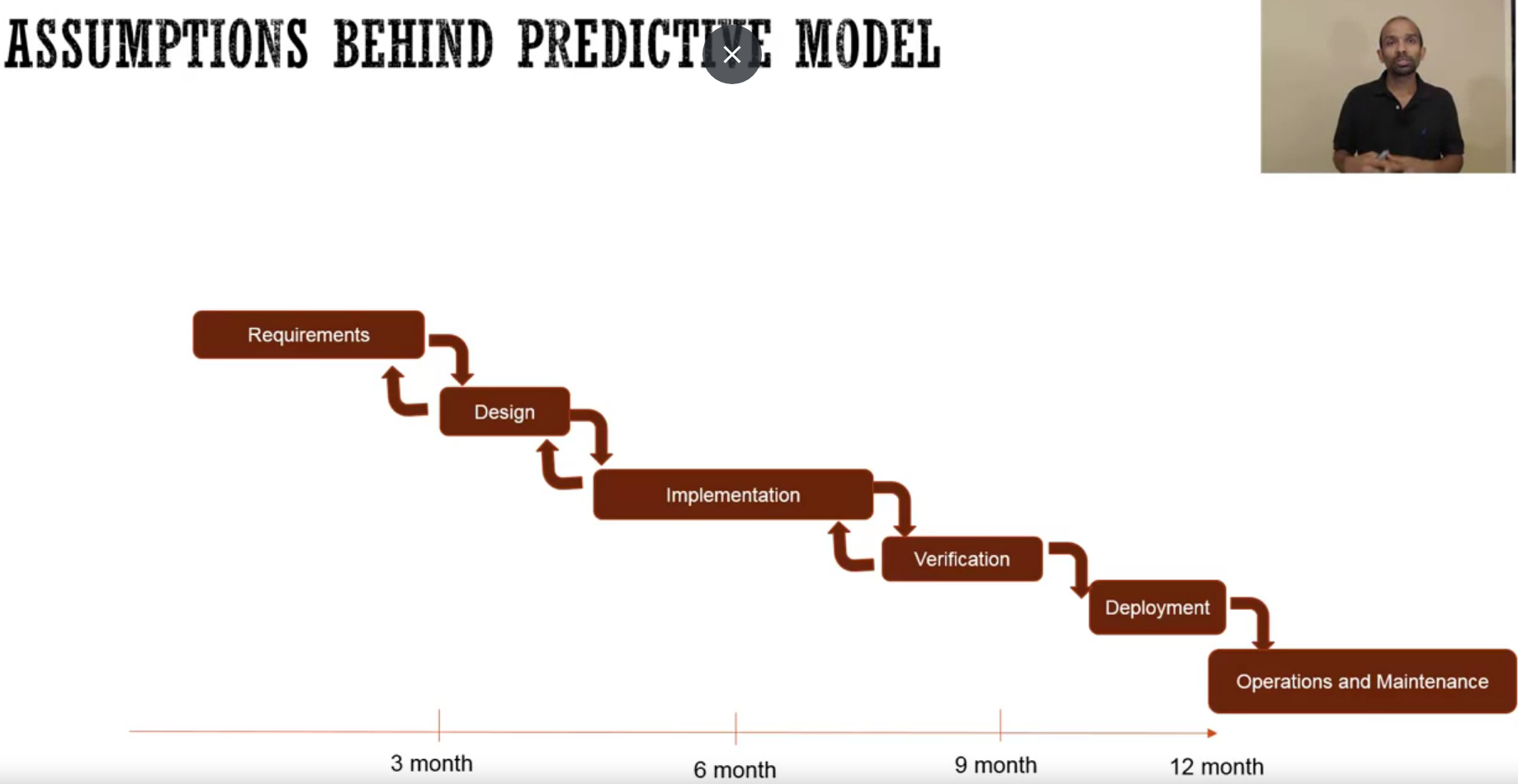
Before we learn more about Agile, the question worth asking is,

why Agile? What led to the emergence of Agile methods, or what problem were we trying to solve?



