What is Legacy Code

Legacy Modernization: What is Legacy Code

How to deal with an old codebase

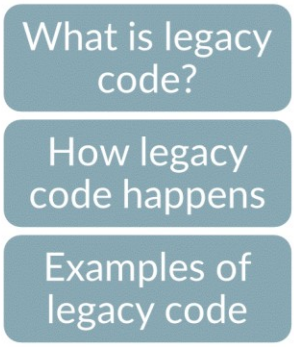
Let's see what is Legacy Code.

What is legacy code?

In this lesson we are going to understand what is legacy code and see some real-world examples.

In this lesson we are going to see what makes our code legacy, and we are also going to see some real world concrete examples. This is in general something that we will strive to do in the whole course, so not making things abstract or theoretical, but just share concrete examples so that we can ensure that we are on the same page. We are talking about the same concrete problems.

What is legacy code?



Contents of this lesson

This is the agenda that we are going to follow. So, recognize legacy code, understand when we have legacy code. We are going to see how it happens that we get legacy code, why our code becomes legacy, and then we are going to see some concrete examples.

So, what is legacy code? It's a little bit like a typewriter. My beloved Aunt Cicci used to write letters to her friends using a typewriter a few years ago, but then she got a computer and she started doing basically the same thing, writing emails instead of physical letters but using this new tool for communicating.

At the same time she could do much more because she could connect on Facebook, share photos and do all the kind of things that you can do on social networks. And you see with a typewriter you can only write letters, but with a computer you can do much more. And so, that is legacy code. Its code that limits your possibilities in the current world. It's a little bit trying to use a typewriter in 2023 and farther.

What is legacy code?

- What is legacy code?
- How legacy code happens
- Examples of legacy code

There are many different possible definitions of legacy code...

What is legacy code?

There are many possible definitions of legacy code depending on technical, business and practical aspects.

...depending on technical business and practical aspects.

What is legacy language?

- One way some code could be legacy code is that it is written in a legacy language
- We could say that a language is legacy if it is used primarily to maintain existing systems, rather than to create new ones

One way to define code as legacy or one reason to consider the code legacy is because it's written in a legacy language, in a language that is not typically used for writing modern application. We could also say that a language is legacy if it's used primarily to maintain existing systems and not to create new ones. So, if no one is considering using COBOL to create a new application, then we can say that COBOL is a legacy language, and therefore code base is written primarily in COBOL, are legacy code bases.

What is legacy code?

- A software can be legacy code, even when written in a non-legacy language
- It is more difficult to provide a simple definition of this case

Software can be legacy code also when it's written in a non legacy language. That case, things are slightly more complicated to define.

What is legacy code?

- It happens when a project is forced into maintenance mode because developing new features is too risky, costly or nobody knows how to do it
- You have code that does something valuable but the value is hard to use fully because of technological issues

We can consider the situation in which a project is forced into maintenance mode because developing new features is too risky, costly, or nobody is able to do that.

So, you have code that does something valuable, but value is hard to use fully because of technological issues with the form in which this valuable business logic is expressed.

What is legacy code?

- It runs on an old platform (OS, hardware, etc.)
- It uses an old paradigm (console, desktop, etc.)
- It is based on an unmaintained framework (e.g. ANTLR 2, etc.)
- ...

For example, maybe the code run on some old platform. It could be code written in, let's say Python, but it depends on some features of some old operative system or some hardware that are old and expensive and difficult to maintain. And so, it makes the code legacy. Maybe it's code that relies on some old paradigms. For example, UI paradigm like console or desktop when everybody's moving to mobile application and Web applications, or could be using some unmaintained framework like ANTLR2. So for example, even an application written in Java, which I will not say is a legacy language, could be legacy code because it's based on some old version of Struts, and no one is maintaining it anymore.

What is legacy code?

We have seen examples of:

- obsolete business practices, such as still using mainframes or not using cloud services, which lead to high initial and/or maintenance costs
- obsolete technical practices, such as using language or framework no more developed

Now, we've seen some examples of obsolete business practice, l using mainframes instead of using cloud services, and this could lead to high initial or maintenance cost, but we can also consider some obsolete technical practices like using languages or frameworks that are not maintained or are not evolving anymore.

