Eviden Handbook

Sustainable Software Development

Preface

In an era defined by rapid technological advancements and unprecedented global challenges, the importance of sustainability has emerged as a clarion call for humanity's future. It is a concept that transcends borders, industries, and ideologies, demanding our attention and collective action. It is within this context that we present this handbook: a guide to integrating sustainability into the art and craft of software engineering.

The path towards a sustainable future has never been clearer. The effects of climate change, resource depletion, and environmental degradation are undeniable, requiring us to rethink the way we operate in every aspect of our lives. And IT, with its pervasive influence on society and the economy, stands at the forefront of this transformation.

Our journey began with a simple realization: the world of digital services possesses an extraordinary potential to drive positive change. Through conscious choices and innovative solutions, we can not only reduce our own environmental footprint but also catalyze sustainability across various industries. The transformation of IT from being a part of the problem to becoming a vital part of the solution is the central theme of this handbook.

The authors of this book have embarked on this mission driven by a shared passion for sustainability and a belief in the transformative power of technology. They strived to create a comprehensive resource that is both practical and insightful, offering concrete steps and strategies to enable sustainable practices in IT.

I would like to extend my thanks to the authors, Joey Bannenberg and Abbas Shahim, whose dedication and expertise have been instrumental in bringing this handbook to life. Their commitment to the cause of sustainability has been an inspiration, and their contributions have enriched the content within these pages.

Now, dear reader, our journey takes a new turn. We invite you to not only read this handbook but to embrace its principles and put them into action. The solutions to our global challenges lie not only in awareness but in tangible changes in behavior and practice. Each one of us, regardless of our role or sphere of influence, has the capacity to contribute to a sustainable future. As you explore the chapters that follow, consider how you can integrate sustainability into your digital services endeavors and beyond.

It is our hope that this handbook serves as a catalyst for meaningful change, empowering you to play an active role in shaping a more sustainable world. The journey towards sustainability is ongoing, and it begins with you.

Let's embark on this transformative journey together.

Jeroen Heikens
Global Head of Portfolio, Application Services

EVIDEN

Table of Contents

1 Sustainable Software Engineering

1.1 Introduction

With the current rate of climate change influencing almost the entire world's population, it is nowadays seriously strived to reduce the pace of global warming. It is the consequence of an increase in the magnitude of the so-called greenhouse effect partially caused by the usage of fossil fuels to generate energy. They are a limited resource, and their exhaustion will lead to higher prices and more inequality in the world. Organizations are therefore discovering ways to optimize the usage of alternative renewable energy resources, such as wind power, and solar energy, that takes time and requires big investments.
A more immediate approach is to make sure that less energy is consumed for daily activities by simply executing them in a green efficient fashion. For example, a transportation company might make improvements to its operations within a warehouse or drive the optimal route to the customer to also reduce energy consumption and hence lower the emissions of CO2. As organizations are rapidly digitalizing and launching digital services to drive the new way of working, it is no longer sufficient to just develop software applications that functionally meet the requirements. The aim is to develop these business assets in the most sustainable way possible in accordance with an internationally recognized framework.
This document is a handbook that focuses on designing and developing software applications and describes the way in which this whole process adheres to a sustainable software engineering guideline.

1.2 Value

The handbook provides a concrete translation of the guideline to a set of easy-to-follow suggestions and instructions. It helps to understand how to build sustainable software and is useful to a variety of stakeholders including program/project managers, business/product owners, (lead) software engineers, scrum masters, developers, architects, hosting parties, and service providers.

1.3 Focal Point

There are a few ways to interpret and deal with sustainable software. Here, we pay close attention to how a more advanced development process can lead to make stakeholders (e.g. developers, users, architects) aware and trained about how to use digital services, to lower CO2 emission of these services, and to contribute to a more sustainable world. For example, more efficient code will lead to less data processing and therefore consume less energy to achieve the same target compared to a less efficient code. In general, the following statement holds:

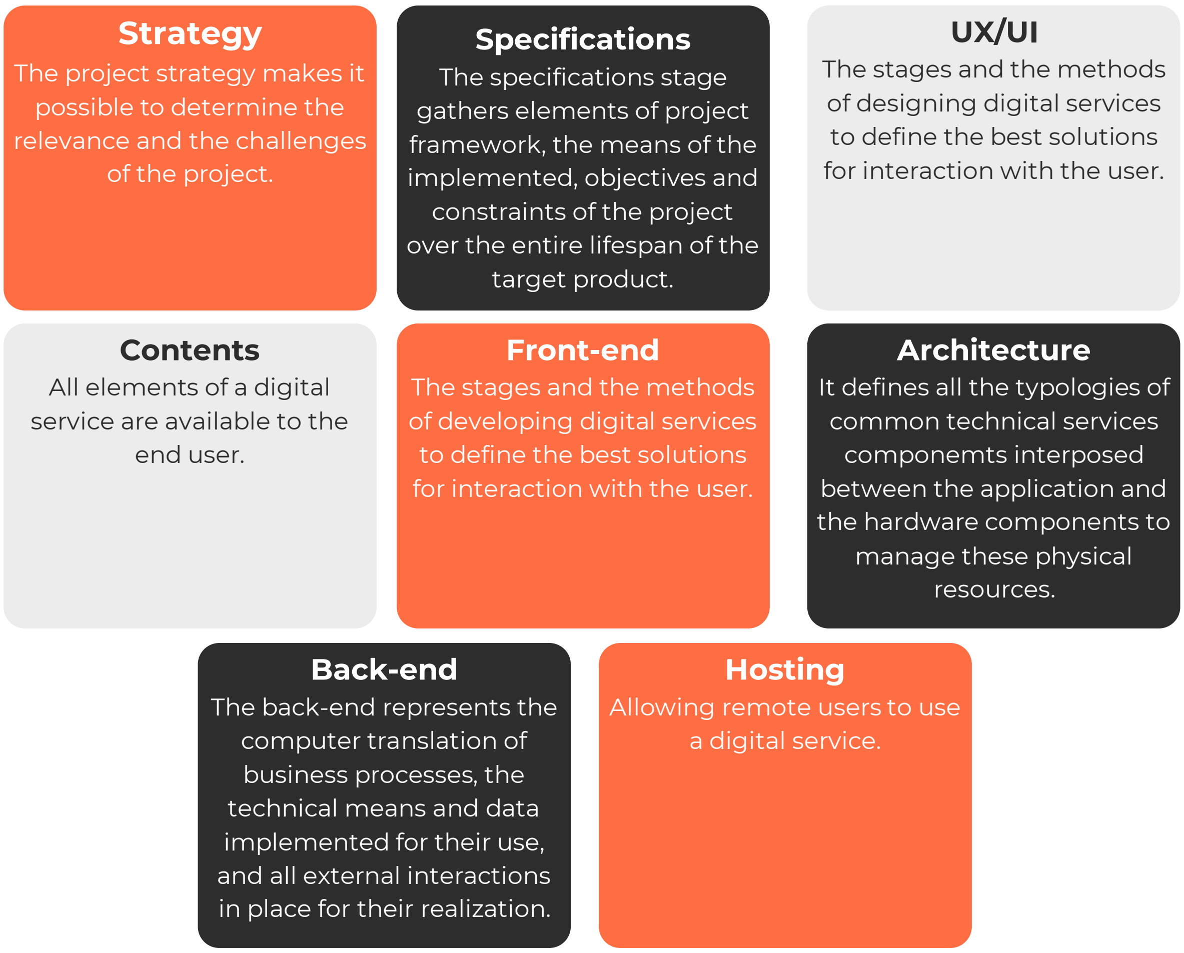
Higher level of software maturity minimizes the human impact, reduces the emission of CO2, and contributes to sustainable prosperity.

A higher level of software maturity can be achieved during all stages of development. The handbook concentrates on the following phases:

1. Feasibility study. This stage includes the business reasoning behind the application, the concreteness of the business need. Basically, you are asking the question of why the application is necessary for the business. Is the balance of advantages versus disadvantages heavily in favor of the positive side? And what consequences would it have regarding sustainability and green software?
2. Setting up agile teams and environments. This phase mainly consists of decisions regarding the team composition, approach, and creation of working environment. Ask yourself what way of working can have a positive effect on reducing energy consumption.
3. Requirements of the application. This stage describes the specifications of the application and what is required from a business and technical perspective. It is challenged whether these needs and wishes would contribute to the emergence of sustainable software.
4. Coding. During the development phase of the application, there are many important decisions to make, ranging from front-end to back-end, architecture, and User Interface/User Experience (UI/UX). It is ensured that the coding guidelines are in line with the sustainable software guideline.
5. Testing: This stage includes decisions of how and what to test. Important aspects to ascertain here are for example IT testing, user acceptance test, unit tests and system tests and the way in which these activities link to sustainability.
6. Installation: This phase includes the hosting party or service provider. It is essential to make sure that the related choices are also made in connection with sustainability.

1.4 Framework

The above mentioned are the common development stages with which most developers are familiarized. The framework for sustainable design of digital services launched by the Institute for Sustainable IT (Institute for Sustainable IT, 2021) is applied as the basis for crafting the handbook. This guideline comprises of eight different families illustrated in figure 1: Strategy, Specifications, UX/UI, Contents, Front-end, Architecture, Back-end and Hosting. They describe different areas of the development cycle where sustainable software engineering can be applied. The families each consist of several recommendations, which in themselves include a vast number of criteria. In total, there are 516 criteria at the time of writing each of which contributes to a greener, more efficient, and therefore more sustainable application. The more of these criteria are followed correctly, the more sustainable the application will be.



Strategy
The project strategy makes it possible to determine the relevance and the challenges of the project.

Specifications
The specifications stage gathers elements of project framework, the means of the implemented, objectives and constraints of the project over the entire lifespan of the target product.

UX/UI
The stages and the methods of designing digital services to define the best solutions for interaction with the user.

Contents
All elements of a digital service are available to the end user.

Front-end
The stages and the methods of developing digital services to define the best solutions for interaction with the user.

Architecture
It defines all the typologies of common technical services componemts interposed between the application and the hardware components to manage these physical resources.

Back-end
The back-end represents the computer translation of business processes, the technical means and data implemented for their use, and all external interactions in place for their realization.

Hosting
Allowing remote users to use a digital service.

Figure 1: The eight families for sustainable software

The families are individually expounded. Their essentials along with how to apply them in the development stages, will be explained in detail in the following chapters.

1.5 Applicable Criteria

The families of criteria, along with their own number of essential criteria, applicable to this handbook are listed in table 1. They can be associated with the development stages, each of which has its own group of audience. The handbook is thus initially all about the essentials of sustainable software engineering.

Table 1: Mapping of families of criteria to development stages

Family of Criteria	Development Stage	Audience	Number of Essential Criteria
Strategy	▸ Feasibility Study ▸ Setting up agile team and environment	▸ Program/Project Manager ▸ Business/Product Owner ▸ Enterprise Architects ▸ (Lead) Software Engineer ▸ Scrum Master	6
Specifications	▸ Setting up agile team and environment ▸ Requirements of the application	▸ Business/Product Owner ▸ (Lead) Software Engineer ▸ Scrum Master	5
UX/UI	▸ Designing ▸ Coding ▸ Testing	▸ (Lead) Software Engineer ▸ Architect ▸ Developer ▸ UX/UI designer ▸ Business/Product Owner	12
Contents	▸ Coding ▸ Testing	▸ (Lead) Software Engineer ▸ Architect ▸ Developer ▸ Business/Product Owner	4
Front-end	▸ Coding ▸ Testing	▸ (Lead) Software Engineer ▸ Architect ▸ Developer ▸ Business/Product Owner	5
Architecture	▸ Designing ▸ Coding ▸ Testing	▸ (Lead) Software Engineer ▸ Architect ▸ Developer ▸ Business/Product Owner	3
Back-end	▸ Coding ▸ Testing	▸ (Lead) Software Engineer ▸ Architect ▸ Developer ▸ Business/Product Owner	4
Hosting	▸ Installation	▸ (Lead) Software Engineer ▸ Architect ▸ Hosting Party ▸ Service Provider	6

The guideline structures the criterion in several aspects, such as a question that states the criterion, an indication of a test, a rule for assessing the level of compliance of the criterion. This structure is used as the starting point to develop this handbook.

1.6 Tool

The sustainable software guideline is logically applicable to several development tools and methods. In this handbook, a Mendix tool called Mendix Studio Pro versions 8 and 9 is used to demonstrate examples in different development stages. An additional Mendix tool is applied to present the agile way of working and developing, specifically Mendix Sprintr. The advantage of this tool is that it is connected to the development tool that is used by the programmers to build the application. When committing a change to the application, the developers can select epics and user stories where the commit is part of.

1.7 Reference Case

To be able to better explain the different criteria, examples are created based on the simplified reference case below.

Jack, a coach of a youth football team, has the responsibility to keep the parents informed of all scheduling and any applicable changes. In the past they would use email for most of this correspondence and there was a general schedule that was sent out in the beginning of the season. While this works to some degree, the schedule quickly became obsolete the moment there were any changes. Changes in the schedule caused lots of confusion and required frequent emails to keep everyone up to date.
What Jack wants is a responsive application for both web and mobile that allows him to display the schedule in an efficient manner, preferably in a calendar view. The schedule will consist of games and practices. Additionally, it would be nice to see a list of all the players on the team. For each game, he would like to get the number of confirmed players so that they know how many players they will have.
Regarding game stats, he would like to keep stats on whether they won or lost a particular game. For player stats, it is necessary to see how many goals they have scored. Because of the quick pace of the game, he will need an easy way to track goals during the game.
Finally, Jack seeks the ability to enter messages that can be displayed which would alert the parents of any changes or notices.

1.8 Layout

In the following chapters, chapters 2 to 9, the families with their criteria will be discussed separately. For every criterion, a detailed description of how to use it during the development of an application is given. The last chapter summarizes the most important aspects of the handbook.

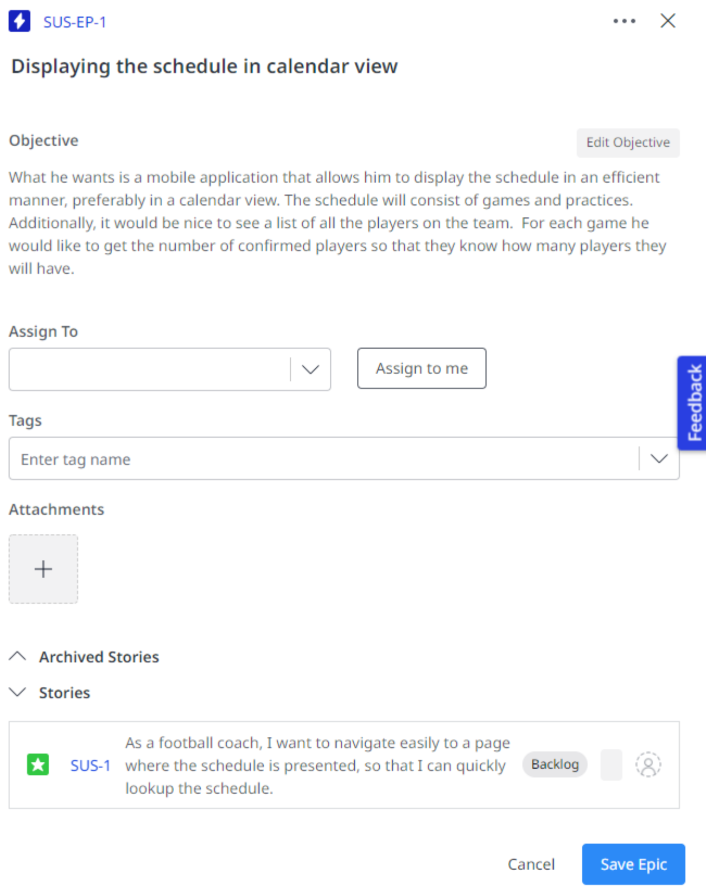
2 Strategy

Strategy is the first family to discuss here which is linked to the early stages of the development lifecycle as listed in table 1. The process of developing an application starts even before a single line of code is written. First, a business problem is stated that is to be resolved. The reference case described in the introduction chapter is presented as an example of such problem statement. Next, a feasibility study is performed to check if there is a business need for the problem to be solved. Basically, the relevance and challenges of the project are determined to decide whether it is worth it to find a solution for the problem statement, considering the sustainability criteria. Furthermore, the availability of human resources in as well as a development environment equipped for the IT sustainability area need to be organized and validated. This activity comes down to preparing a team that can interact with the Program/Project Manager and Business/Product Owner and is able to follow the guidelines of sustainable application development. There are six essential criteria in the Strategy family, each of which are explained below.

2.1 Is the business need expressed?

An essential part of framing the project is expressing the business need. The most important part here is that related uncertainties are taken out of the equation, as uncertainties could lead to extrapolated needs beyond the real expectations (Institute for Sustainable IT, 2021). The goal is to focus only on what is essential for the project, so that the resulting application only contains the features that are essential. Overextending the focus is harmful for environmental footprints due to a variety of reasons including unnecessary usage of scarce resources.
To fulfil this criterion successfully, the business needs are firstly to be formalized, validated, and shared. The best and most common way to do this is by using the above-mentioned tool (or any other commercially available tools such as Jira or ActiveCollab). To formalize the business needs, epics (i.e., clusters of business needs that have a broader strategic objective in common), features and product backlog items, also known as user stories, can be created, validated, and maintained. The program/project manager and business/product owner can fill the product backlog with extensively described business needs. The needs can then be validated by the stakeholders, and effort can be estimated, by the scrum master and the software engineer or developer. It is crucial to always include the latter, as it is the person who can give the best advice on the technical feasibility of the project. If required, the needs can be modified accordingly to make sure that none of them exceed the functional needs and technical limitations. In

addition, these tools have great options to showcase the needs on dashboards and to report and share the needs with all the parties involved.
A good test to assess the level of compliance of this criterion is to check if the business needs are assessed impartially in relation to uses (Institute for Sustainable IT, 2021). This adherence can for instance be achieved by engaging an external expert in IT sustainability. The task is to evaluate the definition and validation of the needs, their relevance, and challenges of the project to independently anticipate the impacts. The better the framing of the project is formalized, validated, and shared, the higher the sustainability score will be.
Given the reference case, attention is paid to the following expressed business need: Displaying the schedule in calendar view. Figure 2 illustrates an example of the way in which an epic can be formalized by the program/project manager and business/product owner applying Mendix tools (Mendix Technology BV, 2022). In the tool, a title, a description, potential tags, related attachments can be added to the epic. It is also possible to assign the responsible person to the epic. The requirements, called user stories, are always part of an epic and are discussed in the next chapter.


SUS-EP-1

Displaying the schedule in calendar view

Objective

Edit Objective

What he wants is a mobile application that allows him to display the schedule in an efficient manner, preferably in a calendar view. The schedule will consist of games and practices. Additionally, it would be nice to see a list of all the players on the team. For each game he would like to get the number of confirmed players so that they know how many players they will have.

Assign To

Assign to me

Tags

Enter tag name

Attachments

Archived Stories

Stories

SUS-1 As a football coach, I want to navigate easily to a page where the schedule is presented, so that I can quickly lookup the schedule. Backlog

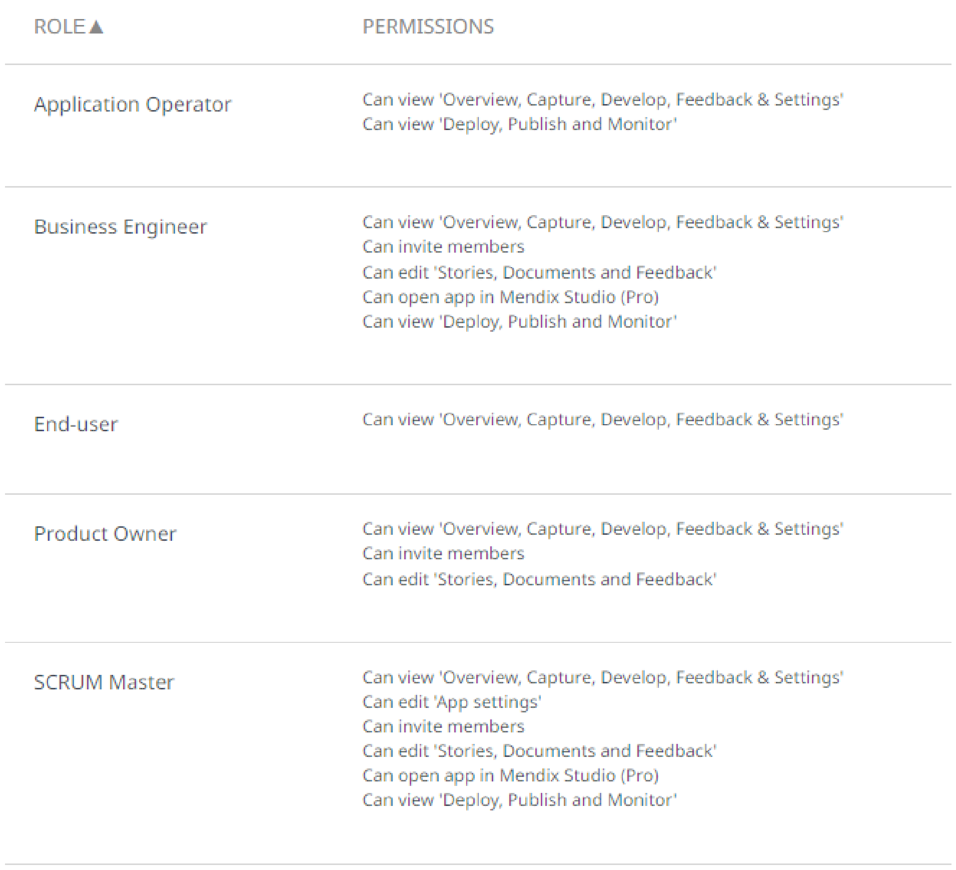
Cancel

Save Epic

Feedback

Figure 2: Example of an epic created within the Mendix Sprintr portal, including an example user story.

