a wait there is before the transition begins, specified in seconds or milliseconds.

vi) **Transforms**(to change in composition or structure)

- Transforms are used for modifying geometry of objects through mathematical operations.
- An **example** of to **transform** is painting a room to change the feel of it.
- An **example** of to **transform** is a new hair cut changing a person's appearance.
- Transforms can be used to do incredibly interesting things with your design.



An example of CSS transforms in the iPhone, from the Apple Safari Visual Effects Guide

v) Animations

- Animations, similar to transitions, modify properties over time.
- Animations can provide any number of intermediate values that are not necessarily linear.

vi) Hover, clicks, and taps

The **:hover** selector is used to select elements when you mouse over them.

Tip: The **:hover** selector can be used on all elements, not only on links.

The user interface on the iPhone does not include a mouse, so certain behaviors must change.

Key among these changes is hovering and clicking behaviors.

Desktop web application that supports a JavaScript menu.

JavaScript

- The scripting language supported by all major browsers is JavaScript.
- JavaScript is entirely client-side: all the code is run on the browser's computer, not on the server.
- JavaScript functions can be included from external files, stored in the document's header element, and inline with the HTML.
- JavaScript is also rather powerful, and is the foundation for a variety of other technologies that power Web 2.0.

i)DHTML

- DHTML uses JavaScript to modify page elements dynamically.
- JavaScript functions modify either the styles or the HTML elements.
- The most common is show/hide, which allows you to hide content using the CSS display: none; property;
- DHTML technique is incredibly useful when designing iPhone web apps,

```
<script type="text/javascript">
<!--
function toggle_visibility(id) {
var e = document.getElementById(id);
if(e.style.display == 'block')
e.style.display = 'none';
else
e.style.display = 'block';
}
//-->
```

ii) Ajax

- Asynchronous JavaScript and XML, is called Ajax .
- Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications.
- With Ajax, web applications can send and retrieve data from a server asynchronously without interfering with the display and behaviour of the existing page.
- Ajax has three parts: the data sent to the server, the function that is supposed to be performed on the data in the form of a URL, and a request for a response.
- The data can be sent to the server in a variety of formats, including plain text, XML, and JavaScript Object Notation (or JSON).
- The request for a response usually includes the name of a JavaScript function in the web page to be called, conveniently referred to as the ***callback function***.

iii)Multi touch

- The iPhone, as of OS 2.0, includes the ability to recognize and handle both multi touch events .
- Mobile Safari with many of the same user interface abilities that you might find in the native API.
- In this example, we combine multi touch events with visual effects to enable dragging.

iv)Fixed Footer

The iPhone doesn't support the ability to display fixed position content at the bottom of the viewport. Richard Herrera created a JavaScript file.

MatteoSpinelli created another version of a fixed footer scroller that is a bit smaller in file size and performs

A **footer** is the information at the bottom of a webpage. ... However, with a **sticky footer** (sometimes known as a **fixed footer**) that information is always present at the bottom of the visitor's web browser as the visitor scrolls down. They do not have to get to the bottom of the page to see it.



Creating a Mobile Web App

- iPhone, was built for mobile browsers, it was simply referred to as a mobile website, or as web content designed to be viewed within a web browser.
- Few mobile browsers support the complex interactions that are often associated with web applications, or application-like experiences using web technologies.

Mobile web applications designed specifically for the iPhone

i) Defining the Viewport

- The *viewport* is the area within a browser where content can be seen by the user.
- On the desktop, the user can resize the browser window and therefore the viewport.
- In mobile devices, the browser area cannot be resized;
- Most Class A browsers—including WebKit and Mobile Safari—create a virtual viewport area, adjusting the content to fit within the screen.
- Default, Mobile Safari assumes a viewport of 980 pixels, the recommended size for desktop sites.

ii)Full-Screen Mode

- To run a mobile web application in full-screen mode without the default Safari user interface, or “browser chrome.”
- This feature can be a great way to leverage web apps to create a native-app-like experience;
- The web app needs to use DHTML or Ajax to display content to users
- When running in full-screen mode, any hyperlinks will load the full instance of Safari, breaking the desired effect.

iii) Changing the Status Bar Appearance

The **status bar** is the area at the bottom of the **Word** window that indicates information about the current document.

- Using `an apple-mobile-web-app-capable`, you can optionally define the status bar style when running web apps in full-screen mode in Mobile Safari.
- In default, black, and black-translucent, which removes the status bar from the page flow, meaning that your content starts at the top of the page, not directly beneath the status bar:

Web Apps As Native Apps

Web apps are not real **applications**; they are websites that, in many ways, look and feel like **native applications**, but are not implemented as such. They are run by a browser and typically written in HTML.

