



# 8 Lessons in Low-Code

How to leverage **low-code technology**  
to create **real business value**

“ Whether you’re thinking about developing software solutions with the help of the Mendix low-code platform, or if you’ve already reaped the benefits of your first low-code project: apply these lessons from Bizzomate and optimize the business value generated by your own low-code solutions. ”

**Johan den Haan**  
CTO Mendix





# Introduction

Disruption defines our world. The world and the needs of customers constantly evolve and organizations are struggling to keep up and stay relevant. In this world software is not merely an enabler but a changemaker. It fuels digital transformation and determines whether companies live or die. Lots of leaders acknowledge this and are looking for ways to speed up digitalization efforts and to cope with a lack of software developers. Low-code is a powerful answer to their needs.

As low-code experts and long time Mendix partner we've advised and supported companies in different industries in numerous digitalization projects. We've helped them challenge set thinking patterns, facilitate better and agile collaboration and fully leverage low-code technology to create solutions that make them frontrunners in their respective business. While working closely with our customers on these projects we've gathered valuable insights that are indispensable for anyone who wants to start, or already is working with low-code.

In order to help more organizations to get the most out of their low-code investment, our experts have therefore assembled our most valuable insights and turned these into the 8 Lessons in Low-Code. I'm convinced that these lessons will help you thrive, be successful in your digitalization efforts and will contribute to building a future-proof business.

**Marc Gelissen, CEO and founder Bizzomate**

# Table of contents

LESSON

**01.**

Start from a shared understanding

LESSON

**02.**

Don't focus too much on the speed of development

LESSON

**03.**

Think outside the box

LESSON

**04.**

Teamwork makes the dream work

LESSON

**05.**

Embrace the cloud

LESSON

**06.**

Work with an open platform

LESSON

**07.**

If you're building software, you're a software company

LESSON

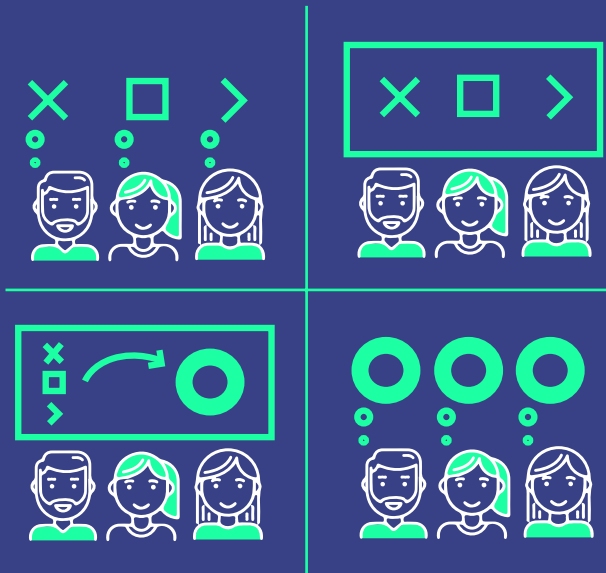
**08.**

Don't waste time reinventing the wheel

CONCLUSION

**Next steps**

The ultimate goal



## LESSON

# 01.

## Start from a shared understanding

What does the ideal wine glass look like to you? This question may sound silly as a wine glass is a pretty standard utensil. However you'd be surprised by how many details a wine glass contains and how they are all critical in the wine drinking experience. So before you ask someone to randomly create a wine glass, you better make sure to provide the craftsman with more details like the context of your intended usage. For instance it makes a difference whether you will use the glass to drink red or white wine or if you're one of these people that like to smell the wine while taking a sip.

Creating the right application with a low-code platform isn't any different from this. It all starts with the end user. What is the ideal wine glass to this person? And does this meet with your idea of the ideal wine glass? This is never a 100% match. So in order to create something that delivers value you need to take a step back and understand where the end user is coming from. In what context will the solution be used? What issues does it need to help solve?

### Identify the right thing to build

At Bizzomate we've experienced that with low-code technology building the application right is easy. The true challenge lies in identifying the right solution to build. To help our customers doing this we've created a four-step approach. Working according to the phases of Discover, Define, Design and Do ensures a creative and pragmatic approach to identify the right problem to solve before you start building a solution. So before you even think about how to build it, take a step back. At



the start of a process to create a digital solution the end user/ domain expert from the business and the developer should come to a shared understanding of what solution is the right answer to the question.

