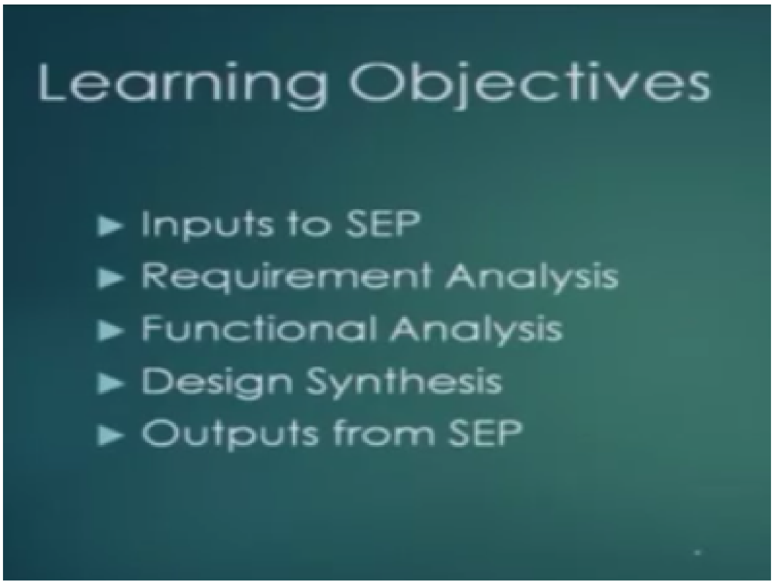
Systems Engineering

Prof. Deepu Philip
Department of Industrial & Management Engineering
Indian Institute of Technology – Kanpur

Lecture - 06
Overview of Systems Engineering Process

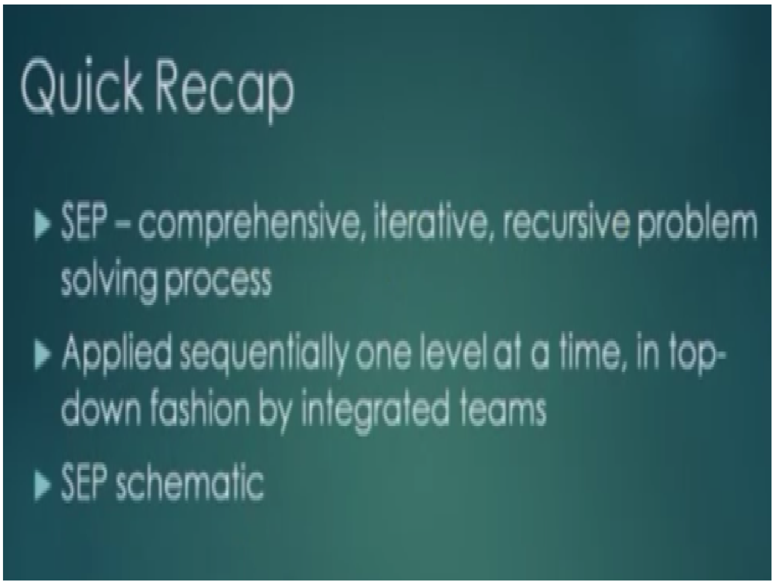
Good Morning, I am Dr. Deepu Philip, I am from IIT Kanpur, and today we are going to discuss in detail into systems engineering process. So we would do a complete overview of what is system engineering process is about.

(Refer Slide Time: 00:28)



The learning objectives for today includes, understanding what are the inputs to SEP, SEP stands for systems engineering process. So, in this lecture whenever you see SEP, we are talking about systems engineering process. When we talk about SE, it is systems engineering. So, we will discuss in details about requirement analysis, functional analysis, design synthesis and outputs from SEP, systems engineering process again.

(Refer Slide Time: 01:05)



So, a quick recap of what we know already about what systems engineering process is. We seen this briefly in earlier lectures, as SEP as a comprehensive, iterative and the recursive problem solving process. Comprehensive means it is all encompassing, it looks at the problem in a comprehensive manner. Iterative, we already studied what iteration is and recursion is, if you remember the yesterday's lecture, we have discussed about it.

Iteration is about the same function being applied to, the same process is being applied to the function again and again to solve a problem that is identified. Whereas recursion is, we are basically applying the same function but to the lower level of the products. So you are keep on drilling down further, drilling down, drilling upwards to realise something, to address an issue.

So, it is a problem solving process, it is iterative and recursive. We already saw these definitions yesterday. And we also studied that applied sequentially one level at a time. So, as we said earlier, systems engineering to a large extend starts with what you call as the customer requirements at one level, and from there you go down to the next level, from where you will identify the functional requirements.