All the epics combined sum up to the business need in total. After creating all the epics, they are then validated, prioritized, and estimated accordingly by the above-mentioned roles and by the scrum master and software engineer or developer. They can do that by opening the epic in the tool and edit the fields they wish to alter. For example, it can be considered that a calendar view is harder to develop than a simple list, which will result in more usage of scarce resources. This and other validations are done to result in the final versions of the business need after which it is shared with the designated parties such as the related business units, the development team, the hosting party, and other external parties. Sharing is done by inviting email addresses to the application on the Mendix portal and assigning them read-only roles. Figure 3 below shows the default roles and their rights within the Mendix platform. The roles can be edited or removed if required.



ROLE ▲	PERMISSIONS
Application Operator	Can view 'Overview, Capture, Develop, Feedback & Settings' Can view 'Deploy, Publish and Monitor'
Business Engineer	Can view 'Overview, Capture, Develop, Feedback & Settings' Can invite members Can edit 'Stories, Documents and Feedback' Can open app in Mendix Studio (Pro) Can view 'Deploy, Publish and Monitor'
End-user	Can view 'Overview, Capture, Develop, Feedback & Settings'
Product Owner	Can view 'Overview, Capture, Develop, Feedback & Settings' Can invite members Can edit 'Stories, Documents and Feedback'
SCRUM Master	Can view 'Overview, Capture, Develop, Feedback & Settings' Can edit 'App settings' Can invite members Can edit 'Stories, Documents and Feedback' Can open app in Mendix Studio (Pro) Can view 'Deploy, Publish and Monitor'

Figure 3: Default role settings of project team members within the Mendix Sprintr portal

2.2 Do the sustainable digital services' goals subscribe to one of SDG (Sustainable Development Goals)?

When framing the project, stating the digital service goals of the project is essential to determine if a project is feasible. Given the focus of this handbook, the seventeen SDG (Sustainable Development Goals) as stated by the United Nations (United Nations, 2022) are listed below:

1. No poverty.
2. Zero hunger.
3. Good health and well-being.
4. Quality education.
5. Gender equality.
6. Clean water and sanitation.
7. Affordable and clean energy.
8. Decent work and economic growth.
9. Industry, innovation, and infrastructure.
10. Reduced inequalities.
11. Sustainable cities and communities.
12. Responsible consumption and production.
13. Climate action.
14. Life below water.
15. Life on land.
16. Peace, justice, and strong institutions.
17. Partnership for the goals.

To contribute to a more sustainable business and indirectly a more sustainable world, it is vital that the goals of the project are in line with at least one of the SDG and at least neutral with the other SDG. Otherwise, there is the risk that services are provided without real or with less sustainable value.
To do this, when considering a goal, the program/project manager and business/product owner should reflect on the consequences for each of the seventeen SDG and decide if it has a positive, neutral, or negative effect. The E-Handbook on Sustainable Development Goals Indicators (Paul Pacheco (UNSD), Ze Yar Min, 2021) can be used as reference to check whether a business need contributes to a goal or not. In such a case, the level of compliance of the criterion is adhered and a full score for sustainability is obtained.
Regarding the football coach reference case, one could argue that the to-be-developed application that improves the quality of life for people that do sports contributes to SDG 3. In addition, because it will reduce email traffic and double work, it will also use less energy (SDG 12 and 13). Finally, it is an innovation in the digital infrastructure for the amateur football club itself and the branch in general (SDG 9).

2.3 Are internal and external stakeholders aware of, or even trained / certified in Sustainable IT?

Another essential criterion is the level of IT sustainability awareness, training, and certification of the stakeholders of the project. Green IT: Information and Communication Technology for a Sustainable Future can be mentioned as a common example of such training that goes along with a certification. Every single one of these stakeholders fulfills a certain role within the lifecycle of development of the application and can consequently contribute to a more sustainable digital service. It is therefore of upmost importance that each stakeholder is at least aware of the sustainable IT and its importance. Even better would be to deal with stakeholders that are trained and certified to be able to apply the principles of sustainable software engineering.
In this case, a part of the strategy should be to allocate budget to increase the level of understanding of stakeholders regarding the concerns and challenges. Every stakeholder should be made aware of the concerns and challenges, which means that the Program/Project Manager and Business/Product Owner create a communication plan to broadcast these concerns and challenges. Regarding priority, it is recommended to start with the scrum master, the development team and related business owners.
The rule for assessing this criterion is basically the number of aware stakeholders relative to the total number of stakeholders. The more aware, trained and certified stakeholders there are, the higher the sustainability score is.
In the football coach reference case, it is important that not only the football coach is made aware of sustainable IT, but in this case also the software engineers, testers, and hosting parties. A part of the budget for this application needs to go to the awareness and training of these individuals. As amateur clubs do not necessarily have the highest budget, the recruitment of already aware individuals is prioritized so that less resources are spent on training.

2.4 Is sustainable IT compliance exposed to users?

Not only the individuals that are directly associated with working on the project should be made aware of sustainable IT, but so should the end-users. This criterion basically states that an application built on the criteria of sustainable software engineering needs to document this perspective and make the document available for the users of the application.
The common way to go ahead is to document all the decisions that are implemented throughout the development lifecycle right at the moment that they are made. It is accordingly easier to update the document which needs to be

accessible in the same way as other links such as for example a legal notice link or an accessibility statement.
To test this criterion, it basically means to check whether the Sustainable IT statement is created, validated and rather easily accessible to users. The existence of the document and accessibility of the document to the users are required to be able to fully comply to this criterion and get the highest sustainability score.
In a Mendix web application, external links to documents like this are mostly situated in the footer of the webpage. For a mobile application, this is an option as well, but it is also an option to create a main navigation item with a list of subitems that each contain an external link or a link to a page in the application that contains this information. Therefore, to test existence and accessibility, these are the locations to look at within an application.

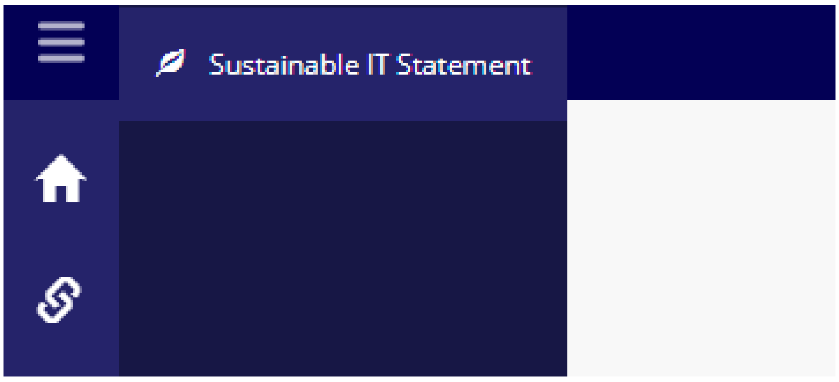
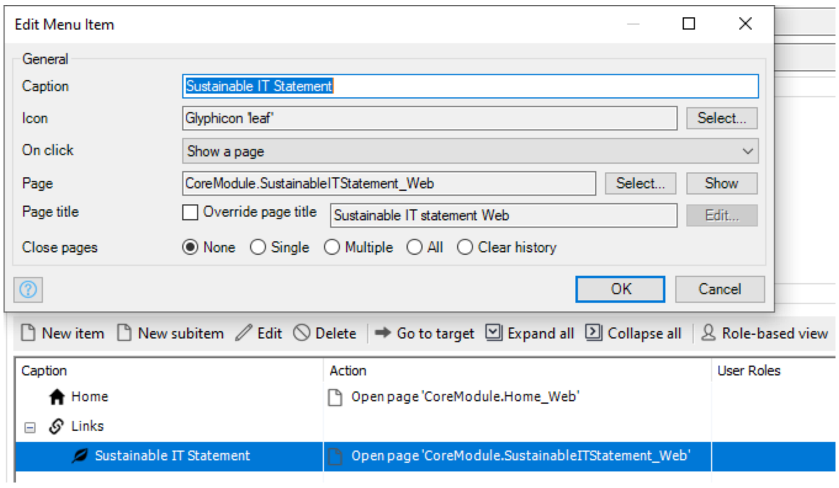


Figure 4: Link to the Sustainable IT Statement as seen by end-user.

In the reference case, links are shown as in figure 4 by using a menu item with subitem “Sustainable IT Statement”, where clicking on the subitem leads to the statement. Figure 5 shows how to add the link as a developer in Mendix Studio Pro.



Caption	Sustainable IT Statement
Icon	Glyphicon 'leaf'
On click	Show a page
Page	CoreModule.SustainableITStatement_Web
Page title	☐ Override page title — Sustainable IT statement Web
Close pages	[✓ None] Single Multiple All Clear history

Caption	Action	User Roles
Home	Open page 'CoreModule.Home_Web'	
Links		
Sustainable IT Statement	Open page 'CoreModule.SustainableITStatement_Web'	

Figure 5: Link to the Sustainable IT Statement as seen by developer.

2.5 Have you built an information system target by seeking consistency and pooling?

When considering the strategy for an application, it is possible to just focus on that application alone. However, instead of only measuring the impacts of that application, it is better to extend the focus to the whole information system (IS). This criterion states that the new project is part of an overall improvement of the IS, by targeting consistency in the way that projects are delivered and managed. Consistency leads to better plans, more accountability, better discipline, more sustainable projects, communicates the value of the IS and to better qualitative projects in general.
The enterprise architect can do this by creating an enterprise architecture cell in the company. In the enterprise architecture, he can use the urbanization paradigm for instance, which has been used for years by many, but is also called outdated by others (Bertin & Crespi, 2014). With regards to this criterion, the most important part of the selected paradigm is that is leads to consistency throughout the IS, so that the target projects are of better quality and more sustainable.
The test for this criterion is basically if there is an enterprise architecture cell in the company that includes this project. The number of projects targeted in that cell relative to the total number of projects is the rule for assessing the level of compliance. The more projects are targeted, the higher the level of compliance.
In Mendix it is common practice for companies to buy licenses for multiple applications as that is most of the time the cheaper option for the long term if there are plans for more than one application. If so, then it is best to share certain aspects of the application between each of the applications. It is very common that a company has a starter app template that contains all the elements of what each application of that company should have at minimum, for example a companywide user management system, a company styling theme or design system, and potentially a single sign-on solution. These apps, and single modules and widgets, can be shared via the company marketplace of the Mendix app store.

2.6 Have you defined and implemented within the DSI an IS debt and obsolescence management system?

This criterion focuses on the management of IS debt and obsolescence management that is decided within the information systems directorate (ISD or DSI). Firstly, functional, and technical debt are common aspects of the IS that need to be managed. There needs to be an accepted balance between adding new features from the functional debt and revisiting older functionality to refactor and lower the technical debt. Furthermore, it is important that obsolescence is anticipated to prevent that components are deprecated when versions are upgraded. Lastly, the variety of the components and their versions needs to be reduced so that it is consistent throughout the IS. Managing these points lead to better securing and controlling the IS.

To achieve the above-mentioned purpose, enterprise architects implement the Application Portfolio Management (APM) strategy within the enterprise architecture unit. The advantages of using APM can be found in appendix I (Mendix, 2022). *APM typically is about executing the following activities to improve the level of sustainable IT* (LeanIX, 2022)*:*

- *Documenting past, present, and future applications deployed or planned to be, inside an organization.*
- *Identifying and/or automating changes to application service lifecycles.*
- *Organizing applications according to business capabilities.*
- *Arranging IT components into technology stacks.*
- *Grading the technical and functional value of applications.*

The test for this criterion is whether an APM has been implemented within the enterprise architecture unit. If there is one, it needs to have defined how the technical and functional debt are handled, how obsolescence is managed, and how the number of components and their number of versions are reduced. Examples include managing the product backlog of each application regarding the amount and type of functional and technical debt, managing the validity of components to avoid them getting deprecated after upgrades, and by using a starter app template that already contains the to-be-used components and their versions. If this is the case, then the full score for this criterion is achieved.

Mendix describes three steps to best apply the APM strategy within low code. They are shortly described below:

1. General Fit Decision: the first step is to define which business initiatives are well-suited to low-code technology. Applications that fall under this category are that from a business perspective *"have a low time to market and, from technical perspective has apps that are data-driven, user-friendly, and multidevice"* (Mendix, 2022). Applications suitable for Mendix have a lower level of predefined requirements and a high expected rate of change.
2. Define Top Candidates: the second step is to define which of the list of step 1 are the top candidates that will bring business value to the company. To do this, five criteria are considered:
    a. The application needs to deliver a 'Wow' response from the stakeholders.
    b. It needs to have a clear business value.
    c. The first release must be an MVP that contains enough features to satisfy the end-users.

d. It should have high exposure to end-users and small to medium size complexity.
e. The application needs to be a t-shirt size S or M, regarding predefined categories such as which teams are involved, how many integrations are required, and how much MVP functionality is expected.
f. The more an application adheres to the criteria, the more of a top candidate it becomes for a first application to build within the IS.

3. Prioritize: the third and last step is to prioritize the top candidates using the criteria described in step 2. Then it is possible to make a chart, such as the example in figure 6. The graph consists of 4 sections on the complexity – exposure axes. Each candidate is placed in one of the four sections. Furthermore, each candidate also contains three attributes, which are revenue impact, wow factor and t-shirt size. In this example there are five candidates listed that all fall in the lower complexity half. Four of the candidates have high exposure, while one has less exposure. Based on the smaller t-shirt size but still a good revenue impact and a wow factor, the Guest Mobile App is placed above the other candidates, even though the Workorder Mobility App has a larger revenue and wow factor.

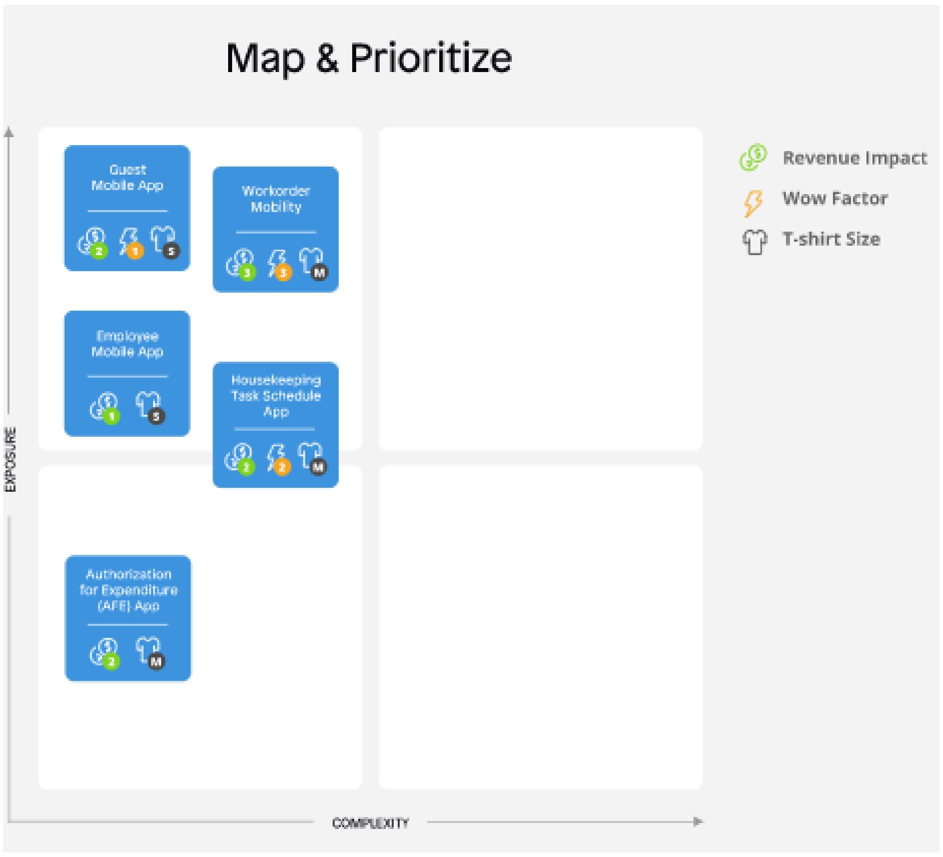


Figure 6: Example prioritization chart of top application candidates for a company.

3 Specifications

The second family to expound is the Specifications family. It is linked to the requirements stage of the development lifecycle as mentioned in table 1. The essentials of this family touch upon a further analysis of setting up the agile sustainable environment and team, including the external service providers. The way of working is determined within the team, and everybody should be aware of it. The scrum master takes the lead in this regard, and makes the decisions together with the product owner, and the software engineer or developer. Furthermore, regarding requirements, the epics are expanded with lists of user stories that are part of the epic. Each user story needs to have a sustainable IT component. Lastly, the technical documentation and specifically the end-of-life information of data and procedures with the documentation are discussed. It is noted that the family Specifications is not finalized at the beginning of the project. Due to the nature of the agile way of working, the execution of the essentials of this family is necessary throughout the lifecycle of the project every single time a new feature is added to the application. Five related essentials will be discussed in the following sections.

3.1 Have the sustainable IT challenges of the project been communicated to the team from the start?

The first of the criteria to go into detail about communicating the sustainable IT challenges of the project to the team. This criterion and the next criterion below are further continuations of the sustainable IT awareness discussed in chapter 2. Where the previous criterion focused more on the stakeholders in general, this criterion focuses more on the scrum team itself, which includes the scrum master, the product owner, and the development team including testers. The Sustainable IT Statement, containing the opportunities and challenges of the upcoming digital service encountered during the Strategy phase, are communicated towards the team.
Ideally, the individuals that were part of the Strategy process are the ones that take this responsibility. It is important that this is done from the start of the project, so that no time and resources are lost to correct not/less sustainable decisions made before the team was aware. The best way to achieve the sustainable IT approach, is to make sure that it is part of a systemic process of application development. It is not something extra to be done, but it is integrated within the way of working. A working solution for a problem is not good enough, only a sustainable solution is. This direction of how to work needs to be always followed or at least considered.

The test for this criterion and the next one is like the one described in the family Strategy. It is the answer to the question of how many members of the project team are aware of the Sustainable IT challenges of the project. Again, the higher the amount of aware team members, the higher the sustainability score.
Looking at the reference case from the introduction chapter, it is decided that for this relatively small application, two developers will be sufficient to work on it. Having only one developer would cause the developer to do everything alone, which results in not enough code review and would cost too much time. On the other side, having more than three could cause many conflicts when merging code for the relatively small application. The scrum master informs the developers with the challenges defined during the Strategy phase. One developer will focus more on the architecture, security, and back-end. The Sustainable IT challenges on these matters lie mostly in the fact that the application needs efficiency in the following aspects: managing the schedule of the matches, keeping up with the statistics during the game as well as providing a way to notify other users of updates. The solutions need to be robust, easy to expand or altered, well-documented, generic to avoid code duplication and not excessive in functionality. The second developer will focus on front-end, UX/UI and contents, where the challenges will contain among others: a responsive application that is both usable by web and mobile screens without having to create the same page twice, an easy to use UI for quickly updating statistics and notifying others, a theme that represents the culture and style of the customer without going overboard with too much content and lastly, an efficient way to show the match schedule that does not require too much computing power.

