

# Prototyping

Department of Computer Science

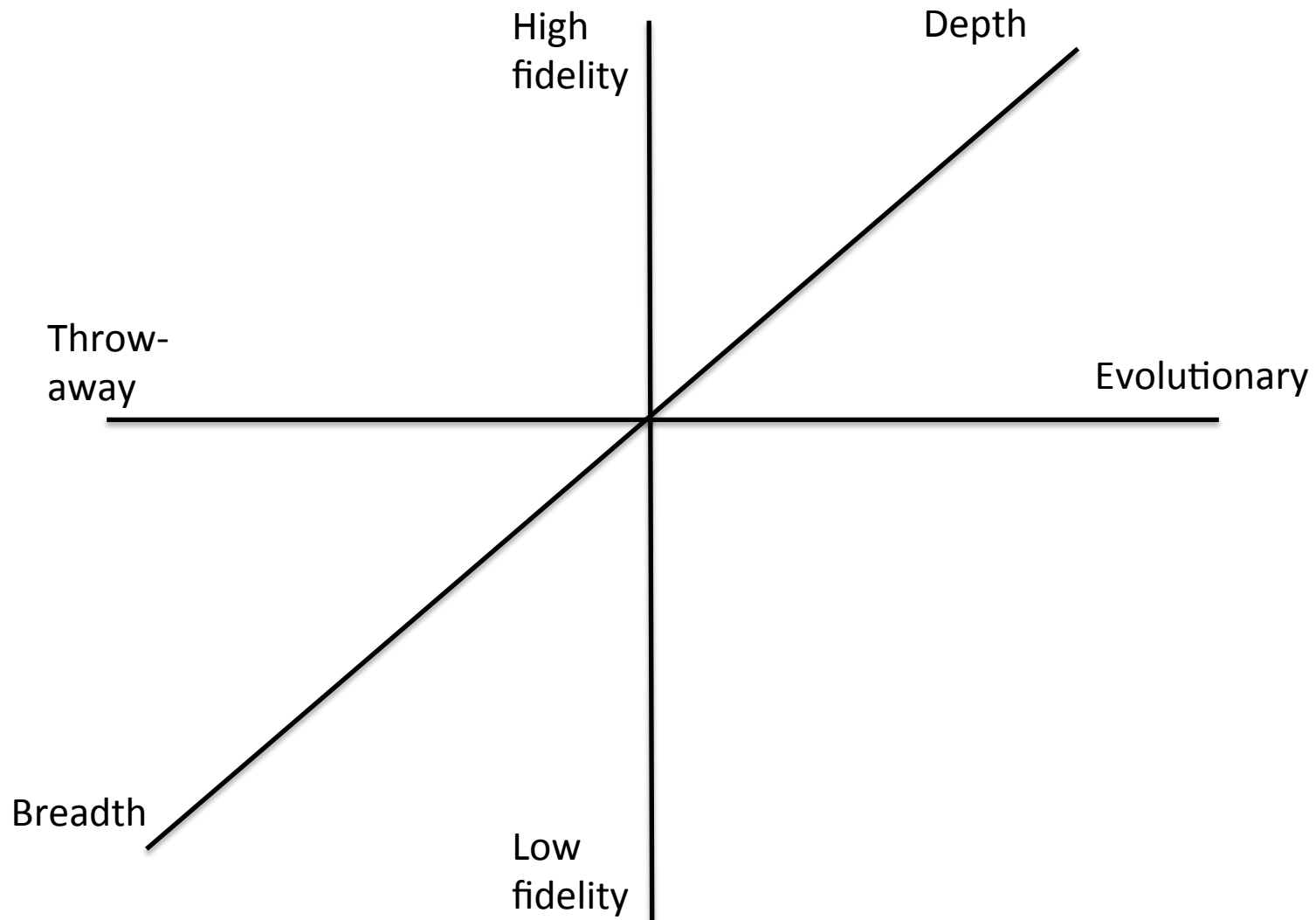
# What is prototyping?

