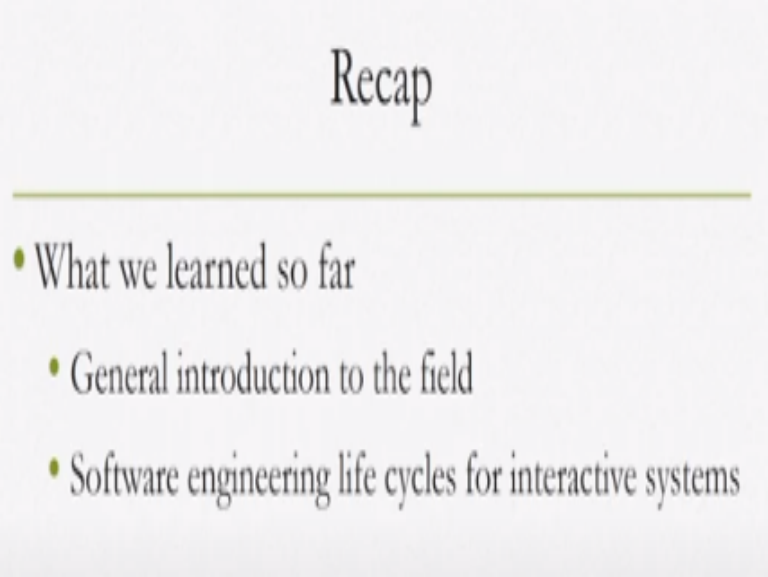
Design and Implementation of Human – Computer Interfaces
Prof. Dr. Samit Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Module No # 01
Lecture No # 05
Usability Requirements

Hello and welcome to the NPTEL Mooc'S course on design and implementation of human computer interfaces. We are going to start a new lecture number 5 on the topic of usability requirements. So essentially it will deal with how to gather requirements to ensure that our product is usable here by product we mean a software application.

(Refer Slide Time: 01:21)



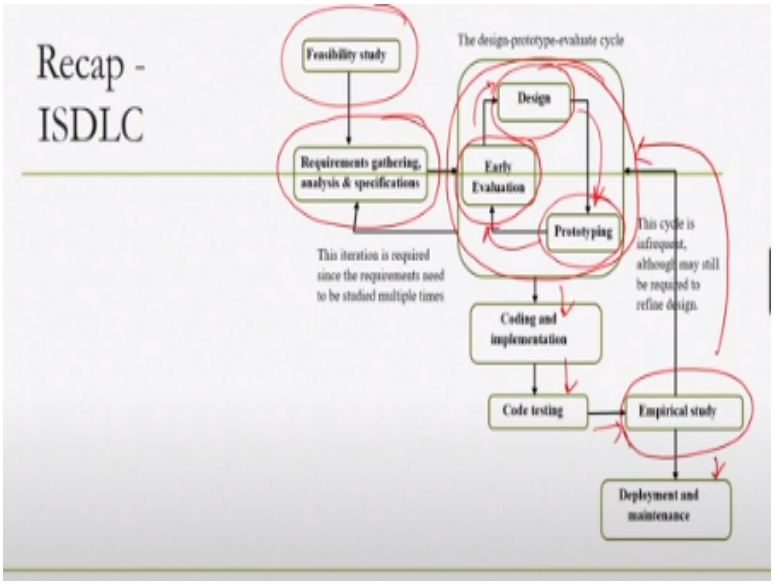
Before we start again as usual we will quickly recap what we have covered in the earlier lectures. So we have talked about interactive systems, which is our core concern here interactive systems and human computer interfaces will be using simultaneously and synonymously. So that term human computer interface will be used synonymously with interactive systems and most of the time we will be using the term interactive system for brevity which will refer to human computer interfaces.

Now when we are talking of interactive systems essentially what we are referring to is a software system which will be used by layman users. So earlier we have learned what are these systems

why we need to be extra causes while designing these systems? What are the things that we should consider while designing such systems? In particular we have focused on the crucial concept of usability.

Now in order to ensure that an interactive system is usable what we should follow that also we have briefly discussed in our earlier lectures. So essentially what we mentioned is that there has to be a systematic and stage wise process to develop an interactive system. Now this systematic development process we are referring to as software development life cycle. So we have seen one such software development life cycle tailored to cater to the need for development of usable interactive systems.

(Refer Slide Time: 03:25)



Let us quickly have a look and recap the development life cycle as I mentioned earlier. So we will be focusing on the individual stages of this life cycle throughout this course the details of these stages will be the main content in this course. So what are the stages there present in the interactive development life cycle? First stage is feasibility study where we first try to see whether the development process is feasible or not with respect to the available resources that includes manpower, cost, time computing resources and so on.

Once it is found to be feasible we go for the second stage that is requirement gathering analysis and specification. In this stage we try to gather requirements end user requirements which will help us develop a system that is going to be usable. Once that is done we enter into this design

prototype evaluate cycle. So, these 3 together constitute a stage now these sub stages in this stage work in a cycle.

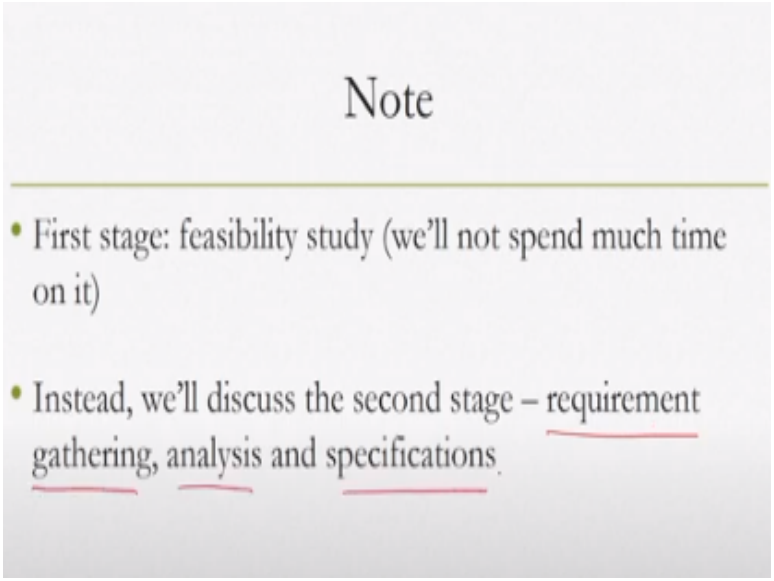
So first we go for designing the system particularly the interface and interaction that design is prototyped for quick feedback on the quality of the design and usability of the design. Based on feedback we refine the design then again refine the prototype or create a new prototype for the revised design. Which again is evaluated and this cycle goes on until we reach a stage where no further changes in design is required.

So once we stabilize on the interface and interaction design or rather once we finalize the design then we go for code design. So essentially we have to build software so we need to design the system so from the design of the interface and interaction we convert it to design of a system or the code design part. That design is then implemented and the code the executable code that we have implemented is then tested in subsequent stages.

Once the testing is over and we get a working system we then have to go for a final round of usability study for the product that is an elaborate process which generally is termed as empirical study or empirical research. In this stage it is possible that we may find some more issues with our design of the interface and interaction. So it may require us although infrequently but still sometimes we may need to go back to this cycle to refine our design and then again recode retest and then finalize the product.