3.2 Does each group of the project team have sustainable IT skills?

To continue the preparation of the project team, this criterion focuses more on the separate groups of the project team regarding sustainable IT skills. Each group encounters different kind of challenges regarding sustainability and therefore requires specific knowledge. If only one group, for example the scrum master, is trained in sustainable IT, but the developers are not, it is highly probable that the development of the application will be inconsistent from a sustainable IT standpoint. It is therefore vital that each group has the knowledge required to tackle the challenges ahead.
One can solve this by creating a generic sustainable IT training plan, which all the groups need to follow. Next, to further zoom in on the specific tasks of the groups, a separate training plan for each group can be created. For example, the generic plan can consist of the challenges of the family Strategy and how to tackle the epics globally. Then, when looking at the requirements, the product owner

specifically looks at the business value with regard to sustainable IT and how to get the correct skills for that, while the software engineer is responsible to come with sustainable solutions regarding the architecture of the application. This way, the training of the groups will be tailor-made for each specific group.
The level of compliance of the criterion is adhered when there is a sustainable IT training plan for each group. Moreover, one needs to look at how well the training plans are specifically made for each group individually. A lower score is achieved if the groups are only aware of, but not trained in sustainable IT.
Considering the reference case, product owner Jack has planned a training about how to formalize the epics and user stories in the best possible way. In addition, he intends to follow a course about testing and how to efficiently do that. Finally, he got certified with the scrum certification PSPO I. The scrum master got certified with the latest scrum certifications PSM I and PSM II. The developers plan to follow Mendix courses about the best practices regarding performance optimization in their respective fields of front-end and back-end.

3.3 What proportion of the functionalities (user stories) have a Sustainable IT component?

The third essential criterion of the family Specifications concentrates on the requirements of the application. In an agile way of working, this is a continuous process that needs to be considered every time new functionalities are added to the application. Every feature and the way it is implemented has either a positive, neutral or negative impact on sustainable IT. When defining a functionality, also called product backlog item (PBI) or user story (US), it is important that a sustainability aspect is part of the user story template by default. Furthermore, additional user stories can be created with the prime reason to improve sustainability of the application.
The best way to define a user story or a defect is by using the same template for every story. This way each team member has to adhere to the same standards of the way of working. A good template should at least contain the following sections:

1. Title. The title of the user story should be in the following format: As a [user role], I want to [functionality], so that [benefit(s)].
2. Description. This describes the user story in a more extensive manner compared to the title.
3. Business value. This is the reason why the user story needs to be implemented.
4. Sustainable IT aspect. Describes the consequences regarding sustainable IT.
5. Acceptance criteria. The acceptance criteria section is the 'definition of done' of the user story. The story is done when all the acceptance criteria are

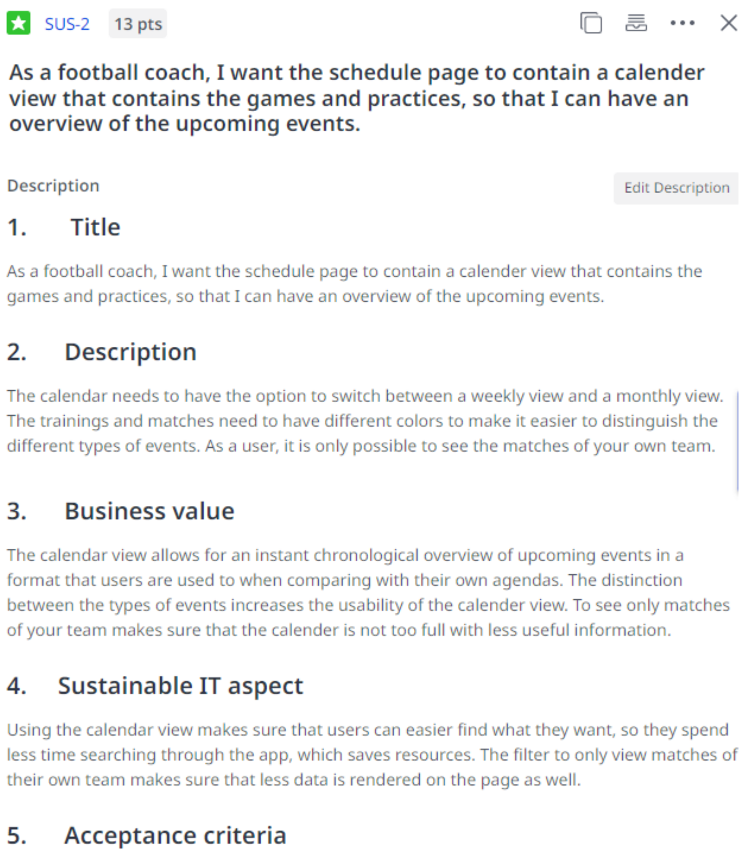
successfully tested. Also include in the acceptance criteria that the documentation needs to be updated, including the sustainable IT statement. Adding this to the acceptance criteria will help the project team to keep track of the documentation more consistently.

6. Suggested (technical) solution. The proposed solution of the user story. Here, it is important to not just go for the easiest solution to develop or deploy, but also consider the sustainable IT aspect of the solution.
7. Stakeholder(s). Mention the stakeholders of this particular story so it's easier to communicate between the parties.
8. Dependencies. Dependencies between user stories or on other parties could mean that the priority changes of the user story.
9. Other points of attention. This can be used to highlight potential roadblocks or difficulties and challenges of the user story.
10. Implemented solution. Always describe the final solution in the end. This way it is easier to know what to test as a tester.

Other important information that is included in the user story is the number of story points, of which sprint and release the story is part, who is responsible for the user story and to which epic the story belongs. Lastly, it is recommended to tag or label the user stories to give additional information. User stories that are created to improve sustainability should get the 'sustainable IT' tag.

The scrum master has the responsibility to test this criterion. The best way to achieve this is by incorporating fixed user story refinement sessions in each of the sprints. The refinement sessions can either be only together with the product owner(s), with the development team or with both parties. External parties can also be invited if necessary. During or before the refinement sessions, the user story template can be filled in after which the stories can be discussed, validated, and agreed on during the sessions. It is important to check if there is a sustainable IT aspect in the story. The more stories with a sustainable IT aspect, or even better, dedicated stories just about sustainable IT, the higher the compliance of this criterion will be.

For the reference case, an example user story is created by Jack using the Mendix Sprintr tool, which is partly shown in figure 7 below. The user story is following the template, which includes a sustainability aspect. During the refinement session with the scrum master and the development team, the details are further expounded and adjusted. One of the changes that is made is that using different colors for different types of events is not necessary, as it is already clear what kind of event it is when reading the event title. The business need for that feature was not outweighing the effort and required resources. In general, Jack and the rest of the scrum team try to include a sustainable IT aspect in 80% of the user stories. The other 20% exists of user stories that are technically required for the app to function properly, but do not have a sustainable IT aspect. Such a user story can for example be about being able to sign in into the application.


SUS-2 13 pts

As a football coach, I want the schedule page to contain a calender view that contains the games and practices, so that I can have an overview of the upcoming events.

Description

Edit Description

1. Title

As a football coach, I want the schedule page to contain a calender view that contains the games and practices, so that I can have an overview of the upcoming events.

2. Description

The calendar needs to have the option to switch between a weekly view and a monthly view. The trainings and matches need to have different colors to make it easier to distinguish the different types of events. As a user, it is only possible to see the matches of your own team.

3. Business value

The calendar view allows for an instant chronological overview of upcoming events in a format that users are used to when comparing with their own agendas. The distinction between the types of events increases the usability of the calender view. To see only matches of your team makes sure that the calender is not too full with less useful information.

4. Sustainable IT aspect

Using the calendar view makes sure that users can easier find what they want, so they spend less time searching through the app, which saves resources. The filter to only view matches of their own team makes sure that less data is rendered on the page as well.

5. Acceptance criteria

Figure 7: Example of a user story filled in using the template using Mendix Sprintr.

3.4 In the purchasing process (Call for tenders) are sustainable IT commitment requirements asked of external service providers?

This criterion of the family Specifications zooms in on external service providers. With very few exceptions, projects are normally carried out partially with the service of external parties. These external service providers have their own standards regarding sustainable IT. When considering which external service providers are best suited to fulfill the requirements as discussed in the last criterion described above. It is important that the selection of the providers is not only based on financial or technical arguments, but in equal measure based on sustainable IT arguments.
To do this, the project manager must consider if the potential external service provider has similar ideals and standards about sustainable IT as the internal service provider. One can look at similar services that the external party provided

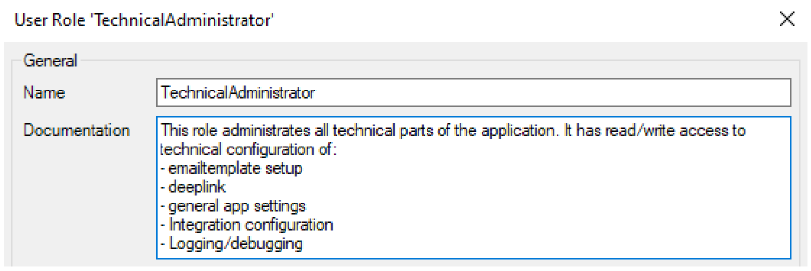
in the past to decide if it lives up to the expectation. If the expectations are met, discuss in detail what the requirements and their sustainable IT aspects are and how the external party plans to achieve the targets, especially regarding sustainable IT.
The rule for assessment of the level of compliance of the criterion is by comparing the number of requirements outsourced to external parties that comply with the standards of sustainable IT to the total amount of requirements outsourced to external parties. The more external parties meet the sustainable IT criteria, the higher the sustainable IT score is.
For the reference case, Jack decided that he needs email functionality to be able to notify users of the app with messages. For the functional email account, he would like to use an external service provider. He reads about Google and their goals of being carbon neutral since 2007 and carbon free by 2030 (Google, 2022) and decides to use a Gmail account.

3.5 Do all specified data and procedures have end-of-life information?

The last of the Specifications criteria focuses on the technical documentation and specifically the end-of-life information of data and procedures. When following the template of user stories in one of the criteria described above, one of the sections is the implemented solution. This is one place to document the technical implementations, including information about data and procedures. The second place is in the development tool itself, where it is possible to document all kinds of information directly next to the code. A few examples of these documents in Mendix are pages, microflows, nanoflows, snippets, etc. The third option to maintain the technical documentation is to create a separate file. With each option, it is important to include not just what technical features have been implemented, but also the end-of-life information. The reasons for the end-of-life can differ from security issues to regulatory or legal issues or from incompatibility to cost related decisions. For the reader of the documentation, it is vital to know when it‘s needed to act based on the end-of-life information.
The preferred way for the development team to create the technical documentation is a combination of the three places of documentation. The first place is useful when one looks from a requirements perspective and wants to know how the feature is technically implemented. The second place is particularly useful for developers, as they can read the documentation immediately next to the code or piece of logic. The separate file is useful for parties that do not have access to the (sprint) backlog or the development tool. Whatever the choice of place of the documentation, regarding the end-of-life information, it is good practice that the used solution includes the technical limitations of the solution,

how to adjust the solution if necessary and when the solution is deprecated and no longer supported. To make sure this is not forgotten, best practice is to add an acceptance criterion to the user story which states that the technical documentation is updated.
The developers test after each feature whether the implemented solution is well documented on the preferred place of documentation. If the documentation is up-to-date, and includes validated end-of-life information, then the level of compliance of this criterion is high. If the documentation is planned or identified but not performed yet, then a lower score is received.
In Mendix Studio Pro, it is possible to document all kind of different aspects of the application. A few examples:

- User roles within the main security section of the application and module roles within each module of the application. Figure 8 shows an example of such documentation.



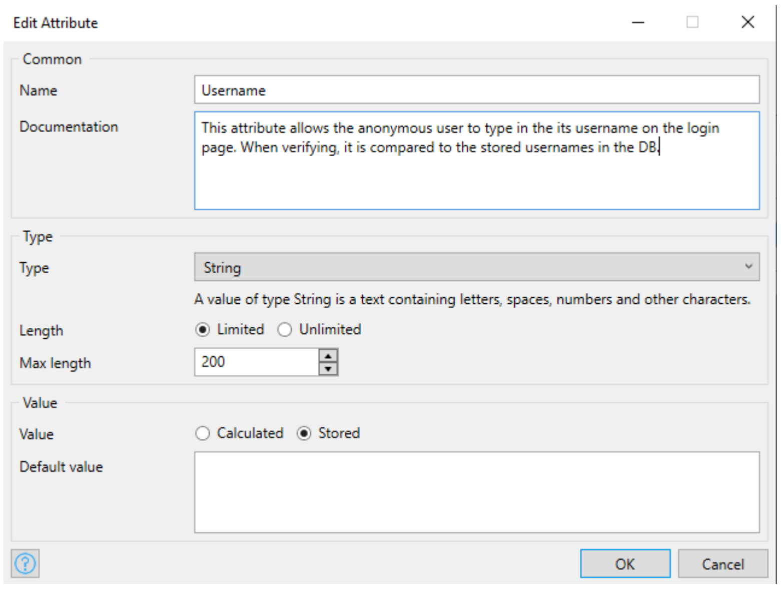
User Role 'TechnicalAdministrator'

General

Name	TechnicalAdministrator
Documentation	This role administrates all technical parts of the application. It has read/write access to technical configuration of: - emailtemplate setup - deeplink - general app settings - Integration configuration - Logging/debugging

Figure 8: Example of user role documentation.

- Entities within the domain model of a module, including attributes, associations, and entity access. Figure 9 shows an example of such documentation.



Edit Attribute

Common

Name	Username
Documentation	This attribute allows the anonymous user to type in the its username on the login page. When verifying, it is compared to the stored usernames in the DB.

Type

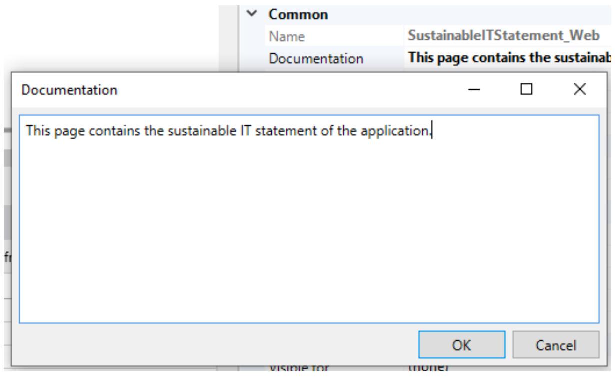
Type	String
	A value of type String is a text containing letters, spaces, numbers and other characters.
Length	[✓ Limited] Unlimited
Max length	200

Value

Value	Calculated [✓ Stored]
Default value	

Figure 9: Example of attribute documentation within an entity of the domain model.

- A page of the application. Figure 10 shows and example of such documentation.



Name: SustainableITStatement_Web
Documentation: This page contains the sustainab

Documentation

This page contains the sustainable IT statement of the application.

Figure 10: Example of a documentation of a page.

Other documents that can be documented are microflows, nanoflows, snippets, import/export mappings, JSON structures et cetera. Within some documents, it is also possible to create annotations to explain certain aspects of a document. An example is shown in figure 11. The annotation describes that the current login method will no longer be used and will reach end-of-life when the single sign-on method is introduced. For the reference case it is decided that the technical documentation will be done within the user stories template on the Mendix Sprintr portal and within the Mendix Studio Pro, such as in figure 10. The separate technical documentation file is not necessary, since it is a relatively small application. In addition, the people who need to know the technical documentation all have access to the Mendix portal or development tool.

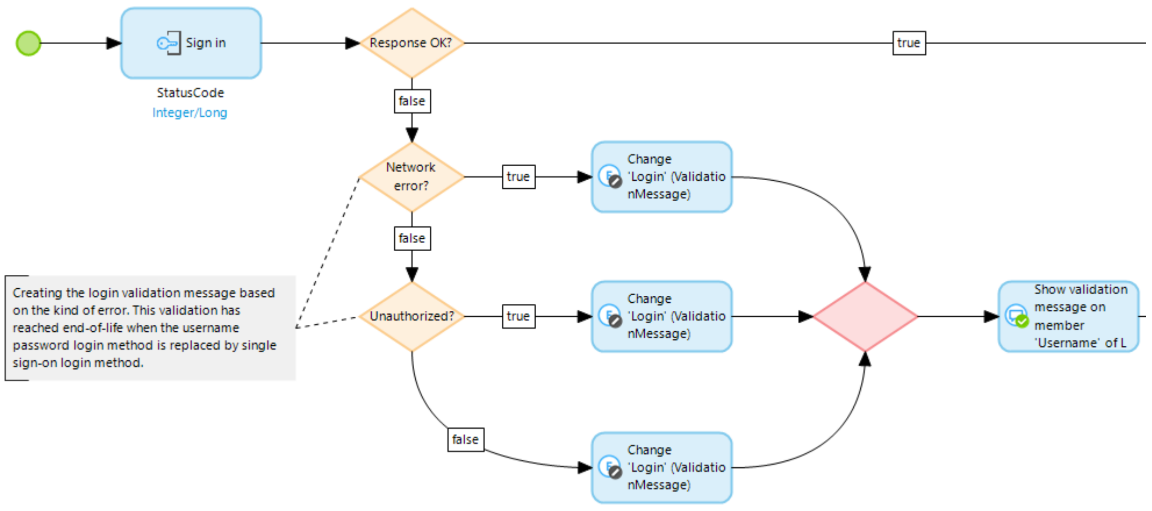

Sign in
StatusCode
Integer/Long
Response OK?
true
false
Network error?
true
Change 'Login' (ValidationMessage)
false
Unauthorized?
true
Change 'Login' (ValidationMessage)
false
Change 'Login' (ValidationMessage)
Show validation message on member 'Username' of L
Creating the login validation message based on the kind of error. This validation has reached end-of-life when the username password login method is replaced by single sign-on login method.

Figure 11: Example of an annotation to describe a section of a nanoflow.

4 UX/UI

The third family of criteria to dive into is the UX/UI family. This is the family with the greatest number of essential criteria that is focused on in this handbook. It is therefore a very important family for applying sustainable IT in software engineering. The UX/UI starts after or during the gathering of the requirements for the application. Before development starts, normally a brainstorming process takes place where mockups are designed by a UX/UI designer to get a generic idea of certain UX/UI aspects of the application, such as navigation through the application, the template of pages and the roles of the pages. One of the essential criteria focuses on making sure all stakeholders of the application are considered during the brainstorm phase. Besides the stakeholders, a very big aspect to consider is the planet, regarding environmental and ecological issues.
Another criterion is that the designs, and the human, ecological and economic values that they bring, need to be valued in the business model. Furthermore, it is again important that all parties involved in the UX/UI decisions and designs are aware and/or trained in sustainable IT. Next the focus shifts to compatibility regarding software versions and their hardware requirements. It is important to make the application accessible to as many users as possible, so the policy for compatibility between software and hardware needs to be established and meet the current accessibility standards. During development, decisions regarding UX/UI continuously need to be made by the project team. One of the decisions about UI needs to be about implementing visual soberness to reduce energy resources and material deterioration. Associated with this is also the criterion about compressing media and the criterion to limit the number and variation of fonts used within the application. Another criterion is that only essential features are implemented, and the interface is simplified. Regarding UX ethics, it is important to avoid dark patterns, and make sure that the users objectively remain the judge of their digital consumption. The twelve essentials will be further discussed in the following sections.

4.1 In the brainstorming process, are all stakeholders taken into account in their entire (human) dimension?

Even though most people immediately think about the planet and environment when thinking about sustainability, the social aspect of sustainability is just as important. During the brainstorming process of designing the application, it is vital to take all stakeholders into account. In this case, the stakeholders are mostly the end users, but also other people that are directly or indirectly influenced by the application. The application needs to be accessible to as many users as

possible, to be easy to use, to help users in their business needs and sustain that accessibility for as long as possible. Other stakeholders such as the developers and the (external) hosting party should not have a too complicated task to develop, manage, host, and maintain the application. Additionally, the application should be easy to showcase by the sales or marketing department. The application should suit all the stakeholders as much as possible until the end of life of the application. To achieve this goal, it is important to know which stakeholders are related to the project. A good way to do this is to create a stakeholder mapping and analysis. The Leaning for Sustainability (Learning for Sustainability - Will Allen (PhD), 2022) describes stakeholder analysis as follows:
Stakeholder analysis is a way to identify a project's key stakeholders, assess their interests and needs, and clarify how these may affect the project's viability. From this analysis, project managers can make plans for how these social and institutional aspects will be addressed. Stakeholder analysis also contributes to project design by identifying the goals and roles of different stakeholder groups, and by helping to formulate appropriate forms of engagement with these groups.
To prioritize the stakeholders and determine requirements and designs of the application, one can create a stakeholder mapping as in figure 12 below. In the figure, it becomes clear that stakeholders can be categorized in four groups depending on high or low influence and high or low importance. Each stakeholder of the project needs to categorize in one of the groups. Then, a strategy for the design and architecture can be relatively custom made to involve the stakeholders based on how prioritized they are. For application development, normally the end-users that will be using the application the most are the most important, but they might not have the most influence. It depends per application what priority is given to each different group of stakeholders.

