

High Fidelity Prototypes

Human Computer Interaction

Luigi De Russis

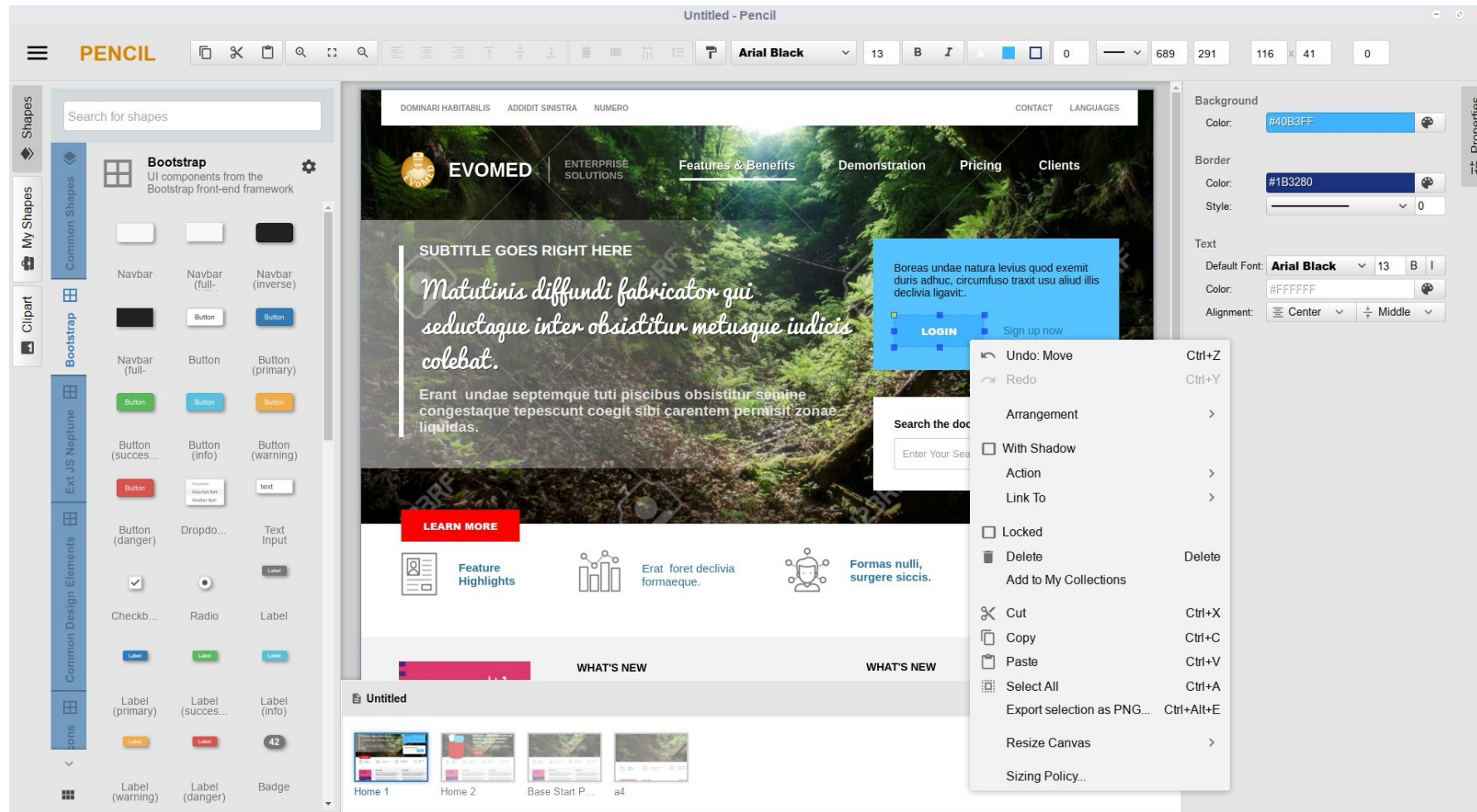
Academic Year 2023/2024

Hi-Fi Prototypes

- Actual computer application, with final-looking layout, colors, and graphics
 - May use design prototyping tools
 - May use real application code
- Much more expensive to build
- More time is spent with graphic design than interaction design
- When tested, people will mostly comment about colors, fonts, ...
 - representation communicates “finished”