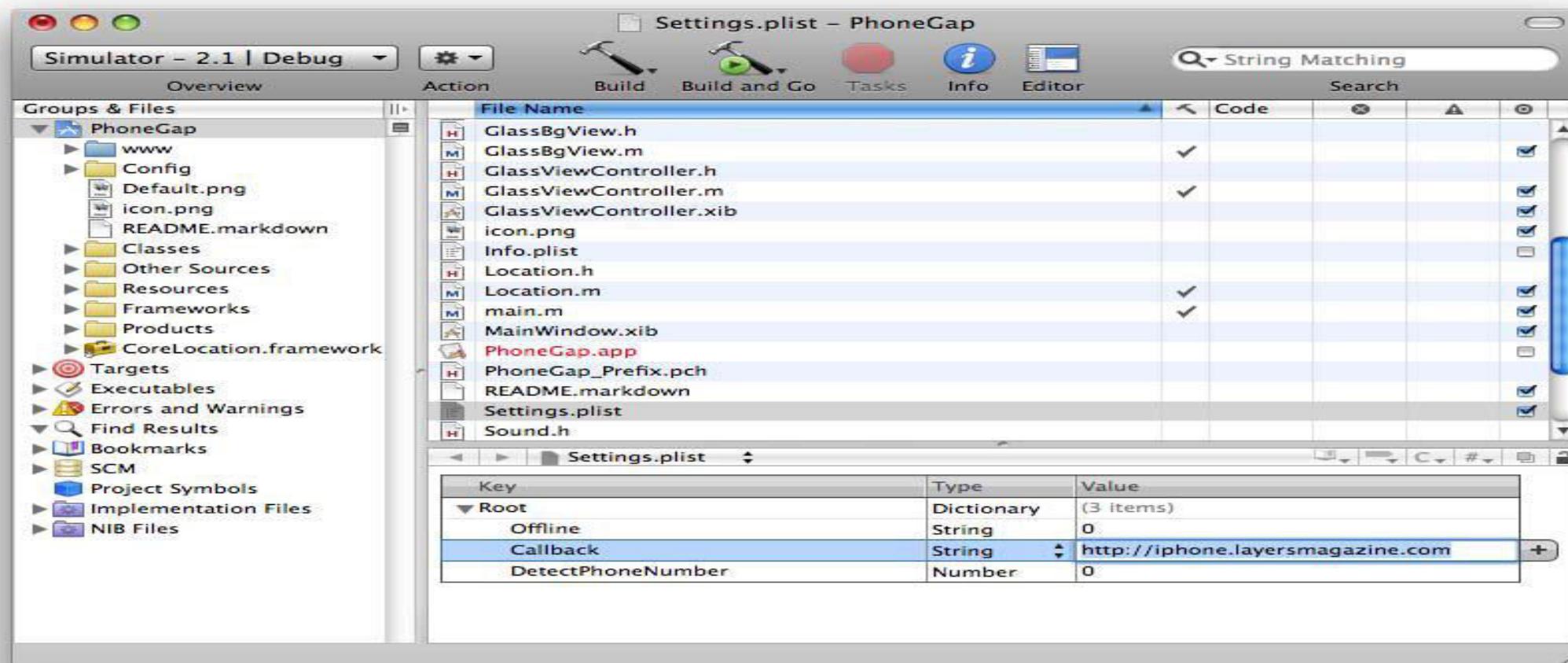
Creating applications for multiple platforms is extremely difficult and costly to do. Making an application for a single platform is not a strategy; it is an opportunity.

- Create mobile web apps that work on multiple mobile devices Many applications need to display web content in windows
- Some applications are full-featured browsers, but more often applications embed web content as a convenience, as in a custom document system.

PhoneGap

PhoneGap is a software development framework by Adobe System, which is used to develop mobile applications. To develop apps using **PhoneGap**

- A great tool for building native apps from web apps is PhoneGap.
- An open source library that enables you to take a mobile web app and compile it into a native app for the iPhone, Android, BlackBerry, and other platforms.
- For example, to create an iPhone application, open up the *phonegap.xcodeproj* in Apple's Xcode
- Take an existing mobile web application designed for most Class A browsers and build a native application in minutes.



Tools and Libraries

Many toolkits and interface libraries have emerged to aid in the creation of mobile web apps, specifically for the iPhone.

i)iPhone GUI

- creating iPhone web apps is having to recreate the iPhone user interface in HTML and CSS.
- complete withall of the iPhone GUI elements needed to create an iPhone-inspired web app, or to be used as the foundation for your own UI.



ii)iUI

iUI is a framework consisting of a JavaScript library, CSS, and images for developing advanced mobile webapps for **iPhone** and comparable/compatible devices.

- During the first iPhoneDevCamp, developer Joe Hewitt created an open source user interface library that mimics the appearance and interactions of the iPhone.
- the project is now known as iUI and is one of the more popular tools used for creating iPhone web apps.
- iUI uses CSS and JavaScript to quickly create menus, animated page transitions, and other effects that make your web app feel at home on the iPhone.
- A page weight of only 30 KB, making it an ideal toolkit for mobile devices.

iii) jQTouch

- Another interface library called jQTouch is designed to include other WebKit browsers based on the popular JavaScript framework jQuery
- It was created by designer David Kaneda, out of the need for a lightweight skinnable interface library for more than just the iPhone.
- jQTouch supports iPhone-style menus, animated page transitions, and effects using CSS transforms.
- It is four times larger than iUI—around 112 kilobytes, half of which is just the core jQuery library.

