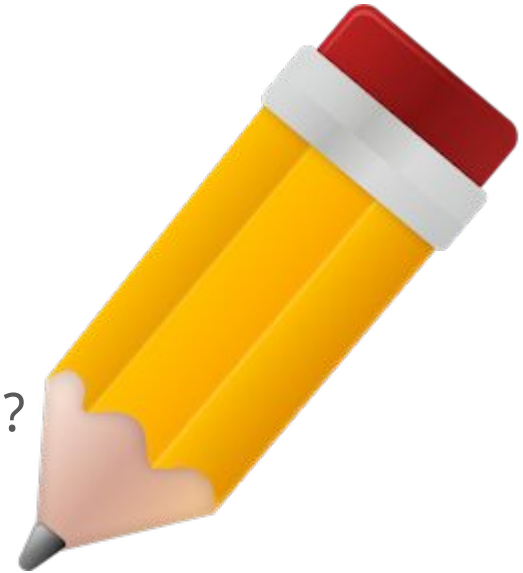
# Ch 5 SAQs (Pop Quiz)

1. Give an example of why Accessibility is for everyone.
2. What is your view regarding 'Combinatorial Impairment'?
3. Pick an interface bridge and describe it.
4. What is the relationship between effective and accessible design?
5. What are the five main principles of effective design?



# UX from 30,000ft Principles of Efficient Experience (part 2)

Chapter 6 - Lecture 10 (50 minutes)

@sharpic

http://sharpic.github.io/COMP33511/

# Collated Principles

| Principle | Appears in Source |
|---|---|
| Closure (Dialog Yields) | Shneiderman[1]. |
| Consistency / Standards | Xerox-Star[2]; Shneiderman; Norman[3]; Nielsen[4]; ISO 9241-110[5]; Dix, Finally, Abowd & Beale[6]; Raskin[7]. |
| Constraints (Exploit) | Norman. |
| Control & Freedom (Support) | Shneiderman; ISO 9241-110; Nielsen. |
| Error Handling (Simple) | Shneiderman; Norman; ISO 9241-110; Nielsen. |
| Familiarity & Metaphor | Xerox-Star; Norman; Dix, Finally, Abowd & Beale; Raskin. |
| Feedback (Informative) | Shneiderman. |
| Help & Documentation | Nielsen. |
| Interrupts (Resumption) | Raskin. |
| Describing (Self) | ISO 9241-110. |
| Heuristic Evaluation | Nielsen. |
| Learnability | Dix, Finally, Abowd & Beale, ISO 9241-110; Sharp, Rogers and Preece[8]. |

Collated Usability Principles, Guidelines, and Rules Harper 2014

# Collated Principles

| | |
|---|---|
| Mappings (Real-Virtual) | Norman; Nielsen; Dix, Finally, Abowd & Beale. |
| Memory Load (Reduce) | Shneiderman; Sharp, Rogers and Preece. |
| Navigation & Freedom (Support) | Raskin. |
| Reversal of Actions (Easy) | Shneiderman; Nielsen; Dix, Finally, Abowd & Beale. |
| Safety | Sharp, Rogers and Preece. |
| Shortcuts (Provide) | Shneiderman. |
| Simplicity | Xerox-Star; Norman; Brooks[9]. |
| Singularity of Focus (Attention) | Raskin. |
| Task Suitability & Conformance | Dix, Finally, Abowd & Beale; ISO 9241-110. |
| Tailor-ability / Flexibility | Xerox-Star; Nielsen; ISO 9241-110; Dix, Finally, Abowd & Beale. |
| Universal Commands | Xerox-Star; Dix, Finally, Abowd & Beale; Raskin. |
| Utility | Sharp, Rogers and Preece. |
| Visibility (Make Things) | Norman; Nielsen. |

