A Report

on

"Usability Engineering"

In partial fulfillment of requirements for the degree

Fourth Year Computer Engineering

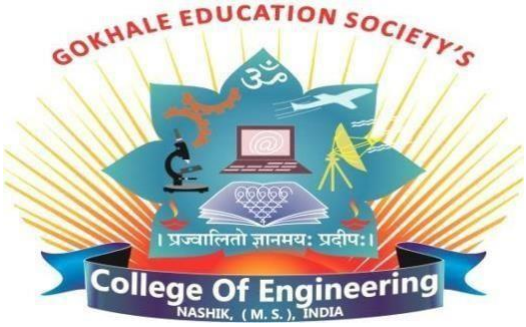
By

Name of the Candidates:

STUDENT NAME	PRN NO.
Ms. Pawar Urmila (A: 69)	**71721570K**
Ms.Deore Komal(B: 46)	**71837348M**
Ms. Rajput Pratiksha (B: 62)	**71837701L**
Ms. Talele Dhanshri (B: 11)	**71721634K**

Under the guidance of

Prof.V.S.Nikam



DEPARTMENT OF COMPUTER ENGINEERING

University Of Pune

Gokhale Education Society's

R. H. Sapat College of Engineering,

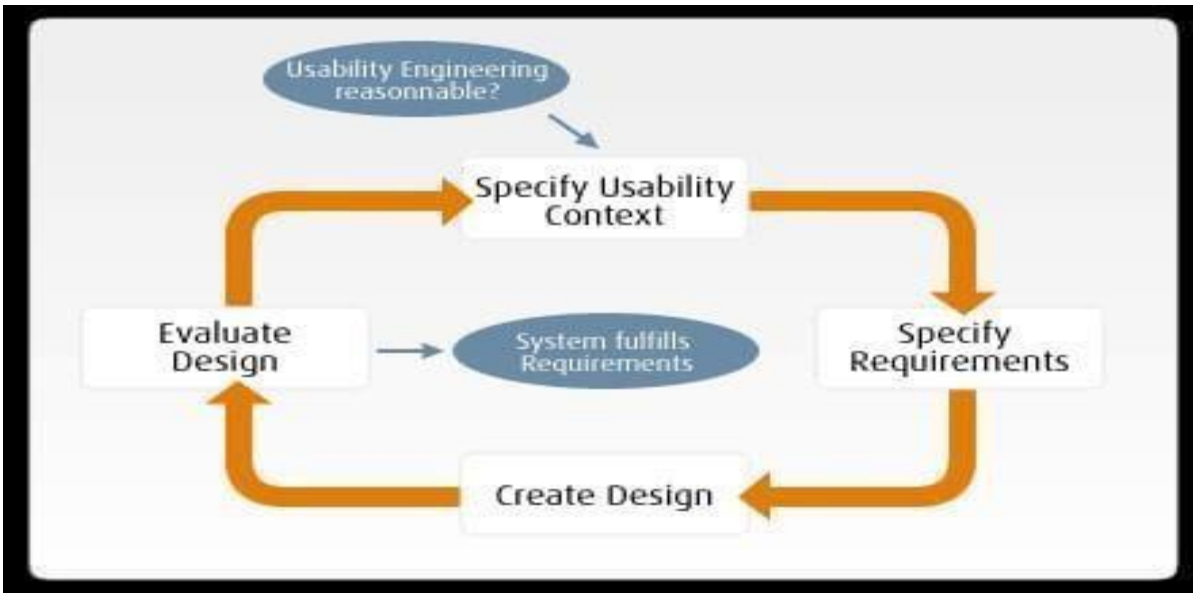
Management Studies and Research, Nashik

- 422 005, (M.S.), INDIA [2019– 2020]

Introduction

The history of usability engineering in this context dates back to the 1980s. In 1988, authors John Whiteside and John Bennett—of Digital Equipment Corporation and IBM, respectively—published material on the subject, isolating the early setting of goals, iterative evaluation, and prototyping as key activities. The usability expert Jakob Nielsen is a leader in the field of usability engineering. In his 1993 book *Usability Engineering*, Nielsen describes methods to use throughout a product development process—so designers can ensure they take into account the most important barriers to learnability, efficiency, memorability, error-free use, and subjective satisfaction *before* implementing the product. Nielsen's work describes how to perform usability tests and how to use usability *heuristics* in the usability engineering lifecycle. Ensuring good usability via this process prevents problems in product adoption after release.

