

YOUR NEW **GOVERNANCE** FRAMEWORK

Bring Value Back to IT Governance

BUCKLE UP



mx mendix



Introduction

How do you lead your organization through digital transformation, while enabling true change management and maintaining visibility into your IT and data landscapes? Step one: your governance framework. Make sure your governance strategy ensures IT and business stakeholders are aligned to business goals and can generate value with every technology decision.

To really help you put this framework into action, we've created a DIY template that you can download and start applying to your business. Look for this icon and click to download.



Table of Contents



IT GOVERNANCE	04
3 RULES + YOUR FRAMEWORK	08
CORE MISSION	10
FOUNDATIONAL PRINCIPLES	12
ORGANIZATIONAL GOVERNANCE	14
TECHNICAL GOVERNANCE	22
GOVERNING THE IT LANDSCAPE	29

**IT Governance:
How We Got Here.
How to Fix It.**

Tied up in ‘Nots’

A question for you: **When did governance become just a means to say “no?”**

It’s frustrating. It’s frustrating to your business stakeholders who hear it, and it’s no doubt frustrating when you’re forced to say it. When implementing governance around your tech stack, your goals are to always keep your company safe and secure. However, security and governance are not synonymous. Keeping your company safe doesn’t mean setting up an impenetrable wall of red tape, saying “no” and “do not” and hiding behind security measures.

You understand that a company’s safety involves more than just security.

Part of keeping a company safe is making sure it’s viable and competitive for the foreseeable future. This means putting into place measures that foster the continuous business-value generation. But bad governance and “no” get in the way of that.

The concept of IT governance emerged around 1993, stemming from corporate governance—a means to manage and operate an organization. The overall goal of IT governance was for IT departments to manage the use of information technology to ensure that they fostered business value creation, as well as asserted accountability for the management of IT departments.

The early goals of IT governance were hopeful:

- 1_Assure that technology is being used to generate business value
- 2_Oversee management’s performance
- 3_Mitigate the risks associated with using technology

Sounds like business nirvana, doesn’t it?

Over the past decades IT governance has morphed into the antithesis of how it began. So much so, that it strikes fear into the hearts of nearly everyone doing day-to-day IT work and can tie you up in knots.

Your governance pain lies in its very definition. Or, rather, definitions. As the concept took further concrete shape, an IT governance ISO standard was adopted in 2008 (updated in 2015) that “applies to the governance of the organization’s current and future use of IT including management processes and decisions related to the current and future use of IT.”



Governance is “leadership, organizational structures, and processes to ensure that the organization’s IT sustains and extends the organization’s strategies and objectives.”ⁱⁱ

– ISACA, COBIT governance framework

IT governance is “specifying the decision rights and accountability framework to encourage desirable behavior in using IT.”ⁱⁱⁱ

– Peter Weill & Jeanne W. Ross

“IT Governance: How Top Performers Manage IT Decision Rights for Superior Results”

Governance is “a formal way to align IT & business strategy”, with frameworks that helps you “determine how your IT department is functioning overall.”^{iv}

– Kim Lindros

CIO.com, “ABC: An Introduction to IT Governance”

The definition of governance keeps growing. On top of COBIT, there’s also ITIL (focused on IT service management), COSO (risk management), CMMI (performance improvement), FAIR (risk quantification).

Pulling the thread from all these different notions of what IT governance is, you can see (and you probably experience it every day) that the goals of governance have become obfuscated. They’ve moved away from managing the decision-makers and strategically generating business value through IT and have moved toward managing the doers and creating business value impediments.

And now you’re in the unenviable position of incorporating a governance framework written on a spool of red tape, one that makes your colleagues in IT grind their teeth, business stakeholders pull their hair out, and generating business value become a third-tier priority.

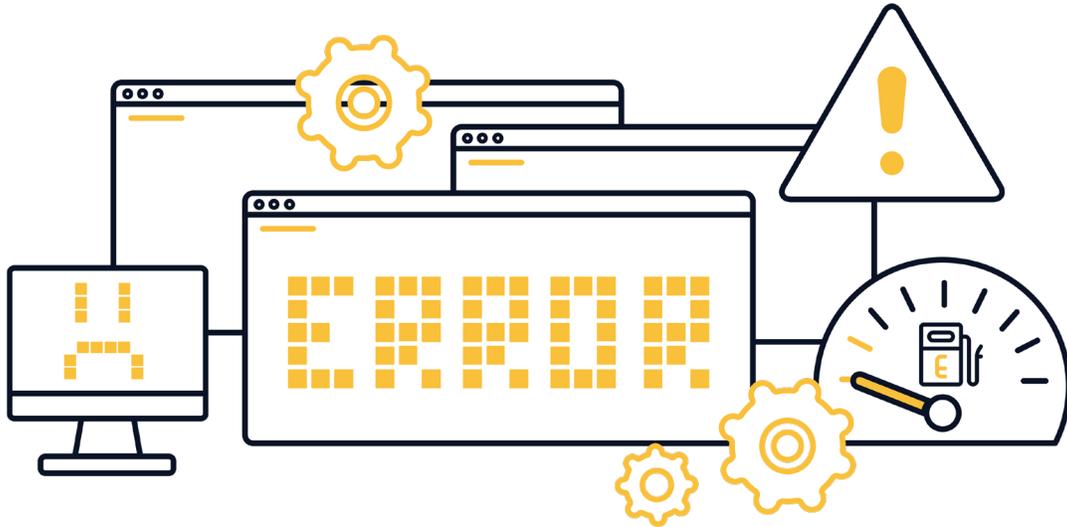
You’re Not Alone

Current IT Governance approaches aren’t only challenging, they’re failing.

