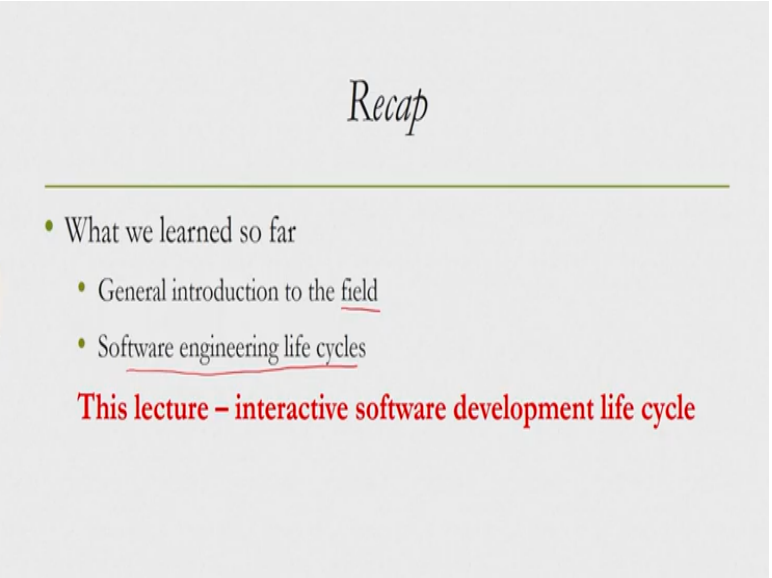
Design and Implementation of Human-Computer Interfaces
Dr. Samit Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology - Guwahati

Module No # 01
Lecture No # 04
Interactive System Life Cycle

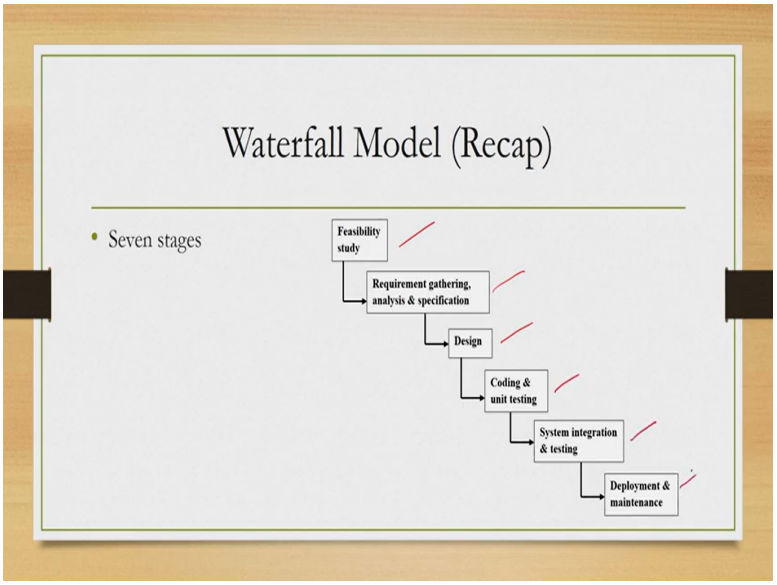
Hello and welcome to the course design and implementation of human computer interfaces. Welcome to lecture number 4, human computer interface development life cycle.

(Refer Slide Time: 00:58)



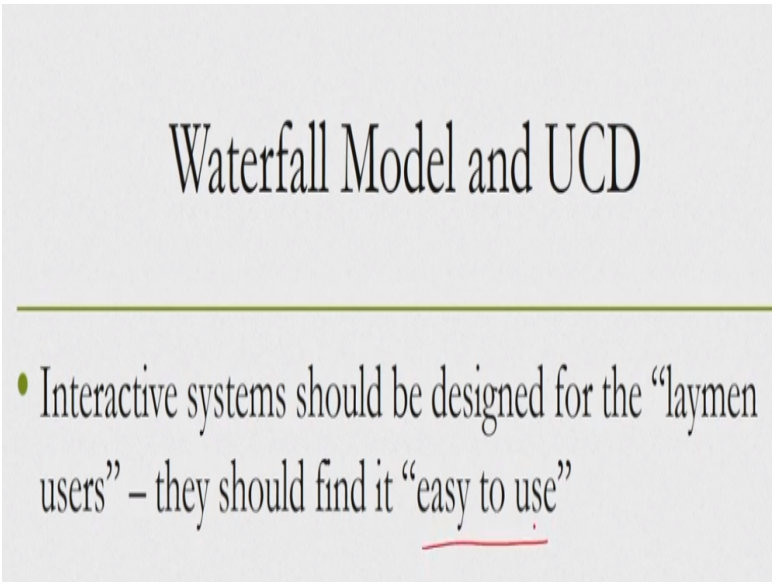
So in the previous few lectures we have learned about the field of interactive systems what are the code design concerns in the interactive software development? And also we have got some idea of software engineering life cycles or system development life cycle or software development life cycles. Now these life cycles provide us a systematic way to think about developing a software product. In this lecture we are going to learn the life cycle for interactive software development.

