STD Session 11

**Karthik C:** [00:00:00] Too

much feedback. Positive feedback.

Hi. What happened to you? Why are you so dull? You're dull only, no? You are. No. Change in energy. The attach is positive, which means you're giving out energy. I am. Yeah. I can [00:01:00] bunk one class.

There any problems? Where are the TAs, bro? Such irresponsible buggers.

Is it not working? Should we go ask him if it's [00:02:00] not working? Can we Sir, should we call the technical team? He's only calling us. I know Ben is a DSF. See, I'm not addressed to the team. Irresponsible DS. She's a redhead. She's a

redhead, no? Bro you What? No. Why? Don't go to that thing. No, you're watching too much of that. I didn't watch anything. You're telling red haired beauty. All of those things. No, I said red haired. No. I said red haired. No. You are meaning that only. No, I don't know what you mean. You exactly know what I mean. I have no clue.

You're always on income, [00:03:00] you know. No, da. Priyank? Yeah? Why? He didn't come yet.

I'm recording, I'm recording, it's going on. Um, Um, Um, Um, Um, Um, Um, Um, Um,[00:04:00]

Um, Um, We wanted to hear, and we wanted to hear, uh uh, Is my voice audible enough? No sir, actually no. Better increase the, the, Mixer volume should be increased. Sir, what about specific questions? Specific volume, like the, the mind. No, not there, there. There, there, there should be some[00:05:00]

Is this better? It's better. It's hissing, right? No, sir. I mean Yeah, but it's better than that. Slightly push on.

Guys, is this better? Okay, so let me use this. Right? So, before I start with today's class, how many of you are yet to submit, uh, the previous assignments? Snack. We submitted, no? Still not everybody has submitted. There are students who have not submitted even assignment 1. Yes?

Right? What I will do, right? Understand this. Guys, if I see somebody talking, I will ask you to leave. 50 percent of the marks comes from continuous assessment, right? This you are well aware, right? So if you get 0, [00:06:00] 10 marks for participation, right? You only get 50 for the end semester, right? Most likely you will not clear the course, right?

The cut off will be something like 40 45 marks, right? Even somebody who gets D grade, that will be the typical cut off, right? It's the normal distribution. So what I'll do, if those of you who have not submitted, Right? Previous assignment. I will evaluate for, with a penalty. Right? Then it will be unfair to other students who submitted on time.

What I'll do? I'll evaluate for 5 marks, I'll evaluate it for 2. 5. If at all you have not submitted, right, I'll give you an opportunity to submit now. Got it? Right? Do submit at the earliest. Next assignment, right, I am yet to post. Today is the deadline, right, for the ISM, right. So, that you submitted by today.

There is one last assignment. I will give you time till next Friday. Right, another 7 8 days I will give you, right. With that, right, all the assignments will be complete. So, I will revise last week's class quickly before I continue today's class, right. Some of you have [00:07:00] doubts, so I will go back. Guys, I, I hear some murmuring.

I will ask you to leave if I see somebody. Right, so this will be the last class for the semester, right? This will be the last class for system syncing, right? So, some of you have severe attendance deficit, right? So though, that separately I will talk to the dean, right? Case by case basis that you will deal with, right?

As long as you have oriented all classes, you do not have to worry.

As usual, we will give you marks for participation, right? So let us start with last, last class.

Guys, why are you so late, man?

Right? So, let's quickly recap. What is this ISM? Right? Let me quickly recap. Guys, you are really getting on my nerves. Right? [00:08:00] Next person I see, I throw you out. Right? Don't get on my nerves. So, let me quickly recap the entire course. Okay? Let's quickly revisit, right? All the way from the beginning. What did we start with?

We started with, right? Let me go back, right? What is design? We said design can be a now, or it can be a work, it can be the process of design, right? You are designing a website, you are building a platform, you are building a back end support system for something, right? Whatever, right? Now, the important thing is, I told you, if you are doing new product development, which we call NPED, if you are doing new product development, you will start with the discovery phase, correct?

Where I have the luxury to go, to go actually ask, talk to my customer, empathize with them, right? Look at my demographics, understand what is their requirements. Take those requirements, right? FPS structure we already looked, right? I'll recap. How many of you have trouble with the FPS structure? Function behavior structure?

Raise your hands. Is everybody clear? No, right? [00:09:00] Somewhat, right? So, FPS I'll revisit again, right? it. So, to take from the requirements to what is called the product specification, right? From here, how many of you are clear, have trouble with this? Sir, can you repeat that? I will, I will do that man. I will revisit everything.

Okay. I will recap everything. Okay. Right. Three of us, uh, scored that enough. Right. So, to go from requirements to what is called the design description. What is this design description? Right. Let us take a simple example. Take your cell phone, if you go to Samsung's website, look at the specification, what does it say?

100 megapixel main camera, 2500 milliamp hour battery. Right? You see that, right? But, that is the end of it. Right? That's the deal. Right? But, if I go talk to a customer, customer's requirements, what will the fellow say? Give me a easy website. I'm building a website to book a plane ticket. Okay? What will you say?

User friendly. What is user friendly? What should be the size of the font? [00:10:00] What should be the color of the font? Where should the login button be placed? Right? All these questions you have to answer. Right? The detailed design you have to answer. Right? So, how do you go from the requirements of a customer which is very ambiguous needs, their needs to what is called function.

Function is there. There are multiple functions your product or your service has to do. Correct? Take any website. Right? Take google. com. What is its primary function? Right? Search engine, you say. But, who are the stakeholders? People also advertise through the platform, right? Right? So, it is providing you a free service.

Technically, it's not free. You are the product, right? So, it's selling you something. If you just Google something, maybe I want to buy a pencil. It says Amazon. You will find this pencil. Go buy it from here. Go to Flipkart, right? Through that, it's making money. So, there are multiple functions Google has to do.

It is doing tracking. It is doing a Tracking of your cookie, where you go visit you, check your Gmail, it read that, right? It read your geolocation [00:11:00] value, you placed, or I have to then connect you to the Indian Amazon website, not the UK or us. Right? All that is doing. Correct. So there are multiple functionalities that it has to perform.

How does it do that? There is a structure behind it, right? Search engine. What is a search engine? Right? It's got a huge database. And what does it do? It goes, crawls the website, right? You've read about all these crawling things, right? It goes and slowly builds a massive They have done it over, like, last two and a half, three decades.

They've been doing it, right? They've built up a massive database. And how do you quickly, when you go to YouTube, right? Let's say, there is one, uh, right, top video, right? What do you call a viral video, right? This has got, like, maybe, I don't know, 10 crore views. All of you go log in, right? Think about this.

You will log in at one instance, I will log in at one instance. Right? There is a physical hard disk that is keeping the data. When you play, you will start at 0, 0. Then you will fast forward. It should show you that instantly with the infrastructure. Somebody, [00:12:00] suddenly somebody from Hyderabad will log into the same, same, they'll go to the same link, start from 0.

Which means same video has to start at different points for different people. How do I do that? You understand? Such a simple thing. Have you given, given it any thought? If Millions of people are viewing the same video. They will view a different time instance, right? And you want it near the instant time.

How do you, how does it make it happen? There is some background structure to the data form that Google has, right? Just like how you have it in your RAM, right? It will store it, it will retrieve the data. There are multiple, right? Have you heard about RAID? RAID 0, 1, 2? Right? There are different levels, backups that Google will have, right?

There's a server in Bangalore, right? There's one in US. Right? One in Singapore. There are multiple copies. Right? But, when you go click, right? All you're saying is, okay, I'm watching a video. But, your request is going, right? It is requesting start. Look for this particular, uh, what do you call it? The video. It has to start playing from there.

Right? This simple process [00:13:00] for different people, you don't think about it. Right? And you will set it at different, uh, uh, Uh, what do you call it? Resolution. You HD. You may view it through your phone or 480 pixel. Somebody would want it at 4K. It has to stream that, right? You understand, so much is happening at the background, which is supported by some backend structure.

All that is going on. Right? This is clear? Right? Just viewing a video or sending a mail, when you start typing, it gives you a suggestion. Right? Predictive, uh, predictive, uh, text inputs. Most of the times it's correct, right? 70 80 percent of the time it figures out correctly. But sometimes, right, today I was typing F I N E, fine.

It was saying fined, fined. It kept re correcting me, wrongly. Right, for whatever model they've used, right, at least for what I was typing. The entire paragraph everywhere, wherever I put fine, it was saying fined. For whatever reason. Right? Structurally there is some problem. Right? We'll see. Fine? So, what I'm saying is, if I'm building a new product from scratch, in other words, If today if I start, let's say I want to be a competitor to Google, [00:14:00] I want to start my own search engine.

Where do I start? That's the, if I ask that question, I am in the discovery phase, right? Next. Okay, sir. I've understood, right? People are looking for certain things in a search engine, they don't want traceability. Your DuckDuckGo, how many of you use DuckDuckGo? Two. Some of you, right? Not many use that, right?

But you're comfortable using it? Any other search engine do you use? Like DuckDuckGo? Yahoo. Yahoo. Yahoo. What is it? Yahoo. Yahoo is like, Yahoo also trusts tracking. I'm talking like, like things like DuckDuckGo. Or ProteonMail, how many of you know ProteonMail? Any of you use it? Right? Not many people even know about it, correct?

Not many people even know about it, but you use it. Why? See, I'll tell you a problem. ProteonMail, the problem I'll tell you. Somebody is using that to send bomb threats. And the police department is unable to track it. That is the level of security they have. Right? It's based in Switzerland. Right? They have very strict, I mean, uh, [00:15:00] privacy laws.

Right? Even the government of Tamil Nadu has a tough time tracking who the culprit is. Right? So, that is, that's what our service. Right? So, that is the service he's providing you. Right? If you pay extra, right, to provide your 10 or 20 per month, he'll give you extra data, all that. Right? So, This is pretty much the discovery phase.

Now to diagnosis, what is diagnosis? You go to a doctor with some ailments. Sir, I am not able to breathe. I have this particular problem. What did they do? He puts a stethoscope. They ask you, what did you eat yesterday? How have you been feeling? Last, take this test. He is doing a diagnosis. So, same thing, the product or service, whatever you are building.

You ask that question. Ok. What is my customer's requirement? What is it doing? What is it not doing? Ok. Understand, this is very important, right? I told you about six thinking hands, right? So, there are two hats you have to constantly swap. What are those hats, sir? First is the hat of an engineer. What does an engineer ask?

As an engineer, what, what, what is the question you are asking? How do I do this? Yes or no? [00:16:00] How do I solve this differential equation? What is this, how, how do I solve it the fastest? Yes or no? Can I numerically solve it? Is there a, does it, does it have a, like, Guessable solution. Huh? Guessable solution.

Guessable solution. Okay, he's saying guessable solution. Right? Or does it have a standard template, if I can ask that? Right? Is there a standard template for it? Is there a generalized form? Right? Does it have a specific complementary function and particular integral for, for this, for this form? Yes or no?

You look at that, right? Or, sir, this is extremely messed up, right? This is a very non standard form. How do you solve it? Numerically only you can solve. Most, most practical engineering problems, right? Numerically only because you are, you will be like fourth order, fifth order right? It'll be crazy. You can't sit and solve by hand.

You need, you need a computer. Even though it may not even have a closed form solution. Right? You need to approach what is called as a numerical solution. Have you heard of what is a numerical solution? What is a numerical solution? Okay. It's like, more like, uh,

Let me,

[00:17:00] Okay. Where does, okay. Can somebody plot this? If I have to write this, X square plus C, how will it look like, or even forget the C. X square, Y equal to X square. Where is the minima for this, you tell me. Zero, zero comma zero. Zero, where? How do you know it's zero comma zero? How did you know that? Because you have that information up there.

Right? Now you tell me, what will be the Okay?

Now, you tell me where will be the minima for this? We have to differentiate it. Ah, that you have to do? Yeah. Okay. Now, let us make it even more messy. Okay. Now, now if you start doing this, right, gets messed up, right? Generally, if I say AX power n plus, right, I can write a generic form. How do you solve this?

You cannot. So, what do you do? Simplest way, sir, let me start substituting numbers. Let me put 1. Oh. X equal to 1. Let me plot. Yes or no? [00:18:00] Yes. That is what you will do, right? So, I can pretty much plot this entire curve and then see where is the minima. Yes or no? That might be the local minima, right? Possible.

Possible, right? I will talk about it. I will talk about local and global minima, right? Not, is everybody aware of local and global minima? Right? Have you been taught, where did you learn this? 12th standard. 11th, 12th. 11th, 12th, right? So, we will talk about, this is in 1D. This is one variable, just x. Okay, let us say if I have an equation like this, plus y square, if I mess up, now what happens to this?

It is in two, it is in two planes, right? I will show you a, a sample graph, right? How do you find the minima for this, right? What he said will come. Matlab simple have you seen? Contour, sir. Right? It is what is called as a convex hull, right? Matlab simple all of you have seen, right? You have a response surface, right?

That is a beautiful example. I will give, I will present an example. You understand? So I can substitute numbers, right? I can And then slowly it is called exhaustively. I can do a exhaustive [00:19:00] search, put numbers one. Okay, let me put one for X. Let me put one for Y. What is that value? Let me put two for X, two for Y.

Let me put minus one for X minus one for Y. This way I can pretty much do it for a compute reason. Easy or not, nothing, right. It'll do like that. Okay. If you put it on a graphing calculator, if you plot this right, how does it solve it? It'll just plug in values. Right? It will generate that. So that's how, numerical limits is how you solve it, right?

Just a very crude example, right? Numerical methods, if you are interested, it's a beautiful, uh, area, right? I digress. So far clear? Fine? So, an engineer is asking the question, how to do this? How do we compute? How do I find a solution for this fastest, with the least effort? Right? A designer will ask the question, when you change the hat, what am I doing and why am I doing this?

Why am I sitting and solving this equation? That's a very important question. Why? Because for an engineer you are [00:20:00] stuck here. You are looking at a problem dead on and then trying to solve it. As a designer you step aside and see should I even bother with solving this problem? Can I do something else?

That's the question you are asking, right? Often you have to swap between these two. Got it? Right? So, so design, right? Typically, right? We say it is an artifact or a process, right? And you have to differentiate what is called conceptual design and detailed design. What is conceptual design? Sir, ok. Let us say tomorrow IRCTC is the same.

We will bring up the website. Right? We will make it even more user friendly. Yes or no? We will make it even more user friendly. How do we do that? How do you do that?

How do you do that? How do you make it? Find the flaws of IRCTC. What are the flaws? Uh? No, no. That is by design. That is by design. I'm, I'm saying some fundamental problem with the website, you tell me. Sir, font size is less. Font size you can, uh Like, for [00:21:00] example, where to log in that, that, that UI. You're talking about UI design, basically.

Right? The user interface is maybe better. That's what you're saying. Right? Can be improved. User interface can be improved. That's what he's saying. Right? So, that is one thing. Fundamentally, I can see. So, I can ask the question, Who is using my website? Thank you. Anyone from 10 years old to somebody who is 70 or 80 years old, okay?

What is their education level? Right? How conversant are they with technology? You understand? Did you ask these questions? You straight, you are looking at it from your perspective. From Badresh, from your footstep. Leave your body. Step aside. Say, how will somebody, right? Who with, with a 10 star education.

This guy should also book, right? How can I make it easier for him? Maybe put it in a local language, regional language. Now it's in Hindi and English. Can I put it in a regional language? Possible? Right? That is one approach. Ok. [00:22:00] Ok. What is the bare minimum knowledge that somebody needs to operate this website?