In a Gartner survey, “Balancing Autonomy with Control: New Governance Models for Digital Business”, the authors cite that traditional governance approaches are under-performing for leaders who want the business and IT to collaborate more.

In fact, 70% of those surveyed stated that their companies’ standards were not designed to apply to digital business teams^v

Qualitatively, bad IT governance can hamper a company’s culture. Rowlands, De Haes, and Grembergen suggest in their work “Understanding the Dimensions of IT Governance Culture” that “culture and behavior of individuals and of the enterprise are very often underestimated as a success factor in governance and management activities.”^{vi}



Rather than fostering business value creation, IT governance is more often hindering it.

IT governance doesn't have to be that way, though.

Cut that red tape

It's time to re-think the way we drive our business forward with IT governance. It starts by going to back to the original intent of the concept:

- 1_Generate business value
- 2_Manage the managers & their performance
- 3_Mitigate risks

But executing, putting digital ink to virtual paper, is a challenge. Where do you begin? Where do you know where to tighten and loosen the needed constraints? Who do you involve in decision-making? How do you do all this without spending years with consultants? How do you evenly distribute the weight of all three goals?

We've come up with a governance framework that works no matter what IT initiative you plan to implement, or technology you use. This framework allows you to keep your company safe: secure and compliant while fostering—not hindering--business value generation.

Throughout this book, you'll find a template that you can use during your day-to-day. We'll show you how to establish a core mission, create foundational principles, and think about the organizational and technical implications of adopting a new initiative or technology. **This is your new governance framework.**

3 Rules & Your Framework

Remember Your 3 Governance Rules of Play

Your governance framework should be:



1 Tailored specifically for your organization. It's not a catch-all.



2 Specific to the program or initiative it's governing. Every technology and line of business has very different needs.



3 Concise and purpose-driven with your framework. Think about the value you will create and how it's measured.

A governance framework needs to be clear and always drive back to the business strategy to which it's aligned—which is why our framework maintains that simplicity. This framework is comprised of four parts and a set of ground rules to make sure every governance model you

build stays on target. Ultimately, when it comes to good governance just remain: Clear, concise, and relevant. Throughout the rest of this book, we'll be providing examples, using the implementation of a low-code development platform as a hypothetical scenario.

This governance framework has four parts

1
The Core
Mission

2
Foundational
Principles

3
Organizational
Governance

4
Technical
Governance

PART I

Establishing & Communicating Your Core Mission

Your core mission is, simply, what is the initiative and why are you doing it? Clearly stating your core mission at the beginning of a project and at the very top of your governance documentation ensures consistency and alignment throughout your entire organization. You want to clearly state why it's important to the organization, and to do it in just one or two sentences.

Examples

Let's say you're implementing low-code technology into your organization. You aim to change the way your entire organization (not just IT) creates software that solves problems and helps achieve business goals. A tight core mission statement may look like this:

The Core Mission of our low-code initiative is to empower our business and IT to co-create and streamline the execution of innovative, high-quality solutions that measurably achieve business goals.

So, what does a bad core missions statement look like? Try this:

The Core Mission of our new low-code initiative can be broken down into two parts. One, bring to key business and IT stakeholders a new set of diverse tools and technologies from the paradigm-shifting low-code market. Second, ensuring that such tools and technologies drive acceptable synergy among departments and increase productivity, innovation, quality, timeliness, and acceptance of solutions measured to current internal standards, while maintaining data security compliancy, as well as alignment with industry best practices.

Wow. We could go on for many paragraphs as to what's wrong with that statement. But a few key points: It's trying to do everything at once, so it's saying nothing at all. It has too many buzzwords that can be misconstrued. Finally, it's too long and can detract the reader from the core mission itself.

Avoid the pomp.

Try it for yourself →



Part II

Laying Down the Foundational Principles

The foundational principles of your initiative serve as the bedrock on which your model rests. **Any action taken or decision made need to align with foundational principles.** Your foundational principles—regardless of technology—should drive management behavior and performance by demonstrating expectations for tools and the support and empowerment you'll need. Foundational principles generally describe means of measurement, empowerment or support, quality assurance, standardization, or innovation—to name a few. Limit your principles between three and eight.

Examples

What does good foundation principle look like? Getting back to our hypothetical low-code development platform initiative, let's see how we would talk about measurement.

