INTERNATIONAL STANDARD

ISO 9241-15

First edition
1997-12-15

Ergonomic requirements for office work with visual display terminals (VDTs) —

Part 15: Command dialogues

Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV)

Partie 15 : Dialogues de type langage de commande

Reference number
ISO 9241-15:1997(E)

Contents

	Page
1 Scope	1
2 Definitions	1
3 Application of ISO 9241-15	3
4 Structure and syntax	4
5 Command representation	6
6 Input and output considerations	8
7 Feedback and help	10
Annex A (informative) **Sample procedure for assessing applicability and adherence**	12
Annex B (informative) **Bibliography**	24

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet central@iso.ch
X.400 c=ch; a=400net; p=iso; o=isocs; s=central

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO 9241-15 was prepared by Technical Committee ISO/TC 159, *Ergonomics*, Subcommittee 4, *Ergonomics of human-system interaction.*

ISO 9241 consists of the following Parts, under the general title *Ergonomic requirements for office work with visual display terminals (VDTs)* -

— *Part 1: General Introduction*

— *Part 2: Guidance on task requirements*

— *Part 3: Visual display requirements*

— *Part 4: Keyboard requirements*

— *Part 5: Workstation layout and postural requirements*

— *Part 6: Environmental requirements*

— *Part 7: Requirements for display with reflections*

— *Part 8: Requirements for displayed colours*

— *Part 9: Requirements for non-keyboard input devices*

— *Part 10: Dialogue principles*

— *Part 11: Guidance on usability*

— *Part 12: Presentation of information*

— *Part 13: User guidance*

— *Part 14: Menu dialogues*

— *Part 15: Command dialogues*

— *Part 16: Direct manipulation dialogues*

— *Part 17: Form-filling dialogues*

Annexes A and B of this part of ISO 9241 are for information only.

Introduction

ISO 9241 covers both the hardware and software ergonomic aspects of the use of visual display terminals. The description of the individual parts of ISO 9241, their interrelationships, and a description of the expected users of the parts is described in ISO 9241-1.

ISO 9241-15 is concerned with the ergonomic design of command dialogues. In command dialogues, users input, by recall, either complete or abbreviated command phrases as required by the command language syntax, and the computer performs the actions associated with the commands and their parameters.

ISO 9241-15 serves the following types of user of this part of ISO 9241:

a) The user-interface designer, who will apply ISO 9241-15 during the development process.

b) The buyer, who will reference ISO 9241-15 during the product procurement process.

c) Evaluators responsible for ensuring that products meet the recommendations in ISO 9241-15.

d) Designers of user-interface development tools to be used by interface designers.

e) End-users who will gain from the potential benefits provided by this part of ISO 9241.

The ultimate beneficiary of this part of ISO 9241 will be the end-user at the VDT. It is the needs of these users that provide the ergonomic recommendations in ISO 9241-15. Although it is unlikely that the end-user will read this part of ISO 9241 or even know of its existence, its application should provide user interfaces that are more usable, consistent and that enable greater productivity.

In order to apply ISO 9241-15 within the overall context of the ergonomic requirements for human-system interaction, it is suggested that users be familiar with the following parts of 9241:

ISO 9241-1 *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 1: General introduction*

ISO 9241-2 *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 2: Guidance on task requirements*

ISO 9241-10 *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 10: Dialogue principles*

ISO 9241-13 *Ergonomic requirements for office work with visual display terminals (VDTs) -Part 13: User guidance*

ISO 9241-15 consists of a number of recommendations, some of which are conditional, concerning command dialogues. Conditional recommendations are recommendations which should be met within the specific context for which they are relevant (e.g. particular kinds of users, tasks, environments, technology). These recommendations were developed primarily by reviewing the existing relevant literature and empirical evidence, then generalizing and formulating this work into recommendations for use by the interface designer and/or evaluator. Sources for the individual recommendations are listed in Annex B.

Designers and evaluators using ISO 9241-15 need to know that they are developing an interface that will meet the recommendations provided therein. Likewise, the buyer needs a means to determine how a product matches the recommendations in ISO 9241-15. The elements can be tailored due to the "if - then" structure in ISO 9241-15. Additionally, it is not the intent of ISO 9241-15 that every recommendation should be applied, only those that are relevant.

The application of this part of ISO 9241 is expected to improve the overall quality of the command language, but ISO 9241-15 (like any other standard) will not guarantee the quality of the interface. Quality depends on specific usability criteria as set by the user, buyer or other command-dialogue consumer which may include specifications based on this part of ISO 9241.

It should be noted that ISO 9241-10 describes dialogue principles that are relevant for the design of command dialogues. These principles should provide the designer and evaluator with additional information concerning the ergonomic rationale for the various recommendations in ISO 9241-15 and, therefore, assist in making tradeoffs. However, it may be necessary to base tradeoffs on other considerations as well.

Ergonomic requirements for office work with visual display terminals (VDTs) —

Part 15:
Command dialogues

1 Scope

This part of ISO 9241 provides recommendations for command dialogues used to accomplish typical office tasks using visual display terminals (VDTs). Command dialogues are sequences of instructions provided by the user to the system which, when processed, result in associated system actions. Users input (from recall, rather than selecting from a menu) complete or abbreviated command phrases (e.g. mnemonics, letters, function keys, hot keys in the order required by the command language syntax and the computer performs the activities initiated by the command(s) and their associated parameters.

Interface design depends upon the task, the user, the environment, and the available technology. Consequently, ISO 9241-15 cannot be applied without a knowledge of the design and use context of the interface and it is not intended to be used as a prescriptive set of rules to be applied in their entirety. Rather, it assumes that the designer has proper information available concerning task and user requirements and understands the use of available technology (this may require consultation with a qualified ergonomics professional as well as empirical testing with real users).

ISO 9241-15 applies to the use of command dialogues, either in conjunction with other dialogues (e.g. menus, direct manipulation) or as the primary dialogue technique (e.g. in the case of "dumb terminals" or where high speed is required in a particular application). In addition, this part of ISO 9241 provides recommendations for those "key" commands (i.e. function keys and hot keys) which represent commands within a command dialogue. If the command functions are evident from the nature of their representation (e.g. pictorial icons) and invoking these functions does not require memory on the part of the user, this would not be considered a command dialogue according to ISO 9241-15. Commands can be accessed through other dialogue techniques (e.g. menu options, forms, direct manipulation). However, these methods do not require recall on the part of the user and will be excluded from this part of the standard and will be dealt with in other parts. It also should be noted that ISO 9241-15 does not provide guidance for dialogues which use "natural" language.

