

# **Comp 388/441 - Human-Computer Interface Design**

Week 9 - 17th March 2016

Dr Nick Hayward

# Usability - I

- may consider an application, product, software as usable if it fulfills
  - *can be efficiently operated*
  - *provides an overall pleasant usage experience*
  - *can be easily learned*
- often difficult to judge the usability of a product etc
  - *rules are often subjective in nature relative to usability*
- each rule may vary greatly from user to user due to
  - *different skill sets*
  - *existing knowledge*
  - *previous experience*
- user's expectations, opinions, general preferences affect perception of usability
- some users are naturally more curious, patient, and persistent
- user experience may also be influenced by
  - *attitudes and experiences of friends, contemporaries...*
  - *general moods*
  - *stress levels, fatigue, distractions*

## Usability - Scissors



Source - RightLeftRightWrong

## Usability - Video

left-handed in a right-handed world...

What It's Like To Be Left-Handed In A Right-Handed World ⌚ ➦



What it's like to be left-handed in a right-handed world - Source: YouTube

## Usability - 2

### Ease of Learning

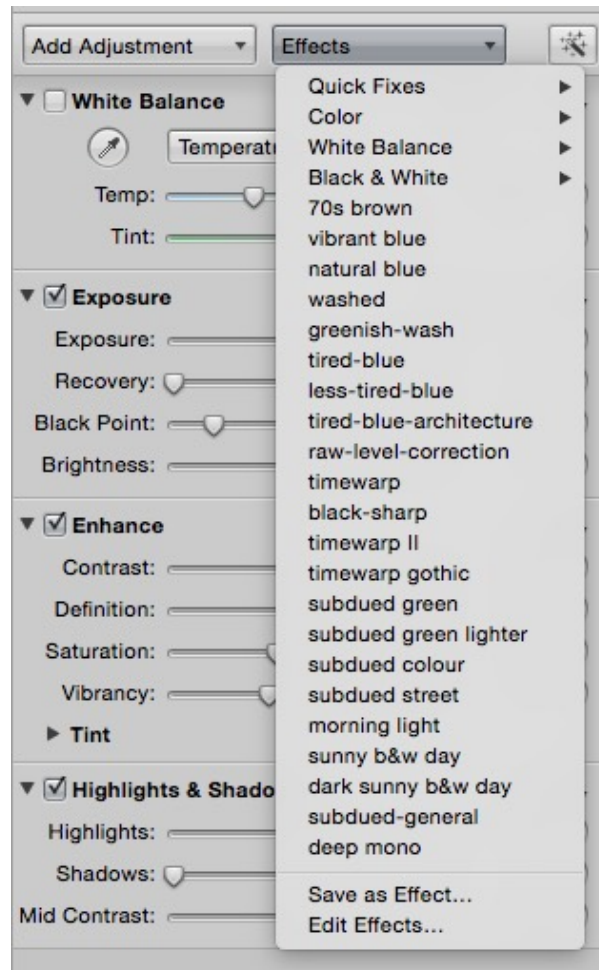
- clear functionality and general operations with appropriate visible controls, labels...
- clear navigation options and paths, plus user's current location
- minimum memorisation and recall for sequences, commands, actions
  - *easy to remember and recall*
- product, application encourages exploration and experimentation
- mistakes are easily recoverable, and operations can be retried if necessary
- assistance and help is easily accessed, clear, correct, and relevant
- consistent interaction behaviour, visual layout, terminology
  - *helps encourage correct user mental model*
- limited surprises for application behaviour and usage
  - *less for the user to learn...*
- where possible, a user is guided through steps to complete complex tasks...
- clear feedback is provided when a user performs an action
- current status of the system is clearly presented and labelled
- application, system, or product should form a coherent whole

