**CS235 Midterm Study Guide**

**By: Team Thundercats**

**Lecture #01 (01/22) – Introduction**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Interface Constraints** – Differ depending on the target application.  **Example:** Desktop has different constraints than web which has different constraints than mobile. | **UI** – User Interface  **UX** – User Experience  **Usability Testing** – Used to determine whether a UI design or UX design is good. | **Data Visualization** – Present analyzed data in a useful and insightful way to the user.  **Create actionable knowledge**. |  |

**Lecture #02 & #03 (01/27 & 01/29)**

**Design Principles and Functional Specifications**

**User Centric Design and Development**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Centric Design** – Involve the user throughout the design and development process. | **User Centric Design Principles**   * **Actively involve users** * **An appropriate allocation of function between user and application** * **Iteration of design solutions** * **Multidisciplinary design teams** | **User Centric Design Activities**   * **Analyze and specify context of use.** * **Specify user and organizational requirements.** * **Produce design prototypes** * **Evaluate designs with users against requirements.** | **Agile Development**  A single burst/iteration of development has three stages:   * **Design** * **Test** * **Code**   Successive iterations during development. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iterative Design** – Continuous cycle of design, prototype, and user testing and evaluation. | **Defining an Application** | | | | |
| **Purpose of the Application**  Clear definition and description of what the application does/should do. | **Users of the Application**  Define the target user(s) of an application. | **Domain Analysis**  Understand the area of expertise of special knowledge for which the application is developed. | **Task Analysis**  Understand the **goals**, **tasks**, and **actions of users** | **Workflow Analysis**  Understand how work can move from one user to another. |

**Mental Models and Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mental Model** – Enables a person to **negotiate unfamiliar situations** and **reason about a situation based on past experience and previously acquired knowledge**. | **“*Easy to Use and Intuitive*” Applications**  Mental model of the user aligns well with the mental model of the application designer. | **Sources of Requirements**   * **Client** * **End Users** * **Application Developers** * **Development Managers** * **Technology Providers** | **Requirements** received from different sources **may be contradictory or change.** |

**Functional and Nonfunctional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Requirements**  Entails what the application:   * **SHALL BE ABLE TO DO** * **ALLOW USERS TO DO** | **Functional Requirements**  Describes **INTERACTIONS** **between the application and the user** **INDEPENDENT OF IMPLEMENTATION**  **Example #1:** The phone **shall** be able to use GPS to determine its location.  **Example #2:** Users shall be able to choose either Option A or Option B. | **Nonfunctional Requirements**   * **Usability** * **Reliability** * **Performance** * **Supportability**   **Constraints the application must meet.** | **Example #1:** The application must respond to the user within 15 seconds.  **Example #2:** The application must run on Windows and Linux servers.  **Example #3:** The new GUI shall resemble the existing GUI. |

**Attributes of Functional and Nonfunctional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Completeness** – All system features are described by requirements | **Consistency** – No two requirements are contradictory | **Clarity** – Each requirement is unambiguous. | **Correctness** – There are no errors in the requirements and each feature is traced to a requirement. |

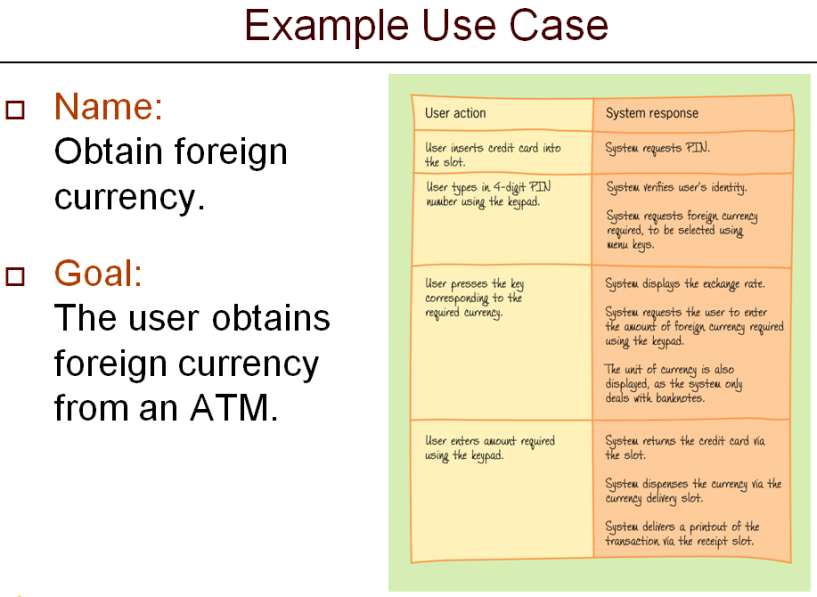
|  |  |  |  |
| --- | --- | --- | --- |
| **Realism** – Able to be implemented | **Verifiable** – Able to be tested. | **Traceability** – Each requirement traced to an application function. | **Strong Declarative Statements** – Use “**shall**” and “**must**” when describing a requirement. |

**Requirement Gathering**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stated Requirements** – What the user tells you s/he wants.  **Implied Requirements** – What you (the designer) think the user wants. | **Prototyping** – A user may not know what s/he wants until you show them a prototype. |  |  |

**Use Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** – **Complete sequence of steps that all a user to complete a task.**  Entails **a task your application must allow the user to do**. | **Components in a Use Case**   1. **Name – In the form *verb object*** (e.g. Obtain Foreign Currency) 2. **Goal** 3. **Sequence of Steps** 4. **Alternate sequences in case things go wrong** | **Format:** Use two-column setup. On left side is the user action and on the right side is the system response. | **Good Uses Case Criteria**   * Make sense to all stakeholders (client, developers, managers, etc.) * Provide real world context. |



**Lecture #04 (02/03) – Design Pattern Overview and**

**Organizational & Navigational Design Patterns**

|  |  |  |
| --- | --- | --- |
| A **design pattern** contains:   * **Description of the problem** * **A solution that can be applied to many programming situations** | **Design Patterns Attributes:**   * **Proven design experience** * **Describe best practices** * **Capture structural and behavioral features of an interface** | **Components of a Design Pattern**   1. **Name** 2. **What** – Description of the pattern 3. **When** – Description of the context to use the pattern. 4. **Why** – Description of the problem the pattern solves 5. **How** – A prescription for a solution. |

