

User Interface Design

Elena Punskaya, elena.punskaya@eng.cam.ac.uk

What is a User Interface?

- Everything designed into a device with which a user may interact.



Image: Bill Stafford / NASA

- Vostok, 1960-1963, Voskhod, 1964-1965 spacecraft



Image: Andrew Bodrov/360cities

- Space Shuttle Atlantis 1991

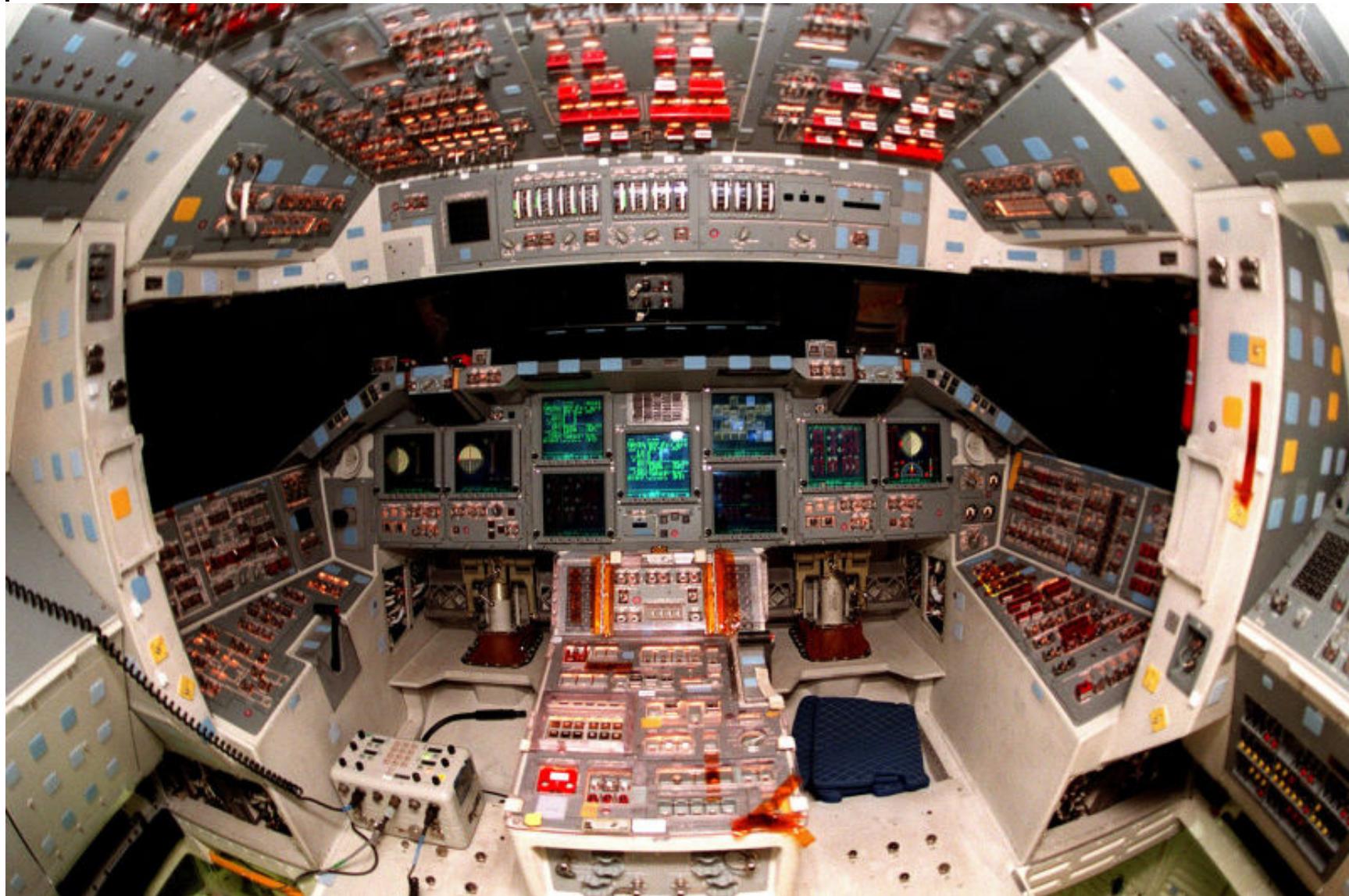


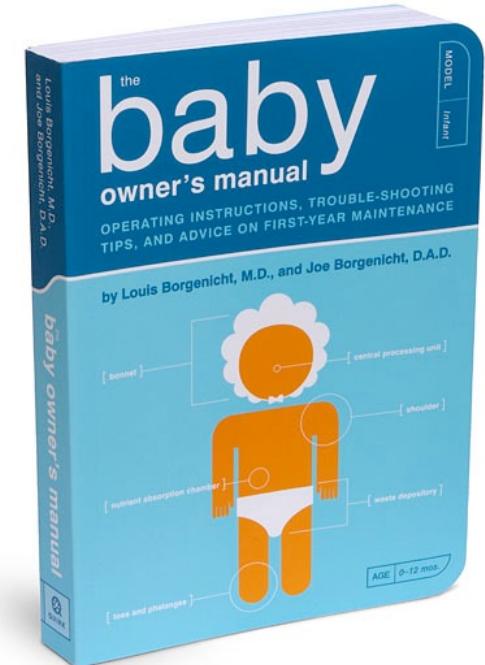
Image: NASA

“A user interface is well-designed when the program behaves exactly how the user thought it would.”

- User familiarity: copy well-known features from other programs
- Consistency: do similar operations in the same way
- Choose default values carefully: minimise the number of choices novice users have to make
- Use metaphors: to establish the user's conceptual model
- Recoverability: ensure UNDO works in every situation
- Provide help: but (!) make it sparing and highly context specific
- Sounds easy, however, there are some constraints

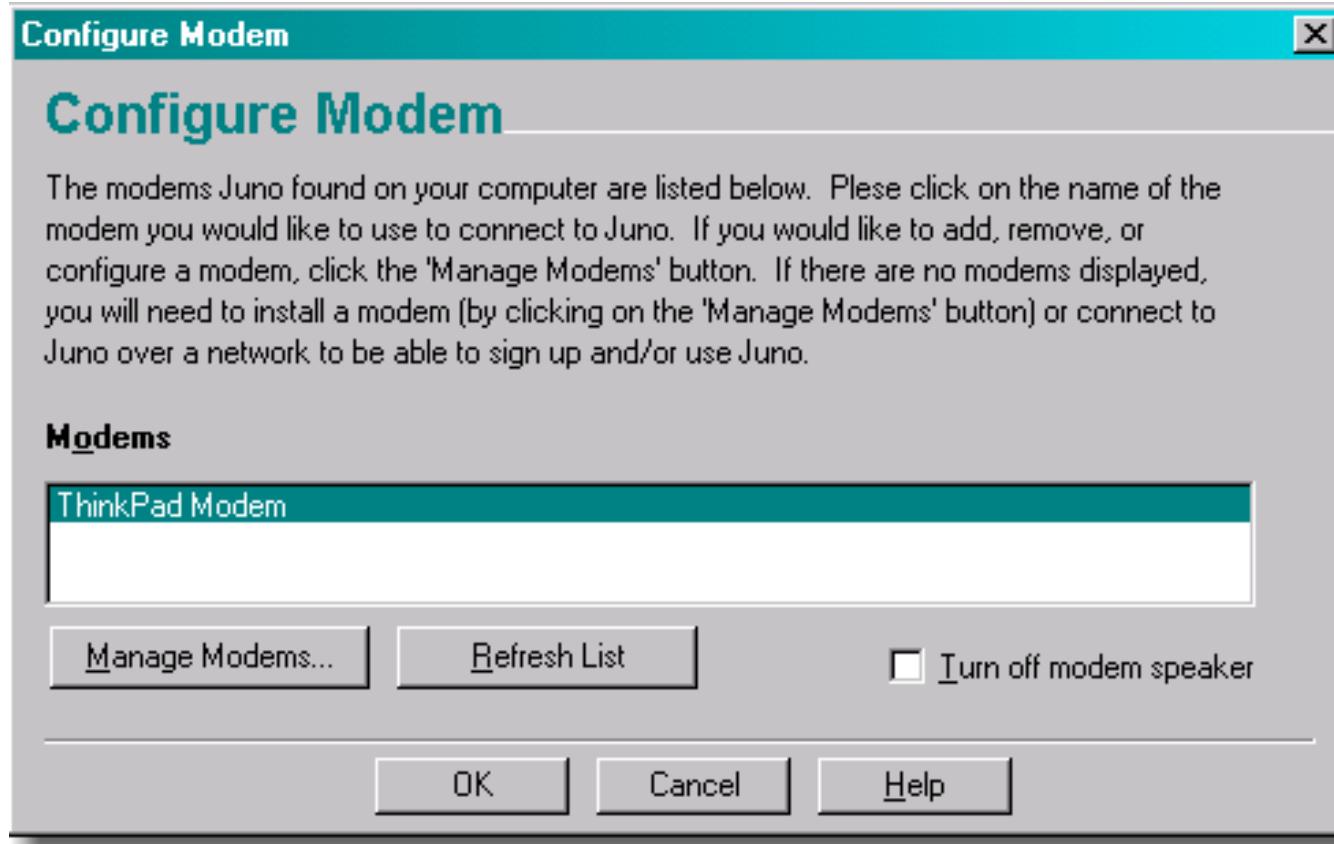
Constraints

- Users don't read the manual



Constraints

- Users won't read more than 2 or 3 words at a time

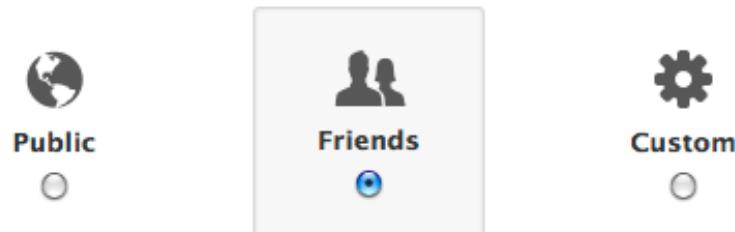


Constraints

- Users don't like making choices

Control Your Default Privacy

This setting will apply to status updates and photos you post to your profile from a Facebook app that doesn't have the inline audience selector, like Facebook for BlackBerry.