Measurement – *As a Foundational Principle, Measurement means that since what is measured drives behavior, we will determine success and improvement of low-code development by measuring KPIs such as time-to-market, revenue generated, and IT savings. KPIs will be agreed upon by everyone in the initiative and they will be reevaluated regularly.*

Another example, this one around the idea of empowerment.

Empowerment – *As a Foundational Principle, Empowerment means that everyone in the initiative will have the tools and support needed for both success and self-determination with a low-code platform, including, but not limited to, training, mentorship, necessary software & hardware, delegated authority, and the authority to change this document via a simple process.*

Here are some unhelpful foundational principles.

Measurement – *We want to make sure that everything is measurable. We'll look at ROI and make sure that people are using the low-code platform we invested in.*

Empowerment – *Train everyone on the platform.*

These are concise, yes. But they're unclear and the KPIs mentioned in the first are somehow both too broad and too narrow. Be explicit.

Try it for yourself →



Part III

Organizational Governance: Following the 4 Ps

Also known as Functional Governance, this aspect of the framework provides ways to activate and enable the people involved with the initiative, the processes to complete it, the right ways to use the platform you're implementing, and that platform's output. The Organizational Governance section is made up of multiple sub-sections that tackle the organizational specifics, policies, and best practices needed to ensure the goals of IT Governance are met for your initiative: business value is created, management performs appropriately by making decisions in alignment with delivering the desired outcomes from the Core Mission, and risks are mitigated through relevant best-practices and policies.

To achieve this, we have seen great success in the field using a model called the 4 Ps: Portfolio, People, Process, and Platform. Let's look into these sections deeper.



Guiding your company through its digitalization journey requires having insight into new solutions being built, who is building them, and how they're built. With Mendix's Control Center, you get built-in governance tooling: a dashboard that gives you complete visibility into your app landscape (developers, access, deployment environments, security measures) so you can effortlessly govern it.

Portfolio

The goal of the portfolio section is to help determine what constitutes the portfolio for a given initiative (e.g. cloud resources, servers, applications, COTS solutions, documents managed) as well as how to manage the portfolio and demonstrate success. Sub-sections here could include Value-Based Prioritization, Measuring Portfolio Value, Demonstrating Portfolio ROI, and so on.

Portfolio – Examples

One typical example we see with our customers is Measuring Software Value, teaching them how to quantify the business value expected and delivered by a given software solution. Here's how that might look in a real governance model for a low-code development initiative with an explanation of value-based portfolio, and a value-based intake form.

Creating quantifiable value is the cornerstone of each decision we make. To ensure we measure success at all stages, we need to define a value-based business case for each app we build with Mendix. Value metrics to consider include: costs saved, revenue generated, time-to-market accelerated, application development velocity, customer engagement and churn, licenses expanded.

Customer Portal (B2C)

Current Situation

Our customers lack an easy way to view their services and products as well as see other services and products available.

These limitations lead to difficulty engaging with the customer, churn, and difficulty in expanding licenses.

Description of the idea (“we believe”)

We believe that with a low-code-built web and native mobile application, we can quickly lower the rate of churn as well as increase traffic to other services that we offer.

Enabled by:	What will it solve/create?	What are the affected KPI's?
1_ Secure log-in procedure. 2_ Responsive design. 3_ Integration with sales and marketing systems.	1_ Opportunity to sell relevant services to customers 2_ Remove friction from the buying process. 3_ Increase renewal rate.	Increase in licensed expansion revenue \$10,000 p/m Increase in customer renewal rate 10%YoY Total Business Value: \$120,000 p/y

People

Regardless of the technology you're trying to implement, the People portion of your governance model is vital. In this section, you express how staff should be allocated, roles required, and how organizational structures should be setup to ensure the

success of the initiative being governed. Sub-sections here could include maturity level organizational structures, Center of Excellence responsibilities, role and responsibility mapping, staffing guidelines, and so on.