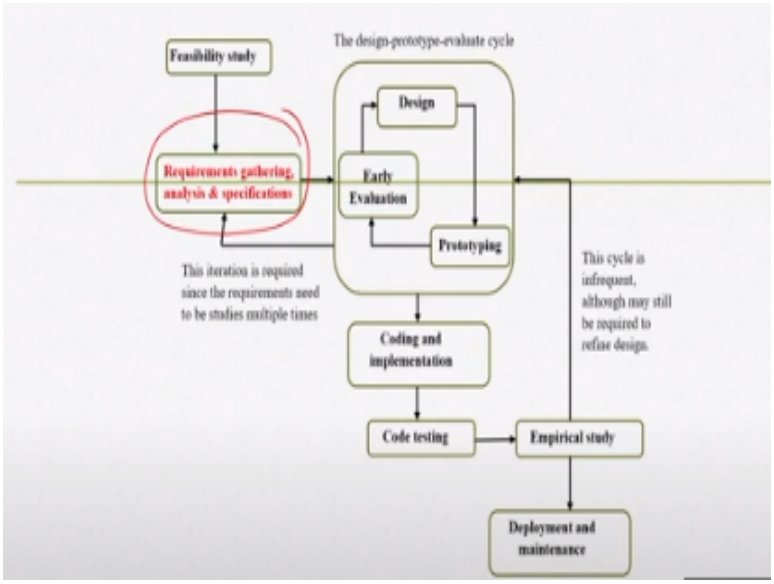
So once finalization of the product is done then we go for deployment of the product and also go for maintenance. So that is in brief the life cycle stages for an interactive system.

(Refer Slide Time: 07:34)



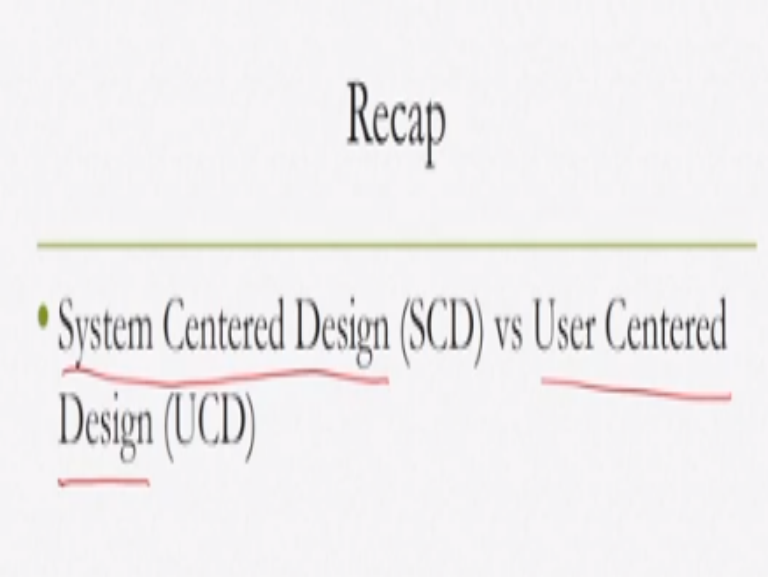
Now in our discussion on these stages we will skip the first stage that is feasibility study we will not spend much time. We will assume that whatever we discuss is on the basis of the assumption that the further development is feasible already that is found out to be feasible. In light of that we will start our discussion so first thing that we will discuss is the first stage after feasibility study that is requirement gathering, analysis and specification.

(Refer Slide Time: 08:20)



So essentially we are starting our discussion on the life cycle stages with the second stage here. Now note here that the term requirement actually is referring to usability requirements rather than system level requirements we shall see the difference in subsequent discussion.

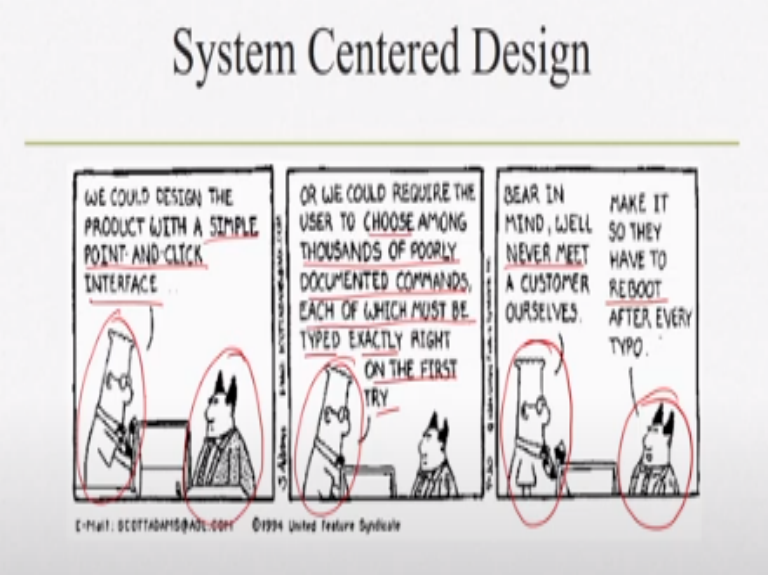
(Refer Slide Time: 08:39)



So once we try to understand the requirements for usability we have to keep in mind that here we are talking of requirement gatherings from the point of view of user centered design or user centric design. So the other approaches where user centricness is not so prominent we can refer to it as system center design. So we throughout this course we will distinguish between these 2 approaches one is system center design or the traditional design of a software and other one is user centered design or design where users concerns are explicitly taken into account in the end product.

Why because we want to have a usable system where the system will be usable for lay persons or layman users.

(Refer Slide Time: 09:50)



To highlight the difference let us see an interesting example. So probably some of you or many of you may be aware of this famous comic character which is called Dilbert. Now this comic strip is taken from one of this Dilbert's series so, here this is the Dilbert figure the character and this person is his boss. Now Dilbert is a software engineer or IT engineer and he is tasked to create a system.

Now as an engineer or developer he is asking his manager for opinion. The first way to develop as per his Dilbert's opinion is that the system could be designed with a simple point and click interface. That means Dilbert is time trying to convey the message that it should be simple to use for the end user. It is also possible as per Dilbert to create a system where the user need to choose among thousands of poorly documented commands each of which must be typed exactly right on the first try.

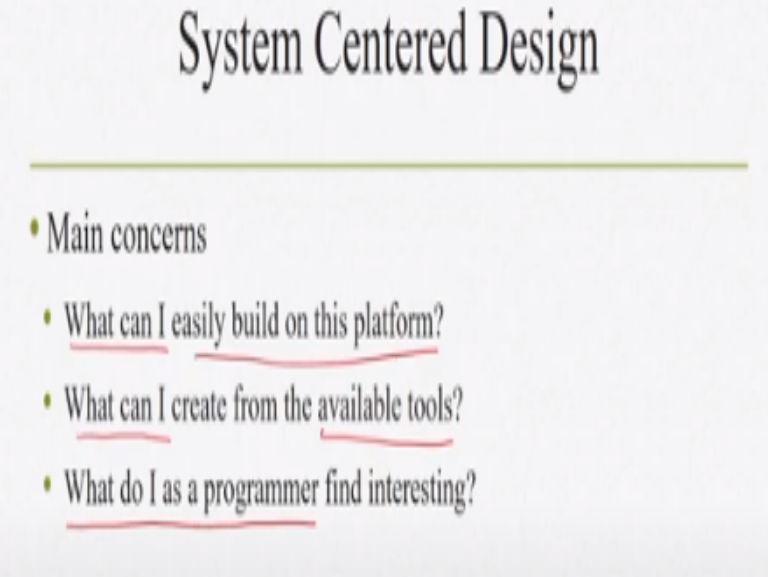
Also in such a case if the customers face some issue then as engineer or developer they will never meet a customer, ourselves to understand their needs expectations problems. And so on it appears that the manager is supportive of this idea and he also adds along with this complexity in the system that, make the system so. So that the customer have to reboot or the user have to reboot after every typographical error.