To answer this question, let's dig deeper into the Predictive model, or sometimes called the Waterfall model.

So, in this model,

1. we start with the requirements. We define all the requirements,
2. and then we do the design, where we design the whole system.
3. And then we implement the system,
4. and then the testing, or the verification.
5. And then finally, we deploy the system.

So, for this model to be successful, one of the assumptions we made was that we can predict the requirements accurately, and it won't change much.

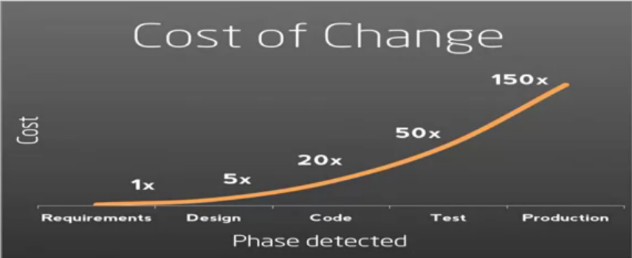
And why is this assumption very important? So, if you think about it, if the requirements are incorrect, the design will be incorrect, and then the implementation will be incorrect. And we'll get the product which is not what we wanted. And so, it is important that the requirements can be predicted accurately at the beginning of the project.

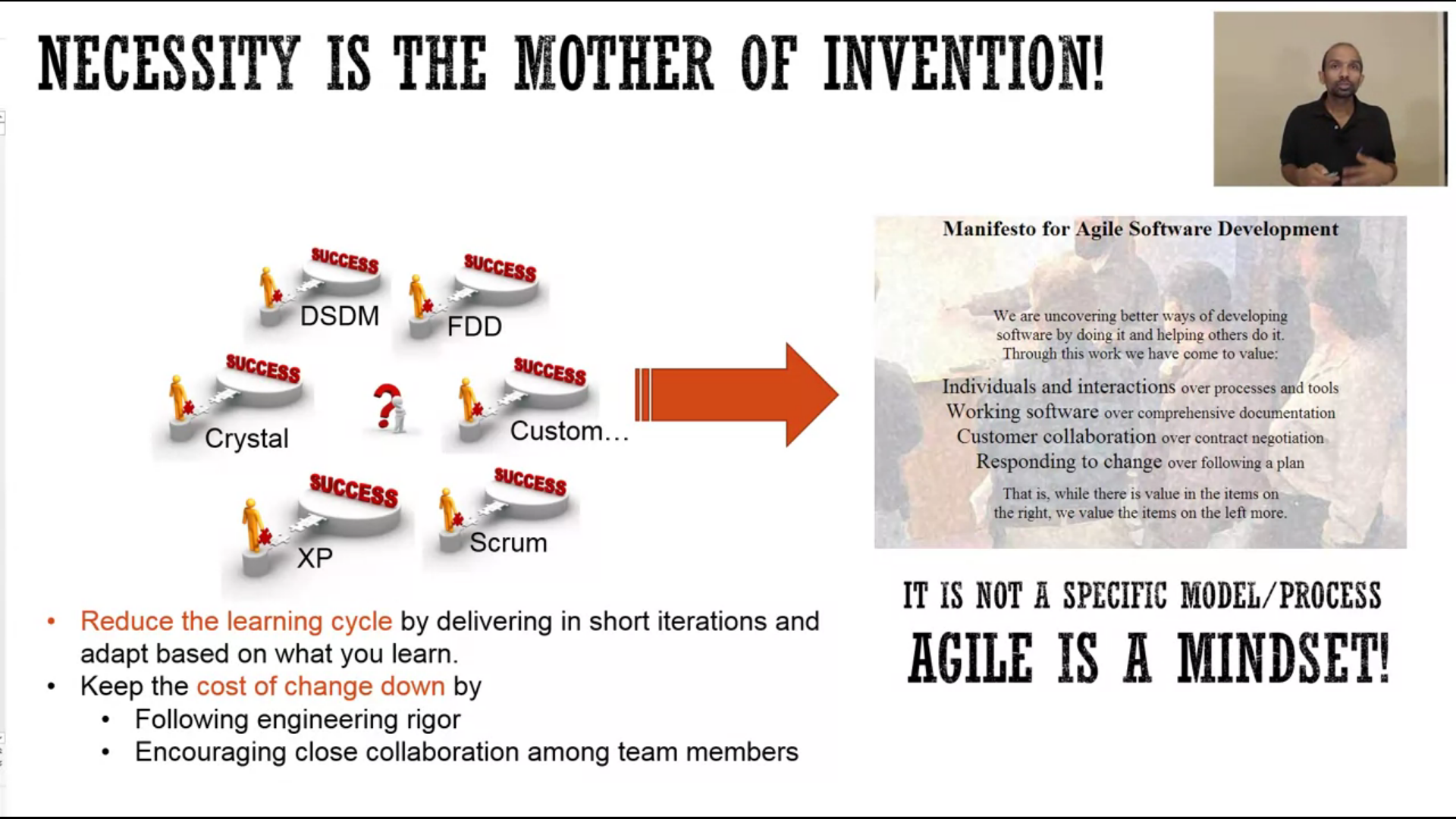
Now, what the industry was finding is that that is not the case. A lot of times, the requirements were incorrect, or what we designed is not what the user wanted. And there were several reasons quoted for that. But one is that, due to the intangible nature of the software, it is very difficult to predict the requirements early on. And then even if we get the requirements correctly, sometimes by the time we are done with the development, sometimes it's 12 months, sometimes even many years. By the time we deploy the system, the market has shifted. So, what was originally thought was needed in the market is no longer needed, or the user needs have changed. And so that's one of the assumptions.