It's not like you plan to get legacy code. This is something that happens to you. You keep walking by the same path that you always used to, and when you got started, it was the very best path possible. And at some point it was still something that was working really, really well. But nowadays, everybody else has found a better path than you, and so they can go to their destination much faster that you can. So, you didn't do any mistakes. It's just that somebody else found a better way to do things, and this is how you get legacy code.

How legacy code happens

What is legacy code?

How legacy code happens

Examples of legacy code

We have briefly seen why some code can be considered legacy. In the rest of the course, we will go more in details about the specific problems, but we've just seen in general what makes code legacy, and this is basically problems making code legacy, problems that prevent to use it properly. Now we are going to see why code becomes legacy, how we end up having a legacy code base.

How legacy code happens

> Legacy code is created by a mix of obsolete business and technical practices

Legacy code is created by a mix of obsolete business and technical practices.

How legacy code happens

> Legacy code was born when the rest of the world moved on to new designs and practices, while your company got stuck on the old ones and you were not able to keep up with the rest.

Now, legacy code becomes legacy, well, not at the very beginning of the project. It's very rare that a project is written from scratch using technologies that are past their prime and are obsolete already at the moment in which the code base is created.

Typically, the code become legacy, because while it was created using the best or at least reasonably modern technical solutions at the time the project was created, eventually the world move on, move to other projects, to other technologies. And so, we have some code that is stuck, is crystallized in all technologies, and we were not able to modernize it iteratively over time. And so, eventually our code ended up being in some technologies that is really not just mature but really outdated and should be dismissed as soon as possible.

How legacy code happens

This change happened gradually. New systems initially had some advantages but they lacked important features and were unproven.

Code doesn't become legacy from one day to the other. It's something that happens gradually. Maybe we write code in, let's say COBOL, and a new language come up, and it's slightly more modern than what we used to have, but we don't want to move. We want to keep our code base in, let's say COBOL. And then new technologies come up, and then more times pass, and eventually our code becomes really, really legacy.

At some point, initially, maybe when we initially brought the application, our code base was modern, then was mature, maybe slightly updated. Eventually, 30 years since we created the application, now it's legacy.

How legacy code happens

> Some new systems showed promise but failed.
>
> Some new systems instead improved and build upon their initial advantages. So they took more development effort, leaving what are now legacy systems at a disadvantage.

This is a natural process because when we write an application in a certain language, after we have created it and we evolve it, we start getting value out of it, new technologies come up, but these new technologies sometimes seems promising, and they disappear two years after.

In some other cases some of these new technologies that comes up become more stable, more mature, and eventually they become de facto standard. Everybody will use them if they had to create a new application in that particular moment. And they become more modern. They become a new point of reference. And in respect to these new technologies, the technologies in which we have implemented as our solution now looks old and updated because we are not as productive, because all developers move to these other technologies.

How legacy code happens

It is not obvious which languages or practices will become obsolete in the future. Something could work out for decades and then slowly become unused.

You could make reasonable choices at every step and still end up in a legacy situation.

Unfortunately, it's very hard to predict which technologies become outdated and when this will happen, and so you can only sometimes react after the fact.

How legacy code happens

- The status of legacy code is context-dependent, not absolute
- Some language may be legacy in the wider world, but still actively used in one industry
- Some languages might slowly become legacy language

Some other important points to consider is that the status of legacy code is context dependent, not absolute. It means that some languages could be considered legacy in general, but maybe they're still actively used in a certain industry. So, in that particular niche, maybe there's still a modern solution or an acceptable solution.

How legacy code happens

C++ is not currently legacy in embedded or system programming.

However, it is no more used in many other applications because of productivity and safety issues. Even the NSA suggests moving from C and C++:

https://media.defense.gov/2022/Nov/10/2003112742/-1/-1/0/CSI_SOFTWARE_MEMORY_SAFETY.PDF

Languages can become slowly legacy languages. For example, C++ is currently a reasonable solution in embedded or system programming. However, it's considered a legacy solution in most other fields. For example, even the NSA suggested moving away from C and C++.

It might be replaced by languages such as Rust in the future.

It is difficult to say now: the language has not evolved for years, but it has seen changes more recently.

In the future C and C++ could be replaced by languages like Rust. Right now at this moment Rust seems very promising, but it's very early to say if 10 years from now everybody will be abandoning C++ and moving to Rust.