So essentially what it says is that as an engineer Dilbert finds it more interesting to build a very complex system which need not take into account the user’s needs, expectations, characteristics

because the engineer will never meet the users and the manager is supportive of that idea. So this comic strip essentially indicates that developers may be having a tendency to go for a design or development of a product in a way such that it is not correlated with the user characteristics.

Developer may like to develop as per his or her own or the teams own understanding of the user rather than actually studying the user.

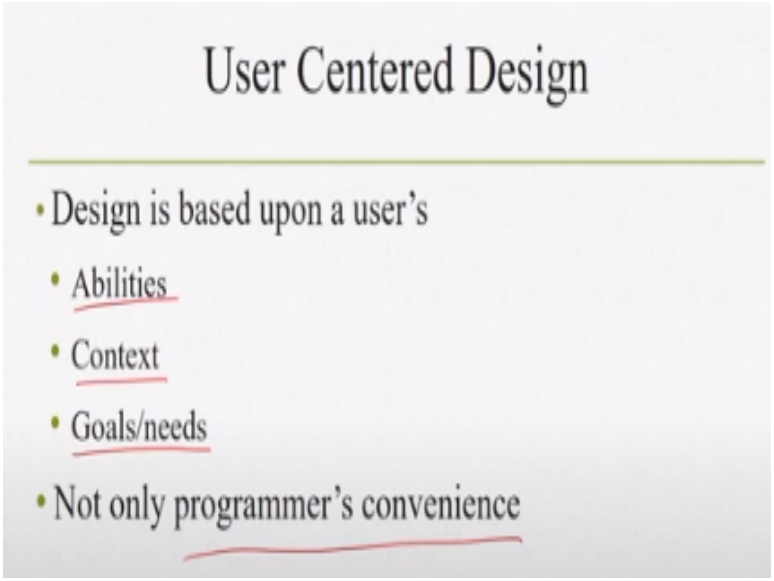
(Refer Slide Time 13:35)



If such an approach is taken which we are referring to as system center design the main concerns of the developer are what can I as a developer? Easily build on this platform rather than what is required by the user. What can I as a developer create from the available tools rather than catered to the user's needs and for that if required create new tools? What do I as a programmer find interesting rather than which where to write a program or add features will help the users?

So essentially the concern is about the developers own expertise skill and interest rather than the user's needs this is what we are referring to as system center design. Where the requirements from the user sides are not considered at all rather the requirements are framed to support the skills expertise and interest of the developer.

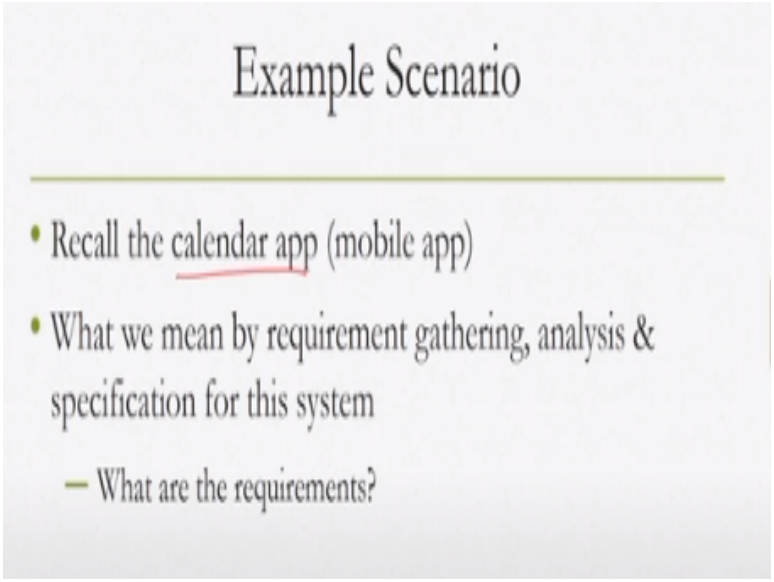
(Refer Slide Time 14:46)



In contrast to this approach we have user center design where the design is supposed to be based on a user's abilities, context, goals or needs. So these are crucial considerations here that what the user is capable of doing what are their goals or needs and what is the usage context? The crucial difference here with the system centered approach is that these considerations are not based on only the programmer's convenience rather these are based on the user's actual requirements.

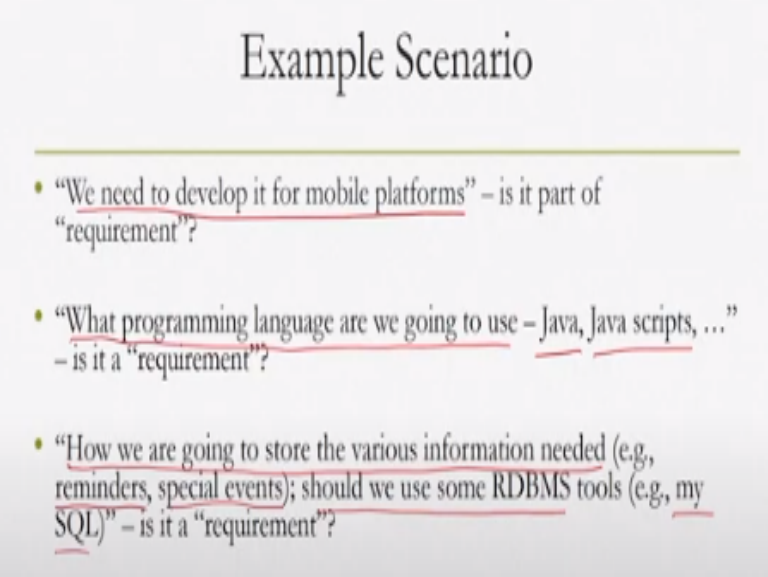
So we will clearly distinguish between these 2 approaches one is whatever programmer feels required another one is whatever are actually required by the users.

(Refer Slide Time 15:40)



To understand this further let us go through 1 example scenario recall the calendar app which we discussed in our earlier lectures. For this calendar app if you are asked to gather requirements what kind of requirements you are going to gather what will constitute requirements? So first question is what; are the requirements and how to gather that? In other words when we are saying that calendar app is an interactive system and we want to build it through stages where the first stage is requirement gathering analysis and specification. So first thing we should try to answer is what are the requirements for the calendar app?