Collated Usability Principles, Guidelines, and Rules Harper 2014

# Collated Principles

[a] B. Shneiderman and C. Plaisant. Designing the user interface: strategies for effective human-computer interaction. Addison-Wesley, Boston, 5th ed edition, 2010.
[b] D. C. Smith, C. Irby, R. Kimball, B. Verplank, and E. Harslem. Designing the star user interface. BYTE, 7(4):242Đ282, 1982.
[c] D. A. Norman. The design of everyday things. Basic Books, New York, 1st basic paperback edition, 1988.
[d] J. Nielsen. Usability engineering. Academic Press, Boston, 1993. Nielsen also lists Aesthetic and minimalist design – we'll look at these more in 'Collated Affective Concepts and Touch-points' (pg. 183).
[e] ISO/TR 9241-110:2006. Ergonomics of human-system interaction – part 110: Dialogue principles. TC/SC: TC 159/SC 4 ICS 13.180, International Organization for Standardization (ISO), Geneva, Switzerland, 2006.
[f] A. Dix, J. Finlay, G. Abowd, and R. Beale. Human-computer interaction. Prentice Hall Europe, London, 2nd ed edition, 1998. Dix, Finally, Abowd & Beale describe '*Learnability*' as: Predictability, Synthesizablity, Familiarity, Generalizabilty, and Consistency. '*Flexibility*' as: Dialog initiative, Multi-threading, Task migratabilty, Subsitutivity, and Customizability. '*Robustness*' as: Observability, Recoverability, Responsiveness, and Task conformance.
[g] J. Raskin. The humane interface: new directions for designing interactive systems. Addison Wesley, Reading, Mass., 2000.
[h] H. Sharp, Y. Rogers, and J. Preece. Interaction design: beyond human-computer interaction. Wiley, Chichester, 2nd ed edition, 2007. Preece, Rogers and Sharp also lists Satisfying, Enjoyable, Fun, Entertaining, Helpful, Motivating, Aesthetic, Supports creativity, Rewarding, and Emotionally fulfilling – again, we'll look at these more in 'Collated Affective Concepts and Touch-points' (pg. 183).
[i] F. P. Brooks. The mythical man-month: essays on software engineering. Addison-Wesley Pub. Co., Reading, Mass., anniversary ed edition, 1995.

Table 5: Usability Principles Collated by Source

Collated Usability Principles, Guidelines, and Rules Harper 2014

# Potted Principles

- '**Stability**' Are the interactions stable?
- '**Scalability**' Does the interface and its data scale?
- '**Simplicity**' Is interface and interaction simplicity encouraged?
- 'Situational **Awareness**' Is perception of the interface facilitate decision making?
- 'Self-**Description**' Does your system describe itself to the user?
- 'Progressive **Disclosure**' Are the interface options presented a step at a time?
- '**Familiarity**' Is your system 'intuitive'?
- '**Learnability**' Are the interactions easy to learn?
- '**Consistency**' Does your system exhibit internal and external (OS) constancy?
- '**Robustness**' Is the system robust to errors?

# SSSADD FLCR ☺

- '**Stability**' Are the interactions stable?
- '**Scalability**' Does the interface and its data scale?
- '**Simplicity**' Is interface and interaction simplicity encouraged?
- 'Situational **Awareness**' Is perception of the interface facilitate decision making?
- 'Self-**Description**' Does your system describe itself to the user?
- 'Progressive **Disclosure**' Are the interface options presented a step at a time?
- '**Familiarity**' Is your system 'intuitive'?
- '**Learnability**' Are the interactions easy to learn?
- '**Consistency**' Does your system exhibit internal and external (OS) constancy?
- '**Robustness**' Is the system robust to errors?

