PROFESSOR: Let's talk about the V-Model, here. We are-- interesting, we have a little bit of a lag here. So the lifecycle management is our last topic for today and that sort of completes the V. And then next [AUDIO OUT] manufacturing.

So what I'd like to cover today is essentially a definition of [INAUDIBLE] mean by that, and then focus on the lifecycle [INAUDIBLE] also known as the [? ilities. ?] Discuss a little bit where they come from, how much we know about them. The bulk of the discussion is going to be a case study called Reconfiguration of Communication Satellite Constellations. And this sort of explains the idea that system, once you brought a system into operation, that's not the end of the story. The system is going to continue to live and change and be modified over its life. I'll try to summarize some of the key concepts over the whole semester. And then I have one slide on career and study recommendations, if you want to go further into systems engineering.

So lifecycle management. So this is my definition of what lifecycle management is. It's essentially the active engagement of-- it's the active engagement of all the stakeholders with the system, between the time when it starts to operate-- so you've done the commissioning, you've done the operations-- until you decide to decommission and retire the system. So it's during the whole operation life. And what you want to do, is you want to maximize the value that you gain from the system's existence. And so life cycle management starts from the very, very, very start.

And so I listed here some activities that are included in lifecycle management. So obviously daily operations, you know monitoring, is the system working as expected? Training and certification of the operators. Sometimes, you know, in long live systems, the people who were the early operators, generation one, are not necessarily the people who are going to operate for the whole life. So you have to have new people operate. Maybe one of the most extreme examples of that is an airplane known as the b-52. Have you heard of this? This airplane might be the first plane that actually has a 100 year operational life. And there are apparently

Navy-- Air Force guys at MIT, you can confirm this, that there's actually like-- the grandfather was the first generation and there's like third generation pilots now flying on the b-52. That's what I've been told. Did you hear this as well?

AUDIENCE: Yeah, That's correct.

PROFESSOR: Do some of these people?

AUDIENCE: No, I've never actually worked on that airframe.

PROFESSOR: OK. But anyway, so that's what I mean by [INAUDIBLE] not just the initial operators, but over multiple generations. Then the third thing here is servicing. And servicing comes in two flavors, of preventive maintenance, you know regular maintenance, and then corrective maintenance, which we also call repair. Dealing with small and large failures, recalls, anomalies. This is a big deal in the automotive industry, as we know. Increasingly protecting the system from either random or targeted attacks. Cyber, physical, right, physical security but also cyber security is becoming a huge topic. Sharing and archiving the data that's being produced by the system. More and more systems produce larger amounts of data. Well, what do you do with it? How do you store that data? Upgrading and retrofitting the system as needed. So retrofitting means we're physically changing the configuration of the system-- right?-- over its life to add new capabilities.

And upgrading; upgrading can be done sometimes without having to change the hardware. So you're doing a software upgrade, or software update, software patching. All of this would go under upgrades. Cross strapping the system with other system in a federation of systems. So what I mean by this is a system may have been designed just to operate on its own, but now you're connecting it with another system. This happened, for example, the electrical grid. Right? The early electrical grids were local, regional power grids. And now we've connected them, in the US we have three major grids, the eastern, the western, and then Texas has its own grid. Texas does a lot of its own stuff. It's called ERCCOT. And then here in Europe, you know the European electrical grid is now continental-wide. It didn't used to be. And of course, that has big implications for operations. Reducing the resource consumption and environmental burden of the system over time, and then finally, decommissioning the system when it's time to do so. And in many systems, this is a big challenge as to finding when is the right time to decommission the system. And that's often the case when, you know, the operating costs exceed the value that the system delivers.

So this is a pretty long list here. And a lot of tasks, a lot of decisions to be made. Any questions about this list? I don't think it's complete, but this is a lot of things you need to take care of during operations. Here's a more graphical view of this. So this is my sketch of the systems engineering lifecycle. So this is part one, right? Conceptual design-- conception, design, and implementation. You start the lifecycle, you do something like a system requirements review, understand the mission, the requirements, the constraints. You do conceptual design, that's where you do creativity, architecting, trade studies. As a PDR, you choose-- so in this case, you know, we're going to go through this triangle concept and then you design all the details within it. That's where we do modeling, simulation, experiments, MDO. And we iterate-- sometimes you have to iterate this [? dashed ?] duration means abandoning the concept you chose and go for a different concept. That's usually undesirable. After the CDR, we implement the system, we turn information to matter, and this all happens in a technological, economic, social context.

So then, the second part of it, which in many systems-- if this is 10 years, this could be 30 years or more-- is the actual operational space. So the system has now been tested, validated, verified, and deployed. And now, you know, we operate the system. Things break, we need to service it, and then I have this-- you see, there's two versions of the system. One is solid, and the other one is kind of faint. That's the virtual version of the system.

So one of the big concepts that's being pushed now, certainly in the US by the DOD, but also other places. Is the idea of a digital twin. The idea of a digital twin is that there's a physical system-- right?-- that exists, and then there's a digital version, a digital twin of that system, somewhere on a computer that mirrors exactly what the real system is doing. So if something breaks on the physical system, well that same component fails in the digital twin. And then, before you actually do a repair or any actions, you execute those actions on the digital twin to see whether the system will operate properly. And so this is kind of the latest thinking, is that for any system we should have a digital twin doing operations.

Upgrading the system. So in this case, we're adding things to it, we're connecting it. You know, and then at some point, the system does degrade. Because, for example, you know, materials, age, they get brittle. Technology becomes obsolete. You know, it gets harder and harder to maintain spare parts or the suppliers of the original spare parts went out of business. So these really old legacy systems can be very expensive to operate. And then at some point you liquidate and it's the end of the life cycle.

So I've already told you about one example of this, which is the space shuttle. And so I'm not going to belabor this again, but I think it's a great opportunity to learn. The space shuttle had a 10 year design life, roughly from 1971 until '81, first flight. And then we operated the shuttle for 30 years and we had a total of 135 launches during that time. We spent $192 billion dollars on the shuttle program. And if you average that, it's $1.5 billion per flight, which is a lot more than was planned. But just yesterday here at the EPFL, we had the annual meeting of the Swiss Space Center. And one of the astronauts here, Claude Niccolier and his colleague, Professor [? Mehron ?] gave a great talk about the Hubble and the servicing of the Hubble and the amazing things we've learned through it. And you guys at MIT, Jeff Hoffman, was part of at least the first servicing mission as well. So there's incredible things that were done by the shuttle.

And so I think we need to acknowledge that, not just the fact that it was very expensive. It accomplished great things during its life. But you know, if we had a chance to do it again, would we-- would we come up with the exact same system, would we make all the same design decisions? You know, we lost two of them. Probably not, we could probably do something-- something different. So here's what we wanted, and then finally, we got a much more complex system in the end. And particularly my sense is that a lot of the things that made the shuttle expensive to operate during its life were things related to maintainability. Right? Reliability and so forth. So the lifecycle properties, which I want to talk about next.

So I'm just going to repeat the questions, so you guys can hear as well. So the question was about reusability. So the shuttle was-- the orbiter was fully reusable. The external fuel tank was not reusable. Right? And then the solid rocket boosters were partially reusable because they had to be stripped down, you know, and rebuilt, essentially, for every launch. The idea of Blue Origin and then Space-X Stage One, flying it back to the launch site, is absolutely the fact that if you can make it reusable, then you can amortize the capex in that element over multiple flights. And it should drop, you know, the cost by a factor of five or 10 or more.

Now, the devil's in the details, right? So if you start and restart an engine multiple times, you can have a big transient every time. So was this system, in fact, designed to withstand these transients? Can it handle multiple starts? What does it do to the materials? And so forth. And so to really model this ahead of time, and then test it over the cycles is the key. So in the shuttle, the big problems-- the two major subsystems in the shuttle, in the orbiter, in particular that made it expensive and the difference between the top picture and the bottom picture was

a, the TPS, the thermal protection system. You know, every tile has hit a different curvature in geometry, so really inspecting every tile, replacing tiles, making sure that the TPS-- because just one weakness in the TPS could basically be fatal on re-entry. And so, TPS was very difficult and-- you know, as opposed to an ablative heat shield that you just sacrifice the heat shield completely.