(Refer Slide Time 16:40)



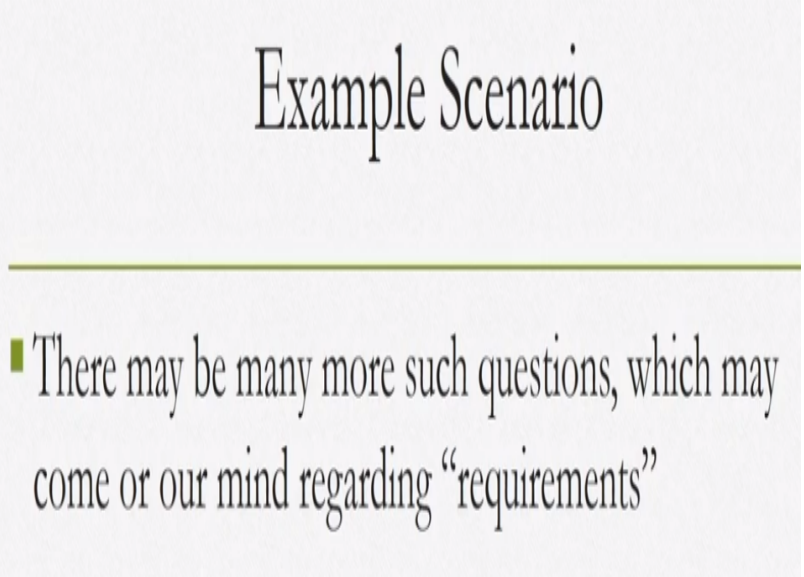
Now note that the calendar app the way we discussed is a mobile app so let us try to see what can be the requirements? When somebody asks you what are the requirements immediately few questions may come to your mind. One of those may be we need to develop it for a mobile platform or for mobile platforms. Can we call it a requirement that it needs to be developed for mobile platforms we shall see.

Another question that may come to your mind while thinking of requirements for calendar app may be what programming language are we going to use? Is it java or shall I develop with java scripts or shall I develop with something else? Can I call this a requirement for building the app? Let us consider a third question which may come to some of our minds? How we are going to store the various, information needed?

For example reminders special events so in a calendar along with the date display we may like to add or show to the user special events such as reminders for classes, quizzes, exams or special events. Now in order to do that what kind of storage technique I should use? Should; I go for RDMS tools where RDMS stands for relational database management system. So some RDMS tools should I use for example should I use my SQL which is one of the DMS tools now if this type of question appears can I call it a requirement.

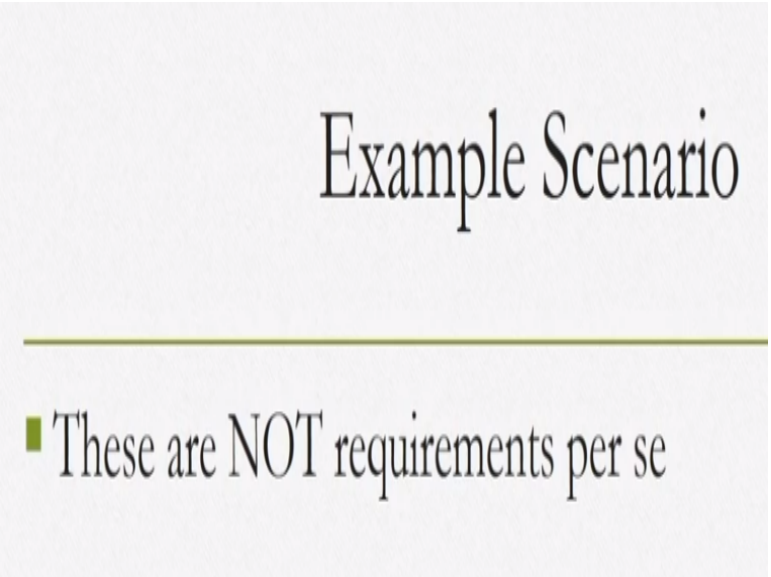
So we are seeing 3 possible queries or questions that may come to our mind when we are asked to find out requirements for developing the app. First one is related to platform we need to develop it for mobile platform should it be considered as a requirement. Second is programming language to be used to build it should it be considered as a requirement? Third is the storage technique to be used and the question is should I consider it as a requirement.

(Refer Slide Time 19:33)



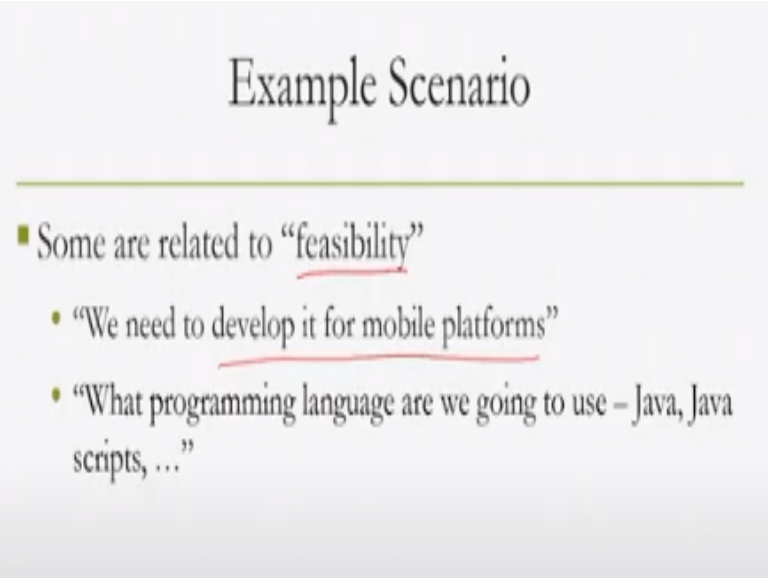
These 3 are not obviously the exhaustive list of questions that may come to our mind there may be many more. Now how do we know whether these are or these questions qualify as requirements to build the system?

(Refer Slide Time 19:53)



In fact the 3 questions that we have seen do not qualify as requirements why?

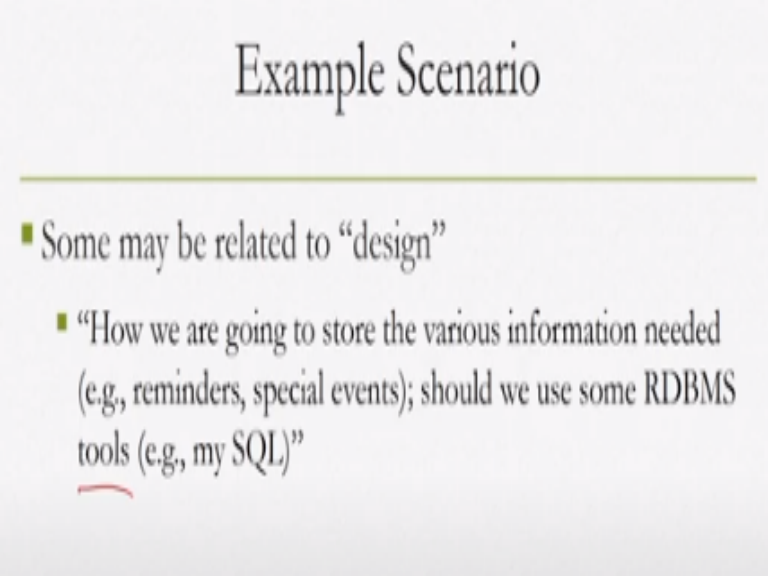
(Refer Slide Time 20:07)



Let us see among these questions some are related to feasibility of the system whether the overall project for building the app is feasible or not. For example the first 2 questions that we have mentioned are actually related to feasibility we need to develop it for mobile platforms. So that is actually a part of the feasibility study stage rather than is a usability requirement. Also what programming languages we should use that depends on the kind of manpower available and their skill.