(Refer Slide Time: 01:40)



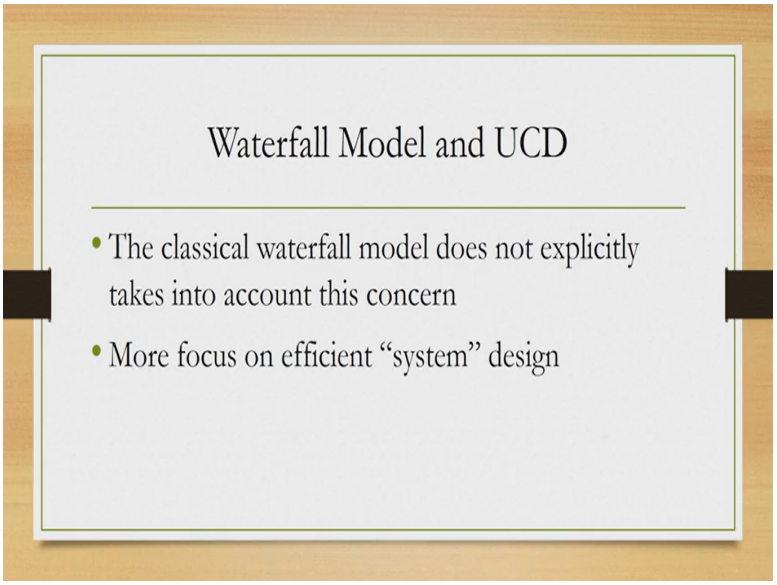
So earlier we learned about the waterfall model which is the most well-known and fundamental of all the Software development lifecycle models. It consists of 7 stages 6 or 7 stages feasibility study requirement analysis and specification, where we identify and specify the requirements design, coding and code testing system integration and testing deployment and maintenance. So with these 6 or 7 stages followed one after another, we can build and deploy any software that is the idea.

(Refer Slide Time: 02:30)



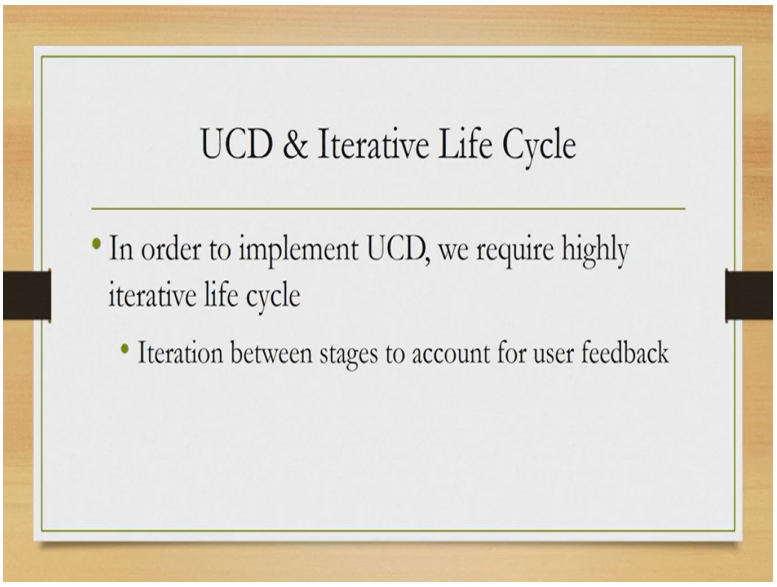
Now our concern is software to be used by layman users. As we have repeatedly emphasized this brings us to the concept of user centric or user centred design, where the software that we develop should be easy to use in other words it should be usable for the intended users.