Previous Topic.....

UNIT –V

Supporting Devices: Having a Device Plan – Device Testing – Desktop Testing – Usability Testing.

Today discuss about.....

Supporting Devices:

- Having a Device Plan
- Device Testing

SUPPORTING DEVICES:

- Maintaining quality and consistency Mobile design and development requires that Flexibility creativity to find the right solution for your resources

Having a Device Plan:

- Testing cycle will be the Web and consistent support for web standards.
- Multiple digital landscapes, including phones, desktops ,always-on devices, the social web.
- A device plan that used a class-based matrix Class A devices are defined as the most advanced web-rendering capabilities.
- Class F devices offer a limited experience, and a progressive enhancement technique creates experiences that degrade.

i)Deciding What to Support

- Two methods for determining the support strategy at the beginning of a mobile project.
- The first is to look at your server logs, seeing which devices currently access your site.
- The second is to use the niche nature of how devices are marketed to consumers.

ii)Example Device Plans

- Creating a device plan is as simple as understanding profit and loss.
- Devices will cost more to develop and test for and certain devices will generate more traffic or downloads.
- A simple numbered scoring system to indicate the possible magnitude.
- A cost score of 5 might be five times more complex, taking more time or resources.

An example device plan with a numbered score system

	Estimated cost	Estimated revenue	Predicted profit
• Class A devices	1	5	4
• Class B devices	2	4	2
• Class C devices	3	3	0
• Class D devices	4	2	-2
• Class F devices	5	1	-4

In this simple example, the Class A device is the easiest to support and Class F is the hardest, whereas the estimated revenue is the inverse, with the Class A providing more of an impact and Class F less of

i) A mobile website

- A mobile website has scoring system.
- *An example mobile website device plan*

	Estimated cost	Estimated revenue	Predicted profit
• Class A devices	1	4	3
• Class B devices	1	3	2
• Class C devices	3	2	1
• Class D devices	4	2	-2
• Class F devices	5	0	-5
• Class A and B devices and testing them throughout the design and development process.			
• Class C devices, I would use progressive enhancement.			
• Class D and F would officially be unsupported			

ii) A mobile web app

- To build a mobile web app, and use actual dollar amounts to determine the plan.
- Class A has a far great margin than the other classes.
- *An example mobile web app device plan*

	Estimated cost	Estimated revenue	Predicted profit
• Class A devices	\$20,000	\$50,000	\$30,000
• Class B devices	\$35,000	\$20,000	-\$15,000
• Class C devices	\$45,000	\$2,000	-\$43,000
• Class D devices	\$50,000	\$0	-\$50,000
• Class F devices	\$60,000	\$0	-\$60,000

- The Class A device is a sure thing. You know you can see a decent profit from the effort—more than covering costs.
- Class B devices are nearly twice as much to support than Class A and have the potential to produce less revenue to support both Class A and Class B devices would result in a loss, making less money than it would cost to support in the first place.

iii) A mobile commerce portal

- To create a mobile commerce portal—a much more complex effort that likely needs more support for more devices, as it wouldn't be fair to users to support just the best devices..
- The revenue comes from the middle, more widely adopted devices, not just the high end
- *An example mobile commerce device plan*

	Estimated cost	Estimated revenue	Predicted profit
• Class A devices	\$20,000	\$125,000	\$105,000
• Class B devices	\$35,000	\$300,000	\$265,000
• Class C devices	\$45,000	\$200,000	\$155,000
• Class D devices	\$50,000	\$95,000	\$45,000
• Class F devices	\$60,000	\$15,000	-\$45,000

- The Class B and Class C devices produce the greatest amounts of profit more expensive to support than Class A devices.

DEVICE TESTING:

- Many devices are subsidized through the operator.

1) Access to Devices

- Access to multiple devices is a challenge for every mobile design and developer.
- Take an application that uses the accelerometer, like being able to shake the device to load a new page like the iPhone application Urban Spoon