2 Definitions

For the purposes of this part of ISO 9241, the following definitions apply:

2.1 argument

Independent variable (including object) used in a command phrase to modify or direct the action of a command.

NOTE Arguments often include parameters.

2.2 command

Whole word, abbreviation, or string of words representing actions requested of the system.

2.3 command dialogue; command language

Command set(s), phrases, structure and syntax associated with a specific interaction of a user with a computer system by means of commands.

2.4 command dialogue structure

Logical structure of the command dialogue (and associated phrases).

2.5 command queuing (stacking)

Accumulation of a series of command phrases in order to allow their input into the system as a group rather than require that they be entered and executed one at a time.

2.6 command phrase

Phrase including the command (words or their abbreviations) and associated separators and arguments (parameters).

EXAMPLE: [Command word] [separator] [argument1] [separator] [argument2] [terminator]

2.7 command set

All of the commands available to the user to perform a given task in a particular application context.

2.8 command syntax

Sequential and other procedural requirements for inputting the components into command phrases.

2.9 command word (name)

Word (or name) used as a command in the command dialogue and representing actions requested from the system.

2.10 command word abbreviation

Shortened version of a command word which is recognizable by the computer as representing the command.

NOTE Such abbreviations may be single or multiple letters of the command word.

2.11 hot keys

Keys, other than numbered function keys (i.e. F1, F2, etc.), not normally used for data entry such as modifier keys (e.g. Ctrl, Alt), or key combinations (e.g. Ctrl/c) which execute immediately without the need for any additional operations.

2.12 keyword

Word in a command phrase identifying a particular argument class (e.g. type font).

2.13 modifier

Argument that alters or limits the action of a command.

2.14 parameter

Value used in conjunction with a keyword to modify the action of a command or argument.

2.15 separator

String of one or more characters, or a pause (for voice), used to separate or organize elements in the command phrase and between command phrases.

3 Application of ISO 9241-15

3.1 Design of command dialogue

In a command dialogue, the command phrase is entered by the user in the specific syntactic arrangement "understood" by the computer. The computer acknowledges receiving the command, indicates whether it is an acceptable command for the current processing state, indicates whether the associated parameters are appropriate for both the command and the current processing state, and if so, performs the requested activities and/or provides the requested outputs. Command phrases may be entered into the computer in a number of different ways, e.g. by means of a "command line", a dialogue box, or by voice input.

Commands may be:

a) Whole words, or strings of words, separated by blanks (pauses, in the case of voice input) or other delimiters, indicating syntax to the computer.

b) Single or multiple letter abbreviations.

Dialogue design determines the way in which a user is guided by the system to make inputs and influences the amount of control the user has over the dialogue. Command dialogues should be designed to support the user in his/her actual work without being bothered by additional work caused by system peculiarities as well as enabling the user to become well-informed and to keep in control of the flow of work (see also ISO 9241-10). Such design goals have to be considered in designing command structure and syntax, command representations, command input and output specifications, and feedback and help mechanisms.

Application of ISO 9241-15 to design and evaluate a system or product requires that the person applying the standard has an understanding of the intended users, their environment and their tasks. User tasks should be listed and the most frequent and important tasks should be explicitly identified. In applying the recommendations, it also is important to consider general laws about human perception, identification and discrimination of information, and psychomotor skills involved in keying in commands.

3.2 Appropriateness of command dialogue

Command dialogues are especially appropriate for one or more of the following conditions, which have been grouped to reflect user and task issues. The applicability of command dialogues becomes greater as more conditions are met.

a) User characteristics

1) Users have good typing skills (if users key in the commands).

2) Users will use the system frequently.

3) Users will receive training on using the command language.

4) Users are familiar with computer technology and command languages.

b) Task characteristics

1) It is not possible to predict the choices of actions that the user may require in the dialogue.

2) Options and/or data may be entered in an arbitrary order.

3) Rapid selection or access to specific system functions is required (in an airline reservation system for example).

4) Extendibility (i.e. creation of new commands, or chains of commands, to suit new situations) is required.

3.3 Applying the recommendations

General ergonomic design objectives are provided in each major subclause of clauses 4 through 7. The individual recommendations aimed at achieving these objectives should be applied within the specific context for which they are relevant (e.g. particular kinds of users, tasks, environments, technology). The format for the individual recommendations is: statement of the recommendation, example (if appropriate), and notes (if appropriate). Examples provided for the various recommendations generally depict an implementation that embodies the recommendation. Some examples also indicate preferred solutions.

Individual recommendations should be evaluated for their applicability and, if judged to be applicable, should be implemented in the relevant command dialogue unless there is evidence that to do so would cause deviation from the design objectives or would result in an overall degradation in usability. When determining applicability, the recommendations generally should be evaluated in the order presented in the relevant clause or subclause. In judging whether applicable recommendations have been met, evaluators should evaluate the product or observe representative users of the product in the context of accomplishing the user's tasks via the command dialogue system. Sample procedures which support the determination of applicability and for judging whether a recommendation has been followed are provided in Annex A.

3.4 Evaluation of products

If a product is claimed to have met the applicable recommendations in ISO 9241-15, the procedure used in establishing requirements for, developing, and/or evaluating the command dialogue shall be specified. The level of specification of the procedure is a matter of negotiation between the involved parties.

Annex A provides a sample procedure that can be used to specify applicability and adherence to this part of ISO 9241. Users of this International Standard can either utilize the procedures provided in Annex A, or develop a comparable set of procedures tailored to their particular development and/or evaluation environment.

4 Structure and syntax

4.1 General

The command language should be designed such that users enter commands in a manner which is natural or familiar to the user without concern for how the computer will process the commands to produce the output (i.e. the command language should reflect the user's needs rather than the computer process and the syntax structure should be consistent with user expectations, task requirements and the input media).

4.2 Internal consistency

The command language should be internally consistent so commands with the same name, function in the same way throughout the application regardless of the context. Commands that do the same thing should have the same name.

NOTE This does not exclude the use of synonyms where appropriate.