**Organizational Design Patterns**

|  |  |  |  |
| --- | --- | --- | --- |
| **Organizational Design Patterns**  High level overall layout of an applications user interface | **Information Architecture** – How to organize an information space | **Interaction Model** – Determines how users navigate and establishes consistency |  |

|  |  |  |
| --- | --- | --- |
| **Feature Search Browse**   * **What:** Contains: a **featured item**, **search box**, and a **list to browse**. * **When:** Engages the user and provides options including searching. * **Why:** Browsing and searching are common. **Hooks user with featured item**.   **Reduce burden on working and long term memory by enabling search.** – **Lecture #09**  **Assists with the creation of a visual hierarchy by making the featured item more prominent** – **Lecture #08**  **Common convention so even in a foreign language, this can be understood by a user** – **Lecture #08** | **News Stream**   * **What:** Displays **time sensitive items** in **reverse chronological order** especially when there are frequent updates. * **When:** Communication channels with **timely content** * **Why:** Allow users to keep up with news streams which they can check often.   **Make crucial element “pop” to draw the fovea to it. This is done by putting at the top of the list** – **Lecture #13**  **Creates a visual hierarchy by giving more important (i.e. newest) higher prominence by being on top** – **Lecture #08** | **Picture Manager**   * **What: Thumbnails of pictures** to browse with **one featured picture or video**. * **When:** Collection of pictures to organize * **Why:** Natural way to organize and browse picture collections   **Rely on recognition rather than recall.** – **Lecture #09**  **Can be used to create a visual hierarchy with the featured image as most prominent and the thumbnails as less prominent.** – **Lecture #08** |

|  |  |  |
| --- | --- | --- |
| **Dashboard**   * **What:** Single page that is information dense and includes **frequently updated widgets** * **When:** Incoming flow of information * **Why:** Users need to monitor the information in particular and **see what’s important at a glance**   **Make crucial element “pop” to draw the fovea to it** – **Lecture #13**  **If the widgets are graphical, allow users to visually recognize issues rather than perform more complex cognitive tasks** – **Lecture #09** | **Canvas and Palette**   * **What: Palette of drawing tools and a canvas for drawing.** * **When: Graphics editor**, **creating new objects**, and arranging objects in virtual space. * **Why:** Closely matches user’s mental model of how to create an artwork.   **Rely on recognition rather than recall.** – **Lecture #09** | **Wizard**   * **What: Several steps to accomplish a task** * **When: Task is complicated** and want to eliminate user thinking and tell them what to do * **Why: Handholding for beginning user** but can frustrate advanced users.   **Good if frequency of practice is low** – **Lecture #14**  **Reduces burden on memory of the right procedure** – **Lecture #09.**  **Lower user risk. Lecture #14**  **Reduce how much user must think. Lecture #08** |

|  |  |  |
| --- | --- | --- |
| **Settings Editor**   * **What: One menu to change settings, preferences or properties** with grouped content * **When:** Many choices to create a profile for an application. * **Why:** Centralizes choices in one place. Easy to find and use.   **Users learn faster if the steps are consistent and predictable. Creating a single page for modifying settings achieves that goal. Lecture #14** | **Alternate Views**   * **What:** Show **multiple views of the same data** in multiple formats. * **When:** A single view is insufficient so **let user choose the view**. * **Why:** Different views show different aspects of the data. Different users may prefer different views.   **Different users may have different goals so by allowing alternate views, users can adjust the UI to fit their particular goal. Lecture #09** | **Multiple Workspaces**   * **What: Multiple tabs or windows to view different pages at one time** (e.g. web browser) * **When:** Application must **manage multiple pages of content separately**. * **Why:** User wants to multiple **multitask without multiple copies of the application open**   **Reduced visual cognitive load – Lecture #05 Similar to “Two Panel Selector” Pattern** |

|  |  |  |
| --- | --- | --- |
| **Multilevel Help**   * **What:** A variety of ways to provide help to the user * **When: Complex application and users need help with specific features or tasks** * **Why: Provide help where and when needed.** Help can be lightweight or heavyweight. **Includes reminders, tips, or tutorials.**   **Lower user risk. Provide easy to access help when the user needs it. Lecture #14** |  |  |

**Navigational Design Patterns**

|  |  |  |
| --- | --- | --- |
| **Navigational Design Patterns**  Patterns for how a user navigates within an application. | **Signpost** – Feature that **helps a user determine where s/he is**. | Navigation errors frustrate users so the number of page jumps should be kept low. |

|  |  |  |
| --- | --- | --- |
| **Clear Entry Points**   * **What: Only a few main entry points into an interface. Entry points are task oriented (clear calls to action.** * **When: Good first time and inexperienced users.** * **Why:** Avoids clutter when an application opens and **does not overwhelm a novice user**.     **Example:** Login Page  **Lower user risk since it gives limited options to a novice user. Lecture #14** | **Hub and Spoke**   * **What:** A page contains links to many other pages. * **When:** Table of contents or portal to show the user where to go from here. * **Why:** The user decides where to go but returns **to main page to go elsewhere**.     **Lower user risk since user always knows hot to return to the main page Lecture #14**  **Operation is simple, consistent, and predictable so easy for a user to learn and use. Lecture #14** | **Pyramid**   * **What: Sequence of pages with back and next links.** * **When:** Sequential text **organized by chapters and return to table of contents.** * **Why:** Matches user’s **mental model of a tutorial**     **Biased by Experience: Familiar Patterns**: **Lecture #09** |

|  |  |  |
| --- | --- | --- |
| **Modal Dialog**   * **What: Show only one page with no navigation options**. User must complete page’s task. * **When: Application cannot proceed without user input.** * **Why:** User cannot ignore page and must provide input. Can be disruptive or annoying to user.     **Lower user risk since if issue is critical, the user must handle it immediately and cannot ignore it. Lecture #14**  **Modal dialog will help draw the fovea to it since its appearance in peripheral vision is like movement Lecture #14** | **Escape Hatch**   * **What:** User can abort a task. * **When: Application takes the user down a path, but user may want to back out.** * **Why:** User changes his mind or recognizes that s/he is going down the wrong path.     **Lower user risk since they know they can backtrack. Lecture #14** | **Sequence Map**   * **What: Show a map of pages in a sequence.** **Includes you are here.** * **When:** Application takes user down a **linear path.** * **Why:** Tells **how** **far the user has gone and how much further s/he has to go**.     **Operation is simple, consistent, and predictable so easy for a user to learn and use. Lecture #14**  **Reduces burden on working memory as users can see where they come from and even where they are going. Lecture #14** |

|  |  |  |
| --- | --- | --- |
| **Annotated Scrollbar**   * **What:** Scrollbar says “you are here” * **When:** Application is document centric or allows pan and zoom. * **Why: While manipulating the scrollbar, user wants to know what content is being displayed**.   Screen Shot 2014-09-07 at 8.02.26 PM.png  **Lower user risk of error since they can see exactly where they are. Lecture #14**  **Reduces burden on memory to remember the context to page number linking** – **Lecture #09**  **Reduces how much users need to think to determine where the current information is in the document Lecture #08** | **Breadcrumbs**   * **What: On each page in a deep hierarchy, show a list of parent pages in order.** * **When:** User needs to keep track of where they are and wants to go back up to any point in the hierarchy. * **Why:** Allows user to move forward or back within a deep hierarchy and user knows where they are in the hierarchy.       **Reduces burden on working memory**– **Lecture #09**  **Lower user risk since knows current location and how to back track. Lecture #14** | **Pan and Zoom** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bookmark** |  |  |  |  |