### Visualisation is a powerful tool

In order to help create a shared understanding visualisation is a powerful tool. This is the clearest, most unambiguous way to transfer knowledge and explore ideas between business domain experts, developers and other stakeholders. For instance let the business colleague sketch the ideal solution without being held back by the limitations of what the technology is able to deliver. The sketch and the discussion that follows is a powerful and fast way to build shared understanding. On the developer side you can benefit from the visual development environment that low-code provides to create a solution. This visual environment enables developers to actually take their business colleagues with them on the journey to build the right solution.

Another critical element in coming to a shared understanding - and to not lose this along the way - is agile working. We know, everybody is agile these days. But, bear in mind it takes a lot more to be agile than just a different way of working. Agile is a mindset and being truly agile implies organisational change. For instance it also means that the power to make decisions on what the result of the low-code solution should look like, is in the hands of those who create it and are impacted by it. In other words agile gives the developer and the domain expert from the business the space and right structure to come to a shared understanding and collaborate to achieve the best result.

## LESSON

# 02.

## Don't focus too much on the speed of development

Dramatically increased velocity in the process of application development is one of the big promises and advantages of low-code. It helps you to move fast and it can take you from idea to production in a matter of days or weeks. However this velocity can also lead to too much focus on the speed of the development process, tight deadlines and a result that isn't the right solution for your problem.

Indeed, low-code development enables you to develop software applications ten times faster than traditional app development. The process is abstracted, automated and model driven. You can easily develop applications with drag-and-drop components without being bothered with what's under the hood. This means you don't have to focus on the technical details like you would in a traditional development process and you can spend more time on the business problem that needs to be solved.

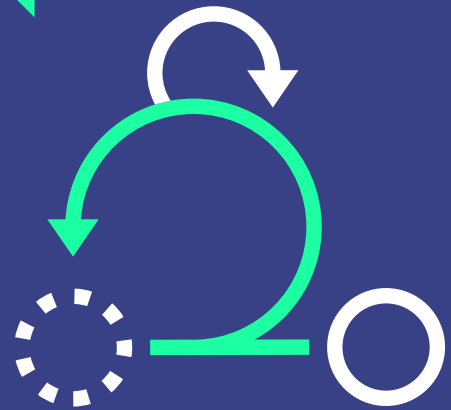
### Experiment, iterate and fine tune

However the real benefit of this increased velocity is that it doesn't only provide you with the opportunity to build applications faster and in the right way. Leveraging the speed of low-code is also - and maybe even more important - about using the flexibility of this technology that enables you to experiment, quickly iterate and fine tune.

Are we still solving the right problem? Is this what the business needs or should we add other functionality and test what works best? The speed of low-code gives you the opportunity and flexibility to innovate, to easily create prototypes and constantly go back to the end user to validate if you're creating the right solution.

### The risk of moving too fast

If you're building the right solution you're ensured that there will be a positive ROI. If you move too fast it can be an expensive ordeal to transform the solution that was delivered into the solution that truly delivers business value and that contributes to the success of the company.





“ The true challenge lies in identifying the right solution to build. ”



## LESSON

# 03.

## Think outside the box

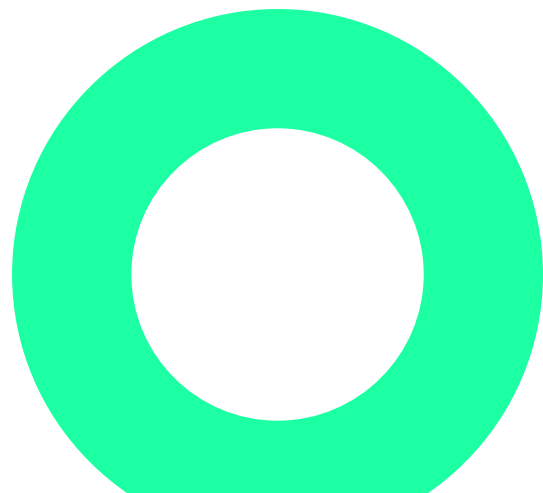
Simply digitalizing a process doesn't mean that you're being successful in digital transformation. In fact digital transformation provides you with the opportunity to look at your current processes and culture in a completely different way. We've experienced that customers that are looking for a low-code application to replace a legacy solution, are having a hard time to break free from the framework that is set for the process they are trying to optimize.