- There is a point where you move from conceptual design to physical design
- Usually, you need to try out several different designs before writing a single line of code
- These designs can be used either to get a preview of what will be implemented, to feed into discussions with stakeholders, or more commonly to iterate in your own design lifecycle

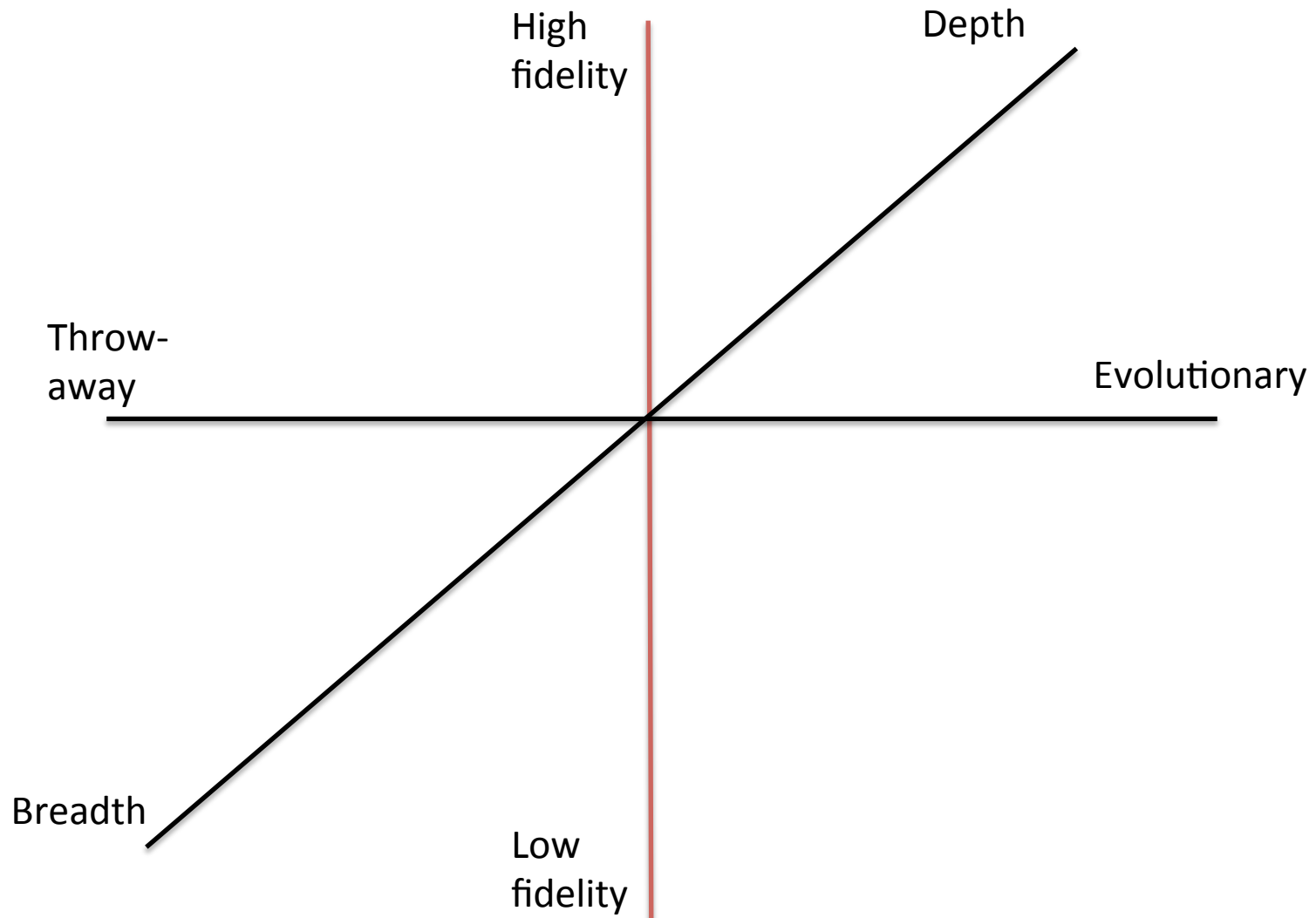
# Why prototype?