**Lecture #05 (02/05) – Gestalt Principles &   
Layout and List Design Patterns**

|  |  |  |  |
| --- | --- | --- | --- |
| **Page Layout** – Manipulates page attention to convey:   * **Meaning** * **Sequence** * **Points of Interaction** | **Tools to Show Importance** | | |
| * **Text Size** * **Text Density (e.g. bold)** * **Background color** * **Visual Rhythm (e.g. how they flow)** | * **Position and size** * **Emphasize small items** * **Group related items** * **List of items** | * **Captions and comments** * **Peer items (same size and shape)** * **One distinguished item (e.g. different border)** * **Containment (e.g. frames around objects)** |

**Four Gestalt Principles**

**Gestalt** (German) – An organized whole that is perceived as more than the sum of its parts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Proximity**  Viewers will **associate items that are placed close together**. | **Similarity**  Viewers will **associate two things if they have same shape, size, orientation, color, etc**. | **Continuity**  Viewers’ eyes want to see **continuous lines and curves** formed by the alignment of items. | **Closure**  Viewers’ eyes want to **see implicit simple closed forms** such as rectangles. |

**Layout Design Patterns**

|  |  |  |
| --- | --- | --- |
| **Visual Framework**   * **What:** A common framework for all application pages but which allows flexibility to handle varying page content. * **When:** An application with multiple pages and **want pages to appear that they belong together.** * **Why: Provide consistency** and help page content stand out (**can provide persistent navigation**).   **A website template across all pages**  **Consistency in format allows users to learn how to use a site faster – Lecture #14 & #09** | **Center Stage**   * **What: Most important part of UI has the largest subsection of the page**. Secondary windows in smaller subpanels clustered around. * **When:** Page’s primary job is to show a single unit of information or permit a single task. * **Why:** Guides the user to what’s most important.   **Make crucial element “pop” to draw the fovea to it. This is done by putting at the top of the list** – **Lecture #13**  **Assists with the creation of a visual hierarchy by making the featured item more prominent** – **Lecture #08**  **Gestalt principle foreground/background. Our brains focus on what is in the foreground.** – **Lecture #08** | **Grid of Equals**   * **What: Arrange equally important items in a grid, each with equal weight.** * **When:** Many items have similar style and important. Allows the user to preview. * **Why:** Neat and orderly. Give each item equal importance. Tells users the items are similar.   **Gestalt similarity principle – Lecture #05**  **No item in the visual hierarchy is given more prominence so all similar – Lecture #08** |

|  |  |  |
| --- | --- | --- |
| **Tiled Sections**   * **What: Visually separate sections of content.**   **Each section has a strong title.**   * **When:** Make the page **easy to scan with grouped content**. * **Why:** Content is structured and easily digestible chunks. Obvious information architecture.   **Gestalt proximity principle – Related items are grouped together – Lecture #05**  **Strong title creates a visual hierarchy – Lecture #05 & 08**  **Nesting shows items associated with the title – Lecture #08** | **Module Tabs**   * **What: Put content modules into separate tabs.** Only one tab is visible at a time. * **When: Multiple content types. User only needs to see one type at a time.** * **Why:** Tabs have a familiar physical model. Less cluttered user interface.   **Reduced visual cognitive load – Lecture #05 Similar to “Two Panel Selector” Pattern** | **Collapsible Panel**   * **What:** User can **open and close secondary content panels.** * **When:** Not enough room to show all content **and/or** different users have different needs. * **Why:** Allow the user to choose what to see.   **Reduced visual cognitive load – Lecture #05 Similar to “Two Panel Selector” Pattern** |

|  |  |  |
| --- | --- | --- |
| **Movable Panel**   * **What: User can rearrange content panels.** * **When: Different arrangements for different purposes or if users have different needs.** * **Why:** Centralizes choices in one place. Easy to find and use.   **Assists with the creation of a visual hierarchy by making the featured item more prominent** – **Lecture #08** | **Left/Right Alignment**   * **What:** Two column interface. **Right aligned labels on the** **left** **and left aligned labels on the** **right**. * **When:** An input form or any other set of labeled items. * **Why:** Neat with strong perceptual grouping   **Gestalt proximity principle – Lecture #05**  **Gestalt continuity principle. Sees this as one element since in a line. – Lecture #05** | **Responsive Enabling**   * **What: Progressively enable** **UI components as the user selects options.** * **When: Leads a user through a complex task step by step.** User can easily change his mind. * **Why:** User can see all options but only required objects are enabled   **Improve scanability and reduce confusing options – Lecture #08**  **Lower user risks since only valid options are selectable. Lecture #14**  **Reduces the amount the user must think as clear since only valid choices displayed** – **Lecture #08** |

|  |  |
| --- | --- |
| **Self-Adjusting Layout**   * **What: Page contents automatically resize to maintain the same arrangement for a different window size or aspect ratio.** * **When:** User can resize window as desired. * **Why:** Users choose window sizes and aspect ratios unpredictable.   **Many demo pages may have this.**  **Improves page consistency for the user making its behavior more predictable and hence easier to learn – Lecture #14** |  |

**Lists**

|  |  |  |  |
| --- | --- | --- | --- |
| **Benefits of Lists** | | **Ways to Organize Lists** | |
| * **Provides an overview of all items** * **Browsability item by item** * **Search for a specific item** | * **Sort and filter items** * **Rearrange, add, delete, and recategorize items.** | * **Length** * **Ordering of the items** * **Grouping of items** | * **Item types** * **User interaction with the list** * **Dynamic behavior of the list** |