(Refer Slide Time: 02:58)



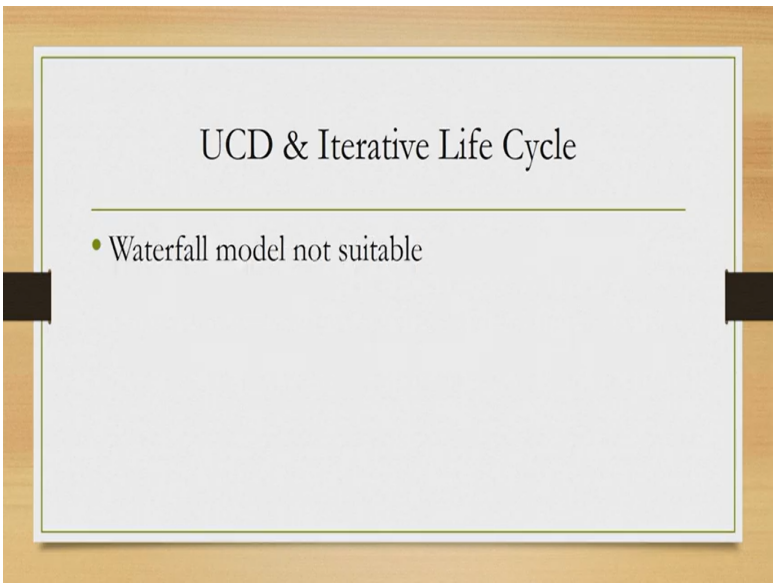
Now the classical waterfall model if we want to follow those stages to build user centric software, the model actually creates a problem. Because our core concern here is usability in order to do that, we have mentioned earlier that we need to take into account the users either in active mode or in passive mode. Now taking into account the users is not explicitly the concern of the waterfall model. Instead the model is designed to build efficient systems rather than usable system.

(Refer Slide Time: 03:53)



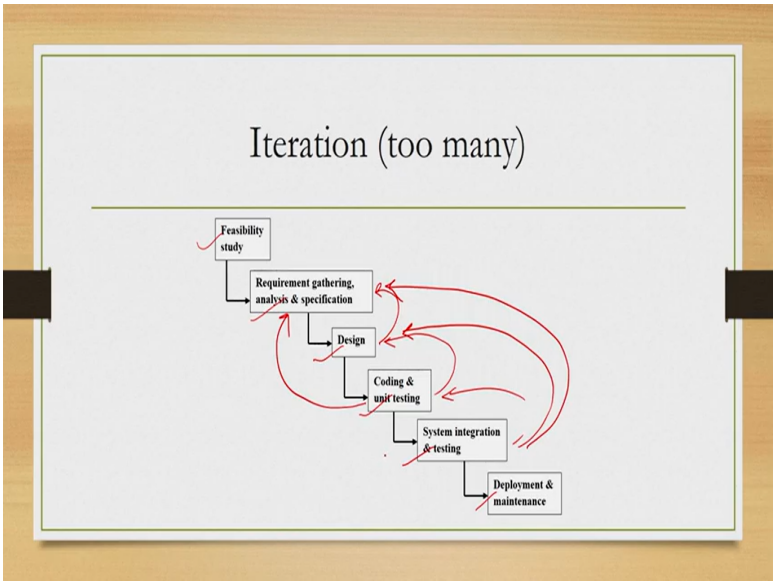
So what kind of problem it creates in order to develop user centric system. We require iterations between stages because that is required to take into account the feedbacks we receive at each stage of the design.

(Refer Slide Time: 04:20)



Now if we want to do that with the classical waterfall model then the model becomes rather messy.