High-fidelity Computer Prototypes

Semi-interactive



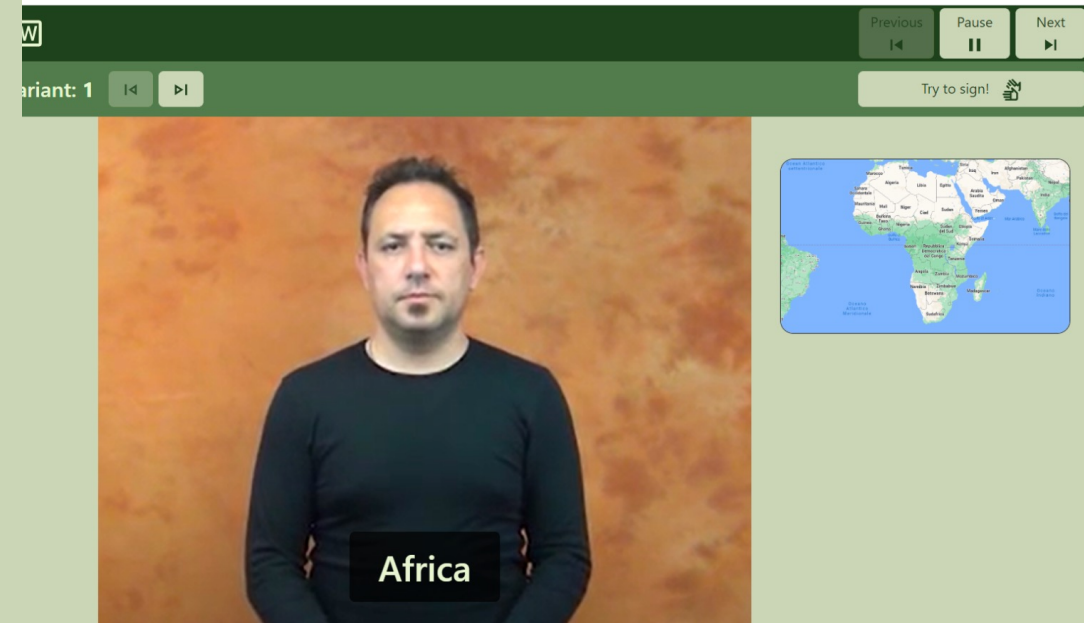
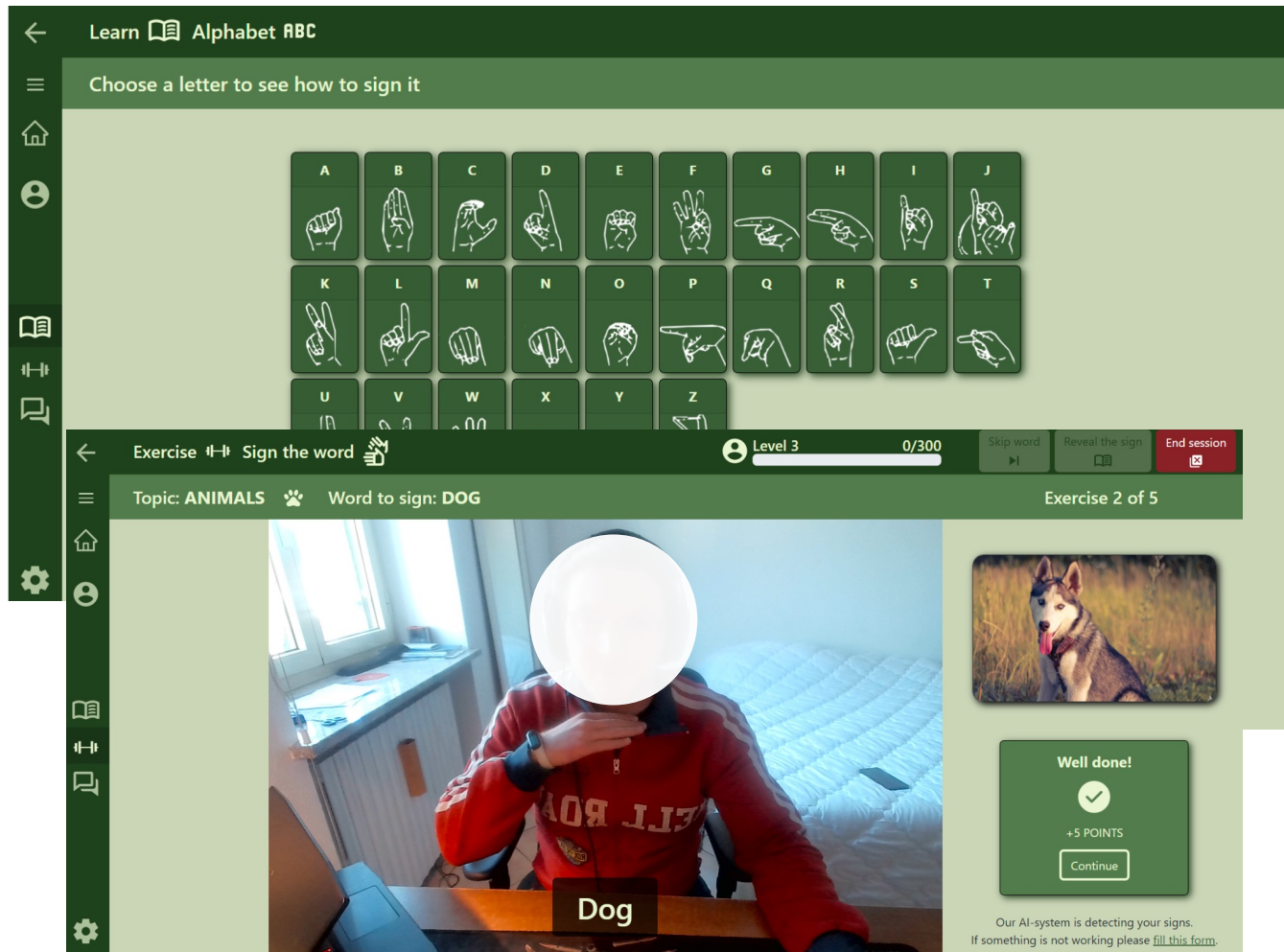
High-fidelity Prototypes