4.3 Command macros

If sequences of command words or command phrases are used frequently, users should be allowed to create and use higher level commands (macros) for these sequences.

NOTE Macro commands should follow the same recommendations as commands.

4.4 Argument structures

Command phrases should be structured to minimize the complexity of arguments.

a) Long lists - If argument lists are long (more than 8 arguments), then additional command names should be created, functions should be combined under single arguments, or lists should be broken into some logical functional groupings.

Dependencies - Dependencies between arguments of a command should not dramatically change the meaning of the command phrase.

EXAMPLES: A dialogue uses:

Command "Quit - filename" to save data to the file named filename

Command "Cancel" to cancel without saving (instead of the more complex “Quit -c”)

4.5 Syntax structure

a) Appropriateness for modality - The syntax structure of the command phrases should be appropriate for the input modality (e.g. voice, typed input, gestures).

EXAMPLE: Voice input is used exclusively and the syntax is completely consistent with spoken language.

b) Consistency within modality - Syntax should be consistent within a given modality.

EXAMPLE: For a screen-based command dialogue, the object follows the action (i.e., action - object syntax) throughout the application.

c) Consistency across modalities - Syntax should be consistent across modalities as much as possible.

EXAMPLE: Voice is used as well as typed input for commands in an application and the syntax is object - action for both modalities.

4.6 Command separation

If the input of multiple commands is allowed, a simple and consistent method to separate commands should be used:

a) Blanks - If system constraints do not require the use of a specific separator, blanks should be used rather than punctuation marks to separate commands.

b) Standard symbol - If system constraints require a separator other than blanks to distinguish separate stacked commands, a simple standard symbol should be used consistently.

EXAMPLE: Using the slash (/) in the sequence of command words "SORT/FORMAT/PRINT".

4.7 Language correspondence

Command structure (semantics and syntax) should correspond to the terminology and data organization familiar or natural to the user.

EXAMPLE: The rules for natural language syntax (e.g. English, French) are applied in designing a query language.

4.8 Command arguments

Command arguments should be easy for the user to specify and to relate to the commands that they modify.

NOTE In some cases, it may be appropriate to represent arguments as names rather than single letters.

4.8.1 Command element linkage

The command dialogue should be structured so that the relationship between the command phrase elements is clear.

EXAMPLE: Print pages=1-15 copies=2.

4.8.2 Argument formats

If appropriate to the task, keyword formats (parameters designated by argument identifiers that precede them) should be used rather than positional formats (parameters designated by their sequential position in the argument string following the command).

EXAMPLE 1 (Keyword format): change shape=round color=red size=4

EXAMPLE 2 (Positional format): change round red 4

4.8.3 Placement of optional argument

If keyword formats are not used, optional arguments should be placed at the end of the arguments list.

4.8.4 Separation of arguments

a) Blank space - If blanks are allowed, a variable number of blanks should be allowed between command elements.

b) Other separators - If system constraints require separators other than blanks to distinguish separate arguments, a simple standard symbol should be used consistently.

EXAMPLE: Using the comma (,) in the command phrase "print fileA,fileB,fileC".

4.9 Quantifiers

The use of imprecise or unnecessary quantifiers should be avoided in a command dialogue.

NOTE In query languages, “few” or “many” are imprecise and users tend not to understand what these terms mean.

5 Command representation

5.1 Command names

5.1.1 General

Command names should be easily related to their function, generally stated as verbs (usually in imperative form), be easily remembered by users, and be consistent with the user's task requirements, experience and language usage.

5.1.2 Distinctiveness

Command names should be distinctive.

a) Distinctive meaning - Command names should be semantically distinct and unambiguous.

EXAMPLE: In English, the words insert and delete are more semantically distinct than add and remove (i.e., add and remove typically have many different interpretations).

b) Specific meaning - Command names whose meanings are specific or constrained should be used rather than those that are more general.

EXAMPLE: Use replace rather than change.

c) Visual/auditory similarity - Command names should be avoided that look or sound similar but have different meanings.

EXAMPLE: In English, store and restore should be avoided because they have different meanings but sound similar.

d) Congruent command pairs - If command operations have inverses or counterparts, congruent pairs of commands for these operations should be provided.

EXAMPLE: read/write, open/close, yes/no.

5.1.3 User orientation

Command names should be chosen that are consistent with the user's experience and correspond to the user's operational language.

NOTE If there are multiple user groups, it may be important to provide different sets of command names for these different groups.

5.1.4 Emotional content

Words selected as command words should be emotionally neutral.

EXAMPLE: In English use "cancel" instead of "abort" and use "delete" rather than "kill".

5.1.5 Command word length

If command input is typed, command words should generally not exceed seven characters.

NOTE 1 It may be appropriate to use command words which exceed the seven-character rule when such words would be more natural than an abbreviation (e.g. "allocate" in English, "einfugen" in German).

NOTE 2 See recommendations on abbreviations (5.2) for long command words.

5.1.6 Suffixes and prefixes

Command words should not incorporate unnecessary suffixes or prefixes.

EXAMPLE: In English, delete rather than deleting, deleted, or deletes.

5.2 Abbreviations

5.2.1 General

If users must type commands, they should be able to use abbreviations instead of typing complete commands.

If it is appropriate to the task to provide command abbreviations, these abbreviations should be obvious to the user, easily remembered, and facilitate command input.

NOTE If the command input is an abbreviation and system constraints allow, the "whole" command name may be displayed prior to, or simultaneous with, execution (especially during learning the command language).

5.2.2 Abbreviation rules

a) Simple rule - If command names are shortened, they should be shortened using as simple a rule as possible. That rule should apply to all commands and those arguments that can be abbreviated.

EXAMPLE 1: truncation ("pr" for print)

EXAMPLE 2: dropping of vowels ("prnt" for print)

b) Truncation - If the task requires the user to generate and remember commands, simple truncation should be used to shorten commands.

EXAMPLE: The users are allowed to drop off characters beyond those necessary to make the command unique (e.g. "Q" for QUIT; or in the case of both QUIT and QUERY, then “QUI” is used for QUIT and “QUE” is used for QUERY).

5.3 Function keys and hot keys

5.3.1 General

If function keys or hot keys are used for command input, their use should be obvious to users or the key assignments should be readily accessible and these assignments should be consistent throughout the application.

NOTE Consider using function keys and hot keys for frequently used commands or when it is important to speed up command entry.