With Urban Spoon you shake to produce a result—something that is hard to test for without an actual device

i) Guerrilla testing

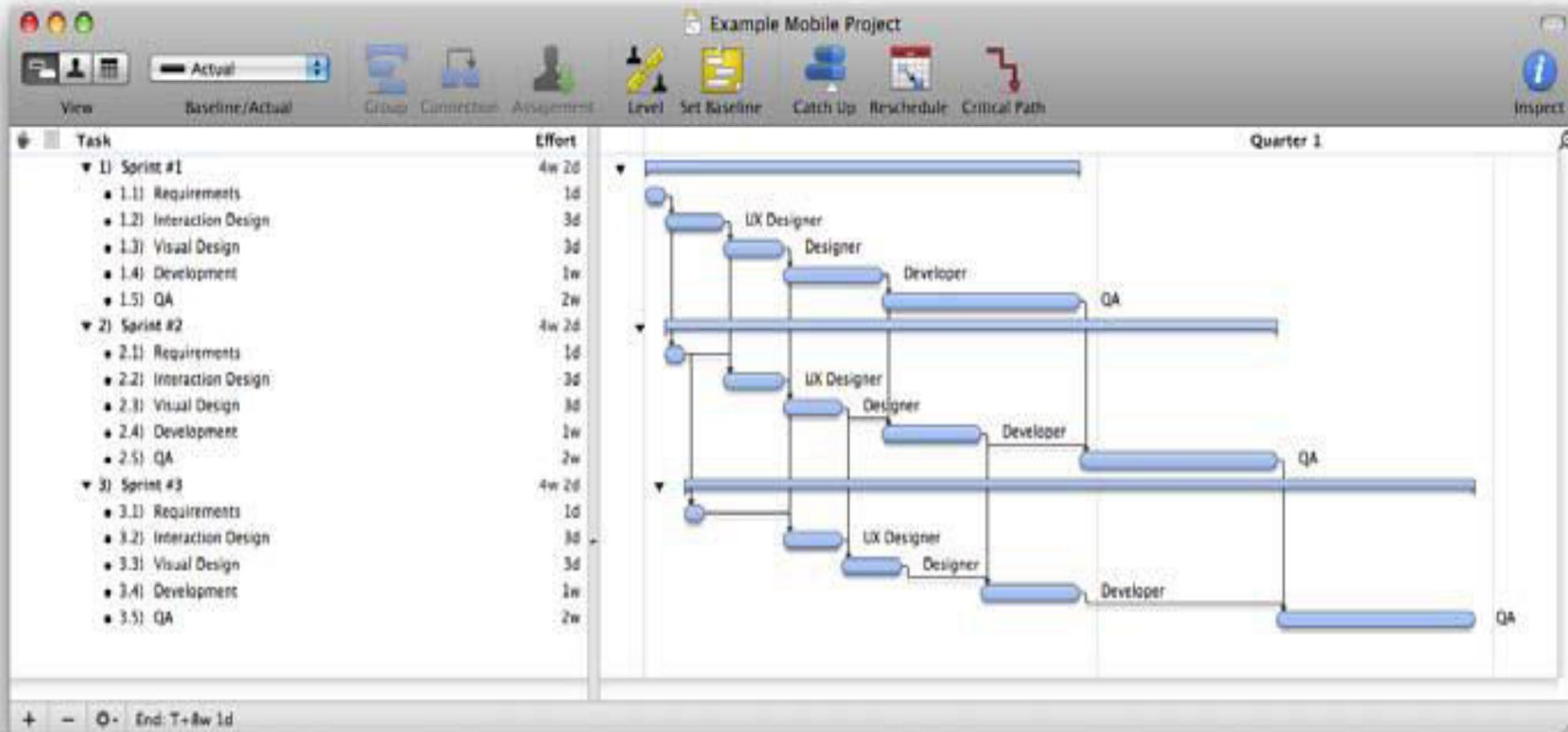
- The guerrilla testing technique is known to recommend in mobile web workshops is to go down to the operator store and test your work on the devices in the store.
- During design or development, certainly we can confirm how it works on several devices that you can hold in your hand all at once..

