

Accelerate the modernization of your core applications: a strategic approach

whitepaper

MASTER your complexity | the best software for unique operations

USOFT


YTI
ONE
YANG MING
K LINE
Hapag-Lloyd
CMACGM
OOCL
MOL
U A S C
TTX

Contents

Introduction 4

1 The clear case for modernization 6

2 Revealing the true legacy of legacy systems 8

3 Is low-code the answer to ever-growing complexity? 12

4 A proven, strategic approach to modernization 14

5 Defining the best-fit low-code technology for your modernization 24

6 The importance of business rules in modernization and other software development 26

7 Conclusion 30



Introduction

The need to modernize is urgent – yet there are considerable headwinds that make this essential task hard for organizations to realize. Numerous hurdles stand in the way; not least the very legacy systems that need to be replaced or modernized.

It isn't always easy to see a clear way forward either. With an array of technologies available, it's easy for organizations to find themselves in a state of 'decision paralysis' - unable to see which technology is best for their unique situation.

With a clear vision of what's needed and how to get there, your modernization projects don't have to wait any longer.

In this white paper, we will explore the many opportunities and challenges of modernizing mission-critical software, and plot a course that will help you define how to realize your objectives.



1

The clear case for modernization

One of the biggest reasons for modernizing your core software is the growing volume of new requirements. These new requirements come from customers, suppliers, or employees. All demand new capabilities, functionalities, and cross-process and cross-organizational support (among other things), in addition to any new legal requirements that emerge from time to time.

The result: a market in which innovation must accelerate just to keep pace with current demands.
Organizations have clear reasons to evolve their digital infrastructure, but it isn't always clear how they can best achieve it. Additionally, a fear of touching those mission-critical systems has led to organizations carrying the weight of legacy for far longer than they should. Furthermore, the longer this continues, the harder it is to replace them with modern solutions.



"We see a huge development in terms of increasing customer demand, for example, changing supply chains, shortening lead times, a very intense focus on error reductions and quality and compliance. And organizations need to make sure their systems can keep up with this speed of change."

Hans Canisius, CEO at USoft



2

Revealing the true legacy of legacy systems

Legacy systems frequently occupy a central role in an organization, but can become a constraint on growth and development when they cannot adapt to current situations, connect with new technologies, or include better techniques. ERPs and other proprietary legacy systems may be highly specialized in the tasks they perform, but lack the needed capability and flexibility to adapt, expand, and scale.

Even today, a great deal of core and mission-critical software is written in legacy code that's reliable, but lacks transparency, adaptability or forward-compatibility. For example, many organizations have critical applications based on Report Program Generator (RPG), the software on the IBM AS/400 and iSeries systems. Another well-known legacy programming language is COBOL. These applications may still prove their worth today, but they can hardly meet today's requirements.
These systems have been trusted, often for decades, to coordinate the most crucial tasks and leverage the business' unique methods and knowledge to give a competitive advantage.

However, as these systems have aged, they drain increasing amounts of resources and become inherently rigid. It becomes more and more apparent that the value they provide is outbalanced by the mounting costs and the lack of flexibility they offer.
With a clear need to modernize right now, it should be easy to get everyone behind this vital project. Yet, there are many reasons that hold back digital transformation and modernization efforts – even when everyone is on board. Many of these come from the legacy systems themselves, so it's important to understand how your 'trusted core software' is harming your business.

The maintenance burden of legacy systems

From our own research of 500 business managers across many industries, we see that business teams are constantly looking for new ways to accelerate innovation and development timelines, but core systems themselves are a major speedbump.

It's common for IT departments to spend 85% of their time and budget just on maintaining their core IT landscape - essentially just 'keeping the lights on'. With a constant backlog, developers are hamstrung and cannot take on new projects, especially daunting ones like modernization.