**List Design Patterns**

|  |  |  |
| --- | --- | --- |
| **Two Panel Selector**   * **What:** Two panels side by side. **First panel shows a list and second panel shows the selected item’s content.** (Two tiered list) * **When:** Show the entire list of items and each item has **subcontent**. * **Why:** Reduced physical effort, **reduced visual cognitive load, less user memory burden.**   **Reduced visual cognitive load – Lecture #05** | **One Window Drilldown**   * **What: A list of selectable items in a single window.** When user selects an item, replace **the list with the selected item.** * **When:** Each item has content; there is limited available space. **Large list with lots of content.** * **Why:** Only option when space is tight.   **Provides feedback to the user that the item was selected. –** **Lecture #13**  **Reduced visual cognitive load – Lecture #05 Similar to two panel selector** | **List Inlay**   * **What:** Display a list of items in a column. **When user selects an item, show the item details in place with the list.** User can independently open and close item details. * **When: Each item has only a small amount of content,** but total content is too much to view. * **Why:** User can choose what to view.   **Reduced visual cognitive load – Lecture #05 Similar to two panel selector** |

|  |  |  |
| --- | --- | --- |
| **Thumbnail Grid**   * **What:** Arrange a **list of items as a thumbnail of images. Each thumbnail is labeled.** Show the larger sized content of a selected item. * **When:** Items have **content recognizable and representable in a smaller format**. * **Why:** Attractive way to display a large list of items.   **Fitt’s Law Thumbnails are large, easy targets to click on - Lecture #14**  **Rely on recognition rather than recall. Lecture #09** | **Carousel**   * **What:** List items arranged in a **scrollable horizontal strip.** * **When: Insufficient space for thumbnail grid.** * **Why:** Encourage users to scroll and browse.   **Rely on recognition rather than recall. Lecture #09**  **Reduced visual cognitive load – Lecture #05 Similar to two panel selector** | **Striped Rows**   * **What:** Table **rows are striped with alternating colors**. * **When:** Not enough room to show all content **and/or** different users have different needs. * **Why:** Allow the user to choose what to see.   **Gestalt similarity principle. Similar things are colored the same. Lecture #05**  **Steering Law for the eyes Lecture #14** |

|  |  |  |
| --- | --- | --- |
| **Pagination**   * **What: Break a long list into pages (e.g. multiple Google search results). One page displayed at a time with buttons for “First”, “Last”, “Previous”, “Next”** * **When:** A very long list of items that would **take too much time to render and load all at once.** * **Why:** Break a list into manageable chucks. **Paging metaphor/model is known and familiar.**   **Biased by Experience: Familiar Patterns**: **Lecture #09** | **Jump to Item**   * **What:** A scrollable list of text items. **When user starts to type the content of an item, jump to that item and select it.** * **When:** A long list of text items. Only a portion is visible and scrolling is tedious. * **Why:** User already knows what to look for. Quick and direct access by the user. Take advantage of the computer’s fast searching.   **Example:** Typing “Ari” would take me to “Arial” font.  **Makes visual element pop to make visual search easier so the Fovea to find it. Lecture #13 Slide #31** | **Alphabet Scroller**   * **What:** Alphabetized list. **Display the letters of the alphabet and user clicks on a particular letter of the alphabet and list jumps to that letter in the list**. * **When: Leads a user through a complex task step by step.** User can easily change his mind. * **Why:** User can see all options but only required objects are enabled.   **Makes visual element pop to make visual search easier so the Fovea to find it. Lecture #13 Slide #31** |

|  |  |  |
| --- | --- | --- |
| **Cascading Lists**   * **What:** Display a **hierarchy of selectable lists** at each level. **Select an item to display that item’s children**. * **When:** List of items arranged in a **deep hierarchy**. * **Why:** Display more of a hierarchy by spreading it across several lists. Users can easily keep track of which level s/he is on.   **Not a single column like tree table. Not selectable by individual items.**  **Reduces burden on working memory since parent items in the list are displayed** – **Lecture #09**  **Reduced visual cognitive load – Lecture #05 Similar to two panel selector** | **Tree Table**   * **What:** Display items in a **single column** using an **outline hierarchy structure.** User can **expand and collapse individual items.** * **When:** A list of items arranged in a possibly deep hierarchy. * **Why:** User can **choose whether or not to view an item’s descendents** and to what level.   **Reduces burden on working memory since parent items in the list are displayed** – **Lecture #09**  **Reduced visual cognitive load – Lecture #05 Similar to two panel selector** | **New Item Row**   * **What: After the last item of a list, create a new item in place.** * **When:** Creates **new items at the end of a list**. * **Why:** Adds new item in final position of the list.   **Reduces the amount the user must think as clear how to add a row.** – **Lecture #08**  **Lower user risk since this prevents users from trying to insert an item in the middle of the list. Lecture #14** |

**Lecture #06 (02/10) – Design Requirements (Guest Lecture)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements Phases**   1. **Pre-project** – Request for proposal. Marketing requirements. 2. **Project Initiation** – Requirements gathering and refining 3. **In-Progress Project Reviews** – Requirements may change 4. **Web, Desktop, and Mobile Applications** – **May have different release cycles** so requirements may change again. | **Causes for Wrong Requirements** | | |
| * **Wrong People** – People (e.g. **managers**) with **limited understanding of the problem** or **no UI/UX experience**. * **Mix of People** – May have different goals or incorrect priorities. | * **Wrong Priorities** – Focused on **PAIN POINTS** rather than on business priorities. **Focused on legacy rather than the future**. * **Copying Other Applications** – Often not appropriate. Leads to “**INTERFACE PIZZA**” without cohesion. **Backward looking**. | * **Lack of Business Knowledge** – **NOT KNOWING WHAT IS POSSIBLE.**   **Examples:** Geolocation, image recognition, audio input, language translation, backend verificaiton services, etc. |

**Getting Good Requirements**

|  |  |  |
| --- | --- | --- |
| 1. **Understand the basics.**  * Use **questionnaires**. * Figure out **likes and dislikes** * **Colors and Fonts** – Check preferences and for any company standards. * **Mood** – **Examples:** Professional, efficient, fun * **Language(s)** * **Target User(s)** – **Examples:** Age, gender, education, training. | 1. **Review documentation and training material**. 2. **Engage and interact with actual users**  * **Helps to understand workflow** | 1. **Observe the system end-to-end**  * **Identify weak points**  1. **Ask lots of questions**  * **In particular ask “Why?”** |

|  |  |  |
| --- | --- | --- |
| 1. **Write (and if necessary rewrite) requirements**  * Create **use cases** * **Validate** with both users ***and*** decision makers.  1. **Build a prototype**  * May require **multiple iterations**. | 1. **Testing**  * **Test with ACTUAL users**  1. **Plan for documentation, help, messages, and training** | 1. **Create UI Transition Plan**  * **Leverage existing learning** * **Remember:** Some users will always resist change |
| **Summary:** **Redevelop and Refine the Requirements** | |

