Human Computer Interaction

Wireframes, Mockups, and Prototypes

Prof. Andrew D. Bagdanov

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze
andrew.bagdanov AT unifi.it

December 15, 2017

Outline

Overview

Overview

Latest and Greatest

- In the first part of this lecture we will have our final Latest and Greatest session.
- The article we will discuss gives a glimpse of a critical problem facing contemporary HCI.

Prototypes and mockups

- In this lesson I will introduce some tools for making mockups and prototypes of our interfaces.
- First, I will talk a bit about some of the motivations for using low- or medium-fidelity technologies for prototyping.
- Then I will give a brief overview of some tools available.
- I will also briefly talk about GUI generation tools.

Wireframes, Mockups,

Wireframes and mockups?

- Wireframes and mockups are two ways of communicating ideas about user interfaces:
  - A wireframe is about functionality. It is usually a simple sketch demonstrating the possible actions in your interface. A wireframe of a website will show the navigation, the main buttons, the columns, the placing of different elements. It is like a bluprint for an interface.
  - A mockup is a realistic representation of the final interface. It is sometimes preferable to draw mockups using graphic software, but sometimes a prototype is more appropriate.
  - A prototype, often confused with a mockup, is a functional representation of the final product. It should allow the user experience content and interactions with the interface and to test interactions.
- Before going more in-depth, we should ask ourselves: why use wireframes and mockups?

Why wireframes and mockups are important

- We use wireframes because they save time, though management (and many engineers) often think otherwise.
- Suggesting to invest time, effort and money in creating non-functional versions of hypothetical software can get you laughed out of a meeting.
- However, wireframing is an essential part of the design process.
- An interface design is a very abstract beast.
- It is usually expressed in an interaction vocabulary that has meaning only to the inner circle of the design team.
- A wireframe begins the first real concrete visual process for a project.
- They turn the abstract nature of a design into something real and tangible without distractions.

Why wireframes and mockups are important

- Wireframes also allow us to clarify features.
- People may not understand what you mean when you say “hero image,” or “google map integration.”
- Wireframing specific project features communicates to stakeholders how these features will function, where they will live, and how useful they might actually be.
- After wireframing we may decide to take out a feature because it just doesn’t work with the overall interface goals.
- Wireframes push usability to the forefront by showcasing control layouts, navigation, and flow.
- They force everyone to look objectively at ease of use and can point out flaws in architecture or how specific features may work

Why wireframes and mockups are important

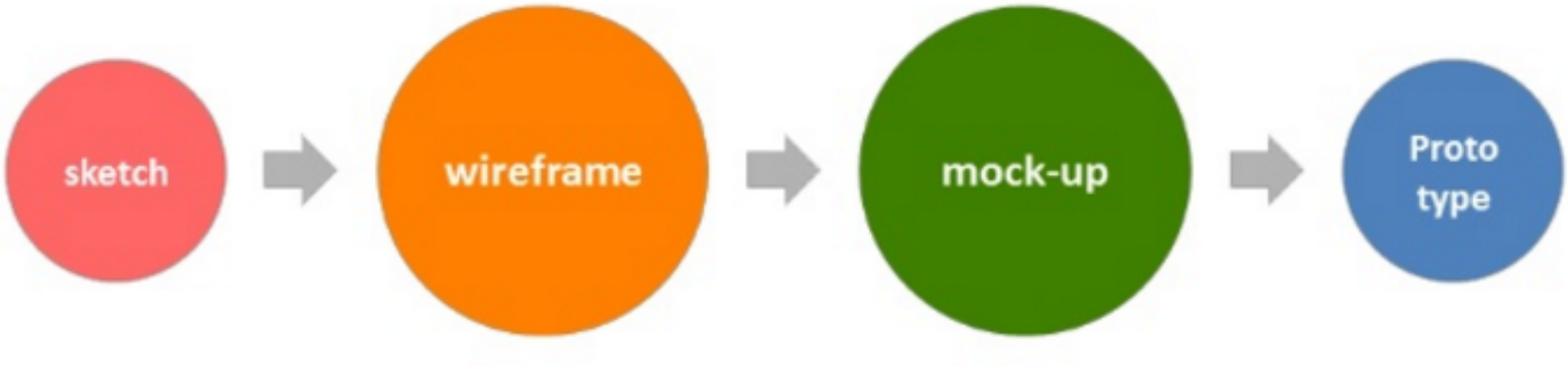
- Wireframing helps make the design process iterative.
- Instead of combining the functionality/layout and creative/branding aspects of an interface in one step, wireframes ensure that these elements are considered one at a time.
- Skipping wireframes delays this feedback and increases the costs of making changes because full design mock-ups must be reworked, not just simplified wireframes.
- Wireframing saves time in many ways:
  - Your designs are more calculated.
  - Your development team understands what they are building.
  - Content creation becomes much clearer.
  - You avoid hacks later on in the process.
- Summary: wireframes are a good way to ensure that all stakeholders are on the same page.