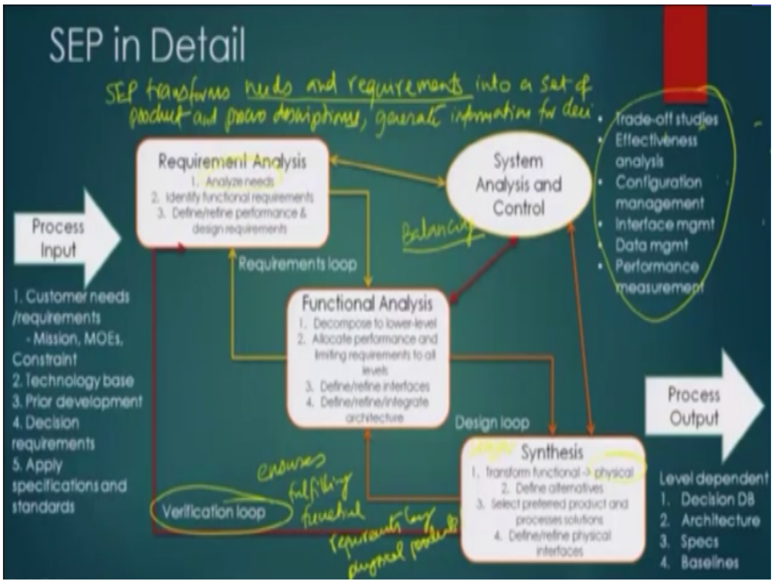
And from there you will do the modules, from modules you will do the sub-systems, like this you will keep on drilling down. But, if you are applying systems engineering to the customer requirements, you apply to the customer requirements throughout. Once it is done, it will give you the output to the next level, which is the functional requirements. And then you apply systems engineering process here, so here is one application of SEP.

Once it is done you derive this, then you apply SEP again you derive the next one. So it is applied sequentially, one level at a time, you go, move from one level to the next level, you do not apply randomly. And it is a top down fashion, so we discussed yesterday, the top down fashion is for basically converting the user requirements to a detailed documentation to the lowest level of a product or a sub-product or a sub-item that can be manufactured or produced or bought or reused.

So, we bring it down to that level of smaller level of items, from there you integrate them back. So, it is a top-down to get this level of sub-items, and then is bottom-up to get to assemble it back to the full product. We saw this yesterday, then SEP schematic we have study, seen the basic schematic of three circles integrating like this, and then we call this point as systems engineering process.

We had some requirement analysis, systems engineering, then life cycle. We have seen this diagram earlier also, we are going to look at it into it in a much better fashion.

(Refer Slide Time: 04:15)



So, in detail the systems engineering process, there is an input to the process, then there is a requirement analysis which is first step that we do, followed by a functional analysis, then a synthesis. We have seen this earlier, which has a systems analysis and control which is also what we call as the balancing act. And once you complete this you actually get an output, in which there is three loops, requirements loop, design loop and a verification loop.

So, if we discuss here in a simple sense, systems engineering process, SEP, transforms needs and requirements into a set of system product and process descriptions, and you generate, along with that you generate information for decision makers, also you do provide input for the next level of development. So, the systems engineering process transforms the needs and requirements, these needs and requirements are specified by the user or the customer into a set of system product and process descriptions.

So, generate those descriptions, it also provides information for the decision makers, so that they can decide, which alternative to chose from, and also provides input for the next level of development, so from one level you move to the next one. So, in the requirement analysis, you see that it analyse the needs, the needs of the user, identify the functional requirements, and define and refine performance and design requirements.

So, the needs and requirements needs to be translated into something called as a performance and design requirements. So, an example, I will suggest you is that, if you talk to the defence people about developing a unmanned aerial vehicle, depending upon whom you talk to, they will have different needs. So, let us say, for example, you are talking to various para-military organisations.

Then they will say that, we need a UAV that need to be fit into the back of a jeep or a gypsy, or like a vehicle that they use for their transportation from place A to place B. The requirement is a need, like we need the UAV to be transported, so the need is, transportable in the tail gate of jeep/gypsy, gypsy I am talking about the Maruti gypsy, fair enough. How do you translate this into what you call as a performance and design requirements?

So the transportable UAV that means it has to be, it means it has to be dismantelable. So, the need is translated to a dismantelable plus quick assembly UAV, is one example. The fit in a tail gate of a jeep or a gypsy is like, it defines the length of each individual part. So, we will say, okay, each part cannot be longer than 1.1 meter, something like this. So then this gives you the design requirements.

The earlier part gives you the need requirements. So you do the performance and design requirements based on the need and the requirements specified by the customer. The customer might specify the needs and requirements in a different fashion in a very loose terms. But the systems engineering is responsible for translating them into, what we call as exact numbers, which can be used by people for developing the system.

So once this is done, so when we discuss these needs, another stuff let us talk about in what forms the input comes from, to the process. The input comes into the process in the form of multiple ways. It is sometimes people's mission, like it has a fight day and night, it has to look at the surveillance, it should identify the people, it should identify vehicles, it should identify big structures, all these kind of things the people, the customer do specify in this.

There is MOE's, MOE's stands for measure of effectiveness. So sometimes people would say that it should be able to fly for twelve hours non-stop. So then you see whether the endurance is available for twelve hours, which is another measure of effectiveness. Sometimes people will say constraints that means, it needs to be silent. So that means you cannot really use gasoline anymore, because gasoline engines make noise.

So then you probably have to go for electric propulsion for the UAV to make it silent. So sometimes customer specifies constraints also, so that we do not want anybody to see it, and we do not want anybody to hear it. So when these types of loose stuff that you, statements that you needs to translate into design constraints, from which the final product specification gets derived out.

And you chose appropriate system to meet the need. Sometimes somebody might say that I want to do something, but there might not be the technology available to do it. A classic example of this is like somebody may say that I want to go to Mars and come back in a day, not possible, because that probably the technology is not available at this point.

So one thing the systems engineer should also be available about, known about is, what is a technology that is available presently, and can the needs of the customer be met by the available technological base within the country or within the organisation or within the area. However, you look at it, the question is obviously, is the technology base available?