- Very useful when discussing ideas with stakeholders
- Encourages reflection on the design
- Allows you to explore alternative designs without committing to one idea
- Allows you to make ideas from scenarios more concrete

# Dimensions of Prototyping



# Dimensions of Prototyping



# Fidelity

- When discussing *fidelity* of prototypes, we mean the amount to which there is real working functionality
- *Low fidelity* prototyping – incomplete functionality, often not programming code, wireframes, sketches
- *High fidelity* prototyping – allows realistic interaction, as close as possible to real functionality that users will interact with

# Low Fidelity: Storyboards

- Storyboard is a technique borrowed from the film industry
- The form we know now was first used by Walt Disney for animation to explain motion of characters
- Only a newcomer to interactive technology in terms of large scale use – become very popular now because we need a design tool to help with the problem of moving away from the desktop
- As ubiquitous computing (continues) to dominate, it is likely we will see more and more storyboarding in design

# Low Fidelity: Storyboards (2)

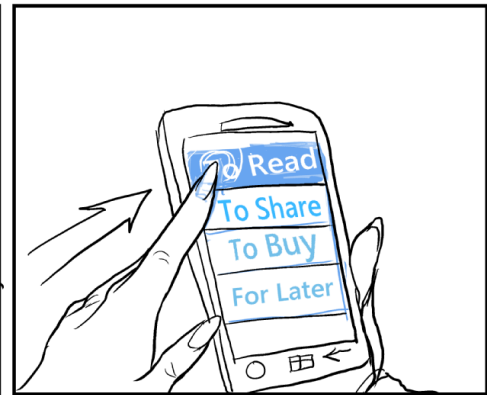
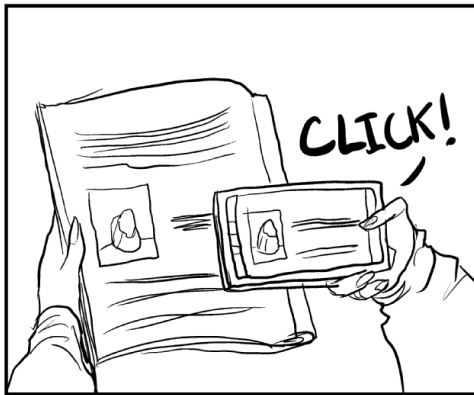
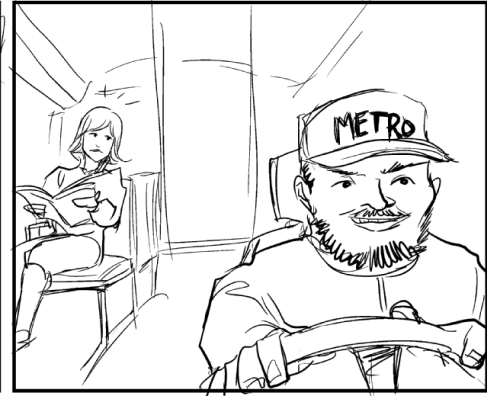
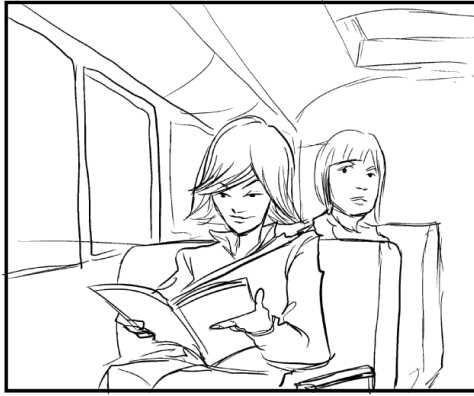
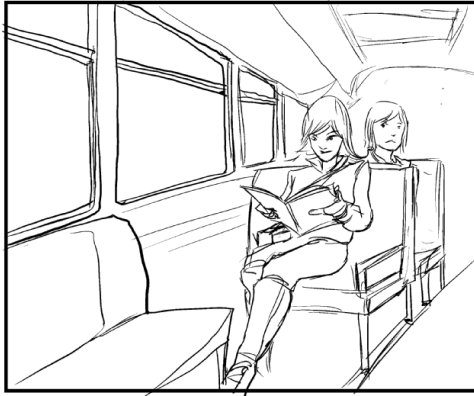
- Each panel demonstrates a ‘beat’ in time – Beats are a base unit of story telling in whatever medium you are working in
- In Interaction Design, we are trying to show frame by frame the user working with an application in context
- These storyboards can then be used to highlight areas where design patterns can be applied or novel designs can be used