How You Connect

Control how you connect with people you know.

[Edit Settings](#)

How Tags Work

Control what happens when friends tag you or your content.

[Edit Settings](#)

Apps and Websites

Control what gets shared with apps, games and websites.

[Edit Settings](#)

Limit the Audience for Past Posts

Limit the audience for posts you shared with friends of friends or Public

[Manage Past Post Visibility](#)

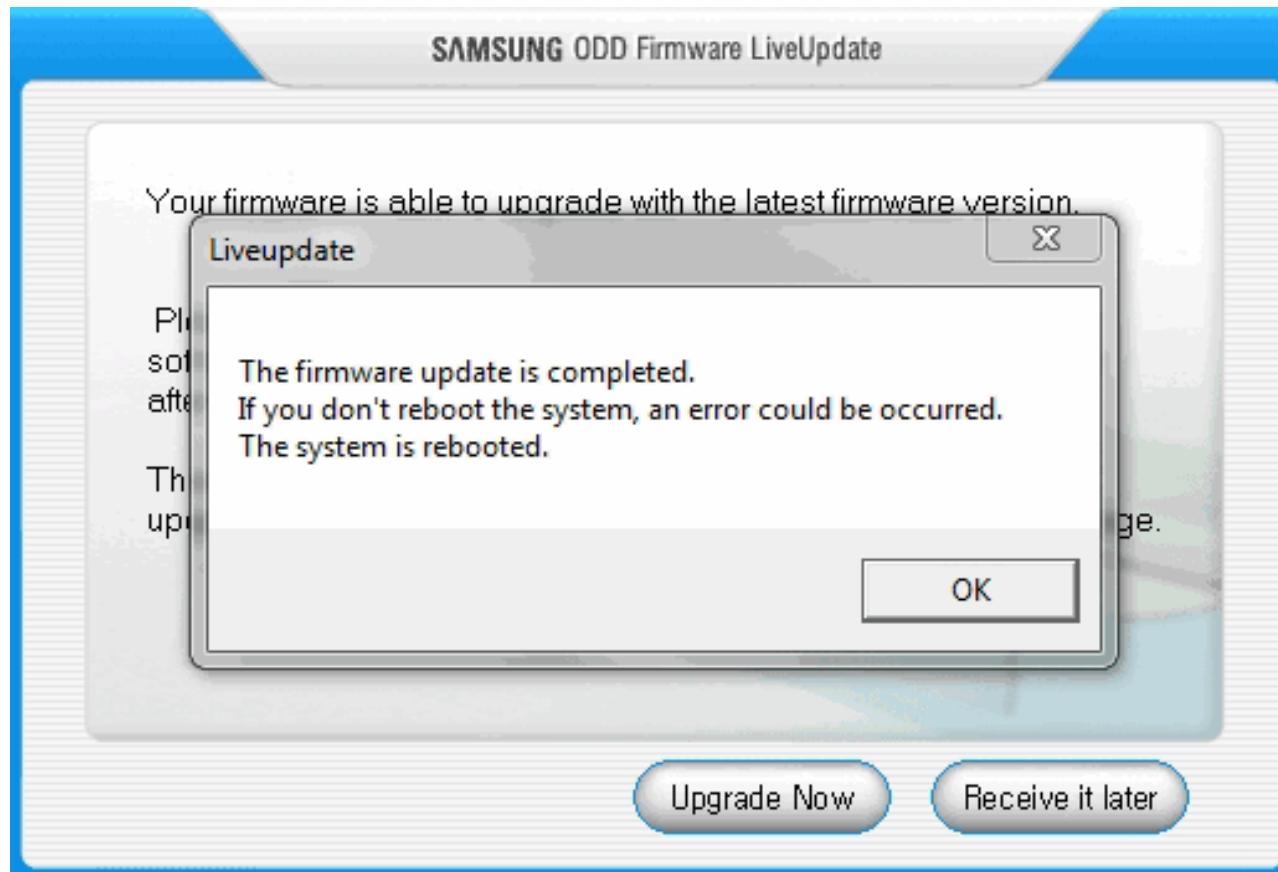
Blocked People and Apps

Manage the people and apps you've blocked.

[Manage Blocking](#)