What is the bare minimum? You understand? Or can I completely get rid of words or only go with pictures? You understand? What am I doing here? I am doing concepts. How am I going to do? I don't know. I am simply saying, can I simply put a map? This is one concept. Chennai map, Hyderabad map. I have to book a ticket, click Chennai, from to Hyderabad you click and you say, you show a family and say father, mother, grandfather, write grandmother, you click, age, right, put a scroll.

As a concept I am saying, sir, let us build a website like this. Somebody who has no education, if they can just click. So, this is a concept that I am presenting. As a designer I am presenting. I am not saying whether that concept is good or bad, this I am, what am I doing? For you in your mind, what is it [00:23:00] you are thinking?

A website has to have user name, password. Yes or no? With this approach, what is the problem? You are stuck solving this equation. What I am saying is, should you even bother solving that equation? You Step aside and ask the question, what is the ultimate objective? To book a ticket. Right? What are all the ways in which I can verify it is one particular person?

This guy's credentials, right? Can I even get it from his phone without even, right, him explicitly inputting that data? God, is it even possible? So if I ask that question, right, then I look at what is called as a conceptual design. You understand? Right, which forms this side of this FPS diagram. R, F and P.

This side is conceptual, sir. Completely conceptual. This course is conceptual. I am not going to detail design. This side, right, this side where we have, right, it is called embodiment design. What is embodiment? Physical structure. How do I convert an idea, just a concept to [00:24:00] something I can actually make touch and feel.

Got it. How do I go from here to here? Right? That is the question we are asking how we go from a conceptual idea to an embodiment design, conceptual design to embodiment design. How do I do that? Right? That's the question. I'm asking. Fine Is this clear? Everybody is clear with this? Yes. Right? So, let us proceed further Right?

So we are here, design and from there design you will go into development. What is development? You have a huge development team, software right? Most of you are, right?

I am a software developer. What do they do? I write some code. What are you writing a code for? Some specific application, sir. Okay. What do you do there? I am a tester, sir. Right? Here, mostly, when you talk about IT, you start with the deployment phase. You are in the deployment phase. Sir, I am in the deployment phase.

Right? They'll say, Right? That is my sister's thing I'll tell you. Right? Okay. Okay. Often, I cite this example because that is what I get to talk to [00:25:00] often, right? So there is this Philips or Siemens, they have got a user database, right? Millions of users are there. They'll log in their problems. They'll say This xray machine brought down like this.

Because of this reason, they'll input, which is in us. There'll be another xray machine, which they might have supplied to Philippines, same problem. Or machines which are supplied to regions where there there is hot temperate in India, Southeast Asia, it'll fail Europe. It'll be fine. You understand temperature sensitivity.

So there is a pattern, right? So Phillips is building, right? And then saying, okay, if I'm building a mission for India, higher temperature, certain components have to change. If I'm building a mission for US and Europe, I have to build it to a different spec. In other words, I have to change the structure, right?

I will get there, fine. So where are we now? We are in the deployment phase, or no. And then every day she'll say, I am seeing a pattern. This sort of, and then. She's writing a software there which will auto flag and then it will give a report, right? [00:26:00] And then it will give it to Philips and then they, when they say, Okay, what is our most failure prone component?

Philips is in the business of healthcare, right? Philips has exited all this lighting and all. Namcaves only they put. Philips no longer builds any consumer devices. They have exited the market. They have given their name to somebody. Right? Philips only today builds in healthcare. If you see the name Philips, right?

It is only for healthcare. Any, anywhere you will say Philips, like Mixi grinder, it's no longer, Philips no longer builds it. They have given it to somebody. Right? Now, in healthcare, what do they build? They build MR, MRI, ECG, uh, X ray, CT scan. Right? All this diagnostic equipment is what Philips build. And then they are asking the question, How can I make it better?

Right? Then you say, Where do my machines fail? How do they fail? Yes, data na, this is all complete data. Right? Okay. There is a failure. What caused that failure? I need more insight. Right? So, they have all these diagnostic tools, right? That is all built by pieces of that. Right? Subsystems [00:27:00] of that is built by CTA, DCA, CTS.

It is all given. Right? There are developers sit and then they break their head and then figure out all this. Right? Complete data analytics. Fine? Where are you here? You are in the deployment phase. Yes or no? Correct? How do I innovate from deployment phase? If I am not doing new product development, how do I do deploy?

How do I innovate? You can ask that question, right? How do I innovate if I am in this phase? So you can innovate. How? You have the existing product. You know how it fails? Sir, I have my current IRCTC website. Sir, I know my Gmail, uh, right? Mailing interface I know. I know where it fails. Right? So, some of the new features they have added, right?

So it lets me compose a mail and then it keeps something packed out at the back and I can refer. So two windows I can open at the same time. This was not possible with the previous iteration. Where am I? I am in the deployment phase. Somebody tells me, Hey, every time I have to go, I have to minimize, go, read, right?

Can I keep two, two mails open? I can refer and that's it. Right? That's a requirement comes from where? Here. So deployment can also [00:28:00] go back to discovery. And that, that is why you can start anywhere in the cycle. There is nothing like you have to, you should start here. You will not have the luxury. Most of the times, right, you will work with the existing product, right?

And then when you work with the existing product, you are in the operational phase. If I am in the operational phase, I can always, the next version, right? This, I dropped once, the cover broke. Only once I dropped, right? So, Targus asked me, make it more robust. Right, I dropped this once and it broke. So, next thing I say, oh, this particular, Right.

Pin broke. Let us reinforce it. Right? So, this is how you can actually come up with new creativity. Where does creativity come from? Where does innovation come from? What is innovation? Building new, new things, sir. Right? True, but it's very difficult to innovate. Yes or no? Right? Latest iPhone, what is innovation?

Hardware has pretty much, has not changed, right? It's all in the software. It's all in the image processing, what it does in the background. Yes or no? Pretty much it's all software today, right? Computational photography, right? Hardware, right, pretty much is [00:29:00] saturated, at least in the smartphone world, correct?

So, the focus in system thinking, right, is to improve effectiveness of the design process and produce better conceptual design that are both functional and structural, right? We'll see how. Okay, fine? Let me quickly recap this. Talk about ISM. Then I will actually get on with today's class, right? So, what is this FPS, sir?

So, FPS is a model, right? It's created in Stanford, right? There are, uh, people who specialize in system thinking who look at, right, all this. And then they have come up with this model, right? What does this model say? Let us look at it, right? So, what is the point of this model? Why am I even bothered with this?

This model helps you to capture conceptual design specs. Okay? In other words, okay, What should be the point? Minimum number of clickable elements in my website, right? What are the bare minimum data I should get from my user? Right? Should I have one factor authentication? Should I have two factor authentication?

All these questions I can ask. Yes or [00:30:00] no? Right? So that is what is called as a conceptual design specifications for my product or my service. Or my product service system. Could be anything. Right? So, you start with the requirements. Right? What is called the requirements document. Guys, I, I hear somebody talking.

Right? So I start with the requirements document. What is the requirements document? What is the customer gives me? It will be very ambiguous. Right? He will not be clear. Customer is extremely ambiguous. I need a good website. I need an easy to use website. Website should load fast. Right? This doesn't help me.

Right? Now I should take that set of requirements and turn it into functions. Right? One level, it's like taking, I give you one liter of milk and I say get clarified butter from this, ghee from that. You have to keep churning, churning, churning, take the butter, then heat the butter, right? Take it into ghee.

Right? That you have to do. That is what we are doing. Taking very ambiguous requirements from my customer. Turning them into functions right? The function should be, right, [00:31:00] it should load fast. Load fast means what? With, even with a, if I give it a 100 kbps connection or 5000 kbps connection. Right, within 2 seconds the website should load.

Right, without any, you have seen this ACID test? Website they have this thing called ACID test, ok? Right, just go read about it. So it should pass that. My browser should pass that test, right? Whether I load it in Firefox, whether I load it in Chrome, or Brave, any, any, any, or, uh, whether it is IE Internet Explorer, right?

It should load. Right? So I can write down a lot of functions. Fine, sir. Functions I have listed down. Right? What are the functions of a cell phone? Should make a phone call, should be able to browse the internet, should be able to, uh, look at maps. It can go on and on. Right? Should be able to take a photograph, video.

So, fine? How do you get these functions? Right? Right. How do you get this? How do you get a service? Something has to work together as a cluster. Some multiple elements. Right? Let's talk about the physical product. Right? I'll go back to mobile phone. Right? [00:32:00] How do you make a phone call? Sir, I need to have a battery first.

That has to give power source. Ok? Where will the power go? Sir, it will go to systems on a chip. Right? It's got a GSM module. Right? So that will basically talk to the cell phone tower. Fine. Right. Now I need a mic that will pick up your voice, that will digitize your data, right? That will turn it into packets, right?

Packets of data. Fine. Now I have a transmitter, physical antenna, that will take this and it will transmit. Fine? You understand? All these components, they are tightly linked, right? Which forms what is called as a structure. This you are clear, correct? This already we have discussed. Right? For me to achieve one function, what is the function?

To talk to another guy with this. Cell phone, right, with a phone. What are the things I need? I need multiple components. I need battery, mic, I need all these electronic circuitry, I need the background infrastructure. This is what we call as a structure, right, to fulfill my functions, right? Guys, listen, this is very important.

Last bench, sit straight. [00:33:00] Don't droop. Right? All of you sit straight. Couple of them are sleeping. Wake your friends up. Behind you. Go wash your face and come back. Right? Right? Guys, today's class, I'll wake your friend up. He is also drifting off. Yeah, you, last guy. Yes. Right? Yeah, go wash your face. Wash your face, man.

Right? So, yeah, go, go. Go wash your face and come. Right? So, you have, guys, if you don't understand this, you will really struggle with the end semester. That is why I am actually coming and then spending time and revising all this. Right? So, you have multiple components, right, that interact with each other.

That will help you fulfill not one function, multiple functions. Got it? Fine? Now, what you have to ask is, what is called B. S.? What is B. S.? Behavior from structure. If I use this size battery and if I use this microcontroller, microprocessor, right, and this particular modem, [00:34:00] this is the behavior I will get.

In other words, if I use all this, my cell phone can actually talk to an antenna which is 5 kilometers away. If I slightly put a bigger antenna, if I put a slightly bigger antenna my cell phone can talk to an antenna which is 7 kilometers away. Or, if I use this battery, my cellphone can only talk to an antenna which is 1 kilometer away.

You can have different structures that leads to different behaviors, fine? Got it? I have one structure where the GPU and the CPU are integrated into one module which is one structure. That will lead to a certain behavior. you. I have another structure where both of them are discrete. The GPU, right? Your Arduino, right?

Mali, whatever you call that, right? That module and my CPU, which is my central processor. These are two discrete elements. That will lead to a different behavior, right? When I have that, different structures will lead to different behavior. Similarly, database, how do you manage all these databases? If I locally store the data [00:35:00] in Chennai, if Google has got a server here in Chennai, right?

Okay. There will be a certain speed with which I can retrieve the data, right? It's close by, server is close by, I can ping faster. But rather if they have parts of the data here. In other words, if I go to Drive, maybe they store the elements of Drive in a server which is far away. Maybe they have a smaller data farm close by.

From your computer you will. So depending on which, uh, device that I use to access Google services, I might actually use different kinds of entry points, nodes to, Connect to my data server. You understand? So which means it's a different structure that I follow. Right? If I do that, right? The behavior from this picture will be different.

If I use browser on web through a mobile phone, it's going to be different. If I use it on a laptop, it's going to be different. Now, this is one thing. What is that we want? We have converted R to F, right? Requirements to functional. What are the functions I need? And, what are the expected behavior? What?

From this product, right? [00:36:00] Sir, the website should load in half a second. Does it load in half a second? Right? That's the question I'm asking. What is the expected behavior? Whether I go on a phone, tab, I don't care. I can use 2G, I can use 5G or 4G. Ok? But my website should load in, in a prescribed time. Fine?

That is what is called expected behavior. But what is actually happening? Sir, that is not happening. You want it, you want the website to load in one second. Ok. But because of the structure I chose, it is giving me 3 seconds to do. Now what you do? Sir, I iterate on the structure. And then what you do? I iterate on the structure.

I iterate on the structure, exactly. Go back, look at the, what is the expected behavior? What is it I want and what is it I have? Compare, edit the structure and you keep doing it. That is why you see this arrow. You see this, right? And you see this arrow, you go back to the structure, edit it, come back, compare, right?

Look at this, comparison. How is that? Compare the expected behavior, compare the behavior from structure, keep comparing and iteratively. There is no end to this. Somewhere you have to [00:37:00] stop. Why? You can never match these two. Why? There is always cost, sir. Physical infrastructure. Of course, if I put a data server farm everywhere, in every city, my access speed will improve, but I have to spend a lot of energy, electricity.

Right? I have to provide security. Right. All that is required. Right. My cost goes up. Right. My customer is not even paying. How many of you subscribe to Google, uh, Google's services? No. We all use it for free. Right. Our institute pays. But for a normal user, I don't pay. Fine. How do I run this service? And yet, keep what is called quality of service.

Q O S. What is the quality of service Google provides? It's pretty good. Right. When was the last time YouTube was down? Never. It's never, right? It's what is called QoS, Quality of Service. Right? It's pretty much 100%. Right? So there was only, my life, there are only two times I know where YouTube service was down.

Only twice in the last 20 years. 2006 they started. Right? It's about 18 years old. Right? Only two times where YouTube went down in my, [00:38:00] at least, what I have personally experienced in the last two decades. Yes or no? Which means QoS is near 100%. Fine? This is clear? Fine? Let's go ahead, right? Now, this I'll skip.

Pure mechanical thing, right? For mechanical Anyway, that's why I gave you examples of website, right? That structure, right? Always remember the data form, that sort of thing. This is not important for you guys. You don't have to use this example, right? Fine? This is fine? Okay. So, here D, when you get to what is called prototyping and testing.

In your 6th semester, you need to be here. What is that? Okay. You need to be in embodiment design. You go from a conceptual design here and you go to embodiment design here, right? So here we have what is called pretotype. What is pretotype sir? Right? Oh. So today somebody shared a document with me, right? If you would like I will share, right?

Literally what it says is, maybe I will post it in the, let me quickly check it, I forgot. [00:39:00] It says, iron on fire framework, okay? It is a new idea. What is this hair on fire framework? Okay. Let's say suddenly my hair is on fire. What will I do? Put water. You will do everything or anything to put it out, correct?

Water is there, I can go grab it. But water is not there, what will I do? So what he says is, even if you have a brick, you will start hammering with the brick. Yes or no? So this is what is called as a hair on fire framework, right? If your hair suddenly goes on flames, how do you put it out? Right? So, in other words, there is something called Minimal Viable Product, MVP.

Minimal Viable Product. Right? What is this MVP? Okay. Let's say, right? Uh, I'll give you another example. Tax, income tax, website you say, right? Infosys is what developed it, right? Now, before they give the order to Infosys, okay? [00:40:00] It's a huge order, right? Okay. It will be, it will cost several crores, right? I don't know, probably they paid 50 crores, I don't know how much they paid, right?

Let's say it cost 20 30 crores to build that website, okay? Why? First of all, data security, all the PAN card information, Aadhar card information, all your, uh, form 16, right? Whatever you are, you are earning, right? Your entire data I have, financial data, right? So I have to first encrypt it properly, right?