ii) Mobile Monday device libraries

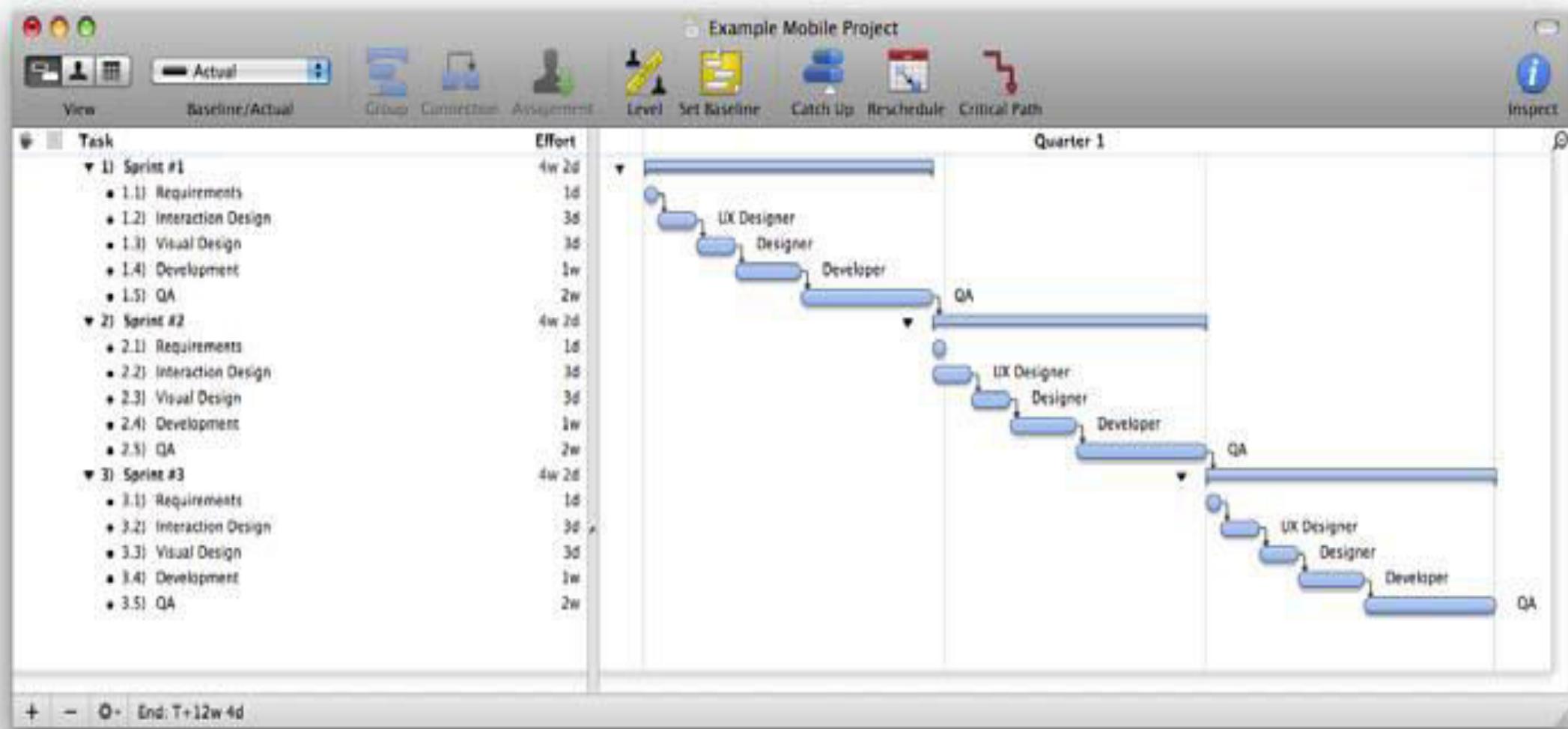
- Create a spreadsheet of who has what devices on what network, then offer to share (and track) them amongst the group, like a library, checking out devices for a day, week, or month.
- Instead of one person buying dozens of devices, you can coordinate with each other so that everyone buys one or two devices and shares them among the group.

2) Estimating the Testing Effort

- When developing a single mobile product or supporting a single device class, an extended testing effort can simply result in a delayed release.
- Supporting multiple classes, the testing effort can hold up resources from moving on to the next phase of development.
- A sprint is a unit of work from the scrum-based methodology that usually lasts anywhere from a week to a month.
- Primary devices in the first sprint, release it, then after a period of time move on to the second class in a second sprint.
- It frees up your resources and doesn't get them bogged down in development hell during QA.



An example mobile project plan, showing how lengthy QA cycles can delay phases



An example mobile project plan, separating support into multiple sprints

2)Creating a Test Plan

- Creating a test plan for mobile devices means testing for every possible problem that the phone might encounter and that might result in it failing to present the desired content to the user.
- A larger number of devices and you can start to see that this can take some time.

Functional tests

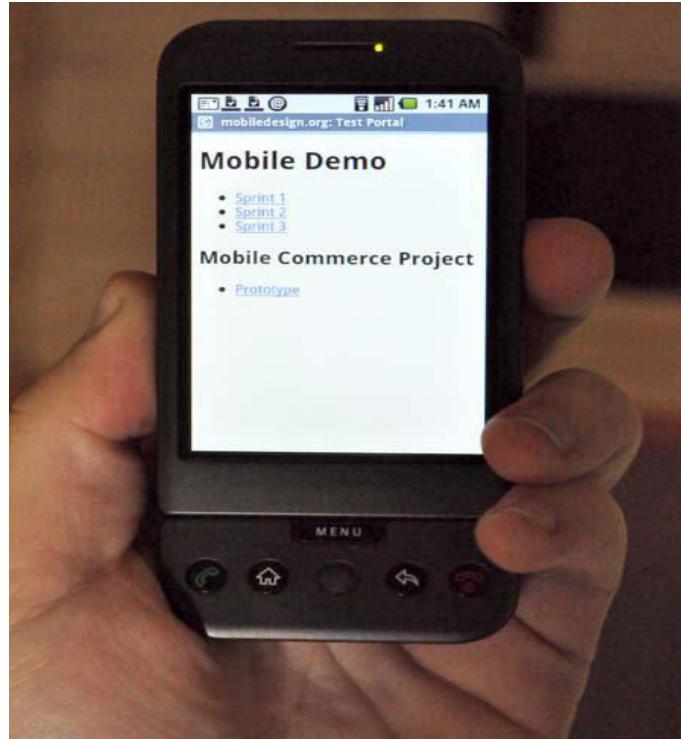
- The first step is creating a test plan based on your feature list.
- All the features of your application and test them one by one on an actual device.
- For mobile products, more detailed feedback and finding more bugs earlier on greatly reduces the time and efforts needed to get the product finished and released.