And then the other was the main engine-- the shuttle main engine. Originally, the idea was to only do very deep inspections and disassemble-- the disassembly of the shuttle main engine after the first few test flights, and then fly multiple times without having to really re-inspect or disassemble the engine. So they did the disassembly of the engines after the test flights and then they just kept doing it for every flight, where the original intent was to only do it for the test flights. And so the shuttle main engines were the second big cost driver in operations. So re-usability is a great idea, but to the degree to which re-usability actually happens in reality, that's really about detailed design decisions you make.

OK. So let me move on then. When it comes to really understanding lifecycle, I do want to point you to the 15288 standard, which is the ISO 15288 standard, which has a fairly complete list of system lifecycle processes. So this is, essentially the system and software engineering, system lifecycle process standard, that has a whole range of processes that are described in quite some detail. And they're not just the technical processes shown here on the right side, we focused a lot on those technical processes in this class, the stakeholders, the requirements, the architectural design-- which is essentially the conceptual design-- but there's also the project processes, right? Executing the project agreement processes. So this would be negotiating contracts, you know supply contracts, acquisition contracts, and then all the organizational things you need to do to create the right organization to execute these projects. So what's nice about the 15288 standard is that it's a pretty broad standard.

OK. So let me talk about the ilities of the lifecycle properties in particular. What they are and how we can describe them and how, especially how they relate to each other. And so the paper that underlie this lecture that you need to get some more detail. And this is one of them. So this paper is called Investigating the Relationships and Semantic Sets Among Lifecycle Properties. And we published this about three years ago in Delft at the CESUN Conference. So the background on here is the following, is that as you've seen, complex engineering systems live for decades-- some of them even for centuries-- and the ilities-- by ilities, we mean properties of systems that are not the primary functional properties. In software engineering

they're often called nonfunctional properties. And the thing that's tricky about the ilities is that you often only can observe them during operations. So you know, you can test systems in the short term, you can see does it work? Does it fulfill its function? But whether certain ilities are present, it often only shows itself over time. And so most of the research that's been done and sort of quantifying the ilities, has been looking at these properties one at a time. So the questions we wanted to go here, wanted to answer is, which of these lifecycle properties are more prevalent than others? You know, the top 20. And then especially, what's the relationship among lifecycle properties? Do they form what we call semantic sets? And then how could you use this information?

And so here, we're going to do this using two different methods. The first method is what I call prevalence analysis. We're going to look in the literature and on the internet, how frequently these lifecycle properties show up, are mentioned, how much we know about them. And then the second method is a cognitive method. Where we ask people to give their understanding of the lifecycle properties and put them into a hierarchy, and then we'll compare the results.

So here are essentially the results from the prevalence analysis. And this is ranked according to the number of journal articles. This is scientific papers written, where this particular key word, by quality or reliability, shows up in the title or in the abstract of the paper. And you can see this is in units of thousands. OK? So quality is number one. Right? Quality is number one. The most scientific papers are written about quality. And you can see, it's almost a million journal articles. This is-- I'll show this to you over time, but since 1884 is the first year. And the databases that we use for this are called it's Compendex and Inspect. These are actually combined-- if you go to a website called engineeringvillage.com, this is sort of a master database for scientific papers and engineering. That was the basis for this. Number two, reliability. Number three, safety. Then flexibility, robustness, and so forth.

And the other bar, the gray bar, is if you Google essentially for this particular keyword and-- you know, keep in mind, this was done about five years ago-- this is the number-- the millions of hits that you will get. And, so you know, there's a factor of 1,000 difference here between journal articles and hits on internet. But still, you can sort of compare the two. And so, in some sense, the black bar is the totality of scientific knowledge about this lifecycle property. And then the gray bar would be the amount of information or uses-- usage of that knowledge, at least as far as it's represented on the internet. And what's interesting is, in some cases the black bar is smaller than the gray bar, which means that common usage is [? leading. ?] So

sustainability would be an example. Everybody talks about sustainability, companies have-- they say, our system is sustainable because, you know, it uses less resources, it produces less emissions. But the actual amount of scientific knowledge, the actual amount of research as to what really is sustainability and how do you design for it, is actually smaller than the usage.

And then there's areas where it's the opposite. Like for example, modularity. Right? You see that in modularity. So in the relative sense, academic interest is leading. We see there's quite a lot of literature on modularity. In mathematics, you know, modularity and software, modularity and design, but most people-- you know, modularity per se is not so interesting and so exciting to the general public. So keep in mind that for some ilities, there's an imbalance essentially, between our understanding of-- the scientific knowledge, our understanding to be design, how to design for it, and you know, how frequently at least a keyword is used.

Now here, this is a little bit more-- this is a little bit harder to see. This is essentially looking at these lifecycle properties over time. So what this graph those you know is starting here in 1884, cumulatively the number of journal articles published about each lifecycle property. And roughly what way you can think about this is that there's some lifecycle property, the top four-- five that we've been actually working on for a long time. Even safety, you know, there's some interesting articles in the 1890s about safety for example, in mines. You know, coal mines. There's an article about how does the impact of lighting-- better lighting-- on safety and productivity in 1890. And so they show that just providing better lighting, actually has a dual benefit; fewer accidents, fewer fatalities, and better production output. And so that's-- and this was a scientific study that was done by comparing data in different mines and actually performing some experiments.

Then you have a group of life cycle properties that we only started really thinking about and publishing about I would say, around World War Two. Such as usability, maintainability, and my interpretation of this is that especially during World War Two, a big difference made logistics. How easy was it was it to use different weapons? How easy was it to maintain equipment? It became a huge determining factor in the outcome of the war. You know, for example, in North Africa in the North African theater, you know tanks, trucks being exposed to the sand and so forth.

So people really started thinking about the military, started thinking heavily about the importance of maintainability in the design of these systems. And after the war then, a lot of

these concepts like usability and maintainability started spreading into general civilian life and products and so forth. The third type of ilities are the newer ones that we've done research on just since the 70s, in the last 30 years. So here in this category I would put things like sustainability, recyclability, evolvability, even interoperability-- which means the ability of systems to connect together and work across system boundaries. Those are pretty recent lifecycle properties and we're still actively researching them.

So another way to show this is by essentially making a network of these lifecycle properties. And so the way-- this is still all using this prevalence data. Essentially what you see here is a network diagram that has these lifecycle properties in a network relationship. Let me just explain, so the size of the nodes relates to how much knowledge we have. Essentially the height of the bars that we saw earlier. And then the strength the thickness of the line relates to the strength of-- really how closely are these two ilities related. And the way this is calculated it is using the so-called 2-tuple correlation. So you take, essentially, articles that have-- you look at articles that have, for example reliability and maintainability in the same article. Right? And then you divide that by the total number of articles on reliability and or maintainability. And that's a ratio between 0 and 1. And then this graph was produced with a cutoff strength of so. 1, right? So if there's more than 10% of articles list these two properties together, then there will be a line here. And the stronger the line, the thicker the line is, the more closely, the more often these concepts are mentioned in the same article or the same piece of work. So that reflects the strength of relationship.

Now, what's interesting is when you first look at this, you don't see much. But after you look at this for a while, you start to realize a few things. First of all, in the center of the graph we have the classic ilities of engineering. Quality, safety, reliability, and I would argue flexibility as well. Those are the top four that we saw before. And then around the periphery of the graph, we have lifecycle properties that are more recent. We haven't really thought about them too much, we don't fully know how to design for them yet. And if you look at them in groups, I will argue that there are three major groups here.

So the first group in the upper left is things like maintainability, durability, reliability, quality, and so forth. So this is all about, is the system made well, with high quality, particularly early in its life. Right? Then we have a group here on the right, and this is about, is the system easy to change. Is it easy to change the system configuration-- flexibility, extensibility, modularity and scalability, interoperability. Those are all different sub-flavors, if you want, of being able to

modify the system over its life. And so that's a group of ilities that that goes together.

And then the third one is resilience, robustness, quality, safety. And so this is really related to performance of the system under different types of uncertainty. You know, either environmental variability or failures in the system, and the ability of the system to withstand, or at least perform-- you know, have good residual performance even in the face of failures. So that's, I think, an important way to think about it. And when you're writing requirements for systems-- like the system should be resilient. Then this helps understand well, what are the other properties that are linked to it and that maybe support that? OK, any questions here or at MIT about-- this is sort of method one, is get data about the lifecycle property and put them in relation to each other. Any questions? Johanna, any question there?