5.3.2 Function key consistency

If function keys are used for entering commands, function key assignments for commands should be consistent across related tasks within an application, particularly for "generic" commands like HELP.

5.3.3 Hot key consistency

If hot keys are used for entering commands, such keys should have the same meaning throughout the application.

NOTE If commands can be accessed by menu dialogues as well as typing, the hot key assignments should be the same as the accelerators used in the menus.

EXAMPLE: Alt/c is used for "cancel" and it is used consistently to provide that action throughout the application.

5.3.4 Consistent grouping of modifiers

If modifier keys (e.g. Ctrl or Alt keys) are used with other keys, there should be a consistent application of the modifier key usage.

EXAMPLE: Alt plus other letter keys is used for navigation and window manipulation and Ctrl plus other letter keys is used for data manipulation.

5.3.5 Limited modifiers

Multiple simultaneous modifier keys should be used in hot keys only if there are more commands than can be accommodated meaningfully by single modifier keys.

EXAMPLE: In a dialogue, Alt+p (rather than Alt+Ctrl+p) is used to issue a print command.

NOTE 1 If possible, use letter keys that are mnemonic in combination with modifiers.

NOTE 2 It may be desirable to require the depression of more than one modifier key to reduce the possibility of accidentally causing a destructive action.

6 Input and output considerations

6.1 General

Users should be in control of the dialogue at all times, be able to easily recover from errors, and not be required to input more information than is necessary for successful task performance.

6.2 Command reuse

If the same sets of commands are used repeatedly during a work session, the system should provide a way of reusing the commands without requiring the user to type them again.

EXAMPLE: Giving users a command history list from which they can select a previously used command.

6.3 Command queuing

Users should be provided with the capability to key in a series of commands (command queuing or stacking) rather than wait for the system to execute each individual command.

NOTE Separators should be provided to separate command strings (see 4.5).

6.4 Error correction

If errors occur, re-entry, or editing, should be required only for the erroneous portion of the command and associated parameters.

6.5 Editing

a) Prior to execution - Users should be allowed to edit commands prior to execution.

b) Editing conventions - If the application has a text editor, the same text editing conventions used in the text editor should apply to command dialogue editing.

6.6 Misspellings

If appropriate for the task and system constraints allow, the system should provide for interpretation and acceptance of misspelled commands unless there is ambiguity as to what command was intended.

6.7 Defaults

Defaults should be provided to minimize typing requirements and to facilitate learning.

EXAMPLE: If the disk drive is not identified it is assumed to be the currently set default drive.

NOTE 1 Arguments that have default parameter values are often referred to as optional arguments.

NOTE 2 A typical technique for making command languages less difficult to learn is to build in defaults (representing commonly used parameters of the commands). This allows the user to learn how the commands work without needing to learn complex functions and syntax. After users become familiar with the command's operations using the default values, they can experiment with altering command parameters.

6.8 Destructive commands

If a command may have unintentional or destructive consequences (e.g. deleting a file):

a) Undo - The user should be allowed to cancel the previous (last) command and its effects.

EXAMPLE: Undo command.

or:

b) Confirmation - The user should be required to confirm the intention of the command before command execution.

NOTE See ISO 9241-13 on user guidance for more information.

6.9 Customization

If system constraints allow, users should have the capability to designate and use synonyms for commands and command macros (see 4.2) and they should be able to revert back to the default names when desired.

6.10 Echoing typed commands

a) Consistent input position - The user's input should be displayed (echoed) in a consistent position.

EXAMPLE: Displayed on a "command line" at the bottom of the screen or displayed after the prompt on the screen.

b) Timing - Typed in command characters should be displayed (echoed) as the user types each character.

EXAMPLE: As the user types the command "print", the letter "p" is displayed when the "p key" is depressed, followed by the "r", etc.

6.11 Output control

If appropriate to the task and system constraints allow, the command phrase should allow arguments for redirecting output, interrupting output, or stopping output.

EXAMPLE 1: Users can specify commands so that output is sent to the display or printer.

EXAMPLE 2: Users can specify that the output stops if it exceeds a maximum number of queries.

6.12 Consistent output format

Commands resulting in similar or related output should present their resulting data in a consistent format (see ISO 9241-12).

EXAMPLE: Use of a single presentation format for lists of files, processes, directories, etc.

7 Feedback and help

7.1 General

Feedback and help should provide users with information allowing them to control the dialogue, recognize and recover from errors, and determine their next course of action.

7.2 Command processing

a) Completion - The system should indicate that the command processing has been completed by displaying the output resulting from the command and/or a prompt for the next command.

NOTE is recommended that this feedback be provided within 2 s.

b) Intermediate feedback - If the command processing is expected to continue for a longer period (more than 5 s), visual feedback indicating that the process is continuing should be provided to the user . Also see ISO 9241-13.

EXAMPLE 1: Hourglass with sand (time) running out.

EXAMPLE 2: Repeatedly displaying a message: "working".

NOTE It may be appropriate to provide such information earlier.

c) Processing status - If appropriate to the task and system constraints allow, users should be provided with feedback concerning the relative amount of time remaining to complete the process.

EXAMPLE: A status bar is shown indicating the amount of processing completed by the amount of green shown within the bar area.

7.3 Error feedback

Error feedback should be provided after the full command (including associated parameters) has been entered rather than as soon as the error is discovered by the system.

EXAMPLE: The user misspells the command "print" by pressing the "t" key rather than "r" key and the system indicates the mistake after the entire command has been entered (and not before).

7.4 Error highlighting

If the feedback on a command entry error is provided on the command line and there is an input error, the unacceptable portion of the command should be highlighted (in the context of the full command or a logical portion thereof).

EXAMPLE: The error portion might be highlighted by using reverse video or different color.

7.5 Command information

If appropriate to the task, the user should be provided on request with information on:

a) Commands available and their meaning.

b) Appropriate syntax structure.

c) Required and optional arguments available (especially if the number is large).

d) Command entry history.

7.6 Performance aids

Performance aids should be provided depicting command characteristics (e.g. name, function, syntax, parameters, abbreviation, hot key, function key assignment).

EXAMPLE: Using a keyboard template to depict function key assignments for commands or using a Quick Reference Card to list all available commands and associated information.