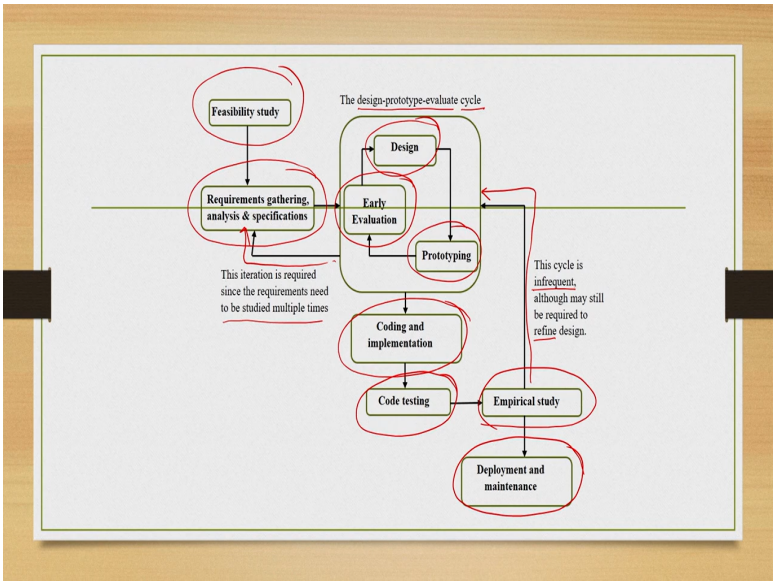
(Refer Slide Time: 04:27)



For example suppose, this is a linear flow of the process from feasibility study we go to requirement stage then design stage, then coding, stage integration, stage deployment and maintenance stage. Now in case of user centric approach we need to take into account users in every stage now suppose in design stage. We find out user input which does not match with the requirement specification we have obtained in the previous stage then we may have to go back to this stage after coding while going for testing.

We find out some issue then we may have to go back to this stage. We may also need to go back to this stage directly rather than going in the design stage after again system testing, we may need to go back to unit level or design or requirement stage. So; too much iteration which makes it difficult to visualize the overall process and creates confusion in the mind of the developer in order to handle that we can refine this life cycle to suit our purpose.

(Refer Slide Time: 05:58)



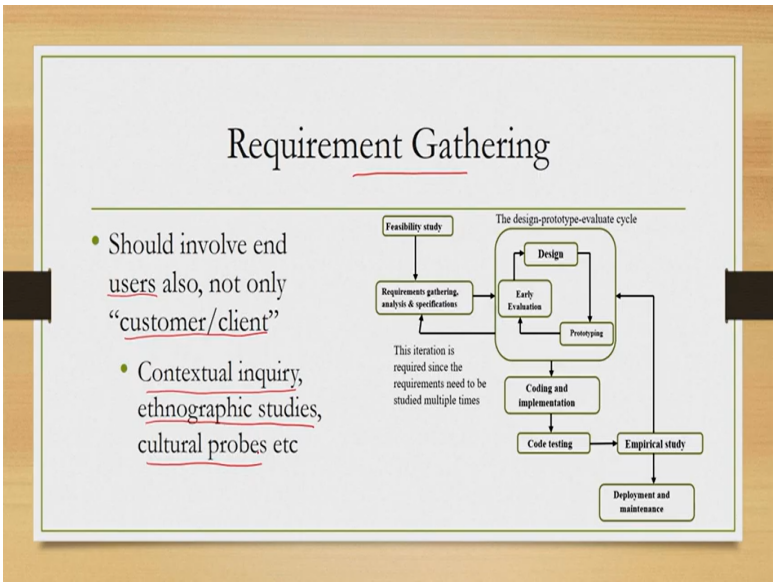
So this figure illustrates such a life cycle for implementation of user centric design approach. Like waterfall model it has several stages starting with the feasibility study. We have requirement gathering analysis and specification stage now. From this stage we go to a design prototype evaluate cycle stage. Here there are 3 sub stages design, prototype building and early evaluation these 3 are connected in an iterative cycle.

Once that cycle stabilizes that means, we do not find any more issues to be resolved based on user feedback, we can go for implementation, we come to the coding and implementation stage once that is done, and we go for code testing. Now after code is tested that means our system is executing as per our requirement. We need to go for another user testing which we are calling empirical study or empirical research.

So in this stage the system is not tested for code efficiency. Rather it is tested for usability. Based on the findings in this stage it may be possible that we go back to this cycle. Similarly based on the findings in this cycle, it may be possible that we go back to requirement stage and the last stage is deployment and maintenance like waterfall model.