With Code



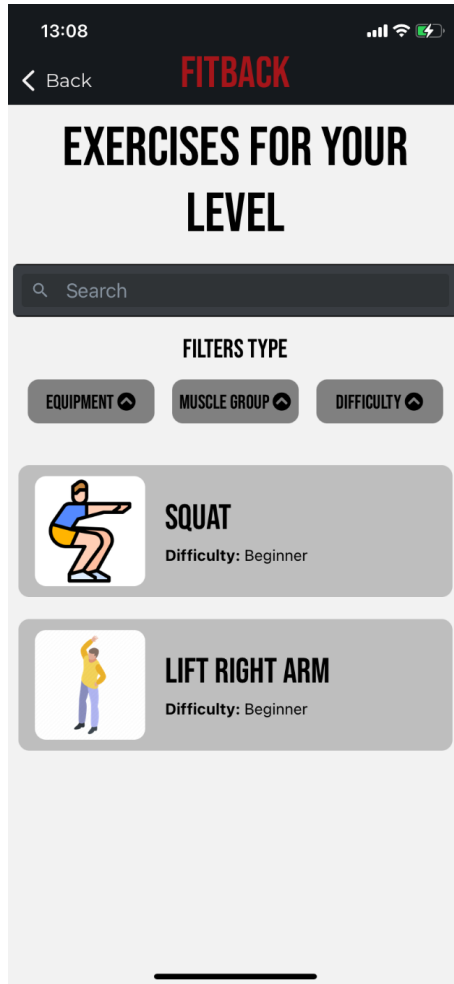
High-fidelity Prototypes

With Code



High-fidelity Prototypes

With Code



What Can We Learn From Hi-Fi Interactive Prototypes?

- Screen layout
 - Is it clear, overwhelming, distracting, complicated?
 - Can users find important elements?
- Colors, fonts, icons, other elements
 - Well-chosen?
- Interactive feedback
 - Do users notice & respond to status bar messages, cursor changes, other feedback
- Efficiency issues
 - Controls big enough? Too close together? Scrolling list is too long?

