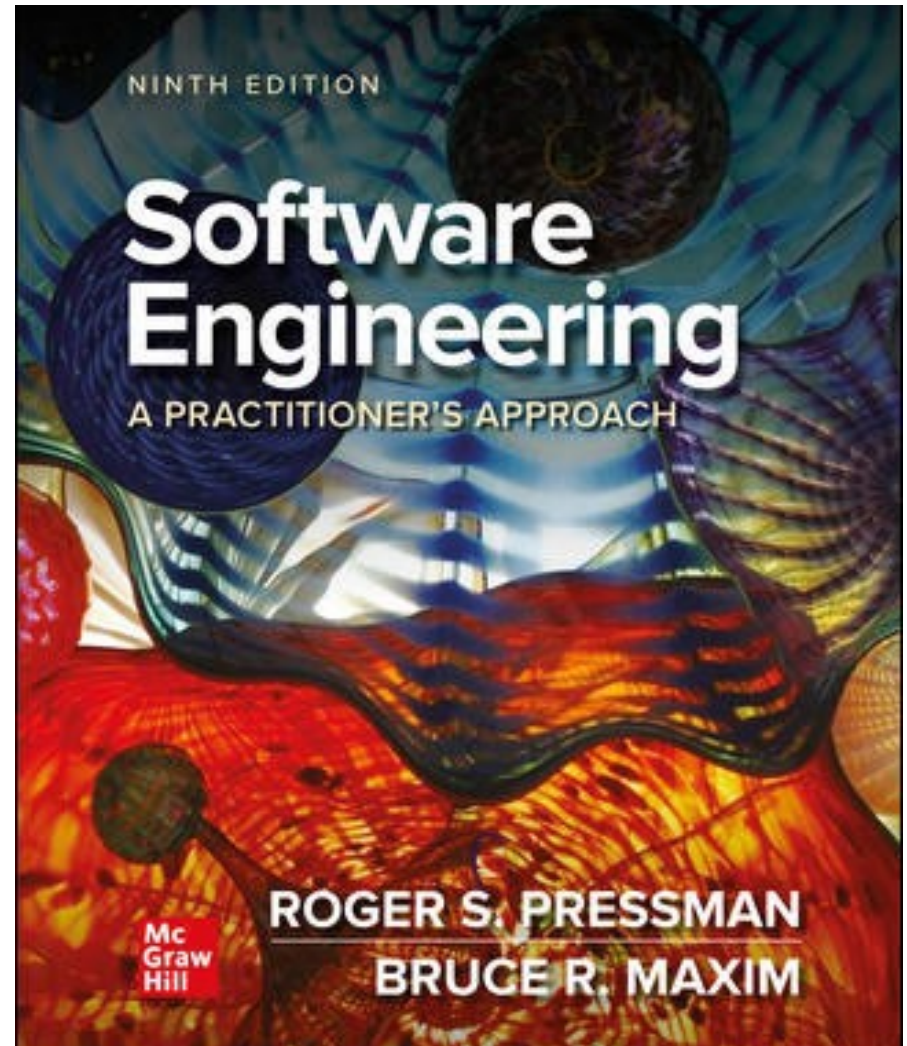


Chapter 13

Design for Mobility.

Part Two - Mobility.



Mobile Development Challenges

- Mobile devices have many features in common with each other, but often provide very different user experiences.
- Some users expect the same features provided on their laptops.
- Other users like freedom that portable devices give them and accept reduced functionality in a mobile software product version.
- Some users expect unique experiences not possible on traditional computing or entertainment devices.
- The user's perception of "goodness" might be more important than any of the technical quality dimensions of the mobile product itself.
- Software engineers should craft a user experience that takes advantage of device characteristics and context-aware applications.
- Testing mobile software products provides additional challenges.