Then you might have done similar development earlier, and from which somebody comes up with some requirement, then you will be okay, that we did this to some extent we did this in some project and these are the issues we came up with. So, we know, if we go through this route, then you might end up with much more bigger problems. And so this prior development knowledge also comes into help, while getting the process inputs from the customer, also the decision requirements.

So the customer says, okay I want this. So when do we make the decision, or what are the parameters against which we make the decision. That is also something that needed to be specified out. So the customer says, you make unmanned aerial vehicle that fly for twelve hours, and it can see both day and night, and it is silent and fine that is a very loose statement.

You make a unmanned vehicle that meets the needs, and the customer comes back and say well, this is kind of fine but I want to see much more sharper images in the night. And then you will be like, then you get into this debate. So before the finalisation comes in you like, okay, if you see that image, in which you can identify it is a human being, cannot identify who is that person, but that is a human being, male or a female at least to some extent, you can identify up to that much.

If that is acceptable, then we consider this project as successful. So, that type of decision requirements are also specified as part of this exercise, or at the time of eliciting the customer inputs. And sometimes you will be forced to apply specifications and standards, like sometimes the customers says, okay, I want to paint it with this particular paint, and you are like it is not allowed in the country.

Because it is a lead-based paint and that is against the standard, that is established by the environmental standards, that is established by the country. So hence let us do something else. So that could be possible, or we want to make a paint like this and want to do it, or we know how to make this paint, we will make it happen and will be painted by this particular paint.

So in this is the first step in which a systems engineer talks to a customer, gets the customer needs in different forms. In this conversation, you use all these knowledge to know and other things to come to a reasonable level of document, which the customer and the systems engineer is capable of agreeing to and say okay fine, this sounds like a reasonably good product to develop.

Then from there the needs are analysed, that is where the requirement analysis comes into picture. And from there functional requirements are elicited out. So as I said earlier, one of the functional requirements that, in the UAV, the para-military specified to us in one of the project that we worked before, was that it should take off from a paved and unpaved surfaces. So unpaved surfaces is quiet broad, it can be a field in which you have wheat or paddy being growing, it could be a river bed, it could be a rocky terrain.

So we were like, can we be more clear. So paved means it could be any runway, public road, a straight stretch from where you have sufficient length for the air craft to run up and take off. And that is a clear stuff about the row. And unpaved what they said was, okay, place where there is no as flat or black top, but it is flat and rolled, and hard surface. Once that was specified out.

Then you could come up with, okay, fine this is what the landing gear should do, this much big tyres be required, all those of aspects where taken care of from those things. So their functional requirement of taking off from paved and unpaved surface, gets translated to the type of wheels are need to be used in the UAV, type of landing gear that need to be chosen, the ground clearance that need to be put.

And all those are the aspects and the damping on the suspensions, all those things has to be decided as part of that. Once, this is done the other aspect that also came out for the UAV was like, the take off should be very short, should be within hundred meters. So now that translates to some power requirements of the engine, so how much power the engine should be able to give at full RPM, and what should be the weight.

So the power to weight ratio was one of the things that needs to be specified out of this particular customer need of hundred meters of take-off. Then once this requirement analysis is completed, you move to the next level, which is called as the functional analysis. As I said earlier, you decompose it into a lower level, so usually the customer specifies higher level functions.

The customer will say, land on a paved and unpaved runway, he or she will never say this should be thickness of the landing gear, this is the particular wheel that you need to be used, this is how much the capability of the shock absorber of the landing gear should be. That specifications is never given by the customer, it is something that is generated by the systems engineer for the product, the person who designs the product.

So the higher level functions are usually specified by which we call as HL functions, which are defined by the customer. It needs to be drilled down into a lower level functions, you decompose them into much lower level functions. And once you decompose them into the lower level functions, you have to ensure that these lower level functions are meeting the need, or the functionality. The functionality is not lost in this process.

So that is where this requirement loops comes into picture. When you start decomposing, you keep on rechecking with the needs, by decomposing this into such a lower level function, are we going to lose any needs out of this. And if the answer is no, then you go ahead and do it, that is the idea. Then you allocate performance and limiting requirements to all levels. The more levels, the sub-levels you keep on going, you allocate performance and limiting requirements.

You say this is what it is supposed to do, and it is not supposed to exceed this, or it is not supposed to perform below this. So those kind of limiting requirements, upper and lower limits are established to each level as you drill down. And sometimes you might come across three or four sub-systems, for a simple small system, and they all need to be integrated. So each sub-system will interface with the other sub-system.

So you have to define the interfaces, and you also how to refine those interfaces. How would those interfaces will work in order to deliver the required function. And you also need to sometime define and refine the architecture, like when we were doing the unmanned aerial vehicle, one of the smaller versions, one of the major requirement that came out of it was to ensure that it fits in the back of a jeep or a gypsy, as I mentioned earlier.

So earlier first we started with designing something up to 1.5 meters in each individual length, then pretty soon found that 1.5 meters is not good, because after loading the UAV into the tail gate of a vehicle, you do not have any space. So to allow for people also to sit along with it, you have to further reduce the size. So each size, each individual part, the maximum length of each individual part was upper limited to 1.1 meters, to provide additional space also.

So this is where you have to refine the architecture. So the product design remains the same, but it was cut down into much more smaller parts for easy assembly. It did increase the assembly time obviously, because there are more parts involved. But, yes, it was a compromise that was to be made and the customer agreed to it. So all these things when you are doing the functional analysis and you are looking at the needs.