How legacy systems affect response times

When employees need to use workarounds and manual methods to enter data (or perform other tasks), they're being denied access to the modern tooling your competition is already using.
By contrast, new market entrants will have an immediate advantage, as they can immediately adopt the most modern capabilities without having to pander to existing legacy systems and the antiquated business architecture that has grown around them.
Imagine: a potential client contacts you for a quote. Naturally they're also checking out the competition - so what if they can provide their quote 2 days before you?
In this scenario, there's a good chance you'll lose business without being able to compete at the standard expected. This is just a simple illustration of the hidden ways legacy systems prevent your business from being able to react at the speed demanded by the market, and this reduces revenue.

The effect of legacy systems on innovation

With excessive resources being consumed just to keep aging systems on life-support, your IT teams are unable to focus on new innovations. As the profitability of the entire business is also affected by the inefficiency of outdated software, there's less budget available each year for new projects, research and development, or marketing and sales events. This has a very significant knock-on effect on your ability to innovate for the future.

Additionally, the lack of flexibility of older systems makes it hard to adopt new technologies, because they cannot properly integrate in a holistic way.

The result is an organization that's chained to the past, unable to move on without breaking its bonds.




msc
tex
msc
msc
msc



3

Is low-code the answer to ever-growing complexity?

Growing data complexity and operational complexity have created huge volumes of information and incomprehensibly complex systems. As demands have grown, businesses have created workarounds that allow them to keep using their dependable legacy systems. These often rely on manual methods for inputting and transferring data across different systems, increasing the risk of incorrect data or other issues.

To overcome these challenges, organizations are actively seeking ways to replace their legacy systems with modern and future-proof alternatives – fast.
One of the most intriguing possibilities today is the use of a low-code development platform to redevelop core systems. Many businesses have taken this approach to develop customized tooling and mission-critical core systems; some 500 million applications will be developed in 2023 using low-code platforms of various types.

However, it's important to recognize that new technology will not fix the problems of ailing legacy systems on its own. No technology is a silver bullet, and low-code, although very powerful, isn't the only option on the table. The opportunities and potential value of modernization are very real, but the path you take must be carefully planned to ensure your new software meets the objectives of your growing business. For this, it's important to understand what the opportunities are and how to reach them.



4

A proven, strategic approach to modernization

To get the full benefits of modernization, you need to employ a strategy that aligns with your business objectives. For this reason, a modernization strategist is often brought into the process to shape the direction of software development. PIT digital transformations is one such partner. They have experience in using a variety of technologies to realize the best results for each distinct business case.

We recently collaborated with PIT digital transformations for a webinar, in which we outlined the following proven approach. It outlines the 3 opportunities of modernization, the 3 pillars that support these opportunities, and a step-by-step method for defining the project scope.

THE 3 OPPORTUNITIES OF MODERNIZATION

Many possibilities can be realized with a well-executed modernization strategy, but there are 3 vital opportunities that are critical to your future success. Capitalizing on these opportunities is supported by the 3 pillars of modernization and the step-by-step approach (below).

3 opportunities modernization can offer

1. Clarity about desired functionality: using feedback from users of current systems, you can gain knowledge about how your current systems fall short and what's needed to make them complete. This is often hidden by departmental workarounds and other solutions that cost time and increase risk.

2. Understanding the current digital environment: businesses gain a better overview of how things work, modeling the business in terms of information flow: what data is needed by which systems, what are the interfaces, and what data people need to work effectively. You also gain a vision over where users need to manually enter data into systems, or manually transfer across 'disconnected' systems. This overview empowers you to make faster, more error-free systems.

3. Redefining how your software supports the business: legacy systems were created in a different age, when interactions with software were different and core systems were expected to do less. By modernizing your systems, you have a new opportunity to support your business better with the best technologies available today, and create adaptable systems that can meet the needs of the future too.

THE 3 PILLARS OF EFFECTIVE MODERNIZATION

The 3 opportunities above are mirrored by the 3 pillars approach to digital transformation and modernization, which outline the core elements of an effective modernization strategy. These are necessary for achieving digital success and are both simple and important. To modernize core applications effectively you have to:

1. **Set a clear course**: the course defines what you want to achieve. What is it that the system should do for your organization - now and in the future? By taking this question very seriously, you prevent yourself from just rebuilding the current system with new technology. It forces you to stand still for a moment and get back to the real purpose of digitalization: making organizations better, faster and less complex. What should the modernized system be like to really contribute to the purpose of digitalization of your organization?