Usability Engineering is a professional discipline that focuses on improving the usability of interactive systems. It draws on theories from computer science and psychology to define problems that occur during the use of such a system. Usability engineering involves the testing of designs at various stages of the development process,experts.



Computer system designers have not always adopted a user-centered perspective on software design. Instead, many designers resolved design questions about the human-computer interface by using introspective criteria such as personal preference or conceptual appeal.

This introspective approach to user-interface design might produce a usable system when software engineers represent actual users.However, computer systems today are being built for a wide range of people whose needs often have little in common with the needs of system designers.

Rather than focusing on finding solutions for usability problems— which is the focus of a UX or interaction designer—a usability engineer mainly concentrates on the *research phase*. In this sense, it is not strictly a design role, and many usability engineers have a background in computer science because of this. Despite this point, its connection to the design trade is absolutely crucial, not least as it delivers the framework by which designers can work so as to be sure that their products will connect properly with their target usership.

In response to market demand for systems that satisfy a growing and varied user community, usability is becoming an increasingly important competitive issue. Designers are striving to create computer systems that people can use easily, quickly, and enjoyably. Indicative of this trend is increased membership since 1982 in
professional groups such as the Association for Computing Machinery's Special Interest Group on Computer-Human Interaction (ACM SIGCHI) and the Computer Systems Group of the Human Factors Society.

Digital's Software Usability Engineering Group believes that engineers must learn about the needs and preferences of actual users and should build systems to accommodate them. With an understanding of customer environments, an awareness of technological possibilities, and imagination, we have produced many idea.

Features & Methods of Usability engineering

- **Usability Engineering in a Nutshell**

To understand why Snapchat's usability has been so vigorously discussed by UX advocates, we first should define the difference between usability and user experience.

User experience spreads beyond simple understanding and complying to a user's desire. It often goes along with Intraction & design, engineering, marketing, and user interface.

Usability means evaluating users' needs and requirements to create a product that will be easy and comfortable to operate. According to Jakob – the man often called "the king of usability" – the main

components of usability are: *learnability*, *efficiency*, *memorability*, *handling errors*, and *satisfaction*.

However, there's a third term that describes the methods of identifying usability problems and offering valuable recommendations. **Usability engineering** is a practice aimed at creating highly effective and user-friendly software products using evaluation methods.

So, what is usability engineering exactly? Good usability specialists work with care for customers in mind and advocate for making user experience as smooth and efficient as possible. It usually means:

1. Understanding basics of cognitive science and psychology
2. Evaluating usability with the help of testing and analysis
3. Working by standards already set in the industry
4. Conducting studies on possible problems and offering solutions

But evidently, the usability evaluation is one of the main and most impactful tasks a usability engineer will have to undertake. Let's explore some of the methods.

- **Usability testing**

This practice is the simplest one yet extremely effective. During a usability test participants are asked to complete typical tasks using the product tested. Moderators observe, listen, and take notes. An effective test, one that will help you get the most insight in the shortest amount of time, requires preparation

- **Usability Evaluation Methods**

Every convenient design is user-centered. Building **user-centered design** implies a series of design techniques focused on

understanding what users need, experience, and value. You should consider how your product will be perceived by a customer in the early stages of product development.

1. **HiDevelop a plan.** Specify which section of a product the test will cover, indicate a specific purpose and a series of questions to be answered, prepare the metrics (both qualitative and quantitative) to measure the satisfaction levels and time spent to complete a task.
2. **Recruit participants.** There are two main conditions your test subjects should meet: They can't be involved in either the design or development of this product and they should represent your target audience. You can use paid services to find eligible participants or engage people in real life – the popular method called guerilla testing.
3. **Run a pilot test.** To make sure that the equipment and your note-taking method suit the job and that you are sure how well participants will understand your scenarios, conduct a pilot test a couple of days before your first test session.
4. **Choose a moderating technique.** Deciding how you're going to moderate a test directly depends on your initial testing goals. There are many techniques you can try, among which are thinking aloud testing and probing questions during or after a test.

The method also involves a wide range of possible techniques: You can do it remotely or on site, both with experts and casual participants and with any budget. When conducting usability testing, don't give users tips or ask leading questions

The Software Usability Engineering Process

The role of engineering is to apply scientific knowledge to produce working systems that are economically devised and fulfill specific needs. Our software usability group has adapted engineering techniques to the design of user interfaces. To understand user needs, engineers must observe people while they are actually using computer systems and collect data from them on system usability.