7.7 Long argument lists

If a command has a long list of arguments and associated parameters, the use of additional dialogue techniques should be considered.

EXAMPLE: For a command language with numerous arguments, the user can access a dialogue box that has a list with parameter values that can be selected for each command argument.

Annex A
(informative)

Sample procedure for assessing applicability and adherence

A.1 Introduction

Annex A provides an example of a procedure for determining whether the applicable recommendations in ISO 9241-15 have been met. It should be noted that the procedure described below is provided as guidance and is not a rigid process to be used as a substitute for this part of ISO 9241 itself. This procedure provides a two-stage process for
1) determining which recommendations are relevant, and
2) whether those relevant recommendations have been adhered to.

Interface design depends upon the task, the user, the environment, and the available technology. Consequently, ISO 9241-15 cannot be applied without a knowledge of the design and use context of the interface and it is not intended to be used as a prescriptive set of rules to be applied in their entirety. Rather, it assumes that the designer has proper information available concerning task and user requirements and understands the use of available technology (this could require consultation with a qualified ergonomics professional as well as empirical testing with real users).

The evaluation procedure should be based on an analysis of typical users, their typical and critical tasks, and their typical usage environments. Command dialogue evaluations generally fall into the two following categories:

- When users and user tasks are known, evaluators evaluate the product or observe representative users of the product in the context of accomplishing typical and critical user tasks in a typical usage environment.

- When specific users and user tasks are not known, evaluators evaluate all of the commands used in the product being evaluated.

Determination of whether a product meets a given recommendation should be based on the set of commands encountered during the evaluation described above. Command dialogues that can be shown to be better than ones that meet the recommendations described in this part of ISO 9241 would also be accepted as meeting the recommendations of this part of ISO 9241.

Users of ISO 9241-15 could demonstrate how they meet the standard by listing the commands evaluated (e.g. all commands or a task-derived sub-set of commands); the method used to judge applicability (as described in A.3 below); the method used to judge adherence (as described in A.4, below); and the results.

A.2 Applicability

The applicability of a recommendation is based on two factors:

— Whether the conditional statement, if included as part of the provision, is true. A particular recommendation is (or is not) applicable when the conditional if-statement is (or is not) true. For example, if blanks are not allowed, recommendation 4.8.4 a) would not be applicable.

— The design environment. A particular recommendation may not be applicable because of user, task, environment and technology constraints such as unknown user community, variations in tasks, noisy office, screen resolution, or lack of a pointing device. However, if the design environment did involve user characteristics, tasks, or technology features addressed by a particular recommendation, that recommendation would be applicable. For example, if command input was allowed by means of unlabelled function keys or hot keys, conditional recommendations in 5.3 should be evaluated to determine their applicability.

The methods which are appropriate to determine applicability of a particular recommendation are:

a) system documentation analysis

b) documented evidence

c) observation

d) analytical evaluation

e) empirical evaluation

The following clause A.3 describes each of the applicability methods in more detail.

A.3 Description of applicability methods

A.3.1 System documentation analysis

System documentation analysis refers to the analysis of any documents which may describe the general and specific properties of the command language. Such documents may include design documents containing system and user requirements, manuals, user guides, etc. For example according to system requirements for a particular application, only the alphanumeric keyboard will be used for command entry.

A.3.2 Documented evidence

Documented evidence refers to any relevant documented information about the task requirements or characteristics, flow of work, user skills, user aptitudes, existing user conventions or biases, test data from the design of similar systems, etc. Such information may be used to determine whether a given requirement or recommendation is applicable. For example, task analysis data may have indicated that users would frequently input the same sequences of commands.

A.3.3 Observation

Observation means simply to examine or inspect the command dialogue for the presence of a particular observable property, e.g. command parameters are used. Observations can be made by anyone who has the necessary skill to systematically check the command dialogue and determine if it has the particular properties associated with the applicability of given conditional recommendations. Due to their obvious nature, such observations readily can be confirmed by another person.

A.3.4 Analytical evaluation

Analytical evaluation pertains to "informed" judgments concerning the properties of a command language by a relevant expert (i.e. of those properties). This method is typically used for the evaluation of properties which can be judged only in the context of other information or knowledge. In addition, analytical evaluation may be appropriate when the system exists only in terms of design documents, user populations are not available for empirical evaluation, or time and resources are constrained. Analytical evaluation can be used to determine whether a particular recommendation is applicable, e.g. to determine if a command may have unintentional or destructive consequences.

Analytical evaluation can be performed by any suitably qualified person who has the necessary skill and experience to judge the relevant property of the command language. Where these properties concern the application of ergonomic principles, the expert needs to possess appropriate skills in software ergonomics. If the properties concern the work environment, system characteristic, or other aspects of the design, the judge needs to be an expert in the particular relevant domain.

A.3.5 Empirical evaluation

Empirical evaluation refers to the application of test procedures using representative end users to determine the applicability of a recommendation. This method is most appropriate when a prototype or the actual system is available and potential or actual user population representatives are available. Many kinds of test procedures could be used, but in each case the test subjects need to be representative of the end user population and be of sufficient number that the

results can be generalized to the user population as a whole. For example, empirical evaluation to determine whether users will frequently input the same sequences of commands could be done by having typical users perform a number of representative job tasks using the command language.

It should be noted that empirical evaluation needs to be conducted by individuals possessing appropriate skills in testing methodology and evaluation techniques.

A.4 Adherence

If a recommendation is applicable on the basis of the criteria described in A.2, it is then necessary to determine whether or not the recommendations have been met. Adherence is determined by using one or more of the methods listed below (note: the methods which are appropriate to determine adherence for a particular recommendation are listed in conjunction with that recommendation in the Checklist in clause A.7):

a) measurements

b) observation

c) documented evidence

d) analytical evaluation

e) empirical evaluation

It is important to note that the results of applicability tests are often important in determining adherence. The various adherence methods are further described below.

A.5 Description of adherence methods

A.5.1 Measurement

Measurement refers to measuring or calculating a variable concerning properties of the command language. An example of such properties is response time. Adherence is determined by comparing the value obtained from the measurement with the value stated in the recommendation.

A.5.2 Observation

Observation means simply to examine or inspect the command language to confirm that a particular observable condition has been met, e.g. command words do not exceed seven letters, that truncation is used in abbreviating commands, etc. Observations could be made by anyone that has the necessary skill to systematically check the command dialogues and determine if a statement concerning an observable property has been consistently applied. The observed property is compared with the recommendation to determine adherence.