It could be very well the case that five years from now, Rust will not have achieved this huge popularity. Maybe it will be in decline and be replaced by some other alternatives that we don't know yet.

So, this is also one reason why many organizations tend to be conservative. You don't want to jump on all the new technologies that come up because you can only verify after a certain time if there are really technologies that are here to stay for a reasonable amount of time or if they're just going to fade out after a brief moment of glory.

How legacy code happens

- There are many ways something can become legacy code
- Whether something becomes legacy can depend as much on the rest of the industry as on yourself

Now, there are many ways in which something can become legacy code, and this can depend on your decisions and on industry as a whole.

How legacy code happens

- Not all old code is legacy code
- For example, Windows codebase is very old. It had originally an old design. It is not legacy code, because Microsoft spent billions of dollars to update the code, kept up with new practices and even developed some of them

Not all old code is legacy code. For example, in the case of Windows, this Windows code base is very old. It had originally very old design, but it's not legacy code because Microsoft has spent billions investing on it and keep it up to date even if the original code base was written a long time ago, and they were able even to develop new practices to deal with this code base.

How legacy code happens

However, many Windows desktop apps became legacy code because developers could not keep up with the changes that Windows made to its own APIs.

However, desktop application that runs on Windows could instead be legacy code today because they were not able to keep up with the new APIs released by new version of Windows. So, maybe they could still be running on recent version of Windows but look outdated, not take benefit of the features or maybe relies on APIs that no modern developers know anymore because they have not been supported or widely used in the last 20 years.

How legacy code happens

> Unless you are Microsoft, legacy code can happen to you, because of a changing environment and decisions made by your ecosystem.

So, unless you are Microsoft where you are able to shape your own environment in a sense, often legacy code happen to you because of a changing environment and the decisions made by your ecosystems, like developers moving away, like framework being not support anymore, like operative systems evolving and changes their APIs. Sometimes it's the environment that dictates the conditions, and these changing conditions could result in your code base become legacy because the world around it change, and the code base was not able to keep up with the world that was changing.

How legacy code happens

A project might become legacy code because your company has changed.

For example, your organization moved to the cloud except for one custom system that you could not find a cloud equivalent for. That system is now legacy.

It *works*, but it does not *work for the new you*.

A project can also become legacy code because your company has changed. For example, if you have 10 different systems, and you move nine of them to the cloud, and now you need to have different processes for the single application that you didn't move to the cloud. Now that single application is becoming more costly to maintain and evolve for you because you move all your processes, all your skills to maintain applications based on the cloud. And so, your only application that is not running on the cloud because of that becomes more expensive and more difficult and cause extra complication to maintain because of your decisions.

So in general, maybe for another organization that instead decided to run everything on their own system, maybe if they had your very same application, that wouldn't be legacy for them because they will be structured in order to maintain that kind of application, or why when you move the decision to move to the cloud, and you reorient your resources to support that kind of application, only your application that did not move to the cloud becomes more expensive and more difficult for you, and so becomes legacy for you.

How legacy code happens

You had the best FORTRAN developers in the world, but they have now retired. So your code is becoming legacy because you cannot find any good FORTRAN developer.

Another very common reason because code becomes legacy is because the conditions around you change. For example, your code base is written in FORTRAN used to have the best and greatest FORTRAN developers but they retired. It's become more difficult to find good FORTRAN developers, and so this cause a difficulty to you. Your code base is more expensive to maintain than it should be because of the technologies used to define it. And so, these characteristics make your codebase legacy.

I think that CDs are good examples of the essence of a legacy problem. You may think that CDs were replaced by a better way to store music, similarly to what happened when cassettes or vinyl came by. However, this is not what made them obsolete. They were actually replaced by an entirely new way of listening to music, streaming. So now, the paradigm must change, while before, maybe instead of cassettes I could have CDs, and I could store them more easily. Now I have streaming that makes possible to have access to a quantity of music that you couldn't really imagine when you were just used to get CDs, right? You can get literally millions if not tens of millions of songs in your pocket or access to those songs.