# Low Fidelity: Storyboards (3)

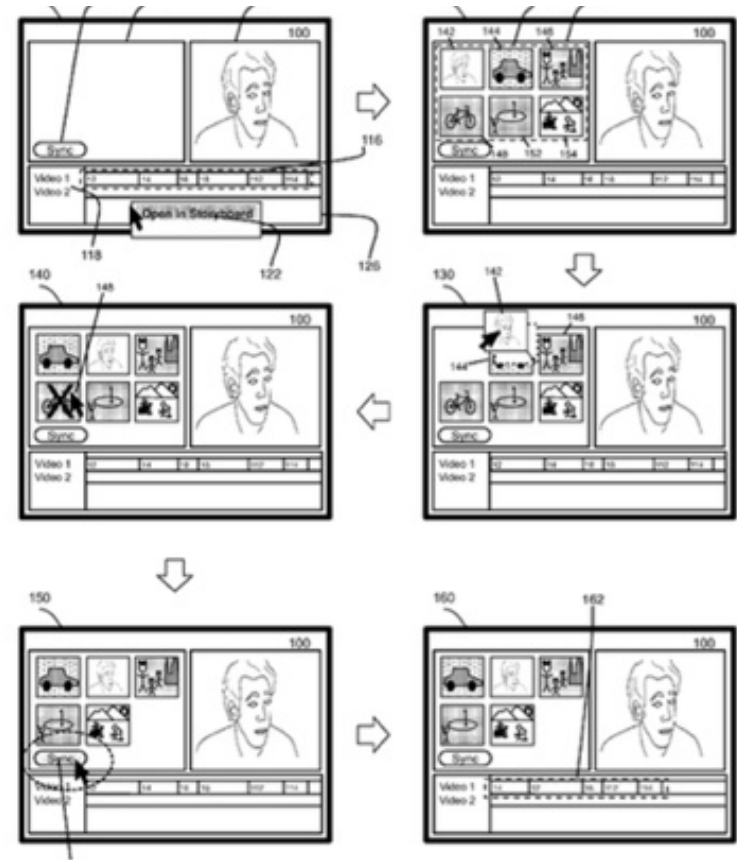
- While there is no standard set of things to define as a 'beat' - things we often see in interaction design include:
  - Change of physical locations
  - Change of context within an application (e.g. starting/stopping a movie)
  - Users inputting information to the system
  - System responding with feedback or errors
  - Any dialogue that occurs in collaborative work