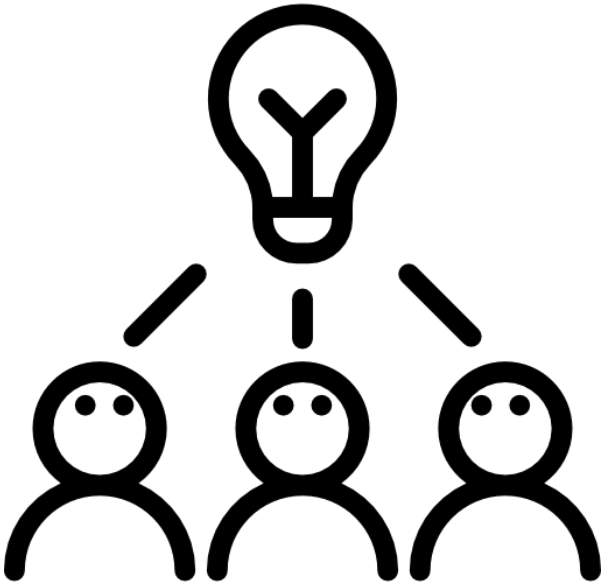
# Notes - In your own words!

# 3 minutes

# Interaction Stability

**Questions to think about as you design your prototype:**

**1 minute**



# Interaction Stability

**This is not an exhaustive list**

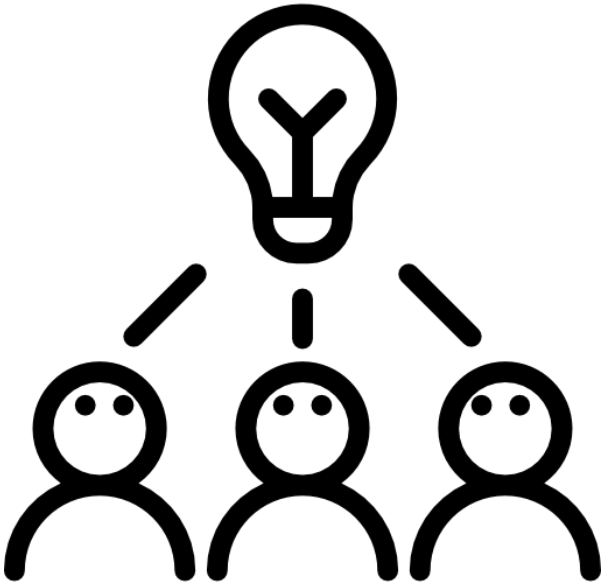
## Questions to think about as you design your prototype:

1. Are you able to resume interrupted actions?
2. Are you able easily reverse an action, incorrectly taken?
3. Are you able to understand your location in the interaction?
4. Does your system recover well from an unexpected event?
5. Does your interactions (including dialogs) exhibit stable non-cyclical behaviour and closure?

# Facilitate Scalability

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Scalability

This is not an exhaustive list 

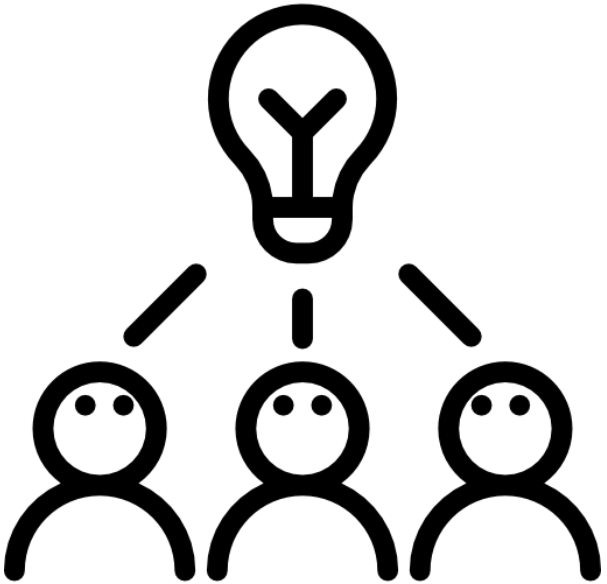
## Questions to think about as you design your prototype:

1. Does your interface scale to handle larger datasets than envisaged etc?
2. Does your system handle data and interaction within an acceptable time?
3. Do complex actions scale up in terms of data and user requirements?
4. Do your interfaces remain simple when information is being dynamically added?
5. Can new functionality be added to your system without negatively impacting on its current interactions and interfaces?