## Usability - 3

### Efficiency

- straightforward, easy for an experienced user to repeat actions or complete tasks
- minimal deliberate or strenuous thinking to perform routine application tasks
- enable and encourage a user to achieve a state of **flow**
- allow a skilled user to achieve a low error rate
  - *clear notification and detection of limited errors and mistakes*
- stable performance and reliability to prevent delays and hindrances
- minimal, if any, surprises and inconsistencies in interaction and design patterns

## Usability - Preset Effects



Source - Aperture

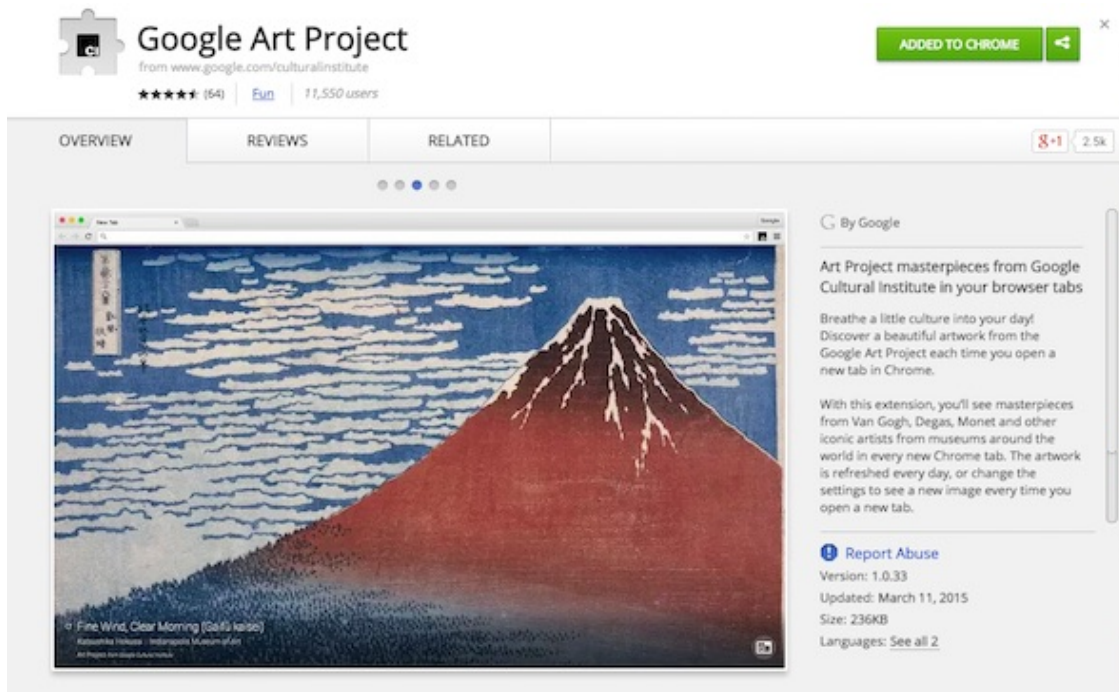
## Usability - 4

### Experience

- possible to consider a product or application relative to its experience
  - *whether it is a pleasant experience or not...*
- is the application's design and interface pleasant and appealing for its users
- does it promote and encourage positive productivity
- eg: if we consider games, does the application's experience
  - *provide enjoyment for its users*
  - *challenge them relative to their abilities*
  - *provide general entertainment and distraction*
- does the user feel rewarded and positive for tasks and actions completed
- again, is the product stable, reliable, and trusted by users
- likewise, are the delays sufficiently limited to avoid frustrations for users
- is the product free of unnecessary annoyances and frustrations
  - *help promote user satisfaction, reduce cognitive overload, and help achieve and maintain a sense of flow for users*



## Usability - Pleasing concepts



Source - Google Art Project

# User experience (UX) - Part I

- broad and over-arching concept
- need to consider many disparate concepts
  - *user's reaction, both positive and negative*
  - *user's general experience with the application including*
    - design and interface
    - potential results and outcomes
  - *general functionality and what an application can do for a user*
  - *does the application, product etc solve a defined problem?*
  - *what can an application help a user to achieve?*
  - *what entertainment value does the application etc provide?*
- software application UX also influenced by acquisition
  - *was it easy to find, download, install, update?*