Observation and data collection can be approached in the following ways:

1. Visiting people while they use computers in the workplace
2. Inviting people to test prototypes or participate in usability evaluations at the engineering site
3. Soliciting feedback on early versions of systems under development
4. Providing users with instrumented systems that record usage statistics

Our software usability engineering process evolves as we use it in product development. As of 1987, the process consists of three principal activities:

1. Visiting customers to understand their needs. By understanding a customer's current experience with a system, we gain insight into our opportunities to engineer new and better systems. We collect data on users' experiences

primarily through contextual interviews, that is, interviews conducted while users perform their work.

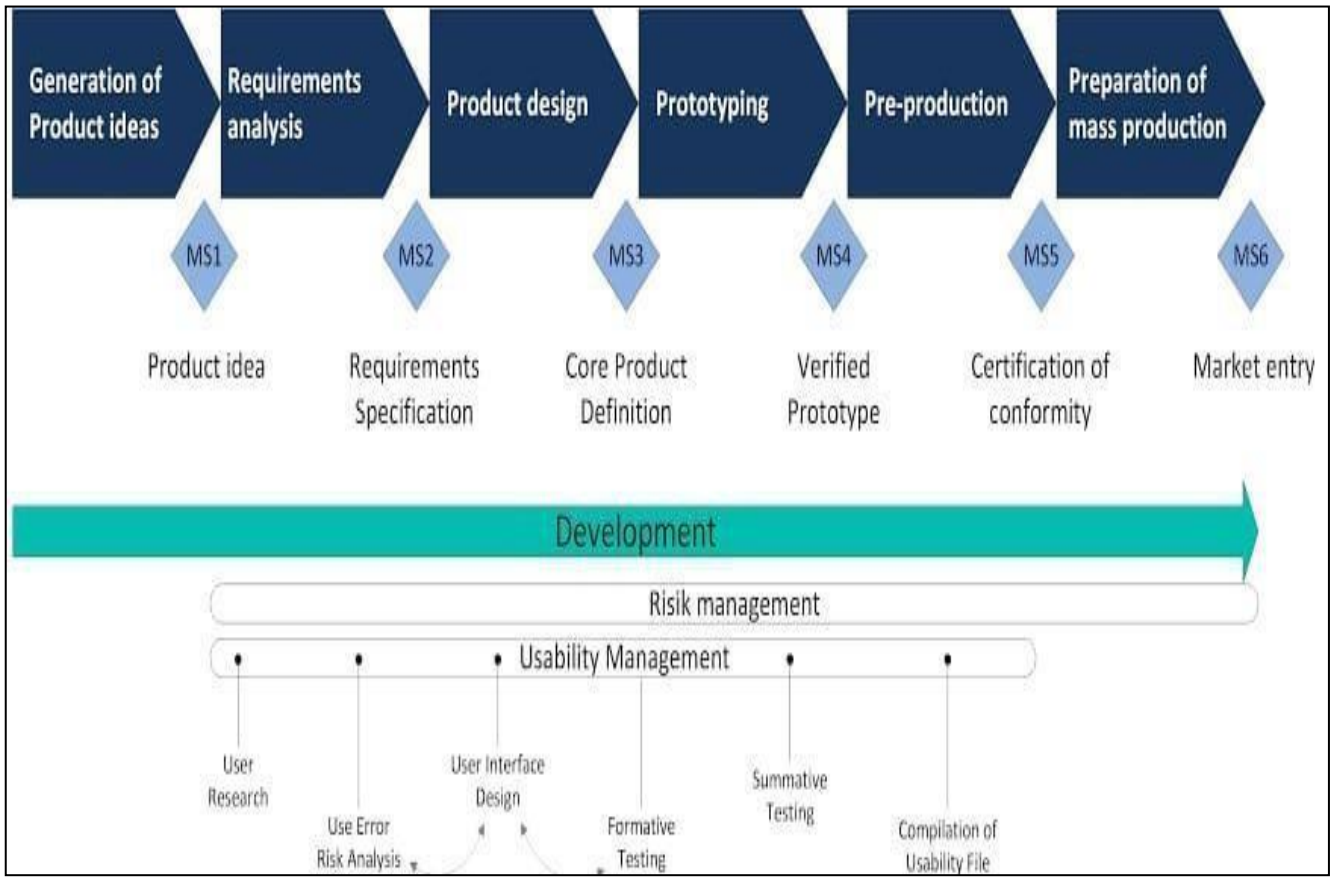
2. Developing an operational usability specification for the system. We base the system specification on our understanding of users' needs, competitive analysis, and the resources needed to produce the system. This specification is a measurable definition of usability that is shared by all members of the project team.