Which means my security, data security has to be foolproof, first thing, right? As Infosys is a company, largely they'll have the means and resources to do that, fine. Now, can you gimme a quick mockup If you go ask him, what's that? I said gimme 50 quotes. Come and meet me. After two years, I'll give you a website.

Would anybody be okay with this? No. What is the first thing I'll ask you? I'll give you a month. I'll give you a month. I don't need complete functionality. Show me. Yeah. Bad minimum order. In other words, show me where. I'll give you panka data. I'll give you all your financial data. You tell me how a customer will go, how they will interact.

Can you build and show me? I'll [00:41:00] give you one. This is what the Income Tax Commissioner, right? The guy sitting on the top, he'll demand. They'll say, yes sir, we'll do that. What are they doing? They are doing what is called, not prototype. It's something what is called, colloquially, pretotype. What is called the first, the zeroth prototype, right?

I start with just the bare minimum. I build a website, just so I can say, see, is this okay? It's like hair on a fire, na? I'll just quickly put together something to show you how the final product will look like. Right? So that is what we call it as pretrotect. Fine? Right? So, in system seeking we are focusing on linkages 1 and 2 from here all the way to here.

That represent 3 stages in fuzzy front end of product design and innovation. Right? This we are clear. Discovery, diagnosis and functional design can bring us clarity on F, uh, behavior from expected behavior and their relationships. Functional design can then be translated into different structural design.

Right? Whatever is required to structural design. One more thing I have to tell you, right? Another important thumb rule. What is [00:42:00] thumb rule? Often I have talked about what is called function and form. All of you understand, right? Sir, what is form and what is function? Sending a mail is function. I have to send a mail, electronic mail, e mail.

What is the sort of interface it's going to be? Can I simply dictate, can I build a, right? If tomorrow, okay, If tomorrow, okay. R Rabbita. There was this, have you seen that? The failure product. What is it called? R Rabbita? Rabbita. Yeah, ulta. I'm saying ulta. Right? Rabbita. It's a mega flop, right? Mega flop, right?

What did they try to do? Guys, I hear some murmur. Guys, I hear some murmur. Right? So this Rabbita, you've seen that orange dupper, right? 200. How much was that? Right? It's a failure product. What did they try to do? They tried to replace a phone with the worst device. Right? Why did it fail? It didn't actually solve any problem.[00:43:00]

Yes or no? It didn't actually solve any problem. It, it didn't even replace a phone. And then what? Finally somebody hacked the back end, and then he ran it as, he turned it into an app and ran it on a standard Android phone. It ran. Right? So that failed. That failed miserably. Why did it it fail miserably?

Right? So you have to ask the question, is form important, is function important? Okay? Always we say, at least, right, most of the times, right, you cannot, I mean there are places where there are exceptions. Form, repeat after me. Form. Form. Form, louder, louder. Form. Form follows. Follows. Function. Function. Most of the time this idea will serve you well, there are exceptions.

Okay, I'll give you an exception. What does this form falls function mean? What is this, sir? I'm saying what does it do? Your service, what is the primary intent? What is the primary target? What is the primary object? That is the important thing. How do you do it? Gives you a structure. The structure is the form.

I might even come up with a [00:44:00] top with my phone and I can dictate my mail, right? Yes, sir. Today I'm only sent, I don't even have to type a mail. You understand? What is it? I have to send? I have to send them mail. I can even be a voice instruction. Right? So the form is different. How I implement is different.

But what is the functionality? To send a mail. To send you electronically. To send a type. Right? I don't have to type the mail. I can even have something like what you learn, the neuronic, right? Might even put a cap. Neuronic. Think. That sends a mail. 10, 15, maybe 20 years later you might have a cap. Everybody may have a headband, right?

They are trying with these head straps. Right? Where you think you send a mail. Right? You control a cursor with your eyeballs. Right? You move, cursor moves. Right? So, how you do it? Right? But what is the function? Right? So, form can, you can take different forms. I can, I can talk, I can type, I can think and then send.

Right? Form follows function. Which means function [00:45:00] takes precedence. Hierarchy. I am going to talk about hierarchy today. So, understand that. Right? So, 3, 4, 5 as I already said. Okay. So that's the definition of a structure, right? It's a structure. We will discuss that in the next class. Our next topic, right?

This is the definition of structure. Structure Right? So on and so forth. Right? This, you all know, right? This I have told you. Science, technology, product, society. Right? This is bottom up or top down. You can look at it. Right? There is an economic issue. Right? Which users have a What are you guys discussing?

Right? I [00:46:00] really don't want to throw somebody out. Right? If I cut your attendance, you cannot write the answer. I am cutting you so much slack, but you are not making my life easier. Right? Right? Right? Right? So, you have a business opportunity, right, and there are users, they have what is called stated need and unstated needs, right, which, right, that functionality is met by what is called as a technology problem, right.

So, we go from problem space to solution space. I have this problem, using this technology, I can leverage this technology to help out this stated unstated need of my customer, right, then that is based on some scientific principle. Fine? This we have already seen, right? Now, what is this ISM? Right? Why did we first construct, we looked at a discovery matrix?

What is the important thing? We said, if I have to innovate, right? What is the example I gave you? If I, if suddenly, group 1, suddenly, Lipton, all these guys are all making tea. Suddenly, I start a new company. And then say, how can I innovate? [00:47:00] Right? If I also make, or let me take some other example. If I also come up with a new mailing service, I'll provide you a mailing service map.

Okay? What does that have to do? One thing is I can completely copy paste Google. I can copy paste other mailing services. Will that sell? Nobody will be interested. It's free, I can say. Already Google works for me. Right? You have to provide me a compelling reason to switch. Right? Even a service like Proteon.

Right? Right, they guarantee security, where they, where if the government goes and asks, they say we cannot give. Right, I am not even paying for the service. Some may be tempted to switch, but even that is not, you understand, that is not compelling enough. Right, I started with Rediff Mail, then I went to Yahoo Mail.

Right, one of my friends came and said, uh, I was in my third year, he laughed at me and said, what are you using? Uh, Yahoo Mail. He laughed at me and said, why everybody's using Gmail? Why are you using Ready Mail and Yahoo Mail? I said, it works fine for me, man. You are losing out a a lot [00:48:00] money. He just took me, he said, he's a CS guy, right?

I was a mechanical guy. I didn't know much about computers. He said, look at this. Can you see the drive? What is this? I don't know. How do you save your data, man, I keep it as an email. Attachment 25 was the limit. He said, that is no limit. You keep one gb. Google gives you one GB of memory for you. Like what?

Okay, this sounds nice, right? Well, attachment I can actually put more than. More than 25 mp. This is nice. Okay, there, tell me more. Okay. Then it shows me and see your YouTube is, think it's one interface built. I am sold that day onwards. Now what has happened now for, for me, for a service to take me from there to elsewhere.

Now you see, how do we do that? It's difficult to answer that question. Right? In Discovery metrics, what we did, right? So Google would've done this, exactly, this discovery phase, and they asked the question, what is, what is my user looking for? So he's using multiple servicer OneDrive. He has it separately, right?

He has got his mailing service separately. He's got his, uh, there is something called Meta Cafe. Have you heard of that website? [00:49:00] Meta Cafe? Uh, Google, uh, video was there, right? Plus, no, no, no. Wait, I forgot. What is it called? Google Video or something from nine five. Oh yeah. Something like YouTube. YouTube was bought by Google.

Google had their own, I think it's called Google videos. videos.google go.com something, right? They had their own. So, you know, it failed, didn't pick up, they shut down the service, right, just need a more data. So, what did Google do? They tried to build a service, they couldn't succeed, they bought out somebody who is better than them, and then integrated as part of their ecosystem, fine?

So, discovery phase helps you do this, helps you say, dude, nobody is giving you one unified interface. Your mailing service, your videos, your data storage, give one login, ask people to do it. And for a enterprise customer, right? We are like an enterprise customer, right? Google, we, all our mailing service, aren't triple LTM, but you log into Google's thing.

Right? That also I give you. I give you security, and I take care of all back end headaches. Right? You pay me, I'm going to take care. [00:50:00] Fine? So, this, how do you find areas of innovation? That discovery matrix helps you do that. That is done. Diagnosis, what did we do? What is the one important rule I taught you?

SNACC. Who are the stakeholders? What are the needs? What are the things you can change, which we call alterables? And what, under what constraints, right? What, what differentiates constraints and alterables? Time. Right? Alterable constraints are the same thing. If given enough time, I can, I can change, it means it's an alterable.

If I cannot change in a very short period, for instance, layout of this room, can I change? No, it's a constraint. Right? Can I put 500 students in this room? It's a constraint. No, I can only have 180 students. 200 students. That's the capacity. That's the constraint. Right? So, given all this, what are the needs?

Right? Then we looked at what are called objectives. Right? You got, you derived multiple, yes or no? Got it? Right? You listed out, right, some 20 25 objectives as output of snack, as output of diagnosis. Yes or no? [00:51:00] Discovery, what did you do? Yes or no? There you try to put a hierarchy, important to what matters to me the most, right?

What is the least important? Here what did you try to do? You put multiple objectives. Now here what we are going to do is, right? Specifically, right? You are going to look at what is called functional hierarchy. I am going to look at multiple of these objectives and say, See, system seeking on a broad level is all about patterns.

Can you see the broad patterns? Right? I will explain more. Right? So, it will be a bit abstract, right? When I explain, you will understand. Right? Now, you are doing what is called functional hierarchy. What is functional hierarchy? My product has got different, different structures, which are clusters of elements that are connected in dense nodes network.

That right? This you know. Now, that, right? Right? Give rise to fulfills multiple objectives of my of whatever I ask of my product. Right. Now the question is okay out of [00:52:00] my structure which is the central node I can ask that question and which is and how many such hierarchies are there? When I say hierarchy, see okay, do not think something is most important to something least important.

Can you have a one break today or do you want two breaks? Two breaks. Two breaks huh Two breaks. Okay, so I'll let you another five minutes, right? So it's only given one break. Always, guys, listen. Listen, right? If you understand this right, next class is like of this. I can do it like 10 minutes. I can finish today's class.

I can actually, I can finish in 10, 15 minutes. It doesn't take long. But you have thoroughly understand this, you right? So I ask what are we doing? Is we're looking at right levels of hierarchy. Repeat after me. Levels of hierarchy. Levels of hierarchy. What is levels of hierarchy? It means. Sir, there are certain objectives which are at one level.

There are certain others which are at another level. That is all I am saying. I am not saying one is important than the other. I am not saying if the teacher is important or if the student is important. No. I am saying you are at one level, I am at one level. That is all I am saying. Right? [00:53:00] Now, what is this decomposition?

Guys. What is decomposition means? Sir, it is decomposing. Vegetable is decomposing means what? It is breaking down into its fundamental units. That's what it means. So when we do what is called functional decomposition, what does it mean? Matrix decomposition, have you guys heard of this? One example can you tell me, matrix decomposition?

LU decomposition have you heard? LU, lower and upper triangle. Are you aware of this principle? You can take any matrix, yes? You can take any matrix, decompose it into upper, upper and lower triangle matrices, right? Why do I do that? How do I compute? One of the fundamental questions, right? I'll tell you, right?

I'll give you an example, right? Then we'll get, go for a break, right? Once you come back, ISML cover. Think about this. All of you have a cell phone? What is the typical resolution of your screen? 1920 x 1080. Full HD, correct? That is PC. Oh, that's PC. [00:54:00] Full HD now, right? We call it full HD, right? 1080p x 1920, right?

Whatever, right? That's like number of pixels, right? Now, how many pixels are there roughly? 2 million? 20 lakhs, right? Now, understand this. The amount of work your processor is doing. I have 2 million. I have 20 lakh pixels. Which are white dots, correct? How do you make a white dot? You mix three colors, right?

Red, blue and green. So, I have my sub pixel array. For every pixel, there are three sub pixels. So, it is actually 2 into 3. How many are there? Six million, if I go at the sub pixel array. So, one frame of my, whatever it is showing is how many pixels it has to compute. Six million dots it has to compute. It has to, and various levels also.

This, right? Typically it is 256, right? How bright is my, right? If you go to color space, right? Then it gets even more complicated. Just think about it, right? I am giving you a very simplified tool. [00:55:00] argument. So, one frame, what is the refresh rate of your phone? 120 hertz. 120 is, not everybody has, 60. We will take 60.

Nowadays everyone has. 60 hertz is what typically most phones have, which means 60 into 3, 2 into 3. How much is that? You understand? The number of dots every second. Your phone has to compute is that many. In other words, you can look at it as a matrix. Okay, let me, let me simplify it for you. Okay, let's not go to colored screen.

Let's look at a monochrome screen. Pixel is on or off. If I take a monochrome screen, is it 1 or 0? Just that. Forget sub pixel, all that. It is just 2 million dots I have to compute. How many times a second? 60 times a second. Right? So 2 million into 60, how much is that? 1. 2 crore. 1. 2 crore. Either 1 or 0, it has to compute every second, right?

If you are playing a video game, something is computing that has to convert to this, she is like, you may not have thought [00:56:00] of this, right? So, what your phone is doing is simple matrix manipulation. 1s and 0s, it is taking a massive matrix and it is doing what? Put 1 here, 0 here, that is all your phone is doing.

Now, this is in n by n, right? Now, you go to Spatial. What is spatial? When I say 60 frames per second, it is one matrix array. Array in two 60 temporarily. This is my first matrix. Next microsecond, second matrix, right? I keep updating now you see right? Temporarily you can move in this fashion. So I have a big matrix that keeps updating 1 0 1 0, right?

And that temporarily goes now for a phone. It is too much for it to handle. So your phone will use a lot of optimization techniques. To compute, right? To reduce the computational load. Software, how are they reducing? Right? So, CS guys like you, right? You have, you sit, right? Bang your head, figure out new algorithms.

Right? You figure out new algorithms to find how do I compute all these matrices with the least effort. [00:57:00] Right? One way is to say, I will keep adding more transistors. That is one way. Or, this is the transistor I have got. In my software side, I Or in other words, from a mathematical side, it's a mathematical problem.

How can I do this quickly? I can ask that question. Fine. If I do that right, then with the same processor I can actually do more. Got it. That is where Nvidia keeps updating your drivers rate. What does it do? Right? They will say, power emissions increase by one person. They'll say 2%. How do they do that?

They, they would've found a new algorithm. Right, which in a certain way, rate tracing. We found a new way to rate race. That reduces mathematical complexity. Now instead of your processor taking probably 85 Watt, it will take 82 Watt. 3 Watt we have reduced. Right? Mathematical workload we have brought down.

Right? This is all they do in the software. How do you come up with these things? Right? Why am I talking about all this? Right? You do what is called a decomposition. If you have to compute, simple question I'll ask. All of you can compute determinant for a 3 by 3 matrix? By hand. 4 by 4? [00:58:00] You can? Painful. Okay.

5 by 5? No way, right? No way, right? Now, imagine computing determinant for 1 million, your cell phone's display. Not possible, na? But your phone is doing it, right? So, what your phone does is, it does this. It takes that motor matrix, decomposes, there are patterns it will see, right? Just your JPEG compression, read about JPEG compression.

How does it take, right? Typical TIF image, which is like 10 MB, all the way to 200 KB. Right? How does it do that? Right? So it will throw out some information actually. It will discard some information. Right? Some which our eyes cannot see, it will discard some. It's called pair mask. Right? Our eyes are sensitive to green.