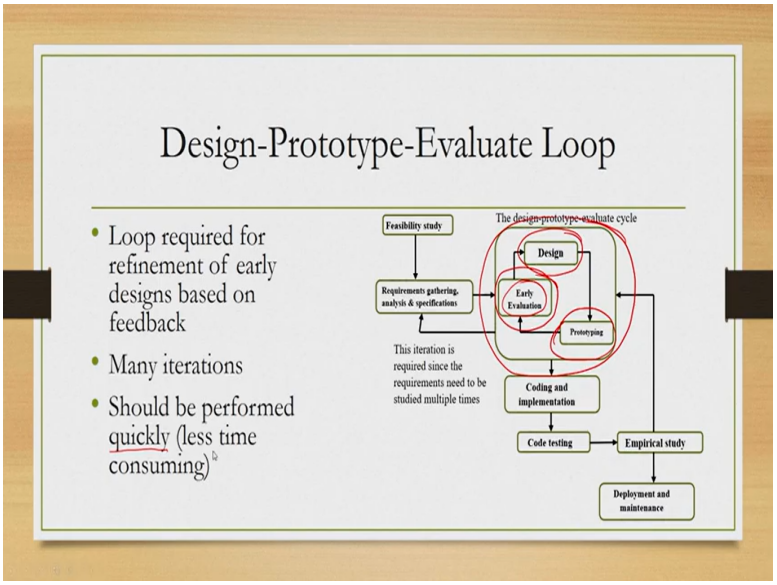
Now this design prototype evaluate cycle to requirement gathering stage this cycle may be required based on the outcome of this evaluation but still it is not very frequent it is expected to be not very frequent. Similarly from empirical study we may have to go back to design prototype evaluate cycle which again is expected to be infrequent although may still be required to refine our designs. So the needs of building an usable software may be taken care of with the help of these types of life cycle model. Let us briefly have a look at different stages of the life cycle.

(Refer Slide Time: 08:53)



Now in the requirement gathering stage since we are dealing with human computer interfaces. So here we need to identify requirements from the user's end users or layman users. Rather than only the customer or client of the product that is very important accordingly we have to deploy different techniques. Those are used to capture end user requirements such as contextual inquiry ethnographic studies, cultural probes and so on.

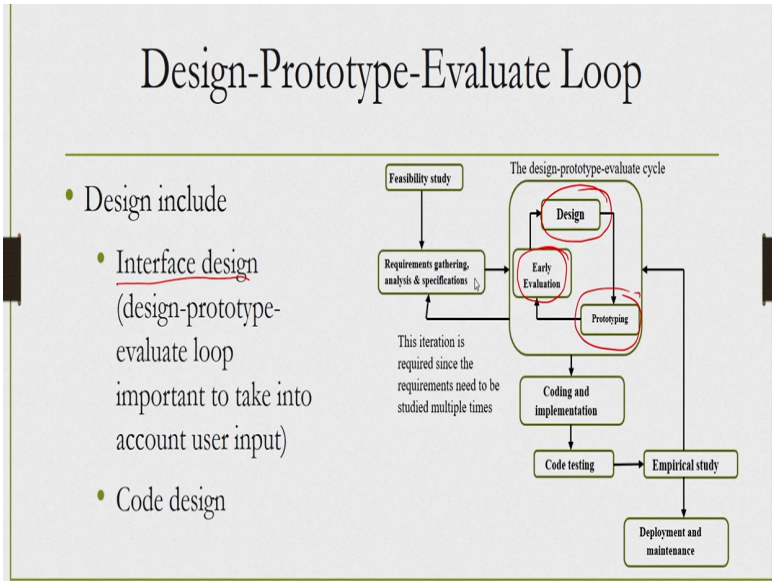
(Refer Slide Time: 09:30)



Next comes the design prototype evaluate cycle so it involves 3 sub stages design building prototype based on design and early evaluation. Now why this loop is required? It is required so first we go for a design based on the requirement. We identified now this design is prototyped and again user feedback is taken based on the feedback. It user feedback or some feedback is taken in the early evaluation stage.

Based on the feedback we refine the design again go for prototype building. And so on till we find out that no further changes in the design is required. That means the design has stabilized so at that point we break the cycle and go to the next stage of coding and implementation. So in this cycle it is expected that there will be much iteration accordingly; our early evaluation and prototyping should be simpler to carry out. So those should be quickly achievable. So that less time is required to build a prototype or to evaluate a prototype.