The way people work is often based on the possibilities and limitations of the solution they use. This means in order to build the best solution, you need to let go of these restraints and have an open discussion about what the real goal of the new application should be. This is also where innovation starts. Begin with the question of how people would want to work in an ideal world and what can be improved in the current process. This way you can create a solution that ultimately supports people in doing their job in the best and smartest way possible.

### Free from limitations

For example: do we really need to build software that can print all kinds of forms, or could we completely eliminate the need for printing by supporting a new way of working? Going back to the basis, the goal of the process and what is actually needed in order to be successful, will help you to create a more innovative solution in less time, with lower costs and more value. By thinking out of the box, you're freed from the limitations that you were not aware of but that kept you confined to a box that prevented you from truly transforming and being innovative.

When building a solution for a brand new process it is easier to think out of the box as you're still working from a conceptual point of view. But even then it's important not to step into the pitfall of going for the answer or 'obvious' solution that's right in front of you.





## LESSON

# 04.

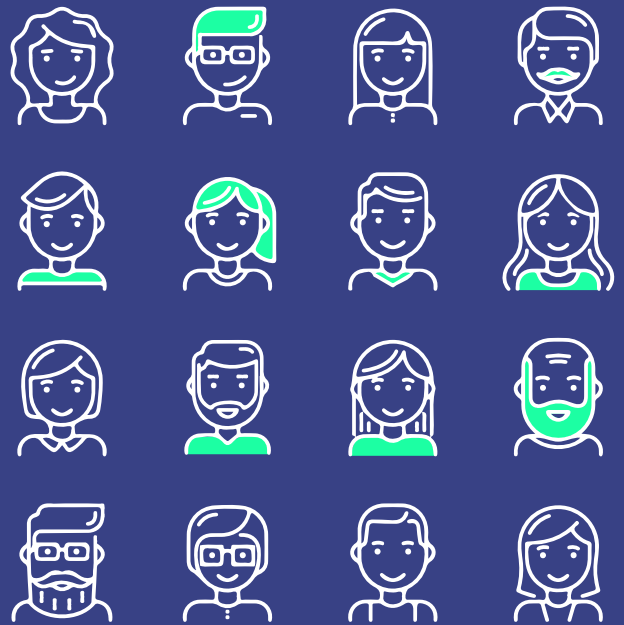
## Teamwork makes the dream work

We already stated that close collaboration between the business domain expert and the developer will lead to the best result of your low-code project. It's important to realize that this goes way beyond the shared understanding that needs to be created before you start building the application.

The great opportunity working with low-code provides, is that you can leverage the possibilities for close collaboration whilst making the application. This is called multi-user development. The Mendix platform enables you to work together on an application at the same time. The platform supports and synchronizes the various workstreams. Developers, from different levels can contribute to the solution and learn from each other along the way. Also the platform makes it easy to assign certain tasks to people with a specific level in software development.

### **Grow competencies**

Approaching the actual development process as a team with people from the business and developers, empowers organization to incorporate different perspectives. Everyone contributes to the end result from start to end. Business professionals get the opportunity to learn about low-code development and grow their competencies. But it's also important to make sure there's a right balance in the sprint. It's not only about educating business colleagues about low-code development, but about guiding the team in the right direction.