Context tests

- After your features are tested, and you know your application is functionally sound, it is time to move on to testing the mobile context.
- Questions to ask about your product during context tests include:
 - How does the user experience render on the device? Are there visible issues on the test device? If so, explain them in detail.
 - Does it load quickly? Does it load correctly?
 - Can you use the physical features of the device as they are intended? Do the softkeys work correctly?
 - Does it terminate correctly?

Creating a Test Portal

- A simple but effective trick for mobile web device testing is creating a test portala web page with a list of links on it to all your web pages.
- This provides easy access to all your development servers that you needto test.
- Typing URLs into multiple devices is time-consuming, especially development URLs, which can be long.
- Creating a test portal lets you add URLs for each new product that needs to be tested.
- bookmark that page on each test device, giving you quick access with a fraction of the typing.



An example test portal

- ★ Another helpful tool is a URL shortener, like [TinyURL](#) or [bitly](#), which shorten long URLs to make them mobile-friendly.
- ★ For example, you can make a URL like *http://m.domain.com/testapp/* map to a longer development URL like *http://dev.domain.com/testapp/session?id=12345*.

QUERIES?

Previous Topic.....

Supporting Devices:

- Having a Device Plan
- Device Testing

Today discuss about.....

- Desktop Testing
- Usability Testing.

Desktop Testing:

- A big advantage of mobile web products is that you can do the majority of your testing from your desktop before ever getting it on a device.
- Desktop testing reduces the time span between developing a feature, testing a feature, and fixing a feature, and ultimately allows you to spend less time dealing with devices.

i) Frames

- Many web browsers have a minimum window size that is larger than your average device screen, making it impractical for testing mobile websites or web apps in a desktop browser. This is called *inlineframesets*, or *iframes*,

Create a web page with an iframe,

```
<iframe src="mobile/index.html" width="240px" height="320px"  
style="border:1px solid;"></iframe>
```

Multiple screen dimensions from a single page.

ii) Opera

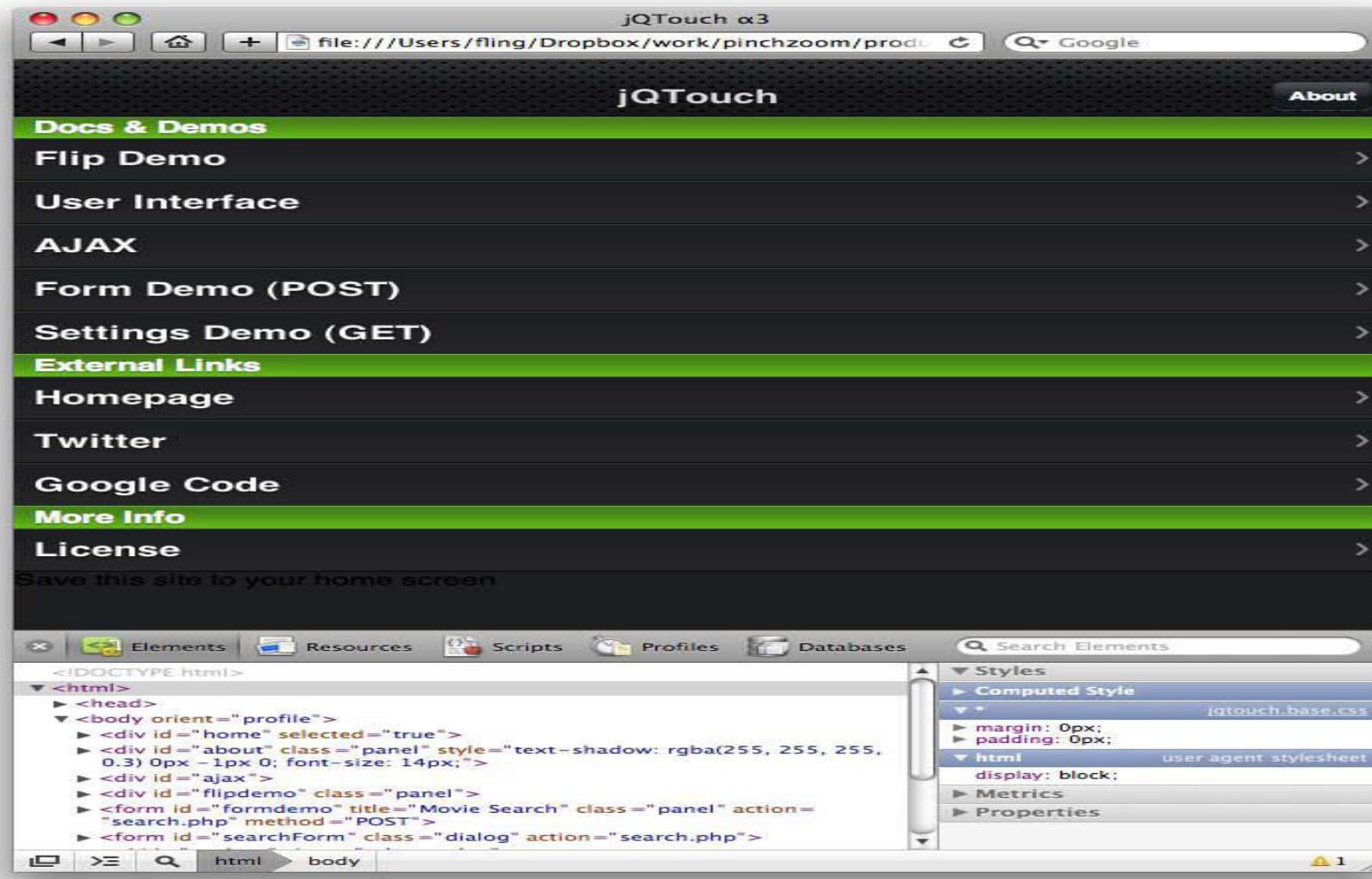
- The Opera desktop browser has a Small Screen view. that mimics a mobile screen, loading the handheld media type if available and presenting the page in a narrower format.
- Opera's Small Screen view is a great way to see how your work might look on a mobileDevice.
- Toggle the Small Screen view by going to View→Small Screen in Opera.