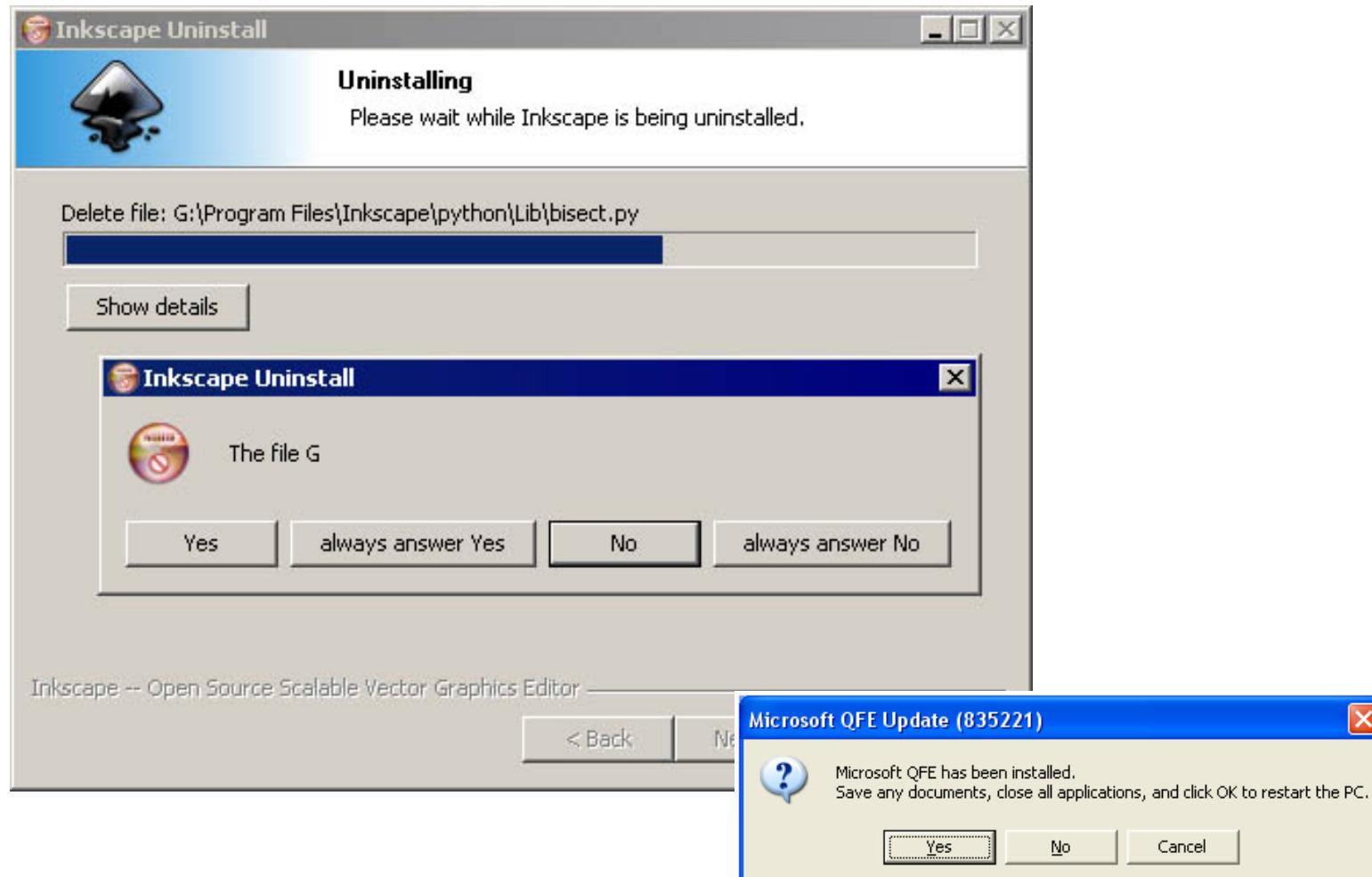
Constraints

- Users don't like being frightened



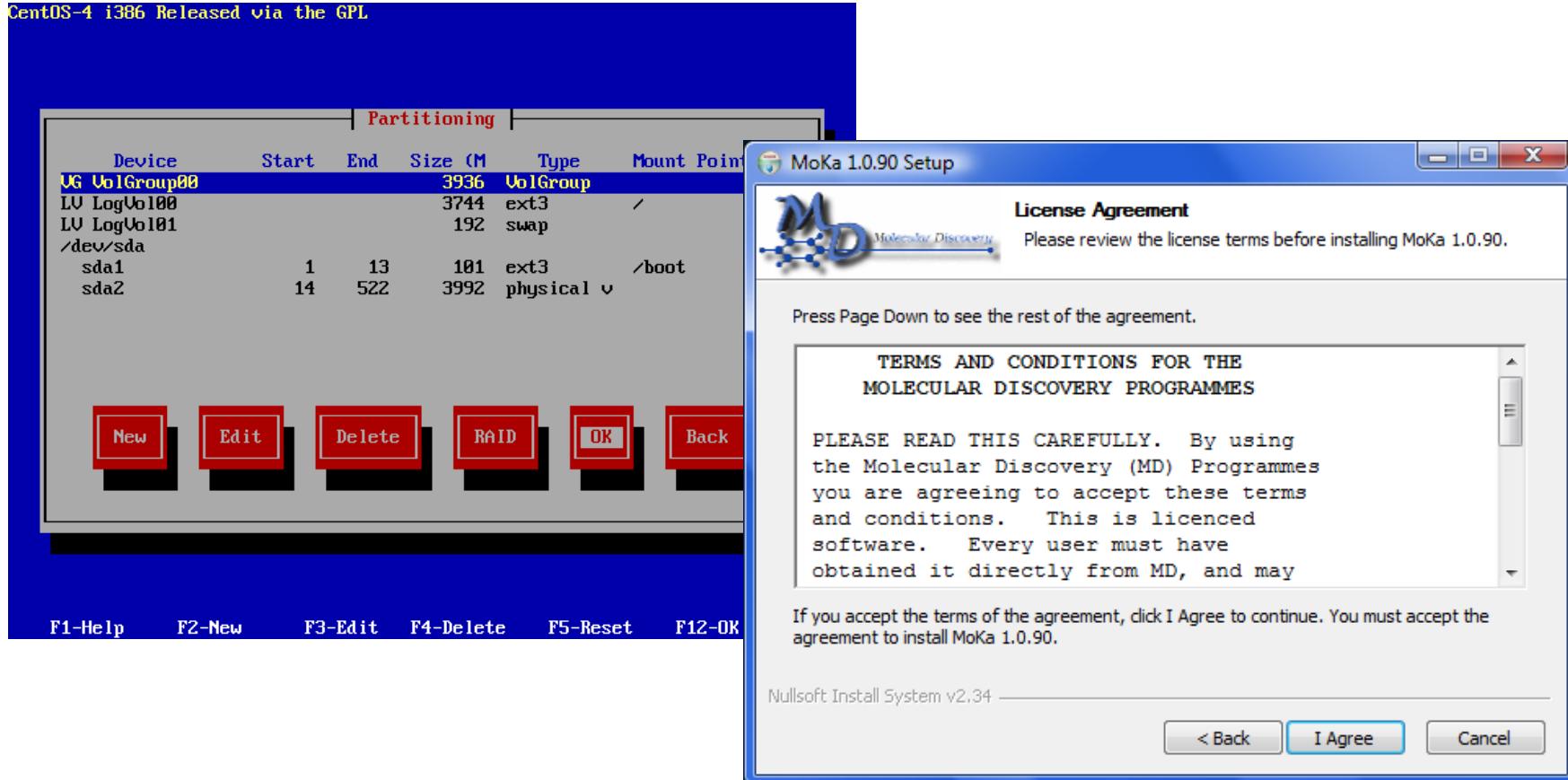
Constraints

- Users don't like being confused



Constraints

- Users ALWAYS click OK



UI Design Process

- Define a set of representative user types
- Identify the most important activities, i.e. use cases
- Assume some user model based on how you think each user will expect to accomplish each activity?
- Design the UI based on the current user model
 - use metaphors to illuminate the user model
 - borrow familiar behaviours from other programs esp. controls
 - keep it very simple, users will always assume the simplest model
- Implement a prototype
- Test each use case with each user type and iterate
 - do not ignore the minority in favour of the majority
 - note areas where people have trouble. These are probably areas where the program model does not match the user model
- Refine the user model and iterate

User Types

- User types are often application specific but common categories are
 - geeks – enjoy figuring out how things work
 - teenagers – like geeks but know no fear
 - experienced users – think they already know how everything works
 - naive users – often too timid to explore the interface
- There is also a dichotomy between
 - People raised with computers/smartphones/tablets/Internet
 - New users trying to use computers
- For each user type selected as appropriate for testing an application, a profile should be created answering the following questions:
 - What are the user's goals?
 - What are the user's skills and experience?
- The answers to these questions will help design a strategy which allows users to exploit their skills and experience to efficiently satisfy their goals
- Ideally, a user interfaces should work well for all user types

From User-Centred to Activity-Centred Design

- Donald A. Norman says

Human-centered design does guarantee good products. It can lead to clear improvements of bad ones. Moreover, good human-centered design will avoid failures. It will ensure that products do work, that people can use them. But is good design the goal? Many of us wish for great design. Great

- and then provokes

Human-centered design has become such a dominant theme in design that it is now accepted by interface and application designers automatically, without thought, let alone criticism. That's a dangerous state—when things are treated as accepted

Activity-centered design might be superior.



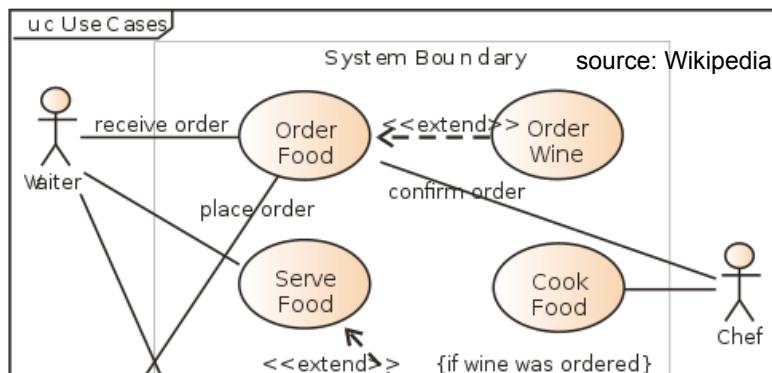
Jakob, Don, Tog

[Jakob Nielsen, Ph.D.](#)
"The Guru of Web Page Usability" (New York Times)
[Donald A. Norman, Ph.D.](#)
"The Guru of Workable Technology" (Newsweek)
[Bruce "Tog" Tognazzini](#)
"Leading Authority on Software Design" (HotWired)

<http://www.nngroup.com/>

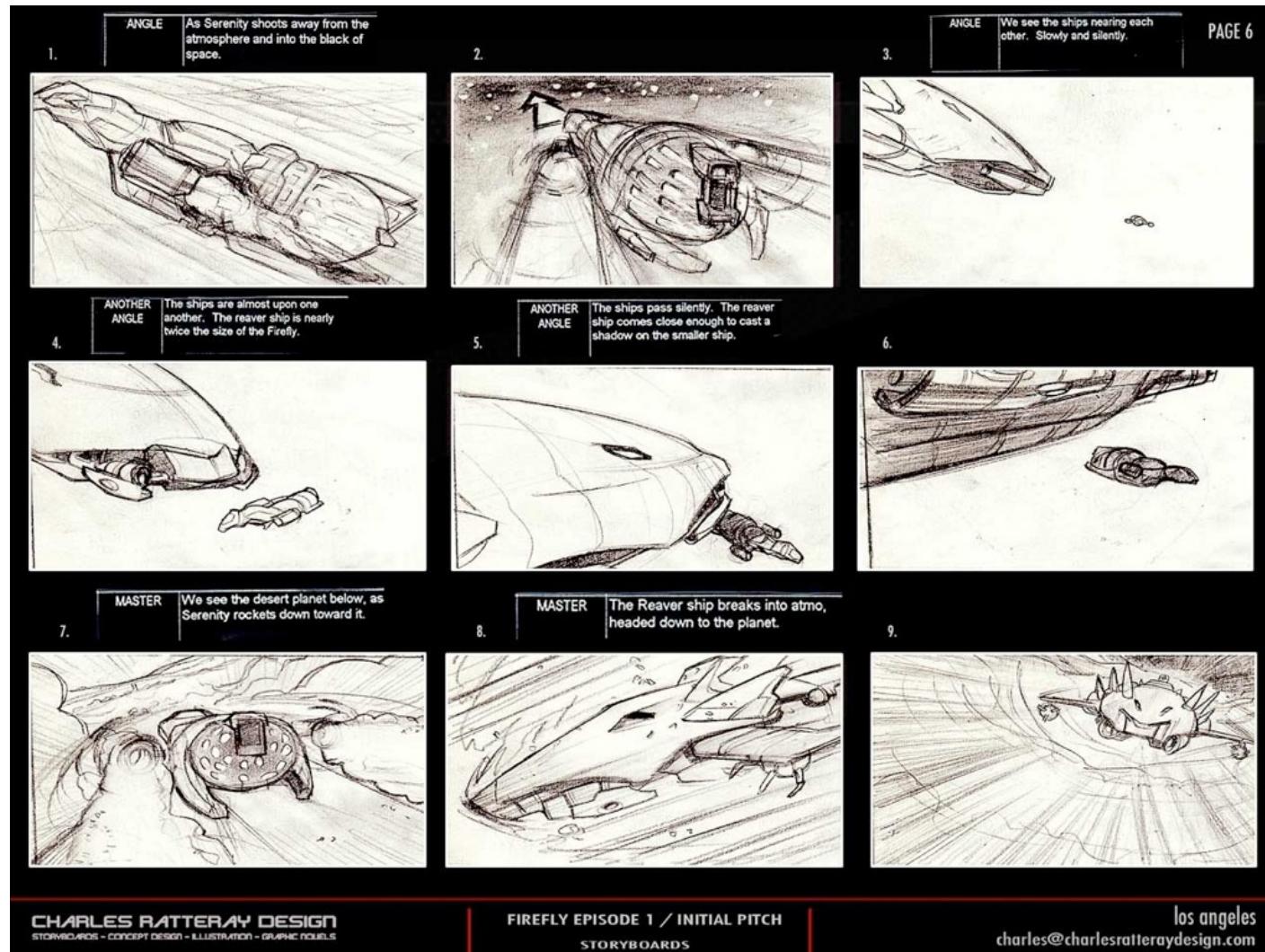
Use Cases

- A use case describes a typical user activity from start to end. A good set of use cases motivates a design, and provides a basis for testing.
- A use case will consist of
 - Definition of the user's goal
 - List of pre-conditions (e.g. information needed before the task can start)
 - Criteria for successful completion
 - List of main steps in the activity flow
 - Any extensions/alternatives to this use case
- This is also called *activity based planning*
- More generally, “*Use cases are a technique for organising and eliciting requirements*” and “*Use cases appear in the UML in the form of use case diagrams*” [10]