**Dealing with Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pre-Project Requirements**   * May need to commit based off bad requirements. * **Minimize Risk** – **Make conditional commitments.** Allow time and budget for changes. * Filter out “**problem clients**” | **Getting Buy-In**   * **Educate your clients:**   + Present UI design principles   + Identify the information you need   + Explain where the requirements fall short.   **Inform client of need for**  **interface review and update** | **Requirements Management**   * Remember **requirements** **evolve** * **Control, limit, and manage requirements changes** * Insist on a single “**authoritative contact**” * **Put everything in writing** * **Get** **everything in writing** * Insist on **timetable for requirements review**. |  |

**Dealing with Graphic Designers**

|  |  |  |
| --- | --- | --- |
| **Importance of Graphic Design**   * **Graphical design is critical to success** * **Graphic design requires specialized skills including:**  Color theory, layout, typography, iconography, graphic development tool knowledge, etc. * **Design fashions and styles change** | **Trusting and Working with a Graphic Designer** | |
| * **Set personal, individual preferences aside** * **Choose a designer based on review of past work** * **Make sure the designer understands the requirements** | * **Tell the designer the required deliverables:**   + Flattened Photoshop files   + Sized Icons   + Font and color specifications   + CSS files * **Incorporate graphic designs in prototypes** * **Separate graphic designs from code** |

**Lecture #07 (02/12) – Action and User Input Design Patterns**

**Action Controls**

|  |  |  |  |
| --- | --- | --- | --- |
| **Action Controls** | | **Action Design Patterns**   * **Ways to present an action** * **Indicate progress of actions** * **Give users a sense of closure** * **Allow users to preview actions** * **Allow users to cancel actions** * **Allow users to undo actions** |  |
| * **Buttons** * **Menubars** * **Pop-up menus** * **Drop-down menus** * **Toolbars** * **Links** | * **Action Panel (Action design pattern)** * **Hover Tools (Action design pattern)** * **Double Clicking** * **Keyboard actions** * **Drag-and-drop** * **Typed Commands** |

**Action Design Patterns**

|  |  |  |
| --- | --- | --- |
| **Button Groups**   * **What:** Group related actions. **Multiple groups of buttons for multiple types of actions**. * **When:** Many possible actions that can be made into related sets. * **Why: Gestalt principle (Proximity).** Easy to pick out clusters (i.e. user is **scanning**) in a complex layout.   **Gestalt proximity principle. Associated/related items are grouped. – Lecture #05** | **Hover Tools**   * **What: A list of selectable items in a single window.** When user selects an item, replace **the list with the selected item.** * **When:** Many possible actions so keep user interface as uncluttered as possible. * **Why:** Show or **enable only relevant controls**.   **Lower user risk and chances to make a mistake since only relevant options are enabled Lecture #14** | **Action Panel**   * **What:** A **panel of related controls** that is richly organized and always visible. * **When:** Controls need to be always visible and there is sufficient space to display the controls * **Why:** Users will always know what actions are available.   **Gestalt proximity principle. Associated/related items are grouped. – Lecture #05** |

|  |  |  |
| --- | --- | --- |
| **Prominent Done Button**   * **What:** Place the button that completes an action **prominently at the end of the visual flow**. * **When:** Whenever you need a “Done”, “Ok”, “Submit”, or “Continue” button. * **Why:** Give the user a sense of close and that the action was done.   **Lowers the risk of using an application since user less likely to select the wrong option – Lecture #14**  **Makes visual element pop to make visual search easier so the Fovea can find it. Lecture #13 Slide #31** | **Smart Menu Items**   * **What:** Dynamically change or disable menu items. * **When:** Different actions for different contexts. * **Why:** Only show actions that are relevant for a given context.   **Lower user risk and chances to make a mistake. Lecture #14**  **Reduces the amount the user must think since only valid choices displayed** – **Lecture #08** | **Preview**   * **What:** Show a **preview** or **summary** **of the results of a pending action**. * **When:** User is about to perform an action that will produce results. * **Why:** Allow the user to choose what to see.   **Lower the risk of an application** – **Lecture #14** |

|  |  |  |
| --- | --- | --- |
| **Progress Indicator**   * **What:** Show how much progress a time consuming action has made. * **When:** A time consuming action that runs in the background. * **Why: Assure the user progress is being made** Indicate how much work remains.   **Perception of time. Perceived responsiveness** – **Lecture** **#14** | **Cancelability**   * **What:** Provide a way to cancel a time consuming activity. * **When:** A time-consuming action runs in the background. * **Why:** Users change their mind and may want to terminate a time consuming action.   **Lower the risk of an application** – **Lecture** **#14**  **Prioritize user actions to improve the perception of time** – **Lecture** **#14** | **Multilevel Undo**   * **What:** Allow the user to **reverse a sequence of actions**. * **When: Highly interactive application with many user actions**. * **Why:** Provides an interface that is safe to explore. Users are more confident changes are not permanent. No need to “checkpoint” data.   **Lower the risk of an application** – **Lecture** **#14** |

**Input Controls**

|  |  |  |  |
| --- | --- | --- | --- |
| **Text Input Controls**   * **Single Line Text Input Controls** * **Multiline Text Input Controls** * **Structured Text (Structured Format Pattern)** * **Text Editor** | **Choice Controls** | | **Tools for Creating Lists**   * **List with Add Button** * **List with New Row Item (New Item Row List Design Pattern)** * **List Builder (List Design Pattern)** |
| * **Checkbox** * **Radio Button** * **Toggle Button** * **Spinner (increment through a list of numbers)** * **Slider** * **Calendar Chooser** * **Dropdown Chooser** | * **Dropdown List** * **Checkbox List (Check individual items in the list)** * **Multiple Selection List (Select multiple items in the list)** * **Combo box (Dropdown and text editable box)** * **Multiple Selection Tree** |
| **Controls for Sort Lists**   * **List with UP and DOWN buttons.** * **List with internal drag-and-drop** |