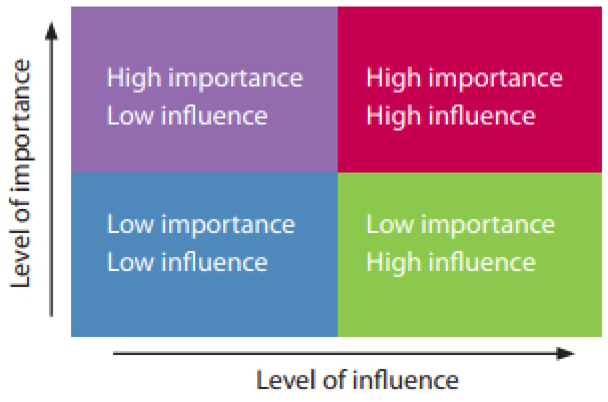


Figure 12: Stakeholder mapping (Learning for Sustainability - Allen, Will; Kilvington Margaret;, 2009)

The test for this criterion is passed when the stakeholder mapping is carried out and the prioritization of the stakeholders is done. If there is a strategy for each of the stakeholder groups and the validated designs and architecture are based on that strategy, then the full score for compliance with this criterion is achieved. Looking at the reference case, it is decided that the stakeholders with the highest priority are the parents of the players in the football teams. The reason for this is that these are the end-users that need the application the most to view the schedule of their children. The parents are also the ones who provide the amateur club with income, which causes them to have some influence regarding decisions within the club, especially the ones who are active as volunteers within the club. The designs of the application will therefore mostly be tailored for these users.

4.2 Is the planet taken into account in thebrain storming process, to integrate the ecological dimension (planet centric design)?

The criterion above focused on the human dimensions of sustainability regarding designs. This essential criterion focuses on the planet and the environmental and ecological issues of the brainstorming process. This might be one of most important criteria of all, as it directly touches the core of sustainability. Without the planet nothing is sustainable, so a big objective when designing the mockups of the application is to make sure that the impact of the application on the planet is as low as possible. At least it needs to be able to control the impact.
The best way to accomplish this target, is to use the planet centric design methodology (Planet Centric Design, 2022; Planet Centric Design, 2022; Planet Centric Design, 2022). This is a methodology for designing services and products without negatively harming the planet. The targets are to reduce resource usage, waste, emissions, and enabling people to have more sustainable lifestyles. There are several principles that need to be followed to be able to achieve these targets. The website of Planet Centric Design introduces a toolkit that thoroughly explains how all the principles can be adhered to. Looking at figure 13, the green circles represent the more common principles of application development. Each design of a feature needs to be financially viable, rather easily feasible and desirable. The planet centric design principles also include three other principles that focus more on sustainability:

- Responsible. Each decision that is made influences the planet. It is important to know what the consequences are of each decision so that responsibility of the decisions can be taken.
- Systemic. The networks of and between important stakeholders can be complex. It is vital that a system is created that connects the right people that can accelerate change and make a positive environmental impact.

- Transparent. It is essential that the development and the culminating features are transparent. Knowledge needs to be shared with end-users on by the designs, which accelerates behavior and thought change regarding sustainability.

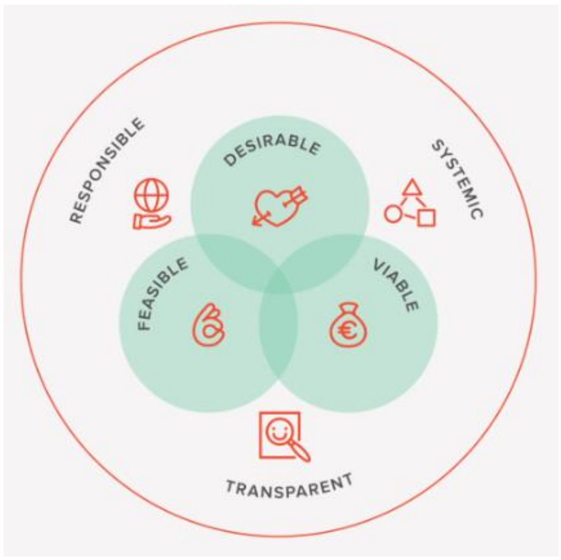
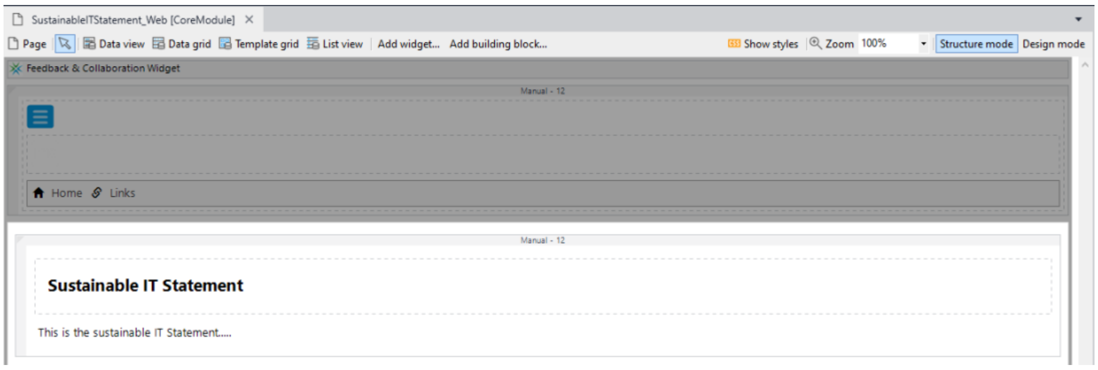


Figure 13: Planet centric design principles. (Planet Centric Design, 2022)

Whether the principles of the planet centric design are followed, and the toolkit is used is a good test to comply this criterion. Again, a better sustainable IT score is achieved when the designs are formalized according to the principles described above.

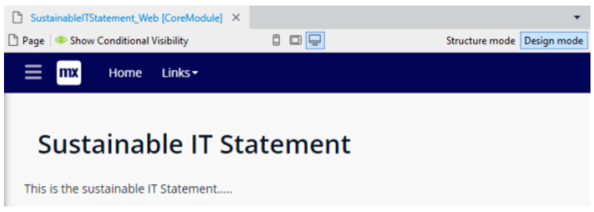


Sustainable IT Statement

This is the sustainable IT Statement.....

Figure 14: Sustainable IT statement page shown in structure mode.

For Mendix applications, mockups are normally created using external applications, for example in Figma. In addition, Mendix Studio Pro has a design mode during development, where it is possible to directly see how a page will turn out to be. This can save a lot of development time over the course of the application lifecycle. The difference between the structure mode and the mentioned design mode can be seen in figures 14 and 15.


SustainableITStatement_Web [CoreModule]
Page
Show Conditional Visibility
Structure mode
Design mode
Home
Links

Sustainable IT Statement

This is the sustainable IT Statement.....

Figure 15: Sustainable IT statement page shown in design mode.

Regarding the reference case, Jack requests from the UI designer that the designs be as basic as possible, as this will make sure most of the principles are achieved. Firstly, it is better financially viable, as Mendix already offers many templates of basic designs that can be used. Secondly, it is more feasible, as they are easier to develop. Thirdly, the players and the parents are not looking for a fancy application. They just have the desire to have the information they require. An application for them is already a great upgrade on the email contact they currently have. In addition, these designs are more responsible when considering the environmental aspects, as basic designs require less scarce resources. The transparency lies in the fact that these templates are publicly available from Mendix. Also, the sustainable IT statement will contribute to making all the decisions regarding sustainable IT transparent. Finally for the systemic principle, the focus of the designs will be on the end-users, as was discussed in the previous criterion. The young players and their parents are the people responsible for the future of the planet. So, the earlier they get in contact with sustainable IT and connect with each other about the subject, the higher the chance of a more sustainable planet in the future.

4.3 Is the extra-financial added value of sustainable digital technology valued in the business model?

The third criterion focuses on the extra-financial added value that is obtained on the long term when incorporating sustainability in the business model. One can imagine the non-financial advantages of applying sustainable IT on the long term, such as a more sustainable planet, a better environment for everyone, equality and so forth. It is sometimes more difficult to understand the financial advantages on the long term. However, there are several financial reasons to incorporate sustainable IT as well. Firstly, the fact that an organization is applying sustainability in its business model will increase the value of the product and attract more customers. Secondly, governments can provide grants to reward sustainability. Thirdly, constant arbitrations on the importance and integration of sustainability in future projects can be avoided if it is covered in the long-term business model. To incorporate this in the business model, it is important to make sure that the human, ecological and economic values are included in the business model

canvas. Then, the next step is that these values are approved by all stakeholders to cover the continuity of the values within the business.
The check to test this criterion is if the above-mentioned values are included in the business model canvas and it there is a clear profit analysis of the sustainable IT contribution. If this is formalized and approved by all stakeholders, the maximum score for this criterion is achieved.
For the reference case, Jack makes sure that the business model included the following financial values regarding sustainable IT. The transparency of the applied sustainable IT in the application inspires the young footballers and their parents to involve themselves in sustainable activities within the club. This means that the club does not have to pay for these activities themselves. Examples of these activities are that players and parents help clean up the club after match days and other events or that team outfits are made more durable and therefore cheaper on the long term. The local municipality is also approached, and they decided to grant the club a new pitch based on their example for the community regarding sustainability. Lastly, more footballers subscribe to the club, just since sustainability is taken in high regard by the parents.

4.4 Are stakeholders informed about Sustainable IT?

This criterion has come up before in the other families described above. To reiterate, it is vital that the stakeholders are informed, aware and trained in sustainable IT. This will improve the level of compliance of sustainable IT within the application and the organization is general. Naturally, this also counts for UX/UI designers. The work they do is what the end-user has mostly contact with by navigation through the application.
In this case, a part of the planning and budget should be allocated to increase the level of understanding of UX/UI stakeholders regarding the concerns and challenges of sustainable IT.
The test is similar for the other criteria. It is basically the number of aware UX/UI stakeholders relative to the total number of UX/UI stakeholders. The more aware, trained, and certified stakeholders there are, the higher the sustainability score is.
In the football coach reference case, it is important that the UI/UX designer that is responsible for that part of the development is made aware and trained. As amateur clubs do not necessarily have the highest budget, a designer is recruited that is already trained of sustainable IT to make sure less resources are spent on training.

4.5 Is a policy for compatibility with obsolete terminals and software versions established?

Due to constant and accelerating technological advances ~~nowadays~~, software is updated on a frequent basis and rapidly deployed. Sometimes, updating the software version can lead to compatibility issues between software and hardware. This criterion and the next concentrate on the importance of accessibility of the application to as many users as possible. Compatibility issues decrease accessibility, as it reduces the number of devices that the application can run on. Each time new hardware is required to be able to run the application, it means that resources are spent that can be used otherwise.

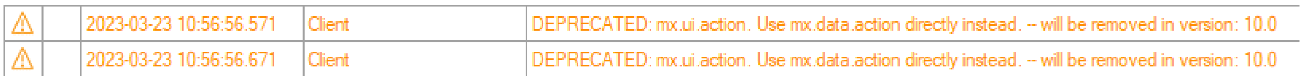
To ensure that this is controlled as much as possible, a policy for compatibility needs to be established that meets the current accessibility standards. The international standards are stated by the Web Accessibility Initiative (WAI) on their website, including the World Wide Web Consortium (W3C) (WAI, 2022). In addition, it should contain the application specific requirements regarding accessibility. A good way to establish the policy is to include end-of-life information and requirements of features in the technical documentation. When developing the (technical) solution, keep the compatibility of the features in mind.

The test belonging to this criterion is to measure how many features are described in the technical documentation and have an end-of-life documentation. Naturally, the longer the life of a feature and the more it is documented, then higher the sustainable IT score will be.

Mendix provides valuable feedback regarding end-of-life information of widgets. Widgets are defined by Mendix as follows (Mendix Technology BV, 2022):

"The widgets available in the Mendix Marketplace are single user-interface elements that can be configured, such as containers, drop-down menus, and buttons."

When a widget is deprecated and can no longer be used in a higher version number of Mendix Studio Pro, the console will display this when running the application, as shown in figure 16.



```
⚠  2023-03-23 10:56:56.571  Client  DEPRECATED: mx.ui.action. Use mx.data.action directly instead. -- will be removed in version: 10.0
⚠  2023-03-23 10:56:56.671  Client  DEPRECATED: mx.ui.action. Use mx.data.action directly instead. -- will be removed in version: 10.0
```

Figure 16: Example of a UI element deprecation message in the Mendix Studio Pro console.

For the reference case, Jack decides that the application will be created in Mendix 9 and he does not expect to upgrade the application to Mendix 10 soon. However, just to be sure, it is decided that widgets can only be used if they are not deprecated in Mendix 10. Furthermore, widgets and modules can only be downloaded from the Mendix App Store if they are regularly updated by the creator. This is to prevent that when upgrading the football calendar app, the downloaded features are no longer up to date. For all the features, the end-of-life information is added to the technical documentation.

4.6 From the design stage, does the service meet the current accessibility standards in (at a minimum)?

This criterion continues the accessibility aspect of sustainability. It states that, at a minimum, the application needs to meet the accessibility standards as described by the WAI (WAI, 2022). The minimum in this case is to adhere strictly to the legal aspects of accessibility. One can go a step further by anticipating future points of and changes to regulations and standards. This way, there is less need for sudden updates or changes to the application, which causes accessibility to be better covered throughout the lifecycle of the application.
Besides the WAI, there are other tools to comply to the accessibility standards, such as the Inclusive Design Toolkit by the Engineering Design Center of the University of Cambridge (University of Cambridge, 2017). They state that every design can either potentially include or exclude users, where inclusive design focuses on making informed design decisions based on user diversity, with the goal to include as many users as possible of the appropriate target market without sacrificing too much of the other business goals of the digital service. This can best be explained by the pyramid models of figures 17 and 18.

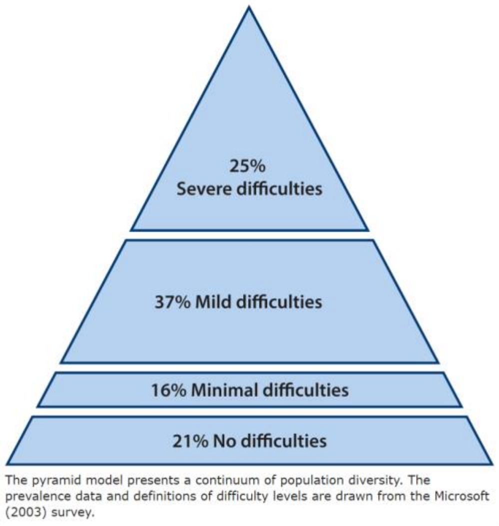

25%
Severe difficulties
37% Mild difficulties
16% Minimal difficulties
21% No difficulties
The pyramid model presents a continuum of population diversity. The prevalence data and definitions of difficulty levels are drawn from the Microsoft (2003) survey.

Figure 17: Pyramid model of application user diversity (University of Cambridge, 2017).

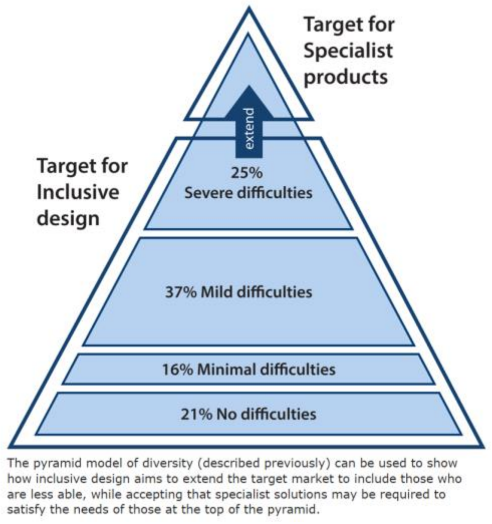
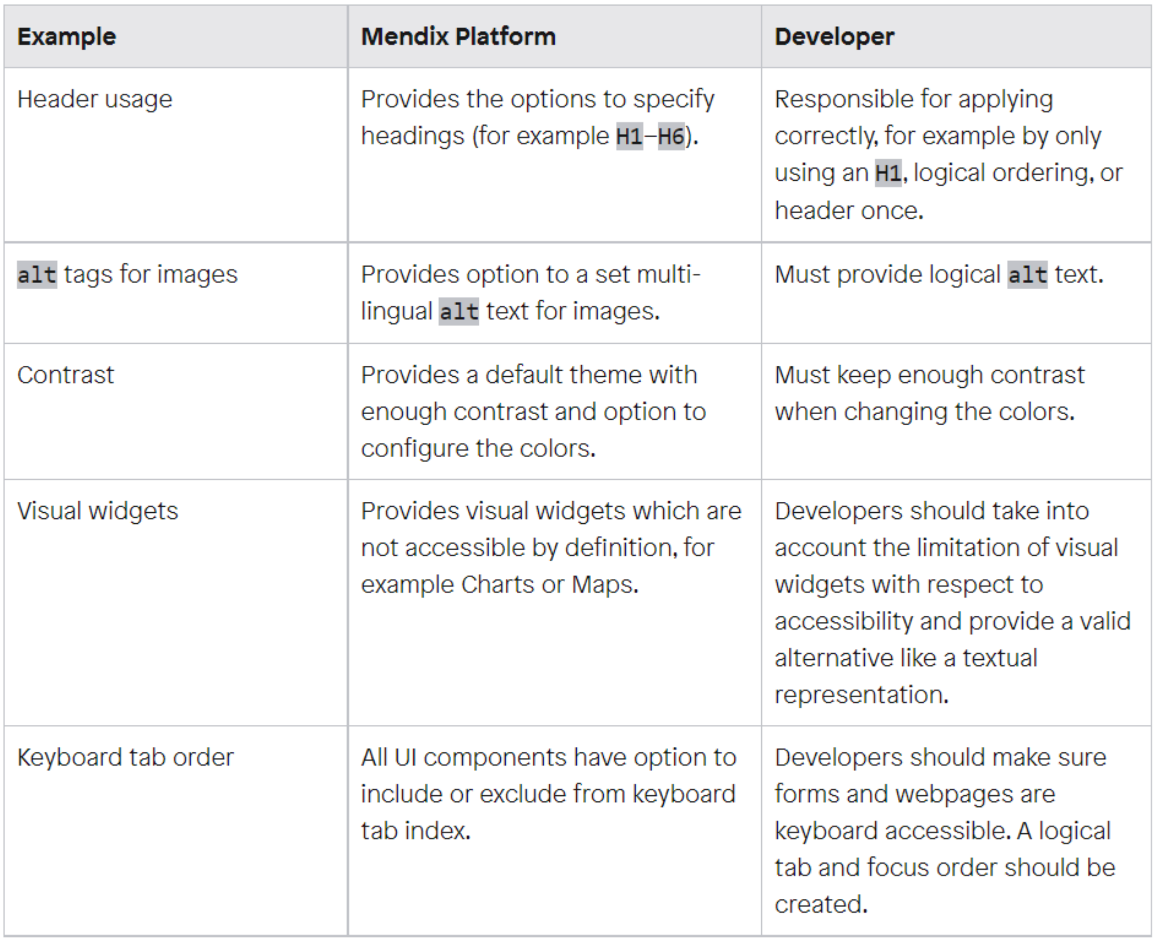

Target for Specialist products
extend
Target for Inclusive design
25%
Severe difficulties
37% Mild difficulties
16% Minimal difficulties
21% No difficulties
The pyramid model of diversity (described previously) can be used to show how inclusive design aims to extend the target market to include those who are less able, while accepting that specialist solutions may be required to satisfy the needs of those at the top of the pyramid.

Figure 18: Pyramid model of inclusive design (University of Cambridge, 2017).