3. Adopting an evolutionary delivery approach to system development. Developers start by building a small subset of the system and then "growing" the system throughout the development process. We continue to study users as the system evolves. Evolutionary delivery is an effective method for coping with changing requirements — a fundamental aspect of the development process.

These three development activities are parallel, not sequential. We do not view user-interface design as a separate and initial part of the development process but as an ongoing process in system development.

These usability engineering techniques apply to most software development environments and are most effective in improving software usability when applied together. However, designers who use any single technique can improve a system's usability.

Flow diagram of Usability Engineering in SDLC:



The Usability Engineering Life Cycle

No matter how you decide to obtain your requirements, you should ensure that you have covered the following points in your requirements analysis. Even if you did create a paper prototype or observe the user at work, be sure to review the following points to ensure that you have all the bases covered.

Bias and Mayhew (2005) created the Usability Engineering Life Cycle (UEL) as a means to build a usability test plan. If you can integrate the UEL into your product development cycle at the beginning, it will provide you with a rigorous analysis and testing

regimen that will help you get the most out of your usability design, analysis, and testing.

- **Level 1 Design**

Level 1 design is the conceptual design level, which is where you design functionality, workflow, and rules. If you and your team have the time, you should get as much information from the users as possible before you decide how to design conceptual models. Models conceived from user input stand a far better chance of being accepted by users during the design evaluation stage in Level 3. The four steps in this level are as follows:

1. **Work re-engineering**—Your project team organizes functionality and workflow design based on the users' tasks and streamlines work before you begin design. No interface design is produced in this task.
2. **Conceptual model design**—The team creates high-level design rules for presenting information and interacting with the hardware, software, or Web site interface. If you have product screens or Web pages, this task doesn't go into that level of detail.
3. **Conceptual model mockups**—You can create paper prototype mockups, as you will learn about in Chapter 4. You can also create wireframe versions, which are small programs that show some functionality but not the entire program, or you can even create a prototype with nonoperating functionality such as small colored paper squares that represent lights on a hardware prototype.

4. **Iterative conceptual model evaluation**—The project team evaluates the mockups and modifies them through iterative evaluation processes. In other words, if the team decides it doesn't like one or more portions of the mockups, it works on those portions repeatedly until it decides that the portion looks good.

- **Level 2 Design**

Level 2 design is where you create the standards for your project. Creating standards is especially important because everyone on the team needs to understand how the project will be put together. Having people creating their own standards as you develop the user interface design is a recipe for chaos. Four steps comprise this design level.

1. **Design standards**—Now that you have settled on a model, the project team must construct a set of interface- or site-specific standards and conventions that will apply to the design of the product.
2. **Design standards prototyping**—The project team applies the interface standards to product functionality. This functionality can be presented in specific screens or Web pages that you create to test the look and feel as well as links to other screens or Web pages.
3. **Iterative design standards evaluation**—The project team conducts formal usability testing or other types of evaluation to refine the screen design standards in the interface. This process continues until major usability issues have been resolved and usability goals are within reach. You'll learn more about usability testing .

4. **Style guide development**—After you have a stable and validated set of screen design standards, you document this information along with the results of the requirements analysis in the product style guide and then distribute the documented information to all project team members. Other style guides, such as a general style guide for the company and the documentation style guide, could also affect the product style guide, and vice versa.

- **Level 3 Design**

Level 3 design is the level at which you actually design the product after making all your preparations in the previous two levels.

1. **Detailed user interface design**—The project designers design the product based on the style guide conventions created in Level 2 design. The product that results is the "beta" version available for internal or external testers to use and test and for which they can provide feedback for the product team.
2. **Iterative detailed user interface design evaluation**—The project team conducts formal usability testing or other typee evaluation to refine the screen design standards in the interface. This process continues until the project team validates the product against usability goals.

