Design Rules

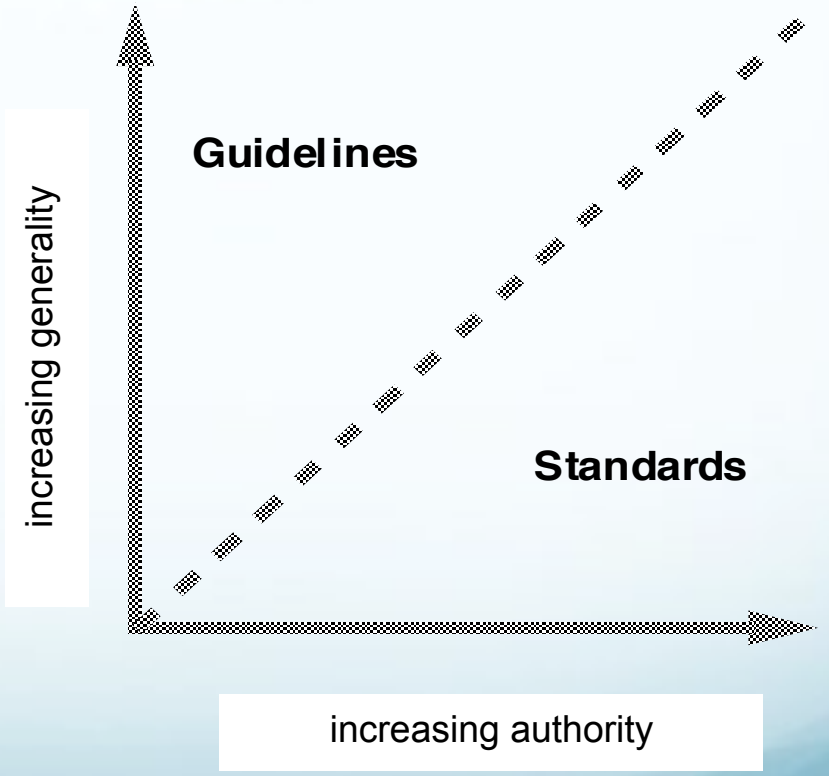
Human Computer Interaction

Design Rules

- Designing for maximum usability
  – the goal of interaction design
- Principles of usability
  - general understanding
- Standards and guidelines
  - direction for design
- Design patterns
  - capture and reuse design knowledge

types of design rules

- principles
  - abstract design rules
  - low authority
  - high generality
- standards
  - specific design rules
  - high authority
  - limited application
- guidelines
  - lower authority
  - more general application



Usability

Usability Engineering

- The ultimate test of usability based on measurement of user experience
- Usability engineering demands that specific usability measures be made explicit as requirements
- Usability specification
  - usability attribute/principle
  - measuring concept
  - measuring method
  - now level/ worst case/ planned level/ best case
- Problems
  - usability specification requires level of detail that may not be possible early in design
  - satisfying a usability specification does not necessarily satisfy usability

part of a usability specification for a VCR

Attribute	Backward Recoverability
Measuring concept	Undo an erroneous programming sequence
Measuring method	Number of explicit user actions to undo current program
Now level	No current product allows such an undo
Worst case	As many actions as it takes to program-in mistake
Planned level	A maximum of two explicit user actions
Best case	One explicit cancel action

ISO usability standard 9241

- adopts traditional usability categories:

- effectiveness
  - can you achieve what you want to?
- efficiency
  - can you do it without wasting effort?
- satisfaction
  - do you enjoy the process?