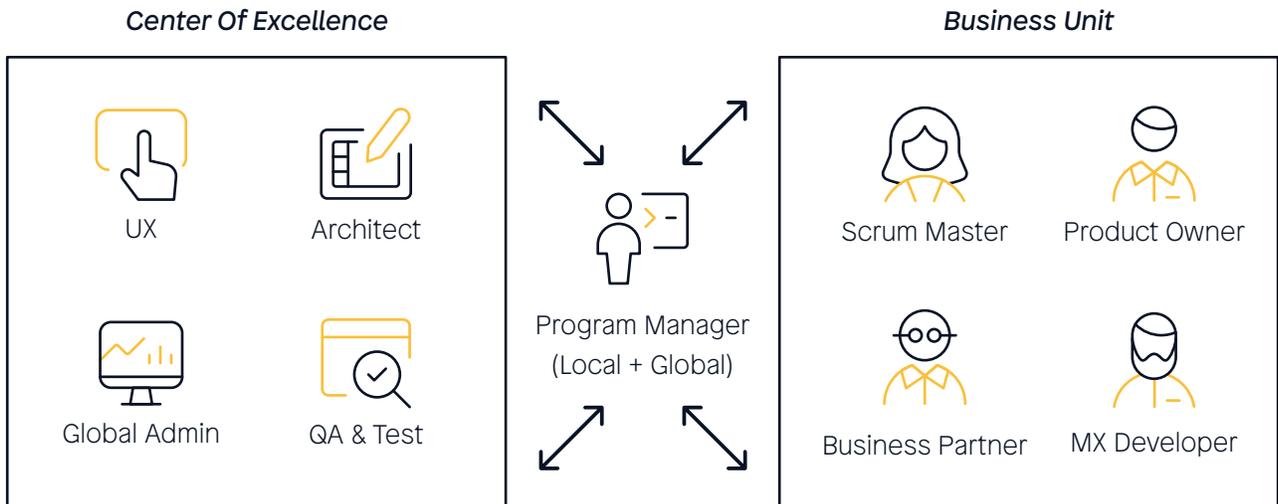
People - Examples

As an example, with low-code development platforms, citizen development—people without programming backgrounds creating technology—comes into play. Ensuring the right people can build and deploy safely requires training and putting into place parameters for building in the platform. But you also need to make sure that your more technical developers are involved in the process as well. Establishing a Center of Excellence allows you to easily dole out roles and responsibilities to people in IT and the business. You can allocate staff and also create rules around collaboration within teams and across silos.



The Mendix Platform provides two skill-based IDEs that allow developers with a range of programming skills to build and deploy solutions in a collaborative manner.

Establishing your Center of Excellence (below), means creating a list of roles and responsibilities (next page).



Role + Placements	Responsibilities
Program Manager CoE	<ul style="list-style-type: none"> · Owns program + continuous improvement · Leads KPI + success reporting · Keeps program aligned to Mendix roadmap
Portfolio Manager CoE	<ul style="list-style-type: none"> · Owns portfolio rationalization + prioritization · Owns portfolio business value KPIs · Allocates COE resources
Enterprise Architect CoE	<ul style="list-style-type: none"> · Drives architecture alignment through collaboration · Enforces low-code technology standards + reuse · Drives roadmap for platform updates
Global Administrator CoE	<ul style="list-style-type: none"> · Owns administration of the MX Platform · Documents COE processes · Monitors the MX Platform
UI/UX Leads CoE	<ul style="list-style-type: none"> · Develops + maintains design language · Creates style guides · Collaborates on specific solution styling + flow
Business Unit Program Manager BU	<ul style="list-style-type: none"> · Owns local program + continuous improvement · Reports local unit KPIs + successes · Aligns program to COE guidelines + Mendix roadmap
Unit Team Lead BU	<ul style="list-style-type: none"> · Enforces technical standards · Participates in development work · Prioritizes technical requirements when needed
Unit Developer BU	<ul style="list-style-type: none"> · Develops and delivers solutions · Collaborates with end users to ensure alignment · Prioritizes technical requirements

Process

For the process section, enablement and reuse are the name of the game. The goal of this section is to describe the relevant organizational processes and policies used for the initiative under governance as well as any new processes that will be put into place as a part of a new project. Creating new or refining current processes allows you to focus on change management, as well as get a better grasp on your application portfolio and how it's composed.

It's very important to remember the goals of IT Governance here: deliver business value, manage the managers and their performance, and mitigate risk. With these goals in mind, processes placed here should focus on the organizational business process, not the technical processes. If a process is already in place—there's no need to reinvent the wheel. Just link to the documentation directly rather than recreating it. At this juncture, it is important to review those documents and make sure that they, too, are clear, concise, and aligned to your organization's strategy. Always be asking if the processes in place are as lightweight as possible and focused directly on the goal.

Some typical sub-sections for Process are Information Security Policy, Infrastructure Purchasing & Deployment, Change Management, Management Performance Evaluation Processes, and so on.

Process – Examples

With low-code, the simplest approach to governance that we've seen our customers take is putting direct links to the software development lifecycle (SDLC) and change management documentation/websites directly in the governance documents. These documents already exist and don't need to be completely rewritten.



In Mendix's Developer Platform, you can collaborate quickly with developers of all skills in your organization. You get Agile-based tools that allow you to create and share user stories as well as a chat function that lets your developers leave messages for each other within the platform.

This makes it clear exactly what's expected and where to find more information. Additionally, our customers can clearly describe intended process changes or link to more efficient processes right in the document ensuring clarity.

Example Only

	Development We use the linked SDLC as per our organizational policies.	View Doc →
	Security We follow all prescribed infosec policies as identified here.	View Doc →
	Change document We follow the organization's change management policies linked here.	View Doc →

Don't rewrite unless these documents need to be aligned to your business goals.

Don't unnecessarily re-type or rewrite a process that already exists, add complexity to simple processes, or relist every existing process rather than limiting the list to the regular one. That just adds more roadblocks and detours on your course.