Figure 17 displays the pyramid model of the diversity of application users. The bottom layer represents the users that have no difficulties in accessing the application, while the users further up in the pyramid have more and more difficulties. The top of the pyramid represents the users that need most attention when considering designs. According to figure 18, inclusive design targets on all the users except for the top of the pyramid. This is called the target audience, and the goal is to optimize the product performance indicators for that target. It is then accepted that the top group requires specialist products.
To test the level of compliance of this criterion, it is vital to know what the intended level of accessibility is. When following the inclusive design toolkit, one must be aware of what the target audience is and how well it is able to work with the application. If the regulation and legal concerns are formalized and shared, as well

as the level of accessibility that is measured is close to what is intended, then the score for this criterion will be higher.
The Mendix portal offers various documents and videos about applying the accessibility standards within applications (Mendix Technology BV, 2022). These should be followed thoroughly to be able to offer the best application on this aspect of sustainable IT. Jack therefore decides that the selection of the UX/UI designer(s) needs to be based on awareness of the standards. A few of these standards are shown in figure 19, including how the Mendix platform provides options itself as well how the developer should apply them.



Example	Mendix Platform	Developer
Header usage	Provides the options to specify headings (for example `H1`–`H6`).	Responsible for applying correctly, for example by only using an `H1`, logical ordering, or header once.
`alt` tags for images	Provides option to a set multi-lingual `alt` text for images.	Must provide logical `alt` text.
Contrast	Provides a default theme with enough contrast and option to configure the colors.	Must keep enough contrast when changing the colors.
Visual widgets	Provides visual widgets which are not accessible by definition, for example Charts or Maps.	Developers should take into account the limitation of visual widgets with respect to accessibility and provide a valid alternative like a textual representation.
Keyboard tab order	All UI components have option to include or exclude from keyboard tab index.	Developers should make sure forms and webpages are keyboard accessible. A logical tab and focus order should be created.

Figure 19: A few examples of accessibility standards and how the Mendix platform and developers can apply them.

4.7 Is visual soberness implemented to reduce energy resources use and the impact on the material deterioration of the visual, sound, and tactile interface's components?

During the development phase of the application lifecycle, decisions regarding UX/UI continuously need to be made. This criterion concentrates on one of these decisions, namely about implementing visual soberness in the application. Firstly, this reduces the usage of scarce energy resources, as it requires less energy to show sober elements or components than elements that are implemented on a purely aesthetic or marketing approach. Secondly, this also leads to a slower material deterioration of the hardware. Thirdly, this is better for the ergonomics of the user.
A good way to do this, is to include these principles already in the graphic design policy and system, by including it in the logo, the brand's colors, and the typography of the brand. Make sure that the graphic charter is not solely based on

aesthetics but also that it is eco-compatible and respects the accessibility standards. In addition, it is good practice to make a dark mode available for the application.
The level of compliance of this criterion increases the more the elements, components and overall graphic charter are more eco-compatible and is more in line with the accessibility standards.
The advantage of low-code and Mendix specifically is that it provides all kinds of building blocks and basic themes from the default Atlas UI. This allows for developers to adhere to this criterion more easily. Regarding the reference case, it is decided that the basic Atlas UI module is used to make sure that the UI has a certain amount of visual soberness.

4.8 Are the features used by the user?

This criterion is a rather easy one to understand, but that does not make it any less important. According to a study of the Standish Group about modernization, 50% of features are hardly ever used, as seen in figure 20 (Standish Group International, Inc., 2010).

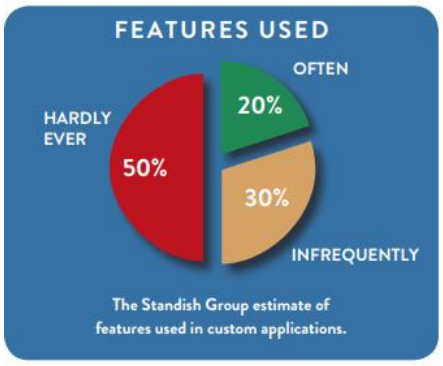


Figure 20: Percentiles of features used in custom applications (Standish Group International, Inc., 2010).

Each feature in an application contributes to a higher consumption of scarce resources and a larger environmental footprint. In addition, it also leads to an increase in the volume of digital services. Therefore, it is most vital to reduce the number of features by only introducing features that will be often used.
To achieve this, it is good to check if a potential feature is expected to be used by the main users and if so, that it brings real value to the application. As discussed earlier in this handbook, it is best practice to define user stories of features in a tool and to use a user story template to describe the story in detail. User feedback can be gathered either automatically, for example by simulating user journeys and analyzing production data, or manually by holding feedback sessions and sending questionnaires to the users. Then based on the results, it could be decided that a feature is no longer needed later, because it is not used anymore.

The tests for this criterion are based on whether user journeys have been simulated and validated to be able to confirm if features are used. The level of compliance is assessed by measuring the number of often used features against the total number of features. The better that ratio, the higher the level of compliance for this criterion.
Each application deployed in the Mendix cloud can use the standard built-in metrics to measure the health of the application. On the metrics page of the application page within the Mendix Sprintr, several graphs are shown with figure 21 as an example of such a graph.

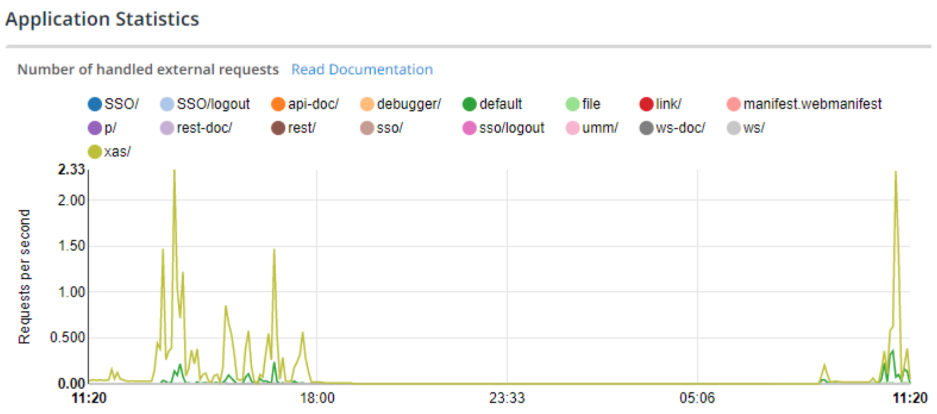
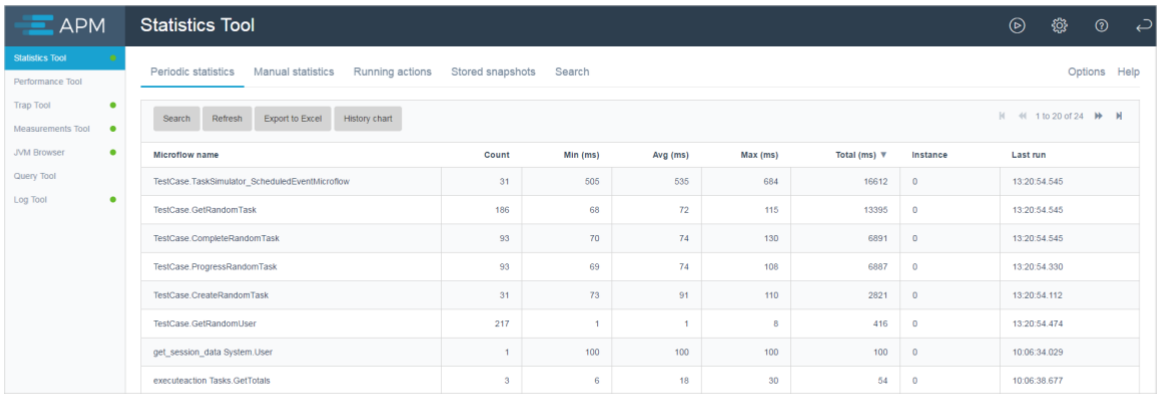


Figure 21: Mendix application statistics: number of handled external requests.

Furthermore, Mendix allows for several add-ons that can be used to improve the overall quality of the application, such as ATS (Application Test Suite), APD (Application Performance Diagnostics), ACR (Application Code Reviewer) and AQM (Application Quality Monitor) (Mendix Technology BV, 2022). For this criterion the APD is relevant, as it has the option to give insights to specific actions within the application, such as how many times microflows are run and how long it takes to run them. See figure 22 for examples.



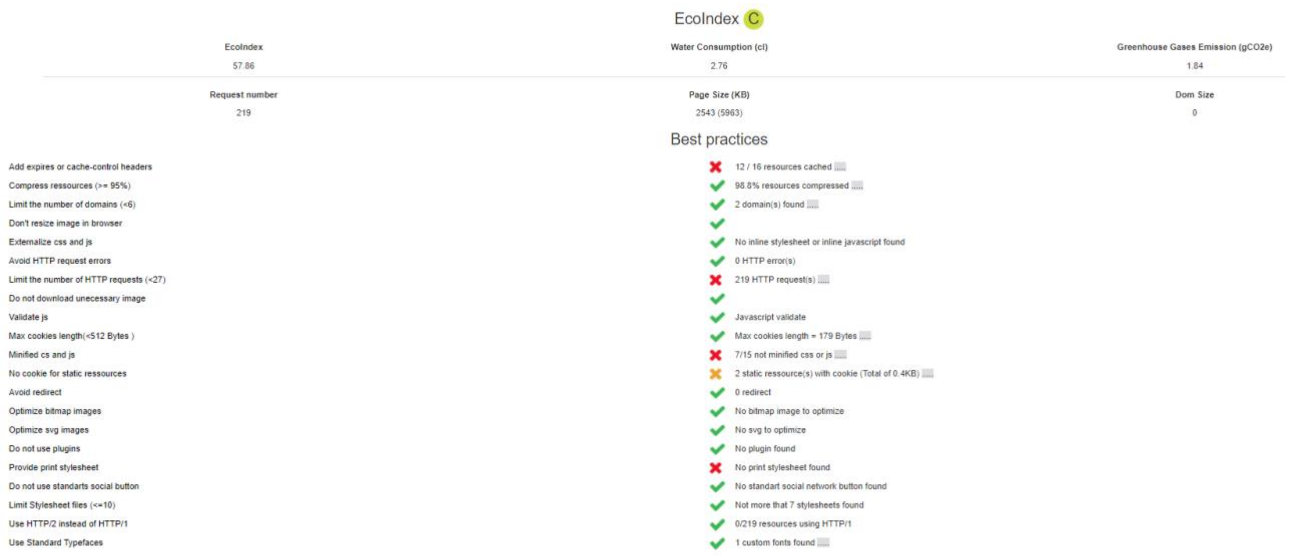
Microflow name	Count	Min (ms)	Avg (ms)	Max (ms)	Total (ms) ▼	Instance	Last run
TestCase.TaskSimulator_ScheduledEventMicroflow	31	505	535	684	16612	0	13:20:54.545
TestCase.GetRandomTask	186	68	72	115	13395	0	13:20:54.545
TestCase.CompleteRandomTask	93	70	74	130	6891	0	13:20:54.545
TestCase.ProgressRandomTask	93	69	74	108	6887	0	13:20:54.330
TestCase.CreateRandomTask	31	73	91	110	2821	0	13:20:54.112
TestCase.GetRandomUser	217	1	1	8	416	0	13:20:54.474
get_session_data System.User	1	100	100	100	100	0	10:06:34.029
executeaction Tasks.GetTotals	3	6	18	30	54	0	10:06:38.677

Figure 22: Example of the statistics tool within the APD to measure microflow usage.

Regarding the reference case, it is decided that only the main features as described in the case are implemented before the go-live of the application. Any additional feature requests are not implemented yet, because they are probably not going to be used often enough to make it viable. These features are described in user stories using the before-mentioned template. In addition, user flows are created to predict the user journey. After go-live, the APD will be used to track if certain features are used as predicted.

4.9 Is a media management/use policy in place to reduce their impact, with criteria for media compression and formats?

Media management and media use is the subject of this criterion. Product Owners mostly want to make sure that applications stand out by using compelling design or by displaying rich media. One can imagine that media with a relatively larger size increases the volume of data that is transferred when showing the media on a page in the application. The higher the data that is transferred, the higher emissions of greenhouse gas, which in turn increases the environmental footprint. The solution of this issue firstly lies in less usage of such media by finding the right balance between sobriety of the pages and the desired needs of the Product Owner. Secondly, when it is decided that media is required, it is important to use media compression and the more sustainable file formats.
A good way to achieve this is by making sure that an environmental quality checklist of the used media is made and managed. This checklist should check for compliance with the best practices of media usage, file compression and formats. Throughout the application lifecycle it is useful to check if pages comply with the best practices. The test for this criterion is whether an environmental quality checklist is used to check the media usage of the service. The higher the ratio of correctly compressed media against the total number of media used, the higher the score for this criterion.
A range of browser tools and consoles are available that can be used as quality checklists. An example of such browser tool is the Chrome extension named Green IT: best practices. For Mendix, it is possible to use this tool when using Chrome as the browser. This extension allows for an analysis of important indicators such as the eco index and shows whether best practices are adhered to. An example of an analysis that is performed on a page in an application is shown in figure 23. The page that is analyzed has eco index C, which can be improved by solving the best practices that are not correctly followed the in the list below. Furthermore, it shows the water consumption, the greenhouse gas emission, and the size of the page.



EcoIndex C

EcoIndex	Water Consumption (cl)	Greenhouse Gases Emission (gCO2e)
57.86	2.76	1.84

Request number	Page Size (KB)	Dom Size
219	2543 (5963)	0

Best practices

Add expires or cache-control headers	✗	12 / 16 resources cached
Compress ressources (>= 95%)	✓	98.8% resources compressed
Limit the number of domains (<6)	✓	2 domain(s) found
Don't resize image in browser	✓	
Externalize css and js	✓	No inline stylesheet or inline javascript found
Avoid HTTP request errors	✓	0 HTTP error(s)
Limit the number of HTTP requests (<27)	✗	219 HTTP request(s)
Do not download unecessary image	✓	
Validate js	✓	Javascript validate
Max cookies length(<512 Bytes)	✓	Max cookies length = 179 Bytes
Minified cs and js	✗	7/15 not minified css or js
No cookie for static ressources	✗	2 static ressource(s) with cookie (Total of 0.4KB)
Avoid redirect	✓	0 redirect
Optimize bitmap images	✓	No bitmap image to optimize
Optimize svg images	✓	No svg to optimize
Do not use plugins	✓	No plugin found
Provide print stylesheet	✗	No print stylesheet found
Do not use standarts social button	✓	No standart social network button found
Limit Stylesheet files (<=10)	✓	Not more that 7 stylesheets found
Use HTTP/2 instead of HTTP/1	✓	0/219 resources using HTTP/1
Use Standard Typefaces	✓	1 custom fonts found

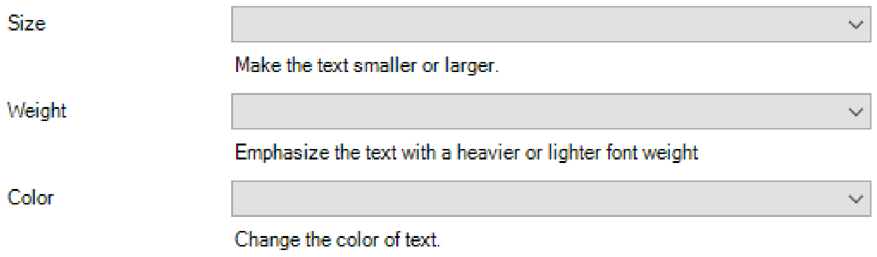
Figure 23: Green IT Chrome extension example analysis.

Regarding the reference case, Jack decided that every new page that is created for the application is tested using the Green IT extension. He also decided that no images or videos are used within the application.

4.10 Are the number of fonts and the variants of fonts called (weight, characters used in the project) limited?

This criterion continues the best practices of the UI of pages. It focuses on the usage of fonts, specifically on the number of fonts and variants of the fonts. Like transferring the data volume of media, fonts can also be large. It is therefore vital that the number of fonts used in an application is limited. The fewer fonts are used, the less the environmental footprint will be.
One of the ways to tackle this issue is by establishing a system typography, that sets a foundation of the font families and weights that are used throughout the IS. The goal is that all the applications within the IS use these standards. In the system typography it is important to not only choose a font family that is used throughout the application, but also to limit the number of font weights to 3, mostly the light, regular and bold weights. One can use Google Fonts webpage to check out the different font families and weights.
To test for the level of compliance of this criterion, it can be checked if a system typography is established. In the system typography, the fewer the amount of font families and font weights are used, the higher the sustainable IT score will be.
Applying the font weights to text is rather easy within Mendix Studio Pro. When editing texts, there are several standard options from which developers can choose, as can be seen in figure 24 below. Besides that, it is possible to style a text

using CSS more specifically. Jack decides to use only the regularly used family 'Open Sans' and three weights: light, normal and bold.



Size	
	Make the text smaller or larger.
Weight	
	Emphasize the text with a heavier or lighter font weight
Color	
	Change the color of text.

Figure 24: Configuring how the text is rendered within Mendix Studio Pro.

4.11 Does the service avoid Dark Patterns?

Another important criterion of the UX/UI family is regarding the avoidance of dark patterns. Dark patterns in web applications, also known as deceptive design, are features of interface design meant to trick the end-users into doing things that they might not want to do but benefit the business of the web application (Brignull, n.d.). An example of a dark pattern is to make it extremely hard to delete accounts, by making it a difficult maze to reach the button that deletes the account. Another example is the usage of colors by using the same color that initially means just to continue to a next page, but suddenly that same colored button leads to a purchase of a product.

Within sustainable IT, the focus is to prevent dependance or addiction of users on the digital service. It is vital that dark patterns are not used to avoid digital over-consumption of users. A good way to do that is by deploying a Captology (Computers as Persuasive Technology) study, which allows for insights of whether users are influenced by the digital service. The Captology study points out where the dark patterns are and where they should be tackled.

To achieve this, the UX/UI designer needs to make sure that this subscription perpetuation is banned from the service. The test for this criterion is then to check if the Captology study identified dark patterns. The fewer dark patterns are found, the higher the score for this criterion will be.

Considering the reference case, Jack makes sure that created accounts can easily be deleted, by asking the developers to add an account overview page, where the users can decide to delete their own account. Furthermore, the application will be free of charge and no purchases can be made within the application. In addition, button colors will be consistent in their function throughout the application.

4.12 Are rules in terms of respect for the user in place?

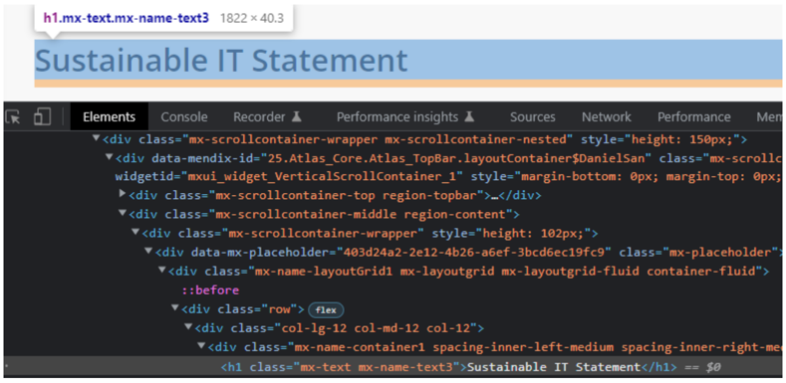
The final criterion of the UX/UI family targets another ethical topic regarding respect for the end-users. It builds on the criterion above regarding the avoidance of dark patterns, as it focuses on potential addictions applications can cause. Some applications are particularly designed to make users feel dependent on it. The goal is that users objectively remain in control of their digital consumption.
A good way to do this is to do a Captology study as mentioned in the criterion above. Other approaches are described in the toolkits of the Ethics for Designers website (Gispen, 2017). A digital ethics charter can be defined that outlines the ways that humans are placed at the center of the digital industries with the main goal to keep humans in control of their decisions.
The test for this criterion is whether a digital ethics charter is defined. If it is defined and it describes how possible addictions are avoided, then a higher level of compliance for this criterion is assessed.
Jack has decided to use the Moral Value Map toolkit of the Ethics for Designers website. This will allow him to see which values are relevant to his design and how his design affects his values. He finds that he values that his users do not have to enter any personal data other than the team they belong to. In addition, he wants that users can easily navigate to what they want within the app without unnecessary clicking through the application. Also, as mentioned above in the last criterion, the makes sure that users can easily remove their account if desired.