A.5.3 Documented evidence

For adherence, documented evidence refers to any relevant documented information related to the command language's adherence to the appropriate conditional recommendations. Such evidence may include existing user conventions or biases, prototype test data, test data from the design of similar systems, etc. For example, test data from a similar system may have indicated that the command names and syntax utilized in the command language being evaluated was appropriate for the types of users and tasks relevant to the application. In this case, adherence is essentially determined on the basis of documented evidence of adherence for that recommendation for the similar system.

A.5.4 Analytical evaluation

As stated in A.3.4, analytical evaluation pertains to "informed" judgements concerning the properties of a command language by a relevant expert (i.e. of those properties). This method is typically used for the evaluation of properties which can be judged only in the context of other information or knowledge. In addition, analytical evaluation may be an appropriate adherence method when the system exists only in terms of design documents, user populations are not

available for empirical evaluation, or time and resources are constrained. For example, analytical evaluation might be used to determine adherence for using command names that have specific or constrained meanings (5.1.1 b). In the above case, "specific" and "constrained" are the judgemental aspects.

As stated in A.3.4, analytical evaluation can be performed by any suitably qualified person who has the necessary skill and experience to judge the relevant property of the command language. For adherence, the expert also should have the skills and knowledge necessary to reliably judge the appropriateness and usability of a particular design solution. It also should be noted that analytical evaluation can verify the tenability of a design, but cannot validate the design. Validation can be accomplished only by using empirical evaluation.

A.5.5 Empirical evaluation

Empirical evaluation refers to the application of test procedures using representative end users to determine the adherence to a recommendation. As stated in A.3.5, this method is most appropriate when a prototype or the actual system is available and potential or actual user population representatives are available. Many kinds of test procedures could be used, but in each case the test subjects should be representative of the end-user population and be of sufficient number that the results can be generalized to the user population as a whole. The task performance of end-users using the command language could be analysed to determine adherence with the various conditional recommendations. For example, by analysing learning time and keying time and errors, it would be possible to determine if the syntax structure of the command language is appropriate (see 4.4). Such tests could be performed both during the development process (e.g. by rapid prototyping) and after the design and implementation of the system (e.g. by system evaluation techniques) and could be based on both objective and subjective user data. Special tests also could be designed to measure the adherence to a particular recommendation. For example, a learning study could be designed to determine if the rules used for command abbreviations are simple (see 5.2.2).

Typically, empirical evaluations are used to determine adherence by comparing the test results against specific command dialogue recommendations. However, it is often necessary to also evaluate test results in terms of effectiveness (e.g. the command dialogues support the user in his/her task in a manner which leads to improved performance, results in a difficult task being performed with less difficulty, or enables the user to accomplish a task that he/she would not have been able to otherwise).

A.6 Procedure

The following procedure (also see figure A.1: Part 15 Decision Process) can be followed in evaluating a particular command dialogue application with respect to the recommendations in ISO 9241-15:

A.6.1 "If" clause conditional recommendations

a) Applicability - Each recommendation has an if-condition either in the statement itself (e.g. 4.2), or implied in the title to a subclause (e.g. 5.3). For each conditional recommendation, the applicability of the if-statement should be determined using the methods proposed to test if the if-condition is true or not (e.g. in 5.2.2, documented evidence, analytical evaluation, or empirical evaluation is appropriate to determine whether truncation should be used for command abbreviations). Also, when there is a set of optional conditional recommendations such as in 6.7 a) and 6.7 b), the applicable approach should be determined using the proposed method(s). The different sets of optional conditional recommendations are further depicted in the Applicability and Adherence Checklist by the use of and/or logic connectors.

b) Adherence - For each applicable conditional recommendation as defined in a), the adherence to the recommendation should be determined using the proposed methods (e.g. if 5.2.2 is applicable, then observation should be used to determine that truncation has been used for command abbreviations).