Platform

As you lead your company through its digital transformation, you need to make sure that the technology you plan to implement matches your business strategy. The Platform section of your governance framework is where you define the requirements your organization has for the platform that you're adopting. If the platform doesn't match those needs, a different vendor or component should be considered. Not only does this serve as a governance model, but it also serves as a guide in the decision-making process. Subsections here (and there should be no more than 5-10 major requirements) include Cloud Strategy Support, Architecture Support, Security Requirements, Regulatory Requirements, and so on.

As with the process section, there's no need to reinvent the wheel when describing these guardrails. If documentation already exists, for example Data Center Rack Support Specifications, just link directly to the supporting document or website.

Platform - Examples

Organizations seeking to digitalize and accelerate their software development process often choose low-code platforms to aid them in that initiative. Changing the way you do business means building scalable, resilient and portable applications that can be used across a range of devices. In this example, your platform requirements for a low-code platform would look like this:

Cloud Native: Various solutions in our portfolio have several deployment needs to mitigate cost risks that come with running them. Therefore, we need to ensure that all solutions built are scalable, resilient, and portable across different deployment landscapes (hybrid, public, private cloud and on-premises). Solutions built on Mendix are cloud-native by default.

Native Mobile: Employees need to be able to download and install a native mobile app that works online and offline. The platform we choose needs to have native mobile as a channel with support of different architecture (progressive web apps, hybrid, etc.)

Security: Because of federal regulations in our industry, the low-code platform and application built with it need to adhere HIPAA, GDPR, and PCI compliance. Please see below documentation for further guidance.

Example Only

	HIPAA This is how we support the HIPAA standard including our certification.	View Doc →
	GDPR We are compliant with Europe's data protection act.	View Doc →
	PCI Compliance Information security standard for major credit card schemes.	View Doc →



Mendix gives you complete cloud freedom. Private cloud, public, hybrid, or on-premises, you can build scalable, portable, resilient applications, all with one click.

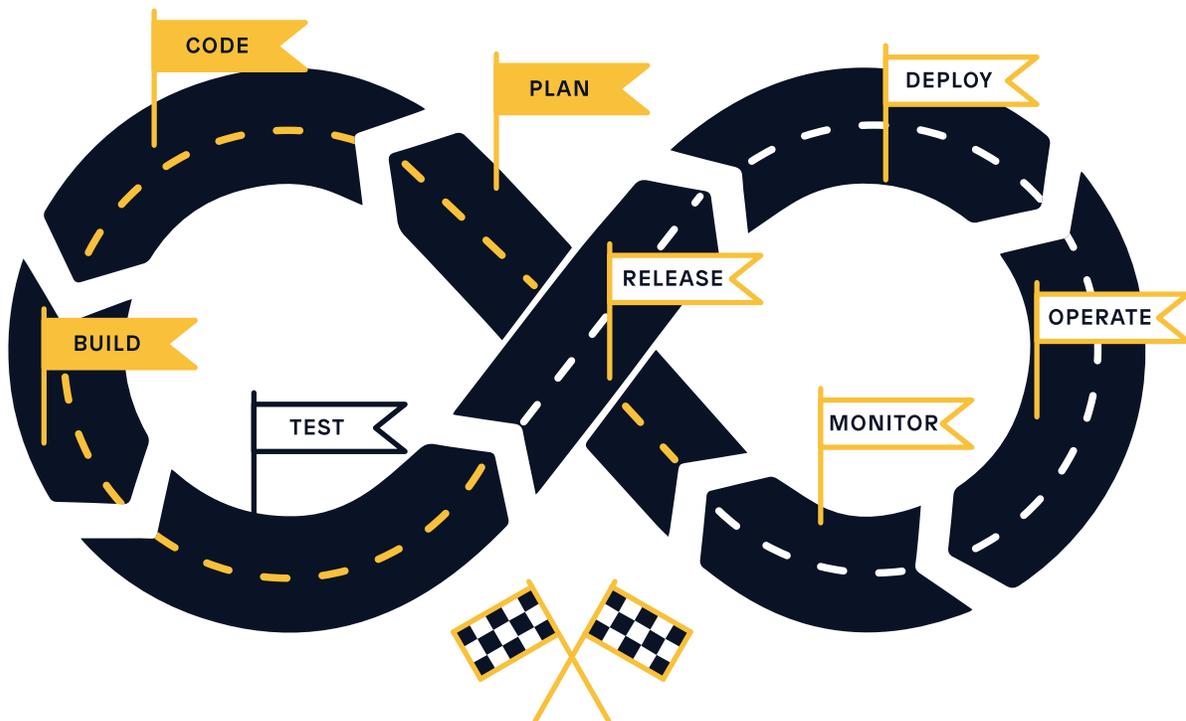
Part IV

Technical Governance: Continuing Control

Like Organizational Governance, this section of the framework is all about enablement as well; it's just oriented toward the technical needs of the initiative. And like the Organizational Governance section, Technical Governance is made up of multiple sub-sections, this time tackling technical specifics, policies, and best practices needed to ensure the goals of IT Governance are met for your initiative. Much of Technical Governance fits into the role of a “how-to” guide for executing on the initiative’s technical portions.