From Activity-centred to Designing Experiences

- Storyboard – a sequence of images for the purpose of visualising the script in film production



charlesratteraydesign.com/wp/project/storyboards-firefly-initial-pitch-boards-episode-one/

Designing Experiences: overall experience

- IDEO project for Amtrack's high-speed trains – Acela Express
- Brief: design the interior of the cabin
- Work: realised that the actual rail journey is only a part of the overall experience and the whole process needs to be designed together, including
 - passenger's decision to travel by train
 - buying tickets
 - arriving at the station
 - waiting for the departure
 - travelling
 - arrival to the destination station
 - transfer to the final destination



www.ideo.com

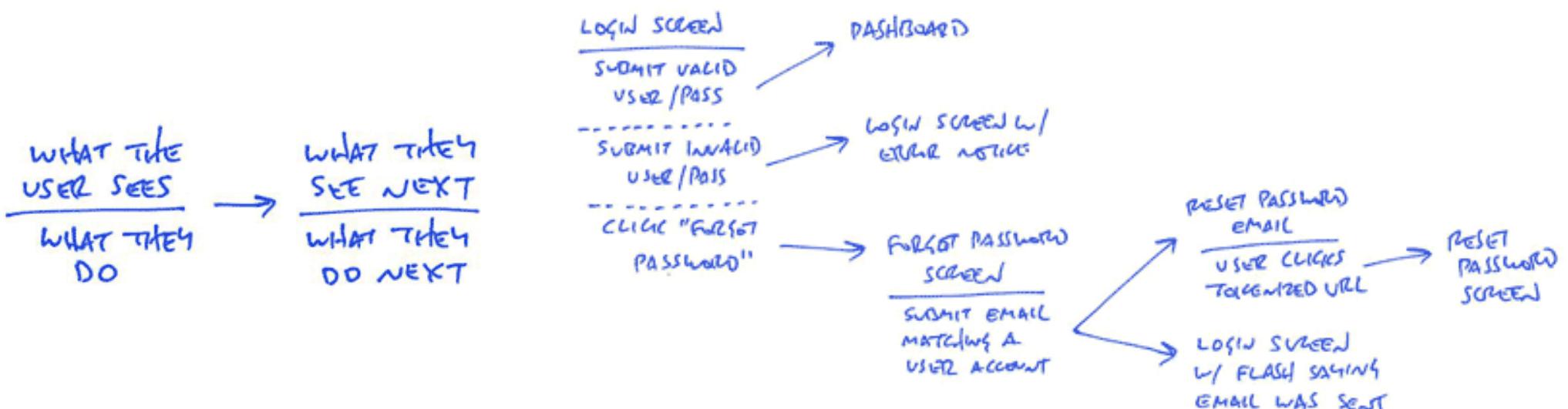
- Unfortunately, over time, the service has failed to live up to the standard (e.g. the speed limited by old tracks) – need to consider practical implications, especially over the long lifetime of the system

Designing Experiences

- Applied to software applications, there are the following stages of user experience
- **Discovery** – a person is realising a need and finding the application to address it
- **Installation** – obtaining a copy of the application and setting it up
- **First experience** – “can I do what I downloaded it for or do I need to read instructions?”, being guided through the first use, building a conceptual model of the application
- **Repeat experience** – “I know what it can do, can I do what do most often in the quickest possible way?”
- **Error handling** – what went wrong, how to recover
- **Help reference** – could facilitate the first use, help to discover advanced functionality or just serve as a reference for less common operations
- **Maintenance** – updates: notifications about availability, applying updates
- **Support** – if needed more help, user groups, support email/phone
- **Feedback** – providing a back channel, identifying issues and features requests

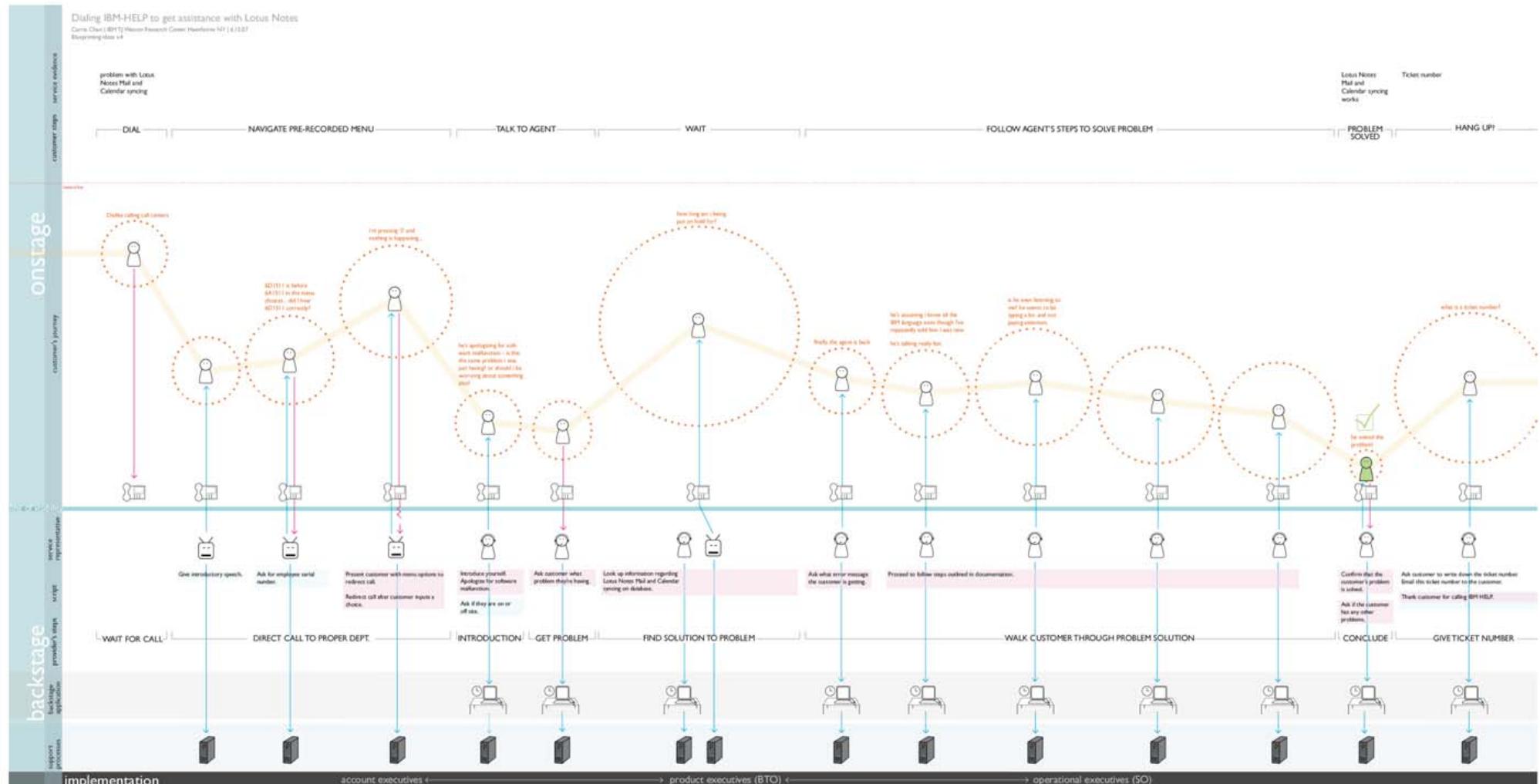
UI Flows

- Need to capture a Screen and Actions possible from it
- 37 Signals' notation:
 - Screens (at the top)
 - Actions (under the solid line)
 - Alternative actions (under dotted lines)
 - Arrows indicate the flow
- More details
 - 37signals.com/svn/posts/1926-a-shorthand-for-designing-ui-flows