You require the input from the customer and go from there. So this requirement loop is an iterative process, that makes it happen. Then once you have the functional analysis done, from there you actually go into, what you call as synthesis, or in another way to think about this synthesis is the design synthesis, okay, you are synthesising the design.

So here you are transforming the functional aspects or the functional requirements that came up from the functional analysis to what you call as a physical system, physical system, or a physical product, or a sub-assembly. So this physical thing can be a product or a sub-system or assembly anything like that, okay. So you identify the physical thing that will actually satisfy the functional requirement, that is part of this analysis.

Sometimes you might want to say, okay, if this does not work, then what is the other option, alternate is. You have to define alternatives. So the customer might say that, okay, we want a UAV that is silent, so, okay fine, let us go with the electric propulsion. Now what type of electric propulsion, DC, AC, brushless DC or high torque motor, there are quite a lot of options that is available.

So we will decide, we decided, okay fine, we will go with brushless DC motors of this particular calibre as the first choice. But as a backup we will go with this one, and this one, power ratings, and if this does not work, then we will work with the third right type of a motor, so we have alternatives. And you need to have them, because sometimes the physical product might not deliver the thing that you expect.

So you need have an alternative at that time, you do not have time to look for alternatives at that time point, you already have to plan for that. Once these alternatives are there, and once you start building the product, when you are synthesising the design, pretty soon you know certain things, okay, this is, this seems like a doable thing, your past experience and other things comes into picture.

So the preferred product and process solution gets chosen out, or kind of emerges out of what is being defined out, with what out of synthesised design. Once it emerges out, then what you do is, you also define the physical interfaces. So you would probably say that the auto pilot or the control the UAV, auto pilot should interface with the electric motor, in which the output is fast the motor should spin.

So the auto pilot is a digital system, which would actually give the input to the motor controller, and the motor controller will determine how to change the speed, RPM of the motor. So those interfaces, the interface between the auto pilot to the motor controller and the motor controller to the motor, all those things need to be talked through and defined through.

And once that is done, the other part that we also need to look at is, what you call as, you erase this point, because it is kind of not clear. So the system analysis and control, we called this earlier presentations as balancing, this is the balancing or the controlling aspect of the whole system. You can keep on doing the requirement analysis, functional analysis, because you need to, some point of time, get out of the loops and move to the, move forward.

So the system analysis and the control allows you to do that. There is bunch of tools that are used by systems analysis and control of the balancing people. That is involves trade of studies, you decide, what trade of, which one to be picked, you might compromise on something, you cannot have everything. So what to be compromised on, that need to be done.

Effectiveness analysis, which is like which one do you chose, and how effective that it is, that also need to be analysed out. Configuration management, interface management, data management, performance measurement, all these aspects that you see here, these are the tools that are used by the system analysis and control team, or the people to balance the product of the system development endeavour.

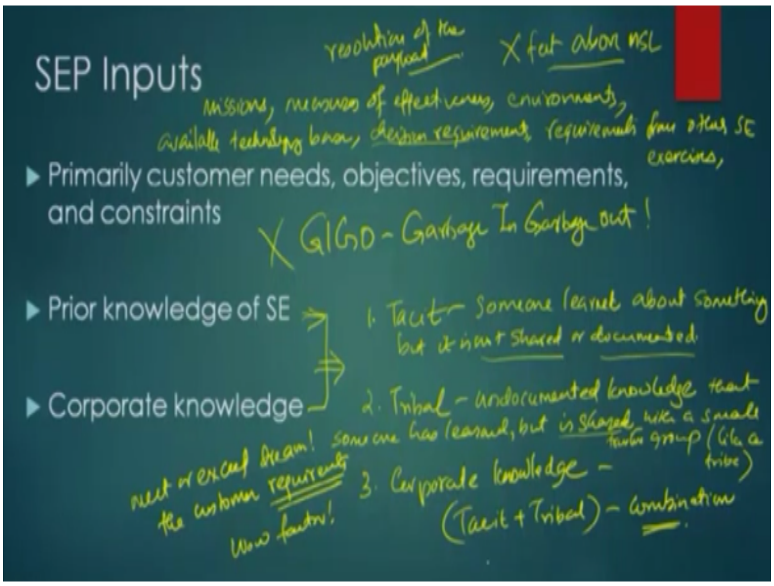
To ensure that the system, the development process moves beyond the loops and you actually do get a process output out of it. And the process outputs are, they are dependent upon each individuals as I told you, is a systems engineering is applied to each individual level and you move in a hierarchical fashion drop down from one level to another.

So depending upon which level you are, your output, process output might be different. It could be a decision database at certain levels, it could be the architecture of the product, it could be the set of specs, which could be used to specify this is the product and you can decide it on to further trade it off and other stuff, and it could also be thought about it as a baseline or like, this is the baseline of the product beyond which we will not be willing to compromise because then you will not be able to satisfy the customer needs.

So once you, we have seen an overview of this, in the mean time one loop that we have not spent too much time on this verification loop. The reason for this is, because we thought it is going to come much later down the whole lecture. But when you do the design synthesis, lot of the time when you define and refine physical interfaces or when you choose a physical product, you need to verify that the choice meets the needs.