When to prototype

- A prototype is a simulation of the final interaction between the user and the interface.
- It might not look exactly like the final product, but should be similar.
- Interactions should be modeled with care and have a significant resemblance to the final experience.
- Interdependence between the interface and backend mechanisms is often omitted to reduce costs and speed up development cycles.
- A simulation of the final interactions is perfect for testing usability before real development actually begins.
- Prototyping can be an expensive and time-consuming form of design communication.
- When possible, create prototypes that can be reused in development.

The relationship between the three

- The three types of low-fidelity simulations of an interface have different levels of detail and effort required to construct them.
- There should be a progression like this:



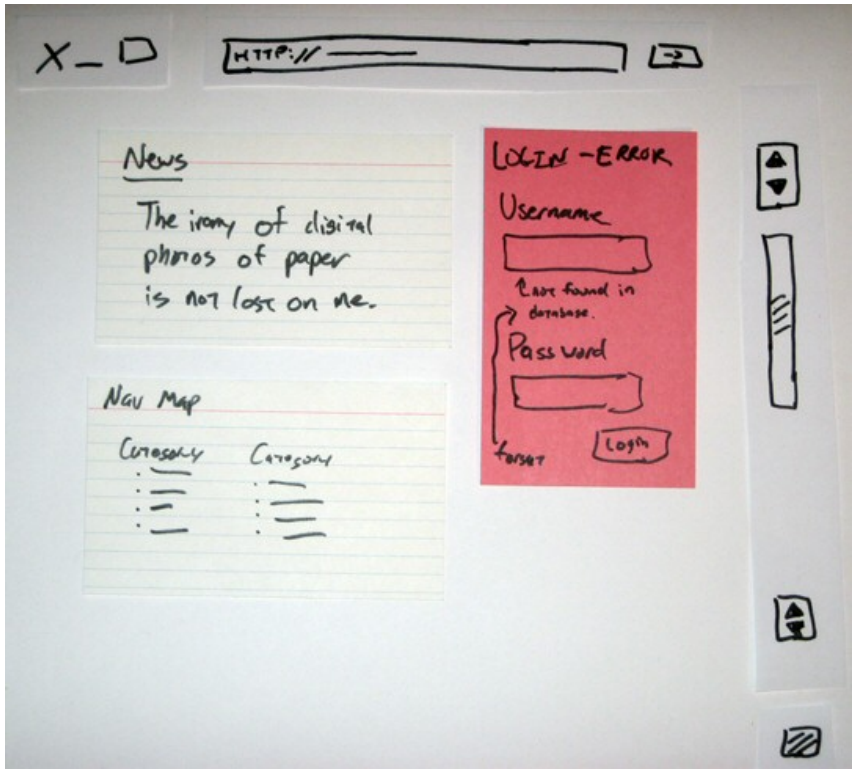
Wireframing and Mockup Tools

Wireframing and mockup tools

- Nowadays there are a myriad of options for creating wireframes and mockups.
- Here I will give a very cursory overview of some of them.

Paper prototyping

- Paper prototyping is exactly what it sounds like: a prototype (mockup, really) of a user interface is created using a combination of pencil, paper, and cutout overlays.
- Paper is an easy medium to work in, and even very simple, low-fidelity renderings can be useful tools.