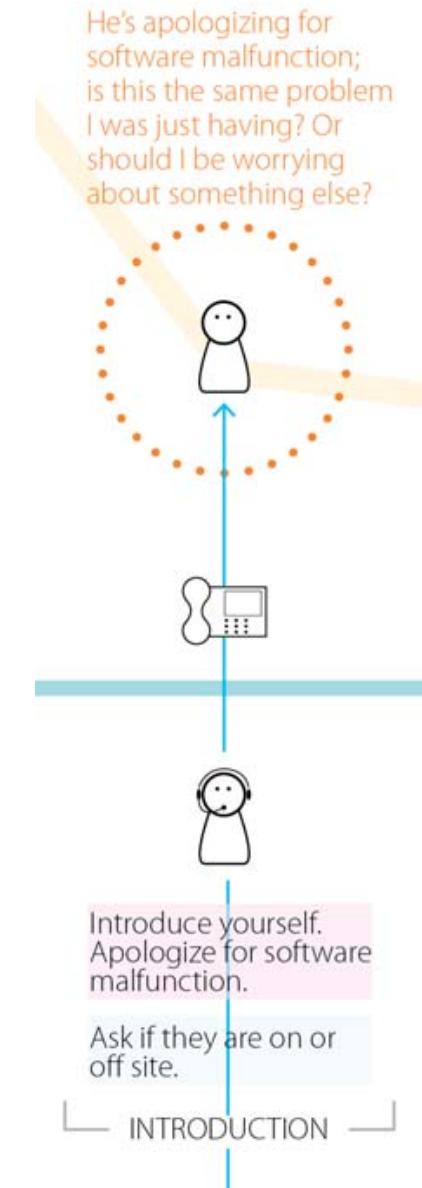
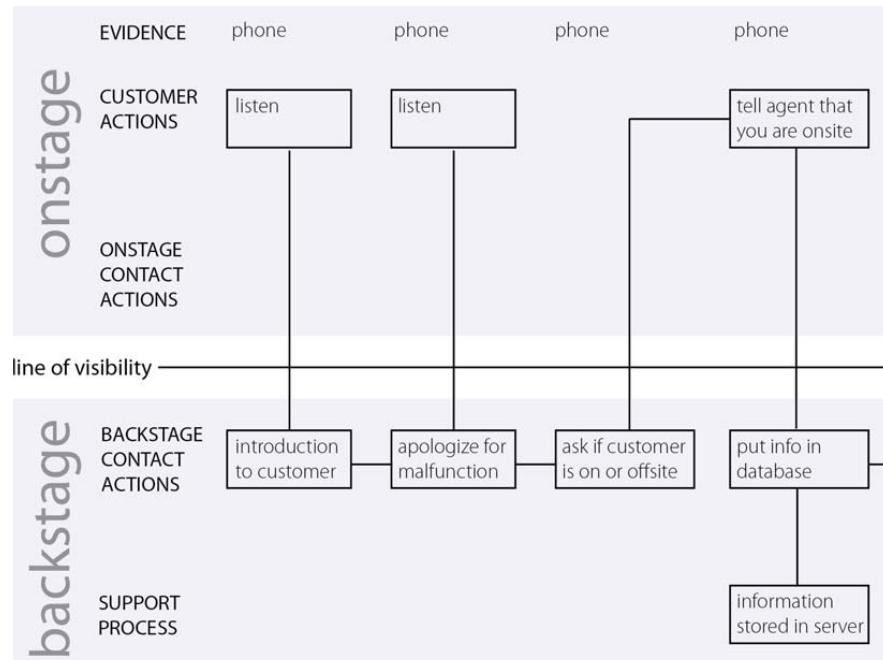
From Designing Experiences to Emotional Design

- Service Blueprinting: When Customer Satisfaction Numbers are not enough
 - Susan L. Spraragen, IBM Watson Research and Carrie Chan, School of Design, MMC110 Carnegie Mellon University



Designing Experiences: Emotions matter!

Service Blueprinting: from traditional to enhanced



Example

- Consider a program to make and send greetings cards
- A simple functional approach would just define the functions that are needed:
 - Create a new blank card
 - Select a style from a library
 - Add/edit text to a card
 - Add a graphic to card
 - Email card or print it out
- A UI to implement this might then consist of a few standard menus
 - An experienced user would have no problem with this. However, a naive user faced with a blank screen and just a few generic menu commands such as the above would be lost
 - A tool bar with graphic icons might help but real novices need guidance.

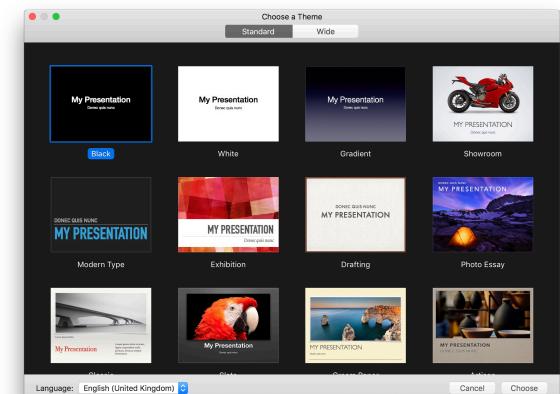


Wizards

- Wizards guide a user through pre-stored use cases one step at a time
- In case of the greeting card, the Wizard would lead the user through a sequence of steps
- Step 1: What do you want to do?
 - 1.Send a birthday card?
 - 2.Send an anniversary card?
 - 3.Send a party invitation?
 - 4.Start with a blank card?
 - User chooses (1)
- Step 2: Select a background
 - User selects from a palette of birthday card designs
- Step 3: Enter desired greeting
 - User types greeting into a text box
- Etc



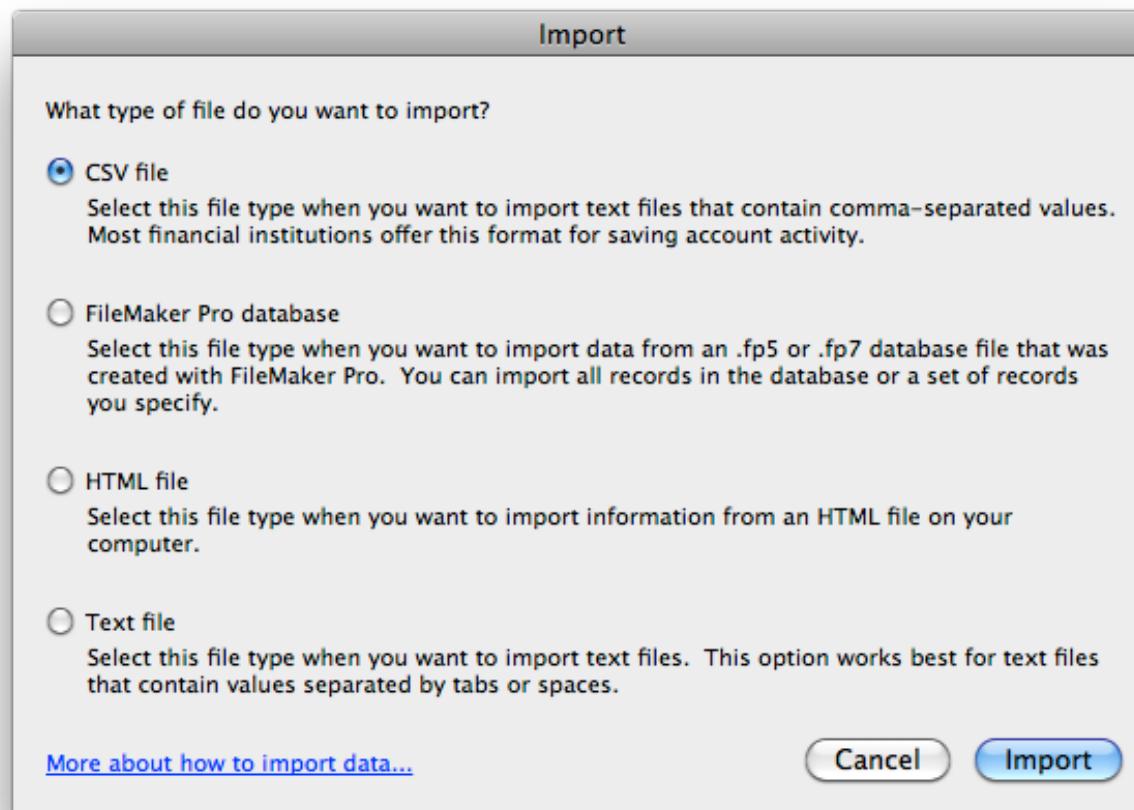
© New Line Cinema and Wingnut Studios



Apple Keynote

Import Wizard

- Allows to guide the user through a process with a multiple decision tree, to help the software do its job better



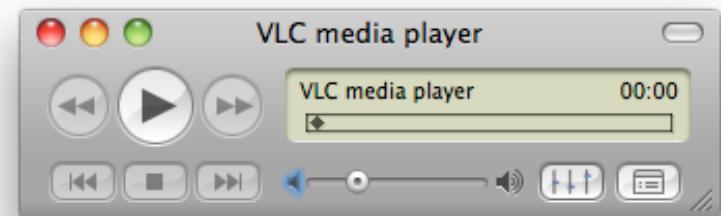
Online Wizards

- Hotel reservation process at Booking.com
- Allows to guide the user throughout the room reservation process