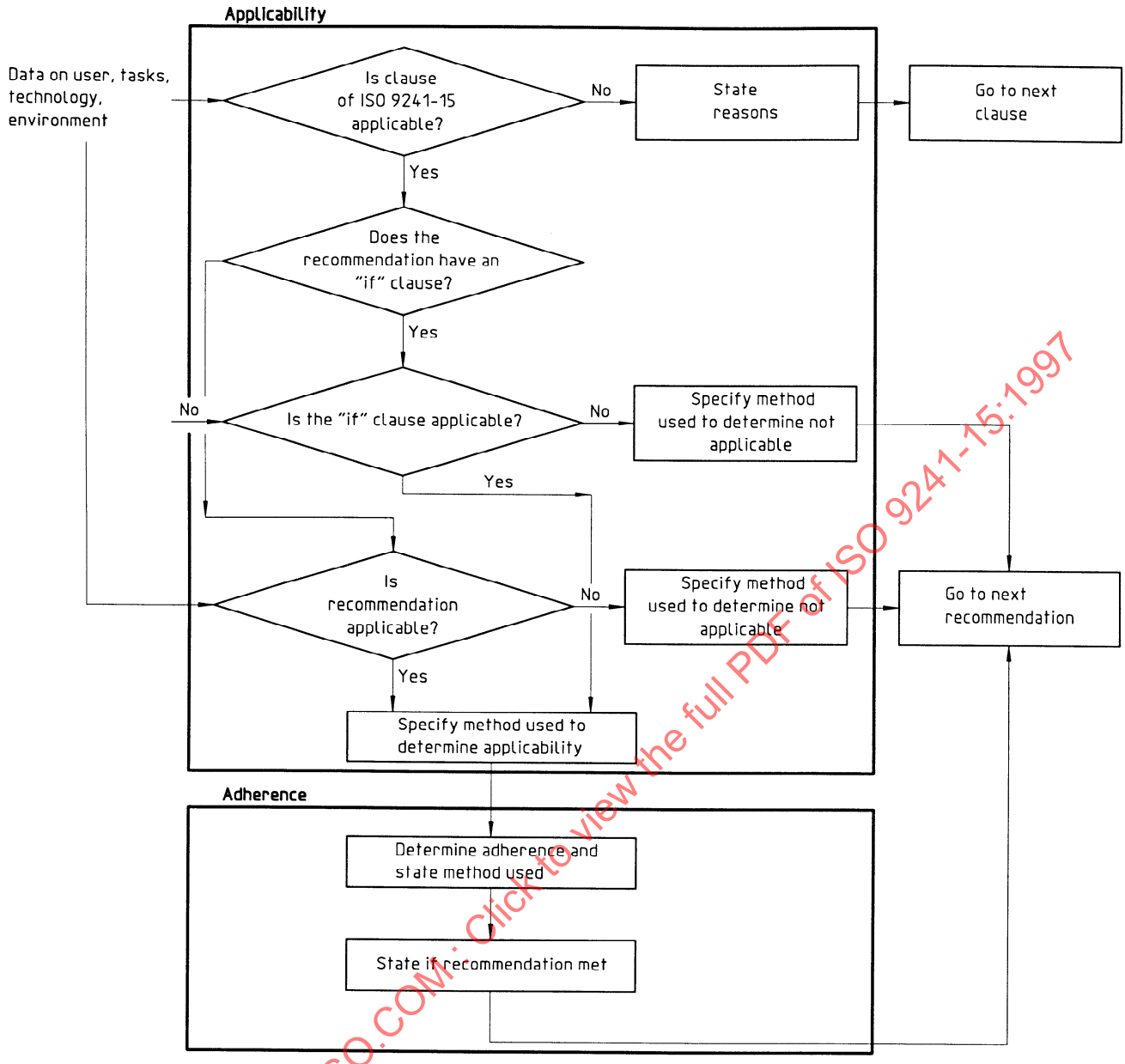

Figure A.1 — Part 15 Decision Process (Evaluation Situation)

A.6.2 Other conditional recommendations

a) Applicability - Non-"if" statement conditional recommendations are generally appropriate to any command language. However, a number of the sections (e.g. 5.3, Function keys and hot keys) are applicable only if the command language utilizes such features. If the command dialogue did use function or soft keys for command input, the conditional recommendations in that section would be applicable (and applicability of the "if statements" would be determined as in A.6.1).

b) Adherence - For each non-"if" statement conditional recommendation as determined in a), information about adherence to the recommendation as described in A.6.1 b) above is necessary. For example, analytical evaluation or empirical evaluation would both be appropriate methods to determine adherence with respect to whether command names are semantically distinct and unambiguous [5.1.1 a)]. If there are valid reasons for not following the proposed recommendation, both the reasons and the design solution chosen also would be of interest to users of this part of ISO 9241.

As an aid for applying the conformance test procedures described above, an Evaluation checklist for applicability and adherence is provided.

A.7 Checklist

The checklist on the following pages is intended as an aid for both designers and evaluators of command dialogues in evaluating both the applicability of, and adherence to, the conditional recommendations in ISO 9241- 15. This checklist contains a "short version" of all of the ISO 9241-15 recommendations and provides a logic structure to assist users in determining applicability. Many of the conditional recommendations allow a number of alternative solutions. The checklist depicts such interdependencies by means of "and" "or" connectors. These connectors are shown only for conditional recommendations within a particular section (it is assumed that the sections have inherent "ands" to the degree that the section is applicable). In some cases, "and/or" is specified because the choices are not mutually exclusive.

NOTE Users of this part of ISO 9241 may freely reproduce the checklist in annex A so that it can be used for its intended purpose and may further publish thier completed checklist.

A.7.1 Description of the checklist

A.7.1.1 Recommendations column

The first column of the checklist contains the "short version" conditional recommendations, connected by the logic connectors, and separated by subsection. Since each conditional recommendation is numbered with its ISO 9241-15 number, users can look up the full text easily in the relevant ISO 9241-15 clauses and subclauses.

A.7.1.2 Applicability columns

The first two columns of the Applicability portion of the checklist are provided for recording the result of the applicability determination by a checkmark in the "Y" or "N" column. In addition, this part of the checklist indicates which of the applicability methods are relevant for each of the conditional recommendations and provides space to "check off" the method used by the designer or evaluator. Those methods that are not relevant for a particular recommendation are shaded to make the checklist easier to use. The codes used for the applicability methods are:

S = System documentation analysis

D = Documented evidence

O = Observation

A = Analytical evaluation

E = Empirical evaluation

DM = Different Method (method other than above used)

If a different method is used (i.e., "DM" is checked), that method can be described in the Comments column. It also should be noted that checking off the applicability methods used is considered an optional feature of the checklist.

A.7.1.3 Adherence columns

This part of the checklist indicates which methods are appropriate for determining adherence to each of the conditional recommendations and provides space for designers or evaluators to "check off" the method used. Those methods that are not relevant for a particular recommendation are shaded to make the checklist easier to use. If the result of the adherence test is positive, the "P" column is checked (for "passed") and if the result is negative, the "F" column is checked (for "failed"). The codes used for the adherence methods are:

M = Measurement

O = Observation

D = Documented evidence

A = Analytical evaluation

E = Empirical evaluation

DM = Different Method (method other than above used)

As for applicability, if a different method is used ("DM" checked), that method can be described in the Comments column. Also as noted for applicability, checking off the methods used to evaluate adherence is considered an optional feature of the checklist.

A.7.1.4 Comments

The comment column provides space for additional statements and comments pertaining to each of the conditional recommendations and can be used to indicate the source of the assessment (e.g. name of expert, title of documented evidence) as well as for describing "Different Methods" when used. Since different solutions (methods) can be appropriate in specific situations, it is best to describe such unique solutions in the comments column. This description can include how these solutions relate to the command dialogue design recommendations and appropriate dialogue principles.

A.7.2 Summary data

Users of the Applicability and Adherence Checklist could summarize the results of the evaluation by computing an adherence rating (AR). The AR is the percentage of the applicable recommendations successfully adhered to with (i.e. the number of check marks in the "P" column divided by the number of check marks in the "Y" column). It is highly recommended that all of the data (i.e. number of Ps and the number of Ys) be reported in conjunction with the ARs. Depending on the complexity of the command dialogue application, it may be useful to complete a checklist for each command set in the system and then average the ARs across the command sets to determine the average AR for the command dialogue application. However, it should be noted that the AR is no more than an arithmetic count which cannot be used as a reliable measurement of the degree of adherence with applicable recommendations without taking into account the respective weights of the items (both by themselves and in the context of use).