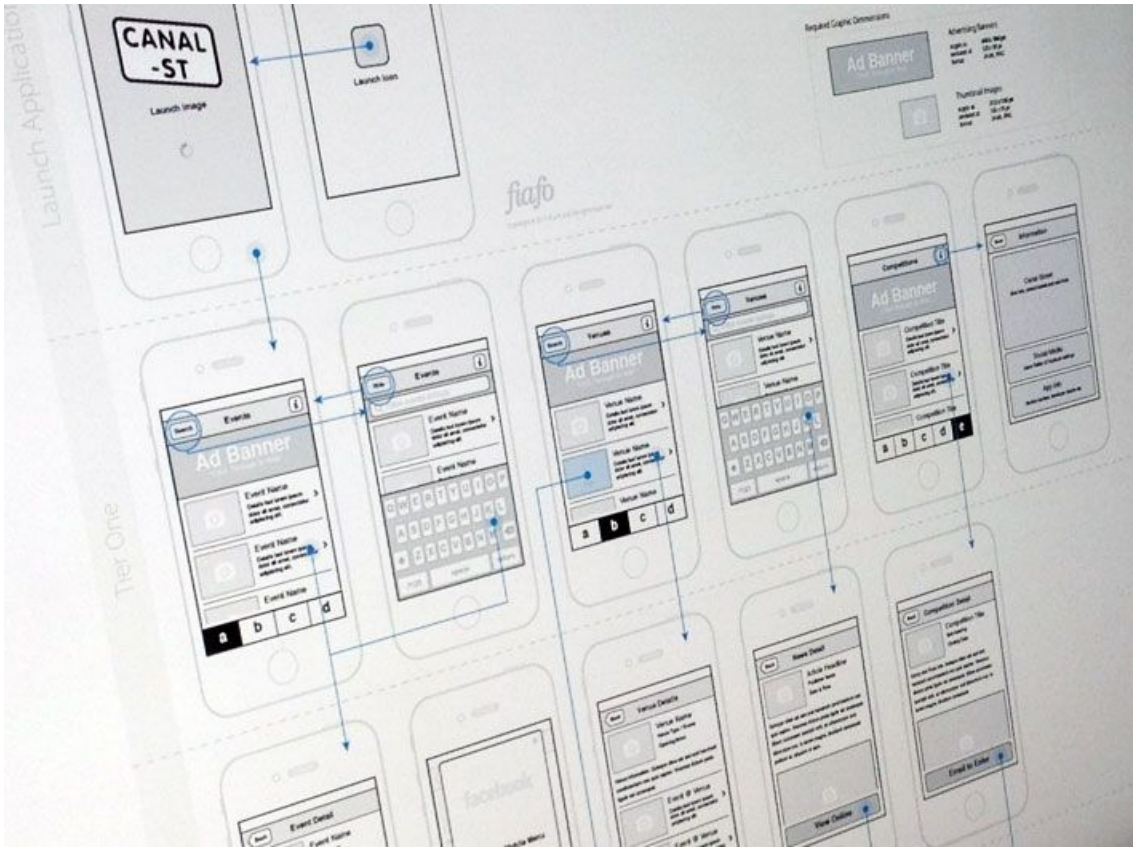


Paper prototyping

- With a bit of creativity, paper prototyping can be used to create simulations of rich interactions:

Paper prototyping

- Paper is an extremely flexible and low-cost medium that also scales in complexity of rendering:



Paper prototyping

- Here are some resources:
  - The Board of Innovation Templates for paper prototyping.
  - Free sketch templates for web, mobile and tablet platforms usink the Sketch.app tool.
  - Some paper wireframing templates from Sketch Magazine.
  - Mobile device templates for sketching mobile interfaces.