2. **Make a practical plan**: once you have defined a clear course you need a practical plan to realize it. Most likely the desired modernized system will interact differently with the digital landscape than the current system does. It will also support the business processes in another (improved) way than the current system does. And it could even interact with your supply chain partners and your customers differently. To conclude, modernization is much more than a technology project. To make it succeed you need to plan carefully, including people, processes, and technology.

3. **Take targeted action**: the plan supports you in taking targeted actions; focused actions that clearly contribute to making your organization better, faster and less complex. We have to be realistic: managing projects - especially those in the center of technology - people, and processes can be a bumpy ride. No matter how well-planned and executed. Without carving it into stone, a good plan can always guide you in these rough times. It supports decision making and focussing on the original goals to be achieved.



When combined, these 3 essential ingredients support the execution of an effective digitalization strategy. Without them, you create an environment that's the ideal breeding ground for problems, conflicts, and hindrances to progress.

Without redefining how your software supports the evolving business, organizations can easily spend considerable resources just to force new technology to mimic the outdated patterns of your current solution. Or you end up with a modernized solution that lacks essential capabilities because you didn't have a clear overview of what functionality was needed from the start.

The frustrations of yesterday can easily be reborn into a newer version, and this opens the door to shadow IT and a never-ending process that consumes time and money. To prevent you from these pitfalls, it's essential to define your modernization project very well.

Step-by-step definition of your modernization project

To capitalize on the opportunities modernization offers, your project should follow a structure that clearly defines what is needed and how to achieve it. This structure is supported by designed workshops and supportive templates and summarizes everything needed to implement the 3 pillars. By achieving clarity about what is needed, drawing up a concrete plan and taking targeted actions to achieve this.

The structure consists of 4 main steps, as follows:

Step 1: Understand the current situation



To really modernize the current legacy system, you have to start with creating a holistic understanding of the current situation. As explained above, the current system is generally designed years ago. Since then, many changes took place, resulting in a change of the required digital support. It wasn't possible to adapt the system in a way that all new requirements could be met, resulting in workarounds, decentral point solutions, informal working processes and all kinds of informal 'data stores' in Excel, e-mailboxes, and paperwork.

For sustainable and effective modernization, you have to dive into the current way of working. You have to create a complete view on the current information flow through the organization:

- Inbound information flows (e.g. new applications for an insurance policy);
- Who supplies information (e.g. new clients);
- How is that information processed (e.g. the new applications are being assessed);
- What (new) information is created (e.g. acceptance/rejection/request for additional information);
- Where is the created information stored (e.g. in the legacy system/in an Excel file);
- Who is the consumer/user of the created information (e.g. another department/the applicant);
- How do the consumers/users receive the information (e.g. via phone call/email/the legacy system).

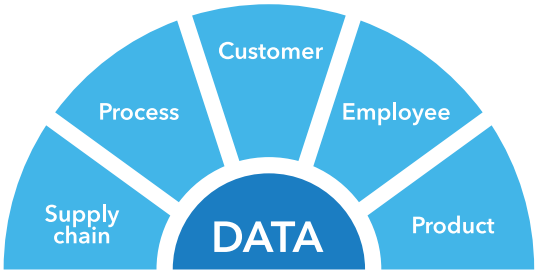
Step 2: Determine the course



Once you have a good understanding of the current situation, you need to answer three questions to create a sustainable and better alternative.

1. What needs to be strengthened or improved

To discover what needs to be strengthened you should concentrate on the core competences of your organization. To strengthen these core competencies means making them part of your organizational competencies and thus independent from the personal competencies of your employees.



To investigate what needs to be improved, you need to identify the inhibitors in your business process. Which elements cost a lot of money, time or effort? Which elements cause errors, irritation or are simply stupid to do?