Because all the time, you have to ensure that you are fulfilling the functional requirements. So this loop ensures fulfilling functional requirements by physical products, the choices you make, some of them you might manufacture, some of them you will buy. But this loop is also necessary to ensure that the physical products are chosen meets the customer needs or the customer functional requirements that are identified in the requirement analysis.

(Refer Slide Time: 25:31)



So moving on, we will little bit discuss more in detail about the systems engineering process inputs. As I said earlier, the inputs can include a wide variety of things, and some of them is includes mission, it could be measures of effectiveness, it could be environments and operation. This could be operation environment, this could be some other environmental criteria.

For example, if the when the UAV development happened, the customer came back and said you should be able to operate in high altitude area. High altitude area is a very loose statement, but that is an operational environment requirement. So you need to now define what is a high altitude area, what height above mean sea level. So in this case then finally we ended up deciding X feet above the MSL, it should be able to take off.

So that was decided by the discussions with the customer. So the operating environments do sometimes translate into certain performance requirements of the product. It could also be on the available technology base, I already mentioned that, available technology base. It can also be into the decision requirements, because sometimes one of the decision requirements that was mentioned was the field commander should be able to look into the image that is, or the video feed that comes back from the UAV, and decide whether the target is hostile or not.

Well that is the decision requirement, the decision requirement for a person who is using the product. Now that translates, directly translates to resolution of the pay load, or the camera, that is onboard on the UAV. So this is how you derive things out of it. And also sometimes, you might also come across stuff like that.

You might say that, okay, requirements from other systems engineering exercises, one of the things that we have learnt from other exercises in the similar UAV field was that, the number of hours after which the engine need to be over hauled. So we had that number with us earlier, so we said, okay, this is what a frequency of overhaul if we use this particular engine, because we have already used it in some other product, are you okay with that or do we need to go for some other stuff.

And when the customer said, yes sir, we think we are fine with the number. Then, that actually helped us to decide the power plan, which power plan to actually go with their engine. I told you about, the decision requirements already, and there is something else also which is part of this. Sometimes you also would understand the fact that, the prior knowledge of systems engineering.

It results in three, the prior knowledge actually requires which we also call as corporate knowledge, results in three things. I would call this knowledges are one thing called as a tacit knowledge, the second one called as tribal knowledge, and third one is what I call as corporate knowledge. So the tacit knowledge is, it is a, it is defined as someone learnt about something, but it is not shared or documented.

So the knowledge that is learnt by someone about something, but it is not shared or documented, it is not at all shared or documented. So tacit is usually with individual person. So if you go to a factory and you see a machine, and you talk to the guy who is operating the machine. He will obviously have quite a lot of tacit knowledge.

Because he has been operating this machine on a daily basis, and he understands what he is, what some other things that need to be done on how can the performance of the machine be better, that is usually called tacit knowledge. But it is never written anywhere in the manual or any stuff, and sometimes the supervisor might not even know about it. So that type of a knowledge is called as a tacit knowledge.

The tribal knowledge on the other hand is also an undocumented knowledge, for sure, it is an undocumented knowledge that someone has learnt, but is shared with a small group, let us call as small trusted group, okay, or like a tribe. So this is called as tribal knowledge, there is no documentation to this knowledge, it has been there, somebody learnt about it, but the person has shared, it is shared, with a small trusted group of people, like a tribe, or a group or a click, they know about it.

That is usually true, because if that is a machining that is people who are working in a manufacturing cell. The people in the cell, there might be four or five workers there. They would have knowledge about each and every machine, so you can call those four five peoples as a tribe. And they know what to do and what not to do.

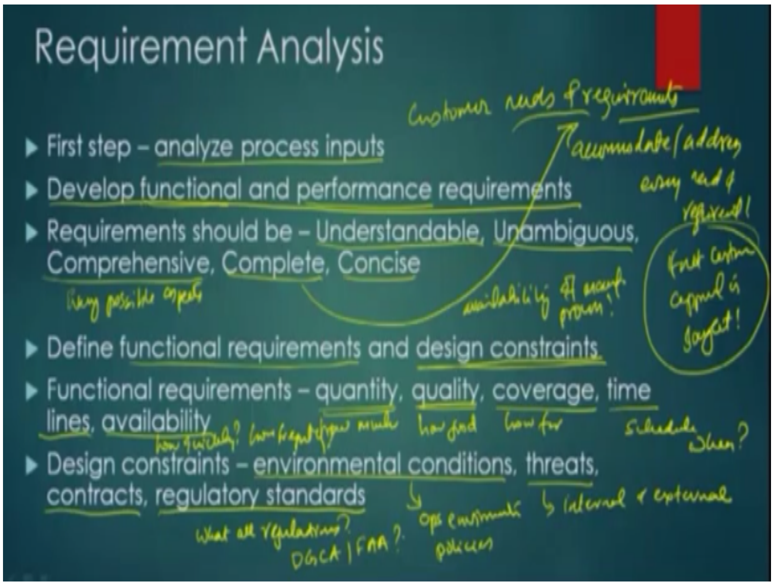
So the corporate knowledge is pretty much tacit plus tribal, and this knowledge is the, is the combination of both, okay. And systems engineer need to be good at understanding or getting to this corporate knowledge, he should know about tacit and tribal knowledge that is happening in the organisation, or across the area where the product or system is supposed to be developed.

But once you have that, you could actually do quite a lot more things, and you can actually develop a product which actually would meet or exceed the customer requirements. Instead of this requirement, I am kind of going to use the word dream. The customer might have dreamt about something, you might be able to exceed the dream, which would be actually some the management fact actually call this as a wow factor, okay.