# Storyboard Example



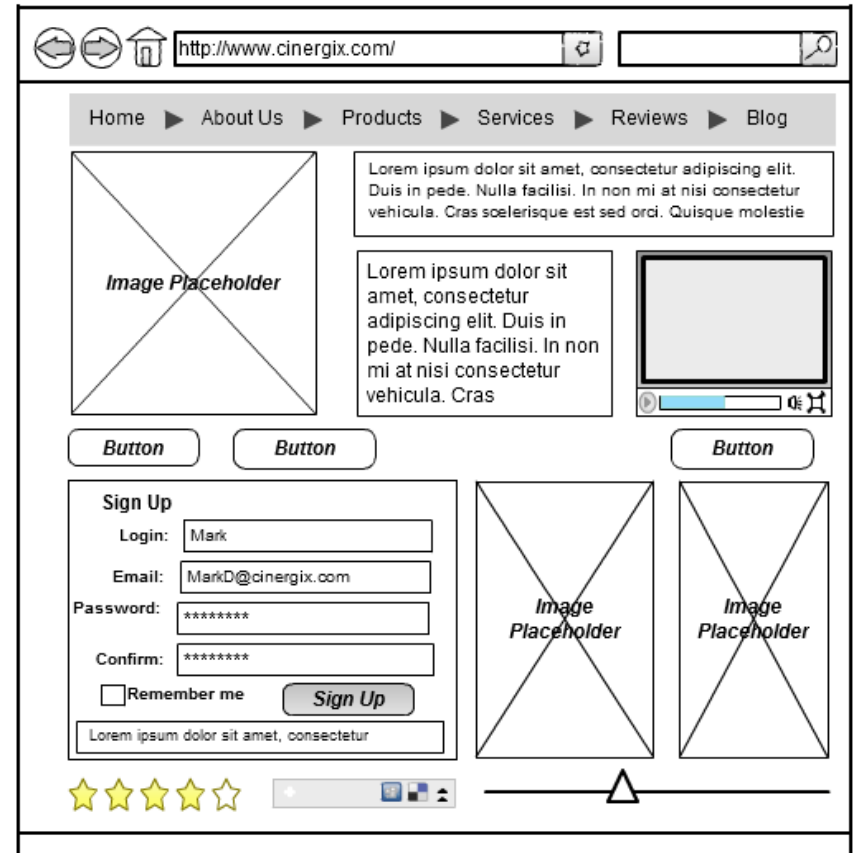
# Low Fidelity: Sketching

- Application sketching is the creation of thumbnails and screen layouts in a very rapid manner
- Produce many designs to explore the design space
- Use all of your personas, scenarios, task models, conceptual designs etc. to inform the designs



# Low Fidelity: Wireframes

- More detailed layout – provides means of walking through tasks
- Approximately correct controls for potential design
- Generally rough sketches, but can use more high definition components from style kits



# Low Fidelity Prototypes

- Advantages:
  - Cheap, simple, fun to make
  - Also cheap, simple, fun to modify
  - Clear to stakeholders that they can be criticized
  - Designers do not have too much at stake in them
- Disadvantages
  - Do not allow realistic use

# Low Fidelity Prototype Resources

- Huge numbers of stencil kits and frameworks now exist for different applications and application styles