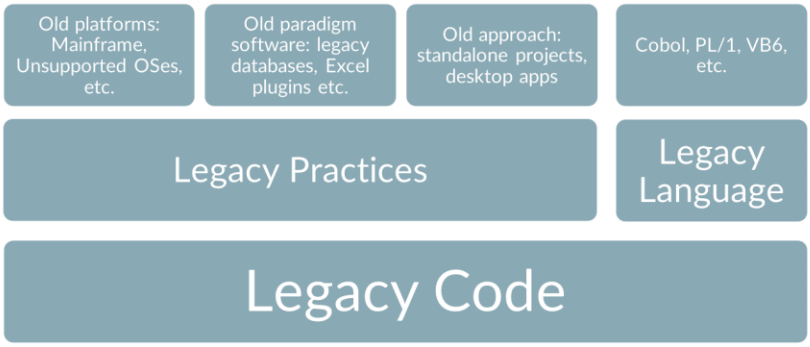
So, it's really a change in paradigm, and that is really the core of legacy code. It's not replaced by something a bit better, but some things that just work in a completely different way that enables you to do different things. And if you get stuck with legacy code, all these new possibilities are precluded to you.

What is legacy code?

What is legacy code?
How legacy code happens
Examples of legacy code

There are many examples of legacy application. Now we would like to share some examples that we have personally encountered as part of our work.

Examples of legacy code



There are code bases that are legacy because of legacy practices, like they're using some old platforms, like they're running on mainframes from older kind of machines that are now not as widespread as they used to be, or they're based on unsupported operative systems.

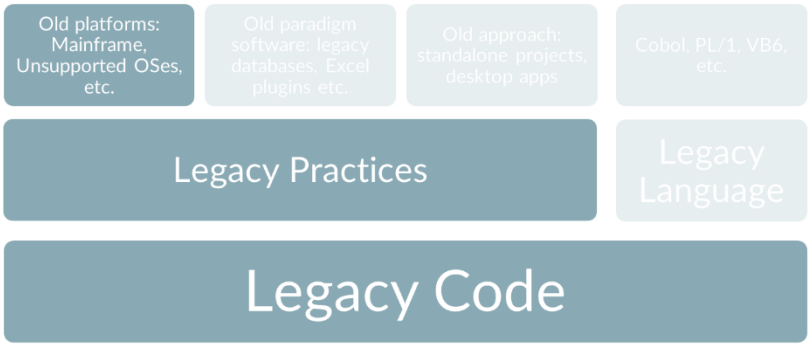
Or maybe we have an old paradigm, like maybe application are based on legacies' databases, maybe old relational databases, or maybe they're based on Excel plugins that are not compatible with a new version of Excel running in the browser. Or they use an old approach, like they are standalone project of desktop application, not integrated maybe in distributed Web applications, or they could be written in own languages like COBOL, PL/1, VB6, RPG, SAS, FORTRAN. There are so many.

Examples of legacy code

We are going to see examples for each group.

Now, we are going to see a few examples for each group.

Examples of legacy code



Let's start with old platforms.

Examples of legacy code

For example, mainframe code is largely legacy code, because most companies nowadays use new paradigms for technical and business reasons.

Applications running on mainframe are now considered, in most cases, legacy code because most companies nowadays are used to different systems either having application running on the cloud or using inexpensive machines like the one using Intel CPUs or, for example, Apple silicon.

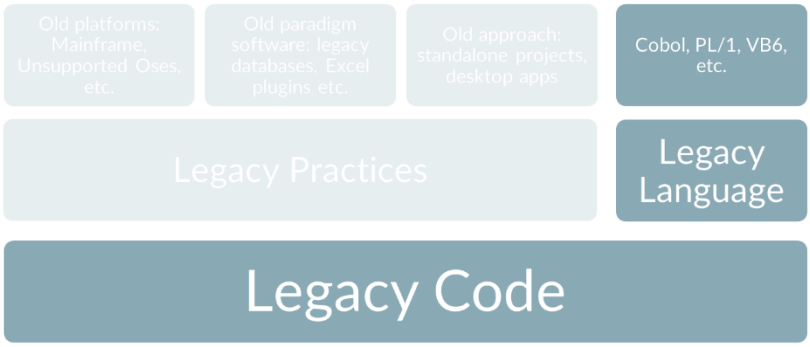
Examples of legacy code

- A financial company needed to move from mainframe computers to standard x86 servers
- This required a complex multi-year project. We helped them with one step: developing a transpiler from RPG to Java

For example, a financial company that we work with needed to move their code from running on mainframe to running on Intel-based servers, servers that they could provisions for a very low cost and were very easy to scale because buying an additional five of these machines would be very, very simple for them.