## User Experience (UX) - Linux Installs

```
* Starting dcron ... [ ok ]
/etc/conf.d/net: line 6: syntax error near unexpected token `"dhcp"'
/etc/conf.d/net: line 6: `config-eth0=( "dhcp" )'
* Starting eth0
* Configuration not set for eth0 - assuming DHCP
* Bringing up eth0
* dhcp
* network interface eth0 does not exist
* Please verify hardware or kernel module (driver) [ !! ]
/etc/conf.d/net: line 6: syntax error near unexpected token `"dhcp"'
/etc/conf.d/net: line 6: `config-eth0=( "dhcp" )'
* Starting eth1
* Configuration not set for eth1 - assuming DHCP
* Bringing up eth1
* dhcp
* network interface eth1 does not exist
* Please verify hardware or kernel module (driver) [ !! ]
* ERROR: cannot start netmount as net.eth0 could not start
* ERROR: cannot start sshd as net.eth0 could not start
* Starting local ... [ ok ]

This is gentoo.localdomain (Linux i686 2.6.36-gentoo-r5) 14:12:19
gentoo login: _
```

Source - Gentoo Linux

## User experience (UX) - Part 2

- user's identification of an **acceptable** product
  - *sense of usability and product preferences*
- Shackel, B. 1991.
  - *product's utility, usability, attraction relative to involved costs...*
- product considered not acceptable vast majority of users seek market alternatives
- UX inherently important aspect of goal to develop and provision successful application...

## User Experience (UX) - Windows

Windows 7 UI



Windows 8: Metro UI



Source - Windows Comparison

## Considering an app's design



Source - All Posters

# Designing our app - I

## Considerations - Part I

- tasks and activities a user can and should be able to perform with the product
  - *ie: what is the considered scope of the product's functionality?*
- as we consider each task, how will the interaction develop and be processed?
  - *in effect, what are the expected steps and actions for the user and the product?*
- we need to consider carefully the overall visual style or appearance of the application
  - *eg: visual design and layout for the basic page templates or screen layout - fonts, colours, typography and iconography, any branding...*
- what are the defined **places** in our application?
  - *eg: pages for a website, navigation controllers and panels for mobile apps, levels in games, and so on...*
- how does our user actually navigate between these **places** within our application?
- as we consider further our app's places, what content and layout will be presented to the user in each *place*.
  - *which controls are available, how will they be presented, arranged, and so on?*

## Designing our app - 2

### Considerations - Part 2

- how will the user interact with these controls?
  - *ie: just mouse and keyboard, is touch accepted?*
  - *are there behaviours associated with these controls?*
- are there any events within our application that are not triggered by the user?
  - *eg: timer driven events, remote calls and services, backup protocols, automatic updates...*
  - *are any behaviours actioned during such events?*
- does the application store, request, manage any data?
  - *what type of data, where, format, protocols, services...*
  - *how do we present this data on-screen and to the user?*
- is there a naming scheme for interface and interaction elements?
  - *eg: data, elements, places, objects, controls, navigation, and any other pertinent concepts...*



## Designing our app - 3

### Considerations - Part 3

- error handling scheme for the app
  - *how will the user be informed? will the user have the option to gracefully recover from errors etc?*
- are there defined user roles in the app?
  - *what actions, privileges are permitted per role?*
- how do our users request or find assistance within the app?
  - *is it an active system or passive? ie: interactive or reference based documentation, tutorials, videos, discussion forums etc...*
- how is the app structured to promote app guidance for users through tasks?
  - *help for the users to work out how the app actually works...*

## Designing our app - 4

- need to engage in a number of related tasks
  - *eg: gathering requirements and their analysis*
- need to understand our user base, the target audience for our app
  - *includes their characteristics, requirements, how they intend to interact with the app*
- as designers and developers we will need to understand
  - *the type of work users want to complete*
  - *the inherent tasks*
  - *the effective problem domain*
- to a lesser degree, this will also require an understanding of the technology requirements
  - *eg: chosen languages, frameworks, device hardware...*
  - *impacts how and what we are able to design and provision for our users*
- need to consider prototypes, mockups, design documentation and specifications, and testing...