The figure above illustrates the organizational perspectives we use to investigate the inhibitors in the business process:

- Supply chain processes
- Internal working processes
- Interaction with the customers
- Blockers/hurdles for employees
- How you deliver your products or services or how users experience it

2. What needs to be anticipated

To create a sustainable alternative, you must take into account the developments you foresee in your industry, with your customers, in the economy and in the applicable legislation. What developments do you see and what are the consequences for your business process?

Moreover, you have to take digital trends into account in the same way. Which digital trends are relevant to your organization or sector and what influence could they have on your business model and process.

3. What course should we follow

To determine the digital course or direction for your organization, and thus for the modernization of your mission critical core legacy system, you now need to combine all your findings of the previous steps.

So, you need to determine what your organization has to be capable of, what it needs, and what it has to do, to strengthen what needs to be strengthened, to improve what needs to be improved and to anticipate what needs to be anticipated.
A constructive and practical way to describe the course is to define organizational challenges that combine and summarize all the findings from the previous steps. We usually end up with 6 to 10 challenges per organization.

Step 3: Make a practical plan



Now you understand the current situation (step 1) and what course to follow (step 2) it's time to create a plan, describing what actions to take and in which order. It takes 3 steps to create the plan:

1. Design the target situation

In step 1 we created a picture of the current situation. In this sub-step we create a picture of the desired situation. What does the desired way of working look like and what does this imply for the support that is delivered by the digital landscape.
You need a free mind at this sub-step; you need to ignore the current digital environment. Just focus on the purpose of all current processes - what is the purpose of all activities executed by the current organization. The target situation should support this purpose, in a modern, efficient and effective way.

2. Determine the transition

Now that you have images of both the current and the intended situation, you can analyze the gap and the transition needed to bridge the gap. As mentioned earlier, focus on people, processes, and technology.

3. Define the approach

Finally, the transition must be divided into logical phasing, steps, and priorities - to make it realistic and manageable. Again, this must be from the perspective of people, processes, and technology. Finally, add budgets for time and money, to complete the plan.



Step 4: Take targeted action



Finally, it's time for action - targeted action! And that's possible, because you've invested in creating a plan, so you know exactly what actions need to be taken.

Each transition is unique but in general the first actions to be taken are the selection of products and suppliers, we will dive into this in the next section.

> **"Users already have experience with the system to be modernized: they know what works well, what functionality is lacking, the current workarounds, mostly supported by shadow IT. They are the most valuable resource for requirements on the modernized system."**

Ruurd Portman,
CEO, PIT digital transformations

Besides from the actions, in our experience, you need to keep some key principles in mind when managing modernization projects effectively:

1. **Focus** - create a plan and work with the plan. It will guide you when the weather and sea get rough.
2. **Agile** - you created a plan; don't stick to it in any case, change it when you need to.
3. **Managed** - work in phases to reduce the chance of a project getting out of hand.
4. **Overview** – always keep an eye on the bigger picture; it helps you to zoom out and put unexpected setbacks into perspective during the project.
5. **Speed** - create and maintain momentum; delivering (small) results fast will create a positive vibe and momentum for further improvements. People generally are happy to contribute to success.
6. **Ownership** - for effective improvement, you have to own the change. All improvements demand effort and energy. And effort and energy demand ownership.

WHICH PATH TO TAKE? 3 CLEAR OPTIONS FOR MODERNIZING SOFTWARE

There are 3 different routes a business can take to modernize its software, and each of these has distinct strengths and weaknesses.

1. Off-the-shelf solutions

You often get a proven product and best-of-breed tooling from these commercial solutions (standard software products). These solutions can usually be configured and/or customized (to some extent) to better match your business needs. However, adaptability is limited and customizations take time to build too. These solutions do not always offer the best value if you just don't need a lot of the functionality they provide. Or when you require a lot of customization.

Advantages: realization speed, reliability, and stability.

Potential pitfall: the 'slippery slope' of customization countering all the benefits of COTS.

2. No-code and/or low-code development

In the case of no-code, solutions can be designed by business process managers, process designers, and the business analyst. No-code generally offers less sophisticated capabilities and is good for relatively simple use cases. Low-code, on the other hand, can be used for more complex applications and certain platforms even for totally rebuilding mission-critical software. As a result, low-code may require a greater need for technical expertise , but allows you to cover a much wider range of use cases.