5 Contents

This section describes the Contents family of criteria. Adhering to the criteria of this family takes place during the whole development process of the application lifecycle. The word ‘Contents’ in this regard contains among others regarding uploading, downloading, formatting, and compressing media. The first essential criterion of this family touches upon the text formatting tags and whether they meet the need of prioritization of information, instead of just for enhancement of presenting the text. The second criterion focuses on compressing images when viewing them on a page. Next, this section zooms in on potential videos and their alternatives. Last, the concentration is on downloading documents and how that can be done adhering to Sustainable IT criteria. The roles that most tackle the issues of this family are mostly the front-end developers, as they develop the how the media is shown on the page. However, back-end developers and architects also have their role in making sure that data is optimally persisted and transferred. The four essential criteria of the Contents family are explained in the subsections below.

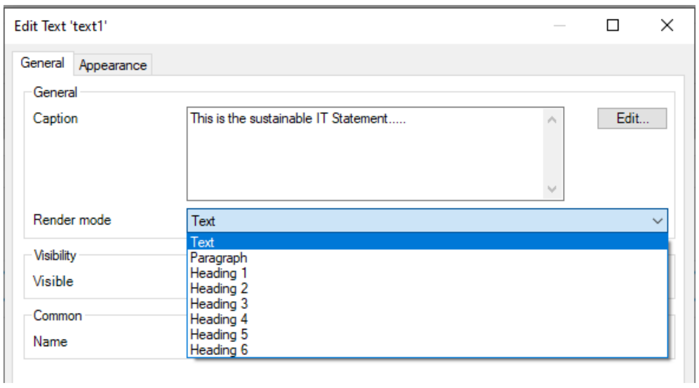
5.1 Do text formatting tags meet the need for prioritization of information, not presentation enhancements?

The first criterion of the Contents family targets the formatting of text and the usage of html tags. Automatons and accessibility assistants use html tags to be able to go through and understand how the text is read. These tags need to correspond to the structure of the content, otherwise these assistants will find it more difficult to understand the content. This criterion is part of the accessibility standards of the WAI (WAI, 2022).
The best way to do this is to make sure that the tags that are used should not only offer layout improvements, but also offer information in a way. Exceptions are the div, span and table tags. The front-end developers should analyze their usage of tags to see if it can be improved.
Compliance of this criterion is tested by measuring the number of analyzed and correct tags against the total number of tags. The higher the number of analyzed and correct tags, the higher the sustainable IT score will be.



```
▼<div class="mx-scrollcontainer-wrapper mx-scrollcontainer-nested" style="height: 150px;">
  ▼<div data-mendix-id="25.Atlas_Core.Atlas_TopBar.layoutContainer$DanielSan" class="mx-scrollc
   widgetid="mxui_widget_VerticalScrollContainer_1" style="margin-bottom: 0px; margin-top: 0px;
    ▶<div class="mx-scrollcontainer-top region-topbar">…</div>
    ▼<div class="mx-scrollcontainer-middle region-content">
      ▼<div class="mx-scrollcontainer-wrapper" style="height: 102px;">
        ▼<div data-mx-placeholder="403d24a2-2e12-4b26-a6ef-3bcd6ec19fc9" class="mx-placeholder">
          ▼<div class="mx-name-layoutGrid1 mx-layoutgrid mx-layoutgrid-fluid container-fluid">
              ::before
            ▼<div class="row"> flex
              ▼<div class="col-lg-12 col-md-12 col-12">
                ▼<div class="mx-name-container1 spacing-inner-left-medium spacing-inner-right-me
                    <h1 class="mx-text mx-name-text3">Sustainable IT Statement</h1> == $0
```

Figure 25: Developer tools view of showing the html tag within Chrome.



Edit Text 'text1'

General | Appearance

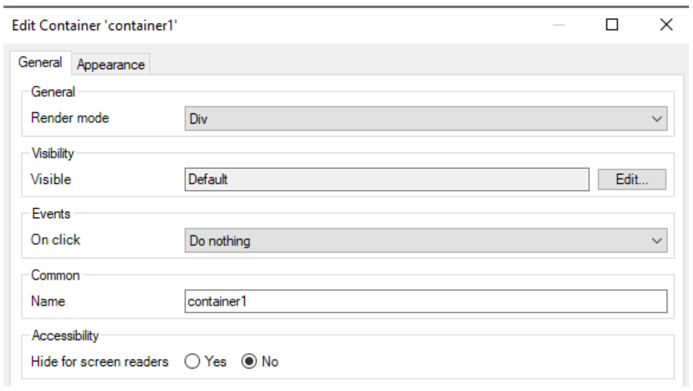
General

Caption: This is the sustainable IT Statement.....

Render mode: Text

Figure 26: The render mode selection window within Mendix Studio Pro.

As shown in figure 25 and 26, a notable example of tag usage on html pages is with the use of Headings. Headers allow readers to browse content by topical groups and provide context for users working through lengthy content. Users are much more likely to discover topics of interest if they are marked with a heading. This structure is especially important for screen reader users. Most screen reader users say skimming for headings is the first thing they do when they open a Web page. In figure 27, developers can choose if they want to include a header or tag as a screen reader element.


Edit Container 'container1'

General | Appearance

General
Render mode: Div

Visibility
Visible: Default [Edit...]

Events
On click: Do nothing

Common
Name: container1

Accessibility
Hide for screen readers: Yes [✓ No]

Figure 27: Screen reader selection window within Mendix Studio Pro.

If "normal" text is only formatted to look like a heading, e.g., by making it bold, it will not appear in this list, making it much less likely screen reader users will notice the content exists. Therefore, it is important to keep your tags structured and use them to convey the right information to the users.
Tab index, as seen in figure 28, is another tag that can be used in Mendix studio Pro. It can be assigned to any button or interactive element. Interactive elements should, under most circumstances, be focusable in the order that they appear in the code. This helps people who are using the keyboard or alternative input devices to follow focus in a logical order. The order that elements appear in the document source should reflect the order they appear visually. Adhering to these rules will decrease the nonessential time a user spends in finding the correct element or information on a website.

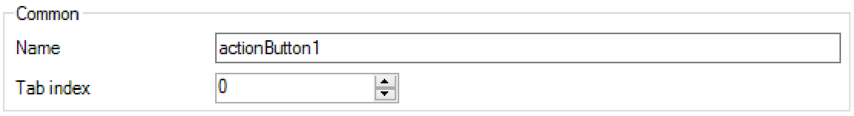

Common
Name: actionButton1
Tab index: 0

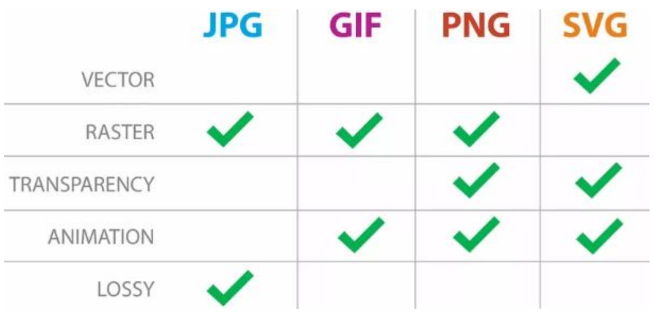
Figure 28: Tab index window within Mendix Studio Pro.

The front-end developers of the football calendar app decide that they will follow the accessibility guidelines as described by the WAI. They decide to only use the tags for text formatting if they give information to the end-user and to give emphasis on important topics. The developers control each other's work with a code review each week, where this topic is also considered.

5.2 Have the images been compressed upstream in a format suitable for viewing?

The second criterion of this family zooms in on images, their formats, and compressions. On one hand, it is essential that data is compressed when it is transmitted to be able to save resources. After all, it costs less energy to transmit an image of 1 MB compared to an image of 2 MB. On the other hand, the

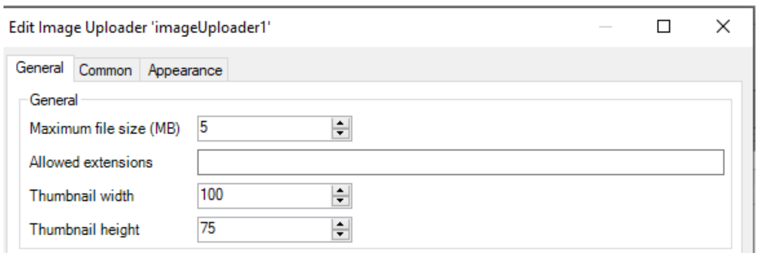
compression algorithm itself consumes technical processing resources. In addition, it requires intermediate storage. It is therefore vital to find the balance between compression of images and not making too many requests of the compression of the images.
To achieve this, it is firstly important that the correct formats are used for each situation. Jeff Cardello discusses the differences between the PNG and JPG formats in his blog on the Webflow website (Cardello, 2021). Here he sums it up as follows: *"The general rule is to use JPGs for photographs, images that don't have a transparent background, and other memory intensive files. And to choose PNGs for graphics, files with transparent backgrounds, and other images where clarity and color vibrancy are important."*
In another blog, Philip Westfall zooms in on what the best format is for web applications (Westfall, 2018). This can be summarized as follows in figure 29 below.



	JPG	GIF	PNG	SVG
VECTOR				✓
RASTER	✓	✓	✓	
TRANSPARENCY			✓	✓
ANIMATION		✓	✓	✓
LOSSY	✓			

Figure 29: When to use image formats for web applications (Westfall, 2018).

The second step is to make sure that the formats are suitable for each environment. They need to be visible without compression every time when changing environment. Another option is to not show the images on certain environments if they are not essential. To see if an image in the application is using the best practices, one can use browser extensions such as the Chrome extension named Green IT: best practices.
The test for assessing the level of compliance of this criterion is based on how well the images are analyzed and within the application. The more images are analyzed and formatted using the best practices compared to the total number of images, the higher the sustainable IT score will be.
Within Mendix it is possible to limit the format extension of images that are uploaded to the application. Therefore, based on the type of image that is expected, one can for example only allow the upload of JPGs for photos. This is done by using the image uploader widget as seen in figure 30. In addition, it is also possible to set a limit for maximum file size, which should always be used as well to prevent usage of unnecessary large image files.


Edit Image Uploader 'imageUploader1'

General | Common | Appearance

General

Maximum file size (MB): 5

Allowed extensions:

Thumbnail width: 100

Thumbnail height: 75

Figure 30: Image uploader widget has options to limit the extensions that can be uploaded.

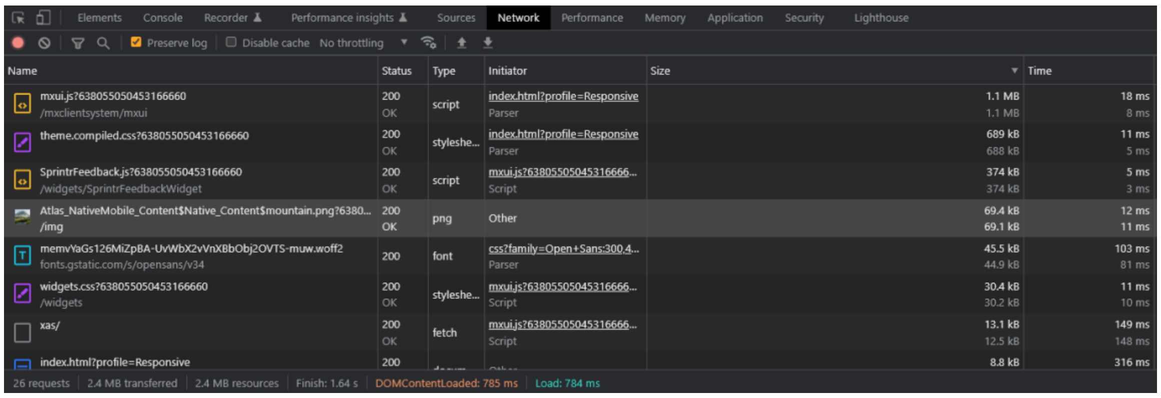
Jack decides that the only image that will be used is the image of the logo of the club that is to be used as button for the home page. This image will be a PNG, since it is required that the image is transparent with the background of the page.

5.3 Can the information carried by the video be replaced by an alternative (computer graphics …)?

The third criterion of the Contents family targets the next media format, namely videos. Firstly, videos take up one of highest shares of the application storage. This means when a lot of videos are stored in the application, the storage will be at maximum capacity sooner. Secondly, when playing a video on a page of the application, it requires a lot of resources compared to static forms of showcasing information. It is therefore vital that developers consider transposing videos into a static form, such as images, computer graphics or only showing the metadata of a video, whenever they can.
One of the ways to check if the content of a video is relevant is by checking the metadata of the video file. Developers can decide there if the size of the video is worth considering the goal and description of the video. If it is not worth it, or the goal of the video can simply be achieved by one of the examples mentioned above, then it needs to be replaced by the alternative. This needs to be done for each potential video.
The test for this criterion is performed by measuring the number of videos that are analyzed and potentially replaced against the total number of videos. The more videos are analyzed and replaced, the higher the score for sustainable IT will be.
Within Mendix Studio Pro it is best practice to choose a widget or module created by your own organization or by Mendix itself, when checking the app store for potential widgets. In case of showing analyzed videos, the Video Player widget can be used, which allows for showing videos from Youtube, Vimeo, Dailymotion and external MP4 files (Mendix Technology BV, 2022). Regarding the reference case, Jack decides that no videos are necessary. In addition, in contradiction to his earlier plan, he decides that no images will be used either to show the players of a team. The name, age and team name are sufficient.

5.4 Are the documents to download compressed, optimized and accessible?

The final criterion of the Contents family concentrates on the downloading of documents. Every upload and download of documents lead to transfer of data, which costs resources. Like images, other documents can be compressed, optimized, and be made accessible so that it is easier to download and upload documents while costing less resources. This is therefore an important goal of developers when creating functionality that deals with documents.
To achieve this goal, developers should use navigation tests to test how the user navigates through the application and check if the available documents are easily accessible to download. Another tool they can use are browser tools. One example is the network tab when inspecting the page using Chrome. In figure 31 this tab is shown.



Name	Status	Type	Initiator	Size	Time
mxui.js?638055050453166660 /mxclientsystem/mxui	200 OK	script	index.html?profile=Responsive Parser	1.1 MB 1.1 MB	18 ms 8 ms
theme.compiled.css?638055050453166660	200 OK	styleshe...	index.html?profile=Responsive Parser	689 kB 688 kB	11 ms 5 ms
SprintrFeedback.js?638055050453166660 /widgets/SprintrFeedbackWidget	200 OK	script	mxui.js?63805505045316666... Script	374 kB 374 kB	5 ms 3 ms
Atlas_NativeMobile_Content$Native_Content$mountain.png?6380... /img	200 OK	png	Other	69.4 kB 69.1 kB	12 ms 11 ms
memvYaGs126MiZpBA-UvWbX2vVnXBbObj2OVTS-muw.woff2 fonts.gstatic.com/s/opensans/v34	200	font	css?family=Open+Sans:300,4... Parser	45.5 kB 44.9 kB	103 ms 81 ms
widgets.css?638055050453166660 /widgets	200 OK	styleshe...	mxui.js?63805505045316666... Script	30.4 kB 30.2 kB	11 ms 10 ms
xas/	200 OK	fetch	mxui.js?63805505045316666... Script	13.1 kB 12.5 kB	149 ms 148 ms
index.html?profile=Responsive	200			8.8 kB	316 ms

26 requests | 2.4 MB transferred | 2.4 MB resources | Finish: 1.64 s | DOMContentLoaded: 785 ms | Load: 784 ms

Figure 31: Example of the network tab within the Chrome browser tool.

A lot of information is shown on this tab. Some notable figures from left to right at the bottom of the page are:

- Total number of requests done.
- Total amount of data transferred.
- The amount of time it took to load the page.

Then, above this information, a list of items is shown of all the requests called the network log. Important for this criterion are the 'type' column that shows what the format of the request is, the 'size' column that shows the size of the resource, and the 'time' column that shows how long it took to upload or download this request. When using the 'use large request rows' option, the size column shows two amounts of data. Here the top value is the uncompressed size, while the bottom value is the compressed size. Therefore, if they are the same size as in the fourth request of figure 28, the developer can see that the compression did not work.

The test for this criterion is by comparing the number of compressed documents to be downloaded against the total number of documents to be downloaded. The higher this ratio is, the higher the score for sustainable IT will be.
In Mendix, as mentioned earlier in this family of criteria, it is possible to limit the size and formats of uploads of documents. It is also possible to write custom java actions that compress images or other files. Jack decides that no images or videos are used in the application. He does however want to be able to export the schedule of a team as an ICS file for users to download, which is implemented as a feature as part of the calendar page.

6 Front-end

The family of criteria that also requires a more detailed inspection is the Front-end family. Front-end development refers to that area of web development that focuses on what the users see on their end. It involves transforming the code built by back-end developers into a graphical interface, making sure that the data is presented in an easy-to-read and -understand format. One of the essential criteria focuses on using environments, tools and functionalities that limit impact on the planet. This relates especially to the use of API's; it is important that the functionalities covered by local actions are privileged rather than API exchanges while also keeping an eye on the front-end load. Using proven development standards throughout development can also make sure that functionalities do not go beyond user needs, as every function with little use has an impact on the environmental footprint. Another criterion is reducing the effects of obsolescence by making sure that a backward compatibility range is defined, this makes sure that all client devices get catered with the right number of functionalities. The UI also plays an important role in IT sustainability, by implementing technical solutions with the lowest impact like having a clean version of a product for print can limit the environmental footprint. The same can be said with implementation of sober behaviors, by consciously choosing which type of media to display will have a positive impact on the processing resources. The five essentials of the Front-end family are further discussed in the following sections.

6.1 Are the functionalities covered by local actions (client side) privileged rather than API exchanges?

The first criterion of the Front-end family targets the use of API's. API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software (Amazon Web Service, Inc., 2023). APIs let your product or service communicate with other products and services without having to know how they're implemented. This can simplify app development, saving time and money. Due to these benefits API usage is increasing rapidly in numerous industries as a critical interface for connecting systems and data.
The loads introduced by using APIs can be significant as introducing an extra service tier will always have a cost in complexity and performance. Software that has multiple web apps or multiple desktop apps connecting directly to the same database and performing the same queries often would benefit from an API service. But if it is a single web app which connects to the database directly then there is less benefit to introduce an additional service layer, where the benefits

then lie with easier regulation of security and access to the application and with data processing. Implementing API's only when its necessary will reduce the volume of exchanges and the number of components used which ultimately leads to a reduced environmental footprint.
The test for this criterion is performed by doing API performance and load testing but keep in mind that API testing is a broad category and running performance tests on APIs cannot be done with just a single test. Instead, tests should be divided strategically into different components and running specific actions to address each one. (For example, standard load testing) The results of these tests then be compared to the local actions that don't make use of Rest/API services.
Mendix has the possibility to incorporate APIs in every application. When considering the reference case Jack decides to only use an API to retrieve scores from past competitions and other teams' information like player names and used strategies from a central database. Other data that contains his team information will be directly pushed to own database without the use of any rest service.

6.2 Is the backward compatibility range defined?

The second criterion of the Front-end family focusses on the backwards compatibility of user equipment. Every software product is bound to change according to the demands of the users. Hence tasks performed on an older version must work as before in the newer versions as well (Devopedia, 2022). Backward compatibility refers to a hardware or software system that can use the interface of an older version of the same product. A new standard product or model is considered backward compatible when it can read, write, or view older formats. Backward compatibility allows newer technology to advance without superseding a current component.
Achieving backwards compatibility starts by writing application code in such a manner that an existing functionality will continue to work and will not hamper the exiting functionality. Backward compatibility doesn't only exist in code, it also needs to be maintained in databases, apps, API's, hardware, and libraries. So, when designing or improving any software, keep compatibility in mind, keep the list of user equipment updated and keep older code compatible with new code.
Backwards compatibility for software can be tested with unit tests that can verify if a functionality is intact with a new release. Tests should be written in such a way that they would fail when encountering any backward compatibility issues. An ideal testing environment would contain a test suite that will fail and alert when there are issues with backward compatibility. This can also be done by automated testing on the CI/CD pipeline which checks for any incompatibility and alerts when there is a violation.

All Mendix Studio Pro versions come with a detailed changelog with backwards compatibility information, furthermore the official Mendix documentation contains all compatible browsers per Studio Pro version. Regarding the reference case, Jack decides that the application will be built using the latest Mendix Studio Pro version (9+) because it supports all the major browsers and even older ones like Internet Explorer.