## Users and skills - I

- continue to consider our application's users
- primary challenge involves consideration of product development relative to both beginner and advanced users
  - *how to make usable and productive app for all concerned*
  - *comprehensible and learnable for beginners*
  - *do not hinder expert users from optimal productivity*
- carefully consider user skill levels
- be aware of changes to skill levels over time
- aware of practical ways to help our users attain and improve skill levels
- understanding user's skill levels helps application of interaction concepts and principles

## Users and skills - 2

### User categorisation - Part I

- we can often categorise users by application skill levels and aptitude
- **evaluation user**
  - *testing and evaluating an app and not yet committed to its usage*
  - *trying to determine its suitability for their requirements*
  - *no pressing tasks or action at hand*
- **beginner user**
  - *trying to accomplish some tasks with the application*
  - *little or no prior experience with the app's usage*
  - *general feelings of uncertainty and learning by trial and error, general experimentation*
  - *some, but not all, will use the available tutorials, help documentation etc*

## Users and skills - 3

### User categorisation - Part 2

#### ▪ **intermediate user**

- *more confident and experienced user, able to complete most of their required tasks*
- *unlikely they will have explored all of the app's features and options*
- *user comfort and fluency will not have been achieved for the application*
- ***perpetual intermediates***
- *Cooper et al. 2007.*

#### ▪ **expert user**

- *greater application confidence and certainty*
- *awareness of product's domain and advanced options*
- *able to complete tasks with ease, solving problems as they arise...*

#### ▪ **power user**

- *considered an extension of an **expert** user with a fascination of the application*
- *normally enjoys customising the application and testing its limits*

## Users and skills - Video

Your First Script - Apps Script Tutorials



Your First Script - Apps Script Tutorials - Source: YouTube

## Users and skills - 4

### Development of skills

- user classification is inherently a simplistic interpretation of skills acquisition and development
- many disparate factors influence development of skills. For example,
  - **domain knowledge**
    - *assumption of underlying, pre-existing knowledge for a given application's scope*
  - **general computing skills and knowledge**
    - *many applications assume general computing skills and knowledge*
    - *eg: simple ability to use similar applications*
    - *ability to use their chosen mode and tools of interaction*
  - **general intelligence and reasoning abilities**
    - *an assumption of general reasoning and extrapolation skills*
    - *ability to read and understand help documentation...*
  - **persistence, motivation, and dedication**
    - *some users will, of course, give up when faced with problems and challenges*
    - *others are more persistent and will try to solve a problem or issue*
    - *gamification and rewards may help this issue...*

## Users and skills - 5

### Assumptions

- consider basic assumptions about users' minimum required skills and knowledge
- often dependent upon goals and functionality of the product, application...
- some inherent assumption of skills for your application
  - *eg: user will be able to use a keyboard, mouse, touchscreen...*
  - *basic level of verbal, reasoning, and mathematical knowledge*
- valid user testing important relative to such assumptions
- testing helps define and highlight unrealistic design choices and assumptions
- modify assumptions and design in response to testing feedback
  - *re-consideration and re-design may be necessary*



## Users and skills - assumptions

- assumption of Domain knowledge - Documenta Latina
- gaming and applications
  - eg: *Royal Game of Ur*