Suggested Video

- **Prototyping: fake it till you make it**
- By Apple Design Team
- <https://youtu.be/3lqh-A5Jy4Q>



Introducing

Some Tools For Interactive Hi-Fi Prototypes

No-Code



<https://www.invisionapp.com/>



<https://www.figma.com>

FROONT

<https://froont.com/>

webflow

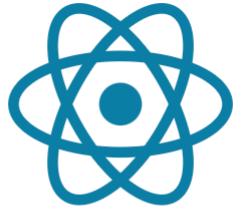
<https://webflow.com/>



<https://principleformac.com/>

Some Tools For Interactive Hi-Fi Prototypes

With Code



React

<https://react.dev>



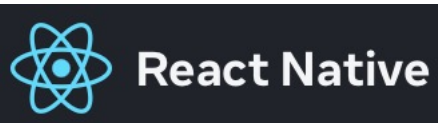
Firebase

<https://firebase.google.com>



<https://ngrok.com>

...



<https://reactnative.dev>



<https://expo.dev>



<https://virocommunity.github.io>

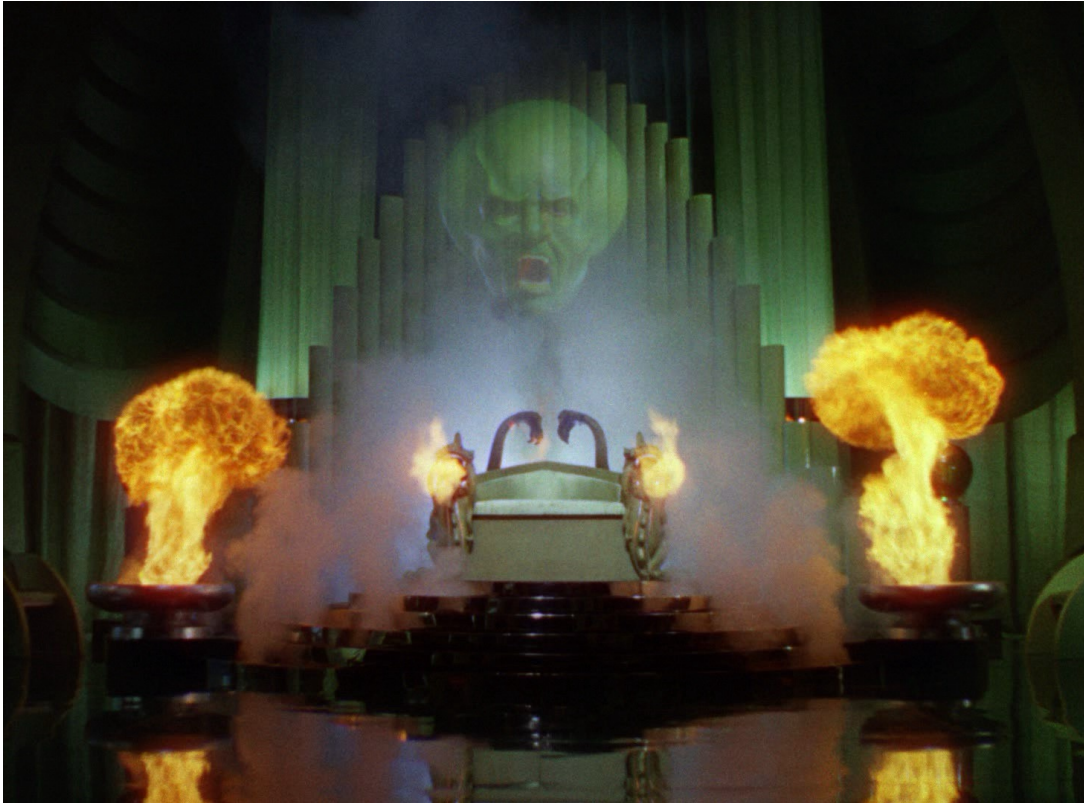
Wizard-of-Oz Techniques

Faking a technology, or filling-in for missing functionality

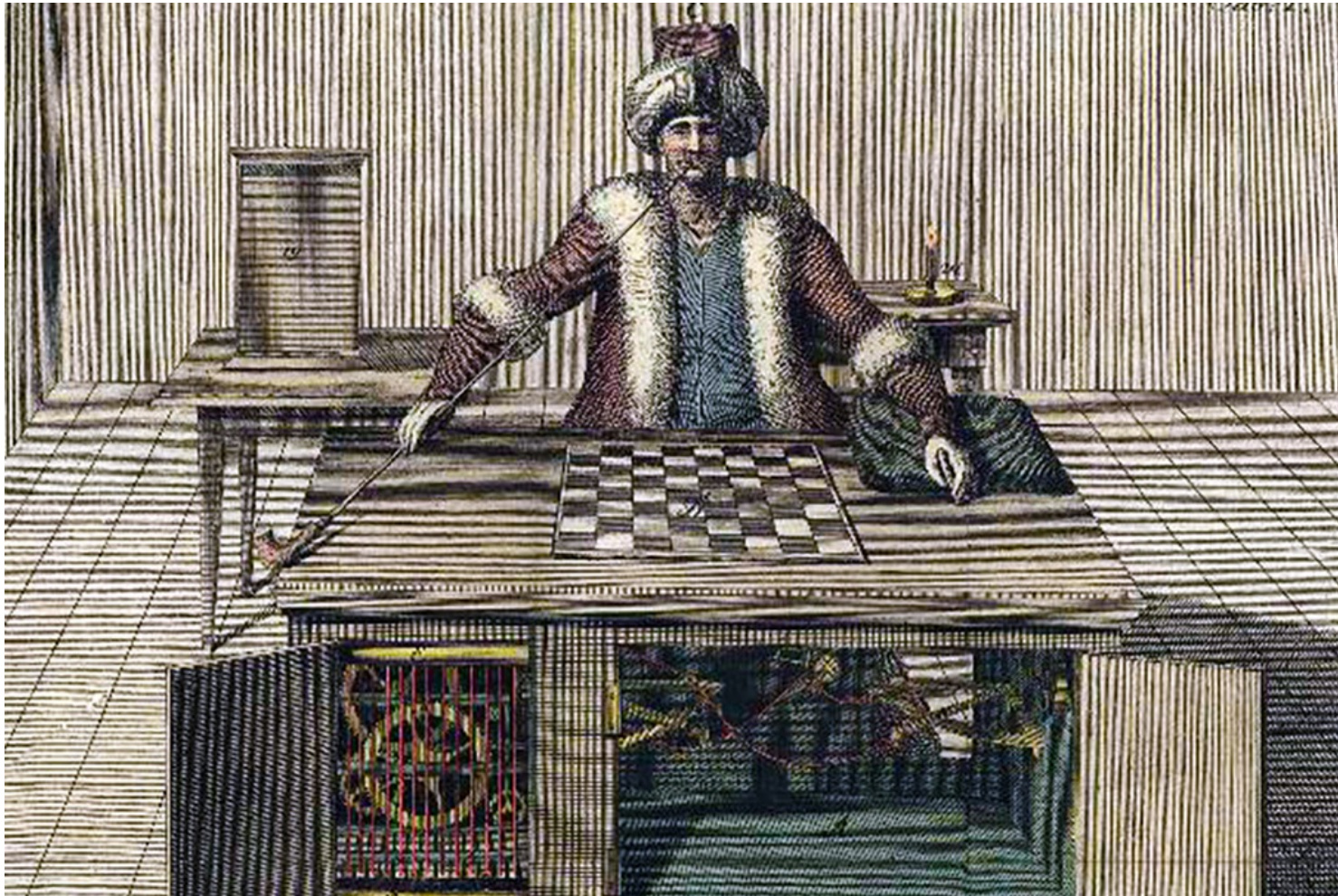
Goal

- How to test an application that is really complete...
 - With finalized user interface
 - With finalized algorithms
 - Also including stuff that we still are not able to implement
- ...but without actually writing the code
 - Except for a semi-interactive ‘dumb’ prototype

Remember The Man Behind The Curtain?