AUDIENCE: I have a question. So is this chart that you have now actually used in the initial design and like conception [? con-ops ?] to like try to tease out the interdependencies, or is it too abstract and just more of an educational tool.

PROFESSOR: It's really at this point-- you know this is fairly recent. And this was just done in the last few years. So I don't think this has fully penetrated system engineering practice yet. But the point-- the point that I want to make here is that there's a huge-- there's a huge gap right now between when people in briefings you know, either at the Pentagon or at a corporate headquarters say, you know we want a sustainable product or we want-- we're a software company, we do optical networking and we want we want to have the most resilient design of the industry. You know, they'll put that as a goal for the project. The question then is, well OK, that's fine. But what does that really mean? What does it really mean, resilient? You have to operationalize that definition such that you can derive from it lower level requirements that you can actually go and design to, that you can test for.

So what this helps you to do is understand what may be supporting-- what are supporting elements that will be related to resilience. You know, what are supporting concepts, supporting lifecycle properties that this property that you're looking for is linked to? Right? So it's an evolution-- you know, we know how to design now for speed and energy efficiency, and you know, the sort of things that were really hard to do 20, 30, 50 years ago. It's pretty standard practice now. You know, how do you design for optimal interoperability? We don't quite know yet, but we're finding our way. And so what I believe is that especially the lifecycle properties on the outer periphery of this chart, those are the ones that need more work and those are the ones that we're really learning how to operationalize. Does that make sense?

AUDIENCE: Yeah, thanks. That makes sense.

PROFESSOR: OK, good. Let me talk about the [? message ?] too. So this was basically trying to get to-- how do people, how do humans-- there's semantics, semantics means the meaning of words, right? Semantics is the science of the meaning of words. How do they interpret these lifecycle properties? And so humans have a deep and possibly varied understanding of the semantics of these lifecycle properties. So what was done here was that a list of 15 life cycle properties, many of them are overlapping with the ones you saw earlier, were presented and then the challenge was to-- the question that these four groups-- 12 participants, four groups-- had to say, is there a hierarchy here? Are some higher level lifecycle properties are some of them lower level properties that support these. So there was a round one, find the parent-child relationships, describe these. Interviews, and then the second round.

So here's essentially what was the results of the first round. So four different groups, so each of them basically came up with its own version of a hierarchy. They didn't talk to each other, they were firewalled from each other. And in all cases this notion of value robustness came out on top. So value robustness means the system should deliver value, right, to the stakeholders. Despite failures you know, environmental changes, so value delivery of the system is the top, the most important thing. Then what's different here is how do you achieve that? So for example, you see group one had robustness and changeability. So robustness typically means even if you don't make any changes to the system, it should continue, you know, it should be survivable. It should be versatile. Changeable means you actually modify the system over its life. And then we have lower level properties like modularity or reconfigurability. And you know, there were differences between the groups but not as big as you might think.

So in the second round, as a result of the second round, then this so-called means to end hierarchy was constructed. So means means, you know, these are the enabling lifecycle properties and ends means this is the final, sort of this is really what you want to achieve. And so this was the result of that. So again, at the top we have this notion of value robustness. And this is achieved by a combination of survivability robustness and changeability. And then these, in turn, are achieved by lower level lifecycle properties. And then at the lowest level we have things like interoperability, modularity, reconfigurability, and so forth. And the difference between the solid lines and the dashed lines here is that if it's a solid line and these are directed arrows, that means that three or four out of four groups, right? So the majority of groups had this is a particular parent-child relationship. And if it's a dashed line it means only

two out of four. And if it's only one out of 4, it's not shown here. OK? So this is kind of a combined result across the four groups.

So you know, what can we take from this? So first of all, lifecycle properties are absolutely critical. You'll find us in mission statements, you know, you'll find it in even in advertising. Right? A lot of companies say we have we have a robust solution, we have a sustainable system, we have resilient networks. So it really is a huge, huge selling point. It's very critical. What I encourage you to do as system engineers is take a critical look at this and say, well how resilient is resilient? You know, how many subsequent failures of the system can you tolerate? Sustainability; what does that mean in terms of kilowatt hours per hour of usage? You have to get down to the details to start quantifying what these lifecycle properties mean and compare them among systems.

Despite differences, the two methods that we just looked at-- so one is the prevalence analysis and the other one is the human semantic exercise-- despite differences, the high level conclusions were similar. So some ilities are closely related to each other and form semantic sets, meaning they're tied together by both [? synonymie ?] or polysemi relationships. [? Synonymie ?] means they essentially mean the same-- they're synonymous-- they essentially mean the same thing. polysemi means-- it is one word, but it means you can have possible sub meanings. Right? So the idea of groups of semantic sets. And those groups essentially are robustness-- so this is the ability of the system to perform its job despite either internal or exogenous disturbances. Right? That's robustness.

Flexible or changeable, which means that you can modify the system easily. You know, if you operate the system for a while and then you realize, Ah, I need the system to do something else. Or I need to adapt it, or make it bigger, or make it smaller, or add some function. So you would modify the system and that's flexibility or changeability. And then resilient and survivable is very specifically the ability of this system to continue performing despite failures or attacks. Right? That's what resilience really means. And so those seem to be the big three semantic clusters that we see in life cycle properties.

And then the third point here is there appears to be a hierarchy of life cycle properties with two or three levels. Where we have the lower level properties of systems like modularity, for example. Really your customer probably does not care about modularity, right? If you advertise the modularity of the product, some of the more educated, some of the more technically savvy customers, they may understand what that means. But most of your

customers really won't appreciate modularity because it's kind of a low level form-oriented technical property of the system. But what they will appreciate is the ability of the system to be reconfigured or adaptive for different pieces, so interoperability, modularity, et cetera are low level lifecycle properties that act as enablers of a higher level lifecycle properties.

And so future work-- future work in this area is to both just to apply these methods to a broader set abilities, larger group of test subjects and more data, but also from a practical standpoint to operationalize better. How do we write requirements? How do we actually design for the lifecycle properties like resilience, flexibility, changeability. How do you really design it? So go from the keyword to real engineering by operationalizing the sub-attributes or factors in the system. OK?

So that's a quick summary of life-- you know, I could go on for hours about this. I'm very passionate about this topic and I will say that people who have dealt with large complex systems you know, whether you're operating the transportation system of a city or in airline operations, or you're running the IP infrastructure of a major corporation. These words, these words are real. These words are you know, real dollars, real challenges. This is really where the action is in a lot of these large complex systems.

OK. Any comments or questions about lifecycle properties? What they are, how they relate to each other, why they're important. Voelker, did you want to maybe say something? OK. All right.

So let me talk about-- try to make this a little more real. Let me talk about a case study, a very specific case study about communications satellite constellations. And this is the second paper that underlies today's lecture. And so let me give you a little context for this first. So this work here was originally published about a decade ago, in March 2004. And that was a few years after the Iridium and Globalstar-- these the two well known communication satellite constellations-- had been launched.

And I have to give you a little context, when I first came to MIT in the mid 1990s, there was a huge interest in this area of satellite constellations. You know commercially, scientifically, in fact, the impression was there were so many applications for satellite compilations filed, that we're going to have so many satellites up there you won't even see the sun anymore. It was just like thousands and thousands of satellites. And the Iridium and Globalstar were really the first two constellations that were fully developed, launched, and both of them failed

commercially within a very short time, a couple of years. And so after this happened the whole market and interest in satellite constellations collapsed for a long time, until about two years ago, two or three years ago. Now people talk about these constellations again. You know, constellations of [INAUDIBLE], Iridium Next. You know, there's sort of new enthusiasm , new wave of enthusiasm for constellations.

So this paper, and this case that I want to tell you about is really about the first wave. And in terms of the ilities, the one that that I'd like to explain to you about here is flexibility and scalability. Rather than thinking about a system as something that you design and build all at once, how do you design a system such that you can gradually deploy it? We call that staged deployment approach. And what's the benefit of that? So here's some pictures you see on the right side, this is what the original iridium satellites look like. They actually use phased array antennas-- and actually both use phased array antennas. So you have individual elements here. And by differentially phasing the signal, you can actually steer the beam. This was a very new technology at time. And then the Globalstar satellite shown at the lower picture.