This case required a very significant migration effort because they needed to move their code base from RPG to Java. So, developing a transpiler that will translate their code from RPG to Java.

Examples of legacy code



Other example with Legacy Language.

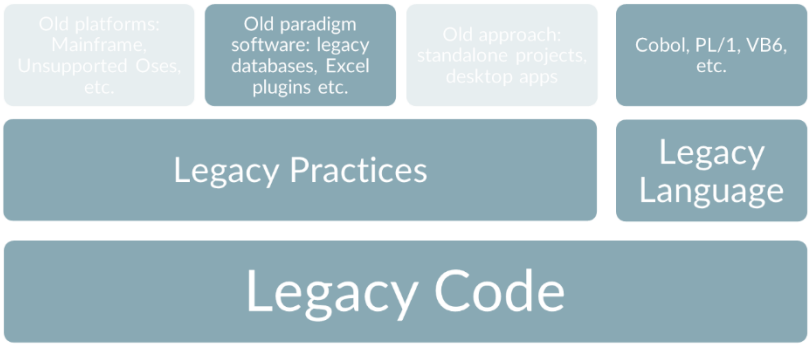
Examples of legacy code

We worked with a consulting company specializing in legacy modernization. They have their own platform to help the company moves from old systems.

We created for them a PL/1 to Java transpiler. They added services and support to the company.

We worked with a company that is offering specialized services and legacy modernization. They have experience working with a certain, specific kind of legacy modernization. And so, we worked with them developing a PL/1 to Java transpiler so that they could use it to provide services to their own clients.

Examples of legacy code



Some time ago, we worked with a company in the broader financial industry, company based in Asia.

Examples of legacy code

- Microsoft Excel is a staple of the financial industry. An advanced feature of the software is that it allows automation and programming using a simple language (VBA)
- Third-party companies rely on this familiarity to build plugins that use VBA to bring added value or even just their own platform that supports VBA

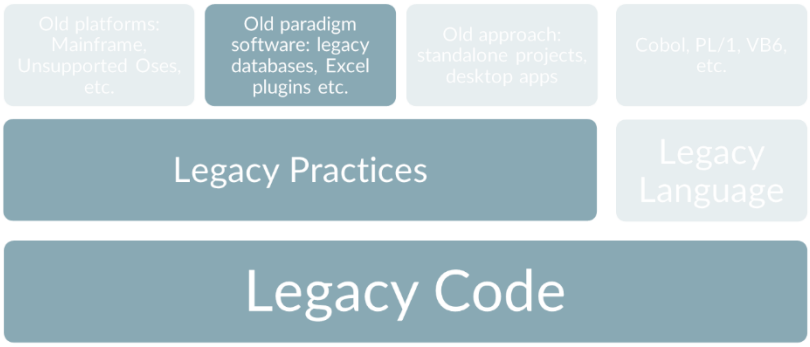
They developed a system based on their clients defining code inside VBA. They were defining VBA script inside Excel, and they had a system that extracted this code processes, process it, and transform calculation. However, this system was not working that well as their system scale because this was not fast enough. And so, the fact that it was relying on VBA and the execution speed of VBA was really preventing them to move forward and was making the system unmaintainable for them.

Examples of legacy code

- We helped many companies deal with these issues
- For one, some years ago, we developed a transpiler from VBA to C++

So, in that case we developed a transpiler that was taking this VBA code, transpile to C++ and execute it much faster so that they were actually be able to work with very large system processing millions of records.

Examples of legacy code



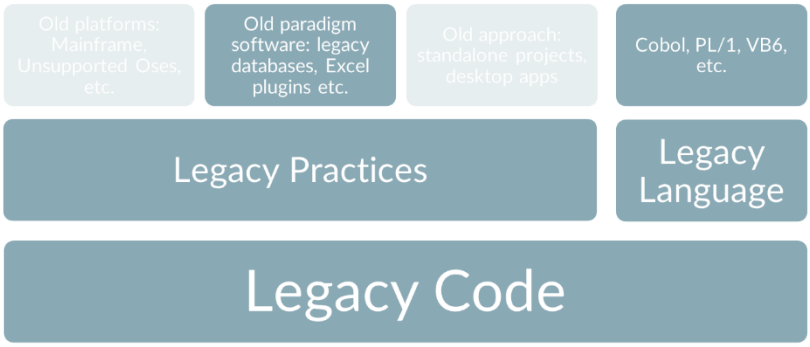
Another example of legacy application that we met is related to parsers.