6.3 Don't the functionalities of the service go beyond user needs?

The third criterion of the Front-end family focusses on functionalities and user needs. The ideal software development process balances the client's business needs and the idea of how the end customer will use the software (Gibbons, 2019). Keeping the latter in mind helps to develop software that not only looks good but also provides value for the users as the number of features available in a digital service directly impacts the consumption of resources and the environmental footprint. By creating functions of little use, they are sources of unnecessary environmental debt. The features also increase the volume of the service, which introduces additional loading times and storage space.
To meet end user needs, first we must understand the problem the product will be solving or what necessities it will be fulfilling and then to define and understand the future user (Gorbachenko, 2023). Achieving this requires a deep dive into researching existing products, solutions, market conditions, user analysis, etc.
As mentioned in the third essential criterion of the family specifications; software development is a continuous process and should be an never ending cycle of receiving, analyzing, and applying feedback to make the product better. This data can come in various forms, including surveys, data analytics, and customer feedback. The collected data on the functionalities can be analyzed to ensure that those features are of real usefulness.
Regarding the reference case, Jack will be implementing page analytics on all functionalities to keep track of usage. Also, by adding the option for user feedback Jack can collect data that can be used to improve the application and by continuous analysis of the application Jack can ensure that the implanted features are also being used.

6.4 Is a clean version available for prints?

The fourth criterion of the Front-end family focusses on the availability of print features that are optimized to reduce space and ink usage. It's important that

applications and websites are accessible in as many use cases as possible, including print. No matter the situation, if a user finds it necessary to print a page on your website, it should be properly optimized.
The flows of dematerialization are sometimes broken and require the production of writing on paper. The user in some cases may also be more comfortable using printed documents rather than 100% electronic. Prints have a significant environmental footprint, whether in the consumption of paper, consumables such as cartridges and printing inks. The presentation needs in electronics or in printing are different; on the other hand, limiting the environmental footprint must lead to a reduction in the density and volume of printed information.
Regardless of the type of content, if users try to print something and they end up with multiple pages of unnecessary elements or content that is unreadable, they will likely be unhappy with the website. You can make sure the content is printable (and improve the user experience) by optimizing your print styles. Printed content should be both readable and as streamlined as possible, with all irrelevant clutter removed from the page (Circle, 2019).
In Mendix we can use CSS on the pages to create print specific styles by implementing the <link> element or by using media queries (Buckler, 2020). Regarding the reference case, Jack will be implementing print specific styles on all pages by making sure that all backgrounds are removed, texts are adjusted for better readability, the correct page margins have been applied and hiding web only functionalities.

6.5 Are cartography objects, animations, videos presented in a static mode?

The last criterion of the Front-end family focusses on the implementation of sober behaviors in applications. Clearly, animations in web design range from the effective to the distracting. That's why it's important to be more mindful on how it's used.
Mapping objects are not always relevant to be presented in the form of dynamic and interactive objects (Scacca, 2020). The static image version of the plans can provide the user with enough information without having to interact with the object. Mapping objects are complex and require processing resources and the use of sensors, which the image versions of plans avoid. The user retains the possibility of interaction by activating the dynamic version but only at his request.
Consider the downsides of an excessive number of animations on your website. If the site is just used as a portfolio to impress potential clients, then the slow speeds wouldn't matter as much. However, if your website needs to grow your online visibility and sales, you must be very careful about how much animating you do, as it could throw the whole thing in jeopardy. If the animations are necessary, look

for ways to further optimize your site for speed. It’s the only way to keep from losing visitors before they’ve had a chance to see them.
Looking at the reference case, Jack decided that no animation will be used in the application. There is no necessary requirement for it.

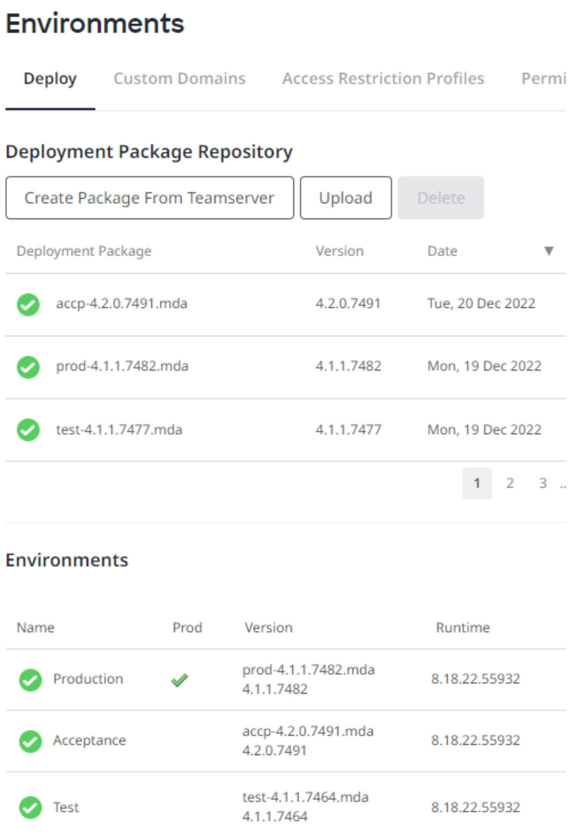
7 Architecture

Another important family of criteria is the Architecture family. Initially, adhering to the criteria of this family is done before the actual development starts. It is however also a continuous process throughout the application lifecycle. In Mendix, typically the architect or lead developer of the application initially sets up the application. After that, the security, main navigation and data models for each module are created. These activities should be part of sprint 0 or at the latest should be first stories picked up in sprint 1. In parallel, the environments where the application will be deployed on are requested. The first essential criterion of this family touches upon the availability of the development and quality assurance environments. The second essential criterion focuses on the lifespan and end of life information of components of the application. The third and final essential criterion zooms in on the identification of the technical equipment that is required for the application. The three essential criteria of the Architecture family are explained in detail in the subsections below.

7.1 Are the environments other than production (DEV, QA, ...) switched off or decommissioned outside the ranges of use (at night, outside the test periods)?

The first criterion of this family concentrates on the used environments and when they should be decommissioned. Applications typically are deployed across several environments. The most important environment is the production environment, which is only used after the app went live. The production environment is where the end users are accessing the application using real data. Then, there is the QA or acceptance environment, which is used for UAT testing. This is normally done by the Product Owners in an agile way of working. Thirdly, there is a potential test or development environment, which can either be local, in the cloud or hybrid. This environment is used by the development team to be able to do the IT testing. Under normal circumstances, the production environment needs to be turned on all the time, except for maybe during major releases and only if no rolling releases are used. The other environments are only actively used during testing periods. This means, that those environments can be turned off during inactive periods, such as overnight. If they are not used for a longer period of time, they need to be decommissioned. Turning off environments will lead to reduction of energy consumption. Research done on this subject indicate that 30% of servers are in a comatose state, which means that nearly a third of the total number is wasting energy (Koomey & Taylor, 2015).

The best way to make sure is to agree to fixed testing times with the IT and UAT testers. This should be documented in the technical documentation. Then all the environments should be turned off during non-active hours. Scheduled events should not be planned during those hours on those environments, as they will not be triggered if the application is inactive.
To test the level of compliance of this criterion, an application should have the above-mentioned documented and executed. If that is the case, then the full score for sustainable IT is achieved. If not, but the problems are identified or there are plans to tackle the problem, then only a part of the score is achieved.
In Mendix, after the environments have been requested, it is rather easy to maintain them as the responsible person. The Mendix Sprintr portal (Mendix Technology BV, 2022) allows for creating and deploying deployment packages on one of the available environments, as well as managing the custom domains, access restriction profiles, permissions, and services. In addition, it is also possible to check the logs, the metrics and create backups of the databases. Figure 32 below shows that for an example application, there are three environments: Production, Acceptance and Test. New deployment packages can both be created here from the team server or directly uploaded.



Deployment Package Repository

Deployment Package	Version	Date
✓ accp-4.2.0.7491.mda	4.2.0.7491	Tue, 20 Dec 2022
✓ prod-4.1.1.7482.mda	4.1.1.7482	Mon, 19 Dec 2022
✓ test-4.1.1.7477.mda	4.1.1.7477	Mon, 19 Dec 2022

Environments

Name	Prod	Version	Runtime
✓ Production	✓	prod-4.1.1.7482.mda 4.1.1.7482	8.18.22.55932
✓ Acceptance		accp-4.2.0.7491.mda 4.2.0.7491	8.18.22.55932
✓ Test		test-4.1.1.7464.mda 4.1.1.7464	8.18.22.55932

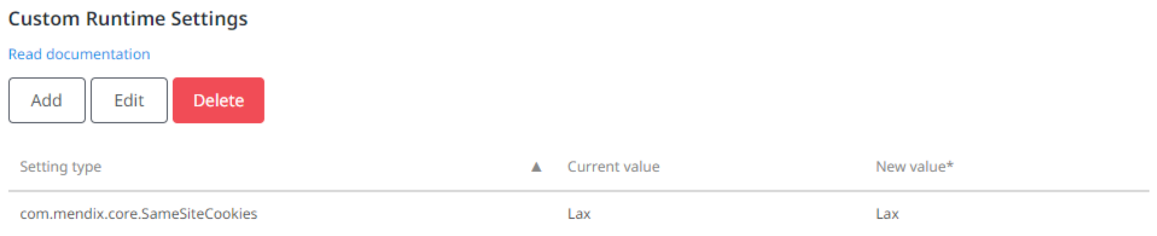
Figure 32: Image of available environments and deployed packages on the Mendix Sprintr portal.

For the reference case, the architect decides that only a Production- and Acceptance environment are required, because the IT testing can all be done locally as well. Therefore, no specific Test environment is required. The acceptance environment is only used during the normal work hours, while the production environment is turned on permanently, so that the end-users always have access.

7.2 Is each component deployed qualified from the point of view of its lifespan, and are the deprovisioning procedures systematically expressed?

The second criterion of the Architecture family is about end-of-life information of component that are deployed. Components, among others the used software, hardware, data center, each have a lifespan. The longer the lifespan for each of the components, the better. Components can also influence each other's lifespan, due to for example compatibility constraints. It is important that it is known what the lifespan of each component is and what they depend on. If inactive components

are badly maintained or forgotten, it not only causes a waste of resources, but it also could lead to weakness in security.
A good way to be prepared for the end of a lifespan of each component is by documenting end-of-life information in technical documentation. It is also good to mention the dependencies there. Furthermore, it is important that a withdrawal process is in place for when the component nears its end of life. The documentation should include information about recurring processing, data and backups to cope with all the consequences of the withdrawal of a component.
The test for this criterion is by checking if the technical documentation contains information about the lifespan of each component and whether there is a plan for each of the components. The more components are documented, the higher the score for sustainable IT will be.
An example of why it is good to document is when the Mendix version is upgraded to a new major version, in the case of this example, from Mendix 8 to Mendix 9. Firstly, the app directory in the explorer is different which for example has consequences to the styling classes. Secondly, some of the widgets are no longer compatible. This means that the features they were used for now either needs to be removed or updated. Thirdly, it can be that imported Appstore modules can only be functioning on Mendix 8 or that the functionality in those modules changed a bit how it worked in Mendix 9. The Deeplink module in combination with Single Sign On for example works slightly differently in Mendix in some cases. This can be due to a runtime setting called SameSiteCookies, see figure 33, which needs to set to Lax in Mendix 9, but this is not necessary in Mendix 8. Now this could lead to a security issue, so it is something to take into consideration when upgrading.



Setting type	Current value	New value*
com.mendix.core.SameSiteCookies	Lax	Lax

Figure 33: Adding a custom runtime setting to an environment on the Mendix Sprintr portal.

For the reference case, Jack decided to use Mendix 9 from the beginning to prevent a later upgrade. In addition, the architect decides that only modules and widgets are used that are also going to be compatible in Mendix 10. The technical documentation is extended with the end-of-life information of each component.

7.3 Are all the technical equipment used by the service identified?

The final criterion of the Architecture family discusses whether all the technical equipment that is used or going to be used, is identified. Throughout the lifecycle of the application plenty of technical resources are used to be able to provide the required solution. This starts with applications used to determine the strategy of the application. Then applications are required to provide an appropriate way of working, such as scrum tools. Other applications that can be used are design tools for the initial designs, communication applications and an availability management tool. Finally, all the components that are required for the actual development and hosting are required. It is important to identify each of these technical resources and their characteristics. Preferably the chosen equipment can integrate with each other so that the combined equipment is more sustainable.
A good way to make sure that all the technical equipment is identified and characterized is by using configuration management. According to Atlassian, configuration management is an IT management process that tracks each configuration item of an IT system through identification, baseline, version control and auditing. Configuration management helps engineering teams build robust and stable systems using tools that automatically manage and monitor updates to configuration data. (Atlassian, Ian Buchanan, 2022).

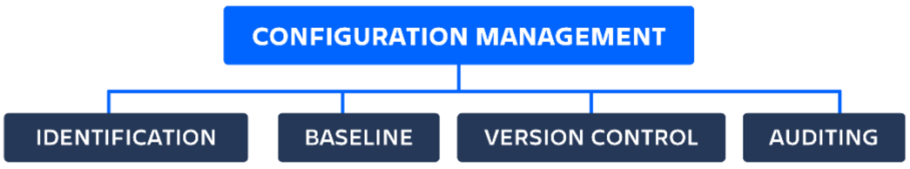


Figure 34: Configuration management by Atlassian (Atlassian, Ian Buchanan, 2022).

The test for this criterion is to check whether it is known what all the technical equipment throughout the development lifecycle is going to be. If they are not or no longer available, it needs to be known what equipment is to be purchased or replaced. The better the state of each technical equipment is identified, the higher the sustainable IT score will be.
For the reference case, it is decided that Mendix is used as both the agile tool and development tool. As discussed earlier, there are a lot of advantages of using the Mendix Sprintr in combination with the development tool Mendix Studio Pro. Figma is used to do the designs. Microsoft Teams will be used for communication within the team. The Mendix cloud hosted on AWS is the chosen platform to deploy and the application on.

8 Back-end

The next family of criteria is the Back-end family. This family is one of the most important regarding the reduction of unnecessary usage of scarce resources. The architect and the back-end developers continuously keep the criteria of this family in their mind when developing. The first criterion zooms in on the reduction of requests, for example by not doing requests in loops. The second criterion focuses on reusability of functionality and the role of documentation regarding this. The third criterion then once again discusses the importance of data compression before transmission. Finally, the last criterion then concentrates on one the most important aspects of a functionality: whether it is useful or not. The four essential criteria of the Back-end family are described in detail in the subsections below.

8.1 Is the number of requests kept to a minimum (no looping)?

The first essential criterion of the Back-end family is one of the most well-known criteria. This criterion focuses on limiting the number of requests made to a minimum. Requests to the database are done in the back-end of the application and it is therefore the back-end developers that deal with this criterion. Requests can for example be a retrieve, commit or delete action. Every time a request is done to the database, it will require resources from both the sender and the recipient to do so. It is therefore better to keep this to a minimum. One of the most common issues is when requests are executed in loops, where for each iteration one or more requests are executed.
A good way to tackle the loop issue is to do the requests outside of the loop. The retrieval of data should be done before the loop commences. If the iterator influences a subset of data that is retrieved before the loop, it is possible to filter the required data within the loop within doing a new request to the database. Then within the loop, all the changes to the objects and variables can be performed, as well as the creation of new objects or the deletion of objects. Regarding the deletion of objects, the objects are not actually deleted in the loop. It is best practice to add the to-be-deleted objects to a list that is marked to be deleted after the loop. After the loop, the changes, creations, and deletions can then be committed to the database all in once for each entity. The Green IT: best practices Chrome extension has the option to check the performance and the number of requests made. See figure 23 for an example of the usage of the extension. The extension can be used to see if a loaded page has a high number of requests.
The best way to test the level of compliance of this criterion is to go through the code and check how many requests are executed within the loop where it could

have been avoided. The more requests are done outside the loop in relation to the total number of requests, the higher the score for sustainable IT will be.
In Mendix, this criterion is mostly applied in microflows and nanoflows, but is also valid for custom Java actions. For microflows and nanoflows, it is best practice to create an empty list before the loop. Within the loop, the list will be filled with the changed or created objects that need to be committed or deleted. where the list will be added to within the loop. Then, after the loop, the list can be committed or deleted with one action. Figure 35 below shows an example of a microflow where a change is committed outside the loop. This microflows retrieves all the accounts as a list and retrieves the language object before the loop. In addition, an empty list is created, which is then filled in the loop. After the loop, this list is then committed. In this case, there are only three requests executed to the database, which is a considerable improvement depending on how many iterations there would be in the loop.

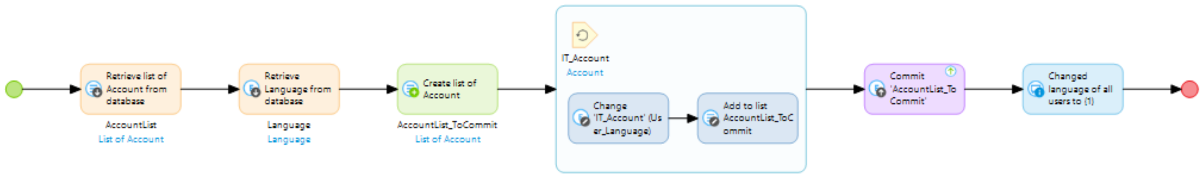
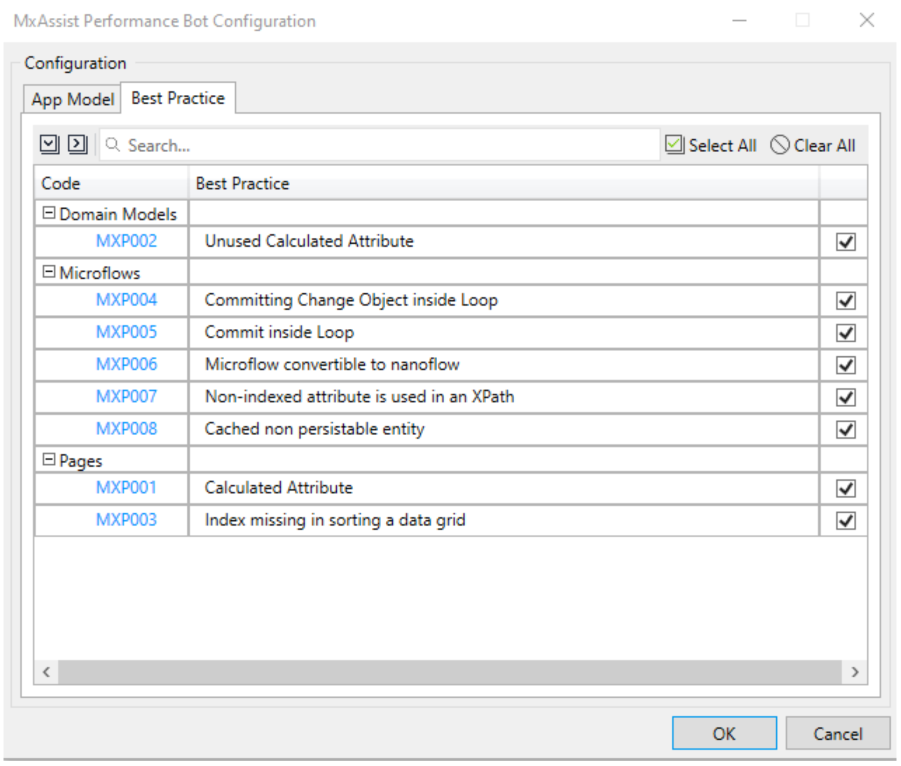

Retrieve list of Account from database
AccountList
List of Account
Retrieve Language from database
Language
Language
Create list of Account
AccountList_ToCommit
List of Account
IT_Account
Account
Change 'IT_Account' (User_Language)
Add to list AccountList_ToCommit
Commit 'AccountList_ToCommit'
Changed language of all users to {1}

Figure 35: Example of a microflow where a change is committed outside the loop.

Mendix 9 has a performance assist bot that outlines when commits are done in a loop. This, and other best practices, can be turned on and off in the MxAssist Performance Bot Configuration screen as seen in figure 36. It is important that these are checked regularly. There are also other code reviewers available, created by external parties that can potentially be used, such as the ACR (Application Code Reviewer) created by Clevr (Clevr, 2022).