  - Keynotopia – HUGE set of stencils for Powerpoint and Keynote
    - <http://keynotopia.com/>
- iPad – the new home of UI Sketching (dozens of apps)
  - App Cooker
    - <http://www.appcooker.com>
  - Prototyping on Paper
    - <https://itunes.apple.com/gb/app/pop-prototyping-on-paper./id555647796?mt=8>

# High Fidelity Prototypes

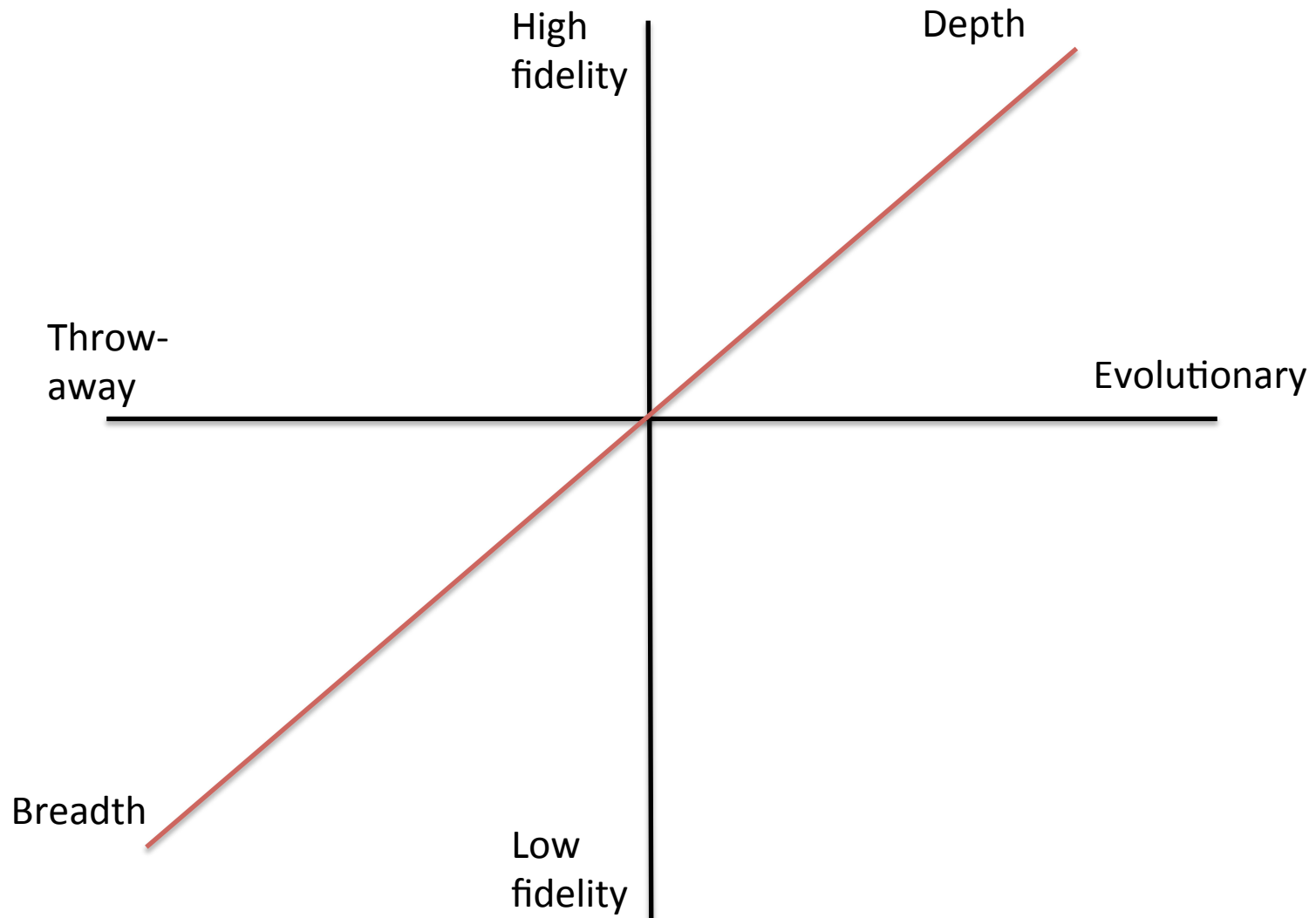
- Uses materials you would expect to find in the end product or system and looks much more like the final thing
- You begin to program portions of the system, implementing either across the breadth of the system or deep into a particular task
- For platform specific software, harder to do, but rapid prototyping tools are in many packages now (e.g. WYSWYG window construction)
- One of the interesting things about the Web is that it's very easy to prototype in HTML and then elaborate the source code