**Input Design Patterns**

|  |  |  |
| --- | --- | --- |
| **Forgiving Format**   * **What:** Input text field that allows **a variety of input formats and syntax** (e.g. zip code, city name, etc.) * **When:** User needs to enter information that can be typed in a variety of ways. * **Why:** Users are unpredictable and this is friendlier for the user.   **Reduces burden on long term memory to remember format. Lecture #09**  **Lower the risk of using an application** – **Lecture** **#14**  **Reduces the amount the user must think as text box says what it does** – **Lecture #08** | **Structured Format**   * **What:** A set of input **text fields that reflect the structure of the request data** (e.g. four 4 digit fields for credit card) * **When:** Input data must have a fixed structure. * **Why:** Provides a clue to user and reduces data entry errors.   **Bias by Experience – Context. Places options in context. Lecture #09**  **Lower the risk of using an application by guiding them to the right approach** – **Lecture** **#14** | **Fill In the Blanks**   * **What:** A sentence or **phrase with one or more blanks the user must fill in.**   **TRANSLATE <BLANKTEXT> to <BLANKLANGUAGE>**   * **When:** User input is in several related parts. * **Why:** Self explanatory interface with hints to the user.   **Bias by Experience – Context. Places options in context. Lecture #09**  **Lower the risk of using an application by guiding them to the right approach** – **Lecture** **#14** |

|  |  |  |
| --- | --- | --- |
| **Tips and Hints**   * **What:** Explanatory text that **provides tips or hints about the desired input. (Can be next to, below, or hover over)** * **When:** A form with a variety of different types of text input fields. * **Why:** Tell user what input is desired so user does not need to consult the documentation.   **(Same usage as “Prompting Text Field”)**  **Reduce burden on long term memory to remember the correct field information. Lecture #09**  **Lower the risk of using an application** – **Lecture** **#14**  **Reduces the amount the user must think as text box says what it does**– **Lecture #08** | **Prompting Text Field**   * **What:** Text **input fields are pre-filled with user prompts**. * **When:** A form with a variety of different types of text input fields. * **Why:** Tell user what input is desired so user does not need to consult the documentation.   **(Same usage as “Tips and Hints”)**  **Reduce burden on long term memory to remember the correct field information. Lecture #09**  **Lower the risk of using an application** – **Lecture** **#14**  **Reduces the amount the user must think as text box says what it does**– **Lecture #08** | **Autocompletion**   * **What: Anticipate possible input data based** on what the user has already typed. * **When:** A list/set of possible ways to complete the input. * **Why:** User friendly time saver. User does not need to remember long input User can reuse previous input.   **Reduce burden on long term memory. Lecture #09**  **Improves perceived responsiveness. Work ahead of the user** – **Lecture** **#14**  **Reduces the amount the user must think since it will give suggestions on possible queries** – **Lecture #08** |

|  |  |  |
| --- | --- | --- |
| **Dropdown Chooser**   * **What: A dropdown menu with a variety of items.** * **When: A user must choose from a set of items.** * **Why:** Users are familiar with dropdown menus. Compact way to present menu items. Display menu choices only when needed.   **Affordance – Control suggests how it should be used** - **Lecture #13**  **Following Conventions – Using conventions reduces how much users need to think. Encourages familiarity** - **Lecture #08** | **Good Defaults**   * **What: Pre-fill input fields with likely user inputs.** * **When:** Reasonable guesses of what the end user will want. * **Why:** User friendly time-saver. Less knowledge required by the user   **Reduces burden on working memory. Lecture #09**  **Reduces the amount the user must think as text box says what it does** – **Lecture #08** | **List Builder**   * **What: Source (input) and destination (selected) lists.** “Add” and “Remove” buttons or drag and drop from one list to another. * **When: User must create a list using items from another list**. * **Why:** User sees the available choices for the destination list and clear what the destination list will be.   **Reduces burden on short/long term memory to remember choices. Lecture #09** |

|  |  |  |
| --- | --- | --- |
| **Same Page Error Messages**   * **What: Display error messages on the same page next to the erroneous input** * **When:** User might enter bad input. * **Why:** Immediate feedback near erroneous input   **Gestalt proximity principle. Message related to field by proximity. Lecture #05**  **Center of vision is relatively small so avoid placing important items where they would be seen by peripheral vision. Lecture #13** |  |  |

**Lecture #08 (02/17) – Usability and Websites**

|  |  |  |  |
| --- | --- | --- | --- |
| **Definitions of Usability** | | **Thinking**  **Users do NOT want**  **to have to think.**   * **Make things obvious for the user.** * **Example:** Make what is clickable (e.g. a button) obvious | **Scanning**   * Users do not read web pages. **Users SCAN web pages. Eyes quickly jump from item to item searching in a non-structured or undisciplined fashion.** * **Pages should be formatted to facilitate scanning.** * Webpage user is looking to get a job done. |
| * **Useful** – Does the page do something users need? * **Learnable** – Can users figure out how to use it? * **Memorable** – Do users need to relearn it each time they use it? * **Effective** – Does it get the job done? | * **Efficient** – Does it do the job with a reasonable amount of time and effort? * **Desirable** – Do people want it? * **Delightful** – Is using it enjoyable or even fun? |

**Visual hierarchies are improve scanability by creating patterns.**

|  |  |  |  |
| --- | --- | --- | --- |
| **“*Satisfice*”**  “*Satisfice*” = “Satisfy” + “Suffice”   * Users do not make optimal choices when searching. * Users “satisfice”. **They make the first reasonable choice.** | **Follow Conventions**  Conventions allows a user to more quickly figure out the intent  **Example:** Red octagon for a stop sign regardless of language. | **Visual Hierarchy** | |
| * **The more important something is, the more prominent it should be.** * **Things that are related logically should be related visually.** * **Use nesting to show what is part of what.** | * Create visual separations that **break pages into clearly defined areas**. * **Heading text should be well differentiated (e.g. size, bold, etc.)** * **Headlines should not “float”.** Place them closer to item they apply to not equally between two paragraphs. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Avoid Confusing Options**   * User may find a process difficult and decide it is not worth it to continue. * **Display only relevant options.** | **Home Page**  Should convey two things quickly:   * **What is the big picture?** * **What is this website about?**   **First few seconds on a page are more critical.**  Since not all users start on the home page, **make the home page easy to return to from anywhere on the site.** | **Web Navigation** | |
| **Two Types of Users**   * **Search Dominant User** – Users look for a search box as soon as they enter a site. * **Link Dominant User** – Users want to browse first. Search only when they have run out of links. | **Persistent (Global) Navigation** – A set of navigation elements that appear on every page of a site. |