Examples of legacy code

Another example of migration is to migrate their existing parser from ANTLR v2/v3 to v4.

Over the years we have met clients that had parser written based on ANTLR v2 and ANTLR v3, and we helped them migrating with ANTLR v4 because ANTLR v2 and v3 were not maintained anymore, was difficult to get fixes for those systems and was difficult to find developers that knew ANTLR, these old versions of ANTLR. So, in that case we helped them migrate to a new framework.

Examples of legacy code



Another example of legacy code is often related to databases...

Examples of legacy code

- SQL is also a standard programming language used in many software
- However, many software implements it with some differences. For instance, Azure SQL is different from Snowflake SQL or Oracle SQL

...and the SQL code that is being written to run on specific databases, like for example, Oracle or Teradata.

And many companies nowadays want to move to other database like Azure or Snowflake and so on. And so, they need to translate their code from a certain flavor of SQL to another one so that they can move to these other databases.

Examples of legacy code

SQL is also used in many ways. Some use it just for standard data manipulation. Others take advantage of procedural functionality to create full programs.

So a certain SQL dialect can be a legacy language or part of a legacy application in some cases. This is even though it is still widely used.

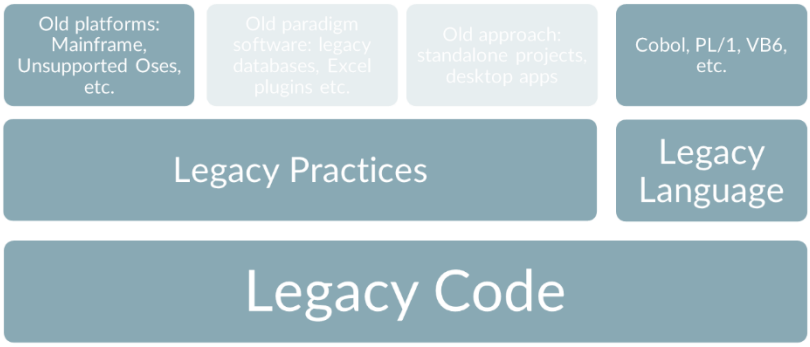
And so, we have helped them translate this SQL code.

Examples of legacy code

- Several companies have the issue of migrating from procedural PL/SQL to Java, Teradata SQL to Snowflake
- Many companies need help to migrate from an on-premises database to a cloud one

For example, in some cases they move from one SQL dialect to another SQL dialect because they want to move from Teradata SQL to Snowflake. In other cases, some companies wanted to move away from dialects of SQL that were containing also procedural code, for example PL/SQL, to proper general purpose languages like Java.

Examples of legacy code



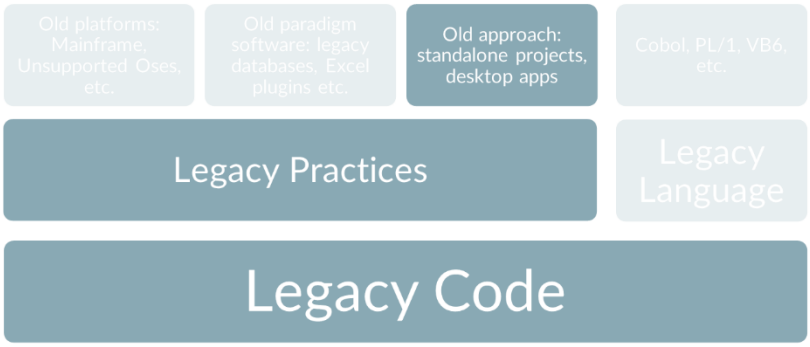
Or we have worked also with companies that wanted to move away from SAS to a combination of Python and SQL.

Examples of legacy code

> It is not just SQL for data applications. There are companies migrating SAS to a combination of Python and SQL to handle both data analysis and storage.

In that case, so they could use the strong points of SQL for processing data and of Python to perform more advanced data analysis.