Both no-code and low-code have the benefits of standard building blocks, easy customization, and the ability to custom-code specific processes. Be aware however, there's a lot of variation among platforms, and each has its own 'sweet spot' for use cases. You need to select the right tool for the job.

For example, no-code platforms are relatively easy, but can show their limitations in terms of what you can do, whereas certain low-code platforms, like USoft, provide a greater level of freedom comparable to high-code development.

Advantages: realization speed, flexibility, ease of maintenance, and cost.

Potential pitfall: don't mistake one platform for the other, they are not as interchangeable as often thought. Selecting the right platform is key.

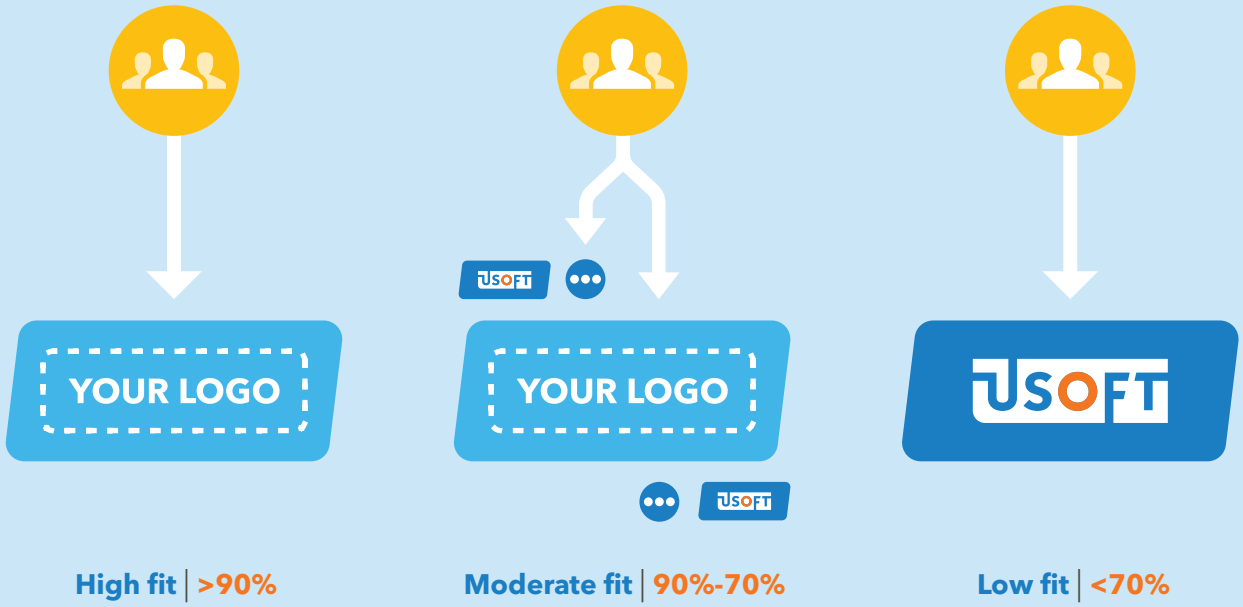
3. High-code development

Manual coding takes a lot more time and resources, but isn't limited by the available components as developers create what they want. The risk of this route may be higher, in part because it takes longer to deliver solutions, you have to deal with resource limitations and the gap between business and IT teams is much larger. In addition, the extended development time means the 'goal posts' are more subject to change as new business priorities are identified (which also increases the time to value).
There's also a real danger that you spend a lot of resources custom-coding a lot of functionality already included in off-the-shelf solutions or available as standard components in low code. Reinventing the wheel has a very low ROI.

Advantages: precision and flexibility of functionality. Build exactly what you need.

Potential pitfall: the argument of vendor lock-in. Yes, with high-code you don't have a vendor lock-in. But with expert resources becoming increasingly rare, don't neglect to account for the risk of resource lock-in.

To decide which route is best for your situation, you should carefully consider the context and, for example, the nature of the processes involved. As support guidelines, you could for example evaluate your need based on the uniqueness of the process or processes involved, and their respective competitive power.