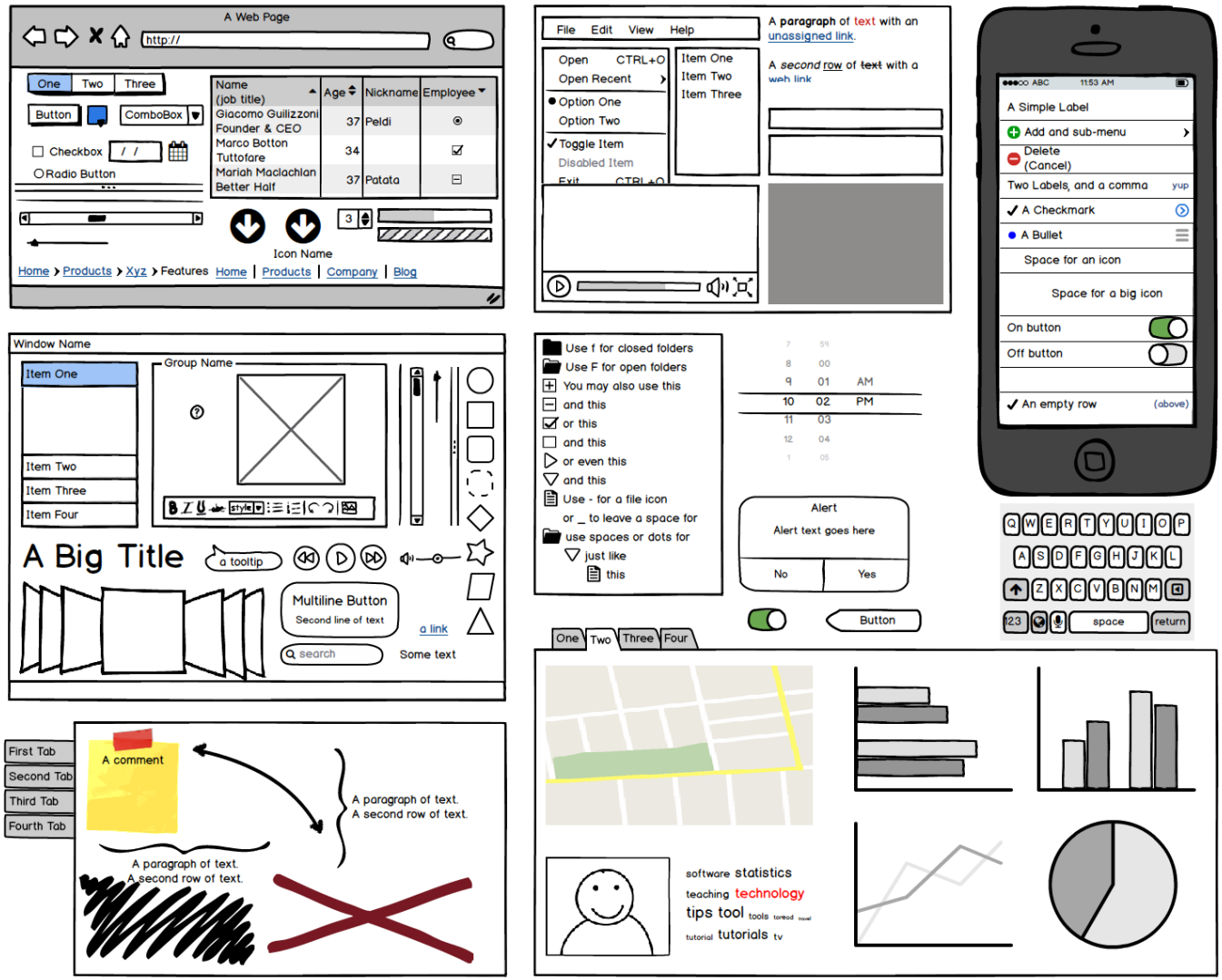
Digital wireframing and mockup tools

- The explosion of internet and mobile startups has led to a mini-bubble of applications supporting rapid wireframing and mockup.
- These applications claim to do everything for you, and to *leverage proactive synergies of your team and their designs, transitioning them into a paradigm-shifting market definer that trends positively*, and lots of other buzzwords that dumb people use to sound important.
- Because of this, I am only going to talk about two tools.
- Both of these have some limited free version available, at least for trial.

Digital wireframing and mockup tools

- A mockup tool worth mentioning is Balsamiq.
- It provides functionality to create medium-to-high fidelity mockups.
- It also provides some wireframing features (i.e. active controls and navigation simulation).
- It is available as a desktop application or as an online, web app.
- It's online version is ideal for collaborative design.
- It has hundreds of templates and pre-defined UI elements for mobile, desktop, and web platform.
- Balsamiq might be the most recommended mockup and wireframing tool available.