(Refer Slide Time: 11:01)

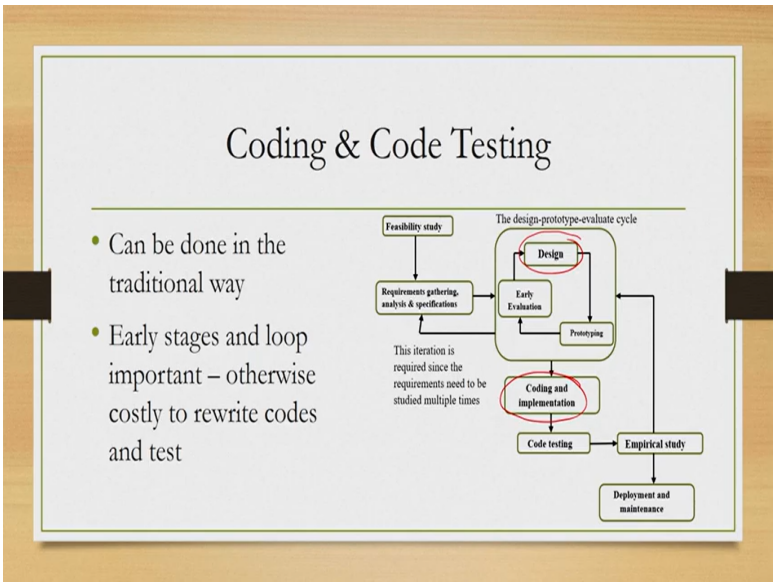


Now one thing we should note here, when we are talking of design, 2 concepts are involved. One is design of the interface that is design of the interface and interaction from the point of view of the end user. In that design nothing related to implementation or coding is involved. And when we are talking of building prototype and quickly evaluating, we are actually referring to the design of the interface and interaction.

So we first design the interface and our interaction to build a prototype get those prototypes evaluated quickly refine our design and so on till we find out no more refinement is required in the design of the interface. And interaction once that design is stabilized then we designs the code or how to implement the system. We go for that design so that is also implicitly involved in this cycle. Although that design of code is also iterative but that iteration does not require end user feedback instead those iterations are performed within the design team.

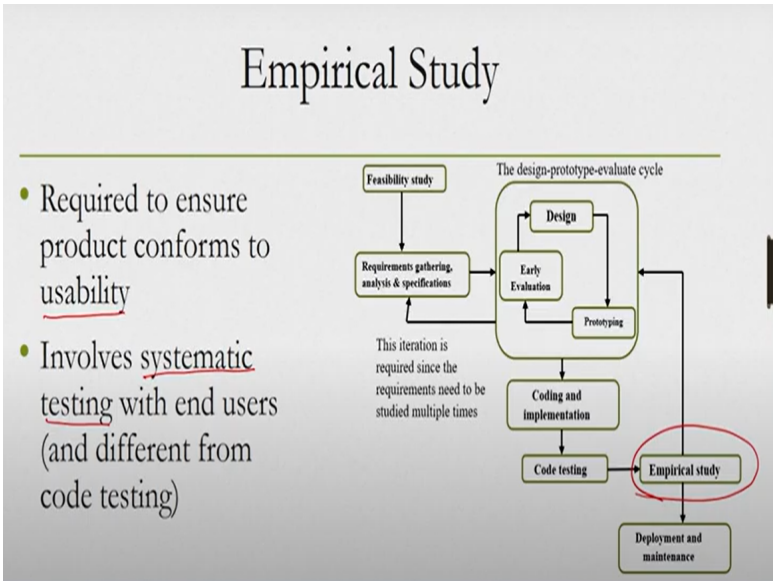
And their prototype is not necessary so essentially that iteration involves designing the code and evaluation by team other team members or brainstorming. To refine the design conceptually 2 different cycles are being mentioned. In this stage one is interface design and the cycle associated with it. The other one is code design and the cycle or iteration associated with it. So the entire cycle of design prototype early evaluation is important for interface design. To take into account user input whereas for code design prototyping is not required and evaluation is basically brainstorming within design team.

(Refer Slide Time: 13:25)



Then comes implementation, so now we are talking of implementing the code design that we have achieved in the earlier stage. Now implementation can be done in traditional way that is by writing code and testing the code or debugging the code. So here one thing to be kept in mind is that before we go for implementation we try to get as good a design as possible. So that we do not need to modify the design later then it will involve changing the code also which is technically costly in terms of manpower.

