How the ESM was built on coding language

By Rogelio Rodriguez, Kalin Anev Janse

26 March 2021

In a decade, the ESM and predecessor EFSF have helped euro area countries overcome crisis, mobilised nearly €300 billion in loans and invested €80 billion in paid-in capital. Without coding, we never could have done it. Established in 2012, the ESM is a new lender, free of legacy technologies that burden many institutions. Instead, we have taken advantage of the latest 21st century technologies, from cloud computing to micro-services. As our financial operations have grown increasingly complex, we have embraced these tools even more. Our experiences have taught us a great deal about the importance of coding.

We have created a single cloud-based banking ecosystem that enables end-to-end workflow for our trades and position keeping, linking eight critical ESM functions covering the whole trade lifecycle. The system optimises time, eliminates paperwork, and reduces operational risk regardless of differing roles and complexities of the tasks.

We built our own data hub in-house using a micro-services architecture, a technology pioneered by Netflix, to integrate our workflow. This allows fast incremental developments, enabling all our data to be accessed in a secure, auditable, user-friendly way, via any device. Thanks to this platform, we make our financial assistance programmes transparent and accessible through our website. This same technology is used internally to, among other things, obtain and process business data, generating dashboards with coded calculations, which reduce the need to exchange manually-generated excel sheets via email.

What is coding?

It's the translation of financial or economic concepts into computer language – code - so that computers can perform complex calculations and visualisations. If we implement a new product, like the ESM Programme Database, it requires

thousands of lines of unique and proprietary code. They are carefully written by our technology team.

Six takeaways from our experience:

Coding is translating. To bridge the gap between business and technology we need to be proficient in both domains. Understanding the user's needs and translating that into a solution in machine language requires explicitly specifying every aspect. To enhance the conversation you bring technology to the front office: We co-locate developers, capital markets staff, and agile project managers in a team. This increases speed to reach a common understanding and rapid product delivery.

Coding is invisible, complex, and critical. Google has two billion lines of code. It would take 63 years to read it, at one line per second. The ESM is only one million lines of code, but it would still take 11 days to read it all. One of our Asset and Liability Management reports produces an output of 14 numbers. These 14 numbers are critical to calculating our cost of funding, used to produce invoices for the ESM's loans. The code to produce only these 14 numbers is 700 lines, a clear indication of how complex coding can be.

In another example, one Asset and Liability Management tool developed within the ESM data hub has around 150,000 lines of code. To put this into perspective, this is three times the length of novel *The Lord of the Rings*. This is manually coded. It is of the utmost importance to get it right.

Coding takes time. Political decisions take time. In order to reach agreement on the ESM's backstop to the Single Resolution Fund it took a few years. Other decisions are faster, like the introduction of the ESM's Pandemic Crisis Support, which took just eight weeks to agree. These decisions need to be coded and the financial operations interconnected. Getting the basics of the Pandemic Crisis Support ready took eight weeks, but finalising a robust solution to cover all the possible variations took five months. It takes time, including many weekends, to deliver for teams ranging from IT to middle and back office – often the hidden heroes who deliver on political decisions.

Code quality is difficult. There are many solutions to the same problem. In the early days of the ESM data hub part of the team was adept at using Google's Angular framework and another part favoured other methods. We later agreed on a single approach to use the Angular framework, but then needed to rebuild the incompatible parts. We had to explain to users that it would take time and resources to rebuild part of the codebase to produce the exact same output in a different way. Thanks to this increase in code quality and cohesion, both maintenance and evolution of the tool is easier today.

Need to prototype fast. Due to negative interest rates, we had to expand our investment classes and implement new tools in core banking systems. The process takes time and requires prototyping before implementation, so to enable business as soon as possible we built a workaround solution. The daily operation of this workaround solution was heavy, so we worked quickly to replace it with the right solution. The need to prototype and to support daily operations needs to be balanced as resources are always scarce.

Complexity is usually underestimated. We are all familiar with technology at some level; computers are ubiquitous in our lives today. Due to this, technology's complexity is usually underestimated and expectations are high. Over time, we are educating our management on the complexity of technology. This mutual understanding marries strategic delivery and technological robustness.