So 50 percent of the image is green, then red and blue. Right? Lot of thing they do. Fine? So similarly, right? I've given you a lot of background. When we want to look at these hierarchies, there are a lot of decomposition methods that are available. What is this? Functional listing. [00:59:00] What is the expertise required?

Very low. Anybody can do this. What is functional listing? Right? You take your product, write on thing and say, which component does what? Battery supplies power, sir. CPU does the computing. Everybody can do this, right? Second thing is black box. What is black box modeling? Right. , all of you know, black box monitor, you don't really understand what goes on inside input and outputs.

You compare, right? So that needs medium expertise. Axiomatic design. What is Axiom? Axiom Solve probability. What does it mean? Basically, what is axiomatic? I'll give you an example. I, I can say it. If I don't know, you don't know. That's axiomatic means. As a teacher, if I don't know, as a student, most likely you will also not know.

That's axiomatic, right? So it uses that principle. You understand, right? What this means. Second, Right? For an engineer, the key skill you have to possess is what is called reverse engineering. What is reverse engineering? [01:00:00] I just go, somebody has written an app, sir. I don't know what the hell this guy is doing.

Right? He's able to scan my face with the same processor. Everything is the same. Hardware is the same. He's doing it in one tenth of a second. And when I put my algorithms, right, it takes about a second. He's doing it in 100 milliseconds. So you go to his code, sit, read through his code, try to, Decipher, or in other words, try to reverse engineer what we doing.

That again, needs no expertise, right? So you look at, and then figure out how has he done this? How has my competitor done this? Right? So what is Microsoft doing? They look at Google, they look at Facebook, right? They look at other competency. How can we stay, stay on top of the game? What is he doing? How does he store data?

Right? What is his architecture for his server form? What, how are we saying. Microsoft even putting this, they came up with a very weird plan. They went to, I think, Norway somewhere, where the sea is very cold. 4 degrees, 3 degrees. Why? Server forms heat up, na? [01:01:00] What they said? Okay, we will submerge the server form.

We will put it underwater. It will be cool. Some crazy idea. Right? They tried this. Microsoft tried this, right? Again, they had other problems, right? Salt. Salt will corrode. Right? Again, if you put it underwater, ships, they will put anchor. Right? Other But they tried this weird idea. It's a different concept Microsoft tried, right?

Another thing is called Benchmarking. What is Benchmarking? What I exactly said. Look at your, and the key distinction you have to draw between Benchmarking and Reverse Engineering is this. Reverse Engineering is, I do this. Ok. This is like this. Ok. Oh, you can put a cover like this. Fine. What about this? A different mic.

Oh, you can put a cover. Ok. right. So, I am just going to tell you a simple equation, right. Now, here there is an index of x. So, we have an index [01:02:00] of x, we have an index of x, ok. And you see, index of x is Right, so what I am going to cover, right, ISM is one decomposition method, right. Go for a break, come back in five minutes, right, we will take a break.

Actually I am a guillotine. I am a guest of honor. I am not a guest of honor. Huh? I am a guest of honor. I am a guest of honor. What happened? Look at the mood. No player will be there. I am a guest of honor. I am a guest of honor.[01:03:00]

Guys, make your, your neighbors. What do you guys ever sleep in the night? Huh? What time do you guys go to bed? Two o'clock, huh? I know that. But what do you do with that? [01:04:00] Aside from sleep. You're sleeping in, brother. What do you do? Guys. Right guys. I, I'll tell you, one hour late, okay? We got, I'm covering last, last week's class, right?

This will take me another 10, 15 minutes to cover today's class and the NK modeling, right? Morphological chart I'll cover. Then we'll wrap up the course

person. Amazing.

So, what is this ISM? ISM is one of the methods that will help you to find what is called the functional hierarchy, right? What we call, we are going to decompose or in other words, [01:05:00] we are going to ask that question, right?

Right, we are going to ask that question, right? In my, guys listen, in my product there are lot of components and they form lot of subsystems, okay. I will cover another concept called DSM, design structure matrix, right? It will follow ISM, right? Same principles apply there. So, in other words what we look at is called modularization.

What is modularization? Many examples I have given you. If you go to Python. If you want to compute, let's say, exponent of a number, e power x, okay, e power 2, what is the value? Do you sit and write a function for that? No, you just call a ready made function. What is sine of a function? What is tan, cosine, whatever, right?

So you have these ready made modules or ready made functions, sub functions that you call, right? If you look at the entire code, it will be in [01:06:00] layers, multiple layers, right, which are composed of? So when I say a, a, a, a, and I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, I, A, B, C, D, E, F, G, H, I, J, I, I, I, I, I, I, a, a, a, a, a, a, a, a, f right, Bye.

For a mathematical function it is easy enough. Right? I can take standard trigonometric function and put modules for that. Standard things like if you want to compute probability, if you have to do any statistical function, statistical analysis, I can build a module. You go to MATLAB. Image toolbox is there.

Statistical toolbox is there. Right? It's got many, many toolboxes. Right? How did they come up with this? In other words, how do I combine what is a module? Fundamentally if you ask, [01:07:00] there are multiple components. Right? That interact with each other, and they are tightly linked. In other words, right, look at this pointer.

There is a battery, there is a transmitter, there is a microcontroller, there are other components. And, if I look at the linkages, right, which component is central to this functioning of the entire device? Battery. Okay, I'll ask you a simpler question. Today all of you are aware of EV, electric vehicles, where is the innovation happening in electric vehicles?

Which part of the EV there is innovation? Battery. Battery, right? The power storage, battery chemistry, right? Can I put lithium? Can I put lithium polymer, lithium ferrite, right? Lithium manganese, right? So, so all this I can put, right? That is where innovation is happening. Yes or no? Right? Which means? Yes. One thing [01:08:00] that I am going to try to do is I will be using this Summary Service which is the latest version of Summary Service that we launched.

Is it the antenna? No, that has changed. Is it the battery? That has phenomenally changed. Is it the microcontroller? That has also drastically changed. Is there any component in your phone that actually has not changed much? Speaker. Speaker, uh? Speakers have gone through a lot of changes. Huh? Charging that charging Jack saying that has also gone through many, many.

C. If you ask me guys, I tell you, if you personally ask me, I will say, should listen, if you ask [01:09:00] me, there was one fundamental shift from 2008, right? 2008 or oh nine. I bought my first touch phone. It had a resistive display. You have to push like crazy for it to work. Right. Then they switch to capacity display after which not much has changed with capacity display.

So no, the touch screen itself, if you look at the last 15 years, not much has changed. If you actually see, of course, incrementally, something has improved. Okay, so if I am doing innovation right, if I look at the touch screen module as one level of hierarchy, not much I can do actually, right? From a technological standpoint.

If I go to battery, I can do a lot. If I go to. Let's say the microcontroller I can do a lot. If I go to the operating system, I can do a heck of a lot. Right? That's where all the thing is happening. Now this is apparent for you. Right? What if you take a product where this is not apparent? That is what do.

Understood? So, this is a semi mathematical [01:10:00] approach. Right? Slowly my battery is dying. Right? This is a semi mathematical approach developed by this professor J. N. Warfield. Right? So, I am going to teach you a method which you can do by hand. You can also go to the website and download a software. Why? The size of the matrix, right?

I am again asking you to do some matrix computation, right? The size of the matrix will be huge. 25 by 25. If you have 25 objectives. Out of your snack, snack analysis you might have, right? You might have written. For you to sit and do it is too much, right? You can actually use a, you can use a little bit of automation, right?

What is the first step? So, what you have to do? You have to define key elements, right? Use this. Terms you say to improve, let's take a phone to improve the battery life, to reduce the heating of the phone, to change the form factor, to introduce a new functionality. Fine. Like this U right now. Okay, now there is a constraint.

What is the constraint? Every element should only [01:11:00] convey one idea. Don't combine multiple things, right? Ideas, issues can be gathered from exports all through systemic analysis, right? You can use, uh, brainstorming, right? So, what is the first step? Look at your stack analysis, could have written, take that, write down, right, using this, write down the key elements.

Write down as many as you want, 25, 30, 40, it doesn't matter. You write down, try to capture. As a group, you sit down and you do that. That's the first step. What is the second thing? Second thing is, what are we trying to, look at the broad picture, what are we trying to do? You are trying to ask the question, what If I realize objective 1, let me go back, if I realize objective, you would have written, right, multiple objectives.

Does it facilitate realization of objective 2? I will give you an example. What is the objective 1? To improve battery life, sir. Ok. How do I improve battery life? If I, [01:12:00] right, just one example I gave you. Computation side, mathematical side, I am going to do something. If I do that, my battery life goes down, sorry, goes up.

Also, what happens? Heating goes down. So, two objectives it is fulfilling. Right? If I do something here, it is touching objective 1 and 2. And, if I realize objective 1, objective 2 is also fulfilled. Then, you can ask the question, which means they are tightly linked. So, you will focus your energy there. Right?

What are we broadly trying to do? Trying to innovate. Right? Trying to come up with new ideas. Right? Ways in which, if I say, how are these things linked? If I touch this, will it influence something else? Right? Second, right? So, you have to use, right, see here, you have to construct what is called the adjacency matrix.

What is this adjacency matrix? List down all the objectives. 1, Just as how you did for discovery matrix, same, same idea. Then ask the question, does 1 influence 1? Which is self referential. Okay, so we say 1 for whatever you've taken. Does realization of objective 1, does it influence, what is the [01:13:00] word here?

Don't even use realization, use the word influence.

Right? Use this word, influence. Does 1 influence 2? Right? No sir, doesn't. Put 0. Does 1 influence 3? Yes, 1. Like this you go on filling the entire matrix. First step is done. Clear? So, I repeat, how do you construct this? Out of stack analysis, you've got a lot of objectives, right? You take those and then you write, element wise you say, to improve, to reduce.

Like that you write down. You can write as many as you want, right? 30, 40, write down. Then after you've done that, right? Use the word influence and ask the question, Does objective 1, does it influence 2? Does it 1 influence 3? So on and so forth. Which means, as a team, of course, I said it is semi mathematical.

Why semi mathematical? This fellow will say. Yes, it influences. I will say no man it doesn't influence. It is a [01:14:00] bit subjective, right? But, broadly, right, irrespective of who does that, right, the inference more or less, the method will ensure you will get, more of, all of us will end up and say, sir, we have to improve the battery.

We have to improve the operating system, right? You will get to that point. This is fine, right? Here you should ensure that only one type of relationship is adopted. Here I have used the word influence, right? That's a relationship. And, transitivity is assumed, right? One element, right, E1 which means E1 influences E3, right?

So, fine, a wider relationship is transitive, may result in feedback loops, right? This, this I'll talk in the coming slide, right? So, this is fine. You have constructed the adjacency matrix. What is the next step? This alone, right, doesn't give us much. Insight. We need better insight. So we are going to do some manipulation to this Ency metrics.

What are we going to do? We are going to construct something called the reachability matrix, right? How do you [01:15:00] construct this reachability matrix, right? So how do I construct this? Take your ad efficiency matrix. Okay? What is the size of it? It is end by end, right? Depends on your problem. You may have 25 by 25, so add identity metrics to it faster.

Take your efficiency metrics right? Look at the dimension of it, right? N by N. 40 by 40, sir. That's the size of my A, okay? Then take a 40 by 40 identity matrix and add it. Okay, I've done it. Now, next, what do I do? Right? So, this is the first step. Then, you take this and you multiply it with itself.

Right? You multiply with itself. A plus I into M, A plus I. Fine, I've done that. Then what? Right? Look at the matrix, the resulting matrix, right? What is the resulting matrix? You will get something, right? A plus A, you will, you will have something? 1, 0, 1, 1, something, you will, you will get something, right? Look at this matrix, ok.[01:16:00]

Now, again multiply A plus A to this. And observe the result. There will come a point where any subsequent multiplication will not alter the matrix, right? That's the point I put here. Reachability matrix can be derived by multiplying the sum of adjacency matrix and identity matrix with itself until no further change in the reachability matrix can happen.

If you look at the reachability matrix, n I have put, a plus i to the power n I have put. What is n? Right? What is this n? Number of times you have to multiply. That will be different for different people. You may get it in three. That fellow may get it in five. I don't know this, this, we could do it in two and this four to two in this case, right?

Fine. This is clear. Yes, element making. This all element here. It can be, I've given example now, see here, self-referential is zero. Here is zero. It is one. It can be. It can be that. It cannot be. Right? I mean, the matrix answers your [01:17:00] question, right? So, fine? Sir, in that matrix, can we get more than one? Like, any of the elements more than one?

More than one as in? The value. You multiply and then check. Because, the last time we did it, one of the indices kept on increasing. Yeah, 2, 3, 4, 5. See, if you land into that problem, maybe the way you formulated, go back and revisit. Okay. Sometimes it will not converge, right? I have seen. Sir, why does that happen, sir?

Logically speaking, why does that happen? Why does it happen? I do not know, right? I am not a mathematician, right? This J. N. Warfel came up with this method. Okay. He has done some lot of incidents. If you want, I will give you his paper. He talks about in great detail. Fine? Just follow this method. I use it like a grey box modeling, right?

Just as a procedure. Fine? Clear? Next, okay sir, I have done this. Then what? Then what you have to do? This is simple, okay? You have to construct this [01:18:00] reachability set, antecedent, intersection and levels. Our objective is to find this level. What is this level? This is why I am struggling to do all the beginning.

What is the question I am asking? Sir, I have got multiple objectives. There are 25 objectives. And those 25 objectives are fulfilled by multiple components interacting with each other. Fine. Which component should I touch so that I can try, what is the relationship? I don't know. Right? It's too confusing.

Right? I'll give you a practical example in the next slide. Right? You understand? For instance, right? Let's take your cell phone, okay? If you go back, right, maybe 10 years back, right? You had monochrome display, okay? It would only, it was two G. It used two G and it used a technology called H. Right. Have you heard of this?

Right? Even though if you put it in two G band, if you go to [01:19:00] browse, it'll put E oh E for H. Okay. Right. Then 3G came, right? You have, uh, then, uh, what is this thing? 4G plus three, right near wall? No, there is something before what? I forget what it's called.

Between edge and, uh, right, this GPRS, there was another thing, I forget what it is called. Right? So, right, now, if you look at that old phone, fine, I can, I can clearly say, because the number of components are fewer, there is no graphic processor, processor does the workload. Why? Because my old, uh, I had a, have you heard of this phone, 3310?

Right? That phone, that's like Nokia's legendary model. Right? If you want an unopened thing, people will pay you 50, 000 1, 00, 000 if you have an unopened, in the box. Right? Still, if you go to eBay, you'll find unopened boxes, [01:20:00] right? People pay a lot of money to buy that. That phone, if you take, right? It's got only 2G band, right?

GPRS, right? I forget the expansion, right? It uses GPRS, and then you can, you cannot even send a multimedia message. If you, if, if it is, phone is that simple, right, then it becomes very clear. Guys, I hear some murmur. I can very clearly know which component influences which functionality. It's apparent. In a modern phone, right?

Sometimes, have you seen these rogue apps? Suddenly, your battery will start dying like anything. Have you seen that? Yeah. And you can't figure out, like, what is happening? Yesterday, my phone was fine. Today, it's, it's like, every two minutes, it's dropping 5 percent and it's heating up like crazy. It's crazy.