Source - Royal Game of Ur British Museum

## Users and skills - 6

### Skill levels and design - Part I

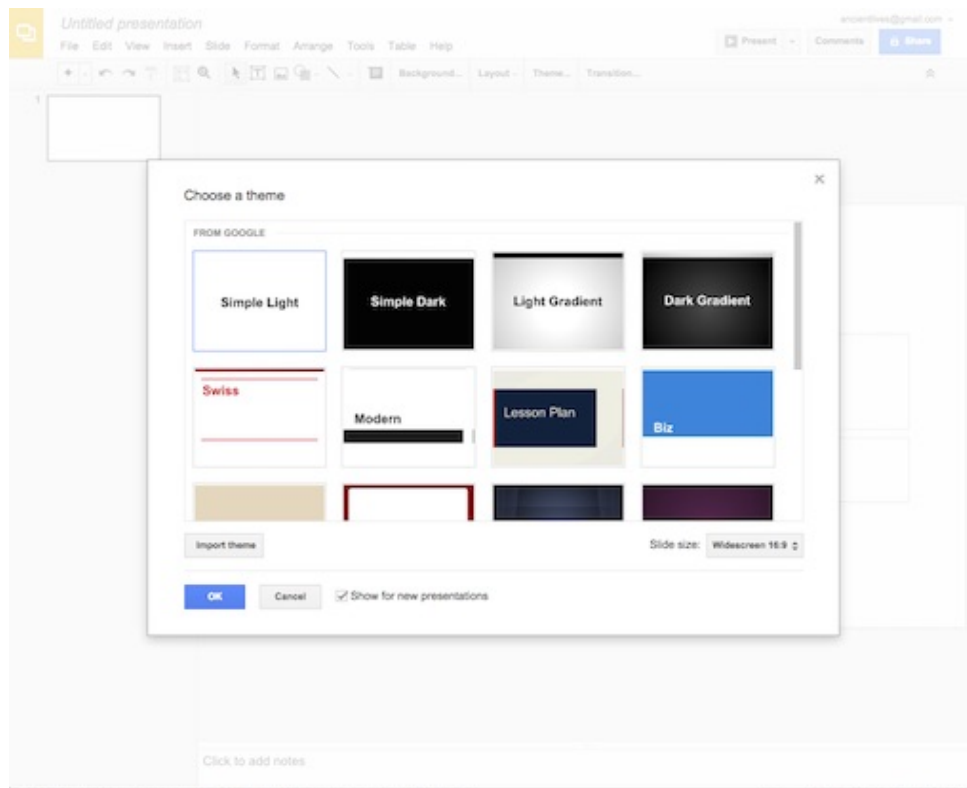
#### ▪ **evaluators**

- *design needs to present good first impression, be pleasing overall, and inviting*
- *should not give the impression of being overly complex*
- *introductory material, such as demo video or guided tour with step-by-step instructions*
- *sample files, demo material allows users to test functionality and see what is possible*

#### ▪ **beginners**

- *functionally easy for our users to learn and discover an application*
- *eg: offer wizard style guidance to create an initial project, document*
- *easy undo/redo errors and mistakes - hopefully promotes experimentation in the app*
- *in-depth tutorials and intro guides, such as manuals, help videos, online help*

## Users and skills - getting started



Source - Google Slides

# Users and skills - 7

## Skill levels and design - Part 2

### ▪ **intermediate**

- *in addition to the above considerations*
- *fully indexed and searchable help resources*
- *allow users to quickly find exactly what they need*
- *online forums and social options and interaction promote sense of community*

### ▪ **expert**

- *quick completion of tasks with maximum efficiency*
- *provide shortcut options, keys, and greater customisation options*
- *bypass and limit beginner tools, wizards, menus etc...*

### ▪ **power**

- *allow greater freedom for users and interaction*
- *user developed scripts, plugins, add-ons*
- *developer tools, APIs, discussion forums, manuals...*
- *carefully consider security implications*

## Users and skills - 8

### Skills change over time

- familiarity, experience, and comfort with an application often increase a user's skills
- skills tend to improve as follows
  - *improved awareness of the application's options, tools, and capacity*
  - *improved and increased awareness of how to perform tasks, handle special cases successfully*
  - *a much lower rate of errors, issues, and mistakes*
  - *increased rate of productivity and completion, speed, efficiency, and so on...*
  - *a general increase in confidence and greater ease at achieving a sense of flow with the application...*
- might also expect general improvement in quality of work
  - *quality often hard to define, measure, and assess*
  - *easier for procedural tasks and jobs than conceptual*

## Users and skills - 9

Practice makes perfect...