# Facilitate Simplicity

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Simplicity

**This is not an exhaustive list**

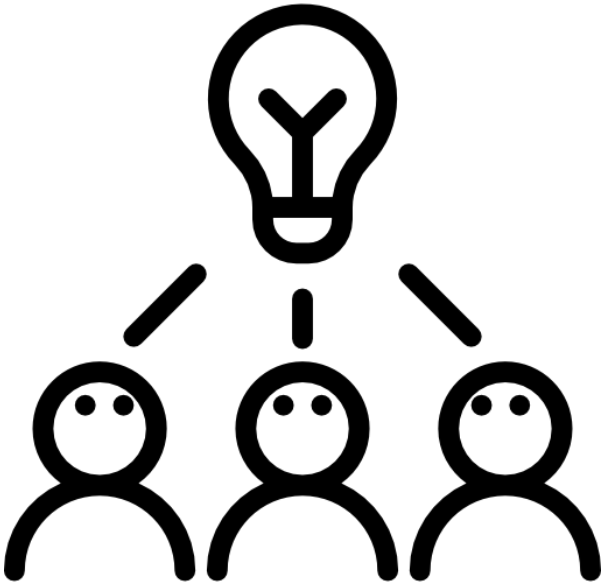
## Questions to think about as you design your prototype:

1. Is your system presented simply?
2. Are the interactive elements simple to understand and use?
3. Can you understand the system behaviour without recourse to manuals of help systems?
4. Does your system exploit natural interactive constraints?
5. Is complexity hidden from the novice user?

# Facilitate Situational Awareness

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Situational Awareness

**This is not an exhaustive list**

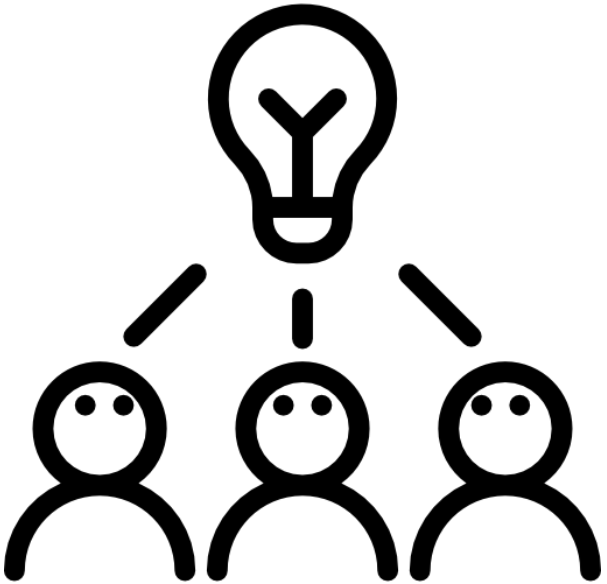
## Questions to think about as you design your prototype:

1. Does your system facilitate orientation both within the interface and within the interaction?
2. Is orientation and navigation, around and through the interface (and interaction), easy?
3. Is error handling simple? Is feedback informative?
4. Are all components, needed for the interaction, visible?
5. Do you maintain a single focus of interactive attention, without distractors?

# Facilitate Self-Description

**Questions to think about as you design your prototype:**

1 minute



# Facilitate Self-Description

**This is not an exhaustive list**

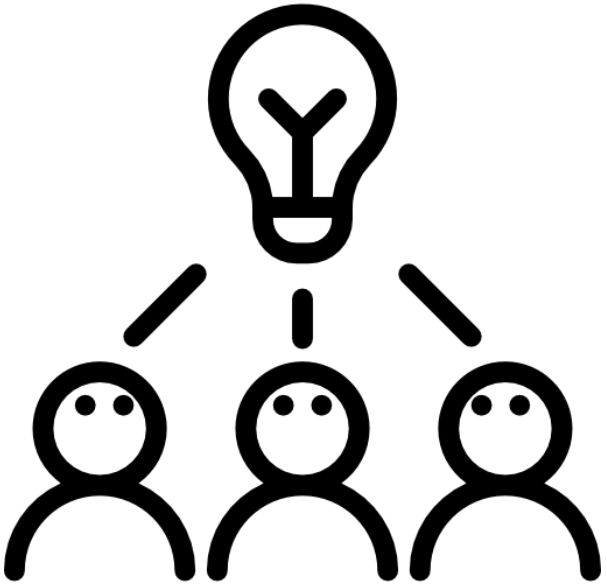
## Questions to think about as you design your prototype:

1. Is your system well documented?
2. Is help present and informative?
3. Is it possible to understand the program functionality without recourse to the manual?
4. Is it possible to understand the interface, widgets, and interactivity without recourse to the manual?
5. Is it possible to fully understand all dialogs, messages, and status'?

# Facilitate Progressive Disclosure

**Questions to think about as you design your prototype:**

1 minute



# Facilitate Progressive Disclosure

**This is not an exhaustive list**

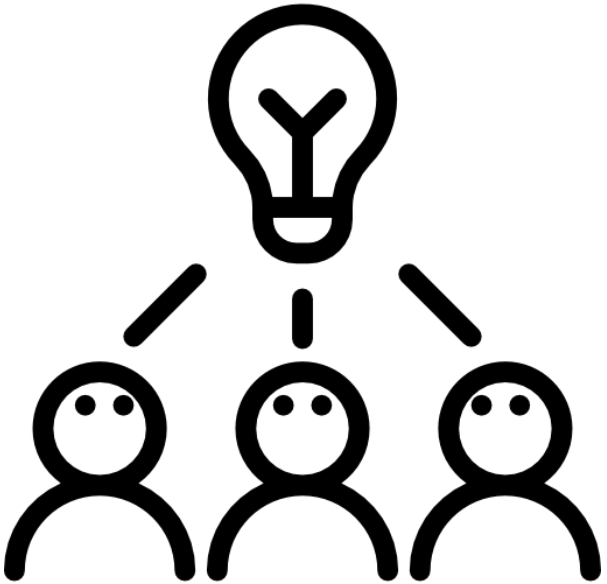
## Questions to think about as you design your prototype:

1. Does your interface look overly complex? If so, simplify.
2. Are there a lot of components displayed at one time? If so clean it.
3. Are there a multitude of possible actions available to the user? If so focus on building one action for one interface element.
4. Is there a tight logical hierarchy of actions?
5. Is the user led along the interactive path?

# Facilitate Familiarity

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Familiarity

**This is not an exhaustive list**

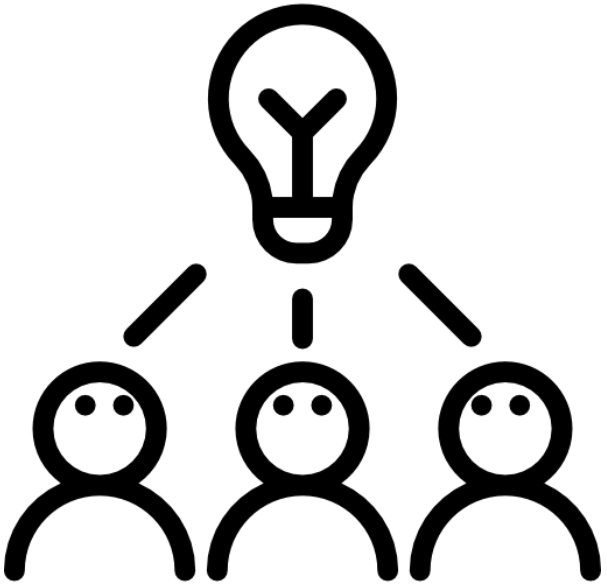
## Questions to think about as you design your prototype:

1. Does your system map real world concepts to the virtual world?
2. Does your system use terms the user is familiar with (including Jargon)?
3. Does the system work in familiar ways, with reference to itself and other comparable applications?
4. Do you assuage `intuition' for familiarity?
5. Does your system use easily understandable (and therefore familiar) messages?