Here is a little bit of that data. So both of these were launched in the late 1990s. Iridium has 66 satellites, Globalstar 48. Iridium is a polar constellation, so the satellites go almost directly over the poles. Globalstar is a walker constellation, so inclined. Doesn't quite give you full global coverage, it's about plus or minus [? 78 ?] degrees, so you can't use Globalstar at the poles. The altitudes are a bit different too. Iridium is at 780 kilometers, while Globalstar is at 1,400. Which-- I know at least one or two of you are working on the Van Allen-- the Van Allen Belt Commission. Right? [? The cube stats ?] to measure the Van Allen Belts. Somebody mentioned that today, who was that? No? Did I hear that wrong? It was mentioned, right? So who was that? Arnold, and he's not here. Ah, see.

So 1,400 kilometers, are you actually pushing, you're starting to push the lower edges of the Van Allen Belt. So one of the big, you know, in some sense, it's easier to be higher because you need fewer satellites to cover the whole earth. But the higher you go, the more exposed you are to the radiation environment. You get closer to the Van Allen Belt, so that's the big trade-off there. You know, the mass of the satellites, they're between 450 and 700 kilograms, transmitter power, around 400 watts. You know, which is not that much, if you think about it, right? That's four very strong light bulbs, the old style light bulbs.

And then what's very different, again, is the multi-access schemes. So Iridium used time division multiplexing and Globalstar, which was supported by Qualcomm, used essentially

CDMA. So you don't chop your frequency band into separate channels, you use the whole frequency band and then you use a pseudo random access code that essentially de-convolves the signal for each channel. The number of channels, about 72,000, about 120,000 duplex channels so that you can actually-- duplex mean you can carry on a two way conversation as opposed to just be asynchronous. And then you can see the data rates, quite low, like per channel. 4.8 or kilobits per second, 2.4, 4.8, 9.6 for Globalstar it's enough for having a conversation. And total system cost, Iridium was about $5.7 billion dollars and Globalstar about $3.3 million. Not including the cost of the ground stations. Both went bankrupt relatively quickly after they launched. However, they've been operating really since then, right? Since this time. Iridium Next is currently under development and is scheduled to launch in 2017. Globalstar is publicly traded and actually valued as a company at $1.9 billion dollars. So they're actually both-- you could almost argue that they were a decade ahead of their time. And

I want to tell you a little bit about the story of, especially Iridium. So here's a couple of press releases, so things that have been written in the press. So look at this one, 26th of June, 1990. Motorola unveiled a new concept for global personal communication. [? Bases ?] a constellation of low earth orbit cellular satellites. August 18th 1999, nine years later, last week Iridium LLC filed bankruptcy court protection last investments are estimated at $5 billion. So the question is why did it happen? The technology actually worked quite well, it was not a technological failure, it was a business failure-- but I would argue a system's failure. To think about the problem differently, so the fundamental challenge is to properly size the capacity of a large system. So if you're designing, you know, a car factory, a new power system and for a future uncertain demand, it's very difficult to do this because demand is uncertain. So is it better to oversized the system or are you conservative and you make it smaller? Market assumptions can change, right? In a seven to eight years. So essentially the v, right? Like getting back to the v. Each of those two systems, Iridium and Globalstar, it took them essentially a decade, almost 10 years-- right?-- to go across the whole v. And when you make your requirements, your stakeholders, all the stuff in the upper left portion of the v, there's so many years that elapse between when you make those assumptions, you write those requirements, and when you actually go to market a lot of things can change. And that's fundamentally the challenge here.

Just to illustrate this for you, showing some data, this is cellular this is not space space, this is on the ground. Cellular subscribers for mobile phones. I know this is hard for your generation

to sort of understand this, but we actually didn't have mobile phones or there were these clunky bricks in your car. You know, it's really remarkable what happened. So look at this data, in 1991 which is when the system was just being developed, there were less than 10 million mobile phone users in the US. It just wasn't-- it was very expensive, it was just not widely-- that technology hadn't been, the networks weren't there. You know, so the green bars where the forecasts that was done in 1991, as to how quickly the mobile user market would evolve in the US. So their prediction was that by 2000, a decade later, there'd be just shy of 40 million users. OK? Now the dark blue bar is the actual evolution. So actually by 2000 you know the US now has 310 or so million inhabitants. So by 2000 there were 120 million users. So the forecast that was done 10 years earlier was off by a factor of three. That grounds, the terrestrial mobile networks, developed three times faster than had been predicted. Now that's great for the terrestrial people, right? AT&T and Comcast-- not Comcast, Qualcomm, et cetera. But the problem is, of course, that because you know ground based communications was so much easier, a lot of the market that had been anticipated for the satellite based communications was essentially eaten away by this competitor, by the ground based competitor.

VOELKER: [INAUDIBLE]

PROFESSOR: Yeah, OK. Can you still hear it, MIT, are you still with us?

AUDIENCE: Yes.

PROFESSOR: Yes, you are. OK. Voelker wants to say something.

VOELKER: Actually, the figures that you showed [INAUDIBLE] and for once, Americans did not look over the pond. Because at that time in Europe, the GSM systems came really strong and very quickly. In the early 90s, you could have already had your Sim card in your cell phone [INAUDIBLE] and in fact in America then, you could only buy the telephone with all the Sim card in-- you had to buy the whole system. And so as the system was controlled by the large companies and not by the users, the large companies, Motorola, thought that they could impose their satellite system on the markets. And in this case, the country to the-- video recorders, hi fi systems earlier, Europe went much faster than America. And actually European GSM market-- cell phones, as you know them-- went much quicker. And then Nokia, which is actually European, overtook and it took probably the next decade for Americans to catch up. So finally here this was, a combination of people had the product and they just couldn't get in

the US. So they were limited to their own markets, and not to sell their products to the rest of the world as they had done in the past.

PROFESSOR: Yeah, good point. And in fact, you know one of the things that-- I think both Globalstar and Iridium had to do-- and this was a very late decision, just in the last couple of years here before launch, 97, 98, is to make their handsets dual use. Like that if you were in an area where there was a cellular network, the phone would switch to that because that would be cheaper, and if you didn't have cellular network access it would automatically try to communicate to the satellite. So there was a lot of-- there were a lot of issues that came up by essentially the markets, I think in Europe, your point, well for-- about the GSM, and in the US just developing quite differently than had been anticipated.

OK so I want to give you a little bit about sort of economics. In the end a lot of this is driven by money, by economics. And so this is satellite economics 101. The key question here is how expensive is it-- what's the cost, and then what is the price that you can charge for one minute of service, right? One minute, one unit of service. And so look at this equation here. This is CPF, stands for cost per function. And so in the numerator we have the lifecycle cost, which is your initial investment. And then we will essentially capitalize that with some interest rate k. Plus then, for each year of operation, you have the operational costs for that year you have to add. So that's your initial-- that's your development costs, your manufacturing costs for the satellite, including the launch costs. And then the ops costs would be operating your ground stations, your networks, any replenishment costs, they would be in your ops costs. And then you divide this by what's called here the number of billable minutes. That doesn't mean you're actually going to bill people, it means just they are potentially billable minutes. So that's the capacity of the system, [? see ?] the best times you know, 365 days, times 24 hours, times 60 minutes, times what's the load factor. So the load factor is essentially the capacity utilization of the system that you anticipate.

OK. So that's the basic equation for calculating CPF, cost per function. Plug-in some numbers here. These are the numbers that have been assumed, you know a $3 billion investment, 5% interest rate, $300 million per year of ops cost, 15 year life cycle-- this is the capital T, over 15 years. 100,000 channels-- that's your capacity. Number of users in this case, so the load factor is simply the number of users times the average activity per user. In this case, it seems to be 1,200 minutes per year, about 100 minutes per month. OK? So that gets you-- that gets you a CPF of $0.20 per minute. And based on that you can-- the business case was made

based on these kind of numbers. And so 3 million users, just 3 million subscribers, right? Not 3 million users at the same time, that number would be much smaller. That number can't be bigger than 100,000 because that's your capacity. So for example, if you run-- if you run a fitness club, right? At any given time in your fitness club you can have 50 or 100 people actually working out, but your number of customers, your subscribers, should be 1,000 or 2,000. And if they all show up at the same time you're in big trouble, right?

So that's the difference between number of users or the number of subscribers and number of active users at any given time. That's a big part of managing these kind of systems. However, what actually happened is the number of actual users grew much slower. So if you plug-in some different numbers, let's keep all the numbers the same, except for the subscriber base-- the number of users. In this case we're going to assume 50,000 users instead of the three million. This is closer to what they had after about a year of operation. Now your CPF goes to $12 per minute, which is noncompetitive. Right? Except for some extremely, like, military applications or making some emergency phone calls on an oil rig, you know, on the ocean. Most people at the time-- now and at that time-- would never pay that for one minute of service.