When it concerns a standard process, like finance or HR, considering the off-the-shelf solutions already available on the market is a very sensible route. In these standard cases, commercial solutions will often fulfill many of the organization's key requirements, needing only minor customization. But in other cases, as the uniqueness and competitive impact of your process increases, this gap will become significantly bigger and require much more customization to align with your needs.

You should appraise commercial off-the-shelf (COTS) software solutions based on how well they match your business requirements and evaluate this match in relation to the uniqueness and/or competitive impact of your business processes. Based on this evaluation, you progress towards selecting the best-fit technology.

You can gauge the suitability of a solution based on how well it covers your requirements. Here is an approximate guideline:

- **>90% fit with business requirements**: there's no compelling business case for a customized solution and COTS is essentially a good choice.
- **70 to 90% fit**: selecting an off-the-shelf solution is a viable option, complemented with customizations built with low-code technology to better fit your business needs. This ensures that you maintain the benefits of COTS by not adding customizations directly in your COTS, which would result in rigidity and substantial cost increase.
- **<70% fit**: the use of a low-code platform to build your own solution is a viable option, so it perfectly fits your business.

USING A LOW-CODE PLATFORM TO LEVERAGE YOUR UNIQUE PROCESSES AS A COMPETITIVE ADVANTAGE

By selecting a low-code platform that meets your business' critical requirements for functionality and customization, you can create the ideal digital toolkit for leveraging your 'special sauce'.

But it's important to focus your attention on where you can have the greatest impact, identifying the value of your unique processes and knowledge. This ensures your resources are well-spent.
It's useful to think of technology as a pyramid, with broad-based off-the-shelf solutions at the bottom of this, and your most customized solutions at the top.
The 'bottom-layer' solutions cover most of the general tasks that are common to all organizations, or organizations within your sector. An example of this is software that is used in departments like HR or finance, as this must meet the same basic requirements in any organization, but confer little competitive advantage.

As you move up the pyramid, you see increasing specificity with how your organization (and individual departments within it) uses software. These layers require more specialized tooling that helps specialists perform important tasks.

And, at the top of the pyramid, you have software that is incredibly specific and potentially unique to your organization. The very top layer represents where your own custom coding can have the greatest potential impact, by offering unique services or processes your competition cannot. This way, you can use your unique processes as a competitive advantage, with your own custom-built technology supporting them.

With the right low-code platform, you can realize:

- **Decreased time to market for new solutions** - low-code enables the rapid assembly of highly capable business-critical software, massively shortening the time to market compared to high-coding.
- **Streamlined business user collaboration** - minimum viable products (MVPs) and proof of concepts (POCs) can be made very quickly, and adjusted easily, especially when business rules are used as the basis.
- **Cost reduction of software development and maintenance** - reduced time spent on development results in a sharp cost reduction. This is enhanced by the relative ease of updating or adding functionality.
- **Improved productivity resulting in more resources being available for innovation** - with a reduced time-to-market, your software can start delivering value sooner. By using an efficient low-code platform, the development process can be more productive - allowing your IT resources to be used to support future innovation (instead of maintaining legacy systems or high-coding custom software).

INCREMENTAL PROGRESS: ONE STEP AT A TIME

No business wants to risk tinkering with their mission-critical software, so the often proposed 'big bang' approach isn't very appealing. The risk of failure is higher, and the impact of things going wrong and their ripple effects are so much greater.

Instead, we recommend an incremental approach. With the USoft approach to modernization, your legacy application is meticulously dissected and analyzed. Each function is modernized, one step at a time. This approach inherently lowers the risk of your modernization process and provides much more control over the entire process and each iterative step.
The platform uses specialized technology that semi-automatically unravels legacy code, and extracts the rules and other knowledge it contains. In the case of COBOL, for example, you can regain between 90% and 95% of the knowledge locked within this legacy code base, including:

- Functional blocks
- Business rules
- Data model
- What code is effective
- What code is ineffective
- The application flow
- The UI flow

Using this automated analysis of legacy code, you can recapture all of this business logic and convert it into natural language and diagrams, in the USoft platform. This ensures that there is a clear overview of what's already there, before deciding (together with the business) which enhancements are required. This repository for regained business knowledge and new insights and business needs, subsequently provides the basis for realizing the new solution or functional area, before moving to the next part.