# High Fidelity Prototypes (2)

- Advantages:
  - You get the real “look and feel”, (some of) the functionality
  - Serves as a living specification
  - Good for exploration of design features and evaluation
  - Good marketing and sales tool
- Disadvantages
  - Disadvantages:
  - Is it worth the expense to develop if it might be radically changed?
  - Evaluators tend to comment on superficial design features rather than the real design
  - Developers can be reluctant to change it
  - Can set user expectations too high



# Dimensions of Prototyping



# Breadth vs. Depth

- Breadth
  - Prototype functionality across the different user journeys
  - Look for journeys with common start points, or share common routes to try to reuse components
  - Example: Prototype the home page of a website and the first page off of each of the navigation bar item
- Depth
  - Provide an entire user journey taking the user from start to finish
  - Example: A e-commerce site taking you from initial login through to final click of submitting an order

# Breadth vs. Depth

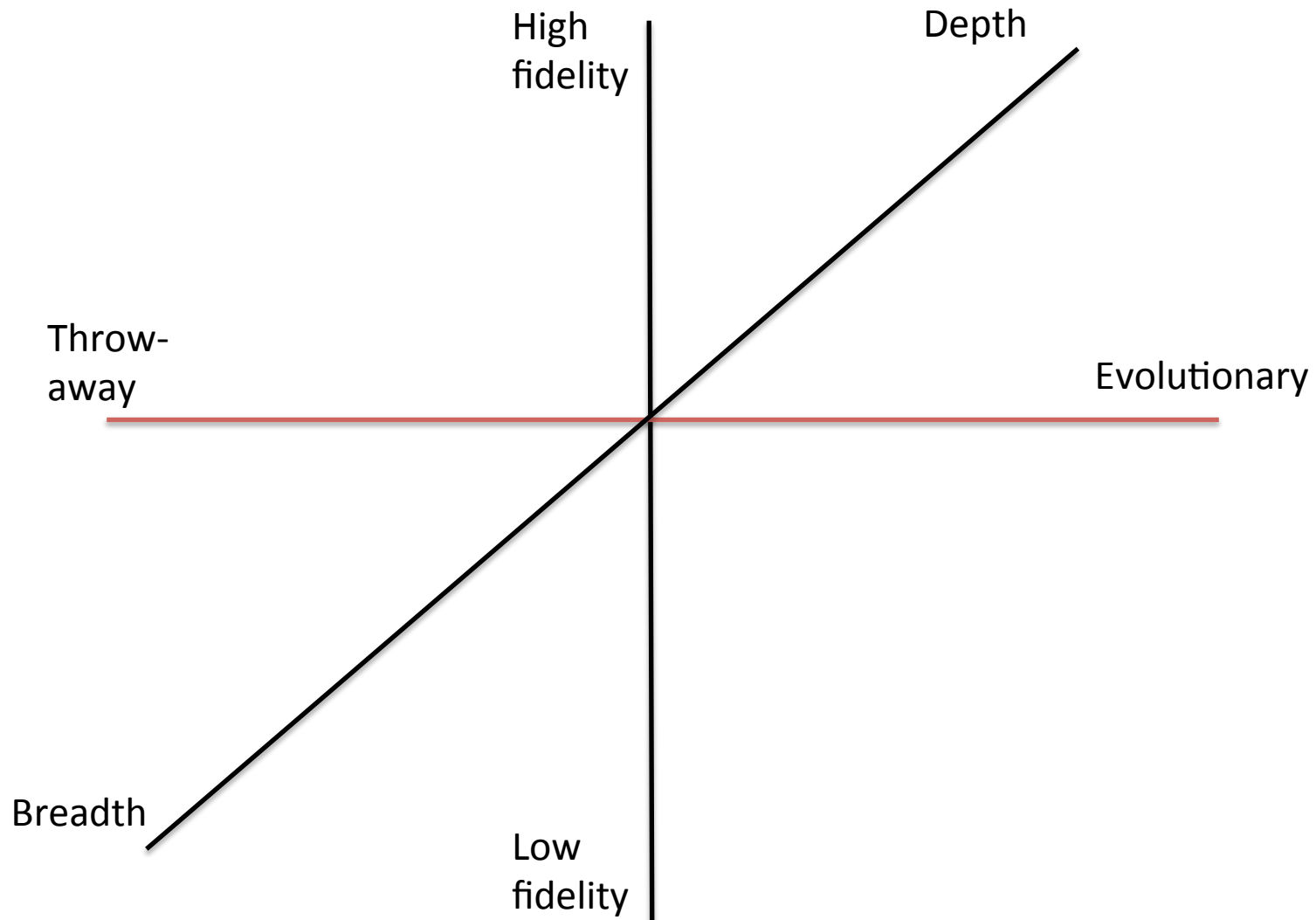
## Breadth

- Advantages
  - Can cover a wide range of possible functionality
  - Can be used to investigate hard problems like navigation
- Disadvantages
  - May lose detail about problems in the dialogue
  - May lose detail of gaps in user journeys

## Depth

- Advantages
  - Can explore a journey through to completion
  - Check for major problems in dialogues or in feedback
- Disadvantages
  - Requires commitment to a particular user journey dialogue
  - Takes more time – difficult to do many deep journeys

# Dimensions of Prototyping



# Throwaway vs. Evolutionary

- On multidisciplinary user experience teams you will usually see a mix of these
  - Designers provide “throwaway” prototypes and developers use “evolutionary” prototypes in their code bases
- Throwaway prototyping uses artefacts that at each stage but does not use the actual components in future iterations
- Evolutionary prototyping provides actual components that are used either directly (e.g. UI code base) or indirectly (e.g. model generation)

# Throwaway vs. Evolutionary (2)

## Throwaway

- Advantages
  - Often low-fidelity so all the good things that come with that
  - Not worried about issues around integration
- Disadvantages
  - Have to “do design twice” in transferring design from one format to working code

## Evolutionary

- Advantages
  - Often high fidelity, so good things that come with that
  - Can run actual scenarios
- Disadvantages
  - Poor coding techniques persist
  - Code base creep
  - Poor high fidelity prototyping which alienates stakeholders

# Summary

- Prototyping comes in a number of different flavours.
- Usually multiple rounds of prototyping, and a number of different types of prototypes will be used.
- Early design iterations benefit most from low-fidelity, throwaway, easy to present/modify prototyping while later iterations benefit from more high fidelity evolutionary prototypes.

# Readings

- Rogers, Preece et al. 3<sup>rd</sup> Edition
  - Chapter 11 – largely about prototyping and design