And so that was the fundamental problem, is that the user base did not materialize as fast as planned. Therefore this cost per function was way higher and they did charge, you know, $3 to $5 per minute of usage of the system, which as you try to squeeze your existing users more, you're not going to get the ramp up and scale up in the system that you need. That was the fundamental problem with the economics of the system.

So let me talk a little bit about the design decisions the conceptual design of what this design space looks like. So fundamental-- oh, the other thing I should mention to you, what was interesting is after the bankruptcy, both of Iridium and Globalstar, both of the chief engineers for both of these systems took refuge at MIT. Essentially, they came to like-- Joel Schindall, who was the chief engineer for Globalstar, great guy, both very competent people, came to MIT-- still there, still a professor there. And then Ray Leopold was one of the three architects for Iridium, also came to MIT during the time. So I had extensive discussions with them, and what you see here on slide 25 is one result of those discussions. Which is, the key design decisions that they have to make, fundamentally, there is-- you won't be surprised to see this, this is the magic number seven, right? Seven key design decisions. When you design a satellite consultation for communication purposes, constellation type-- polar or walker, orbital

altitude, minimum elevation angle above the horizon-- that you can communicate-- satellite transmitting power, the size of your primary antenna, your multi-access scheme. So this is time division or code division multiplexing. And then the last one is about the network architecture-- do you have inter-satellite links, yes or no? Inter-satellite links means satellites can talk to each other. And Iridium chose that, Globalstar did not. So in Globalstar, the Globalstar satellites cannot talk to each other directly in space. They can only talk-- it's a bent pipe system to a ground station. OK? So if you this is like the morphological matrix that you learned about, you just pick design decisions in this morphological matrix and you come up with an architecture. In fact, a full factorial search of the space would reveal 1,440 different satellite architectures.

So that's on the input side, what's the output vector? What do you care about in terms of output of the system? Well first of all, performance. So the performance-- this is for voice communications. In a sense it applies to data communications as well. It is your data rate per channel, right? 4.8 Kbps, your bit error rate, like what is the average number of bits that are wrong? 10 to the minus 3. That's actually pretty-- not a very-- that's not a very stringent requirement. And the reason is this is for voice communication, this is not for sending, you know, commands to a spacecraft going to Mars. If you send commands, it would have to be 10 to the minus 10, or 10 to the minus 9. Much, much better bit error rate. But this is OK for voice. And then the link fading margin, 16 DB. This is the strength of the signal, which which will dictate whether or not you can use the phone under trees or in buildings. So you want this number to be higher, but the higher it is, you know given the power you have on the satellite, the fewer channels you have. So there's trade-offs.

So what was done here is to keep the performance foxed, and so that you can compare architectures, compare apples to apples. Then we have capacity, which is the number of simultaneous duplex channels. And then finally, the lifecycle cost of the system, which includes research development, test, and evaluation, satellite construction and test, launch and orbital insertion, and then the operations and replenishment. So if the satellite fails in orbit, either you have to already have prepositioned to spare, or you're going to launch a replacement. And this actually happened in both cases, both constellations.

So in order to then, connect the input to the output, you need to build a simulator or a model of this system. And this is a high-level view of what that model looks like. So we take our input, our design decisions, and certain constants that we assume-- a constant vector. And cascade

this information from the constellation module, takes the altitude and the minimum elevation angle, and produces-- t is the number of satellites and p is the number of orbital planes, so this is sort of orbital dynamics. The spacecraft module calculates the satellite mass, the satellite network builds essentially the communication pathways. The link budget will calculate the capacity of the system. You know, given those performance constraints. The launch module will determine how many launches do you need, from where. And then the cost module basically calculates the total cost of the system, the lifecycle cost, and finally you get essentially a trade-off of lifecycle cost versus capacity of the system.

Let me just show you some, you know, you say, well that's tying together a lot of information into a multi-disciplinary model. So what kind of information do you have? You have a mix, really of physics based models-- so this is a very well known equation, the eB/N0, that this is the energy bit over noise ratio. This is a closed form physics-based equation that tells you how much energy is there per bit-- what's the signal to noise ratio on a per bit basis. And this is a function of transmitter power, receiver and transmission gains, various losses in the system. And then some of the equations-- some of the information is empirical. For example, the relationship between spacecraft [? wet ?] mass and payload power. You know, if wanted to have a closed form or you wanted to have more detail, you would have to almost build a CAD model, right? A separate model. You'd have to-- for each of the 1,440 architectures, you'd have to manually construct an individual detailed design. And that's not feasible, so what you do instead is you use some prior data-- and you can see here a scaling relationship. It's not perfect, but we have error bars, we know how good it is. Satellite mass, satellite wet mass is a function of transmitter power and, in this case, propellant mass. So two kinds of equations.

Benchmarking. So once you have this model, you need to ask the question, how can I trust this model? Does it give me a reasonable answers? And in this case, benchmarking the process of validating simulation by comparing the predictive response against reality. So just quickly here, showing you four kinds of data. So one is the simultaneous channels of the constellation, this is the prediction of capacity. You can see it's pretty good. In this case, the model is actually a little conservative. So the blue bars are the actual planned capacities. The red ones or magenta is simulated, and you can see that the simulation under-predicts, slightly, the true capacity of the system.

Life cycle costs, you saw that Iridium was just a bit more than $5 billion. Globalstar was between $3 million and $4 million. And so here we're in the right ballpark. And then in terms of

satellite mass and the number of satellites in the constellation required, we matched that very closely. And the reason for this is, fundamentally this is just geometry. Right? If I can tell you the altitude and the minimum elevation angle, and I'm telling you I need I need global coverage and I want dual redundancy, so you can always see at least two satellites, it's just geometry to be able to figure out how many satellites and how many orbital planes you need. And that's why the model, or the simulation, and reality match very, very closely. So this gives you some confidence that this model is reasonable.

So what you can do with it is now what we call Trade Space exploration. Which is, in a sense, what you did for the [INAUDIBLE] competition. So this picture here, this graph shows you the life cycle costs over those 15 years, versus global capacity of a system. Each of these blue dots represents one of those 1,440 architectures. And you can even see on this, where Iridium actual versus simulated falls and where Globalstar actual versus simulated falls. So both of them were actually off the Pareto frontier, which was interesting and led to quite some discussion. And you can-- the Pareto frontier itself was, of course, very useful. One of the reasons that Iridium is not on the Pareto frontier is that fundamentally, polar constellations are inefficient. And, if you think about it, when the satellites are crossing over the poles, they're very close to each other. And they're actually not crossing exactly over the poles, or you could actually collide, right? So you offset them slightly to avoid collisions. But some of the satellites actually get turned off when they cross the poles. So you're not utilizing your assets super efficiently in the polar constellation. Which is one of the reasons why they are not on Pareto frontier.

Now, the way you would use this in a traditional system engineering approach, which is traditional system engineering approach means give me the requirement, write down the requirements, and then find the minimum cost design for that requirement. So a requirement could be we need a capacity of 50,000, for example, and then you find this intersection with the Pareto front, and that's the minimum lifecycle cost design that gives you that capacity. Right? And that's the system you pick, and that's what you go and build. And that's essentially what they did. The problem with this if there is high uncertainty is, the true requirement is kind of unknown. The market will tell us in the future what the true capacity should be. So how do you deal with this? Well, you could be-- we're going to be on the safe side, right? We're going to oversize-- so if the true demand, the true capacity that you would need, is higher than what you just had, your system is going to be under capacity, right? Your system is going to be undersized for what it should be.

And so what happens in practice? What do you think happens in an under capacity situation? What would you say? The system is too small relative to the market that you're in. You're not capturing as much as you can, meaning a competitor will probably capture. I mean, I guess you can change your price policy, but fundamentally, you're missing out on market opportunity. The other situation, which is actually what happened, is that the demand is much less than we anticipated. Your system is oversized and remember, once you launch a satellite constellation, it's not like a fleet of taxis, right? Where you just park them in a parking lot. You've already-- the fixed cost is very high.