You can't figure out right? You go check, and then you find out, this is taking memory, there is some rogue app, right? And then maybe you restart your phone, still doesn't help, then you go back, uninstall a few apps, you have to figure out right? Is there a method to the madness? That's the question I'm asking.

Sometimes you don't know right? So here, [01:21:00] right? If, since we have many many elements, or in other words many many, Components for a physical product and those components make up a subsystem, and those subsystems fulfill multiple objectives, right? How are these linked? That insight is what I'm going to see.

Get fine. So look at one, one here, one list. Uh, I cannot say, say, I can say right one somehow. Inferences. 1, 3, 4. So put that 1, 3, 4, 2, 1, 1, 1. Everywhere. Put 1, 2, 3, 4. Right? So you fill the reachability set. Fine? Look at 1's right? This you have done. Next, antecedent, right? We went row wise, now let's go column wise, right?

Look at 1, yeah, that's influencing all. So 1, 2, 3, 4, right? 2 influences only 2, right? And 3 influences all, right? So you put these two. Intersection A union B, right? So you do that, you get this. Right? Not A union B, A intersection [01:22:00] B. Right? It was not A. Right? So you take the intersection of these two. Right. And then you end up with this.

Now you see a pattern here, guys. This is what we've been struggling for. You see a pattern here. What is the pattern? I see 1, 3, 4, 1, 3, 4, 1, 3, 4 is one. Pattern two, all one is like an orphan. It is saying separately. So I say these are at one level, and this is at one level. So when I say level one, it doesn't mean it is important.

All I'm saying is it is at one level. This is at another level, right? So you can graphically represent it like this. If I touch 1, it influences itself, right? 3, 4, and 2 doesn't get affected. You understand? So, this gives me a clear thing. If I touch this objective, right? If I touch objective 2, everything else is getting influenced.

If I touch objective 1, 3 and 4 only is getting influenced. 2 is not getting influenced. So I can have a [01:23:00] better insight. So, ISM. gives me the functional hierarchy of all the objectives that I have. This is clear? Now I can do the same thing I did for, uh, discovery matrix, right? D plus R, D minus R. Right. You can take the in degree out degree, right?

And then plot it like this, right? So this is an example, right? Only this one example I, I'll quickly cover because I, I still have to cover, okay? I still have one and a half hours. Let me quickly cover this. Okay? So this is from, okay, this is like, uh, this story is probably 15 years old. What you see is 15 years old, okay?

15 years old. There is no int banking, there is no pay dm. There is no bim, nothing. Okay? This is from 2006, seven. That being the case, banks are asking the question, right? How can we, one, get more, for a bank, somebody has to come and deposit [01:24:00] money. I have 10 customers here, they are rich customers, they will put a lot of money.

That is one strategy. A bank like Standard Chartered, how many of, how many of you have an account with Standard Chartered? Not many. Indian Bank? Many. SBI? Many. Right? So, Nations Banker. That's what LPSA, right? Nations Banker. You go to any corner of the country, you will have State Bank of India. Right? You go to, right, at least in South, you go anywhere, Indian Bank will be there.

Right? So, a bank will have a certain set of customers. Right? So, what is the fundamental thing? Either, what I have to do, I have to give more service to the customer, so he will invest more money. That is one strategy. Or, I have to expand my customer base. Two strategies I can follow. Fine? Okay. Third thing is I should retain my customer.

This fellow should not suddenly say, uh, access back is good, let me retain. First I should retain, no? All this, I have to do, right? So one is retention. Customer Retention. I have to retain my customer. Have to keep them happy. Give them more functionality, and keep adding more customers to [01:25:00] my service portfolio, right?

That's the question, right? Let's say, for example, ICA is asking this question, okay? So, functional architecture, ok. So, then they, then they understand. People, right, when they say digital payments, they mean your credit card, right? Your, basically your car, car payment, right? So, right? So, expectations and payment experience.

For a customer, right, what is called expected behavior? This is expected behavior. My company has done this for over a decade. This is one of the best guys! He's the father of golf. That is amazing. What do you think is the fashion? What is the fashion of golf? Sorry about that. What do you think is the fashion of golf?

Time for [01:26:00] a question. payment. Whatever. Right. The infra, the payment methodology or the technique I'm, I'm giving you right, the digital payment, right. Experience I provide you, but there are some negative attributes associated with it. What are that? Loss of privacy. The moment I swipe, they've got my credit card number.

They know who I am, they know what I bought, where I bought all the data is going, so I may not be comfortable. Payment cycle. Right. Then lack of control on spending. What happens sometimes, right? It just keeps swiping. Right? You go to Amazon, Diwali, right? Vega Sale. What you do? You keep buying stuff. Right?

It's also on a credit card. Why? I don't, I'm spending money I don't have. I'm borrowing from my future. Right? Oh, there is this fantastic watchman. Okay, Carmen. 20, 000. Okay, let me buy it. There's an offer. 2, 000 rupees offer. 20, 000 rupees watch. Right? So, there is lack of control on spending. Right? And, lack of real time information.

Lack of card acceptance, right? So I have visa, maybe this guy is declaring. I have got [01:27:00] master card, maybe this guy is declaring. Then inclusion, right? Is the service available to everybody, inclusively? Right? So you can look at mobile complimented features, right? So you can say, right, I can bring in real time info, interaction, location intelligence, computing power, right?

So, here finally what they found out is, right, after doing all this functional decomposition, right, ISM, is it? Okay. So, not everything can be addressed. You can see the linkages, right, each. They say you can use a mobile phone, right, to address the fundamental limitations of cards and reduce the use of cash.

Why, why do you want to use, why do you want to reduce the use of cash? There is a sinister reason. What is the, cash gives you some anonymity. I go and pay a hundred rupees. There is a business transaction. Who paid? I don't know. What did he buy? I don't know. I just sold this. Who was the customer? I don't know.

I buy it with my card. [01:28:00] Who paid? I know. What is his age? I know. What is his earning potential? I know. Right? Got it. Right? That is where your G pay, right? This fellow pretty much gets all the things. Right? Banks, they really love to get rid of hard cash. They want to get rid of cash altogether. Because cash is your only potential.

Right. I can put black money. I can go buy a house. Nobody will know. RBI does not know. Right. Income tax requirement does not know. I want traceability. Right. Any illegal activity. Right. I want to stop that. Right. If it is a digital payment, I can track, I can trace it back to who is the perpetrator. Right. So, finally what I found out is, right, a mobile based implementation would overcome some of the problems with card.

Card theft. Card is stolen, sir. Somebody is now swiping. So, let us get to today's class. So, I will quickly run through lecture 10. Fine? Guys, ISM is clear [01:29:00] now? Right? Now what we are going to do, we are going to tweak this about ISM. What does it tell me? It tells me some, right? If I touch this, I mean if I raise this objective, this happens, and this component, if I touch this happens.

That or not? Guys? Right? That insight it has given me. Now what I am going to do, okay, with this new found knowledge, right, I am going to, right, come up with some creative ideas of actually making my product better. How do I do that? Right? Let us look at that. Fine? This we already explained. So I do not have to go back.

Fine? So what do you have to do? Okay. There are three steps you have to follow. You have to do what is called, you are going to generate a structural design. What is a structural design? Right? I will explain as we go forward. Take [01:30:00] functions with, I told you right, when you constructed the dot, you can look at d plus r, d minus r, all that you have.

Now, you take functions with high d minus r and low d plus r. Put together, right? Then similarly d minus r close to zero, high d plus r. Then functions with the negativity minus R and low. D plus R then functions with the D minus R and D plus R, close to zero, right? Like this, you make you group all these functions.

Fine. Guys, listen, this chart is very . If you don't follow you, you will not. If you don't listen to me, you cannot follow. Now you are going to create what are called multiple concepts or multiple options. Remember the example I gave you, right? If I am building a website which allows me to book a ticket, what is option 1?

Should I even build a website? Guys listen, so the first question I am asking is, right? I need to book a train ticket, how can [01:31:00] I do that? What is the conventional method? Go to the counter, pay money, fill a form, you will get a ticket, fine? How else can I do it? Internet is there. Go to the computer. Log into your, uh, Chrome.

Go to the website book. Or install an app. Go. Fine. You have three options. Now, how can you innovate? Right? You need to come with J options. What is J? Can be anything. 10, 20, 30. The more crazy you go, right, is when you come out of a paradigm. Right? So, I have to talk about something called Guys, listen. Right?

Very, very important principle, right? For CS guys, this is extremely important. There is something called, there is a guy called Kuhn, K U H N, Kuhn, ok? He gives you, guys, I hear a very prominent voice multiple times I'm telling you, I don't know what the hell you are talking, right? Another one and a half hours, you give me [01:32:00] your attention, right?

Then we'll be done for the semester. So, this guy called Kuhn, K U H N. He comes up with a theory, okay. He came up with a theory. What is it called? Kuhn's theory of paradigm. What does it say? He says, okay, what example I have given you, right? Maybe I will show you a picture and give you a better idea. Right?

So,

Everybody likes this, right? Right? All boys, if I ask them. Ninja, right? Kawasaki Ninja. It's like probably your dream bike. Right? Will you be okay, I'm having this?

He is laughing at it. See, I showed him [01:33:00] this. Why is he laughing at this? It looks cool, but it also looks weird. Out of place. Exactly. That's the word I was looking for. It looks out of place. Why does it look out of place?

Right? This is an important principle, right? If I show this, everybody in the class is happy, right? And so, no, you really like it now, but if I show this right, and if I give you option A and B and say you can take anything, how many of you'll go with option A, raise your, you'll go with. So, who wants option B, the second one, you tell me.

Why? Sir, actually we can try something new. You can actually? Try something new. Try something new, ah? See, let us say this is your mode of conveyance.

Guys, guys, no, no, I am trying to make a point here, ok? See, this is the problem, no? I try to kidney [01:34:00] your brain and you go somewhere. Guys, listen! Right? So, in the first, this is called, right, Kuhn, right, Thomas Kuhn. What he says is, guys understand, this is a very important principle, right, if you want to complete a charge, chargeability, stick, stick with what I am saying, right, it is very important.

Chargeability cannot do this, it cannot do this, right, I am, I am giving you a secret, I am giving you a secret sauce, right. This is one paradigm, right, what is this paradigm, sir, this new word is very confusing, right, what do you mean by paradigm? What I mean is, You build a bike with two tires, right? And you sit on it.

You hold on to the handlebar like this. There is a brake which you press with your foot and there is an accelerator, fine? And you sit in this upright position, right? Or maybe you have to lean down a bit. How did this come about? Any idea? How did this come about? It's arbitrary. [01:35:00] Please understand, there is no reasoning behind it.

Right? So, this we call it as a arbitrary paradigm. So, if you are in this paradigm, right? If you want to improve in this paradigm, what can you do? Huh? You keep changing within the constraints of that paradigm. Snack. S N A C. Remember. With this paradigm, I've got a constraint. What is the maximum I can accelerate?

The engine is here, centre of gravity is here, right? If I, if I accelerate more than 2G, right, G is what the acceleration due to gravity, the vehicle will topple. If I have to go faster, what do I do? Actually I have to do this. This is a crazy looking bike. This is a crazy looking bike. Why? Look at this feet, where is it?

It is called feet forward position, right? Let's see, like this. Right? Here, how do you keep? You keep it like this. Feet backward. Right? Here, feet forward. So, this is one paradigm. This is another [01:36:00] paradigm. If you have to innovate, you have, you can, there are two things you can do. Stick, sit within this paradigm and do all optimization.

What did I say in the beginning of the class today? Engineers are worried about how? How do I make this go faster? How do I improve the handling? Designer does what? They will come out of this paradigm and say, why am I sitting like this? Right? Can I sit like this? Right? That is the question he is asking.

So they are asking the question, what am I doing? Why am I doing? Right? So you challenge the narrative. There is this guy, I forgot his name, he has written a book, I will give you one line summary, he says. If you have to, if you have to innovate, if you have to make a lot of money, you have to bet on a controversial idea, he [01:37:00] says, right?

If you have to win, if you have to innovate, if you have to break the norm, if you have to stand out of the crowd, if you have to be someone like Steve Jobs, you have to bet on a controversial idea. Why a controversial idea? Yeah. A phone should not look like this. It should not have a display. That's a controversial idea.

But will it succeed mostly? No. There is a reason why it is called a controversial idea. It challenges as narrative. You don't find a lot of bikes like this, yes or no. Right? This is a controversial idea because we started with this. It's an arbitrary paradigm. If we had started with this, what would've happened?

Everybody will say, this is the norm, sir. Right? So you have a. Watch and watch goes like this and what do we call that? Clockwise. I have, if you want, you come to my room. I have a clock with a mirror image of this. In that clock, clockwise is this way. [01:38:00] You understand? The entire dial is backwards. I often ask students to read the time.

They will be messed up for a couple of minutes. Right? If you are interested, you come to my cabin, I will show you. Right? So, what is clockwise? What is clockwise? It's arbitrary. Somebody said, you put a watch face like this, 1, this direction, right? And that is clockwise. Please understand that it's arbitrary.

I could have, in my world, I could have put a watch this way and call that clockwise. That's also arbitrary, right? So, this is the key tenet of Kudsi theorem paradigm. All paradigms are arbitrary. And if you want to innovate, come out of the paradigm, because within a paradigm, there is only so much you can do.

Understood? Right? This is the generic design principle. Right? So just to illustrate this, right? Let me finish this slide.

Right? All of you know this handle, right? We are all familiar with [01:39:00] this. What about this? There is also a handle design like this. Look at what he is doing. This is a 20 year old design. Right? Lab, I work with a lot of mechanisms like this. It's called a complaint mechanism. But you look at this. There are no moving components, right?

Everything deforms. Right? So this is another paradigm. This is how you innovate. How do you come up with this? How do you come up with this? It's not easy. It's not straightforward. So you ask that question. Sir, I have to put oil. That's stupid thing. Keep, jams. Okay, jamming issue. Write down. Okay, next. I have to maintain.

Okay. Lot of parts rub. They wear down. How do I do this? I don't have an answer. But, I've got ISM I can do. I know the relationship. Then I sit. I talk to domain experts. I look at evolving areas. I say, dude, is there something, uh, new? Yeah. I've got this new thing. It's called a complaint mechanism. I think we can put it here and solve this.

So you go talk to a domain expert, somebody who is [01:40:00] doing something differently. This is actually bio inspired. This is actually bio inspired, right? So, fine? Some of you, how many of you are from Designers Club? Right? Only two, three, right? So generally these, right, as part of that I talk about there. Uh huh.

Uh huh.

Semester. One second semester. Yeah. Session. No Orient. Oh yeah. Second time I came. First time go.

Fine. So [01:41:00] why am I, why did I go on a long window tour on theory of paradigms and all that story? You need to create multiple. Now you now does it open your mind? Right? Don't be constrained with one saying, Sir, website means it should be like this. There should be username, password. This is how I've been doing.

And copy paste the same thing on a mobile app. Okay? That is one paradigm. Come out of that paradigm. Can you think of something different? That is what we mean by option 2, 3, 4, up to G. Come up with wild ideas. Right? Ideas that might not even work. Right? That might not even work. Right? You come up with all these.

Right. Crazy ideas, right? That's why we put Option J can go 2025, right? Right. Now, after you've done that, guys understand this is the first step, right? Option one, right? This, right Fulfills these functions, right Functions with high D minus R, low D plus R. This fulfills like that. You [01:42:00] go on filling the entire.