Mobile Technical Considerations

- Multiple hardware and software platforms.
- Many development frameworks and programming languages.
- Many app stores with different rules and tools.
- Very short development cycles.
- User interface limitations and complexities of interaction with sensors and cameras.
- Effective use of context.
- Power management.
- Security and privacy models and policies.
- Computational and storage limitations.
- Applications that depend on external services.
- Testing complexity.

Mobile Development Life Cycle

Inception. Goals, features, and functions of the mobile product are identified to determine the scope and the size of the first increment.

Design. Developers define the app user experience using screen mockups and paper prototypes to help a user interface design that will take different screen sizes and capabilities into account.

Development. Mobile software is coded, test cases are created. Usability and accessibility tests conducted as the product evolves.

Stabilization. Most mobile products go through a series of prototypes: feasibility (one logic path); alpha prototype (minimum viable product); beta prototype (largely complete); and release candidate (all required functionality) ready for product owner review.

Deployment. Once stabilized a mobile product is reviewed by a commercial app store and made available for sale and download.

Useful Mobile User Interface Design Models

- A *platform model* describes the constraints imposed by each platform to be supported.
- A *presentation model* describes the appearance of the user interface.
- The *task model* is a structured representation of the tasks a user needs to perform to meet her task goals.

Mobile User Interface Design Considerations

- Define user interface brand signatures.
- Focus the portfolio of products.
- Identify the core user stories.
- Optimize user interface flows and elements.
- Define scaling rules.
- User performance dashboard.
- Champion-dedicated user interface engineering skills.

Mobile User Interface Evaluation

In trying to meet stakeholder usability expectations, mobile developers should attempt to answer these questions to assess the out-of-the-box readiness of the device:

- Is the user interface consistent across applications?
- Is the device interoperable with different network services?
- Is the device acceptable in terms of stakeholder values in the target market area?

Mobile Design Approaches

- **Usage Scenarios.** Must consider context variables (location, user, and device) and transitions between contextual scenarios (locations and settings, movement and posture, devices and usages, workloads and distractions, user preferences).
- **Ethnographic Observation.** Used method to gather information about representative users of a software product by observing them as they use the product in a natural setting. Can be tricky to observe them without interfering with their product use.
- **Low-Fidelity Paper Prototypes (for example: cards or Post-it notes).** Cost-effective usability assessment approach in user interface design that can be used before any programming takes place. It is important for these prototypes to be similar in size, weight, appearance to allow their use in a variety of contexts.

Mobile Design Mistakes

- **Kitchen sink.** Avoid adding too many features to the app and too many widgets on the screen.
- **Inconsistency.** Set standards for page navigation, menu use, buttons, tabs, and other UI elements. Stick to uniform look and feel.
- **Overdesigning.** Remove unnecessary elements and wasteful graphics. Do not be tempted to add things without thinking.
- **Lack of speed.** Users do not care about device constraints—they want to view things quickly. Preload what you can.
- **Verbiage.** Unnecessarily long, wordy menus and screen displays
- **Nonstandard interaction.** Take advantage of the user's experience with the way things are done on the platform.
- **Help-and-FAQ-itis.** Adding online help is not the way to repair a poorly designed user interface.

Services Computing

- Focuses on architectural design and enables application development through service discovery and composition.
- Allows mobile app developers to avoid the need to integrate service source code into the client running on a mobile device.
- Runs out of the provider's server **and is loosely coupled** with applications that use it via messaging protocols.
- Provides an API (application programming interface) to allow service to be treated like an abstract black box.