To structure the Technical Governance section, we find the continuous delivery/integration methodology provides a good foundation for thinking about technical delivery, even for non-software related initiatives such as Data Center Operations, or IT Support. Ultimately, use what fits your company and initiatives best. If you have a better, leaner model, for example, go with that.

As with Organizational Governance, if there is already an existing guide, how-to, or checklist in place then just link to it rather than trying to reinvent it. **The same rules apply: keep it simple, keep it focused, and update/remove anything that isn’t efficient enough or relevant.**



Plan

The planning section is focused on planning the next cycle of delivery. Whether it's a new software feature, delivering a laptop refresh, or deploying a new server rack, this is the stage that enables people to understand what they're trying to do, what they need to think about to execute, and what needs to be in place for them to be successful. Sub-sections here might involve delivery team structure, new services vs service updates, provisioning environments, and so on.

Plan – Examples

One example we see often at our customers to One example we see often with our customers. This allows their appropriate employees to understand the organizational specific steps necessary to get cloud resources for a new project, and reduces friction as the build, test, and release phases go on. Here's how this might look in a low-code governance model.

Environment Provisioning: All purchases of a container should be requested up front and made through the CoE for approval. The below documentation shows you how to acquire more licenses to enable users to access the application, purchase containers for the low-code application to run, and add users to an existing license.

Example Only		
	License purchase Process on how to acquire Mendix licenses.	View Doc →
	Container purchase Process on how to acquire Mendix containers.	View Doc →
	Adding Mendix Users Process on how to add users to an existing Mendix license.	View Doc →

The goal in these documents is to keep the steps to purchasing and acquiring to a minimum. Even our largest or most heavily regulated customers have fewer than ten steps to get necessary resources.

Build

This section contains the best practices and typical conventions for executing on delivering on the initiative. The content here should be focused on being guidelines and guardrails rather than strict commands. If they are required, commands should be kept to an absolute minimum.

The goal is to enable professionals to do their job, not tell them how to do their job. Typical sub-sections might include coding design pattern best-practices, laptop provisioning checklists, API best-practices, architecture guidelines, and so on.

Build – Examples

For example, when implementing Mendix, it'd be helpful to provide a set of Mendix developer best practices that we share with customers.



Sometimes the best documentation is the one that's already written for you. We've compiled developer best practices to get your developers quickly up-to-speed on how best to build solutions with Mendix. [Click here to see more.](#)

Another good example would be a list of preferences for how and when to use your platform. With a low-code platform your list may look like this:

- Prefer solving problems using the built-in Mendix capabilities rather than custom code or Java Actions whenever possible.
- Prefer using Mendix Marketplace modules supported by Mendix or with paid support over those with community or no support.
- Prefer breaking down problems into smaller applications or microservices rather than creating monoliths.
- Document the security roles in your app so it is clear who owns what.

Test

Testing functionality is critical to creating a stable technical foundation throughout your organization. The testing section focuses on how you define the different types of testing for your initiative, how and when the different types of testing should be applied before delivering a unit of execution, and any special tools or procedures required for testing. Sub-sections here might include testing definitions, security testing tools and process, load testing guidelines, server testing processes, and so on.

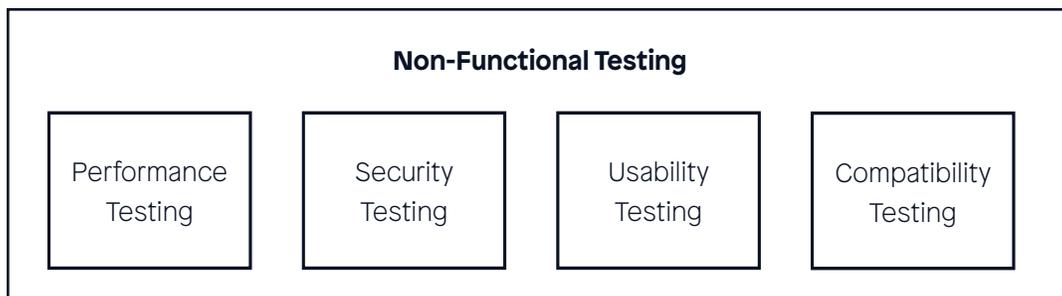
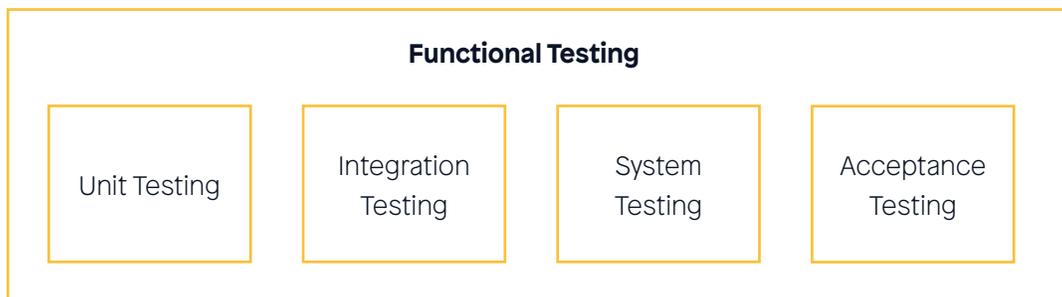
Test – Examples

With a low-code platform like Mendix, consider that a lot of testing is automated, and there are a lot of functions—like SSO for instance—that can be built and tested once. Then those functions work in perpetuity as a reusable component—at least, until that component needs to be updated.