Right? Now, each column should be filled taking needs, preferences of a key stakeholder, right? Customer, enterprise, regulatory body, so on and so forth. Right? So, pick the appropriate technology for the critical functions. What is the most critical function? Right? So, you fill up. Next, what to do, sir? Ok, I have done that.

This, I am doing it at a functional level. I am looking at a multiple concepts. Now, I am going down at the component level, right? I am saying function 1 is fulfilled by component 1 also, 2 also, 3 up to component j. Function 2 is not fulfilled by component 1, 2, so on and so forth, right? I will give you an example.

What is function 1? Let us go back to a cell phone, right? To make a phone call, what is component 1? Battery. Yeah, battery has to work. To make a phone call, should the display work? Yes, it has to work. What about the antenna? Yes, it has to work. Right? You can go on filling. Next function, sir. What is the next function?[01:43:00]

Just name a function. Cell phone has got multiple functions. Name one function where you don't need a battery. There is no function like that. Pretty much you need battery for everything. Is there a function where some component doesn't participate? If you make a phone call, camera is not needed. Right? If you are making a phone call, for phone call component one, camera is not needed.

Battery is needed. Processor is needed. So on and so forth. Probably my GPU is sleeping, because I am making a phone call. GPU does not need to work. So, you go on filling all the functions with all the components of your product. So, once the high level product target picture is defined, which is this. You have to use the matrix to date the other components in the structure.

So, you have to create a detailed component list. Now, we come to the last part of this. What is this? This is called design structure matrix or product architecture. Repeat after me. Design structure matrix. What is this design structure matrix? [01:44:00] Here I have listed down functions and components. Now you ask the question, what about the relationship between components themselves?

Right? Take component 1, that is it. Influence component 1, yes. Does component influence component 2? No. Like that you can go on looking at it. In other words, How tightly is battery linked to your processor? How tightly is your battery linked to your display, so on and so forth. Once you do this right, fine.

I'll explain more in the coming slides. All of you have seen this. You're familiar with this picture. How many of you, Sarah, have never seen this? Is there anyone in the class like that? You've never seen this? Sir. What is this? I put the name right. This is the schematic representation. Right? Guys, since all of you are not familiar, listen to me.

What does a jet engine do? Right? It's simple. If you have to move a plane, you have to pull the plane. [01:45:00] I can tie a long rope to the plane. If I just drag the rope, what will happen? It will move and start flying. It's that simple. And something has to pull that rope. Right? If you have some big giant and then that fellow, you see it right?

How do you fly a kite, man? How do you fly a kite? You just pull and run, na? That's what you do, right? If you want to fly a kite, what do you do? Does a kite have an engine? No. Right? So, if there is no wind, if there is no wind, what do you do? You pull this string and you start running. That's how a kite starts flying.

So, every plane flies like that. Only thing is instead of a string, some guy or some white kid running, you have a jet engine which does the pulling. That's it. Now, what this engine has to do, it has to just take in air and accelerate at the backward, right? Your ance third lock, right? It pushes air out. That pushes the engine and that drags the brain.

Find clear now to do that. Okay? All these bears you see. [01:46:00] Are simply the intake fan. It's the compressor section. Takes the air, compresses the air, right, burns the fuel, and it runs the turbine and exits fine. What is a turbine? Your windmill. What does it do? If wind blows it turns right. Same thing. So wind blows.

This turns, this is connected to the intake. So this fan turns. What does the fan to that sucks in? And the cycle continues, fine. So what does it do? Initially there is a motor, external motor that will spin the engine. Ok, there is a battery and that spins the engine. It is called a starter motor. That starts the engine.

So what? It starts spinning, right? Once all these plates, right, these are called plates, you have multistage fans. These are multistage compressors actually. I should not call them fans. These are multistage compressors, right? It is called axial flow. This is an axial flow turbine, axial flow compressor. So axially, along the axis, Add flows, and as the add flows, you can see the cross section areas strings, right?

[01:47:00] So air gets compressed and after the air gets compressed, this is the combustion chamber, right? Your fuel ke is injected and burns and the burn gas is hot, it wants to expand, it expands and goes up quickly. What is it? Turns the turbine. Turbine moves the compressor. This is clear, right? Very easy, right? So what have I done?

I've taken a fan and this fellow, and I've connected it to her. Wind turbine buying. There is no natural wind. So what am I doing? I am burning fuel to generate heat, which expands fuel. That's it. I've taken a fan under kind of conceptually concept. Conceptually. What about the embodiment there? This is a concept diagram, somebody right?

Frank Whittle, uh, is the guy who came up with this idea. Okay? So a British inventor came up with this idea. again, it is disputed. There is a German guy he says, I invented again, you know, British is always not like Newton. They gave him credit. Right? So there are a couple of guys who it's like Live and Newton, right?

Why? There was actually a South Indian mathematician who came up with the [01:48:00] concept of re. Many of you don't know, right? Our people have come up with the concept of infinite, infinite as well, just that British rule. They set their wrong. You are all your, uh, calculus is credited to not even liveness, only to Newan because who was the, I think president of Royal Society, Euan himself, right?

He was the chairman. He said, my work is the greatest. It's like award himself and how we say Newtonian physics on all the credit goes right. You should understand this, right? This is fine. This is a conceptual idea. Fine guys, understand now. Fine. What about the embodiment? I told about embodiment, right?

This is the embodiment. Can you understand anything? Right? This is easy to follow, right? Nicely blue, orange, right? But the actual thing, if I showed you that, do you understand anything? It is actually still simple, ok? This is a high bypass fan, ok? This is a turbofan engine, right? So you see these [01:49:00] blades, right?

And, but you see lot of other components. Right? If you can see some similarity, right? Yes or no? You can see the overall skeleton, but it's actually a lot more. You don't see these tubes, right? All these cooling tubes that goes, right? Why am I, why I'm telling you this is, between a conceptual idea, right, between a concept and an embodiment, this is the difference.

And look at the level of detail this requires. There are probably 10, 000 engineers who worked on this. Probably more than that, right? There are only three companies in the world which can make this, Prath and Mitni. Rolls Royce, General Electric, that's it. Airbus buys from Rolls Royce, right? Boeing buys it from their own thing, GE or sometimes they buy from Rolls Royce.

There is Pratt and Metina, another American company, right? So why? Because it is so technologically complex. China cannot make it even. They can make planes, but they cannot make engines. They are slowly learning how to make. Okay? HAL, right? DRDO, they had a [01:50:00] joint project called Kaveri Engine. read about it.

It's called Kaveh region. It is a 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 12, 13, 13, 15, 16, 16. It is a 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 14, 13, 15. Oh[01:51:00] [01:52:00] [01:53:00] [01:54:00] [01:55:00]

Listening test, English, listening, listening test. Sir, [01:56:00] can[01:57:00]

you switch off the A. C.? I don't know, Mani, but I feel terrible. So cold. So cold. So cold. Yeah, I don't know where the sun hole is. It's so cold. It's so cold. It's so cold. You won't be.

You won't be. Right now also I am not buying it. But, it's number [01:58:00] doesn't show. It's very It's very It's very I don't know. Alright. Take it off. It's not worth it.

Good morning, you can come right now. I'll be here in just a minute. Can you help me with the laptop? Because I have some questions. I can't hear. First, watch the video this morning. The class will be quite long. I want to make clear. You are in your room. Don't worry. Students, listen. Settle down. We have one hour.

Let me, let me wrap up. Settle down fast. [01:59:00] Guys, settle down. What if you come after the second break to class, you will have a free attendance? It's for audio, anyways I have to come. So you might as well So, we just saw, right, the example of this engine. Now you think, now if I ask this question like a cell phone, right, does [02:00:00] battery talk to this?

You cannot ask that question, right? It is pretty, pretty difficult, right, for such a complex system. Look at the design structure when we start, right? It is insane, right? Look at the number of, this is just part of it from the book, right? So, please check out this video right, Real Engineering, if you just go Google this, right.

It will actually show you the complete right, breakdown of an engine if you are interested, right. So, here right, what is the important thing? For instance, right, let us look at aerofoil, right, does it influence fan containment unit? No. Does aerofoil influence, right, I cannot read it so bad here. Shroud, right, does it influence?

Yes, right, does the stator influence? So, what we have to do is, we have to look at the, we have [02:01:00] to look at the, what we have Test it out.

Right. What this method shows you is just, if I look at it, what do you see? You see all these clusters, guys listen, you see all these clusters. What does it tell you? It tells you components that have lot of interconnectedness, right? Once you see that right? What is the difference, between the normal and the low level information?

If you want to have wide range of information, there is a, there is a limitation on what Let me tell [02:02:00] you the definition of wide range of information you can do, which is, an array of information. There is a built in GPU, some light gaming graphics you can play. But for any heavy lifting, you have to go to a dedicated GPU, right?

That you know. Your phone, right, everything is on a tupper, they call it a SOC, systems on a chip, right? SOC. Now, there because the communication lines are very close together, they can talk to each other very fast. If they are in two separate things physically, right, you have to have some copper trace, right, that connects these two.

So, if I look at it something like this, what it tells me is part consolidation. What is part consolidation? Say there is this screw and there is a washer. Can I combine this into one component? Right? So, if you look at, uh, some of you may have [02:03:00] water bottles with flip, flip cap. Anybody has that? That one, yeah, that green one, right?

If you look at that. Right? So if I am working on something like this, what do you do?

Right? Does it have a hinge? Look at the door. Right? You look at the door. All of you have seen the hinge on a door, right? If I open this, right? There are how many? 1, hinges. Does it have a hinge? No, sir. What is a hinge there? It is the plastic part. You will have two protrusion like this. Sir, but there is no hinge as such, right?

It is only, it is, it is, it is a gap in the There is a hinge. What is a hinge? It is just a pin, correct? It is just a pin. You have a water bottle. Just take it out. Right? This one, man. Right in front of you. You are right. Yeah, open that. Right? In fact, if I, [02:04:00] if I break, right? If I break it, I have to get you a new water bottle.

Okay. I'll get you a new one.

Look at, look at this. How old is it, man? Sacrifice. One year, sir. One year. I'll get you a new one. You get back and I'll buy you one. Okay. It's a sacrifice. Look, look at this small protrusion you see. No, every time, no, this is not tupperware, this is Milton, no. Tupperware is flexible enough, no. I just told him I break it, bro.

You see this short of protrusion here, guys?

Guys? Look at the protrusion. We are talking about path conformation, right? Now, there is a dedicated hinge. Right? There is a dedicated hinge. That is a [02:05:00] separate component. Right? This pin, how can you achieve that? Sir, my point is, in this, in this if you have something like, there'll be some attachment to it.

It won't be like a whole hinge. There's not a hinge that's moving like this. What you're talking is a live hinge. It's like a pencil box. In fact, why do you want to go that far, right? Look at this, uh, Right? What you're saying is something like this? Yeah. It's not something that's happening. Guys, compose yourself.

It's not that funny, man.

Right. So a meat for the land, right? This is a separate component. If I can come up with a design map, this is part of this, why is this happening to, if I can come up with [02:06:00] a separate design where I take this and integrate it as part of this, right? Then I've done what is called part consolidation. Okay. Why is this important?

When you have in my, in a, in a DSM, you can look at this matrix and see, okay, there is a tight integration here. These components are, these components are there. Can I put them, put them together as one module? Can I put them as one subsystem? Why is it important? Fast communication. Right? If you look at watches, right, what is called movement?

What is movement? See I have, uh, Casio, right? It uses the Casio movement. Casio makes Casio movement, right? Casio also sells movement to other companies, right? So, they also sell the movement to other companies. Some other manufacturer will buy from Casio. They will integrate it, put a different dial case, right?

Gold plating, whatever, right? Then they'll sell it in the market. How is this [02:07:00] possible? It is possible because of modularization, right? Look at most of your phones. Who makes the camera module for the phone? Sony. Sony makes. Oppo buys from Sony. OnePlus buys from Sony. Or probably Carl Zeiss. He makes the lens, right?

He applies to everyone else, right? So the point is, when I modelize, right, when I consolidate path consolidation, you have 10 components. Can I make it nine? Can I make it seven? Let's look at from a coding perspective, right? Mm-Hmm. How do you modernize of code? Your code has to be flexible enough to run on any platform template.

Whether they put it on a arm, whether they put it on a Intel or X 86 architecture, right? That's what, that's what, that's what they say anywhere. Right. So, wherever I put it, right, it should be modular enough to run. Today it's not possible, right? ARM is picking up, right? In mobile, [02:08:00] they're really, they're pretty strong.

Even in standard PC, right? But what is the problem? Windows doesn't run well. It's not optimized for. It's, it's, since time immemorial, right? It's designed for what? Intel and AMD. Right. X86 architecture, right. That's what it's designed for. Now, if my code can be more generic, it can be platform agnostic, we say, platform independent, then I have a robust code, right.

From a CS perspective, right, that is the fundamental challenge, right. There are applications I have, right, which will only run in Windows 2000, it will not even run on XP. If you go visit any factory, right, old factory, they will have machinery which are 20 years old, which is made in 95. What computer did you have in 95?

Windows 95. Right? If I have that, if I take a modern computer, I can buy a modern computer, not a problem. But what is the software that runs the computer? It runs on Fortran. How many of Fortran? Guys, how many of Fortran? F O R T R [02:09:00] A N. It's a program in COBOL, BASIC. BASIC you know, you might have heard. Right?

Fortran is, Fortran used to be like the king. I'm talking 35 40 years back, before C. Before C, Fortran used to be BASIC and Fortran. Right? Fortran used to be, like the defacto standard if you want to write a piece of code, Especially if you want a piece of code that runs on an Embedded system Right? Guys right?

On an Embedded system So, if i have a Fortran Code, probably it is meant to run in Windows 95 If you put it on Windows 7 or 8 or 10 it will not run Because why? Software Incompatibilities Right so this tells you What all the component that are there How tightly they are What is the difference between, between, between, between the transcripts?

What is the difference between the transcripts? Yes you can. So, you do not need to carry. It will be, it will [02:10:00] be a thing you have to, to, to, to, to, to, to, to, to reinforce on the site of the code. That's beautiful right if I can do that But there is a problem with that. What is that? What is the problem with that?

Anyone? If I can have one component, rather, one phone, there is only one component. That will be ideal, right, for a manufacturer? If one thing goes wrong, the entire thing you have to change. You don't have redundancy. Right? What you are saying is called redundancy. Right? For mission critical performance.

Right? You have what is called redundancy. What is redundancy? If this one component goes, ask the reverse question. This function is done by, right? We are looking at it that way. Look at it column wise. If this component goes, how many functions will it take it out? Right? In other words, if your battery dies, phone is gone.

Right? So, how do we solve this? Have a backup. [02:11:00] Right? So, in aerospace industry, this is very important. Right? You have what is called fly by wire. What is fly by wire? I will give you a simple example. You apply a brake, all of you know to ride a bicycle? Yes. You apply a brake, somewhere you can feel the brake grab the wheels, right?

In winter have you felt it? It will just slip and it will not stop, right? You have what is called feedback. Okay. Now think about it this way. Can I have a motor there that pushes the brake shoe and push, keep a button here? If I press the button it brakes. Will that not work? It work, right? So what have I done?

I have a cable that connects my input control. If I press a lever, that's a cable connecting my breakthrough that stops the by second. Now what is the problem? I have to apply all the force right now. Look at a plane. World War I [02:12:00] right? Just. At that one, like 1914 to 18, right. Plane. When did they, right? In what?