"You can custom code everything from scratch or you can use a technology like low-code that bridges the chasm between commercial off-the-shelf and high-code, and speeds up the development process and ensures that what you are creating is specifically tailored for your specific needs. Today and towards future changes and innovations."

Hans Canisius, CEO at USoft



5

Defining the best-fit low-code technology for your modernization

As we've already discussed, you can't just throw technology at a problem and expect it to solve it for you. There are multiple low-code and no-code platforms on the market, and each of these have unique strengths.

The best fit will depend a lot on your organization and the use cases you need to tackle. For example: some platforms will be ideal for simple to moderately complex front-end applications, while others are designed for 'citizen developers' who want to create their own tooling and workarounds. However, for a business-critical application your platform must be capable of much more.

The USoft platform, for example, has been designed with a specific focus on complex core systems that revolve around using high volumes of data, with extensive business rules, processes, and transactions. Of course, you can still build simple applications with USoft, but it is designed to do much more. This kind of platform is what organizations require for a successful modernization of complex and core systems.

Standardization vs. customization

As we've seen, it makes sense to focus on customization for processes that are not covered by commercial off-the-shelf solutions. As mentioned before, this often concerns the processes that are increasingly specific to your organization and contribute to your competitive power. Looking at your business from this perspective makes it easier to focus on how you can differentiate with your digital capabilities to win.
And, looking more closely at customization, whether it concerns a complete system or small add-on solutions that extend an off-the-shelf system, this can be done with high-code as well as within a low-code platform. However, the use of the right low-code platform can offer significant benefits, including a lower total cost of ownership, speed in delivery and ease of maintenance.
For example, when you look at the DevOps burden for complex systems (e.g. ERP) built with a low-code platform like USoft, we typically see a team size that is 20-40% of what's required for off-the-shelf or customized solutions. That's a significant cost improvement! Even apart from all the agility benefits toward future developments.

"We believe organizations should focus as much time, energy, and budget into digitalization. This allows you to excel in areas where you can differentiate with a superior digital offering (and win against your competition) or adopt fresh, smart business models where new possibilities are unlocked, such as strategic partnerships."

Hans Canisius, CEO at USoft



6

The importance of business rules in modernization and other software development

It's not what you do, but how you do it. Each business has processes that are unique; your expertise and the way you operate is your competitive advantage. However, to compete in today's digital landscape, your business needs to offer more than just processes. It must act with intelligence too.

To do this, you must use business rules as a necessary companion to your business processes. These rules are written in easy-to-understand sentences and describe specific criteria for an action or process (as well as defining key terms). An example of a business rule might be: 'when an order is sent to country X, then it must be sent via logistics company Y.' With just a few words, it is clear what can/needs to be done in a given situation.
By working together with business rules, your business processes can become more intelligent and more adaptable. And so can your software.

Although they're often hard to extract, your business processes always include business rules in some form, even if they exist scattered throughout various documents, memos, and ingrained behaviors.
Therefore, a critical factor in the success of your modernization is your ability to capture and reapply these business rules. Without proper attention to your business rules, the software you build will be unable to do what you require. But by using business rules as the foundation for your software, you ensure the finished product is and remains fully aligned with your business.
Unlike business processes, which change over time, your business rules tend to be more stable. They're highly business-focused, specific, and easy to validate. This makes them the perfect way to model your business in a highly granular way.
Software built based on business rules is generally more robust and future-proof compared to flow-oriented software. With this in mind and to enable a business rules-centered development process, the USoft platform includes a powerful business rules management feature that keeps development focused on the business perspective.
As a result, the solutions built using USoft are often 200% or 300% more efficient, compared to commercial solutions or custom-coded software.

WHAT ARE BUSINESS RULES?