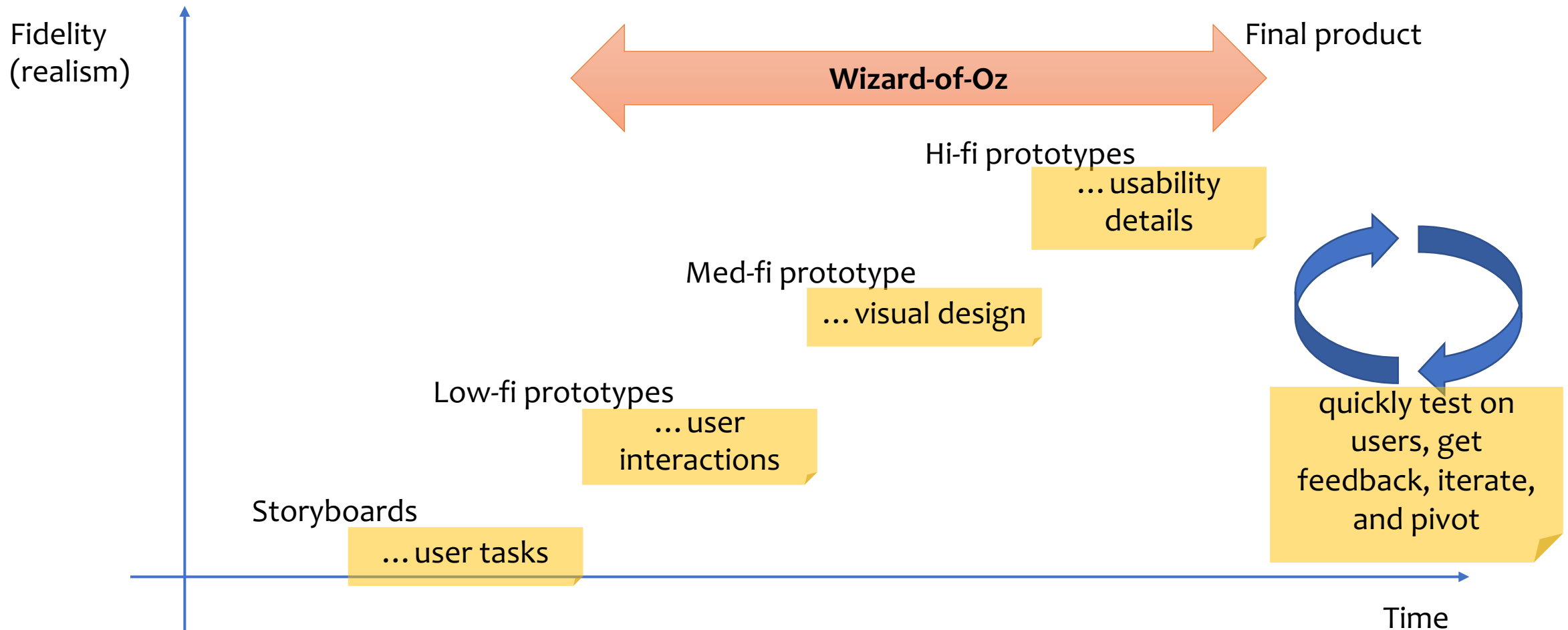
Remember the Mechanical Turk?



Wizard-of-Oz

- Software simulation with a human in the loop to help
- “Wizard of Oz” = “man behind the curtain”
 - Simulates the machine behavior with a human operator
 - Wizard is usually but not always hidden
- Often used to simulate future technology
 - Speech recognition
 - Learning
- Wizard may be hidden or visible
 - Must always be revealed, at least at the end

Prototypes Facilitate Conversations About...



Implementing a Wizard-of-Oz Prototype

- Choose supported tasks and scenarios
- Create User Interface mock-ups
 - Implement a part of the system
 - Leave “hooks” for the Wizard’s actions
- Implement a back-office interface for the Wizard
- Define “rules of behavior” for the Wizard
 - When he should respond
 - How it should respond (the “algorithm”)

Benefits

- Faster and cheaper than most interactive prototypes
- More “real” than paper prototyping
- Creating multiple variations is easy
- Identifies bugs and issues with current design
- Can envision applications that are difficult to build
- Playing wizard allows a better understanding of algorithmic requirements