**Web Navigation Headers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Page Naming**  **Every page should have a prominent name.**  Page names should be **consistent** (**“Consistent Terminology” – One and only one name Lecture #14**). | ***“You are Here”***  Always make the **current location stand out**.  **Breadcrumbs** – **Show both where you are AND how you got there** (see navigation design pattern) | **Tagline** | |
| * Good taglines are **clear and informative** and **explain what the website or organization does**. * Six to eight words generally (**long enough but not too long**) | * **Convey differentiation and a clear benefit.** * Not generic. * Personable, lively, clever (but not too clever to obscure the benefit) |

**Lecture #09 (02/17) – Biasing, Perception, and Memory**

**Usability Testing Tips**

|  |  |  |  |
| --- | --- | --- | --- |
| Maintain an ongoing conversation with the tester. | Ask the tester to think out loud to get a sense of their thought process. | Do not hand hold the tester. Let the user explore naturally to get a true sense of their experience. |  |

**Cognitive Science**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cognitive Science**  Interdisciplinary study of mind, intelligence, and behavior. | **Factors Influencing the Mind’s Processing of Information** | |  |
| * **Perception** * **Language** * **Memory** | * **Reasoning** * **Emotion** |

**Bias by Experience**

|  |  |  |
| --- | --- | --- |
| **Bias by Experience: Priming**  Recent experience affects how the brain reacts to new data.  **Example:** Life hologram image with “life insurance” | **Bias by Experience: Familiar Patterns**  Humans **EXPECT** and **DESIRE** consistency.  **Example:** Brain assumes things are the same without checking (“Back” and “Next” button swapping. | **Bias by Experience: Context**  Brain process inputs differently based off what is around the object.  **Example:** French Napkins, Polish Silverware  vs.  Fold Napkins, Polish Silverware |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Perceptual Input Types:**   * **Visual** * **Auditory** * **Olfactory (Smell)** * **Gustatory (Taste)** * **Tactile Sensory System (Touch)** | **Perception Bias by Goals**  **Goals influence where we look and what we are sensitized to see.** | **Perception and User Interface Design** | | |
| **Avoid Ambiguity** – Ensure all users interpret the UI in the same way. | **Be Consistent** – Use consistent colors, fonts, shapes. Place controls in the same place. | **Understand Goals:** Different users may have different goals. Ensure UI directs user to the right goal. |

**Types of Memory**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Long term memory** – Changes in the neurons that are involved in a **neuron activity pattern**. | **Long Term Memory** | | **Short Term Memory**  Combination of perception, attention, and retrieval from long term memory. | **Working Memory**  **Main component of short term memory.** Tiny subset of all available information from perceptual system and long term memory that **we are aware of right now.**  **NOT A MEMORY STORE** |
| **Recognition** – New perceptions similar to previous ones reactive the same pattern of neurons. | **Recall** – **Other brain activity** cause a pattern of neurons to active. |

**Attention and Working Memory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attention is highly focused and selective.** | **Capacity of Attention**  4 items plus or minus 1 | **Volatility of Working Memory**  Easy to forget goals and details. | **Examples:**   * **Remind users of search terms** * **Leave breadcrumbs for the user** * **Do not force users to memorize instructions.** |

**Long Term Memory**

|  |  |  |  |
| --- | --- | --- | --- |
| **Long-term memory is a memory store.**  **Metaphor:** A dark warehouse | **Deficiencies of Long Term Memory**   * **Error prone** * **Weighted by Emotions** * **Retroactively alterable** | **Long Term Memory Best Practice**   * **Do not burden long term memory** * **Allow users flexibility to help with memory**  (e.g. Generate their own verification questions) |  |

**Attention, Recognition, and Recall**

|  |  |  |  |
| --- | --- | --- | --- |
| **When focused on a goal, humans pay little attention to things not related to the goal.**  **Attention Blindness** – When occupied with a task or goal, we fail to notice objects and events we would otherwise notice.  **Change Blindness** – When our attention is focused, we fail to notice change. | **Recognition is Easy**  **Recognition = Perception + Long Term Memory**  **Recognition is fast**   * **Faces** * **Threats** | **Recall is Hard**  **Human brain did not evolve to recall facts.**  **Example of Recall Aids:** PowerPoint Slides, account books, address books, calendars | **Examples:**   * Thumbnails rely on recognition rather than recall. * Docked icons (OSX) convey function via recognition (e.g. camera, calendar, calculator). |

**Lecture #13 (03/05) – Gestalt Principles, Color,**

**Vision, Affordance, and Feedback**

**Gestalt Principles - Expanded**

|  |  |  |  |
| --- | --- | --- | --- |
| **Proximity**  Items grouped together are assumed to be related. | **Similarity**  Objects that look similar appear to be grouped. | **Continuity**  Humans are biased to see continuous forms rather than disconnected segments. | **Closure**  Our visual system automatically tries to close open figures in order to see whole objects. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Symmetry**  We parse a complex scene in a way that reduces complexity and gives it symmetry. | **Foreground/Background**  We automatically separate what is in the foreground from what is in the background. | **Common Fate**  Items that **move together** appear grouped. |  |

**Colors and Color Blindness**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contrast vs. Brightness** – Human eyes are optimized for contrast, not brightness. | **Factors Affecting**  **Color Discrimination**   * **Paleness** * **Size** * **Separate Distance** | **Color Blindness** – Affects 8% of men and 0.5% of women.  **Affected Colors**   * **Dark red and black** * **Blue and purple** * **Light green and white** | **Color Blindness Testing** – Check how the OI looks in grayscale to determine how it looks for the color blind. | * **Distinguish colors by saturation (colorfulness) and brightness not just hue (color)** * **Use distinct (significantly different) colors** * **Avoid color blinds that are indistinguishable to the colorblind** |

**Vision**

|  |  |  |  |
| --- | --- | --- | --- |
| **Human Eye**  Composed of three vision detecting components: rods, fovea, and cones.  **Rods** – Detects light levels but not colors (**better in low light**)  **Cones** – Detects red, green, and blue  **Fovea** – Small region in the center of the visual field. 1% of the retina. | **Pixel Density**  Fovea is about **17 times** (158K **cone** cells per square millimeter (mm2) in fovea versus 9K mm2 elsewhere) more cellular **cone** dense than the rest of the retina.  **Data Compression**  Only ganglial neuron per cone cell in the fovea where multiple cone cells per neuron in the rest of the retina (**i.e. lossy data compression**) | Outside the center of the eye, there are more rods which do better in low light. In the center of eye (fovea) there are more cones. | **Peripheral Vision**  Edges of the visual field  **Roles of Peripheral Vision**  **Guide the Fovea** – Eyes move several times a second and visit “interesting” and “crucial” elements in the visual field.  **Detect Motion** – Peripheral vision better at detecting motion.  **See better in the dark** – Rods (more common in peripheral vision) see better in the dark. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Vision and Error Messages** | | **Vision and Searching**  **Make crucial element “pop” to draw the fovea to it.** |  |
| Error messages far away from the item of issue. Relies on peripheral vision to see it. | **Error messages should stand out.** Otherwise, Gestalt similarity principle says they will not be visually distinctive. |