- improve skills through regular practice
- for our applications and products
  - *ensure users practice and repeatedly perform given tasks*
- some application scenarios naturally make it easier for users to practice
- simple act of repetition of regular tasks often mimics regular practice
  - *practice due to necessity*
- *"people generally become skilled in whatever becomes routine for them."*
  - *Card et al. P.188. 1983.*
- **deliberate practice** is the act of intentionally practicing with focused attention
  - *specific goal of improving skill levels, working and training at increasing levels of difficulty*
  - *often requires careful monitoring and evaluation of work and results*
  - *motivation and self-improvement important*

## Users and skills - 10

### Monitoring practice and skills

- **Power Law of Practice** - Card et al. 1983
  - *applies to most mechanical and cognitive skills, not always relative to knowledge acquisition*
- as users gain in experience relative to increased practice
  - *related application performance tends to increase rapidly, then slow to a steady rate*
  - *steady peak normally reflects attained peak performance for the practiced skill*
- lack of practice naturally leads to loss of performance and skill
  - *drop in frequency and intensity of practice*
  - *motor skills do not normally atrophy as quickly as knowledge based skills*
  - *simple to refresh these skills with a period of further training and practice*
- designers need to be aware of this potential for skills atrophy
  - *complex, detailed applications may consider detailed help systems, options*
  - *allow a user to quickly refresh knowledge using practice exercises, tests, incentives...*

## Users and skills - Power Law of Practice



Source - Wikipedia



# Users and skills - I I

## Gaining competence

- practice allows us to determine improvement relative to a given activity
- **four stages of competence** model suggested by Robinson in 1974
- this model suggests the following stages a user may follow to mastering a skill
  - **unconscious incompetence**
    - user is unaware of how bad he or she may be relative to a particular skill
    - may even be unaware that the skill exists
  - **conscious incompetence**
    - as user attempts a given skill, they become increasingly aware of a deficiency of skills
    - realise need to improve that skill through further training, learning, practice...
    - may be a daunting and overwhelming realisation for many users
  - **conscious competence**
    - practice allows a user to engage in training sessions, exercises...
    - effectiveness of such training can vary greatly
    - often dependent upon task itself, suitability of chosen practice and training
  - **unconscious competence**
    - complete a task without really thinking
    - act of working, completing an exercise has become natural to the user
    - do not really need to think about the given act...
- games are a good example of hands-on training and practice

## Users and skills - video

Nintendo 3DS - Brain Age: Concentration Training Launc...  