So if demand is below capacity, you have all this investment here, lifecycle cost has been wasted because you oversized the system. And the challenge is that the true requirement is kind of like this-- there's this probability, density function. Right? It's a probabilistic requirement, essentially. And there's no set of experiments because of all the [INAUDIBLE] uncertainty. It's not a systemic uncertainty. There's no set of experiments you could do today that will help you refine the requirements. Now they did do market studies, but again those market studies were years ahead of when the actual system was launched. So they were unreliable, fundamentally.

So what I'm arguing here is in this kind of system where you have large uncertainties as to the true requirements, you shouldn't just guess and then put billions of dollars on a guess. This is like playing in the casino, right? Essentially. What you should do is think about the problem differently. And in this case, the answer is staged deployment, or one of the answers is staged deployments. So you build a system such that it can adapt itself to the uncertain demand. So you build, initially, a smaller, more affordable system. But-- and then the system has already built into it the flexibility or scalability to expand if needed. But only if needed. And there are two major economic advantages to that. One is that you don't need to spend all the capital up front. As so you're deferring capital investment. And you retain the decision freedom to reconfigure or not, and that's typically what we call the real option. You've created a real option for the future.

So the question, then, is well how valuable is that? It is worthwhile doing this staged deployment approach? So there's probably different ways of doing this, and I just want to share with you-- and this is from the paper-- how this was done. So step one is you partition your design vector. So you basically decide which part of the system is flexible-- because you can't make everything flexible typically-- what part is flexible and what part is fixed, is the base,

you can't change it? So in this case usually the idea is that the satellites themselves, the design of the satellite, the transmitter power, that their protocol, and the network architecture should be fixed, right? Those things are difficult to change. We're going to allow the astrodynamics, the actual shape of the constellation, to be flexible. So keep the satellites the same, only change the arrangement in space.

So what you do is you partition the design vector into the flexible and inflexible parts. And when you do this, in step two, what's nice about this is you can then actually find families of designs, right? You can find families of designs where the-- we have a little lag here-- that you can find families of design that share the constant parts. That have the common-- So what's shown in this graph here-- this is again, our design space, kind of zoomed in a little bit more-- every one of these points that are connected with these lines uses the same types of satellite [INAUDIBLE]. Same transmitter power, same antenna diameter, and all of them have inter-satellite links. So effectively, you're partitioning the design say into subsets that share common features, and then we call this a family of designs. And the idea is to start small, so on the left side, and then you grow the system over time right? But only if needed.

So when you do this, you 1440- strong design space turns out to be decomposable into 40 different paths. Now, which path would you choose here? Well you want to be as close to the Pareto front possible. So this is an example of all of a path. In this path, if we started on the lower we would start with 24 satellites and then we would actually move them to a lower altitude. And add 30 more in step two, and then you would gradually grow the constellation over time. You see that? This is actually not too different from how GPS was done, right? The GPS constellation was deployed in phases, but it was not market driven, it was just sort of phasing the development for risk reduction, and spreading off the capital.

The other thing that's interesting here is that as you grow the constellation, it moves further away. It moves further away from the front. So when it's small, this particular [INAUDIBLE] is close to being optimal, and as you scale it up it becomes more sub-optimal. And there's other paths that have the opposite behavior; where when small it's kind of suboptimal, and as you make it bigger, it gets closer to the optimality. Pretty interesting. So that's step two, find the paths.

Step three, you now have to model the actual uncertainty. And in this case there are different ways to model uncertainty. In this case, what we used was a GBM model-- geometric Brownian motion. This is well-known in physics and applied to statistics, physics, this has been

applied to the stock market, right? And the idea here is that in this case, the man behaves like a particle in a fluid. Right? That's kind of moving in an unpredictable fashion. So this is the basic equation for-- this is a discrete version of geometric Brownian motion, you have some trends nu, nu times delta-t, so this is the delta and demand, or the change in demand, divided by demand, so this is normalized change in demand, is nu times delta-t, so your trend times delta-t plus sigma times epsilon times square root of delta-t. Epsilon is a standard, normally distributed random variable between 0 and 1, and epsilon and sigma is your volatility, essentially.

So here's some examples, if you start with an initial demand of 50,000, which is what they actually saw as an initial demand early on, you have a growth of about 8% per year. This is your trend. Plus in this case, a volatility of 40% per year, which seems high but in this kind of very new technology, new system, is actually not that far fetched. You get three different-- these are just three examples of how demand might evolve. These are just three scenarios.

So demand can go up, it can go down, and-- GBM is very nice, but one of the downsides of GBM is there's infinitely many scenarios that you can generate because it's fundamentally-- even though it's discretized in time, it's not discretized and in demand. So a simpler version of this, a more discrete version, is a so-called binomial lattice model. So the way this works is again, you start with some initial demand here on the left and then you discretize this such that moving through time, you can look at different scenarios. Right? These scenarios could be-- the best scenario is things just keep growing. Right? Grow, grow, grow. Here, time, the 15 years, have been sub-divided into five three-year periods.

So here's a sample scenario, demand goes up, goes up, and then it goes down twice, and then it goes up again in the period. And so this is a discretized random walk. And you can choose the numbers, so the sigma, the volatility, the amount of the up and down movement, and the probability, p, of going up, and then the probability of 1 minus p of going down, are chosen such that they're consistent with the GBM model. So this model is the equivalent of the GBM model, it's just discretized. So the beauty of this is that you can now simplify this to 32 different scenarios instead of infinitely many. And each of those scenarios is not equally probable. There is actually probability weighted scenarios, depending on what nu and sigma are that are underlined.

So now we have 40 paths, right? We have 40 evolution paths of the system and we have 32 different future scenarios. So what we do now in the next step, four, is to put the two together

and calculate the cost of each path. Calculate the cost of each path with respect to each of the demand scenarios, look at the weighted average of all the possible paths, and the one thing-- the one tricky thing about this is you need to build in a decision rule. Such that, if the system-- if demand exceeds capacity the system, you're going to expand and move to the next stage, you're going to stage deploy the next phase. And there's many different ways of doing this decision rule.

So the simplest one was chosen here, and of course costs are discounted. So let me explain to you how this works. So we start the simulation, so the first initial deployment. This is our initial stage one-- right-- for the constellation. And then we start operating for two years, and demand in this period goes up. You see the ops cost in this case goes slightly down, this is due to the discounting. This is a discounting effect. We arrive at the end of our first three year period and then the question is here is our capacity of the system. What would you do in this situation? Let me ask somebody at MIT, make sure you guys are still with me. So what would you do in this situation? You're now at the end of year three. Anybody?

AUDIENCE: We lost you for a second halfway through your statement, could you repeat?

PROFESSOR: Yes, so you've done your initial three years, you've deployed your initial constellation. And you now have the situation shown on this chart. What's the right decision, according to the decision rule? What's the right decision?

AUDIENCE: OK, I'd keep it the same. Because--

PROFESSOR: Yes, exactly.

AUDIENCE: --it has not exceeded demand yet. But prepare to change--

PROFESSOR: Exactly, you wait, right?

AUDIENCE: Because you have to lower your capacity.

PROFESSOR: You keep it the same, you don't do anything. You just keep operating. Exactly, right. So you have another three years, right? During these three years, demand keeps growing. And you now, at year six, arrive at this point here. So what's the right decision now?

AUDIENCE: Can I ask a question about the decision to stay the same?

PROFESSOR: Would you say the same? What's happening to the demand line? Veronica, what's happening

to the demand line?

AUDIENCE: Can I ask a quick question first, about the decision to stay the same?

PROFESSOR: Yes.

AUDIENCE: I feel like by waiting until we've exceeded demand to choose to expand the constellation, we're introducing a lag between demand and capability that creates an inefficiency that may actually drive users away from the system. And I see this kind of oscillatory effect, where you would expand, and then people would meet, and then the system would be insufficient and then people would move to a different system. And you kind of have a yo-yo around the maximum carrying capacity. And that doesn't seem efficient to me, so I'm wondering if you could speak to that.

PROFESSOR: Yeah.

AUDIENCE: So what you're essentially looking for is what I would [INAUDIBLE] decision rule. So you would actually-- so the decision here is to deploy, right? You've got to deploy your second stage because you've saturated the system. What you're arguing for is before this occurs, you know like at 80% saturation, that's when you trigger the next stage, right? In order to anticipate the saturation. And so, absolutely, you could do this. There are many different decision rules that you can try when you design a flexible, deployable, scalable system. And that's part of the decision space. So in this particular example, we just implemented the simplest possible decision rule, which was when saturation has occurred you do deploy the next stage, but not before.