**Affordance and Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Affordance**  Design of the control should suggest (i.e. **afford**) how it should be operated by the user. | **Feedback**  **It should be obvious to the user of a control that the control’s action has taken place.** |  |  |

**Lecture #14 (03/10) – Fitts’ Law, Steering Law, Learning**

**Hand-Eye Coordination and Fitts’ Law**

|  |  |  |
| --- | --- | --- |
| **Hand-Eye Coordination**  Pointing at objects on a display and moving pointers along a constrained path follow consistent quantitative laws. | **Fitts’ Law** **for Pointing**  **at a Displayed Target**  – Time to move to target  – Measure of the ease of starting and stopping motion  – Measure of the average difficulty of moving the hand  – Distance to the target  – Width of the target | 13.1_Fitts_Law_diagram.jpg |
| **Fitt’s Law**   * Users hit on-screen targets **faster the larger and closer they are**. * **Beyond a certain size, making a target even larger adds little benefit.** * **Below a certain distance, making the target closer does not help much.** | 13.2_Fitts_Law_graph.jpg |

**Steering Law**

|  |  |  |  |
| --- | --- | --- | --- |
| **Steering Law** - If a user must keep a pointer within a certain **confined path** while moving it to a target, then the wider the path, the faster you can move the pointer to the target. | 13.8_Steering_law_diagram.jpg |  |  |

**Brain Plasticity and Practice**

|  |  |  |
| --- | --- | --- |
| **Brain Plasticity** – Your brains learns a new perception or behavior by rewiring itself.  **Neurons that previously fired independently now fire in concert.**  **Neurons in a previous network can be rewired into another.** | **Keys to Faster Learning**   * **Practice** that is frequent, regular, and precise. * **Operation** that is task focused, simple, and consistent * **Vocabulary** that is task focused, familiar, and consistent. * **Risk is low** | **Frequency of Practice** – How often a user uses the tool. If frequency of practice is low, guide the user to a solution (e.g. **use the** **wizard pattern**).   * **Habits form faster if practiced regularly**. * Design UI to encourage people to use it regularly. * When practice is precise, neuron networks fire in concert with less “noise.” **Encourage precise and careful practice of an activity**. |

**Gulf of Execution, Task Analysis, and Conceptual Model**

|  |  |  |
| --- | --- | --- |
| **Gulf of Execution** – The gap between what an application user wants and the application’s operation.  **Matching User Expectation/Wants**   1. **Perform task analysis** 2. **Design task focused conceptual model of object-action pairs** 3. **Design a UI based strictly on task analysis and the conceptual model** | **Task Analysis Questions**   * **User’s goal(s)?** * **Necessary tasks to support?** * **Important/common tasks?** * **Results/outputs of each task?** * **Terminology related to tasks?** * **Problems users have when performing tasks?** * **Doer of a task?** | **Conceptual Model** – Model of the application the designers want the users to understand.  **User’s mental model should match as closely as possible with the conceptual model.**  Conceptual model should be **task focused and as simple as possible**. |

**Learning**

|  |  |  |  |
| --- | --- | --- | --- |
| **Consistency** – Users learn faster if an application is **consistent and predicable**.  Unpredictably forces a user to constantly learn something new. | **Task-Focused Terminology**  Vocabulary should be task relevant.  **Examples:** Uses words the user would normally use/associated. Avoid technical jargon. | **Consistent Terminology**  Every concept has **one and only one name**.  **Same name, same thing.**  ***Different name, different thing***. | **Lower User Risk**  Users explore and learn more if the application has less risk.  **Users may be afraid to use unfamiliar features.**  Do not make it more likely a user makes a mistake |

**Perceived Responsiveness and Time**

|  |  |  |  |
| --- | --- | --- | --- |
| **Perceived Responsiveness**  **More important than effectiveness**. Even with good performance, user may perceive poor responsiveness. | **Improving Perceived Responsiveness** | | **Busy indicator** – User for any action that blocks further user action even if the function is executed quickly (**Example:** Hourglass cursor)  **Progress indicator** – Display for any action that takes longer than a few seconds.   * **Show work completed and work remaining** * **Start at 1% not 0%** * **Do not display 100% for more than 2 seconds.** * **Use human scale** precision (e.g.  “4 minutes” instead of “240 seconds”) |
| * **Keep up with the user** * **Do not make the user wait unexpectedly.** * **Keep the user informed of the application’s status** | * **Indicate how long an operation will take.** * **Allow the user to do things while waiting.** * **Anticipate a user’s common requests (e.g. auto-complete)** |

|  |  |  |
| --- | --- | --- |
| **Delays**  A delay **between** unit tasks of a large operation are less annoying than delays **within** unit tasks.  **Delays within a unit task forces users to keep the ultimate goal within working memory**. This can cause a user to lose track of what is happening. | **Task Closure**  **High Level of Closure (Larger Task)**  Less sensitive to time delays.  **Example**: Saving a file  **Low Level of Closure (Smaller Task)**  More sensitive to time delays.  **Example**: Typing a character and seeing it on the screen. | **Display Important Information First**   * **Example #1:** Open a file and display first page while the rest of the information loads. * **Example #2:** Display found items immediately while continuing search. * **Example #3:** For a large image, so a low resolution version first while larger resolution downloads. |

|  |  |  |
| --- | --- | --- |
| **Work Ahead**  **Stay ahead of the user when possible.**  Use low CPU-load periods to pre-compute responses to high probability requests. | **Prioritize User Actions**  **Process user actions with higher priority** not the order they are received. Reorder items in the task queue.  **Example:** Stop button on a web browser. | **Timely Website Design**   * **Minimize the number and size of images** * **Provide quick to display thumbnails or overviews of large amounts of content.** * **Use Cascading Style Sheets (CSS)** * **Use built in browser dialog boxes** * **Use browser side scripting and applets** |