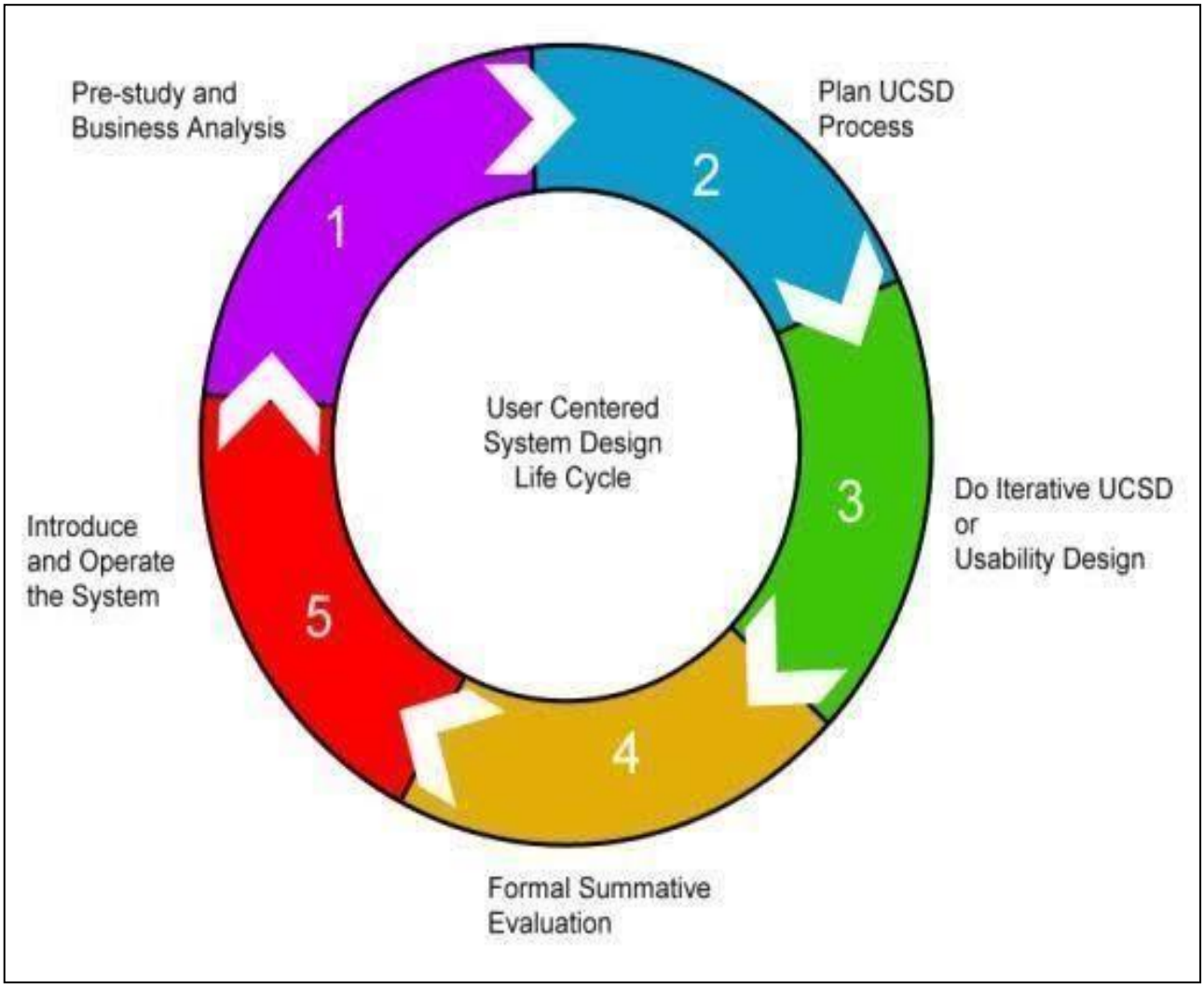
Installation and Feedback

After the product has been installed and used for a period of time, the company should gather feedback from users about what they like and don't like about the product and how they use it.

You can obtain feedback in any number of ways: by e-mail, phone, mail, or on your Web site. You can send surveys to customers, and you may want to offer prizes or special offers to entice customers to return the surveys, especially if the surveys are long. You may also want to conduct focus groups either in person at your company building or at the client, or online using a collaborative software tool that employs real time videoconferencing such as WebEx, Microsoft LiveMeeting, or Raindance.
The Never-Ending Process

One thing to keep in mind is that the UEL really never ends. Feedback during the development process will ensure that you don't have many problems to fix after the product is out the door—and good feedback is always a feather in your company's cap. You will also need product feedback from your customers after the development process ends.