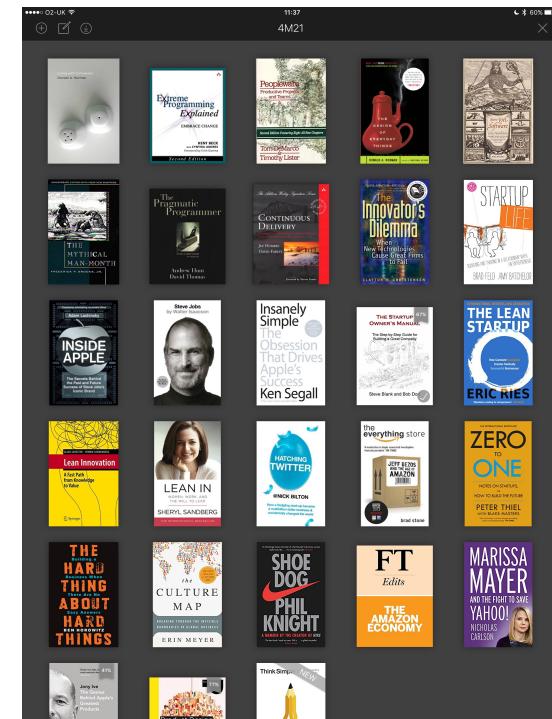
1. Your room	2. Your details	3. Your address	4. Booked!
<p>Your booking</p>  Hotel Continental Zurich - MGallery Collection Location: Zürich Check-In: Sunday, 4 March 2012 Check-Out: Monday, 5 March 2012 For: 1 night, 1 room, max. 2 people.  Sign in for speedy booking First name <input type="text" value="James"/> Last name <input type="text" value="Smith"/> <small>Please fill in a valid email address</small> <small>Email address <input checked="" type="checkbox"/> You'll receive a confirmation email</small> <small>Confirm email address <input type="text"/></small> Room: Standard Room with 1 Queen Bed - Non refundable Full guest name <input type="text" value="James Smith"/> Max persons 2 guests Smoking Non Smoking Only Special Requests <small>Please write your requests in English or in the language of the hotel.</small> <input type="text"/>			
1. Your room	2. Your details	3. Your address	4. Booked!
<p>Your booking</p>  Hotel Continental Zurich - MGallery Collection Location: Zürich Check-In: Sunday, 4 March 2012 Check-Out: Monday, 5 March 2012 For: 1 night, 1 room, max. 2 people.  Your address Name James Smith E-mail smith@nowhere.com Address <input type="text"/> City <input type="text"/> Zip/Post code <input type="text"/> Country United Kingdom Telephone <input type="text"/> So the hotel can reach you <p><small>i Your credit card is needed because it will <u>guarantee your booking</u>. Hotel Continental Zurich - MGallery Collection may charge your credit card once you confirm your booking.</small></p> <p>Credit card type <input type="button" value="-- Select --"/> Credit card number <input type="text"/> Card holder's name <input type="text"/> Expiry date <input type="button" value="01"/> / <input type="button" value="2012"/> CVC-code <input type="text"/> <small>I agree with the booking conditions and general terms by making the reservation.</small></p> <p>Make the reservation <small>No booking fees!</small></p>			
<p>CVC-code</p>  <p>You can find the CVC-code on the back of your credit card. It's the last 3 digits printed on the signature strip.</p>  <p>Do you have American Express? The CVC-code is on the front of your credit card. It's the 4 digits above your credit card number.</p>			

User Conceptual Model

- User conceptual model is the user's mental model of what is happening
- Note that different users will have different mental models. A good UI design encourages every user to adopt the same mental model
- Eliciting user models:
 - usually 5-6 people is enough to gather a consensus
 - present each user with a scenario and a task to complete
 - use simple prototypes, screen mock-ups or just sketches to describe scenario
 - ask user questions to understand how they would set about the task
 - hence infer their mental model
- Using a metaphor can help a user align their conceptual model with the way that the program actually works



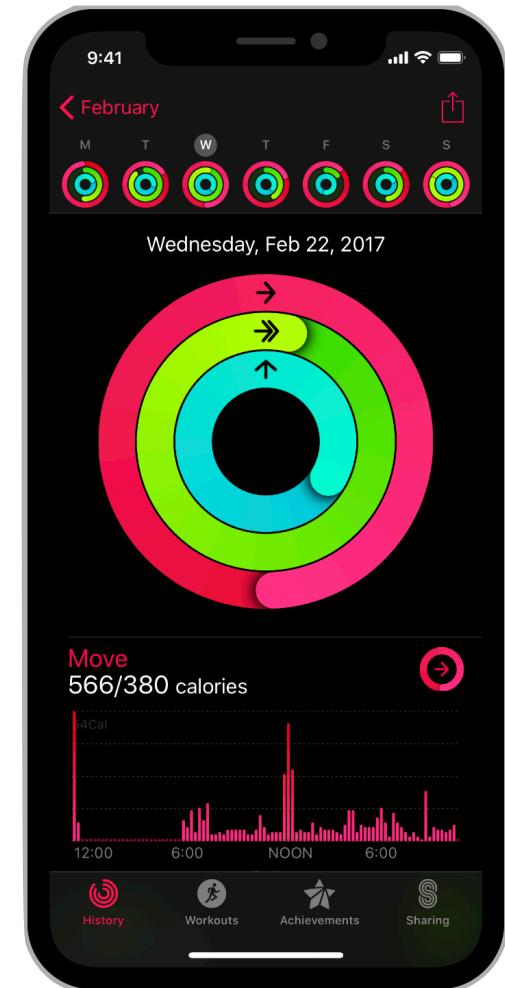
www.videolan.org/vlc/



Amazon's Kindle for iPad

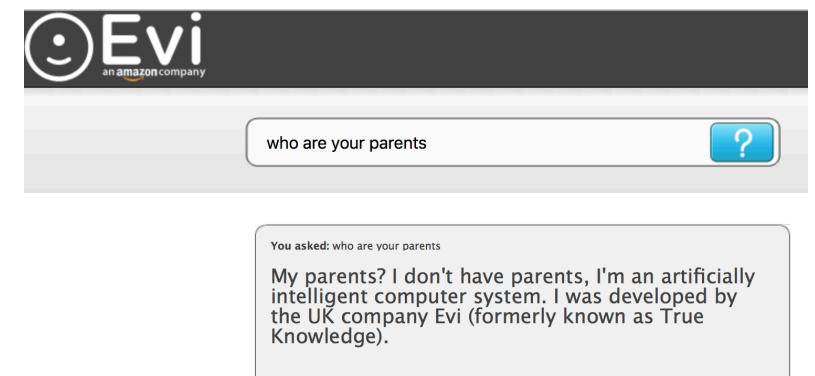
Information Presentation

- All interactive systems need some way of presenting information to the user
- It is good design practice to keep the presentation separate from the information itself; this is sometimes called the document-view architecture
- The representation on the screen can be changed without having to change the underlying architecture
- There is a vast choice of display styles: bar charts, thermometers, dials, as well as just displaying text and numbers.
 - In general, graphical display methods should be used as they can present trends and relative values. A digital display should only be used when precision is required, and even then graphical highlighting could be used to highlight changes
- Colour is very useful for highlighting or increasing clarity
 - However, don't use more than 4-5 colours, and be consistent with their use (e.g. red for danger)



User Input

- Direct manipulation: typing, gestures, e.g. Media Players, Graphic programs
 - Fast, intuitive and easy to learn, but hard to implement and only suitable when there is an appropriate visual metaphor
- Menu selection, e.g. Mobile phone
 - Avoids user error and needs little typing. Could be tedious for experienced users and is not suited to many complex options
- Form fill-in, e.g. Stock control
 - Good for data entry, requires significant screen space
- Command language, e.g. Travel booking system
 - Powerful and flexible for experienced users, hard to learn
- Natural language, e.g. Evi (Amazon) or Siri for iPhone
 - Accessible and easily extended, could be unreliable but will get better as the computing power and connectivity of the devices increase



Dealing with Errors

- “Life has no error messages”
- Computer/Video games have no error messages (mostly ;)
- In real world learning could be achieved by trial and error because possibilities of alternative actions are limited
- CrontoSign device has three buttons
 - the big button in the middle turns it on
 - some users might try pressing the smaller buttons first, but would not stop there – 100% would discover that the middle button turns the device on
- However, if there are too many choices the user will give up
- In complex systems, errors are “wonderful teaching opportunities”, e.g. dynamic input validation in web forms, also “most people want “just-in-time” learning



Payee name:

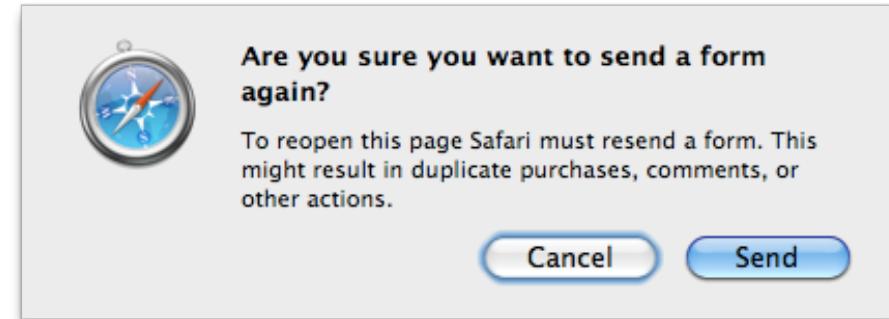
ABCDE

* Some of the characters used
are not allowed.

Error Messages

- Error messages should be there to help the user

- polite
- concise
- consistent
- constructive
- appropriate to user's skill level
- accurate
- informative
- appropriate to setting

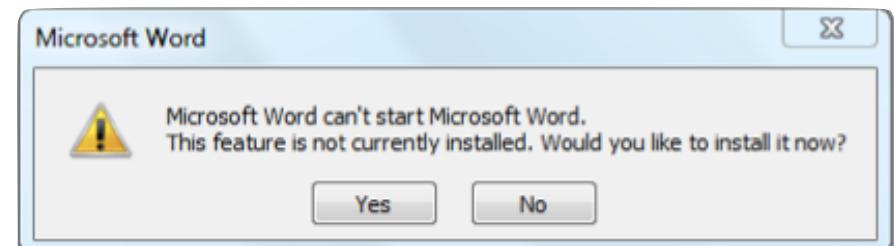
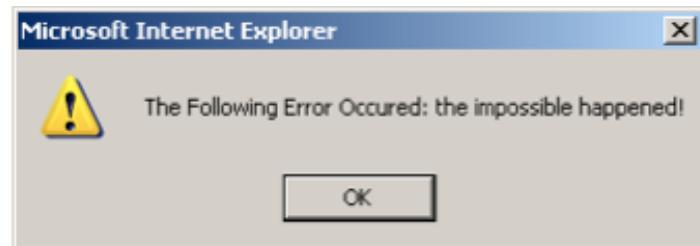


- How NOT to do it:



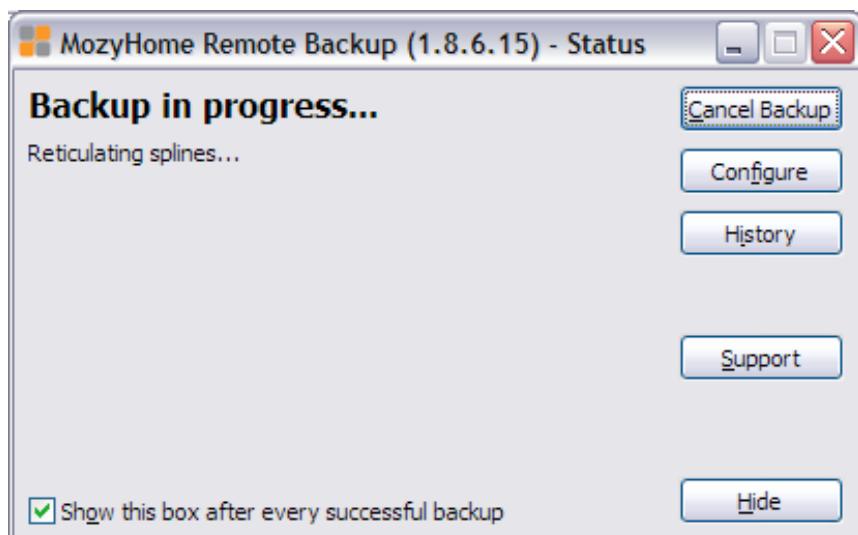
Server Error in '/' Application.

Unknown error: Go Away

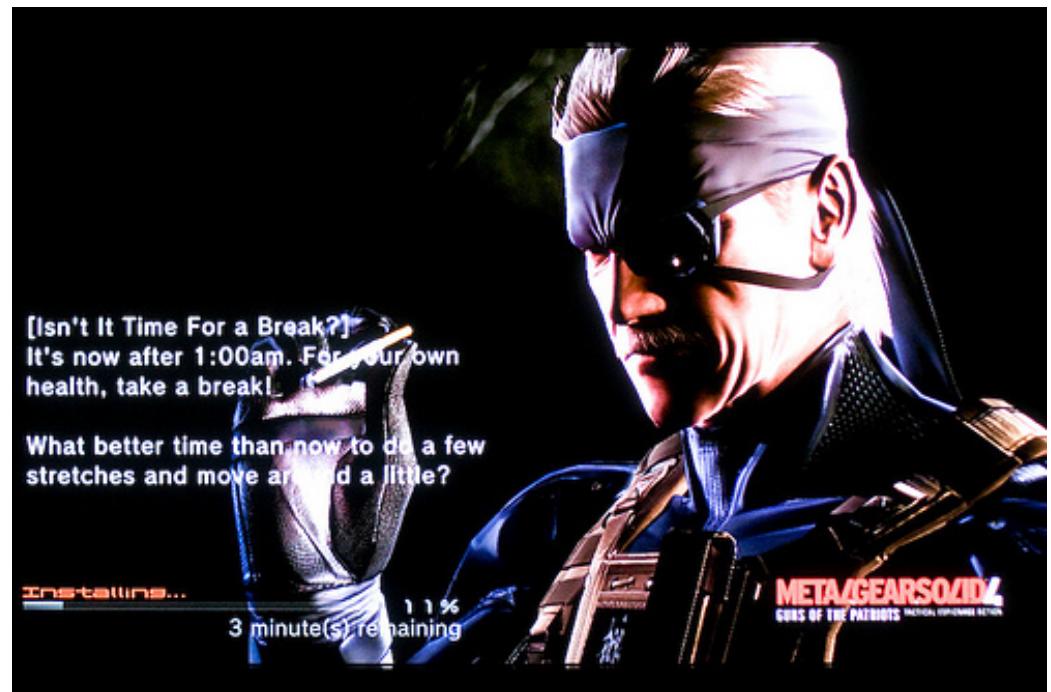


Progress Indicator / Waiting Screen

- Waiting is the most frustrating part of any user experience
- Providing a good progress indicator helps to elevate this pain and it should be:
 - dynamic: confirms that something is happening
 - informative: how much is remaining/ has been done



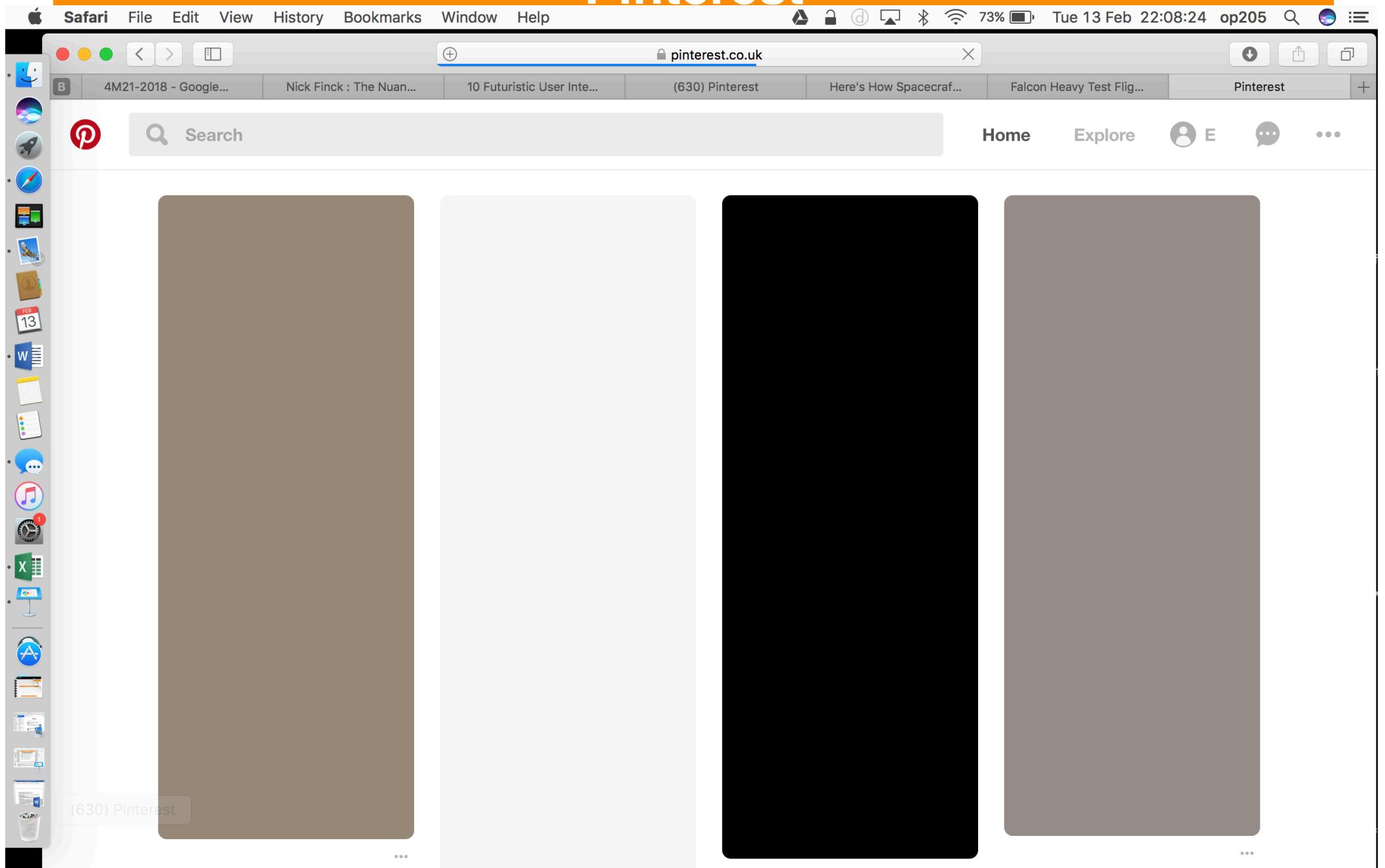
MozyHome, © Decho Corp



Metal Gear Solid 4, © Konami

<http://www.ign.com/blogs/tengu230/2011/07/14/the-unnoticed-video-game-loading-screens>

Pinterest



Safari File Edit View History Bookmarks Window Help

73% Tue 13 Feb 22:08:27 op205

pinterest.co.uk

4M21-2018 - Google... Nick Finck : The Nuan... 10 Futuristic User Inte... (630) Pinterest Here's How Spacecraf... Falcon Heavy Test Flig... (630) Pinterest

Search Home Explore 9+ ...

6+ Sensory Bin ideas!
All from the Dollar Store

change IN seconds
MOTIVATION TO LOSE WEIGHT

Fat Burning Evening Workout

50 Jumping Jacks

30 Squats

20 High Knees

9+

Ball Maze Game

FrugalFun4Boys.com

(630) Pinterest

User Interface Evolution

Past



www.oldcomputers.net



computinghistory.org.uk

- Keyboard
- 280x192 pixels screen
- 9" 512x324 pixels screen
- Mouse



Present



Future?