Nintendo Brain Age: Concentration Training - Source: YouTube

## Principles for usability - I

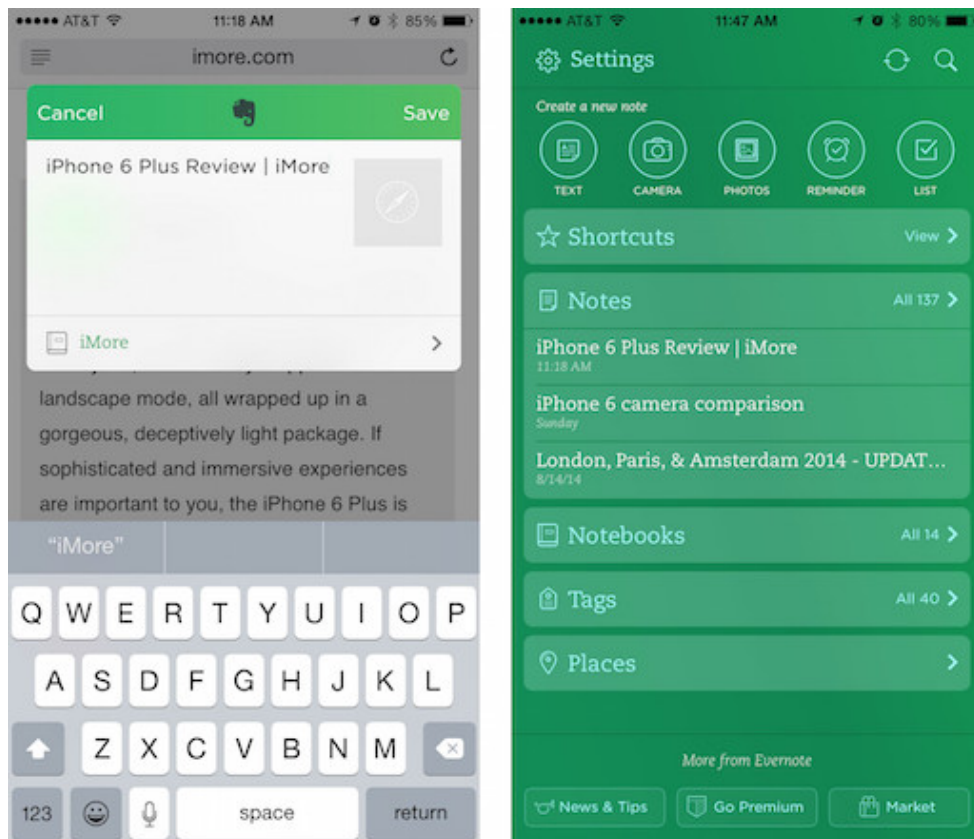
- consider some of the underlying design principles that help guide our designs
- eg: Don Norman's design principles for usability
  - *Norman, D. The Design of Everyday Things. 1988.*
- Norman introduced a set of basic design principles and concepts
  - *consistency*
  - *visibility*
  - *affordance*
  - *mapping*
  - *feedback*
  - *constraints*

## Principles for usability - 2

### Consistency

- one of the primary ways our users learn is by discovering *patterns*
  - *new situations easier to learn by reference to existing patterns of knowledge*
- *Consistency* is key in helping our users recognise and apply such patterns
- overall, things that look the same should perform the same general way
  - *same button, same colour normally infers same pattern of interaction and usage*
- behaviour and actions should also follow a similar pattern
  - *sound, animation, vibration etc should follow a similar pattern for users*
- design inconsistency can cause confusion and overload for our users
- memorisation of exceptions may also increase user resentment towards the app
- internal design and interaction consistency crucial for our users
- external consistency equally important and useful
  - *consistency between OS and app design guidelines*

## Principles for usability - Evernote for iOS 8



Source - Evernote

## Principles for usability - 3

### Visibility

- users normally learn app functionality by visually inspecting the UI
  - *eg: available menus, menu items, icons, buttons, links, tools etc...*
- sequential tasks should be well labelled and navigation obvious
  - **next** button obvious, and highlighted
- usability and learnability naturally improved when options and commands clear and visible
  - *controls should be easily visible, contextually appropriate, logically placed*
- functionality within an application that is not visually represented often hard to discover
  - *keyboard shortcuts often a bad choice for sole command option*
  - *shortcut combinations often noted in visual menus*
- visibility does not, necessarily, infer that all options and functions be graphically represented
- impractical for many complex applications
  - *need for careful, considered design choices and contextual awareness*

## Principles for usability - video

Photoshop: Selecting from a contextua...  



Photoshop: Selecting from a contextual menu - Source: YouTube

## References

- Card, S.K., Moran, T.P. and Newell, A. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates. 1983.
- Robinson, W.L. *Conscious competency - the mark of a competent instructor*. Personnel Journal, 53. PP. 538-9. 1974.
- Shackel, B. *Usability - context, framework, design, and evolution*. Human factors for informatics usability. Cambridge University Press. PP. 21-38. 1991.