In addition, you may have upgrades to your product that need to be produced—or updates to the documentation you may want to place on the company Web site. So be sure to include the additional costs of implementing continual feedback as needed, especially between.
Lifecycle:



Heuristic Evaluations of Usability Engineering:

The simplicity of heuristic evaluation is beneficial at the early stages of design. This usability inspection method does not require user testing which can be burdensome due to the need for users, a place to test them and a payment for their time. Heuristic evaluation requires only one expert, reducing the complexity and expended time for evaluation. Most heuristic evaluations can be accomplished in a matter of days. The time required varies with the size of the artefact, its complexity, the purpose of the review, the

nature of the usability issues that arise in the review, and the competence of the reviewers. Using heuristic evaluation prior to user testing will reduce the number and severity of design errors discovered by users. Although heuristic evaluation can uncover many major usability issues in a short period of time, a criticism that is often leveled is that results are highly influenced by the knowledge of the expert reviewers. This "one-sided" review repeatedly has different results than software performance testing, each type of testing uncovering a different set of problems.

The main goal of heuristic evaluations is to identify any problems associated with the design of user interfaces. Usability consultants Rolf Molich and Jakob Nielsen developed this method on the basis of several years of experience in teaching and consulting about usability engineering. Heuristic evaluations are one of the most informal methods of usability inspection in the field of human–computer interaction. There are many sets of usability design heuristics; they are not mutually exclusive and cover many of the same aspects of user interface design. Quite often, usability problems that are discovered are categorized—often on a numeric scale— according to their estimated impact on user performance or acceptance. Often the heuristic evaluation is conducted in the context of use cases (typical user tasks), to provide feedback to the developers on the extent to which the interface is likely to be compatible with the intended users' needs and preferences.

The final set of heuristics that are still used today were released by Nielsen in 1994. The heuristics as published in Nielsen's book Usability Engineering are as follows:

1. **Visibility of system status**:

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. **Match between system and the real world**:
   The system should speak the user's language, with words, phrases and concepts familiar to the user, rather than system- oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom**:
   Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. **Consistency and standards**:
   Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention**:
   Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. **Recognition rather than recall**:
   Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.
   Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use**:

Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design**:
   Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. **Help users recognize, diagnose, and recover from errors**:
   Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

The most effective usability objectives contain several key elements:

1. Task – including a quantitative goal so that success can be easily defined, usually in terms of time, accuracy, task completion rate, etc. The highest risk tasks and the most important tasks for using the device effectively should be included when defining tasks.
2. User profile – specific user group that will perform the task.
3. Use environment – context of device use for a particular task, including physical and social environmental factors. Use environment is not always specifically called out in the usability objective but is often implied. It should, at a minimum, be defined for the device in the usability documentation and considered when developing usability objectives.

The value of Usability Engineering

Focusing on and prioritizing Usability Engineering activities will generate numerous cascading benefits:

1. Medical devices that can be used safely and effectively
2. Fewer late-stage changes in the development, ultimately leading to faster time-to-market
3. Regulatory compliance – Even though not always submitted, documentation is still required and can be requested by authorities
4. Simpler and more intuitive product user interfaces
5. Minimized need for training and support
6. Improved user satisfaction leading to improved sales
7. Improved market life and customer loyalty
8. Minimized risk of liability claims and product market withdrawals

This list clearly illustrates that even though Usability Engineering activities takes time and resources, it is definitely an investment worth making.

Save time and money by focusing on Usability Engineering

For some, it might seem obvious that for a product to be successful, the User Interface needs to be easy, safe and effective for the users to use, which requires a dedicated focus on designing and evaluating the interface. For others, it might seem excessive to spend a lot of resources involving users – "we know how the users will interact with the product" – seems to be the opinion.

However, our experience is that there is great value and potential savings in both time and money in focusing on Usability Engineering when developing products, and Medical Devices in particular.

Three levels play important roles in the use of a product: Utility, Usability and User Experience.



- **Utility** – does the product fulfill the users need by having the right features for the users to obtain their goals?
- **Usability** – is the product safe, effective, easy and efficient to learn and use? In medical devices, the focus is on safety and effectiveness.
- **User Experience** – does the user have a good experience when using the product? This includes the users' satisfaction and aspects surrounding the product itself