So again that is of part of the feasibility study stage where we study about the available manpower resources etcetera. So this is again not a requirement as such when we are talking of requirement we are implicitly talking of usability requirements, usability has nothing to do with platform or programming language. So these are or these cannot be considered as usability requirements or requirements which we are trying to find out in this lecture.

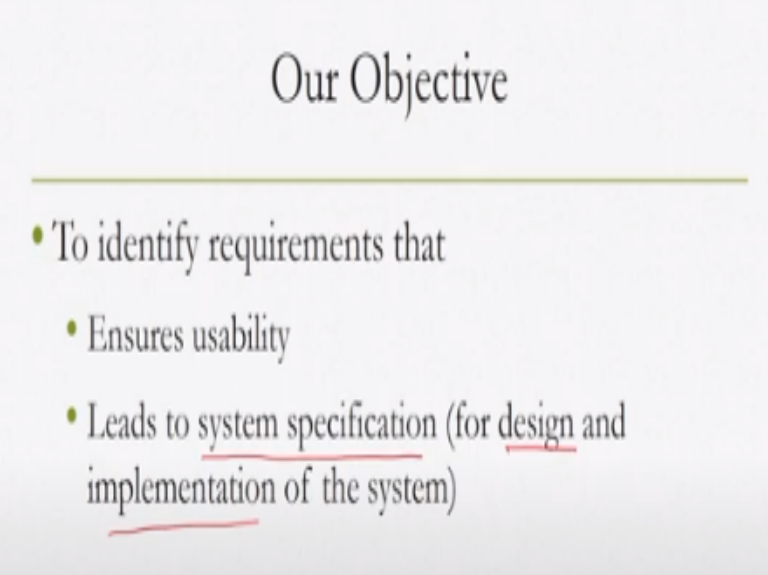
(Refer Slide Time 21:29)



There may be some queries which are more related to the design phase rather than requirement phase. For example the third question that we have discussed how are we going to store the information should we use some RDBMS tools? This actually relates to the design of the system rather than requirement for the system. So while going for identifying requirements we should be very careful whether what we are identifying are related to the feasibility of the project or design of the system or testing of the system so everything is not requirement.

Requirements are different than what we are what probably immediately comes to our mind so we should be very careful.

(Refer Slide Time 22:36)



So what is our objective then? Our objective is to identify requirements that ensure usability. That is our core objective we need to identify only those requirements that ensure usability other things that we identify are not related to requirement identification stage. Also along with identification of requirements in this stage of requirement identification or rather requirement gathering analysis and specification.

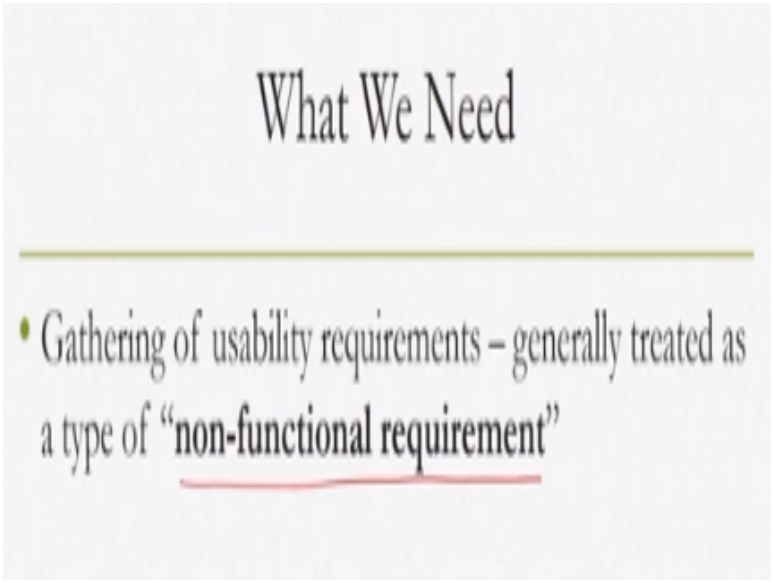
So there are 2 more components analysis and specifications so whatever usability requirements we gather or identify we need to analyze them so that we can convert them to some specification system specification. Which will make it possible to design and implement the system, so first thing is we need to identify the requirements and second thing is we need to convert it to some system specifications.

So that the next stages can be carried out namely design and implementation of the system. So there are 2 objectives one is identification one the other one is specification which involves analysis of the identified requirements.

(Refer Slide Time 23:58)

So then what we need we need to gather the requirements and we need to analyze and specify the requirements these 2 things we require.

(Refer Slide Time 24:10)



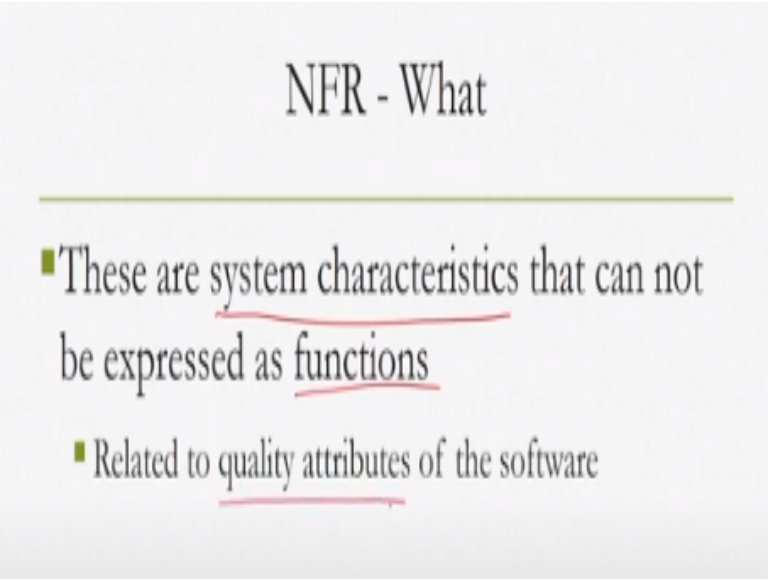
Now one thing we should note here so when we are talking of usability requirements these come under non-functional requirement. Later on we will see that there is another type of requirement when we talk of system development particularly software system development that is functional requirement. However usability requirements are typically considered to be under non-functional requirements. So essentially we are trying to gather one type of non-functional requirement namely usability requirements.

(Refer Slide Time 24:45)

So before we proceed further let us quickly learn about this concept of non-functional requirements. So when we are talking of building a system so obvious thing is to identify what we need to build that is requirement gathering. Now we have already seen that some things need not be requirements they may be part of other stages of the development. However whatever are, part of requirements can broadly be divided into 2 categories functional and non-functional.

So we will talk about first non-functional requirements because we are primarily concerned about usability requirements.

