



Introduction



Legacy Modernization: Introduction

How to deal with an old codebase



Let's start with an introduction to legacy modernization.





When Legacy Code is a Problem

Your old codebase is a problem.



We talk about legacy modernization when there is a problem with legacy code, so your old codebase is causing you a problem.





When Legacy Code is a Problem

Your old codebase is a problem, it is costing you money, it hinders opportunity... but is also giving you a lot of value. It might even be the backbone of your business.

Let's not throw away the baby with the bath water.



Now, your old codebase could be a problem for different reasons. It could costing you a lot of money. It could limit the opportunities that you are able to capture, but on the other hand, the same codebase is providing you a lot of value. It could even be the backbone of your business, so we need to find a way to save the value while throwing away the problems. In a way, we want our cake and we want also to eat it.





When Legacy Code is a Problem

- An old codebase can have a lot of value, but it is difficult to take advantage of the full value
- It has high costs (e.g., hardware, maintenance, development costs, etc.)
- It slows you down (e.g., inability to find developers, complicated changes, etc.)



So legacy code is a problem because, while your old codebase has a lot of value, this value is somehow trapped. It's not as accessible as you would like it to be.

It could have very high costs, for example, due to the cost of hardware, to the cost of maintenance, to the development cost, but also to licenses cost. It slows you down because you cannot develop your system as fast as you would like to because, for example, you cannot find developers or because changes are complicated, or because there are not libraries that you can just reuse, as is the case with more modern programming languages.





Legacy Modernization

The objective is to free up the value trapped in your old codebase



Our objective with legacy modernization is just to free up the value trapped in your old codebase. We need to understand that there is a ton of value in this codebase and a lot of problem, and we want to separate the two and free up the value so that we can really leverage this value within our organization.





Legacy Modernization

- The process of Legacy Modernization can help you free this value
- *Freeing* can mean reducing costs, releasing features quicker, launching new products or services, etc.



The process of legacy modernization is exactly the process to help you free this value, with the goal of reducing costs, launching new products or services, or adjusting features way more quickly and so on.





Legacy Modernization

Your legacy codebase constrains you: it gives you value, but it is hard to maintain and develop.

You can use Legacy Modernization to improve your codebase. The problem is that Legacy Modernization itself can be costly and confusing.



So your legacy codebase, and the legacy part in your legacy codebase, is what is constraining you. It's limiting the value that you can extract from your codebase because it's hard to maintain and develop. Now, legacy modernization per se is also not simple. It can be potentially costly, and it can be confusing.





Legacy Modernization

- You are unfamiliar with the process
- You do not know what you need and what solution to pick
- The uncertainty of the process makes it riskier
- You cannot find trusted partners



The main problem, in my opinion, is that you are unfamiliar with the process. Typically, you don't go through legacy modernization processes very quickly. For many organizations, they need to solve this problem once every 20 years, every 30 years, so it's a problem that is new to them. They have not the experience to easily solve this problem. And given you do not have the experience, it's difficult to know what you need to do and which different alternatives you should go for. And given there is so much uncertainty, the project is perceived as risky.

And what do you do when you feel that something is risky and confusing? You just wait and hope for the best, which is not exactly the best strategy. It's also difficult to find partners that you can trust with these, given that you don't have experience, so it's difficult to assess which partners really know how to do these jobs and who has the experience and the skills to solve your exact problem.





Legacy Modernization

The end result is that companies get stuck and do not improve their legacy codebase.

If they do start the process of legacy modernization, they are unsatisfied with the result and the process.



The end result is that many companies get stuck, do not improve their codebase, they just keep it as it is. Some others do start the legacy modernization process, but they end up being unsatisfied with the results and with the whole process.





So here, let's take a look around us and find the direction that we are going to take for the rest of the course.





Legacy Modernization

- No easy, 1-2-3 steps automatic solutions
- No simple, catch-all solutions



First of all, a disclaimer. Unfortunately, there are no easy solutions that work in every case, and we do not have a formula with one, two, three, or seven steps that you can just follow and get your problem solved. It's a little bit harder than that, and so this is also why this course is not just 10 slides, but it contains a lot of content.





Legacy Modernization

Legacy modernization is a process that operates on software that is highly valuable for your company.

Therefore it requires time and care to understand what is the best solution for your use case.



Now, legacy modernization is a process that focuses on software that is highly valuable for your company. This is software that has provided a lot of value to your company. And if it wasn't the case, you would've thrown that code a long time ago. But that code, you have kept it because yes, it provided value to your organization for a very long time until it eventually became legacy. And given this is so valuable, we need to be careful and understand what is the best solution to move forward with it, because this legacy codebase is probably a very important piece of your business, and so we should treat it with respect.





Legacy Modernization

In some cases, you will need to deal with an old language for which there are no developers...

