Decide when to test 🡺 can test any time during the nine stages of the document development

Design phase 🡺 predictive test to test the suitability of design specs and production goals, high degree of flexibility.

Writing and draft phase 🡺 remedial test immediate change and re-test to the document, moderate degree of flexibility.

Field evaluation 🡺 evaluative test changes have to wait for next release.

Tie testing to document Goals 🡺 may be you do not have a time to test all your documents, concentrate on how to apply the program to the user’s workplace

A test point 🡺 an issue or feature of a document that might interfere with the efficient and effective application of a program to a user’s work activities.

Test tasks 🡺 with high chance of user failure or high cost.

Test complex task 🡺 one of a kind, high abstract or technical task.

choose type of test :

1 🡺 Performance Tests test whether users can 🡺 successfully complete a given procedure.

2 🡺 Understandability Tests test whether users can 🡺 provide evidence of what they have learned.

3 🡺 Read-and-Locate Tests (can they find it test) test how effectively   
 users can locate a given topic of information in a documentation set

Objectivity 🡺 that you try to set up the test in a way that you don’t prejudice the outcomes too much

The tester🡺 the person who administers the test, arranges the meeting with users, sets up the test situation, records the test activities, and so on.

The evaluator 🡺 the person taking the usability test.

Do a pilot testing**,** Test the test 🡺 is a way of reviewing your test to see if your testing material with gather the kind of info you want them to do.

Instructions 🡺 do the guidelines you give to evaluators are enough to test correctly?

Terminology 🡺 are the terms easy to understand?

Timing 🡺 can the user perform the test in the allocated time?

Testing 🡺 resembles reviewing

Review 🡺 produce comments

The substantive edits (development edit):

🡺 language edit 🡺 clarifying the expression of ideas in a document

🡺 Information edit 🡺 ensuring that all elements of a report work together and in the right order

The copy edits:

🡺 Format edit 🡺 all the parts conform with acceptable format

🡺 Mechanical style edit 🡺 all abbreviations, capitalization are consistence with a specified in house style

Proofreading 🡺 last stage of edit, making sure all the parts match each other

managerial editing 🡺 should attend meetings, edit documents

substantive editing 🡺 should check document as they are being developed and advises the writers on how to organize the contents of the document.

Copyediting 🡺 done after the document complete.

Proofreading 🡺 to double check things.

flip test 🡺 ten second per manual, layout of pages, overall look.

Skimming 🡺 six to ten pages per minute, spelling, punctuation.

reading selectively 🡺 two to three minutes per page, grammar, complete sentences.

reading analytically 🡺 five pages per hour, missing information, technical inaccuracies, paragraph organization.

the long look 🡺 one to two minutes per page, omission in title pages.

Novice editors 🡺 see their work as making document conform to style guideline or looking for mistakes in grammar.

Expert ones 🡺 they do this and more as a level of detail, sentence structure, and language.

Take a constructive attitude **🡺** People often see editors as grammar police.

Consult standard style guides **🡺** when controversies occur you need to consult a general *style guide*

Headers 🡺 help users locate the information within the context of the entire manual.

Introduction 🡺 helps users see the workplace application of function.

Heading 🡺 help the users see the hierarchical structure of information.

Icon 🡺 helps user recognize key information.

cuing pattern (italics, bold) 🡺 helps users recognize key information.

numbered steps 🡺 create clear areas for commands

lists and tables 🡺 helps users decide how to apply program functions to workplace tasks and give users a sense of control.

page numbers 🡺 helps users navigate among abstract concepts.

Color 🡺 helps users recognize task name.

Introduction 🡺 helps users apply functions to workplace tasks.

Icons 🡺 helps users identify screen elements quickly.

cuing (bold) 🡺 helps users recognize keystrokes easily.

hypertext links 🡺 helps users access related commands and tools.

Try out ideas on users,involve users in the process 🡺 eliminate alternatives that would not work for them.