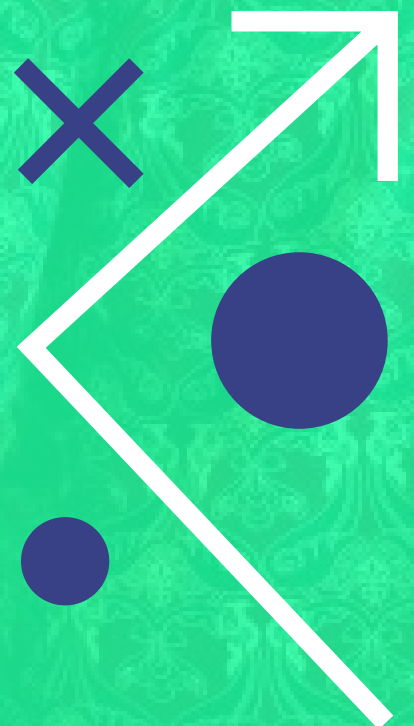


If business colleagues collaborate with developers it becomes much clearer what implications including specific functionality will have on the development process. This way a consultant can easily clarify that building something exactly as requested could take as much as three weeks while an alternative that may fulfill (almost) the same business requirements, could take as little as a few days. By actually showing what it means in practice, multi-user development doesn't only facilitate collaboration but also boosts a shared understanding during the development process.

We've also experienced that multi-user development can lead to surprises as some domain experts show a special talent for development. As they also have the domain knowledge this makes them extremely valuable team members. On the other hand there are professionals who are strong in conceptual thinking and add value from this side. It's all about looking who can contribute what and reap the benefits from different viewpoints.



““ The speed of low-code gives you the opportunity and flexibility to innovate, to easily create prototypes and constantly go back to the end user to validate if you’re creating the right solution.””





## LESSON

# 05.

## Embrace the cloud

It's a no brainer: when creating your low-code application, it needs to be available. If people can't use it, it doesn't deliver value. Therefore the ability to easily deploy your solution and the guarantee it's scalable, are vital for the success of your low-code application. In order to ensure this the cloud is indispensable.

The cloud isn't only a key factor in the deployment of the application. Also during the development process it ensures fast collaboration and iteration. Working with the Mendix platform it takes you merely 15 minutes to set up a production environment for a cloud native application. This way you can easily start using 'version 1' of the solution you're working on with your stakeholders. Working with the cloud therefore also supports agile working.

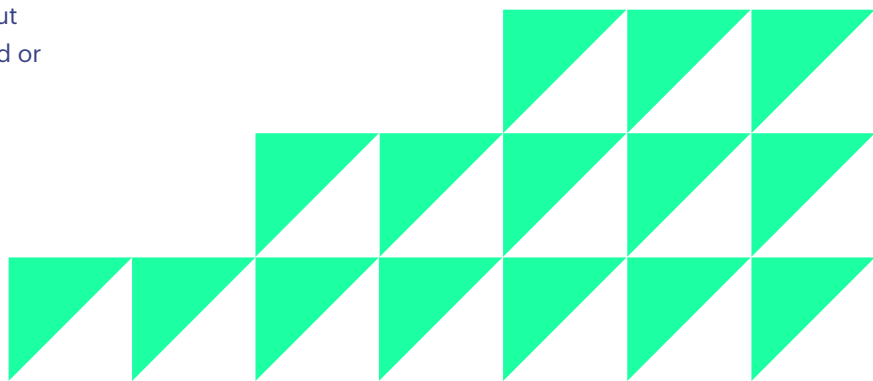
### **Scaling up fast**

Think about how much time it would take you if you needed to place your application in a private cloud or on-premises yourself. How fast would you be able to demo the app in between iterations? How much time would you ultimately lose in bringing the app to market? And how would you make sure the application is able to handle 100.000 users without faltering? Placing the application in a private cloud or

on-premises and scaling up fast when needed, would also require system managers with other skills and knowledge and a robust process.

By integrating this into the platform and having a dedicated cloud-team in place, Mendix ensures that you can easily develop, deploy and run your application in the cloud with a smooth experience for the end user anytime, all the time. Moreover with the growing popularity of microservices, the cloud is the go-to environment that connects teams and applications in the same project.

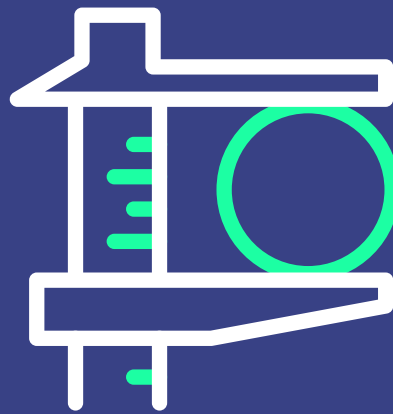
Another important element that you can leverage by working with the cloud is data. Smart applications and services generate massive amounts of data. Machine learning and AI can help analyse this data and provide you with significant insights. If your application doesn't run in the cloud, it will be much harder to integrate analytics and to use this data for strategic decision making.