Cloud Computing

Cloud architecture has three service layers

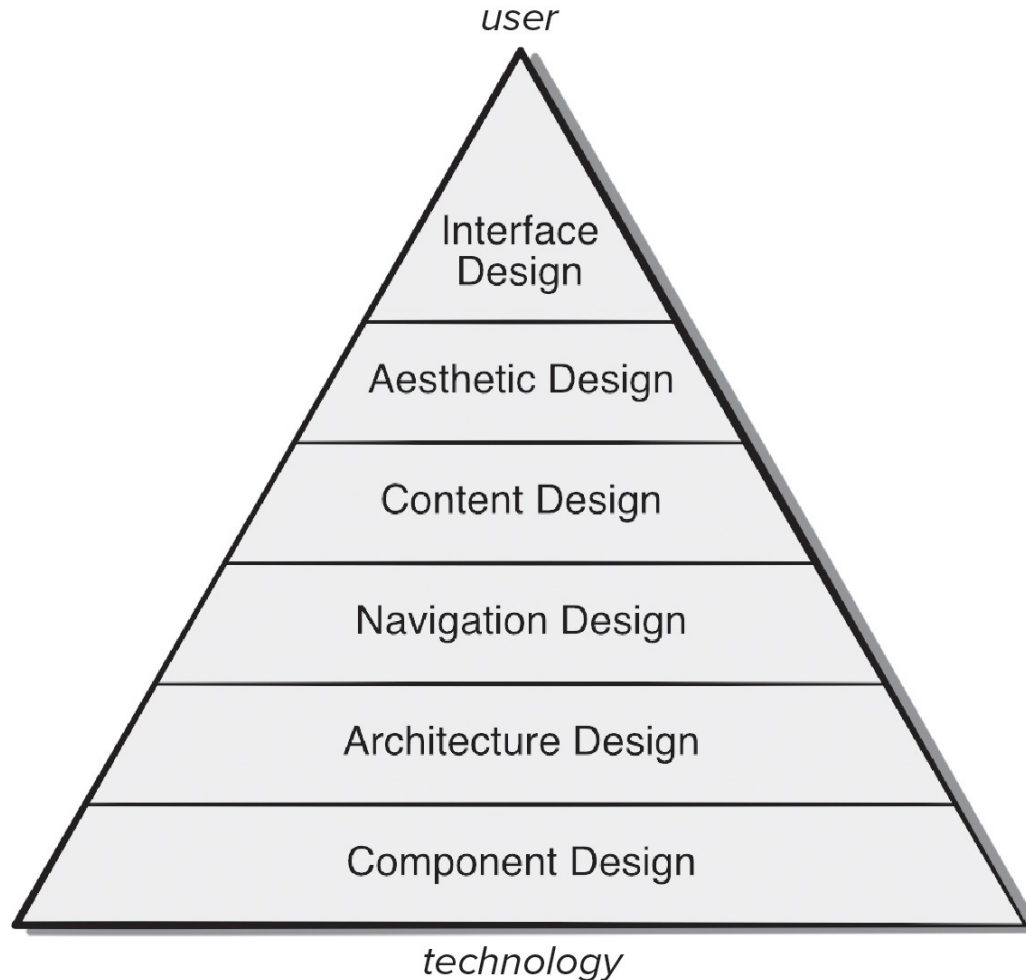
- **Software as service** layer consists of software components and applications hosted by third-party service providers.
- **Platform as service** layer provides a collaborative development platform to assist with design, implementation, and testing by geographically distributed team members.
- **Infrastructure and service** provides virtual computing resources (storage, processing power, network connectivity) on the cloud.

Context-Aware Apps

- Context allows the creation of apps based on the location of the mobile device and the device functionality to be delivered.
- Context helps tailor personal computer apps for mobile devices.
- Mobile computing merges real and virtual worlds by allowing devices to be aware of other objects and its surroundings.
- Device must detect the presence and identity of a user, as well as the attributes of the context that are relevant for that user.
- Extracting relevant context information from several sensors is challenging (for example: noise, miscalibration, wear and tear, weather).
- Event-based communication is preferable to the management of continuous streams of high-abstraction-level data in context-aware applications.

Web Design Pyramid

Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



[Access the text alternative for slide images](#)

Objectives of Web Interface Design

- Establish a consistent window into the content and functionality provided by the interface.