Sometimes you will have to change a desktop app to a mobile app...

There is no unique approach that works for everybody.



Now, there are different situations that make your code legacy. In some cases, the problem is that you're dealing with an old language and you cannot find developers. In some other cases, the problem is with the paradigm used to write the software. Maybe you have a great application, it's just a console application, and now everyone wants maybe a desktop application, or it's a desktop application and everyone in your sector is moving towards mobile applications.

So depending on the situation, there are different approaches that we can consider.





Legacy Modernization

We are giving you the tools to understand what to do and a strategy to follow so you can build a solution that works for you.



With this course, our goal is to give you the tools to understand what to do and define a strategy to follow so that you can implement the right solution for your needs.





Legacy Modernization

This is similar to the work of a tailor. A tailor takes your measurements and then they creates a suit that fits you best.



The key is to find a tailored solutions for your specific needs, because under the term legacy modernization, there are so many different situations, so we should see in which specific situation you are and which is the perfect strategy for you.





Legacy Modernization

There are several techniques available, each one best fit a specific situation.

You will need to do some work to analyze your situation.

We will help you every step of the way.



There are several techniques that are available and we will help you through this process.





Course Organization

The course is divided into three parts:

- Analysis
- Diagnosis
- Resolution



The course per se is divided in three parts: the analysis part, the diagnosis part, and the resolution part.





Analysis

In the analysis section you will:

- Learn about the different types of Legacy Code
- Understand the effects of Legacy Code



In the analysis part, we will learn about the different types of legacy code and we will learn how to understand and examine and measure the effects of legacy code.





Analysis

At the end of the analysis section, you will know how to recognize legacy issues.



At the end of the analysis section, you will know how to recognize these issues brought by legacy code in your specific situation, and see which of the several potential problems you're actually experiencing.





Diagnosis

The diagnosis section will:

- Present you the 7 techniques used to solve your legacy problems
- Explain which techniques to employ depending on the legacy issues you have



In the diagnosis sections, instead, we will present the seven techniques that you can use to solve your legacy problem. We will explain the different techniques, and in particular specify which techniques can be used to solve which problem.





Diagnosis

At the end of this section, you will know what legacy modernization technique to use for your case.



At the end of the diagnosis section, you will be able to understand which technique you should use.





Resolution

The resolution section explains how to solve your legacy issues:

- We will provide general guidelines to successfully implement a legacy modernization project
- We will discuss how to implement each of the 7 strategies



Finally, in the resolution part or resolution section, we are going to see how to actually apply these techniques and solve your problem. We will provide general guidelines to successfully implement a legacy modernization project, and we will discuss the specificities for implementing the seven different techniques or strategies.



“ Resolution



At the end of this section you will have a roadmap to follow to deal with your legacy issues.



And at the end of the resolutions section, you should have a roadmap that you can follow to actually solve your legacy issues.





Why you should trust us

We are a company specialized in language engineering solutions.

We created this course because companies came to us to solve their legacy code issues.



Now, why you should trust us? Well, we are a company specialized in language engineering solutions, and so this is what we do for a living. This is not just problems that we solve once because a random client asked us to look into this. We do that regularly, and we created this course because we are in contacts with many companies that have problems with legacy code, and so they reach out to us.





Why you should trust us

We implemented legacy modernization solutions for many years and many companies.

We helped Fortune 100 companies and small businesses, from America to Asia.

We worked on different technologies and we keep solving legacy problems every year.



And we have worked with them implementing legacy modernization solutions for many years, for many companies, from Fortune 100 companies to small businesses, from America to Asia and Europe and Africa.

We didn't yet work with people in Australia or New Zealand, but hopefully we will do that soon. Now, we work using different technologies and different solutions to solve these problems every year, and so we have a broad experience in this field.





Why you should trust us

We have seen the issues and needs that companies have.

We have seen fears and hopes of companies in dealing with a legacy codebase.



And so, sometimes we act as psychologist for these companies because the companies reach out to us because they have this problem, they're confused, they don't know how to deal with it, so we sit down, discuss about the reasons that are bringing them to tackle this problem now. We look at the different alternatives that they have, and we guide them through this process.





Why you should trust us

We found that we could directly help some with our skills while others were better served by other solutions.

We thought we could use our experience to provide guidance for other companies dealing with legacy issues.



We are not always able to help all the companies that reach out to us because sometimes there are very good solutions that they should employ. They are not the solutions that we provide. So what we do is suggesting them to go for these other solutions that we do not provide, maybe to work with other service providers, with other vendors and so on.

And so over time, we developed some guidelines to give suggestion to these companies that reach out to us and help them pick a strategy that will work for them.





Legacy Modernization

Let's start your legacy modernization journey!



So, this is the end of our introduction. Now let's start with your legacy modernization journey.