AUDIENCE: OK, thank you.

PROFESSOR: Is that clear?

AUDIENCE: Yes, thank you.

PROFESSOR: OK, so we deploy the second stage now, and you can see there's another spike here-- right-- of capital that's needed. It's not as high as the first initial stage, but it is substantial. So now as we deploy the next stage, our capacity went to the higher limit. So we're now at capacity level two. We're now operating the system. Demand keeps growing, that's good. Now we have year nine, you can see. But we haven't yet saturated our new capacity. So again, the optimal decision is-- or I shouldn't say the optimal decision, but the decision according to the rule is

you just wait, right? You keep operating.

Ah, now from year nine to year 12, demand starts to go down. Right? And this happens in some systems, right? They are growing and they peak, and then things go down. So in this case, we wait. And then in this scenario it goes down again. You see how that works? So what's nice about this is you can now-- in this sense, all the 32 scenarios of possible futures, you can run them against the 40 different families of designs or evolution paths. And out of that you can find the best path, right? This is the path, the path that on average-- I should point out on average-- will satisfy your demand at minimum lifecycle costs, given the uncertainty models that you have composed. Yes, [INAUDIBLE]

AUDIENCE: [INAUDIBLE]

PROFESSOR: Why did it go down?

AUDIENCE: Like, [INAUDIBLE]

PROFESSOR: This here?

AUDIENCE: [INAUDIBLE]

PROFESSOR: No, the capacity stays the same. You don't go down in capacity. The reason is-- this is a good question-- this particular system, a satellite constellation is dominated by fixed costs, right? So if you wouldn't retire half your satellite, because then you would lose coverage also, potentially. Because you've moved them through orbits that give you the right coverage, you just increased your capacity. This is different from a system that is dominated by variable costs. Like for example, if you operate a fleet of taxis and and for whatever reason, demand goes way down, you can go and park half your taxis and just not operate them. You may still have to pay a lease on them, but essentially you can downward that the capacity of the system. You can't do that here, it doesn't make sense. Right? So that's the big difference between systems that are fundamentally fixed cost dominated versus variable cost.

So in this case, we just don't use it as efficiently in the last six years. The answer here is that there's an optimal path, and that's shown here, that will, for a given targeted capacity, you can compare essentially this blue path against the traditional fixed design. And in this case, the traditional architecture-- fixed architecture-- would cost about $2 billion dollars to build, both and on average the red point is the average lifecycle cost of the evolvable system, is 1.36.

Right? Lifecycle cost of the rigid design versus the expected lifecycle cost of the best deployment strategy or staged deployment strategy, which is-- essentially the way that's calculated is the probability weighted lifecycle cost of each of the scenarios, right? Against this path. And the difference between those two numbers is about a third, $650 million. And that is the value of the real option. That's the value of designing the system with scalability, with flexibility. OK? Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yes. That's true. So the lifetime-- the question was, is the life expectancy similar? Of course-- so the 15 year life of the whole system is the same for both. Of course, in the staged deployment strategy, some of the satellites are going to be younger at the end of the 15 years. And so they may have longer residual life left, which is actually not included. That's not even included in this real option value.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Right.

AUDIENCE: [INAUDIBLE]

PROFESSOR: That-- that's right. So you could sort of refine this model to include a more-- a staged decommissions, right? Or staged transition to a next generation system. So in this case, in this model, after 15 years-- boom-- you know everything, you just finish. No more revenues, you just decommission. A hard end, a hard stop after 15 years. But you could actually build transition and decommissioning models as well. Good point.

So that's essentially the case study that I wanted to show you. And, you know, you say that's kind of like in the US we talk about Monday morning quarterbacking, you know? You come in and all the football games that happened and the mistakes that the coaches made, and I would have done this or I would have done that. So yes, you know, this is sort of like looking at this problem in hindsight, but the reality is that this really has had a big impact and a big dampening effect and that the new generation of systems-- I think-- are built much more intelligently with this kind of evolution in mind.

I will also mention to you that Iridium is actually the system that went bankrupt first. And the reason for the bankruptcy-- well, one of the immediate reasons for the bankruptcy-- is because of the way that the project was financed. So about one third of the funds-- the $5

billion-- one third of the funds for Iridium came from equity from Motorola. So that was their own money that they lost. About one third was an IPO, initial public offering, you know, shares sold to the public. And about one third was bank loans. And these bank loans were relatively expensive and the banks expect that they get paid back at a certain speed, depending on the market evolution. And it's essentially the inability to pay for and service their loans, that caused the Iridium bankruptcy. And it turns out that the loans were about one third of the total capitalization of the project.

So what I would argue-- I would argue that if they had done this more flexible, staged approach, they could have saved one third, which is about the value of this option and capitalization, and just build the system only with equity and with the money from the IPO. And that would have given them a lot more time, you know they wouldn't have had to service those loans and it would have given them a lot more time to wait for the market to develop, which it eventually did. So the whole financial architecture was poorly done as well. It's a whole other-- that's a whole other question.

AUDIENCE: Professor, quick question?

PROFESSOR: Yeah.

AUDIENCE: Are there situations where [INAUDIBLE]

PROFESSOR: Yes. So the question, I'm just going to repeat it for you guys at MIT, the question was are there cases where the flexible approach is not the best? And the answer is yes. So for example, if you're in-- I know that you're studying the energy market-- if you're in a situation where you have guarantees, a guaranteed customer, so the government or one of the big utilities is entering into a long-term agreement with you, that they will buy the power that you produce at a certain price for the next 20 years. And so you're not subject to that market volatility, but you have a long term agreement, including clauses that say well if the government doesn't do this then they'll compensate you for any losses, this is a financial contraction.

If you basically have eliminated demand uncertainty, because of the particular contractual arrangements, then there's no reason you shouldn't build the system for a fixed capacity. And then you can build something right on the pareto front, because that particular uncertainty has been eliminated for policy or contractual reasons. So, good question.

OK, so we're almost there. Let me just try to summarize some--

AUDIENCE: Can I ask you question on that last slide?

PROFESSOR: Please, go ahead.

AUDIENCE: On the slide before, 37. It seemed like the probability weighting was for a general capacity. Like something that could be adapted over the life cycle but here we're pinning down a specific capacity. So why is it the sum of probabilities, then? Haven't, kind of, the probabilities been realized, don't you know exactly what your capacity is that you're operating at?

PROFESSOR: Yes, so just to be clear on this. So the probability, those p's of i's, they come from the lattice model. They are determined by the lattice model. So what you do is for those 32 scenarios of the future, this is the pi, right? The i is the index, is the-- for this particular future scenario, for each future scenario, you always start with a1, which is your initial configuration of the constellation. You always start with a1. And for each of those future scenarios, you then simulate that future and for those futures, where the demand doesn't ever really take off and materializes, you never actually trigger that next expansion stage. And it's actually that asymmetry that gives you the advantage. And it's essentially not deploying capacity when you really don't need it. And so you do this for all of this n scenarios for the future, and the pi, the probability, weighting of each future scenario comes directly out of the lattice model. That's not something that you have to manually determine.

AUDIENCE: Right, no, I understand that. But here, since we've kind of pinned down a capacity already, we know what endpoint we're at. I'm just wondering how the probabilities come into that, or is it kind of working backwards now, of all the paths that go there on this specific plot?

PROFESSOR: I see what you're saying. OK. So the last point, the a1 point, so the a1 point is driven by-- you need to make some assumptions about initial capacity and then the endpoint, the a4, is driven by how large you made the trade space. So you could make the trade space, you could make the design space even larger, right? This is a discretized full factorial design space, you could make it even larger. And then a4 would not be the end form. This is just sort of a finite size effect, given to the size of the trade space.

AUDIENCE: All right, thanks.

PROFESSOR: All right, so the way I want to do the summary here is just go back to the learning objectives.

You know, I always do this in every class, you know, let's close the loop. What did I promise you in lecture one that you would learn in this class? And, you know, each of you, you're going to have to decide for yourself, you know, did I actually learn this? Did we actually do this? So this is sort of due diligence on the learning objectives. And I have one slide on each of them. So the first objective was to-- for you to really understand the system engineering standards and best practices, as well as more approaches. Number two is understand the key steps in the system engineering process. Number three was analyze and understand the role of humans in the system as beneficiaries, designers, operators, and so forth. Number four was being honest and characterizing the limitations of system engineering as we practice it today. And then objectified learning and objectifiable was applying these methods and tools to a real-- even if not so complex-- cyber electromechanical system.