Examples of legacy code



Now we can see an example that is a bit more advanced and more rare.

Examples of legacy code

- A company develops software for managing infrastructure (i.e., climatization systems for buildings, industrial technology, etc.). Over the years they developed a variety of systems to serve different clients' needs or pieces
- They needed to support a new technology, they developed a new software

This is an example of a client of ours that is a large company, a large conglomerate. They're writing software to managing infrastructure, like climatization system for building, industrial technologies and so on. And they developed over the years a series of different systems as they work on different projects, work with different clients and so on. When they needed to support a new piece of technologies, they just add the technology piece to their system.

Examples of legacy code

- The cost of maintaining old software kept going up and even their own teams had difficulty understanding their codebase
- They decided to change their overall approach. They put their domain knowledge in a DSL that could power all their software for decades to come

And eventually the system was supporting so many different technologies, was very difficult to maintain.

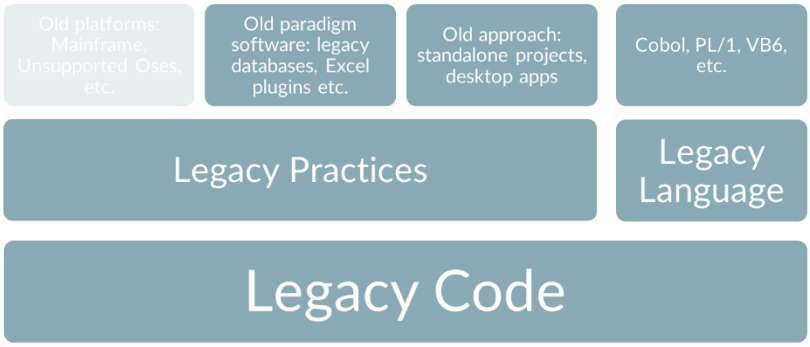
So in their case, we develop a domain specific language. So, language that they could use to express everything they needed to and simplify the adding support for a new technology.

Examples of legacy code

- They expressed their business advantage, their expertise, in code written in DSL
- They re-build their client apps on top of this DSL
- If you are not sure what is a DSL, do not worry, we are going to talk about it in a following lesson

So, we then help them migrate the existing code to this DSL.

Examples of legacy code



We have seen that software does not rot, per se...

Examples of legacy code

> Software does not rot. However, changes in business practices and environment can create legacy code.
>
> There are changes in vendors, changes in available people, alternative solutions that perform better, and changes in the way we work.

...but changes in business practices, and the environment can make your code legacy, because now there are better alternatives or because the conditions made more difficult to maintain the systems. So, changes in services provided by vendors, changes in skill and competencies that are available, or just the fact that there are better alternatives can make your code legacy.

Examples of legacy code

For example, the migration from desktop to web or mobile apps. In some sectors, desktop apps are becoming like mainframe code.

We have seen many examples. For example, the migration from desktop to Web or mobile applications all of a sudden made desktop applications looks very outdated, even if nothing changed about desktop application per se.

Examples of legacy code

We worked with a company with apps in Visual Basic.

When Microsoft stopped developing Visual Basic they could not improve their apps. But it was still not convenient to recreate the same app in another language, so they got stuck with legacy code.

Or imagine again a company that has a series of application written in Visual Basic. When Microsoft stopped developing Visual Basic, they could not improve their apps, so they needed more features, like for example, the support for Unicode. They got stuck.

And given there is a huge barrier in moving to another language because the typical approach is taking your application and then rewrite completely in another language. Because of this huge barrier, they keep remaining stuck in code that is written in Visual Basic that becomes more and more outdated as the time pass.

Examples of legacy code

They decided to move from a desktop app in Visual Basic to a web one in JavaScript.

One such client, we have them migrating from Visual Basic to JavaScript so that they could move, transform the desktop application in Web applications. And we did that through defining a transpiler.

Summary

In this lesson we have seen that legacy code is not just *old code*.

It is old code that still matters and impacts your business.

In this lesson we've seen, we've seen that legacy code is not just old code. It is code that is difficult to maintain, costly to maintain, that is worse than the alternatives because of changes in the environment, typically. But this code, it's important to consider that it's still valuable. There are problems in the form in which this code is expressed, but the logic that is expressed, the function that is performing are still very valuable because if this is not the case, you can simply throw away that code.