The second assumption that was made, is that the translation is going to be perfect.

Which means that, as you can see from the requirements to the design, to the implementation, the translation is going to be perfect. But what was really happening, is that whatever was written, the requirements, the designers, or the architects understood it a little bit differently, and they designed a different system. And since there was no check, then it went into the implementation. And then they did another translation issue, and they built something else.

Now the problem was not that these translation issues were found, or these requirements were incorrect. The problem was the time we were finding these issues.

So, in this model, we were finding the problem very late. And as you know, due to the cost of change, that the later we find the problem, the more costly it's going to be. So, these issues in the industry led to the industry to look for new methods. And as these issues were going on, there were several teams around the globe who were trying different models, like Scrum, XP, Crystal, DSDM, and so on. And they were finding success in their methods. And the foundation of these methods was, they were trying to reduce the learning cycle. So, all these issues that we found in the predictive model, translation issues, or requirements were incorrect. They were trying to make sure that we can learn about those early on. So, they were trying to reduce the learning cycle by delivering in short iterations, and then adapt what they're building based on what they're learning. And the second thing they were trying to do, is to keep the cost of change down by following the engineering rigor and encouraging close collaboration. So, instead of those phases where the things are being handed off from one phase to another, to another group of people. They were encouraging close collaboration among the people, so that it can keep the cost of change down. And so, as these teams were successful, the leaders of these teams or industry experts.



They were thinking and saying, why don't we get together, and find out what is it that we are doing which is different? And that led to something very significant. It led to the development of Manifesto for Agile Software Development. And these individuals got together, in 2001, in a resort in Utah. And they skied, relaxed, and they also defined these four values, and twelve principles. As you can see, they didn't define a new process, or new practices. They said, this is the mindset that is common to the method that we find successful. And so, they defined a mindset that can help software development teams be successful. So, in summary, what the industry realized, is that software development is a creative process, and we need adaptive methods to be successful at it.