- Keyboard
- 27" 5120x2880 "retina" screen
- Mouse
- Virtual Keyboard
- 4.7" 1334x750 "retina" screen
- Multi-touch
- Shake
- Voice
- 1.5" 390x312 "retina" screen
- Force touch
- Digital crown
- 25" (equivalent) 640x360 screen
- Touch
- Wink
- Nod

User Experience in Context



bloomberg.com



commerzbank.com



- **Desktop**

- complex tasks
- long duration
- rich presentation and functionality, all accessible in the same time

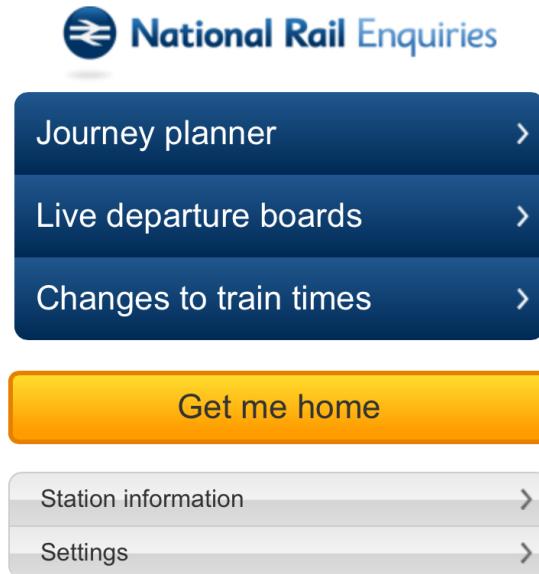
- **Mobile**

- simple tasks
- short duration/ on the go
- selected presentation and features, prioritised to match most likely user actions

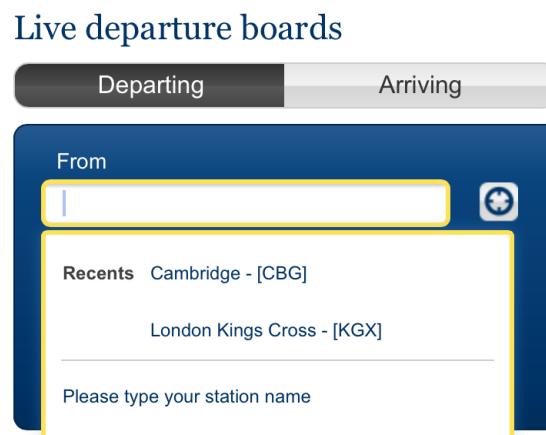
- **Wearable**

- notify and suggest an action
- extremely short duration
- even more selected/ prioritised presentation of informations and features
- hierarchical UI flow

Example: National Rail App



- Start page actions
 - Journey planner
 - Live departure board
 - Changes to train times
 - “Get me home”



- Live departure boards
 - smart suggestions

Graphical User Interfaces Frameworks

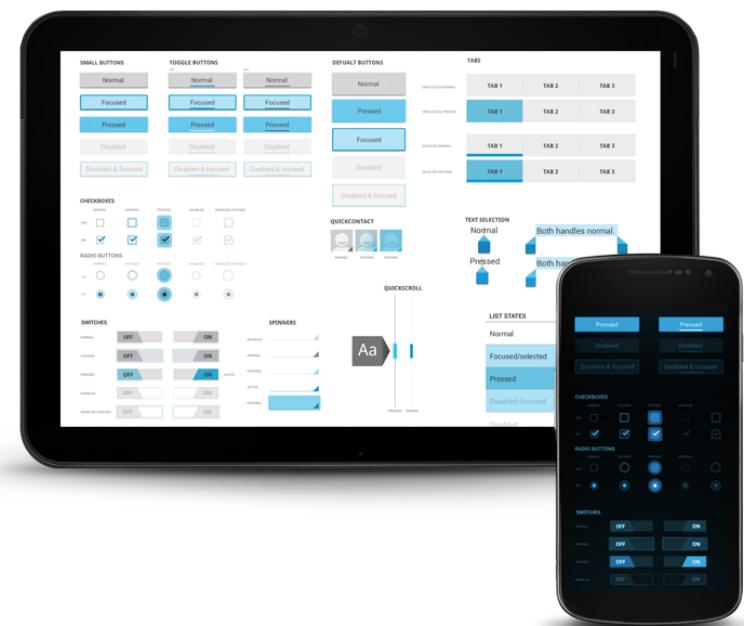
- GUIs are often implemented using application framework class libraries
- Pioneered by Apple via “MacApp”, then copied by Microsoft with the MFC (Microsoft Foundation Class) library
- Basic functionality is coded in base classes e.g. windows, menus, scrollbars, buttons, etc
- “Look and Feel” is then dictated by class library so all applications built using the library adopt similar devices and metaphors
- Developer can customises base classes to achieve required functionality e.g.
 - decorator pattern used to add widgets to windows
 - observer pattern used for synchronising multiple views of an object
 - today there are literally hundreds of GUI toolkits available targeted at specific languages, operating systems, cross-platform, etc. Examples are Swing, TK, GTK+, OpenUI, Zinc, Motif, Qt, XVT, FLTK, Wx

GUI Frameworks

- GUI Frameworks could be provided by the target platform: Android, iOS or be a part of the language libraries: Java Swing
- Each framework can have its own concepts and design principles that should be followed
- Each framework allows to extend standard components (Views, Buttons, Lists) to implement custom behaviour



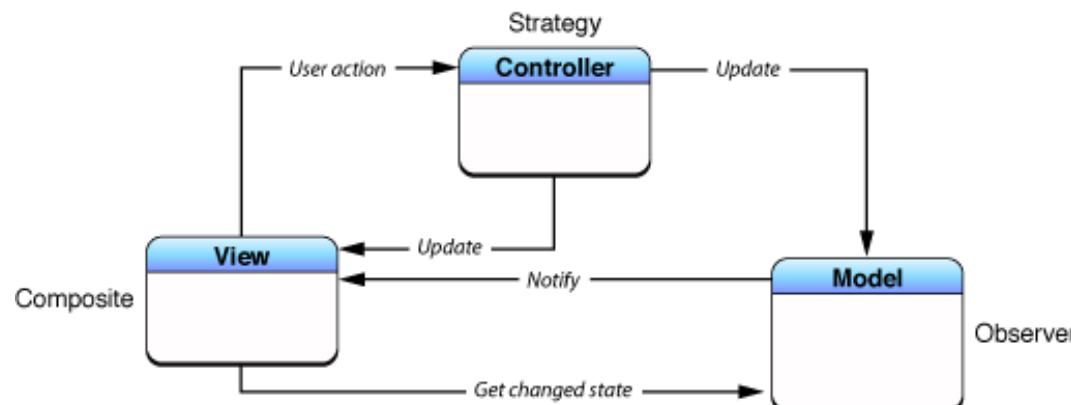
<https://developer.apple.com/ios/human-interface-guidelines/overview/themes/>



<https://material.io/guidelines/#introduction-principles>

What is behind?

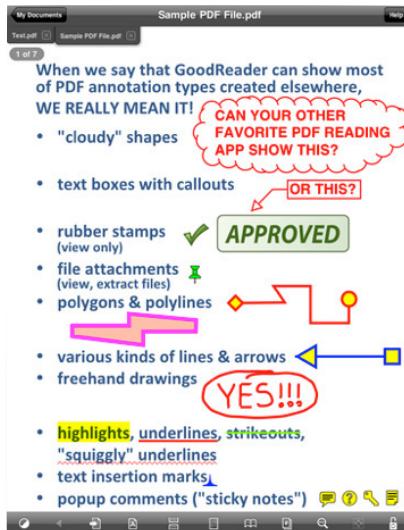
- Model View Controller (MVC) is the common pattern that separates the data, its presentation and the way it is manipulated by the user
- Observer design pattern is a key part of MVC
- Model – an abstraction of the data that provides ways of accessing and manipulating it
- View – presents a specific purpose view onto the Model, captures User's input and passes it on to the Controller
- Controller – interprets User's input and invokes corresponding actions on the Model



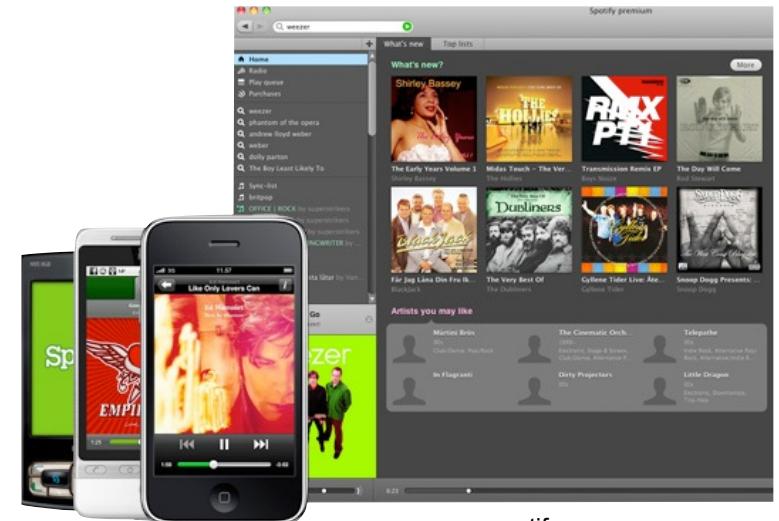
developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/CocoaFundamentals/CocoaDesignPatterns/CocoaDesignPatterns.html##//apple_ref/doc/uid/TP40002974-CH6-SW1

User Interface Summary

- The key ingredient to a successful UI is to ensure that the user adopts a mental model which is consistent with the underlying program model
- To achieve this a compromise is needed
 - use metaphors, wizards, etc to guide the user to the right model
 - modify the program model to match user expectations
- Use cases help design the UI and guide testing
- Many toolkits are available for rapid GUI construction
- Good Designs make Winning Products and make Users Happy!



Instagram by Burbn
instagr.am



Example Questions

Question 2.

Look at the products and processes around you and identify three examples of bad designs and good designs. Identify pitfalls to avoid and features that make the product/processes work well. Apply design theories and suggest improvements.

Question 3.

Open one of the popular websites or mobile phone app and identify both the good and bad user experiences. Apply the basic concepts of user experience design to suggest improvements.