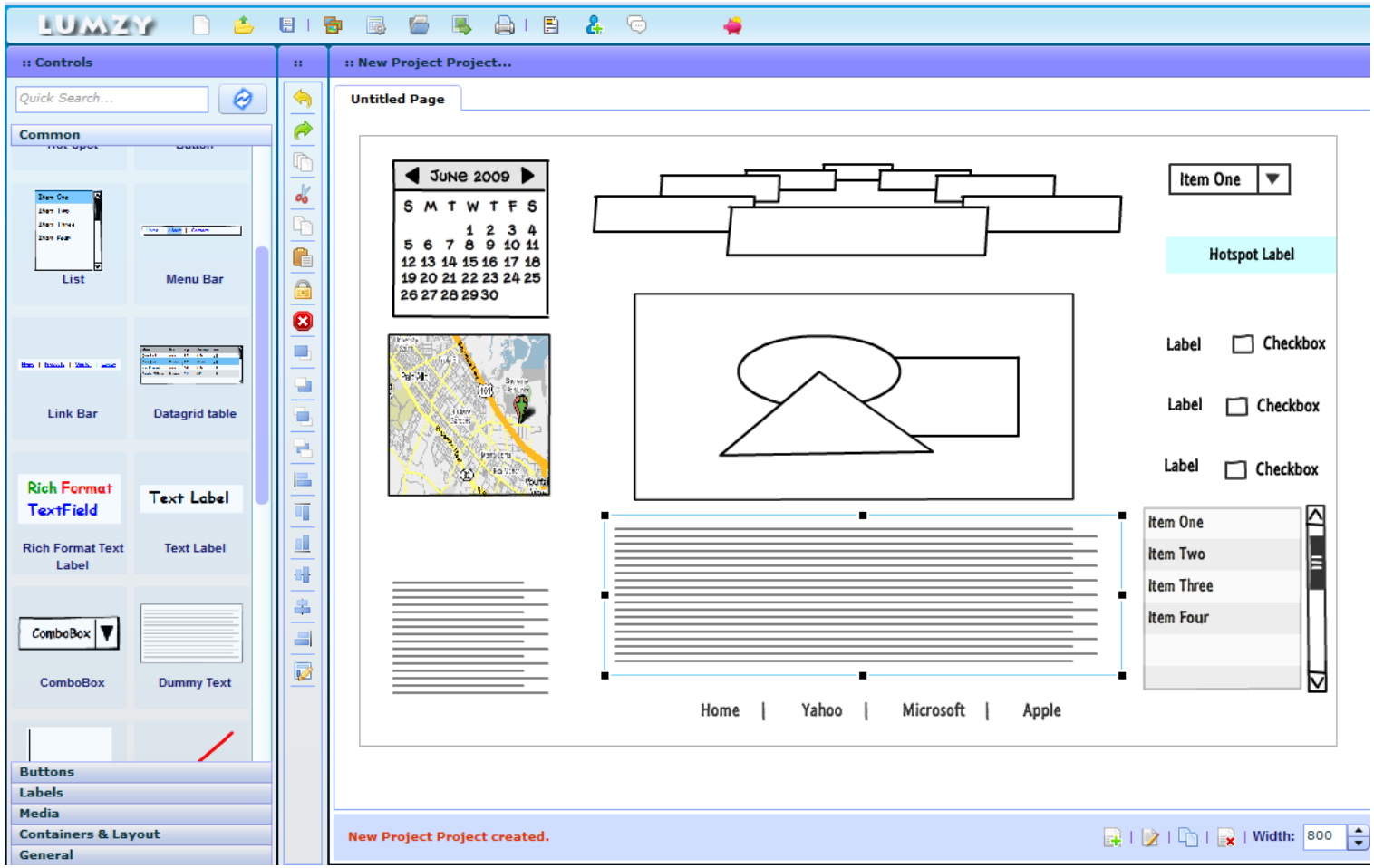
Digital wireframing and mockup tools



Digital wireframing and mockup tools

- A wireframing tool worth mentioning is Lumzy.
- It provides tools and templates to prototype complex interactions using working simulations of common controls.
- In Lumzy you can tie actions and events to controls and emulate your project.
- It is a functional prototyping tool where user actions can trigger popups, alerts, page navigation, etc.
- It is also a web application, which facilitates collaboration and sharing.

Digital wireframing and mockup tools



Prototyping

Prototyping

- Often, during the course of a project, there are needs that require a more functional simulation of interactions than those offered by mockups or wireframes.
- In such cases, prototyping can be used to create incomplete realizations of the functionality of your interface.
- Prototypes allow you to explore interactions and to also understand backend issues and other issues related to information architecture and infrastructure.
- They should be considered incomplete and disposable drafts of the final interface.
- Given their nature, it is important to minimize the effort required to build them.
- Or, it is important to maximize reusability of specific components.

Prototyping: use a dynamic language

- One of the most effective ways of minimizing cost and effort of developing prototypes is to use a dynamic programming language.
- This is, of course, why I chose Python for use in this course.
- Dynamic languages, like Python or Lisp (e.g. Clojure on the JVM) are extremely flexible.
- They are also interactive and directly support incremental development.
- Dynamic languages, however, do not necessarily optimize reusability of prototype components.

Prototyping: use a GUI designer

- Most high-level GUI toolkits (like GTK, WxWidgets, and QT) have GUI Designers that allow you to paint your interface.
- They allow you to compose widgets and containers using the actual elements of the GUI toolkit.
- These designs are flexible because UIs can be easily reconfigured and recomposed.
- They all have some method of generating code that implements the skeleton of your user interface.
- Some care must be taken:
  - Do NOT modify the code generated by the UI designer.
  - You should generally inherit from the generated classes.
  - These derived classes will implement the logic specific to your application.

Summary

Summary

- We haven't even really scratched the surface of the tools available for mockups, wireframing and prototyping.
- Wireframes are useful for exploring and simulating interactions in user interfaces.
- Mockups are useful for envisioning the final look-and-feel of an interface.
- Prototypes are more functional than wireframes, and allow you to link interactions with (partial) backend implementations (like data files and databases).
- GUI Designers can be an excellent way of progressing from wireframe to prototype in a way that ensures reusability of your design elements in the final interface.