some metrics from ISO 9241

Usability Objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

Iterative Design and Prototyping

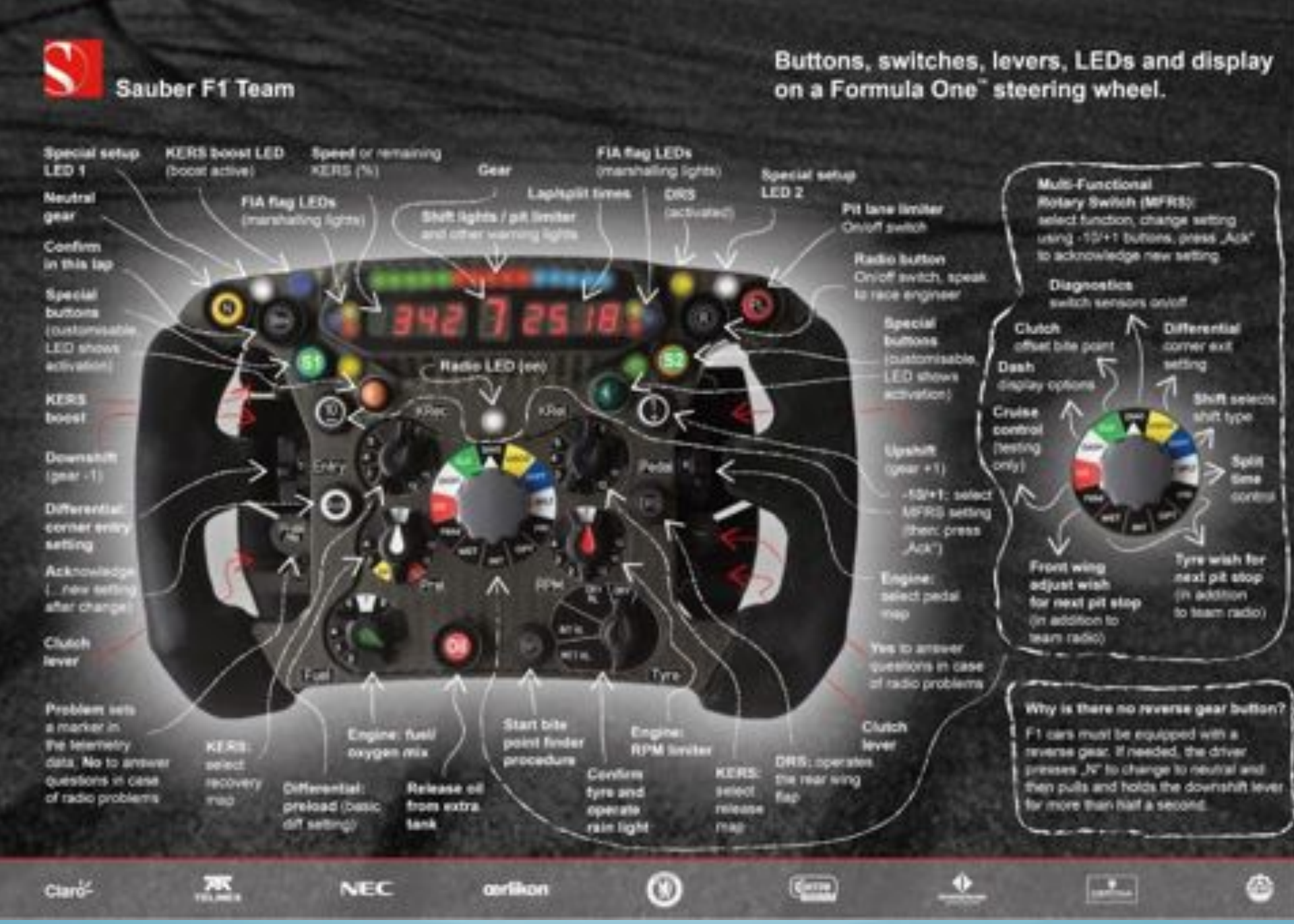
Iterative design and prototyping

- Iterative design overcomes inherent problems of incomplete requirements
- Prototypes
  - simulate or animate some features of intended system
  - different types of prototypes
    - throw-away, incremental, evolutionary
- Management issues
  - time
  - planning
  - non-functional features
  - contracts

Techniques for prototyping

- Storyboards
  - need not be computer-based
  - can be animated
- Limited functionality simulations
  - some part of system functionality provided by designers
  - tools like HyperCard are common for these
  - Wizard of Oz technique
- Warning about iterative design
  - design inertia – early bad decisions stay bad
  - diagnosing real usability problems in prototypes....
  - .... and not just the symptoms