The Design Problem 🡺 The problem of software design results from each reader needing to apply the system to a multitude of tasks.

أتوقع سؤال توصيل

No one carefully reads more then 2 sentences at a time. 🡺 Solution: Make paragraphs short. Include tables and lists whenever possible.

Most users begin using the table of contents before they ready the manual. 🡺 Solution: Make table of contents complete. Use abbreviated, complete and **chapter**-by-**chapter** table of contents.

Most users go to the manual or help only after they have failed to perform tasks. 🡺 Solution: Describe error recovery clearly and completely

Most readers do not read instruction first. 🡺 Solution: Replace introduction with information about users needs, special documents features, or helpful routing information.

Most readers do not read any sections in its entirety. 🡺 Solution: Tell users which section to go for particular tasks/problems.

Navigation 🡺 Navigational aids are elements of a document that tell the reader where to go next for what kind of information.

Cross reference 🡺 point to other sections or chapters with related info.

Layering 🡺 is having two versions of information on the page at once

Indexes and TOC **🡺** They are the two most important user tracking and navigation devices.

Cuing 🡺 it refers to the technique of including visual patterns to make a certain kind of information memorable.

non scrolling regions 🡺 appear on top of screen and stay there.

keyword and whole text searches 🡺 search box.

links and jumps 🡺 allows user to go from one topic to a related topic.

expanded text 🡺 also called “stretch text” allows you to embed more detail into a topic so that the user can click on the expanded text to view the detail.

Indexes 🡺 show an alphabetical view of all the important topics and terminology used in a help system.

pop ups 🡺 to handle glossaries the user just has to click on the term to see its definition.

context sensitivity 🡺 the ability of a help system to present info based on the current state of the program.

Histories 🡺 history buttons allow user to trace their steps, easily go back to previous topics.

Grid lines 🡺 lines drawn where page and column margin would fall.

Margins 🡺 spaces between text and page edge.

Columns 🡺 spaces between the gridlines marking columns

Gutters 🡺 space between columns.

Baseline, grid line 🡺 at the bottom of the text and graphics area that define the bottom margin**.**

degree of modularity 🡺 means breaking the information into chunks of text and graphic units to make them easier for the user to digest.

in online help 🡺 can overcome the modularity problem by pop ups, and hypertext.

degree of structure 🡺 which means that we place the info on the page according to patterns, with certain kinds of info only in certain places.

Two column formats 🡺allow the reader to distinguish between the guidance information and support information.

One column format 🡺 this will arrange both graphics and texts in the middle of the page.

left margin 🡺 which rule the page, so to speak, because most of the items on the page use the left margin as a starting place.

Columns 🡺 newspaper column(snake text) or table columns (discrete item)

headers & footers 🡺 contains product name, version number.

Screens 🡺 full and partial.

Rules 🡺 lines of varying width and length

Pagination 🡺 sequential and modular(2-10)

Windows screen format 🡺 it contains a non-scrolling region, usually it uses one column format.

Manual pages format 🡺 it consists of a handy format for dumping print documentation, it has no left margin.

Multiple window management 🡺 don’t remove or destroy all the traces of user’s work- allow the help screen to cover part of the screen, avoid window clutter.

Graphics 🡺 use simple graphics.

Screen grids 🡺 use narrow margins, less indentation

line spacing 🡺 single space will do it.

page size 🡺 the smaller the page the smaller the size and the less dense font.

Media 🡺 pages allow you to use smaller, more detailed fonts, whereas screens allow more limited range of fonts.

user expectations 🡺 the designer should pay attention to what kinds of type users see regularly in software documents.

Headings 🡺to help reader locate important info. easily distinguished from the body text.

Hints, notes, and cautions 🡺 they must read easily and should catch the reader’s eye.

User input, computer output 🡺 writers usually change the font of input and output messages from that of body text.

Tables and lists 🡺 make the tables different in indentation and column layout.