1903, right? Three or five, right? Exactly. Right. So you, the planes are not even 15, 20 years old and they're fighting with planes. Those planes, if you have to move the control systems, right? All the wings, everything. Oh boy. Physically the pilot's out of huge muscles if you have to break, right? It's not easy, right?

What if I just give you a button? You press the button and it breaks. That's beautiful. Right? Why are we not doing it? Sensitivity. Sir, we don't know how much you want to break. Okay. Say, I'll build that into your system. I'll give you a feedback. I'll put an encoder. And I'll put a motor here. So it precisely emulates how much force you're applying.

I'm, I, I will convince you you're haptic feedback. In other words, look at your cell phone. When you type, it vibrates, right? A, B, C. Yes, yes, [02:13:00] yes. That gives you feedback that you have touched. It is not as good as a keyboard, right? That tactile feedback is not there, haptic feedback is not there. But when I touch the screen it vibrates, right?

Yes or no? Now that is called feedback. Let us say I give you the feedback. Now won't you use it? You will use it, right? What is the problem with that system? It needs electricity, right? What took a mechanical cable? Now what have I done? I put a encoder here, I put a motor here, I put another motor there.

There has to be a communication channel. I need a wire. And if some rat comes, hits the cables and what happens? What will you do? You will do this, nothing will happen. Right? So this is called fly by wire. There is a wire. I am flying my plane with what? Right? It is called a yoke. What the pilot pulls, right?

In movies and all you have seen, right? What the pilot moves is a encoder. It is like your mouse wheel. That is what the pilot is moving. What is your [02:14:00] pilot moving? This is what he is doing. That is an optical encoder. So what your pilot is moving, it is like a joystick. How is it connected to the control surfaces?

There are hydraulic cylinders in it. They are like mota mota, right? There is a separate power pack. What talks to these two? What is the mode of communication? There are wires. You understand? Now, that wire is also a component. That can fail. If it fails, what happens to the functionality? Sir, I can no longer fly the plane.

What happens? Gone, right? You'll crash the plane. So, these kind of systems, right? You have redundancy. There are dual backup. There are two encoders. There are two separate wires. So, there is a good chance my, the rat, even if it comes, It will not chew through both the wires. First I will make it rat proof.

Even if it does end up eating one wire, there is a secondary channel it will switch. Right? I will give you a software example, sir. Again you are talking, uh, this I understand, but can you give me a [02:15:00] software example? I will give you a software example. Let us look at, right, robotic assisted surgery. What is robotic assisted surgery, sir?

I am using a robot to do surgery. Now, every robot is controlled by some operating system. That has to be something. Right? So I'm doing some surgery. Right? It's happening on the patient. Have I shown you any video of it? No? Right? I'll, I'll show you a video. Right? Otherwise, uh, you'll have no relevance. Sir, is it that small change, big thing?

Big change thing? What small change, big thing? Butterfly effect, huh? Uh, no, sir. Your, your research video, right? Like you will, you will do a change here and there it will do.

Small change, big thing, what is the way to describe that? See, I will just go, look at, look at here. See, this is what is called teleoperation, okay. So, instead of the doctor directly working on the patient, right, [02:16:00] here I am actually sitting and I have got this control set, you can see these black things. When I move, the robot moves actually.

Okay. So, it is here, see look at my hands, right, as I am moving this, this is a 10 year old video, ok. See, you can see it move, right, so there is an amplified view, ok, this is the actual scale of it, right, you can see, right, it is itna bada, right, this small, what you are seeing is like this small, right, tip of my finger, and this is a very magnified view, you can see, right, fine.

So, the important thing is, how am I controlling the robot? I move something in space here. It's like a mouse, I'm moving a mouse in 3D space. How is that moving? There are a lot of sensors that read my hand position. And it says, this fellow is moving his hand here in space, spatially, [02:17:00] right? Your direction cosines and x, y, z, in Euclidean space.

Six coordinates it's tracking, right? My three positions and three direction cosines. That is Somebody get a A robot, right? Now there is some software running, right, that's talking to these and keeping them in sync. If that crashes, what happens? Windows, blue screen of death if I get what happens to that?

You understand? It cannot happen, right? Right? So first of all, it doesn't use Windows, right? Linux makes this. Linux, or what else makes this, not what, you, what is the joke? About two. He makes a very nice joke about the Windows, right? Okay. So, Windows first of all, right? The very first thing, I uninstalled Windows.

Right? What was working will not work. Now, in such a case, right? I will build, right? Guys, listen. I will build software that I rigorously test. And there is always that edge case scenario. It can fail, right? Nobody can guarantee. So, what I do [02:18:00] is, Guys, listen. I will have two hardware, which runs both. I will have a duplication.

Whatever algorithm this is running, I am running it in parallel. If this crashes, what happens? Immediately the system will detect and it will hand over to this fellow. So when the doctor is working, sometimes it will crash. The doctor is not even aware. Why? There is a secondary backup in place. There is a redundancy in place.

Understood? Right? So when you do part consolidation, this problem is also prevalent. It will fail. If it fails, it should fail safe. Ok, this pointer goes bad, what happens? Nothing. No problem. I will keep it at that. Right. I can still work my car's brake doesn't work. What do you do? Right? Those of you're interested, right?

Quickly and explain, okay, your car has got what is called four channel a s, right? I you if your main break doesn't have what you do in this crash, so generally how the connected is this wheel and this wheel, they are connected. Okay? [02:19:00] This wheel and this wheel, they are connected, right? So in sort of a X right architecture.

Now, these two lines, right, are connected to one, what is called a master cylinder. Okay? So, there will be two. For whatever reason, right, this line gets cut, right? Our rad friend comes, right? Choose through the cable. What happens to the host? Pressure, Pascal's law, you know. Pressure applied at a point is equally applied everywhere.

What will happen? If there is a pressure loss here? Pressure loss will be everywhere. Now, in this case, what have I done? I have isolated the circuit. These two wheels won't work. What? These two wheels run on a separate line, right? Yes, right? So there is a redundancy. So my system will run at half the capacity.

It will break, but it will not break 100%. At least I have some break rather than nothing. You understand? So any system that you build, depending on the criticality of [02:20:00] the application, you will actually build a redundancy. Coding. This is very, very important, right? Uh, surgical devices, I cannot emphasize enough, right?

Robustness, right? There are, there are, uh, uh, a standards for this. There is, uh, not, uh, not only SA what is the other? Uh, uh, not IES There is international standards, right? Forget the, uh, uh, right. There are codes that are for medical devices. There is a separate regulatory body that actually, that classifies software that brings it right.

What is called regulation? What is regulation? FSSAI. Have you seen? Food? Agmark? Agmark they have removed. Now it's all FSSAI. Right? Government comes and check. OK. Are you doing it properly? OK. Take the regulatory approval. Right? You are doing it in certain practices. Right? But, for a software. Right? Suddenly I hack the system.

This system, I can hack the system. Right? If this system gets hacked, what happens? [02:21:00] See the, the first guy who sent a virus, right, read about that story, it's interesting. He was a hobbyist. There was no, this concept of virus was not prevalent. He sent a virus, right, it caused a lot of damage, right? I think it was his brother or friend, right, who inadvertently gave away his name.

And then he got arrested, right? He was the first year to be convicted for cybercrime. Right? He thought he was fooling around. If you're fooling around this, these are wireless systems also you can, so you can easily hack, right? Wi Fi, you know how easy it is to hack, right? You have this, what, WPA2, right? You have all this encryption, right?

But it's, any encryption can be broken, right? Given enough effort. So, you have to build in all these. The point I'm trying to emphasize is for software especially, that runs, the car has got 50 to 70 microcontrollers. How many? All of you have seen an Arduino? How many of you have heard of Arduino? Raspberry Pi?

Raspberry Pi, Arduino, right? That is one microcontroller. A [02:22:00] typical car has how many? Anywhere from 50 to 70. The length of wiring, you have any idea how much it has? 5 kilometers. The average length of wiring in a car is about 5 kilometers. Why? So many things are happening, right? Engine itself has got, right, fuel injection system, safety systems, right, infotainment.

My GPS, so many ADAS, right? Modern car has got automatic driver assistance system. So many things, right? It is what is called CAN, Controlled Area Network. Like internet, your car has got its own. It's called Controlled Area Network, CAN Network. Right? That, that. Every component talks to every other component.

If something goes bad, right, it will actually report and it will run at reduced performance. Because why? If the brake doesn't work, if the steering fails, Right? I might cause, I might die or I might kill somebody. Right? Understand the importance of this. Especially from a software perspective. Right? Most modern devices run some software.

Please understand. Everything is IOT connected these days. Right? Something fails? Gone. [02:23:00] And finally, every software backup will have a hardware backup. Say, what is this? I'll give you a simple example. All of you use the NIFT, right? Chumma you'll run in the middle, door will close. You'll run in the middle.

Right? Okay? Okay? What is the most common way a lift can fail? Can somebody tell me? Uh? One at a time. I have not given any marks for participation. If you can tell me your roll number and answer, that will be helpful. Yes? Sir, line cut. Ok, cable gets cut, it drops, and then you fall to death. Is that what you are saying?

Yes, sir. Ok, that's the typical answer I get. What's your number? CS 23. CS? 22? 23. 23? B105. 10? B10? Yes. What is your name? Ernesh. Ernesh. So, Ernesh says the cable gets cut, right? That is very unlikely. Any other answer? Any other answer? Power shutdown when the lift is. Okay, you will [02:24:00] not die now. Okay, lift it.

Number? 14, sir. I 14. I, ah? Yes, I 14.

Yes. Huh? Yes. Yes. Yes. Door crushes you. Door crushes you. Okay. That's a possibility. What's your role number? I 5,400. Right. You know what is the common failure actually? Right. Closing doors, which is what I said. Right. Anyway, I'll give you a card for that. Right. What's your number? I 27. I 27 1 0 2 7. Okay. He, he said this.

He said this happens. Right? You meet me after class. Guys, listen. Write jokes aside. So, lift runs on [02:25:00] some sort of operating system. Do you know that? Our lift is a dumb lift. Why is it a dumb lift?

It is called a simplex system. What is a simplex system? There are two lifts, right? What do you do? Press this. It does not come. You go back. Press the other one. But how, but if you look at other buildings, they will be talking to each other. Oh, yeah, yeah, yeah. Right? You press. The nearest lift will come. Yeah, yeah, yeah.

Right? So, once I went to a building, there were ten lifts. Right? I pressed. I was waiting. Lift behind me came and waited. I did not know that. See, I am so used to our system, right? I went, right? Then, uh, somebody calls in. That is waiting for you. Oh, okay. Right? It's at my back. I didn't notice that. Right? So, there are five in the front, five in the back.

I didn't notice the ones in the back. They talk to each other. Right? Now, the point is. So, let's listen. Right? The point is. Right? These things talk to each other. Right? Now, sir, software can, how can it fail? [02:26:00] The most common cause of death in lifts is not the cable getting cut, it's not the door crushing, it's actually the lift starting to move when you enter.

Think about it. That's good. That is the most common failure. Right? How can that happen? You get in and it starts moving. That crushes you. Right? So, there are interrupts. Coding, have you heard of interrupts? Right? There are also called interlocks. Okay. So, the logic of what the lift follows is this. First it will check.

Have the doors closed? Right? There are two sensors. Ok, it says yes. Door? Right? There is a logic, no? Zero or one? One. Doors are closed. Fine. Right? Has both the doors closed? Right? Inside and outside. Yes, they are closed. Fine. Is there an instruction to move? Yes. Where is the lift? Right? Is the lift level? It checks the level.

Yes, lift is level. Right? If it is out of level, it will not move. That is why it gets stuck. Even in the smallest problem, right, it is irritating. Right? [02:27:00] Yes or no? Yeah. But why does it get stuck? It is designed, right? If anything is wrong, it will just stop. It is designed to simply stop. Why? If it moves, you will get crushed.

Right? That is the simple failure mode. So, there are three, four things your lift will check. Right? Are all the doors closed? Not only in this floor. Every other floor. Because somewhere, some other door is open, it goes, you might even fall, fall, fall inside or outside. Right? Understand that. So, first thing is, all the doors it checks, the entire length.

In other words, all the nodes are serially connected. If you open the chain, what happens? It goes from 1 to 0. Easy na? Take all the sensors, connect them in series. Even if it 1 opens, and logic, right? 1, 1, 0, 1, everything then becomes 0. Your logic becomes 0. Right? Like that, there are multiple redundancies in your lift.

And for whatever reason, right, your microcontroller gets, there is some, moisture. And then there is some short circuit and something tells it wrongly in [02:28:00] the microprocessor. Pin, right, are you taught 8085? Right? So typically, your writer has got like 40 pins, ok. If there is a short, how can a short happen, drop of water.

Just a small condensation. Or an ant going and then getting electrocuted. It can conduct. So what happens, let's say if this pin captures this Right is the pin that checks the door. Door is open, but some anti shots are getting, and what happens? The lift thinks the door has closed. Why? Because the micro control is reporting, but physically the door is not moved.

This is why I say for a CS engineer, it is not just coding. Please look at the hardware you're working with. If it's, if you're doing just this window on the front end, backend, all these things happen. Right? Now, the manufacturer is intelligent. He would have planned for this, right? This is a failure mode, right?

So he will check this and he will also check other logics. Okay? This is what [02:29:00] is called, right, in communication we call this, uh, I think it's called parity checking. So when you send a signal, right? Right? When you send a signal, A, let me call it A. I will also send the A compliment.

You understand? Every communication path, your RJ45 cable, it has got A, A complement, B, B complement. Why do I set a complement? For whatever reason, this bit gets altered. It becomes zero. What, what happened to the complement? It is not zero. There is a disparity. System will actually stop and ask, what is happening?

Maybe the wire is cut. Maybe some band is short circuiting, whatever. Right? So you have So even in, right, physical systems, right, in softwares, you have to think about redundancy. Keep this in mind. This is a central thing because in your coding, especially with any device, [02:30:00] whether it is a lift, whether it is a car, could be a medical device, anything, any false positive or false negative can cost somebody some life, right?

Clear? So, this again, you see, uh, this thing, right, I'll skip through this, right? So, You see this, right? Same thing. This is the air conditioning system, right? I don't want to go into the depth of, right, how this all works, right? But understand the same, same logic on a broad level. You can see these boxes, right?

You take and then you combine them into what? What are they called? Subsystems, right? This is how your systems on a chip all these came into being, right? So, DSM for risk assessment, right? Now we are asking this question. Sir, what if this module fails? Right. Which is most likely to fail? Areas of high development risk.

Right. These are areas, right, where it can really, this is talking about a Mars rover, right, when it is landing. Right. This is for NASA's, I think, [02:31:00] Pathfinder. Right. About 13, 14 years old. Right. So they have done this. And then they say, areas, whichever you see in red, is mission critical. If this fails, the entire mission is gone.

Right. Probably has spent 2, floors, gone. Right. So you can identify. Right with DSM, right? And then you can actually look at high potential areas with high failure prone or high risk areas, right? With high severity. Fine. So tweaking a design, sir, how do we tweak a design? Okay, so all of you have seen a treadmill, right?

It's a excess equipment, and then you have what is called as a foldable, right? Mm-Hmm. electric treadmill. You take all the components and then you see the. Look at the relationship, right? And when you see this, right? From this to a foldable, right? Look at the arrow direction, right? The original, uh, relationship, right?