To achieve a consistent interface, you should use visual design to establish a coherent “look.”

- Guide the user through series of WebApp interactions.

You may draw on an appropriate metaphor that enables the user to gain an intuitive understanding of the interface.

- Organize navigation options available to the user.

It is important to note that one or more navigation mechanisms should be provided at every level of the content hierarchy.

Aesthetic Design

- Don't be afraid of white space.
- Emphasize content.
- Organize layout elements from top-left to bottom right.
- Group navigation, content, and function geographically within the page.
- Don't extend your real estate with the scrolling bar.
- Consider resolution and browser window size when designing layout.

Content Design

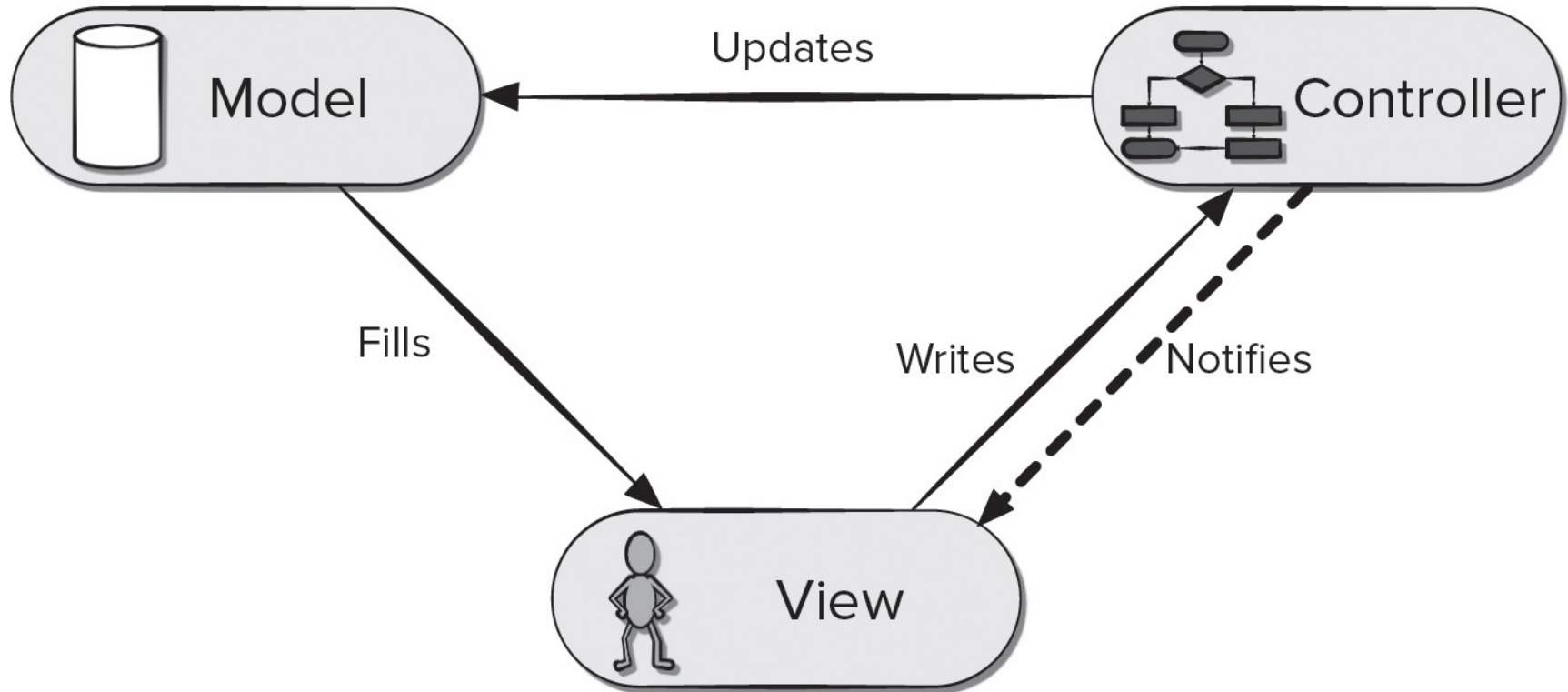
- *Content design* develops a representation for content objects.
- A *content object* has attributes that include content-specific information and implementation-specific attributes that are specified as part of design.
- For WebApps, a content object is more closely aligned with a data object for conventional software.