Applicability and Adherence Checklist

Recommendations	Applicability								Adherence								Comments (including sources)
	Results		Method Used						Method Used						Results		
	Y	N	S	D	O	A	E	DM	M	O	D	A	E	DM	P	F	
4 Structure and syntax																	
4.2 Internal consistency Commands function in the same way throughout the application regardless of the context.																	
4.3 Command macros If commands used frequently, users allowed to create and use macros.																	
4.4 Argument structures a) Long lists - If argument lists more than 8, additional command names, combine functions under single parameters, etc. and																	
b) Dependencies - Dependencies between parameters of a command should not dramatically change the meaning of the command phrase.																	
4.5 Syntax structure a) Appropriateness for modality - Structure of the command phrases consistent with the input modality. And																	
b) Consistent within modality - Syntax should be consistent within a given modality. And																	
c) Consistent across modalities - Syntax should be consistent across modalities.																	
4.6 Command separation a) Blanks - If delimiters required, blanks should be used rather than punctuation marks. Or																	
b) Standard symbol - If delimiters other than blanks required, a simple standard symbol used consistently.																	
4.7 Language correspondence Semantics and syntax correspond to terminology and structure familiar to user.																	
4.8 Command arguments																	
4.8.1 Command element linkage Relationship between the command phrase elements clear. and																	
4.8.2 Argument formats Keyword used rather than positional formats. Or																	

Key

S = System Document. Analysis
D = Documented Evidence
O = Observation
A = Analytical Evaluation
E = Empirical Evaluation
M = Measurement
DM = Different Method
Y = Yes (if applicable)
N = No (if not applicable)
P = Pass (meets recommendation)
F = Failed (does not meet)

Recommendations	Applicability								Adherence								Comments (including sources)
	Results		Method Used						Method Used						Results		
	Y	N	S	D	O	A	E	DM	M	O	D	A	E	DM	P	F	
4.8.3 Placement of optional arguments If not keyword , placed at end of list. And																	
5.7.4 Separation of arguments a) Blanks - If allowed, variable number allowed between command elements. Or																	
b) Other separators - If other than blanks, a simple standard symbol used consistently.																	
4.9 Quantifiers Imprecise or unnecessary quantifiers avoided.																	
5 Command representation																	
5.1 Command names																	
5.1.2 Distinctiveness Names should be distinctive																	
a) Distinctive meaning - Names should be semantically distinct and unambiguous. And																	
b) Specific meaning - Specific or constrained names used rather than general. And																	
c) Visual/auditory similarity - Names avoided that look or sound similar but have different meanings. And																	
d) Congruent command pairs - If operations have inverses or counterparts, congruent pairs of commands provided. And																	
5.1.3 User oriented Command words chosen that are consistent with the user's experience and operational language. And																	
5.1.4 Emotional content Words selected emotionally neutral. And																	
5.1.5 Command length If typed, command words should not exceed seven letters. And																	
5.1.6 Suffixes and prefixes No unnecessary suffixes or prefixes.																	

Key

S = System Document. Analysis
D = Documented Evidence
O = Observation
A = Analytical Evaluation
E = Empirical Evaluation
M = Measurement
DM = Different Method
Y = Yes (if applicable)
N = No (if not applicable)
P = Pass (meets recommendation)
F = Failed (does not meet)

Recommendations	Applicability								Adherence								Comments (including sources)
	Results		Method Used						Method Used						Results		
	Y	N	S	D	O	A	E	DM	M	O	D	A	E	DM	P	F	
5.2 Abbreviations																	
5.2.1 General Abbreviations allowed for typed commands. And																	
5.2.2 Abbreviation rules a) Simple rule - Shortened using a simple rule that applies to all commands. And																	
b) Truncation - If users need to generate and remember commands, simple truncation used to shorten commands (also see 7.2).																	
5.3 Function keys and hot keys																	
5.3.2 Function key consistency Key assignments consistent across related tasks. And																	
5.3.3 Hot key consistency Key should have the same meaning throughout the application and associated modes. And																	
5.3.4 Consistent grouping of modifiers If modifier keys used with letter keys, logical consistency of the key usage. And																	
5.3.5 Limited modifiers Multiple modifier keys only if more commands than can be accommodated by single modifier.																	
6 Input and output considerations																	
6.2 Command reuse If command sets used often, reuse without retyping provided.																	
6.3 Command queuing Users provided capability to key in a series of commands prior to executing.																	
6.4 Error correction If errors occur, re-entry, or editing, required only for the erroneous portion.																	

Key

S = System Document Analysis
D = Documented Evidence
O = Observation
A = Analytical Evaluation
E = Empirical Evaluation
M = Measurement
DM = Different Method
Y = Yes (if applicable)
N = No (if not applicable)
P = Pass (meets recommendation)
F = Failed (does not meet)

Recommendations	Applicability								Adherence								Comments (including sources)
	Results		Method Used						Method Used						Results		
	Y	N	S	D	O	A	E	DM	M	O	D	A	E	DM	P	F	
6.5 Editing a) Prior to execution - User can edit commands prior to execution.																	
b) Editing conventions - Same conventions as for text editing.																	
6.6 Misspellings System provides for interpretation and acceptance of misspelled commands.																	
6.7 Defaults Defaults provided to minimize typing requirements.																	
6.8 Destructive commands If a command may have unintentional or destructive consequences:																	
a) Undo - Users allowed to cancel the previous (last) command and its effects. Or																	
b) Confirmation - Users required to confirm the intention of the command before command execution.																	
6.9 Customization Users provided capability to designate and use synonyms for commands and command macros (see 5.2) and able to change back to defaults.																	
6.10 Echoing typed commands a) Consistent input position - If the command is typed, the input displayed (echoed) in a consistent position on display.																	
b) Timing - Characters displayed (echoed) as each character typed.																	
6.11 Output control If appropriate to task and constraints allow, user can redirect output, interrupt output, or stop output upon requests.																	
6.12 Consistent output format Commands resulting in similar or related output, presented in consistent format.																	

Key

S = System Document. Analysis
D = Documented Evidence
O = Observation
A = Analytical Evaluation
E = Empirical Evaluation
M = Measurement
DM = Different Method
Y = Yes (if applicable)
N = No (if not applicable)
P = Pass (meets recommendation)
F = Failed (does not meet)