- Desktop testing using Opera's Small Screen view



iii) WebKit

- The WebKit browser engine can also be used for desktop testing for WebKit-based mobile applications.
- You can download the nightly builds of WebKit and run them on your desktop, giving you a close representation of how it may render on the target device.
- WebKit has an excellent debugging tool, called the Web Inspector.
- It allows you to see how styles are rendered, offline storage information, the page weight and the estimated time it takes to load resources.
- It is helpful for fine-tuning the mobile web projects.



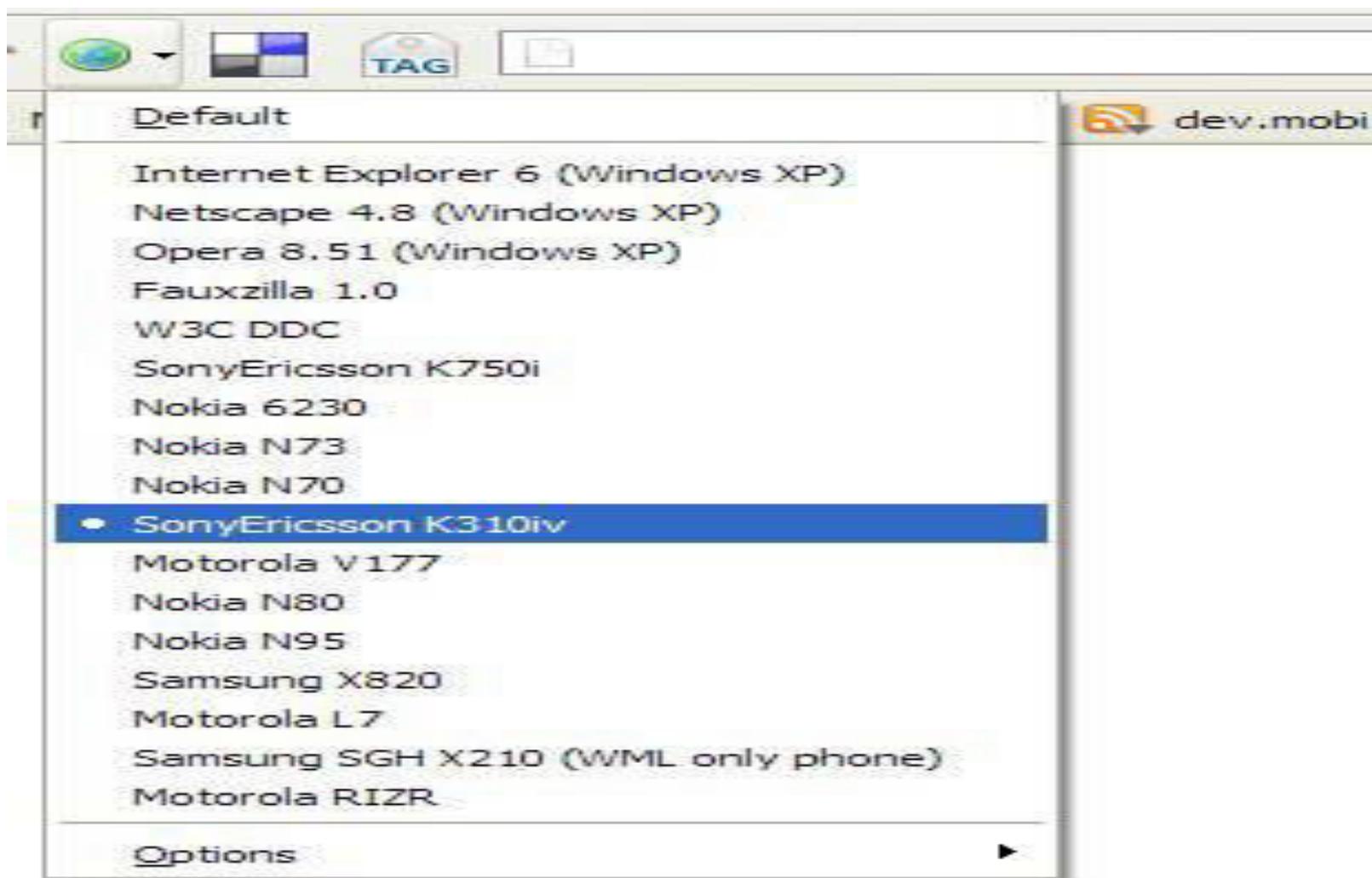
Desktop testing with the WebKit browser

vi) Firefox

- The Firefox User Agent Switcher extension lets you change the user agent information you send to the server
- Once you add the data from the supported mobile user agents, you can test how each of your targeted sites renders for the requesting device.
- Few mobile browsers are based on Mozilla's Gecko rendering engine, this form of desktop testing isn't so much for testing how content renders, but is instead for your device detection and content adaptation functional tests.
- The Web Developers Toolbar and Firebug Firefox extensions to provide XHTML and CSS debugging, similar to the WebKit Web Inspector.

v)Collecting User Agents

- Test detection and rendering of multiple devices on the desktop means having valid user agents
- Collecting user agents is the open source WURFL database, which contains a large number of user agents submitted by the community and from device makers.
- Each of those browsers likely has a unique user agent. Searching the WURFL database shows 100 devices and therefore 100 user agents for devices called the “RAZR”



Using the Firefox User Agent Switcher to test device detection methods

vi) Simulators and Emulators

- Every mobile framework comes with an emulator—the hardware in your computer is different than the hardware on the device, it has to run in an emulated environment that allows you to test your work in a virtual environment.
- It causes inconsistencies between the emulated environment and the real environment.
- Every new device that is released doesn't necessarily get its own emulator.
- We use the RAZR as our example, emulators are only available for a small handful of the total number of RAZRs.
- Emulators are often out of sync with the actual devices, mobile web products, you can usually get just as accurate and far faster test cycles using the iPhone Simulator (Figure 15-11) that comes with the free iPhone SDK. It is an accurate representation of the iPhone environment. Using desktop browsers as your first-pass and hands-on device testing for your second pass.



The dotMobi emulator