Architecture Design

- Architecture design is conducted in parallel with interface design, aesthetic design and content design.
- *Content architecture* focuses on the manner in which content objects (or composite objects such as Web pages) are structured for presentation and navigation.
- *Information architecture* is also used to connote structures that lead to better organization, labeling, navigation, and searching of content objects.
- *WebApp architecture* addresses ways the application is structured to manage user interaction, handle processing tasks, effect navigation, and present content.

Model View Controller (MVC)

Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



[Access the text alternative for slide images](#)

WebApp MVC Architecture

The *model* contains application specific content and processing logic:

- all content objects.
- access to external data/information sources,
- all processing functionality that are application specific.

The *view* contains all interface specific functions and enables

- presentation of content and processing logic.
- access to external data/information sources,
- all processing functionality required by the end-user.

The *controller* manages access to the model and the view and coordinates the flow of data between them.

Navigation Design

Begins with a consideration of the user hierarchy and related use-cases

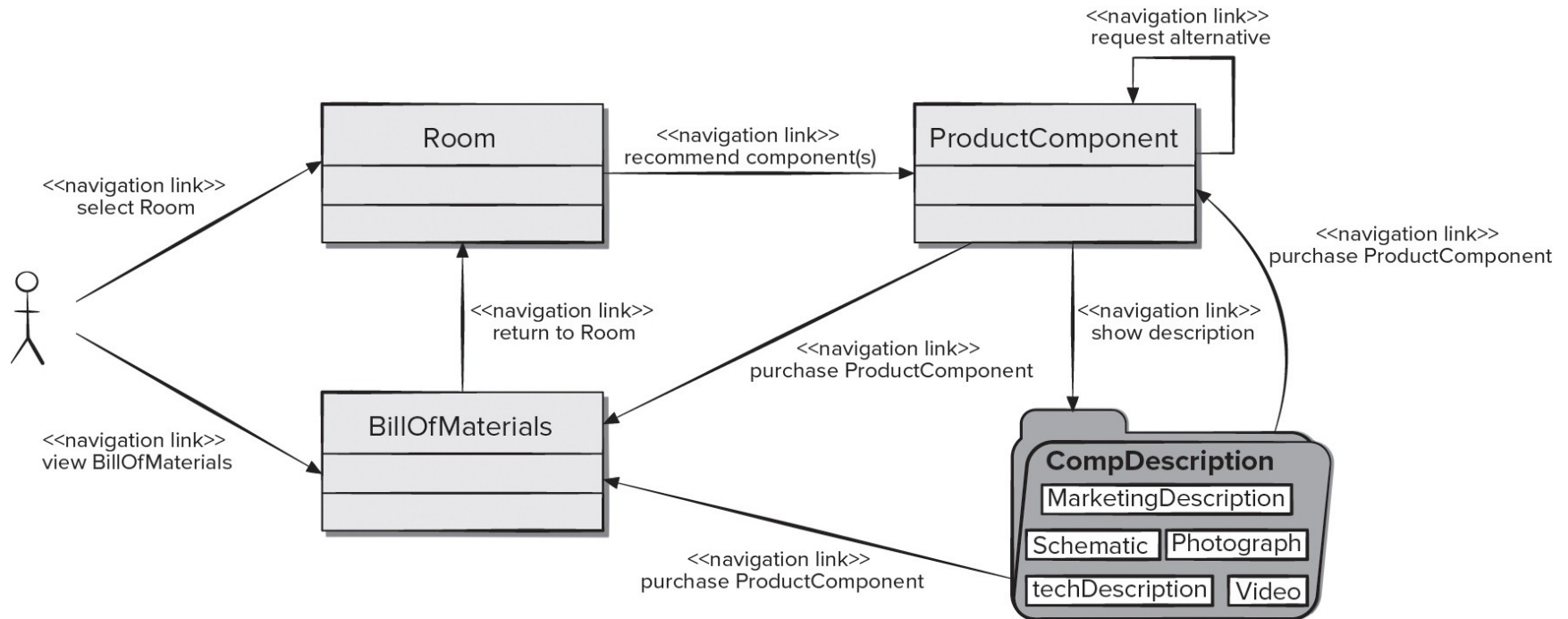
- Each actor may use the WebApp somewhat differently and therefore have different navigation requirements.