PROCESSES
Focus on HOW you do things
Sequence orientated



BUSINESS RULES
Focus on WHAT you do
Conditions and criteria orientated

1. Both are part of **business conduct**
2. They **work in unity**
3. **But** many business aspects **do not surface** in processes

THE POWER OF BUSINESS RULES

1. Business rules are **specific**
2. They are **easy to read**
3. They are **autonomous**
4. They are **very stable** over time
5. And they are **100% business orientated**

HOW THE USOFT LOW-CODE PLATFORM USES BUSINESS RULES TO BUILD ROBUST CORE SYSTEMS

Most low-code platforms work on the principle of process flows, which are then automated – but the USoft platform is quite different.

Instead, the focus is on extracting business rules and then automating these rules in software. We place your data at the center of this model, with a protective layer of rules surrounding this data. Like a shield. This ensures that the only way to interact with the data is by complying with your business rules. This ensures that your information is continuously safeguarded, resulting in a very high quality and consistency of the data – an essential criterion for mission-critical software.
Another advantage comes from using your own business rules to shape the development trajectory. This makes the finished solutions very specific to the organization and its unique advantages.

As already touched upon, while processes may change over time, business rules are very stable and easy to update without affecting the processes themselves. By building your core software on such a stable framework, the finished solution retains the same stability and reliability, while being able to make changes and new wishes develop within your business processes.

> **"It's a business process that you want to automate and that you want to support, so it should be 100% business focused."**
>
> **Hans Canisius, CEO at USoft**



7

Conclusion

When it comes to modernization, it's not enough to simply rebuild what you already have in modern code. Doing this will not enable you to optimize your competitive advantage and make it harder to stay competitive in the future.

To be effective in your modernization efforts, you need to identify your business needs going forward, and focus on defining a clear path from where you are now to where you need to be.
In this white paper, we've outlined the 3 pillars of modernization, and the steps you need to define your objectives. We've also seen that there are many different technology options for building the solutions you need, including how low-code technology can be a driver for transformation and contribute to accelerating your transformation efforts.
It's reassuring to know that you don't need to replace your entire legacy system in one 'big bang' modernization push. Instead, you can apply an incremental and low-risk approach, supported by (semi-) automated tools that analyze your current code, help to distinguish between functional units within this code, and extract most of the rules and knowledge it contains. This enables a sharp focus on each function, and allows you to perform a low-risk incremental modernization.

Step by step, you can gradually migrate your legacy application to a new and future-proof solution, with your own business rules at the center of it all.

As an organization, it's essential you retain your unique competitive advantage. This is why you need to focus modernization efforts on the areas that make the most difference, without reinventing the wheel by re-coding things that already work well or are readily available. This way, you conserve time and resources for where they make the greatest impact.

This is an efficient way to keep all the systems that work well, and concentrate only on what needs to be modernized. And the result is reliable core software that better supports your organization with additional intelligence and modern functionalities.

The information in this ebook has been compiled with care. Nevertheless, USoft cannot accept any liability for the consequences of incompleteness or inaccuracy of the material in this ebook.

No part of this document may be reproduced in any form without written permission from USoft B.V.

Authors: Hans Canisius, Tom Schellekens
Design: brandfabric
Photography: Unsplash

About USoft

At USoft we understand that organizations desire the freedom to design their own business operations. We believe that software may never pose an obstacle and should instead clear that path. USoft offers organizations a low-code development platform for developing and redesigning business-critical systems.

The intelligent, flexible low-code platform is the perfect solution for clients who want to innovate fast and adapt to the needs of the market. Our technology is engineered around the power of business rules so that your organization's context is always at the heart of everything you do. To achieve the best result, USoft always immerses itself in the essence of the client's business operations and has a great track record for client satisfaction and excellent service.

USoft's low-code platform is used worldwide in government, aviation, education, financial services, transportation and logistics.

For more information, please visit **usoft.com** and **community.usoft.com**.

USoft B.V.

IJsselmeerweg 1

1411 AA Naarden

The Netherlands

+31 35 699 0699

info@usoft.com

© 2023 USoft B.V. | All rights reserved | Classification: Public

usoft.com