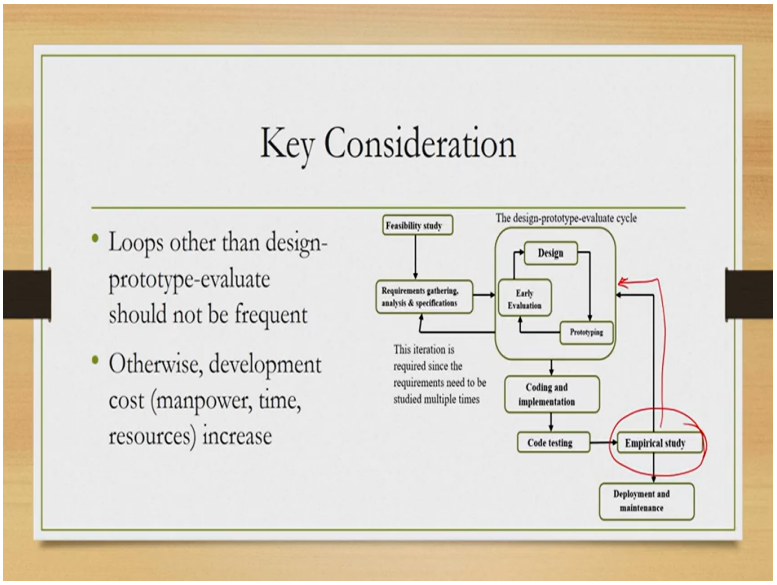
(Refer Slide Time: 14:10)



So along with code we need to test for the code so there are standard techniques available the other important stage is empirical study. So once the system is implemented and code is tested that means the system gets executed without any flaw. We still need to know whether this overall product is usable or not code testing will not give us that knowledge instead what we need to do is to get it tested with the end users again.

And that requires a systematic and scientific approach which is generally called empirical research or empirical study. So this is required to ensure that the product conforms to the usability requirements. And this is not an ad hoc process it requires systematic testing approaches. But these testing approaches are totally different from what we have employed for prototype testing or code testing.

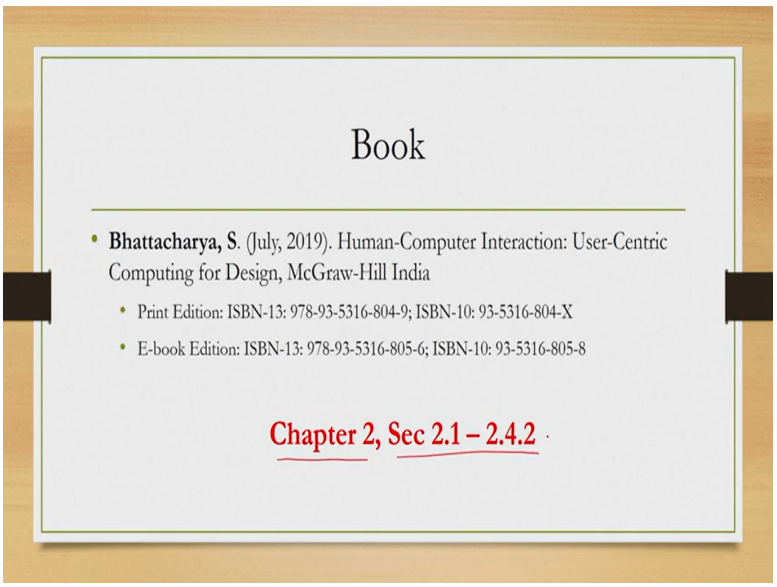
(Refer Slide Time: 15:23)



Again the earlier stages are very important because empirical study is also a costly affair .In terms of time resource manpower cost. So we should ensure that before going for empirical study we should ensure that the implemented product is as good as possible. There should not be too many problems otherwise; we have to repeat this loop multiple times it can be repeated once or twice. That is still acceptable but if we have to do it multiple times because the earlier stages we did not do properly then that increases the cost as well as turnaround time of the overall system.

In fact it may lead to non-delivery of product also so to summarize, we have discussed a software development lifecycle that is tailor-made for user centric design implementation. It involves important cycles these cycles are important to take into account user feedback and ensure that the product is usable. In subsequent lectures we will go through the details each of these stages.

(Refer Slide Time: 16:52)



Whatever I have discussed in this lecture can be found in this book. You are requested to refer to chapter 2, section 2.1 to 2.4.2. So that is all for this topic looking forward to meet you in the next lecture thank you and goodbye.