## LESSON

# 06.

## Work with an open platform



The great thing about low-code is that it provides you with building blocks to build your application. This ground level of functionality ensures rapid development and enables developers of various levels to build applications. However from experience we know that there is always something you want in your application that isn't provided by the building blocks of the platform you work with. This then needs to be created separately with another system. Therefore working with an open platform that doesn't limit your possibilities, is incremental.

As much as we love that with low-code the sky's the limit, a side note is in place here. Consider carefully what functionality you really need. Although API's and SDK's make it very easy to connect to functionality created outside the low-code platform, adding lots of other stuff can make your project complicated. It will also make it harder to ensure governance. Also when developing additional functionality yourself, it's smart to closely collaborate with a technical team member to make sure that what you're creating will integrate with the rest of the solution. However when you have a solid process in place for this, the sky truly is the limit.

### Talking about integration

This is another great benefit of openness. Your low-code application needs to seamlessly integrate with systems and data sources that are currently used by the organization. It needs to stand the test of time by being able to integrate with new technology. Furthermore the application itself must be flexible to adaptation and optimization. This way you will create an application with a positive ROI, now and in the future. Our biggest lesson learned on the integration topic is to involve the integration experts early in the project in order to run in to delays. It is even our recommendation to make the integration experts part of the SCRUM team.

Another advantage of working with open technology is that you can profit from a vast community that also is developing solutions. When working with the Mendix platform you can leverage all kinds of functionality and widgets from the app store. This way you can reap the benefits of what other innovative minds have created and incorporated this in your project. And since Mendix also enables you to transform the logic you've used to another development language, you're free to check out anytime to like.



“ If people can't use it, it doesn't deliver value. ”



## LESSON

# 07.

## If you're building software, you're a software company

With so many software solutions on the market, you can also think why bother to create it yourself? Custom solutions often don't meet the specific needs of an organisation. It's too rigid and puts you into a straightjacket. Not ideal if you're looking to make a difference instead of blending in with the crowd (and your competitors). Low-code offers the opportunity to make your own solutions on your own terms and build them together to meet the unique needs of your organization. These own solutions enable your organization to differentiate against the other players on the market. However what a lot of companies don't realize: If you're building software solutions, you're a software company.

### 🕒 Document important elements

At Bizzomate we've noticed that the implications of this are very much underestimated. If you build a solution, you also need to think about governance, control and the lifecycle of the product. This entails having a process in place and people who are responsible.

It all starts before the development process takes off. The biggest pitfall in agile working is the often used mantra 'communication over documentation'. However if you're building a software solution you need to make sure to document important elements. For instance define what the modeling guidelines are so you're ensured that everyone works in a consistent way that ensures quality. This goes from a uniform way of naming the application to the way you build these. Also it's advisable to document what kind of solution you've built: what are the different elements of the application and how does it integrate with other solutions? Furthermore you need to

agree on the definition of done; when are specific parts of the project, user stories and sprints, and the project done according to the standards that are set? If you all speak the same language quality is guaranteed and it's much easier from a support perspective to solve issues.

### 🕒 Lifecycle management

When we deliver an application to our customer, we also train someone from the organization to handle first line support. Also we make sure that the responsibility of the lifecycle management of the application is assigned. Agile working helps to organize this effectively as it implies that you put the responsibility for the application with the people who are closely involved with the application on a daily basis. Also in house knowledge enables you to set up a DevOps-structure and to continue optimizing your application.

### 🕒 Pentest for business-critical applications

With growing threats and cybercriminals that are becoming smarter to make their way into the organization, it's obvious that the security of the applications needs to be guaranteed. Thanks to the robust Mendix platform and the modeling guidelines, a high level of security is taken care of. However if your applications handle sensitive, business-critical, personal information, it's wise to have a security policy in place. Also measures like execute risk assessments and carry out pentests decrease the risk that your application will fall victim to hackers.