Because there are many different types of testing, it's important to note what they are, which are mandatory or recommended, and which type has been automated.



With the [Mendix Application Test Suite](#), you get a suite of tools for embedding automated testing into your application lifecycle. Mendix also supports integration with third-party testing tools and supports Selenium and Junit testing frameworks.



Unit Testing: A Unit test is an automated piece of code that invokes a unit of work in the system and then checks a single assumption about the behavior of that unit of work. The developer performs this test.

Acceptance Testing: The product owner or mandated business representative tests and accepts the functionality.

Release/Deploy

This section is crucial. Without it, there's no clarity on how to deliver completed work. If delivery methods aren't clear, that means delays. And if your solution is delayed, so too is the business value. Subsections here include: production release processes, laptop delivery processes, cable run completion check list, dependency registration, and so on.

Release/Deploy – Examples

Low-code platforms help to accelerate time-to-market speed. To guarantee that business value is generated as quickly as it can be, it's good to put a production deployment checklist together and list it out in this section. This would ensure that you're deploying new software to production while adhering to your specific change management process. Here's what a deployment checklist may look like with Mendix.

Deploy of Production Checklist

- AQM Installed – AQM score must meet or exceed quality guidelines
- APM Installed – Mendix Admin can access for solution
- Application use Mendix License AD group for access
- GDPR Risk assessment has been completed
- Regression Test written by the Business Analyst or Tester are available



Mendix AQM offers a dashboard that gives instant insight into the quality of the application models you build with the Mendix Platform against the ISO 25010 industry standard for maintainability.

Operate/Monitor

This section focuses on how the deliverables of your initiative are operated, monitored, and supported once live. This can range from handling updates and patches coming from the solution's vendor to creating deprecation plans. This section greatly connects to the Organizational Governance portion of your framework, because it's also where you continuously monitor the business value you're creating.

Operate/Monitor – Example

The beauty of a low-code platform like Mendix is that because of reusable components, the timescale surrounding operation and monitoring processes are greatly reduced. The change management team that has to review each solution in the past doesn't have to use the same rigor they've used with traditionally built (.NET, Java, etc.) applications.



Deep monitoring or monitoring apps in a microservices architecture requires a tool that goes beyond standard Java monitoring. This is why we've integrated Data Dog with our technology so you can monitor enterprise-grade apps with ease.

Key to monitoring the value of a low-code-built application is making sure that the application is supported once it's deployed and in the field. IT or Support Teams have their existing governance already to provide ongoing support, so it may not be necessary to rewrite the rulebook. However, it's worth updating to consider new tools and processes to ensure that IT provides support for both Citizen and Professional developers. Create a technical support structure that is clear for users. A Helpdesk-Experienced Support-On-Call Developer model works well here, especially aligned with a ticket system.

In production support tickets will be submitted through the HelpDesk. The Frontline Support team will attempt to solve the issue. If needed, Frontline Support can escalate the ticket to the COE Admin for resolution or to the Unit's Mendix Support. If the issue cannot be resolved then, a Developer can act as a last resort to solve the issue.