So I see one, essentially, hopefully-- no, we didn't check the readings but I'm hoping that you did your readings, that you really feel like you understand the NASA System Engineering Handbook, which is our quasi-textbook for this class. But there's other standards as well, such as the [INAUDIBLE] handbook, which is actually the basis for certification. So several of you-- your [? EPS ?] fellow mentioned to me that you're interested in certification, so I encourage that. Most of that is based on the number two. I did mention the ISO standard 15288. That's probably the best known standard; ISO is a very-- they're located right here in Geneva. ISO is a very important organization, you know? I know ISO standards are not the most exciting things to read, but they have huge impact and so 15288 is something that worldwide-known and across all industries really.

And then I want to mention also the European Systems Engineering Standard. And Foster Voelker and I had a discussion about this earlier today. This is issued by the European Space Agency. And it is a little-- there are some subtle differences between ISO approach and the NASA approach. And so for those of you on this side of the Atlantic, I do encourage you to take a look at that [? ESES ?] standard as well. And then we augmented this with different papers, and readings, and so forth.

So I want to do a very quick-- this is our last concert question for the class. So here's the link, [INAUDIBLE] and it's essentially I want to get some feedback what you think about the standard. All outdated and surpassed by the digital revolution, one of you thinks that's true. 90% of you think they're useful codification and best practice. 14% think they can be dangerous if you adhere to them too closely. About 60% of you think they're essential reading

for any serious system engineer. And 10% said that you would never use them as a daily reference book.

VOELKER: OK, so that's good. I think that's-- what do you think, Voelker, pretty reasonable outcomes? OK. But what is true is that they're going to be referenced, in contracts. You know if-- if you're doing-- you know, it depends, again, on the industry that you're in. But you're going to adhere to some of these things in contracts, and if things don't go well, people will actually check whether, did you do this step, did you do that step. So it does have real consequences.

PROFESSOR: OK, great. Thank you very much. That was that was good. Step two, you know the key steps, v model, I think I'm not going to explain it again. You know, hopefully this is something that you will not forget. And just one point about the v model is it does not imply that system engineering is always sequential. You know, it's possible to iterate between steps and across the whole v as well.

SE3, stakeholders and value of network. So this is the role of humans. We talked about this very early on, several of you talked about it during the oral exam here today. You know, the hub and spoke versus the stakeholder value network. The method underlying this is critical, but I do want to mention, very quickly-- and this is something we didn't spend a lot of time on-- is human factors. Right? How do you design systems so that humans can effectively and safely use them? You know, interfaces, procedures, and so this is a couple of slides from one of my colleagues, Missy Cummings. This is actually some-- in a nuclear plant-- Katya, you're going to like this-- lots of these dials, and this is actually a Russian nuclear plant. So it has it has a very specific layout.

And so the important questions in human factors are, how do you best display status information? What tasks do humans do? What level of automation? What are the training requirements? And these are also very important questions. And so you can apply the human systems engineering process, very much analogous to what we covered in class so far. So you do mission and scenario analysis, function allocation. And then in human factors, you talk more about task analysis. But essentially, eventually, leads to system design. Like what are the buttons that you push, what does the layout look like, or the user interface, and so forth.

So, you know, I'm not I'm not presenting human factors as a separate topic, but the human factors requirements should be built in right from the start. And essentially the functional allocation is sort of a key question here. How much is done by automation, how much is done

by humans, and how do you split between the two? And one of the most important things to consider here is this human performance curve. And we know this pretty well now, that when humans are extremely highly loaded-- you know, when you have a huge workload-- your performance goes down.

But interestingly, if you're under challenge your performance goes down as well. And you know, so for example, people who are in power stations or mission control centers where nothing is happening for days and weeks, you know their attention goes down, and they don't perform at a very high level. Humans perform best at a moderate workload, and this is well-known. And it should influence how you define the human interface and the split among automation and humans.

SE4, system complexity. Last time I will mention the magic number seven, plus or minus two. The real limitations come in and when we operate at levels, you know, three, four, five, and six. And the main reason for this is because now a single human cannot understand, cannot remember, cannot deal with all the detailed components. And you need to split the work among teams, among organizations, and creates extra interfaces and complexities. OK? And just to show you, Iridium, we talked about Iridium. This is a very recent news story this is from October 29th. The new Iridium Next has again been pushed back by four months. This gentleman here is the CEO of Iridium, chief executive for Iridium Next, Mr. Desh. And he's talking here about a particular [INAUDIBLE] and then Viasat is one of the contractors. There's a particular component that's been giving trouble, and that's sort of in level three or four in the hierarchy, and that's essentially delaying the whole system. So there's an example of how the large complexity of the system is causing issues.

AUDIENCE: [INAUDIBLE]

PROFESSOR: But they're actually a supplier to [INAUDIBLE] through the whole chain, right? Oh you're saying there's more going on than meets the eye. Yeah, who knows? Who knows?

OK, and then finally here I want to mention application to a case study. You know, we used the CanSat 2016 competition as a quote, unquote safe case study. And I have to say I'm really pleased with how this worked out. You had the 47 requirements as a starter, you approved, them you group them, and my comments about the PDR that we had a week ago are all teams passed successfully, the PDR. You know, if this was a real PDR, we would have issued a couple of [? ridds, ?] I think. There was some-- a couple of teams were over budget, or

needed to work out their aerodynamics in more detail. But by and large I thought this was an excellent application of system engineering concepts. Went beyond my expectations. Several of you mentioned the importance of concept generation, you know hybrid use of structured and unstructured creativity techniques. I know at least one team applied to actually go to the competition.

And this shows you a couple examples. So we have one team here at the EPSL using a bio-inspired design. This is an actual seed airfoil that actually occurs in nature. And then here's an example from MIT, the Rogallo Wing Team 7. So, I'm sorry I didn't mention all the teams here, but it was really great to see the application of this in the CanSat competition case.

So the last thing I want to do-- and I only have one minute left, actually I'm already slightly over time-- but I just want to give you some career and study recommendations. So first of all, you know, I want to make sure that you-- and I said this in the first lecture-- there's a lot more to the system engineering and systems research than we were able to cover in this class. So this class is really what I call a door opener to the world of system engineering. If you want to go deeper, there are deeper subjects. You know, model based system engineering is the big trend. System safety. And then in the spring I'm teaching a class called Multi-Disciplinary System Optimization. And there is actually a WebEx access to it as well. So it's not going to be officially offered as a [INAUDIBLE] class, but if individual students here are interested, please contact me.

Self-study, you know the System Engineering Journal, there's IEEE journals, there's also-- you know, for some of you this is maybe a little too soon-- but there's professional degrees in system engineering. For example at MIT we have the SDN program, System Design and Management. The average age of the students in that program is 33, but it is-- they're sort of coming back to get their masters in system design and management. And there's quite a bit of SLOAN content as well. So finance, you know, understanding the financial-- how to build a business case around systems as well.

Professional experience. You know, getting experience in actual projects. Like at MIT, for example Rexus. You know, you've got the Clean Space One project, we heard about [? Optaneous ?] One, Solar Impulse, there are a lot of opportunities. Or-- we also had a couple of people here mention that you're starting your own company, your own venture. And you know, when you're doing, that you have to be the system engineer. You have to understand all of these things, interfaces, suppliers, requirements, markets, all that has to be integrated.

Finally, you did hear about INCOSE. We had a quick dial-in with at an earlier session. So if you're interested, you can join either as a student or a professional member. And also certification. This class was not [INAUDIBLE] for that, but if you're interested in this site, [INAUDIBLE] number of years of experience at the [? CCF ?] level. And then finally, please keep in touch. And last but not least, I want to thank all of you, the students at MIT, at EPFL, our TAs, [INAUDIBLE] here at EPFL, Johana at MIT, the technical staff has been helping run the technology, Voelker, thank you. And-- and that's it.

So I want to thank all of you. Next week we do have a voluntary seminar on sort of future trends in manufacturing, but again, it's not mandatory. It's going to be in the same place, but like I said, it's not part of the official class. So with that, sorry for running a little bit over time but it's been a joy to teach this class in this kind of new format.