So the inputs as you know, the famous statement called GIGO, Garbage In Garbage Out, so the systems engineer needs to be ensure that the GIGO does not happen in this process, use whatever information is available, whatever the customer specifies, all those things need to be taken into consideration even including the tacit and tribal knowledge and include the corporate knowledge, whatever you can and come up with a really good requirements document, from where a good system development can translate.

This is probably the first and foremost the most important activity a systems engineer would actually do.

(Refer Slide Time: 33:35)



Then we move into the next one what we call as the requirement analysis, and it is the, in a simple way the requirement analysis, its aim is to develop the functional and performance requirements. So once you have specified the customer has given you the needs and the requirements, then your first job is to analyse these, you know, inputs, which are the customer needs and requirements, as we saw earlier.

Once you analyse that from there, you develop functional and performance requirements. Functional requirements means, you are capable of doing something, can you decipher a person in the night, that is a functional requirement. If you are using a UAV and you have a camera on it and it can decipher a human being from an altitude of 2000 feet, standing in the ground, that is a functional requirement.

The performance requirement would be the resolution, it can be basically like, how many times you can do it, can you do it in all the weather, or do you require a clear sky, can you see through these clouds, that would also be part of the performance requirements. So both functional and performance requirements together define the system.

And the requirements, most important things of the characteristics, whether it is functional or performance, the requirement should be, that should be understandable, understandable means people to be able to read and understand okay so this is what is being meant. Unambiguous, there should not be any confusion, are we talking about. So somebody might say that we want a camera that can see in the night.

And if you leave it that way, then obviously the ambiguity is that are you using an illumination source. If you say that, we need to have a camera that would sense the heat in the night and using that thermal imaging happens, then people, okay fine, they are talking about the infra red cameras. So it should be unambiguous, there should not be no confusion about what you are trying to achieve.

It should also be comprehensive, comprehensive because all aspects everything, that needs to be considered for the product development should be considered as part of this. Also what we call as comprehensive review, every aspects, every possible aspects, should be considered in this. It should be complete, because once it is done none of the needs and requirements should be left out, completeness means accommodate or address every need and requirement.

If we cannot, then we should get approval from the customers saying that this cannot be met, okay let us remove this, okay. So if not, the customer approval is sought. If you cannot fulfil that, only the customer says, okay fine, I do not want it, okay, I am okay with to compromise on that. This also is part of the process.

And concise, it should be in such a way that people would be read it and understand. People should not come to a level where say that, okay fine, it is there, but I am not going to read this, because it such a huge document. So it should be concise, it should be enough for somebody to read it and understand and then say, okay fine, I can read this document.

Also please understand the fact that when you define the functional requirements, it also to some extent the design constraints also need to be specified. Sometimes there might be certain aspects that you will constraint the product that you are going to design like, the availability of the manufacturing process. Sometimes might say, okay, this is the max I could do, then those constraints also be specified as part of the requirement analysis.

And the functional requirements, if you want to spell out what are those functional requirements, if you want to look into detail what are the functional requirements. Some of them is quality, sorry, quantity is how much, quality is how good, coverage is how far, and think about in that way, timelines that is schedule or the question is when, and the availability, how quickly, and how frequently, that is also fine, okay.

All these questions is functional requirements, once you answer these things, once you address these aspects of the functional requirement, you will come across, you will have a good functional requirement set up for a particular product or system. The design constraints are typically derived out of the environmental conditions, threats, threats can be both internal and external, the environmental condition.

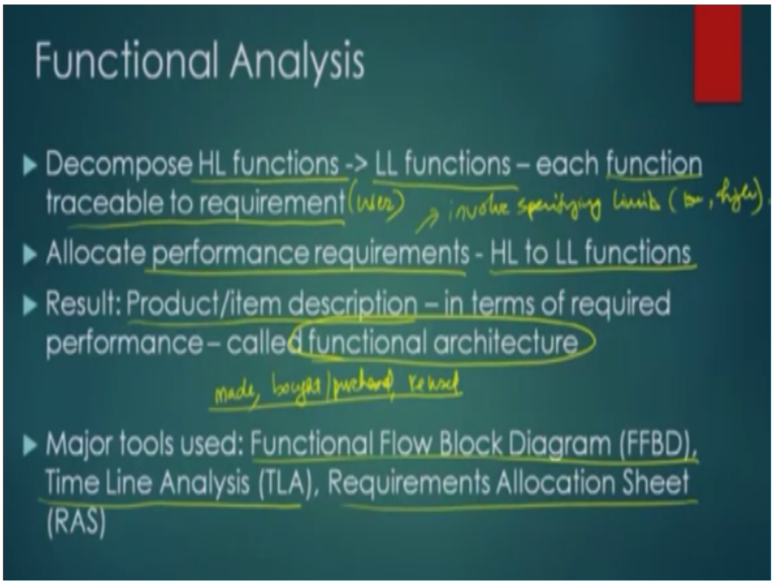
As I said earlier, can be the operating environments, it could be policies, many things like this, okay. Threats can be internal and external, because there is like, for example, when we were developing the Tejas aircraft, one the problem we faced through, faced was not being able to develop the engine, and the important restrictions on certain engines delay the process.