# Facilitate Learnability

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Learnability

**This is not an exhaustive list**

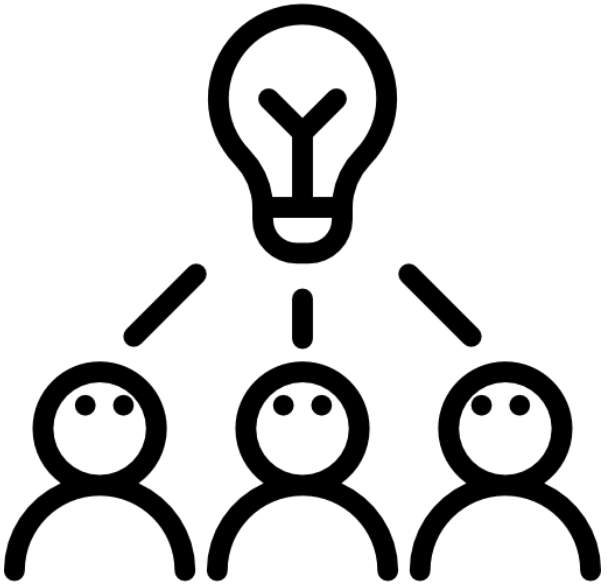
## Questions to think about as you design your prototype:

1. Is your system behaviour predictable?
2. Can users easily transit from novice to expert?
3. Can you understand the system behaviour without recourse to manuals of help systems?
4. How easy is it to learn any bespoke system functionality?
5. Does your system facilitate self learning, and functionality investigation?

# Facilitate Consistency

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Consistency

**This is not an exhaustive list**

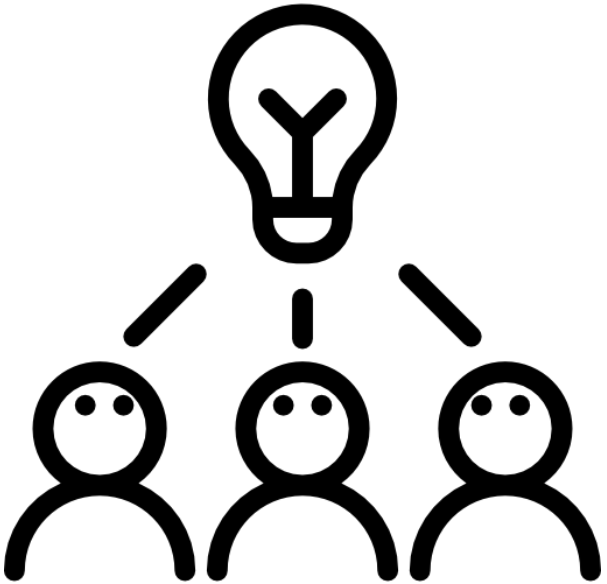
## Questions to think about as you design your prototype:

1. Am my developing a consistent interface?
2. Are the interactions consistent across the platform and development?
3. Is my command and event structure universal across the development and platform?
4. Am I following standards and best practice?
5. Am I following the platform design guide?

# Facilitate Robustness

**Questions to think about as you design your prototype:**

**1 minute**



# Facilitate Robustness

This is not an exhaustive list 

## Questions to think about as you design your prototype:

1. Does your system recover well from an unexpected event?
2. Are errors easily dealt with?
3. Are incorrect user actions easily recoverable?
4. Is the user-state saved between sessions in the event of a system failure?
5. How does your system handle abnormal input?

# To Do

1. Read Next Chapter.
2. Be Ready to Answer the Chapter SAQs (Pop Quiz).
3. Is there a Discussion Topic.

# See You Next Time!

Open House / Surgery - 2.60

Friday 09:00-11:00

@sharpic

# Break Time - Pause Recording

Back in 10 Minutes!

Come see me now if you have Questions Regarding this Lecture!