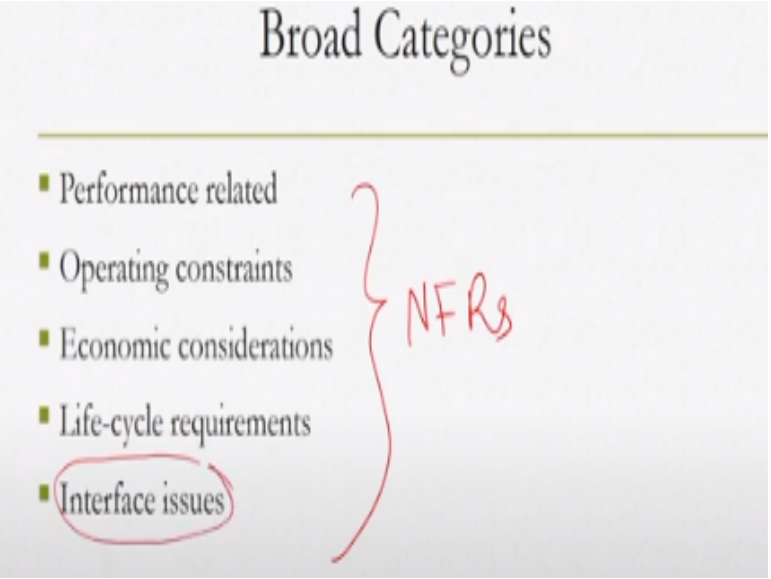
(Refer Slide Time: 25:34)



So what are non-functional requirements? These are system characteristics that cannot be expressed as functions now. When I say functions this is a loosely used term but it refers to the general concept that whenever there is a function it takes some input and produces some output. So if a requirement can be specified as a function where there is some well-defined input and well defined output then we call it as a functional requirement.

When we cannot do that then we call it as a non-functional requirement. Typically such requirements are related to the quality attributes of the system or the software.

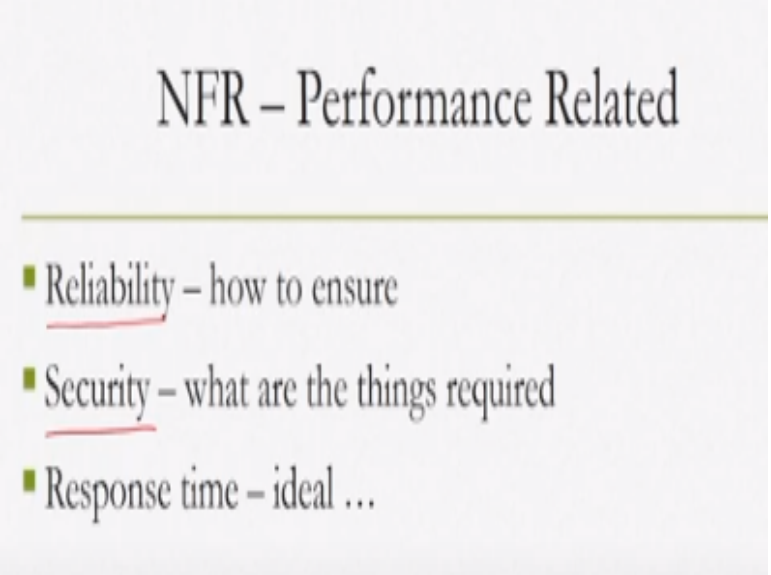
(Refer Slide Time: 26:30)



Now these non-functional requirements can be of different types so there are few broad categories of non-functional requirements. We have performance related non-functional requirements, operating constraints which are considered to be non-functional requirements. Economic considerations these are also non-functional requirements, life cycle requirements, interface issues.

So these 5 broad categories of requirements are generally considered to be non-functional requirements or NFRS. Among these our core concern is usability requirements which comes under this fifth category of interface issues but let us also try to get some idea on the other categories.

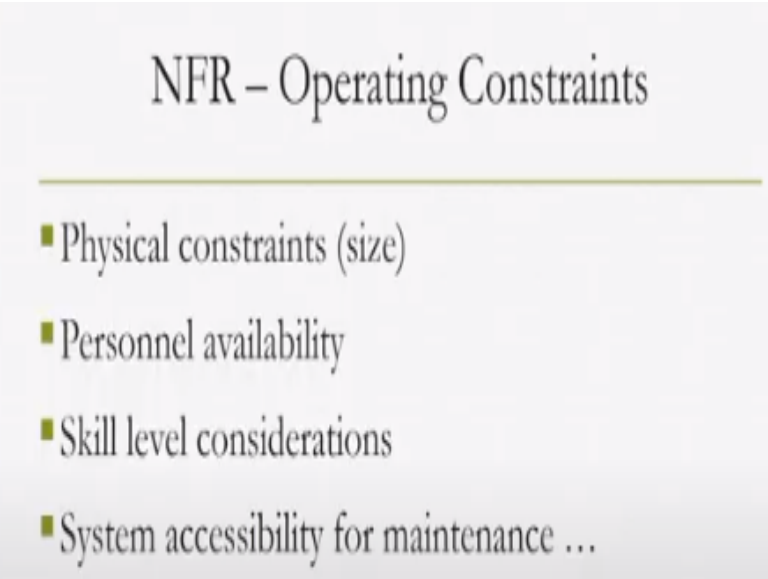
(Refer Slide Time: 27:36)



So the first one is performance related non-functional requirements there can be many such non-functional requirements. One is reliability which says that how to ensure that the product is reliable how we can ensure that. Security is another performance related non-functional requirement and which typically refers to the things that are required to make the product secure.

Then response time another performance related non-functional requirement which includes ideal response time and practical times. So, broadly these 3 can be considered to be performance related, reliability, security, response time.

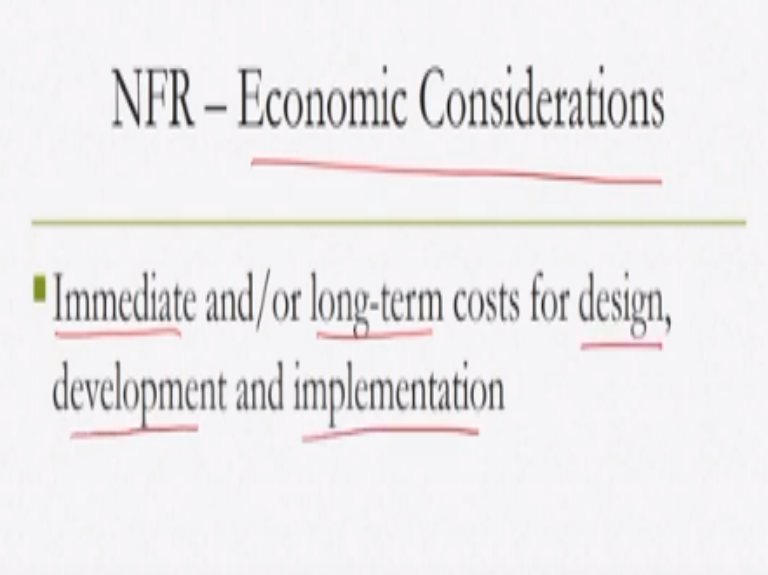
(Refer Slide Time: 28:37)



Then comes; the second category that is operating constraints so one of these is physical constraints the size of the system. Then availability of personnel to maybe maintain, develop the system, skill level considerations again to develop and maintain the system. These are few things that we need to consider while building the system also system accessibility for maintenance. So the extent of accessibility which is required to maintain the system that also comes under operating constraints.