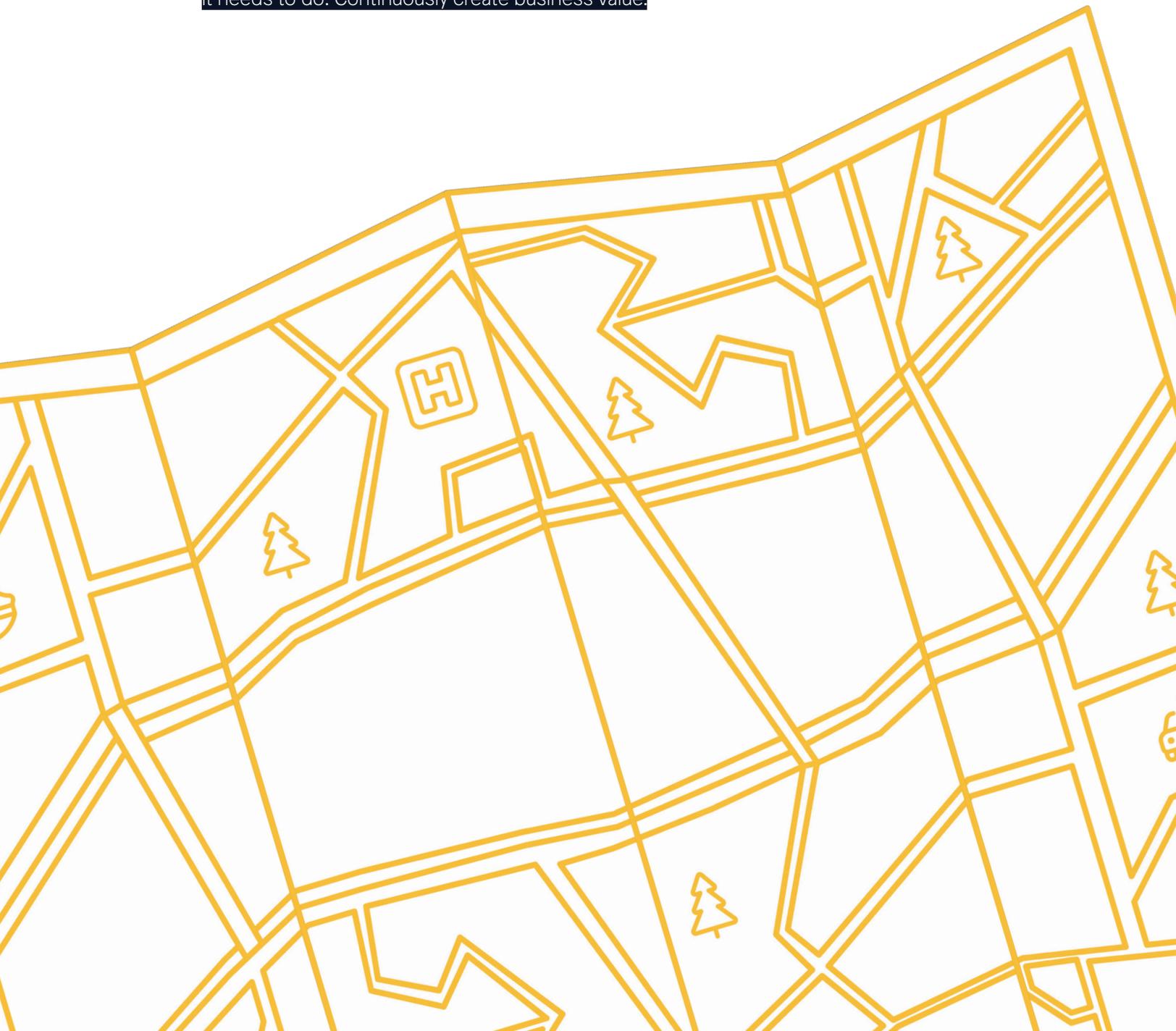
The Mendix Platform empowers you to maintain and observe your solutions with the out-of-the-box Mendix APM. APM provides tools that records all levels of logging, identifies and analyzes the performance of your applications and measures CPU and memory usage.

Governing the New IT Landscape

Gaining that competitive edge. Guiding your organization further into the digital age. Ensuring its future. None of this is possible if you don't have a governance framework that is in lock-step with your organization's digitalization strategy.

Using the framework we've provided, in conjunction with the Mendix Platform and its governance tooling, you get that holistic, 360-degree view that you need to rationalize and manage your application and data landscapes. This framework is your roadmap, but it's also the signpost for your organization that reads "Up ahead: Competitive advantage" and "Next turn: Change management" and "Next stop: Digitalization."

Most importantly, it's a means to empower your organization to do what it needs to do: Continuously create business value.





YOUR NEXT STOP



Security by Design w/ Low-Code Development



Learn more



Forrester named Mendix a leader in low-code with an

ELEGANT

collaborative platform*

See why

*The Forrester Wave™: Low-Code Development Platforms For Professional Developers, Q2 2021

WEBINAR ON-DEMAND

Business, meet IT. IT, business.

See how Accenture is bridging that IT/Business gap.



Register to watch

accenture



About Mendix

Mendix is the fastest and easiest platform to build and continuously improve mobile and web applications at scale. Recognized as a market leader by leading analysts, we help our customers digitally transform their organizations and industries by building, managing, and improving apps at unprecedented speed and scale.

More than 4,000 forward-thinking enterprises, including SUEZ, Philips and North Carolina State University use our platform to build business applications to delight their customers and improve operational efficiency.

GO MAKE IT.

- i. iso.org. 2021. ISO/IEC 38500:2015(en) - Information technology — Governance of IT for the organization. [online] Available at: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:38500:ed-2:v1:en>> [Accessed 6 August 2021].
- ii. 2018. COBIT 2019 Framework: Introduction and Methodology. Isaca.
- iii. Weill, P. and Ross, J., 2004. IT Governance: How Top Performers Manage IT Decision Rights for Superior Results. Boston, Massachusetts: Harvard Business Review Press.
- iv. Lindros, Kim. 2017. 'What is IT governance? A formal way to align IT & business strategy'. [online] Available at <https://www.cio.com/article/2438931/governanceit-governance-definition-and-solutions.html> [Accessed 23 July 2021].
- v. Gartner, 2020. Balancing Autonomy with Control: New Governance Models for Digital Business. Gartner.
- vi. Rowlands, B., De Haes, S. and Van Grembergen, W., 2015. Understanding the Dimensions of IT Governance Culture. International Journal of IT/Business Alignment and Governance, 6(2), pp.56-66.