Matrix relationship is like this. You see [02:32:00] how different parts are linked. And then graphically you, you convert that. This is the same data in graphical form, right? So you put a hierarchical graph. And then you remap it, right? So this is what I mean by tweaking the ISM. Tweaking the ISM. Here. Right. DS instruction metrics, right?

To get a, what is the fundamental problem, right? It takes a lot of space. Your typical treatment takes a lot of space. Now, if I can fold it, if I have to fold it, what should I do? Right? You can look at the relationship, different components. I look at this, right? By modifying it, it's less complicated now, right?

Less complex. No. Right? So you get this, and this is a clamping chuck, right? So what is this used for? Foundation course you were doing some, uh, right? All of you are doing, right? Some wooden, wooden, uh, flute. You had to clamp the flute onto something? Yes. You used a vise? Bench vise? So this is a bench vise, right?

So it's got six components, right? So you work out, right? Same ISM logic and you find levels of hierarchy. Okay. Now what is [02:33:00] the point of doing ISM, right? With this simple example you can see. If you look at component five, right? One and this three bolt. This is a separate cluster and this is a separate cluster.

Guys, listen. I hear some murmuring. Right? Listen. Right? So, I have two separate clusters. So if I have to reduce cost or if I have to make a change, which cluster should I touch? This cluster or this cluster? Second. Which is easier? First. First is easier. See, if I touch this, it is, it is affecting everything.

You can see, right? If I touch this screw, it is going to affect everything. If I touch this, what happens? Only this changes. If I touch this, only this changes. So if somebody, if my boss comes and says, okay Karthik, reduce this cost, maybe 5 rupees you reduce. What will I do? I will say, okay, I will do this and say, okay, right, there are two clusters and in this cluster, right, I can touch this, only this much is going to affect it, right.

I can selectively pick areas where the [02:34:00] interaction, right, the coupling is actually low. I can change the material, I can change the process. Even in coding, right, there is something called kernel. All of a kernel, Linux kernel, right? Generally, you are not allowed to test the kernel. Why? Any idea why? You are not allowed to test the kernel.

Why? It controls the hardware. It controls the hardware, exactly. Right? It's at a fundamental level. Right? It is the bottom most layer. On top of that, everything runs. Correct or not? Who answered? Correct. Right? What's your number? i1003, right? So, kernel it runs at a fundamental level, right? If I go mess up with that, that's going to ruin.

Why? I'll give you an example. For instance, it may control the speed of my computer fan, right? If I go do something, mess up the fan speed, what happens? My processor will burn. It'll ruin my hardware. [02:35:00] So generally, there are certain things I am prohibited from doing. Even in Android, if you write an app, there are certain things the kernel will not let you do.

It will stop you. There are some safeguards in place. Software thresholds, right? You cannot, for instance, right? It will, the system will not allow you to write, let's say you want to write a rogue program, right? Chuma, I want to ruin people's phones, ok? Chuma, I write a program that loads the processor like crazy, ok?

Trains the battery and it, it'll do after, let's say two days after installing. So Google, when they check, right, let's say come up with a logic like that, right? There are safeguards in place. Your Android got safeguards in place. There are some people who are doing all this nonsense. The initial Android is right, I can write it.

You can, anybody can go do that, right? But your current rate, it'll actually throttle the process, right? That's why your computer is slow down. Have you seen, if you hit a hundred percent, what will happen? Your system will hang. Why does it hang? , it can actually continue to work, but it cause physical damage, right?

So [02:36:00] you understand that. So what should I touch, right? So you, I will help you do that last, right? Important principle. I have over 15 minutes. Let me cover this guys. This is the single most important thing I can teach you, right? So that's why I kept it as a last slide. So the question is, right, how do you come up with a new thing?

Okay? What is this morphology? Right. ISM. Okay. Creating variety. What is variety? There are so many water bottles. Can you see? Look at in this classroom. Look around you. So many different designs. Right? And then if I ask you, come up with a new design, how do these, uh, generative AI work? Use the existing model.

Take the cap from that, body from this. Right? Yes or no? It will mix match. Right? It will take different elements and it will try different connotations and combinations. Yes or no? That is all it does. Right? Okay. That is all we are going to do now. Right? For an example, what I have taken is six shavers.

Right? Probably from [02:37:00] Philips, Philips, Braun. Right? Different manufacturers have taken different. And then what we have done? Right? Take the top part, bottom part, the button, cover. Right? You can take it into pieces. Right? So I have a Targus presenter. You can go to Logitech, buy them. Maybe buy it from Philips, right?

And then not everything looks the same, correct? But buttons more or less may be same or may be different, right? So I come up with this, what is called morphological analysis. So it's called a morphological chart. So this method is called morphology. Okay, morphological analysis. So you take the body, you take the holder, take the keys, take the bottom, right?

And then you look at the relationship between Can I take this top, right, and will it assemble to this bottom? Can I take this button and put it on this? You understand? So what am I doing? I am creating new variety. [02:38:00] Or in other words, what did I say? Variety is a measure of complexity. Remember, we saw this in the earlier classes.

So, with this as a starting point, I have created six more designs. Ok, how did I do that? By mixing matching. Ok. Right? Different components and they have to fit together. Please keep that in mind. You cannot randomly assemble them. Right? That's why this shows you. Right? How these things are interlinked.

Right? Using this, you can actually create new designs. Right? From existing designs. That's the thing. Now, if somebody is asking, give me a good shaver. What do you do? Why did you buy a particular phone? There are aspects of your phone you hate, right? You wish, you will say, I have OPPO, but that OnePlus, that power button is nice.

I don't know, but this, I bought this phone for this particular feature. Maybe the camera is good, maybe the display is good. Somebody might have to say, battery life is beautiful, but camera is bad. You'll say, right, can you take the best feature of [02:39:00] every phone, put them together to create sort of a, right, your, whatever, your dream phone or whatever phone.

Right? But they should fit together, right? Components may not agree with each other, right? So that chart is what you see. This is what is called as a morphological chart. You take different designs, break them into what is called primitives, right? Different component level. Same thing you can apply for, you know, like a application development, right?

What does Samsung do? Or what does Apple do? They benchmark each other. There are some good designs Samsung comes up with which Apple copies. There are some good designs that Apple comes up with which Samsung copies. What is the end result? Both look the same. If you actually look at the latest, uh, there was a comparison I was reading.

The, uh, what is that the, uh, Watch. Smartwatch. Looks exactly the same, more or less. Right, why? They were copying each other so much. Right, [02:40:00] back and forth. Right? Apparently the CEO of something got so irritated. He said, next design, he says, we will try to, you know. Because they used to actually experiment a lot and fail also.

Right? But what have they done? They have become so big, they fear failure. Right? That's why smaller companies like LG, you LG. But, it never took mainstream, right? People didn't like it. So, they went bankrupt and then they closed the division. This morphological chart is fine, right? One more concept, right?

That I've reserved, I still have time, so let me quickly run through it. You know Huawei? Yeah. Of course. Until Huawei was there in US, Okay. Guys, he was talking about optimization at the beginning of the class. So, this is optimization. Okay. It's called NK model. N you know, nodes and relationships, right?

Already we have seen. Okay. [02:41:00] So, we are going to take network concept, right? And create new product, products. How do we do that? Right? So, this is an existing product, right? You have nodes and relationships. How can I create a new product? Mechanism here. Addition of new notes. You put new elements at, put new components.

Your phone doesn't have a camera. Guys, your phone doesn't have a camera. Put the camera module right, or your phone doesn't have a heart rate sensor. Put that right. So that's a new mechanism linking of new nodes with each other. You put new notes and you can make them talk to each other. That is another way.

What is another way? Linking of new nodes with existing nodes, right? For instance, right. I've put a temperature sensor right now. That temperature sensor. This is, this is, this is, this is, this is, this is, this is. This is a, how do [02:42:00] I say, this is a, what is this? Google has got an excellent thing in the name of, what is this thing called?

Jamboard, no? Anybody use it? Yeah, sir. They want to discontinue it. Yesterday, yesterday I got mail. Yeah, they want to discontinue it. I don't know, right? Nice feature. Right? Removal of existing links and linking of existing nodes with each other. Right? These are six mechanisms by which you can actually, right?

So, you have, how many of you have heard of Prusa i3? How many of 3D printers? How many of you have used the 3D printers that have used 3D Printer? Myself. Have you design manufacturing lab? Huh? We had ID lab. ID lab. Did you use it? Yes. Right. So Cruiser is an open source printer. Okay? So it's an open source printer.

So what they wanted to do is, right, the idea tell you what is a 3D printer. 3D printer can make any component. So the thing is, can a 3D [02:43:00] printer print itself? Can a 3D printer make a 3D printer? That, it's called a replicator. The name, what they came up with. Can it replicate? This is in theory, this is the idea.

It's a beautiful idea. Not practically possible, right? Because not bearings, right? Certain linkages you cannot 3D print. So, you still have to use that. But, his original thinking was, I will make one 3D printer, that is going to print, it's going to exponentially 3D print everything. That's a beautiful idea, right?

He called it replicator. Okay? That was the motivation. So, the first evolution was, guys listen, right? Prusa is what you will find today. Go to Google and, uh, Prusa. Prusa you can buy, right? As a kid you will buy. About, uh, 20, 000, 25, 000. So, the initial thing is Mendel, Huxley, right? It evolved, right? And then if you look at the trend, Reduction is average degree of network, right?

Decrease in number of links due to integration and combination of parts. Why? DSM, this is what I am talking, right? You have to consolidate parts, put them together. Because having too many, right? That's everything. Average cluster coefficient decreases decrease [02:44:00] in the degree of coupling in design due to multi-functionality.

One part is doing many things right and decreases in the shortest path. Reduction, simplification of parts and correctivity, right? So same thing, right? So it was done for gas lesson, right? For two electric bikes, right? Electric bicycles, right? So you see right standardization of components, right? And then they say centrality, you look here.

Battery, right? The power module tires. Right? You can see there are some densely connected nodes. What are they? The battery system, obviously. And specifically the cable housing, right? And chain stay. Fine? Last concept. Ok. Now, how, how do I, how do I optimize? Right? I can ask that question. This is what is called, the end is called genes.

In other words, the number of parts in a component. Guys, listen. How many? What is the number of parts in a component, right? K is what? The [02:45:00] average degree, right? So 1 is connected to 2 and 3, 2 is connected to 1 and 3, 3 is connected to 1 and 2, right? And there is something called variant. What is a variant?

Guys, listen. What is a variant, man? iPhone 16, 16 Pro, 16 Pro max. That is a variant. You buy this, you will get two cameras, you buy this, you will get three cameras, so on so forth. So here, what is a variant? you In one variant, okay, let's say you buy a pencil. I'll give you a simple example. It's made of plastic, sir.

You buy another pencil, it's made of metal. So, there are two variants, right? So, in this, if you buy a pencil, one pencil is made of plastic, another is metal, another may be made of, right, some other, maybe wood, some other material. Fine? This is clear? So, in my, in my argument, to keep it simple, I have three parts.

I have average area of two, There are two variances, fine. So, which results in 2 power 3, what is that? [02:46:00] 8 possible variances, ok. So, 1, 2, 3 how can you put? 0, 0, 0, 0, right. This logic you know, you write down, right. And you give weightage, right. And then you say, you ask the question, ok, if If I make part one out of wood and my customer like it, would it fulfill their functionality?

Right? If I make my part out of aluminium, what happens? Right? If I change this part to something else, I can ask these questions and then give weightage. Right? So, fitness value of a particular product varies with configuration. When I say fitness value, it means how close is this fulfilling the requirement of my customer.

Functions, right? What are functions I have? So, I can actually start here, right? Slowly, right? Local peaks, I will show you. Right? So, This is how typically, right, your fitness and your, what is called as a search space. What is search space? I have to do a, see, this is just three parts, keep it in mind. [02:47:00] How many parts will, right, how many parts does this thing have?

Links, ok. This is number of links. How many parts? Hundreds of components. So when you have hundreds of components, this is just three, so I can visualize it in, right? If there is a fourth dimension, I cannot even visualize it, please understand. Only if there are hundredth dimension, I cannot do that, so.

Right? So it becomes what is called? It goes into hyperspace. Okay? So you have 1D, 2D, 3D. Is there a 4D? What is the 4th dimension? Time. What is 5th dimension? Physicists say there are 11, 12 dimensions. Right? I can't even visualize it. Right? So here, right, you want to maximize something. Yes or no? There are two things, n and k, right?

When k is zero, right, when the number of linkages are zero, right, average linkages, right, you can actually travel in this, [02:48:00] what is called fitness landscape. It is like a landscape, right? You can change multiple things and that will result in different fitness levels. So you can actually traverse it. But where do you start?

Where do you start? This is visible, so you can see. Now this is in, like I said, if you have 100 dimensions, you can't even visualize it like this. It's impossible. So how do you do that? Right? You simply put a computer program, let it reach what is called the global minimum or global maximum. You have to do what is called as an exhaustive search.

I mean, there are different methods, right? You can do bracketing, right? Newton Raphson, have you heard of Newton Raphson? There are many methods like that. You don't have to use one very basic method, right? There is cuckoo algorithm, like, like how a cuckoo bird will go. They use all this. It will be here, suddenly it will fly, go here, start searching.

Right? Your search space is so huge, if I start searching now, there is high likelihood. It's like looking for water. [02:49:00] Where will I go? Let me search here. Okay, let me go there and search. Let me go there and search. Right? It's called cuckoo algorithm, right? People have come up with different different algorithms, right?

So Pretty much right. This summarizes the concludes, right? The anti-system seeking course. Right. Any questions you have our last couple of minutes. Any questions you have, guys? Any questions at all next week? Do you want a session

said yes please? Sitting here? No. If you want, I'll come here. Okay. If you have anything you can ask me. If you want. CR, where is the CR? Or you do one thing. You come to my cabin. Okay. 2 to 3 I will be in my cabin. During regular class hours you come to my cabin. 310J. Come to my cabin. Anything you ask me there.

Fine? Any questions at all? No sir. [02:50:00] No work is still there? Uh? Okay. You're welcome. Guys. Alternance. So, see I, I will not ask anything unreasonable. Okay. Right? I'll be very reasonable. Okay. Generally our institute doesn't, uh, encourage open book test. Otherwise we can even have an open book test. Right? For course like this, that's, that's actually something I've been asking.

Okay. So, what I will ask is, okay, I'll tell you this. Please go and revise. Till the 10th slide I will post, right? Probably this, right? NK I will not post, right? 11th lecture, forget it. Just for information. Till morphological chart. Till morphological chart, chart comprises of 10 slides, right? Within this, I will test, right?

Your understanding of discovery matrix, your understanding of SNAC, right? ISM, ESM, whatever methods I have taught you. I will [02:51:00] ask you your fundamental understanding. The question will be set up such that, right? It can be 3 marks, 5 marks, 15 marks, right? It will be a combination of that, right? Fundamentally, I will test your understanding.

The question will be posed in such a way, I will test whether you have understood and you can apply it to any, any, any problem that I give you, any real world problem I give you. Got it? Right? It will not be like, chumma, you cannot just pie heart and come, chumma, Right. It will not, exam will not work. You cannot.

My heart, that will not work. You have to really understand. Right. If you have any trouble with any part of the, everything you have to understand. Right. All the best for your exams. I will post your next assignment today. Okay, I'll give you a week next till next Friday. Right? Finish it. Right. Go today and then.

Attendance.[02:52:00]

Attendance down.[02:53:00]

all very much.