If you, I would recommend you guys to read that story, how the non-availability of a certain engine delayed the indigenous development of the Tejas aircraft, which is kind of like a, to an extended threat, it is a external threat. Contracts, how to define make the contracts, is sufficient people available to make the contracts, what should be written in the contract, what should not be written in the contract, that is all there.

And then also there is called as a regulatory standards, like what all regulations, like for example, when you do a unmanned aerial vehicle, UAV design, then you need to look into what is the, in India it is a DGCA or FAA regulation standards, what are they. And then you go follow that, and agree that, and design a system in such a way that fulfil those standards as well. So that when somebody is going to take the UAV and fly it at some place.

The regulatory agency does not come back and say, not acceptable or this does not meet the standards and he cannot fly. So I hope you guys understand the requirement analysis to a complete extent.

(Refer Slide Time: 40:20)



Once it is done, the second one is the functional analysis as I said earlier and the user specifies what we call as HL functions or high level functions and is a job in the functional analysis is to decompose these high level functions in to what we call as a low level functions or LL functions and only criteria is each function, each function should be traceable to a particular requirement. Whose requirement? The user requirement.

The requirement is that of the user. So the user requirement should be, so each function that you put the low level functions that you derive from the high level functions, each function should be traceable to a requirement. In a way, we are saying that we are adding this function to ensure that the user, you are adding this low level function to ensure that the higher level function is specified by the user is fulfilled. So that is the part of this.

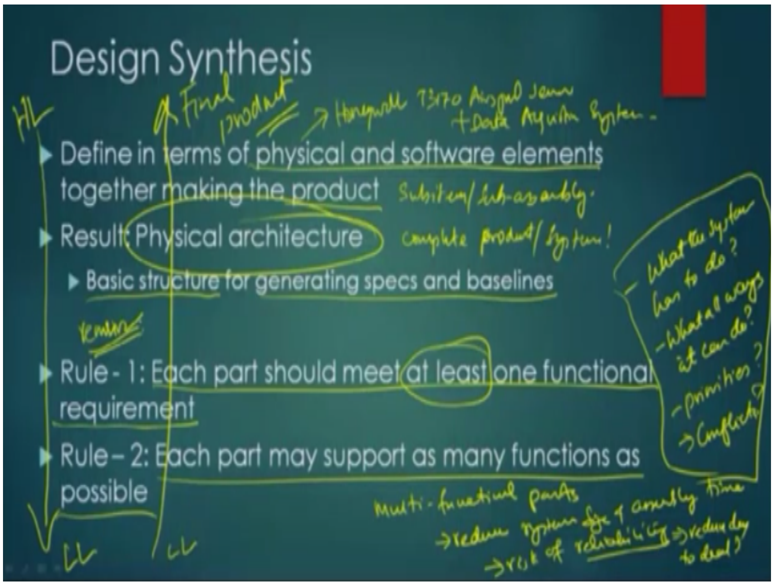
Also you need to allocate performance requirements, so once you allocate the function, how would you measure that the function is fulfilled, so that that requires performance requirement to high level and low level functions. So as you drill down the level from high level to low level, each level of functions, you will keep on adding performance requirements and as well as restrictions also that it cannot go upper and lower limits.

So the performance requirements also involve specifying lower limits, low and high. So once you go through all these things when you do the breaking down of the functions and allocate the performance requirements to the functions, what is the result, what you get out of it, the end result? The end result is a product/item description. You will finally bring down it to product/item description in terms of the required performance.

So this product or an item that is something that can be manufactured or made or it can be bought or purchased, or it could be reused. So you bring it down to the lowest level of the product/item description, which will satisfy the performance requirements associated with the function. This is called as the functional architecture of the system. The major tools used in functional architecture, which we will see in the later stages of this course is the functional flow block diagram, which we call as FFBD.

So how do the flow block diagrams are used to do the functional architecture. TLA is the time line analysis and RAS is requirement allocation sheet. So these are the major tools that are used by the people who are in the functional analysis to come up with the functional architecture of the product or a system.

(Refer Slide Time: 43:07)



Then we talk about the third major aspect of the diagram, the design synthesis where we do is, you define in terms of physical and software elements that together make the product. Here, you are basically specifying what are the physical and software elements. What parts actually do make? So here you are like, saying that I am going to use this particular sensor, then you would say, if you are buying the sensor, you will say the manufacturer.

For example, we will say Honeywell T3170 air speed sensor something like this. So you would define it in terms of the physical and you might require a software that will actually give the data to a data capturing device. So you specify all of these plus data acquisition system, something like this, okay. So this together will make one part of the product. It does not have to be the hard product, a sub-product, a sub-item or a sub-assembly.

So in this, we had a functional architecture earlier. The result of the design synthesis is we have the physical architecture. The physical architecture is defined as the basic structure that is necessary that is needed for generating specs in the baselines. So you have the physical architecture of the complete product or the system. During the design synthesis, we follow a kind of two rules, which we do not violate as much as possible.

So each part should meet at least one functional requirement. This is the rule number #1. So every part that you decide should meet at least one functional requirement. You should not add parts for sake of adding the parts. It should be there to ensure that at least one functional requirement. This is the important word at least. You can exceed more than one, but it should be there. It should not be there if it does not have any function in it.

Rule #2 is each part may support as many functions as possible. You can support more functions or if you have multifunctional parts, then you might be able to reduce system size and assembly time. However, you might stand the risk of reliability if the multifunctional part fails, then multiple functions of product might not work. So then, this is where you do the trade of, okay. Do I use this part or do I provide redundancy to deal with it.