As each user interacts with the WebApp, they encounter a series of *navigation semantic units* (NSUs)

- NSU—“a set of information and related navigation structures that collaborate in the fulfillment of a subset of related user requirements”.

Creating NSU

Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



[Access the text alternative for slide images](#)

Mobile Component Design

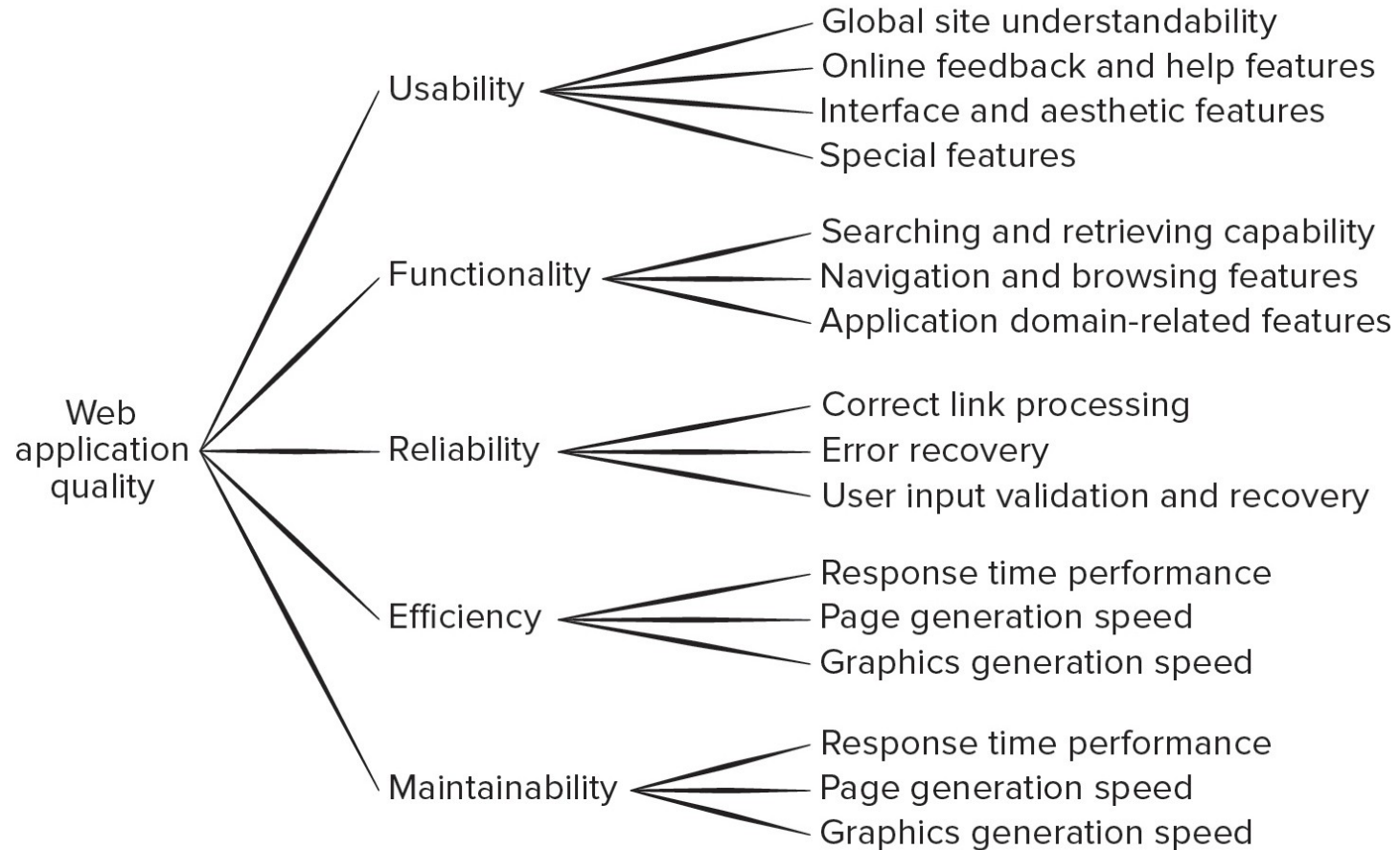
Mobile apps deliver sophisticated processing functions that:

1. Perform localized processing to generate content and navigation capability in a dynamic fashion.
2. Provide computation or data processing capability that are appropriate for the app's business domain.
3. Provide sophisticated database query and access, and
4. Establish data interfaces with external corporate systems.

To achieve these (and many other) capabilities, you must design and construct program components that are identical in form to software components for traditional software.

Mobility and Web Quality

Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



Source: Olsina, Luis, Lafuente, Guillermo and Rossi, Gustavo, "Specifying Quality Characteristics and Attributes for Web Sites," Proceedings of the 1st International Conference on Software Engineering Workshop on Web Engineering, ACM, Los Angeles, May 1999.

[Access the text alternative for slide images](#)

Mobility and Design Quality ¹

Security

- Rebuff external attacks.
- Exclude unauthorized access.
- Ensure the privacy of users/customers.

Availability

- Percentage of time that an is available for use.

Scalability

- Can the app and the systems with which it is interfaced handle significant variation in user or transaction volume.

Mobility and Design Quality ²

Time to Market

- Measure of quality from a business point of view.
- First mobile product to address a specific market segment often captures the most end users.

Content Quality

- Lots of competition on the web.
- How does the user assess the quality (for example: veracity, accuracy, completeness, timeliness) of the content that is presented within a mobile product?
- This is part of what data science tries to address.

Mobility Product Quality Checklist ₁

- Can content and/or function and/or navigation options be tailored to the user's preferences?
- Can content and/or functionality be customized to the bandwidth at which the user communicates? Does the app account for weak or lost signal in an acceptable manner?
- Can content and/or function and/or navigation options be made context aware according to the user's preferences?
- Has adequate consideration been given to the power availability on the target device(s)?
- Have graphics, media (audio, video), and other web or cloud services been used appropriately?

Mobility Product Quality Checklist ²

- Is the overall page design easy to read and navigate?
- Does the app take screen size differences into account?
- Does the user interface conform to the display and interaction standards adopted for the targeted mobile device(s)?
- Does the app conform to the reliability, security, and privacy expectations of its users?
- What provisions have been made to ensure app remains current?
- Has the mobile product been tested in all targeted user environments and for all targeted devices?

Mobility Design Best Practices

- Identify the audience.
- Design for context of use.
- There is a fine line between simplicity and laziness.
- Use the platform as an advantage.
- Make scrollbars and selection highlighting more salient.
- Increase discoverability of advanced functionality.
- Use clear and consistent labels.
- Clever icons should never be developed at the expense of user Understanding.
- Support user expectations for personalization.
- Long scrolling forms trump multiple screens on mobile devices.

End of Main Content



Because learning changes everything.®

www.mheducation.com