Y
X
B
A



http://www.businessinsider.com/f1-steering-wheel-picture-2013-6

Principles to support usability

- Learnability
  - the ease with which new users can begin effective interaction and achieve maximal performance
- Flexibility
  - the multiplicity of ways the user and system exchange information
- Robustness
  - the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

Principles of learnability

- Predictability
  - determining effect of future actions based on past interaction history
  - operation visibility
- Synthesizability
  - assessing the effect of past actions
  - immediate vs. eventual honesty

Principles of learnability (ctd)

- Familiarity
  - how prior knowledge applies to new system
  - guessability; affordance
- Generalizability
  - extending specific interaction knowledge to new situations
- Consistency
  - likeness in input/output behaviour arising from similar situations or task objectives

Principles of flexibility

- Dialogue initiative
  - freedom from system imposed constraints on input dialogue
  - system vs. user pre-emptiveness
- Multithreading
  - ability of system to support user interaction for more than one task at a time
  - concurrent vs. interleaving; multimodality
- Task migratability
  - passing responsibility for task execution between user and system

Principles of flexibility (ctd)

- Substitutivity
  - allowing equivalent values of input and output to be substituted for each other
  - representation multiplicity; equal opportunity
- Customizability
  - modifiability of the user interface by user (adaptability) or system (adaptivity)

Principles of robustness

- Observability
  - ability of user to evaluate the internal state of the system from its perceivable representation
  - browsability; defaults; reachability; persistence; operation visibility

- Recoverability
  - ability of user to take corrective action once an error has been recognized
  - reachability; forward/backward recovery; commensurate effort

Principles of robustness (ctd)

- Responsiveness
  - how the user perceives the rate of communication with the system
  - Stability
- Task conformance
  - degree to which system services support all of the user's tasks
  - task completeness; task adequacy

Standards

- set by national or international bodies to ensure compliance by a large community of designers standards require sound underlying theory and slowly changing technology
- hardware standards more common than software high authority and low level of detail
- ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks

Guidelines

- more suggestive and general
- many textbooks and reports full of guidelines
- abstract guidelines (principles) applicable during early life cycle activities
- detailed guidelines (style guides) applicable during later life cycle activities
- understanding justification for guidelines aids in resolving conflicts

Golden rules and heuristics

- “Broad brush” design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen’s 10 Heuristics (see Chapter 9)
  - Shneiderman’s 8 Golden Rules
  - Norman’s 7 Principles

Shneiderman's 8 Golden Rules

- 1. Strive for consistency
- 2. Enable frequent users to use shortcuts
- 3. Offer informative feedback
- 4. Design dialogs to yield closure
- 5. Offer error prevention and simple error handling
- 6. Permit easy reversal of actions
- 7. Support internal locus of control
- 8. Reduce short-term memory load

Norman's 7 Principles

- 1. Use both knowledge in the world and knowledge in the head.
- 2. Simplify the structure of tasks.
- 3. Make things visible: bridge the gulfs of Execution and Evaluation.
- 4. Get the mappings right.
- 5. Exploit the power of constraints, both natural and artificial.
- 6. Design for error.
- 7. When all else fails, standardize.

HCI design patterns

- An approach to reusing knowledge about successful design solutions
- Originated in architecture: Alexander
- A pattern is an invariant solution to a recurrent problem within a specific context.
- Examples
  - Light on Two Sides of Every Room (architecture)
  - Go back to a safe place (HCI)
- Patterns do not exist in isolation but are linked to other patterns in languages which enable complete designs to be generated

HCI design patterns (cont.)

- Characteristics of patterns
  - capture design practice not theory
  - capture the essential common properties of good examples of design
  - represent design knowledge at varying levels: social, organisational, conceptual, detailed
  - embody values and can express what is humane in interface design
  - are intuitive and readable and can therefore be used for communication between all stakeholders
  - a pattern language should be generative and assist in the development of complete designs.

Summary

- Principles for usability
  - repeatable design for usability relies on maximizing benefit of one good design by abstracting out the general properties which can direct purposeful design
  - The success of designing for usability requires both creative insight (new paradigms) and purposeful principled practice
- Using design rules
  - standards and guidelines to direct design activity

Questions?