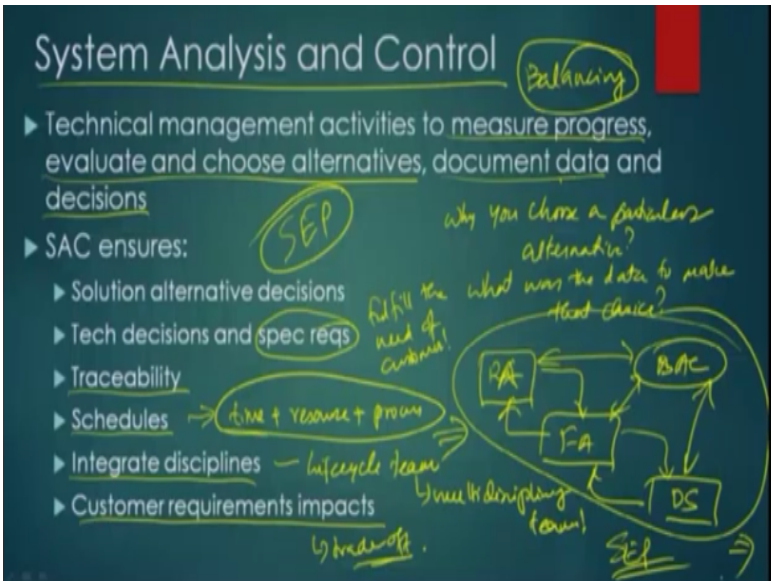
So maybe you provide two of those parts, so if one fails, there is somebody else to take over. So this redundancy analysis also comes out as part of this kind of a trade of multifunction parts and function analysis what we do. So in a way, this whole analysis helps us to understand or better understand, first one is what the system has to do. This is the first thing that we understand better and the second thing that we do is what all ways it can do, the system can do.

What it has to do, how many ways or what are all ways it can do, what it has to do, and we will also talk about the priorities or we understand the priority system and what are the conflicts. What are the issues that we might have with the system. These many things are understood clearly by the time when we are done with this complete exercise.

Once the systems engineering process is over, you have clarity on these activities and you have developed a physical architecture, you have a functional architecture. So you have gone down, drilled down from the highest level all the way down to the lowest level possible. So you are started at the high level, you went to the low level and you have the product definitions and you have the physical and software elements defined out.

Now, you go back from here to realize the product. You started at the lower level and you reached the final product. We have already seen that in the previous lecture how do we go down, drill down and drill up, so this drilling down and drilling up is also why we call it as cursive approach, okay. That is also there. We have already seen that in the previous lectures.

(Refer Slide Time: 48:00)



So the one part that we did not really spend too much of time on is, the systems analysis and control or what we call as the balancing bubble in the diagram, okay, the balancing group. So what is the job of the balancing stuff. It is actually the technical management or in a way in the previous diagram, the modified systems engineering, you might have seen these three stuff where we had a technical management group here with subgroups underneath this and we drew diagrams like this.

Well, this is more what the systems analysis on control. This the equivalent of that SAC in the modern, in both cases, the systems analysis on control. It is the technical management activities. You are managing the technology in this case and the aim of this is to measure the progress, so how good are we or how much progress we are making, so this is kind of something related to the project management, evaluate and choose alternatives.

The design document might have specified many alternatives. You might have to decide, which alternative to choose and you have to evaluate each alternative and then come up with the choice, which is a decision making process. You have to document data and you have to document decisions. So you have to document why you choose a particular alternative. You have to document what was the data to make the choice.

So this needs to be documented as well. So the system analysis on control, SAC ensures, it does all these, it ensures all these aspects, while you are doing these three loop things. You are doing a requirement analysis, then you have a functional analysis, then you have a design synthesis and you go through this loops, but the balancing SAC ensures that all these things are kept under control by, you come up with solutions to alternative decisions.

So what are the solutions, how did you choose the alternative decisions, technology decisions, okay at what particular point you have, there is a debate on which technology to be used, and then a decision need to be made, then SAC ensures that appropriate technical decisions end up being made. It also states, okay, what is the spec requirement.

Somebody says okay, this is what the spec is, somebody says this is what the spec is and if then the SAC ensures that finally a spec, appropriate spec is chosen, which will fulfil the need of the customer or the functional requirement of the customer and traceability that whatever is being done. There is a rhyme and a reason for why it was done, that is done, schedules which is the as we said earlier time plus resource plus process.

So that aspect is specified how are you going to develop the schedule in which the system will be developed. Also there will be multiple disciplines involved in what we talked about it as lifecycle teams, the multidisciplinary teams. How does those different disciplines will be integrated together that is also ensured by SAC and the impacts of the customer requirements on the system development, that is always kept in check with the SAC to ensure that okay.

What happen if we this also is a trade of. If one of the requirement is traded of, then another, what is the impact on the product and the SAC ensures that the customer is also kept in group in that aspect. I hope with all these things we have a much more better clarity on the systems engineering process, SEP as a whole. And we have seen the different aspects of the SEP, this whole thing is what we call as the SEP, systems engineering process with the inputs and outputs.

In the next lecture, we will actually go through different functional tools available for functional analysis, requirement analysis, how things are being down, certain flow charts, certain diagrams, and customs built systems engineering tools to really realize the development of a system. So thank you for listening and see you next time.