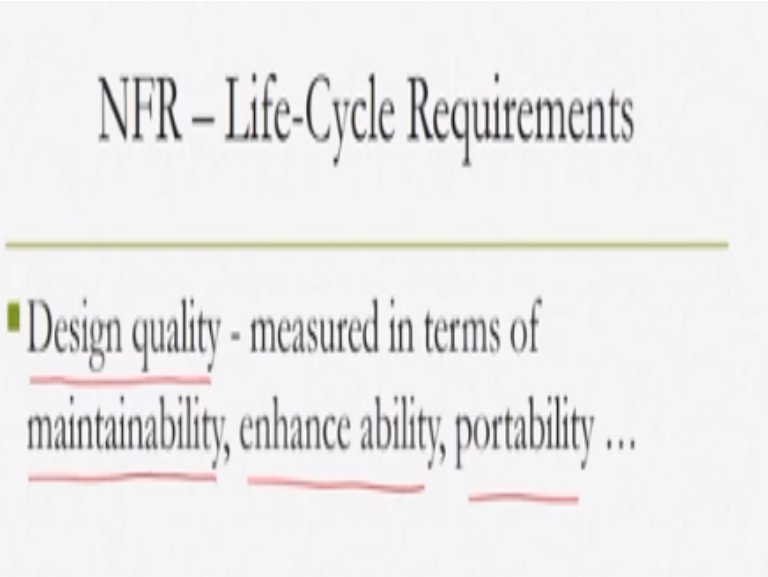
So these things actually we need to specify as requirements while delivering the product. Like the size and how to manage then available personal skill set accessibility. So, that it can be smoothly developed maintained updated and so on.

(Refer Slide Time: 29:59)



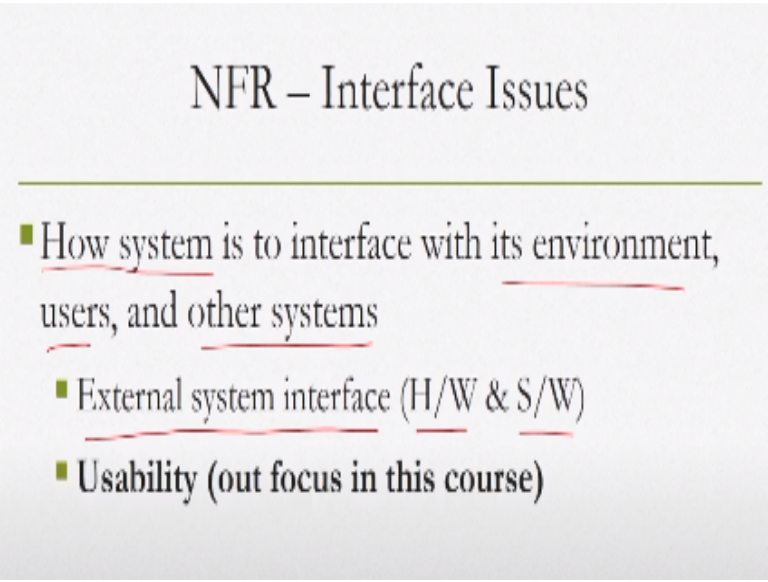
Then comes the third category economic consideration here the non-functional requirements relate to immediate and or long term costs for design development and implementation these three things. So, essentially these are also required to be specified as non-functional requirements.

(Refer Slide Time: 30:31)



Fourth category is life cycle requirements under this comes the design quality requirement for the design quality. Here by design we mean the design document that is obtained after the design stage both designed for interface as well as design of the code. So this quality of the design is measured in terms of maintainability of the design, enhance ability of the design, portability of the design and so on. So we need to specify the requirements in terms of these measures requirements for the quality of the design.

(Refer Slide Time: 31:24)



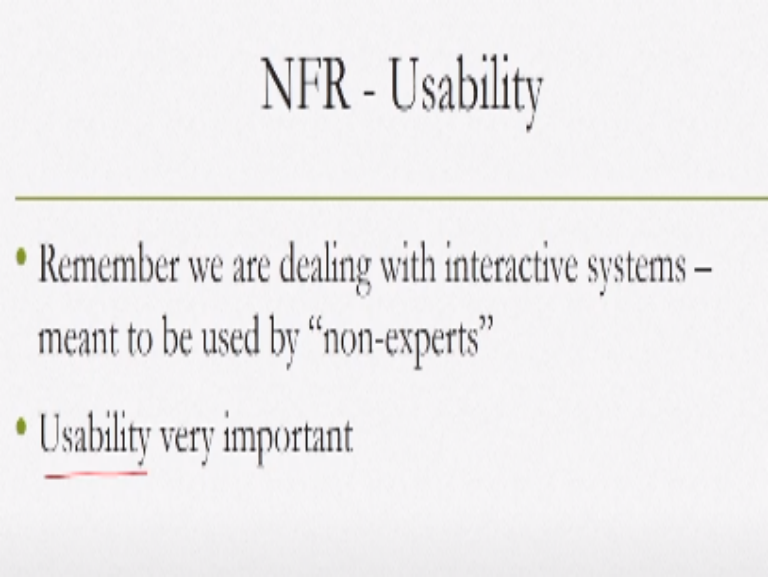
And finally comes the fifth category of non-functional requirements namely issues related to the interface here we may add. That this is not only interface so whenever we are using the term

interface we are referring to 2 conceptual things one is interface one is interaction so this is related to interfaces as well as interaction. Now this nfr deals with how system is to interface with its environment users and other systems.

So it is not only related to user interfaces but also interfacing with its environment as well as other systems. So it includes specification of this particular non-functional requirement includes external system interface in terms of hardware and software requirements or in other words peripheral devices or API's. So how to interact with the external systems and usability requirements that is what are the requirements to make the system usable.

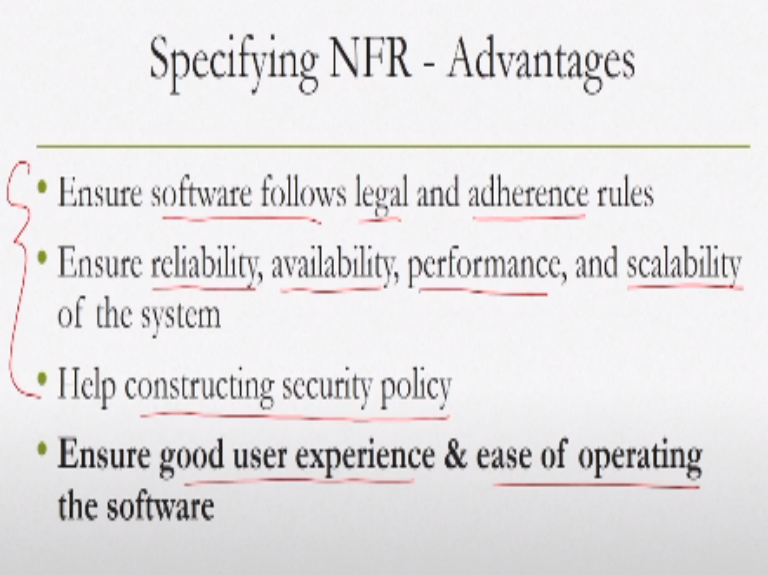
So, these 2 comes under interface issues related non-functional requirements so our focus is on the usability aspect of the interface issues.