- iPhone and iPod touch run on Mac OS X, the simulator is able to run the iPhone OS natively on your desktop, not virtually, as long as your desktop is running Mac OS X.
- The iPhone Simulator also runs the same version of WebKit iPhone Simulator is a far more accurate means of testing
- mobile web products for the iPhone than simply using the desktop version of WebKit.

vii) Remote Access

- Remote access services let you remotely control an actual device through your desktop, but few exist; DeviceAnywhere is one.
- Remote access provides a compelling method for testing, with the added convenience of a desktop emulator it showing the displayed characteristics of the actual device.
- Many devices and accessing devices in other countries on their native networks.



Device Anywhere

Usability Testing:

- The goal of a usability test is to identify actionable faults in the system from subjective behavior and opinion.
- In mobile development, the challenges of the device can add additional subjectivity.
- The goal is simply to get the experience in front of real users in some sort of analog of the target context
- Alternate forms of usability testing, such remote testing, don't usually work well.
- Café testing—get casual unscheduled feedback from participants at a nearby café—works great, given the casual nature of the locale.

Mobile Usability Test Tips and Tricks

- *Know your users*
- Knowing your target users before you start usability testing helps to draw them out
- while you perform the test

Make no assumptions about your participants

- There are no right or wrong answers in a usability test, and there are no right or wrong participants. Your job is to listen, watch, and learn from your participants.

Test early

- Usability testing just at the end of the project does no one any good.
- A user verification and feedback test after each phase of design and development.

Go to the user; don't have them come to you

- To perform usability tests in the participants' home, office, or some sort of neutral location. You need to test mobile products in the wild, not in the lab.

Record everything

- Small gooseneck digital video cameras can be positioned to capture the device screen as well as the users' reactions

Have someone else record results

- Ttaker—a third person whose job is to record everything during the test the notes and extra perspective will come in handy later when reviewing the test material.

Test often

- Usability testing is not a one-time event; it is something you need to integrate into your process. Plan to test as often as you possibly can.

Keep it simple

- Usability testing is more art than science.
- It is easy to turn the simple act of talking to your users into an insurmountable chore

QUERIES?