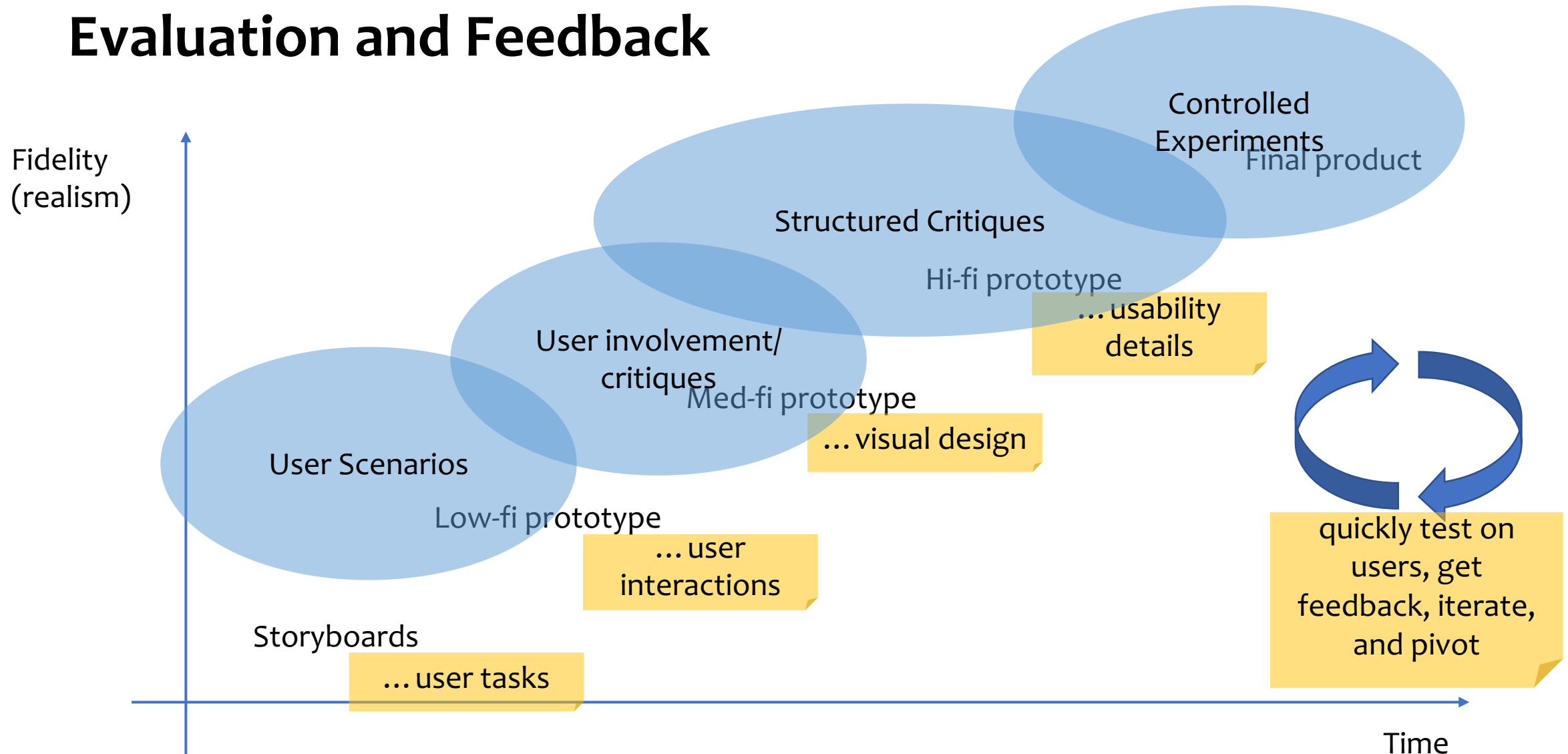
Risks

- May be over-optimistic
 - Speech recognition that always works (instead of having an error rate)
 - Super-intelligence (that will never exist)
- Wizard behavior is difficult
 - Take into account system limitations
 - Emulate expected system response
 - Within acceptable timing
- Needs at least two researchers

Wrap-up

Many different techniques, applicable to different goals and contexts

Evaluation and Feedback



References and Acknowledgments

- Google, Begin Today With Rapid prototyping,
https://www.youtube.com/playlist?list=PL9KVIdEJ2K8NDpsiyYpcbB_qifd3y5CYZ
- MIT, http://web.mit.edu/6.813/www/sp18/classes/11-prototyping/#reading_11_prototyping
- Scott Klemmer, Storyboards, Paper Prototypes, and Mockups,
<https://youtu.be/z4glsttyxw8>
- Thanks to Fulvio Corno, past teacher of the course, for his work on some of these slides

License

- These slides are distributed under a Creative Commons license “**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**”
- **You are free to:**
 - **Share** — copy and redistribute the material in any medium or format
 - **Adapt** — remix, transform, and build upon the material
 - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
 - **Attribution** — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
 - **NonCommercial** — You may not use the material for [commercial purposes](#).
 - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.
 - **No additional restrictions** — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.
- <https://creativecommons.org/licenses/by-nc-sa/4.0/>