(Refer Slide Time: 33:09)



So to repeat here we are dealing with interactive systems that are to be used by non-expert users or lay persons and for them usability is very important. So we need to ensure that the system is usable so that lay persons find it easy to use. And our whole objective for this stage of identification of requirements is centered around; identification of usability requirements.

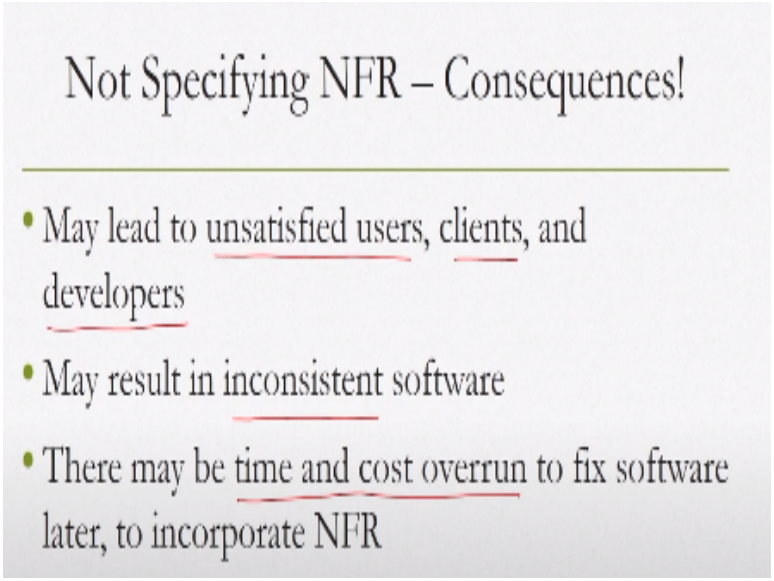
(Refer Slide Time: 33:41)



So in subsequent lectures we will see what we can do to identify usability requirements but before going to that let us first discuss what are the advantages; of specifying non-functional requirements. So if we specify this that then that ensures that software follows legal and adherence rules these are very important to maintain some standard. It also ensures a product that is reliable, available on demand; ensure performance as well as scalability of the system.

So if we specify non-functional requirements then these are ensured because we are specifying what is expected out of the product it also helps us construct security policy of the product. So these are obvious because the corresponding non-functional requirements are to be specified. Together all whatever we have mentioned ensures that at the end we get a product which achieves good user experience and ease of operating the software by the user as well as the technicians who are tasked to maintain the software. Both are ensured if we specify non-functional requirements properly those are the advantages.

(Refer Slide Time: 35:34)

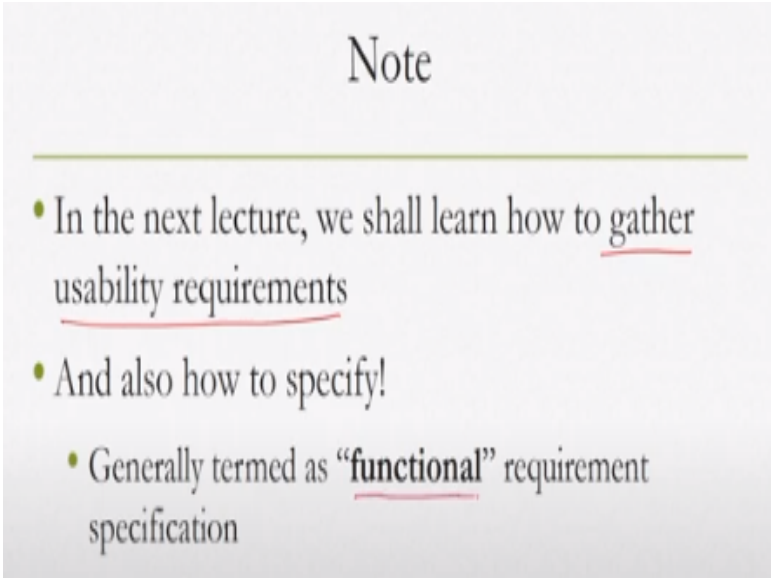


Now if we do not do that then what are the disadvantages it may lead to unsatisfied users or clients or both as well as unsatisfied developers. So developers will not be given clear idea of what they need to develop then at the end if they are told that whatever they have developed does not mean meet certain requirements then definitely that will not be very satisfying for them. So at the beginning stage all these non-functional requirements should be clearly specified as much as possible.

Not specifying the non-functional requirements properly or clearly may also lead to inconsistent software which is not a desirable outcome. Also not doing so in time may lead to time and cost overrun to fix certain issues later; which of course is not a desirable outcome because we did not specify the requirements earlier. So we need to fix it later so then that leads to time and cost overrun and it may even lead to not delivering the product at all total wastage of time and cost.

So these are significant and severe consequences; of not specifying non-functional requirements in time or before the development begins. So we got some idea of what we want to gather? We want to gather usability requirements; which is one sort of non-functional requirements. Non-functional requirements are of many types usability is only one of those but as we have seen in developing a product software product we need to specify non-functional requirements properly so that later on we do not face any issue.

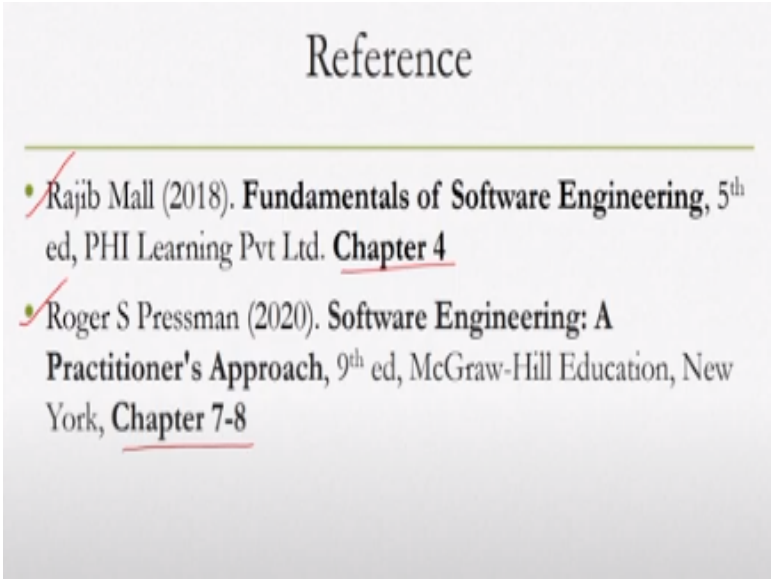
(Refer Slide Time: 38:02)



With that brief introduction to non-functional requirements we conclude this lecture and in the next lecture we shall continue discussion on how to gather usability requirements. So we will focus only on usability requirement gathering rather than gathering of other non-functional requirements. Along with that we shall also learn in details how to specify such requirements. Specify in a way so that it becomes easier to convert the specification to system design.

Now that specification is generally called as functional requirements. So we shall also learn about functional requirements in subsequent lectures.

(Refer Slide Time: 38:53)



So that is all for this lecture whatever topics we have discussed today may be found in these books. So for further information you may refer to these 2 books fundamentals of software engineering or the second book software engineering a practitioner's approach. In the first book you may refer to chapter 4 whereas in the second book you may refer to chapters 7 and 8 to get more information on these concepts.

So with that I would like to conclude this lecture and hope to see you in the next lecture soon hope you have got some idea on the concepts that we covered today thank you goodbye you.