

Requirements and Functionality Principles

Requirements Match (RM)

The set of functions offered by the system (ie. what the system does) should cater for the needs and requirements of the users for whom it is intended. The characteristics of those intended users, and their requirements, should have been accurately determined.

Functional Utility (FU)

The system should provide the best means of performing the required operations. Thus there should be no redundancy or under-provision of system functions, and the way in which functions are organised should be appropriate to the expectations and knowledge of the intended users.

User-System Principles

Navigational Effort (NE)

It should be as easy and efficient as possible for the user to move (in steps or stages) between system states and between functional components. Thus there should be the minimum number of steps between related components and states, and no unnecessary retraction of steps already made.

Memory Load (ML)

Users should not have to make excessive demands on their memories, nor should they have to remember information between states. Thus complex input formats should be avoided where possible (if these are necessary, indication should be given of the required format), and default selections and inputs provided. The same operations should be performed in the same way between states and functional components.

Error Management (EM)

Prevention of erroneous user actions (before the action) is preferable to identification (after the action). User actions with potentially serious consequences should be completely prevented, or warning given before commission to the system. In both cases, the consequence(s) of the error should be indicated, along with any alternative action(s). User actions with less serious or trivial consequences should be immediately retractable. Compound or complex user entries should be retractable and modifiable before commission.

Feedback (FE)

System status (ie. what the system is doing) should be visible to the user at all times. Thus every system state should be labelled, and information always given of system activity, whether user-initiated or user-independent. Immediate confirmation of user-initiated processes should be given, and all system processes should indicate that they are continuing. For processes of any length, indication of elapsed duration or completion time should be given. All continuous user input should be matched by appropriate feedback. All user inputs should be immediately confirmed.

Location and Navigation (LN)

The relationship of every system state to other states should be indicated. Thus as well as labelling, each state should indicate the user options which are available from it. These should always include a return to the previous state; it should not be possible to enter a system state from which there is no exit. Where states represent different functional modes, those modes should be clearly distinguished. In an overlapping state system (eg. windowing), the 'top' state should be clearly indicated. In a multi-state (eg. multi-tasking) system, it should be possible to determine which states are currently active, and to switch between states.

Choice Availability (CA)

At every system state, the range of user options should represent those which are appropriate from that state. Thus neither too many nor too few choices should be available from any state, a balance being maintained between the number of steps required for particular operations and the number of options available at each step. Each option available from each state should be functionally distinct from other options.

User Control (UC)

Users should be encouraged to feel that they and not the system are in control. Thus as far as possible the user should be allowed to initiate and control interactions, and to explore. Excessive use of functional modes (components with distinct and un-related means of performing operations) should be avoided. The tone of error and other feedback messages should be affirmative rather than negative. The user should not have to wait until system processes are complete before initiating new processes.

System - User Match (SUM)

Terminology, language and the assumed level of background knowledge should match those of the intended users. The system should encourage users to create for themselves a coherent conceptual model of its functions and organisational structure. This can rely on metaphors from the user's environment or task domain, using representations of real-world objects and tasks.

User Principles**Modifiability (MO)**

The system should allow users to modify or adapt its functional scope and organisation to fit with their level of experience or preferences. It should also provide shortcuts for user inputs, so that experienced users may reduce the number of steps required for particular operations; these shortcuts themselves may be modifiable.

Flexibility (FL)

Any one operation may be performed in more than one fashion. Thus it may be possible to initiate an operation from more than one system state or component. In mixed-input systems (eg. requiring combinations of keyboard and mouse inputs), it should be possible to perform most operations using one input type (eg. keyboard or mouse) alone.

Appropriateness of Content (APC)

The informational content of the system should be appropriate for its purpose, and no more. Thus the size, format and complexity of each piece of information should be minimally sufficient to convey its intended meaning. This includes messages, feedback and other forms of system output.

Accuracy of Content (ACC)

Each piece of information should be completely accurate, unambiguous and explicit. This includes messages, feedback and other forms of system output.

Salience (SA)

Some informational content may be of particular salience, either exceptionally (unlike the remainder of the system, occurring rarely) or prominently (in terms of its importance for system operation). Such content should be given appropriate emphasis.

Comparative Principles**Consistency (CON)**

Each operation should require the same user actions, resulting in the same system response, regardless of from which functional component it is initiated. Thus the steps required to accomplish each operation should be consistent. Consistency of movement between components should also be maintained. The layout of each state's content should always be the same, as should the range of options available from each state (though some may be unavailable at some times). Terminology and language use should remain consistent across all components and states, as should the format and style of each type of informational content. All messages and feedback should be consistent in style and content. All salient (exceptional or important) content should be consistently emphasised.

Grouping and Linking (GL)

The relationship between different functional components should be consistently maintained. Thus those components which are functionally independent should remain so, while those which are dependent on other components should have those links and dependencies maintained throughout.

System Performance Principles

Manipulability (ML)

Unless there are good reasons for not doing so, the user should have the maximum freedom to switch between states and functional components and to arrange interface elements. Thus in a multi-state (eg. multi-tasking) system, there should be means of access to states which are hidden or occluded, and means of switching between active states. In a graphical system, containing objects such as windows should not be immovable, or prevent interaction with other objects, without reason.

Responsiveness (RP)

Those system components which are physically moveable by the user should present no resistance. Thus windows, icons and other graphical objects should be draggable, expandable (etc.) in real-time; where it is possible to scroll through files or lists, there should be no delay. There should also be no delay in initiation of system processes, nor in response to user inputs.

Perceptual and Motor Principles

Visio-perceptual Load (VL)

The visual load presented by any state (or, in a multi-state system, combination of states) should not be excessive. Thus containing objects such as windows should not appear cluttered, and objects with related functions should be grouped together. Alignment of related objects should be maintained. The number of colours used should be kept to a minimum. In a multi-state system without manipulable (moveable, occludable) states, the number of states which are concurrently visible should be kept to a minimum.

Audio-perceptual Load (AL)

The auditory load presented by the system should not be excessive. Thus audio output should be used sparingly, such as to indicate salience, and volume levels should be adjustable.

Motor Load (ML)

The number of physical actions required of the user should be kept to a minimum. Thus in mixed-input systems (eg. requiring combinations of keyboard and mouse inputs), it should be possible to perform most operations using one input type (eg. either keyboard or mouse) alone.

Perceptual Clarity (PCL)

All text should be readable (via font size, type and line separation), and smaller graphical objects both discernible and distinguishable from each other.

Perceptual Contrast (PCO)

The contrast between visual objects, including text, and their background should be sufficient to discriminate those objects, but should not be excessive. Positive text polarity (dark on light) is preferable to negative polarity (light on dark), and common colour clashes (red-green, blue-black, blue-red, yellow-blue) should be avoided. Bright colours should also be avoided unless used to indicate exceptional salience.

User Support Principles

General Help (GH)

Online help should be provided. It should be possible to search and browse the material in more than one fashion. Principles for navigation through Help facilities are the same as for the remainder of the system, and the means of doing so should be consistent with those elsewhere. The amount of material should not be excessive, and, as a minimum, should focus on the steps necessary to accomplish operations. Illustrations and examples from the system should be used wherever possible.

Context-sensitive Help (CSH)

It should be possible to access specific and relevant parts of the online Help material from every state in the system, especially error states. Material relating to error states should amplify the information given in the original state, including the likely cause(s) of the error and any remedial action(s). It should not merely replicate the Help material relating to the attempted operation.