LESSON

# 08.

## Don't waste time reinventing the wheel

Nobody wants to waste time developing a solution that has been created before. The great thing is that low-code doesn't only boost productivity in software development by providing ready to use building blocks, it also opens up a world of best practices. All there for the taking.

Through the Mendix App Store and developer community you have access to a vast amount of proven pre-built components, connectors, features and functions. Developers can take what they need and include it into their application. New features and

functionality can be added in the blink of an eye. Also there's a lively forum where developers are eager to help out if you're stuck in a project.

Not only Mendix makes it easy to leverage the work of other smart developers. At Bizzomate we also have our own app store with proven, successful components and solutions. Besides this we enable our customers to learn from best practices of projects we've done in various industries. This way they can benefit from our knowledge and experience in solving common problems in their respective industries.



## CONCLUSION

# Next steps.

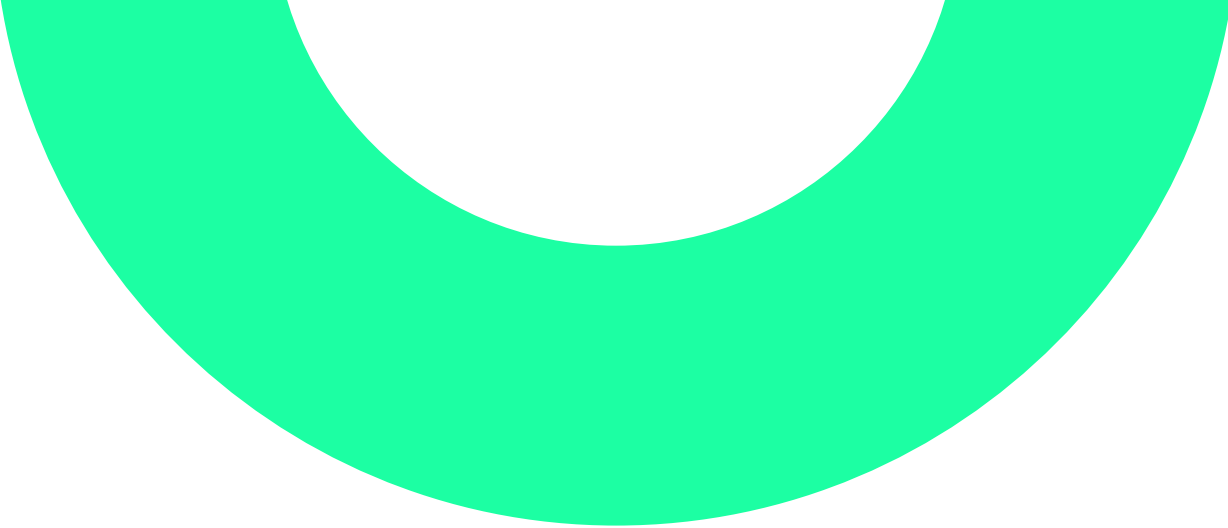
## The ultimate goal

Ultimately the goal of working with low-code is to deliver the right solution that delivers business value. We hope these 8 lessons give you a good idea of the critical success factors that help you get the most out of low-code technology.

- 01. Start from a shared understanding**
- 02. Don't focus too much on the speed of development**
- 03. Think outside the box**
- 04. Team work makes the dream work**
- 05. Embrace the cloud**
- 06. Work with an open platform**
- 07. If you're building software, you're a software company**
- 08. Don't waste time reinventing the wheel**







# Contact us now to rock your organization with low-code

Are you ready to rock your organization with low-code? Or do you want to know more about these lessons? Contact us through:

**Bizzomate**

+31 (0)43 3111 05 70  
info@bizzomate.com  
www.bizzomate.com

**Bizzomate Valkenburg**

Nieuweweg 25  
6301 ES Valkenburg (LB)  
The Netherlands

**Bizzomate Dordrecht**

Burg. de Raadsingel 93  
3311 JG Dordrecht  
The Netherlands

**Bizzomate Amsterdam**

Boeing Avenue 215  
1119 PD Schiphol-Rijk  
The Netherlands