MxAssist Performance Bot Configuration

Configuration

App Model | Best Practice

Search... Select All Clear All

Code	Best Practice	
⊟ Domain Models		
MXP002	Unused Calculated Attribute	☑
⊟ Microflows		
MXP004	Committing Change Object inside Loop	☑
MXP005	Commit inside Loop	☑
MXP006	Microflow convertible to nanoflow	☑
MXP007	Non-indexed attribute is used in an XPath	☑
MXP008	Cached non persistable entity	☑
⊟ Pages		
MXP001	Calculated Attribute	☑
MXP003	Index missing in sorting a data grid	☑

OK Cancel

Figure 36: MxAssist Performance Bot Configuration.

The back-end developers of the football calendar application decide to have the best practices in place regarding the commits outside the loop. They decide to use the assist bot to regularly check this, as well as code reviewing each other with each new feature that is built.

8.2 Is the documentation of the functionality available to enable its reuse?

The second essential criterion of the Back-end family zooms in on documentation of functionalities. This criterion has been focused on as well in the Specifications family. This criterion specifically zooms in on the documentation that enables easier reuse in other projects. Most often, in an organization there are common functionalities that are used in multiple applications, such as design systems, common modules or APIs that can be called. Examples of modules that are reused are user management modules or audit trail modules. To be able to profit from the already created functionalities, it is important that they are documented well, specifically the sustainable IT aspect, but for other aspects as well.
The best ways to do this have already been described in the Specifications family. To reiterate, documentation can be done in three locations:

- In the user story describing the functionality.
- In the code within the development tool.

- In a separate technical documentation document.

The second and third options are preferred when looking at reuse of functionality, as a new user story would be created for the functionality in the other application. The code can be reused, including the comments and annotations. The technical documentation might be expanded based on different input or output parameters of the functionality for each of the applications, but it should stay as generic as possible. What is important for documentation of the piece of code, module or API when looking at reuse, is that it is describing:

- How to implement and use it.
- How updates and versioning is handled.
- The end-of-life information.

The test for this criterion is whether the documentation describes how to reuse the functionality in other projects. The more functionalities this describes, the higher the level of compliance for this criterion will be and therefore the higher the sustainable IT score will be.

Regarding the reference case, reuse is not really prioritized as it is currently the only application of the football club. However, each functionality will be documented as described in the Specifications family.

8.3 Is the data exchanged compressed before transmission?

The third essential criterion of the Back-end family discusses data compression. The compression of media files such as images and videos have also been discussed in the essential criteria of the UX/UI family. The idea behind data compression before transmission applies to any kind of data. When sending data, it requires less resources when data is compressed. However, compression and decompression also require processes to be run, which could negate the created benefit if required too often. An example is when archiving data, it is decided to compress the complex data that consists of several tables in a data model into one archived table with a JSON string attribute, where the JSON string represents the complex data structure. This new table requires less resources to transmit. However, it is harder to perform complex tasks on the data, such as data analysis. Therefore, when data analysis is required, the data needs to be decompressed again, which requires resources. If the data analysis were to happen often, it might not be worth it to compress it to the JSON structure.

The goal is to find the right balance. For that it is necessary to know what the required resources are when compressing the data. This is then compared to the required resources of transmitting decompressed data. The comparison is done taking into consideration also how often the data needs to be transmitted versus how often it needs to be compressed and decompressed. In addition, it is common

to use browser tools to check the best practices such as the earlier-mentioned Chrome extension GreenIT: best practices.
The test for assessing the level of compliance of the criterion as stated by the Institute for Sustainable IT is by comparing the size of network transfer compressed against the total size of network transfer (Institute for Sustainable IT, 2021). The higher the compressed size, the higher the sustainable IT score would be. However, this does not adjust for the losses caused by compression and decompression. Therefore, a better test would be to check for the losses of compression and decompression against the losses for transmitting decompressed data for each functionality and make a conscience decision for each functionality. The score for sustainable IT is then decided by comparing the total number of conscience decisions taken in this regard against the total number of functionality where data is transmitted.
Looking at the reference case, it is decided to compress data that is transmitted quite often and does not require complex tasks performed on it. On the other hand, data will not be compressed if it is not transmitted often or is used for data analysis, such as the statistics of the matches,

8.4 Is the envisioned functionality useful?

The last criterion the Back-end family discusses the usefulness of functionalities. Earlier, it was mentioned that only functionalities that are useful should be considered when designing the application. Only functionalities where benefits outweigh the costs are classified as useful, with special attention to existence of the sustainable IT impact of the functionality. During UAT and after go-live, functionalities are either considered useful or not, and can therefore be adjusted or removed from the application if they are not useful.
A good way to decide if a functionality is useful from an end-user perspective is by collecting data of the usage of the functionality. Analytics can be used to decide whether it is worth keeping the functionality. In general, the more often a functionality is used, the higher its usefulness. This is not always the case, however. In some cases, an action is only performed once every year, but is still vital for the business workflow of the application. The decision to keep a functionality should be based on several factors such as frequency of usage, vitality for workflow, sustainability impact and others. To collect the data of a functionality, the GreenIT browser extension or the Opquast checklist can be used.
The test for this essential criterion is to check if each functionality has an identified use in the application. The level of compliance is based on the ration between the functionalities that have an identified use against the total number of functionalities. The more functionalities have an identified use, the higher the score for sustainable IT will be.

In Mendix the APD can be used to measure microflow usage, see figure 22. This can be used to decide whether a functionality is used often or not. Another way to do this is by using logs. Regarding the reference case, Jack decides to use the APD for this and asks the developers to use the GreenIT extension with each sprint as well. If a functionality is hardly used by the end-users, then it will be removed from the application to save resources.

9 Hosting

The last family of criteria is the Hosting family. This family of criteria is more focused on the hosting party and to a lesser extent on the application developers and architects. The first essential criterion focuses on the waste of the data centers and recovering or recycling. The second essential criterion then discusses the Power Unit Effectiveness (PUE). Thirdly, this family concentrates on the sizing regarding the physical infrastructure compared to the number of virtual machines to be run. The next essential criterion zooms in on the certification of the servers and technical equipment. Furthermore, sustainable IT performance indicators of the hosting party are discussed. Lastly, the question is asked whether the end-of-life hardware equipment is recycled or upcycled. These six essential criteria are each discussed in the subsections below.

9.1 Is the waste from the data center recovered or recycled?

Hosting an application has an impact on the planet, simply because it requires energy and resources. According to network-king.net, “in April 2019, it was estimated that datacenters worldwide used more than 2% of the world’s electricity and generated the same volume of carbon emissions as the global airline industry (in terms of fuel consumption)” (Zabeu, 2022). Furthermore, data centers produce e-waste such as racks, computing equipment, monitors, circuits, and any other electrical components within the infrastructure (Walbank, 2022). Reducing the impact of the hosting platform is important when looking at sustainability. One of the topics in this regard is reducing the e-waste that data centers produce while hosting an application. The first essential criterion focuses on recycling or recovering this waste.

The waste management and recycling need to be managed. To achieve this, the technical documentation should contain a section about tracking waste management. The documentation describes the methods that are applied. A well-known method to manage the e-waste of data centers is to use a circular design instead of a linear design, see figure 37. Circular Design Guide describes the difference as follows (Ellen MacArthur Foundation; IDEO, 2018). A linear design consists of three parts: make, use, and dispose. The disposing part is therefore creating waste that is not used again. On the contrary, a circular design makes sure that the generated waste itself or its individual parts are reused or recycled. Circular Design Guide offers content to learn about understanding, defining, making and releasing the methods of circular design. There are more methods to improve the efficiency of data centers, for example by using liquid immersion cooling instead of air cooling, which reduces the electricity needed by half (Green Revolution Cooling, 2022).

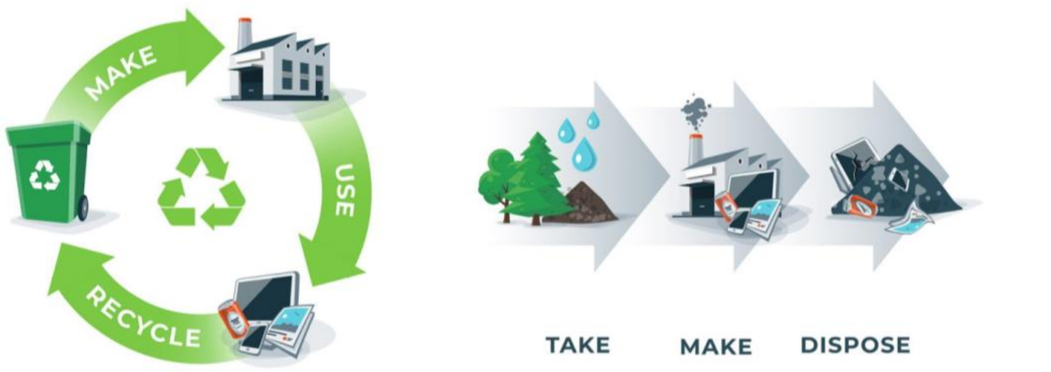


Figure 37: Linear design (right) versus a circular design (left) (A & A Packaging, 2015).

The test for this criterion is based on whether the means exist to check the waste treatment process. The more it is checked, and more waste is prevented or recycled, the higher the score for the sustainable IT will be.
When considering the reference case, Jack is looking for a sustainable solution regarding the hosting party. He decides that the hosting party of the football calendar app is selected based on the treatment of waste management. The way the hosting party is managing their waste is vital for this decision. Preferably, a party is selected that utilizes a circular design for data centers.

9.2 Is PUE (Power Usage Effectiveness) available?

The second essential criterion discusses the PUE (Power Usage Effectiveness) and whether it is tracked. This continues on the last criterion by looking at the power required by the data centers. The power usage is calculated by taking the total power consumed. This is done for the data centers and for the IT equipment within it (Gillis & Fontecchio, 2022), which leads to the following equation:

$$PUE = \frac{Total\ Power\ Data\ Center}{Total\ Power\ IT\ Equipment}$$

This ratio defines the overall efficiency of the entire infrastructure. The goal is to make this ration as small as possible, preferably smaller than 1.2, as stated by the Institute for Sustainable IT (Institute for Sustainable IT, 2021).
As discussed in the previous essential criterion, one of the methods to improve the efficiency of a data center is by using liquid immersion cooling instead of air cooling. Tech Target lists the following methods (Gillis & Fontecchio, 2022):

- Virtualize servers.
- Improve cooling systems:

- Optimize cool air production.
- Replace inefficient hardware.
- Use an energy-efficient uninterruptible power supply (UPS)
- Use energy-efficient lighting.

The test for this essential criterion is to calculate the PUE and to check what the value is. If the value is between 1.0 and 1.2, then the highest score for sustainable IT is gathered. Otherwise, the higher the value, the lower the score will be.
Mendix offers various solutions for hosting an application, including public cloud, virtual private cloud, private cloud, hybrid cloud, multi cloud, and traditional (virtual) servers. Jack decides to deploy the applications on the public Mendix cloud, because it is the most optimized cloud to run Mendix applications, according to the Mendix evaluation guide (Mendix Technology BV, 2022). This gives the best chance at achieving a low PUE.

9.3 Is the size of the physical infrastructure in line with the number of virtual machines to be run?

The previous essential criterion already mentioned that virtualization is useful when it comes to efficiency of the data center. The third essential criterion zooms in on this virtualization. Firstly, virtualization techniques allow for pooling the physical resources of a server and storage space. This means that the physical resources are utilized less and therefore require less scarce energy. This can optimize the use of the equipment. There is a risk however that the physical equipment is underused and therefore did not achieve its optimal capacity. This could also lead to the deployment of new equipment while the old equipment would still answers the needs. The goal is to find balance so that the physical park is in line with the virtual machines and to assure that both are optimized.
In general, it is desired that the virtual machines cover at least 60% of the usage rate, according to the Institute for Sustainable IT (Institute for Sustainable IT, 2021). To accomplish this, it is important that the sizing that is done initially for the application is documented in the hosting section of the technical documentation. Then, during deployment the actual usage rate of physical and virtual machines can be compared. If necessary, it can then be adjusted to make sure the usage rate of the virtual machines is at least 60%. Make sure that the physical park is not underused by reducing its size.
The test for assessing the level of compliance of this criterion is whether the relative ratio of the allocation of physical resources to virtual machines is monitored. Based on the usage rate of the virtual machines, if it is at least 60%, the full score for sustainable IT is achieved. Then, the lower the usage rate of the virtual machines, the lower the score will be.

As mentioned before, Jack decides to use the Mendix cloud for the deployment of the application. This allows for the optimal usage of the virtual machines due to its scalability. In addition, the usage rate is monitored so that future decisions can be made more easily. The Mendix metrics page in the developer portal are used for this (Mendix Technology BV, 2022).

9.4 Are the servers and technical equipment certified?

The fourth essential criterion of the Hosting family targets the certification of the servers and technical equipment. Nowadays it is becoming more and more common for servers, storage spaces and network equipment to adhere to the environmental standards, labels and certifications. It is vital that the application is hosted on an environment where the hardware is subjected to certification via an eco-label. This makes sure that the impact of the host on the planet is low.
The certification of the equipment needs to be documented in the technical documentation. Make sure that each of the servers, storage spaces and network equipment have a valid and recent eco-label. Examples of certifications are ASHRAE and TCO8. The latter, with generation 8, has a wide range of updated and new criteria and is designed to promote a circular design as described in one the essential criteria above. It gives importance to transparency and responsibility in the supply chain. In addition, it supports progress towards the SDG goals of 2030 as stated by the United Nations (TCO Development, 2023). There are several product categories of products that can be certified. For this handbook, the most important are the following:

- TCO Certified, generation 8, for network equipment (TCO Development AB, 2019).
- TCO Certified, generation 8, for data storage (TCO Development AB, 2019).
- TCO Certified, generation 8, for servers (TCO Development AB, 2019).

Each certification has an extensive guide which can be followed to get the certification, which this handbook recommends doing.
The test for this essential criterion is rather simple. It states that each of the technical equipment needs to have a recent and valid certification that follows the latest sustainability guidelines and strives towards the SDG goals. If each of the equipment has this certification, then the full score for sustainable IT is achieved. If not all the equipment has a certification, then only half of the score is achieved.
It is already decided by Jack to use the Mendix cloud, which is hosted on AWS, for the deployment of his application. AWS improves the sustainability of their cloud by focusing on power efficiency, cooling efficiency, and tracking performance. They state that AWS’ infrastructure is 3.6 times more energy efficient than the median of U.S. enterprise data centers and up to 5 times more efficient than the average in Europe. Furthermore, their goal is to power AWS with 100%, which they

plan to achieve by 2025. This would lead to 96% reduction of carbon footprint of their customers' workload (Amazon.com, Inc. or its affiliates, 2023). Jack also decides that all the information about the technical equipment is documented in the technical documentation.

9.5 Does the supplier share all its indicators?

The next essential criterion discusses the Sustainable IT performance indicators of the hosting party. Examples of these indicators are gas emissions, usage of renewable resources, resources consumption and waste. As mentioned before, the efficiency of data centers is important when looking at sustainability. To make sure what this level of sustainable IT maturity is, it is useful to use performance indicators. When these indicators are reported, it demonstrates an advanced level of maturity. It is vital though, that these indicators are relevant and reliable to be able to draw any conclusion about the level of sustainable IT.
The hosting party should provide the display and technical documentation of the performance indicators. The documentation should also list the sources of the indicators, so that can be determined whether an indicator is relevant and reliable. The test for this essential criterion is by checking whether all sustainable IT indicators are reported, and if they are reported, whether the source is reliable. If the answer to that question is yes, then the full score for sustainable IT is obtained. If only a few of the indicators have a reliable source, then half of the potential score is achieved for this criterion.
For the reference case, Jack decided to deploy the application on the Mendix cloud, hosted by AWS. Amazon provides a recurring sustainability report that describes all the indicators of sustainability (Amazon, 2021). This can be used to provide the technical documentation with the right information. The metrics on the Mendix developer portal could also be used to get an idea of the performance indicators (Mendix Technology BV, 2022).

9.6 Is the end-of-life equipment recycled or upcycled (sale / donation / recycling, etc.)?

The final essential criterion of the Hosting family is recycling or upcycling equipment that has reached the end of its life. As discussed in the first essential criterion of this family, technical equipment produces e-waste. That criterion specified the waste of data centers, whereas this criterion takes a view on the recycling or upcycling end-of-life equipment in general. Equipment that after its end of life is still capable of serving less intense uses should be reconditioned and reused in other areas of activity. If that is, however, not possible, the equipment

should be recycled, where each step of the recycling channel must be conform the sustainable IT requirements.
A good way to make sure that upcycling or recycling is following a sustainable process, is to trace the e-waste process of all the equipment. This should be covered in the technical documentation and be updated whenever the equipment reaches its end-of-life. Unitar, the United Nations Institute for Training and Research, has created several reports about monitoring e-waste. In one of their reports, they have created a mathematical model to measure the generated e-waste within countries (Forti, Baldé, & Kuehr, 2018). Although this is based on countries, the theory can be applied within organizations as well.
The test for this criterion is to check whether end-of-life equipment is recycled or upcycled, i.e. part of the circular economy. If so, then the chain of recycling should be documentation in the hosting technical documentation, which results in full score for sustainable IT. Otherwise, the less recycling or upcycling is done, the lower the score for sustainable IT will be.
As mentioned earlier regarding the reference case, Jack has decided to deploy on the Mendix AWS cloud. Amazon provides a recurring sustainability report including information on recycling (Amazon, 2021). This can be used to provide the technical documentation with the right information.

10 Summary

In current society, the problems regarding climate change are becoming more and more important and urgent. Especially the increase in magnitude of the so-called greenhouse effect is a common point of discussion. This effect is partially caused by the usage of fossil fuels to generate energy. The exhaustion of this limited resource leads to higher prices and inequality in the world. To tackle this, governments, organizations, and people strive for a more sustainable world. One of the ways to do this is to use alternative renewable energy sources, such as wind power, and solar energy, which takes time and requires big investments. Another more immediate approach is that less energy is consumed by daily activities by executing them in a sustainable efficient way. As organizations are rapidly digitalizing and launching digital services to drive the new way of working, it is no longer sufficient to just develop software applications that functionally meet the requirements. It is also desired to do this in the most efficient and sustainable fashion.

This handbook has focused on designing and developing software applications and describes the way in which this whole process adheres to a sustainable software engineering guideline. Examples have been described for Mendix applications, which are mostly also applicable for other applications. The handbook has provided a concrete translation of the guideline to a set of easy-to-follow instructions.

The handbook has been written to help organizations develop their low-code applications in a sustainable manner. It has helped to understand how to build sustainable software and is useful to a variety of stakeholders including program/project managers, business/product owners, (lead) software engineers, scrum masters, developers, architects, hosting parties, and service providers.

10.1 Families

The instructions have been based on the families and essential criteria of the Handbook of sustainable design of digital services (Institute for Sustainable IT, 2021). The first family is the Strategy family. The Strategy family contains criteria that target the early stages of the development cycle. Firstly, it is important that a business need is stated, so that no unnecessary features are developed. Secondly, a vital criterion is that the application and its features are subscribed to one of the SDG goals. Thirdly, an essential criterion is that the team and stakeholders are aware or certified in sustainable IT. Next, the decisions made regarding sustainable IT need to be exposed to the users. Furthermore, the IS is discussed, where it is important to have a consistent target for all the applications within the

IS. Finally, APM is used to tackle topics such as the balance between functional and technical debt, obsolescence, and variety and versioning of components.
The second family, the Specifications family, also targets the early stages of the development cycle as it focuses on stating the requirements of an application. It is important to know that in an agile way of working, this family stays relevant throughout the cycle. The first two criteria zoom further in on the awareness and training of the scrum team regarding sustainable IT. Then, a large criterion describes the product backlog items and that they should contain a sustainable IT component. Next, external service providers should be selected based on their sustainable IT ideals and standards. Finally, the chosen way of technical documentation needs to include end-of-life information of the used components.
The next family is the UX/UI family, of which the criteria are essential throughout the entire development cycle. For this family, the essential criteria mostly focus on the people and planet aspects, although the prosperity aspect is also important. Important topics are the human dimension of the stakeholders, accessibility, planet centric design, the extra-financial value of sustainable digital technology, more focus on awareness, compatibility, visual soberness, whether features are actually used by the user, media management, number of fonts and font variants, dark patterns and respect for the user.
Next up is the Contents family, which focuses on the kind of content shown on pages to the user. This family is again applicable throughout the whole development cycle, as new content is subject to the criteria of the family. The first essential criterion states that text formatting needs to meet the need of prioritization of information, instead of just for enhancement of presenting the text. The second criterion focuses on compressing images when viewing them on a page. Next, this section zooms in the usage of videos and their more sustainable alternatives, such as images or computer graphics. Last, downloading documents and how that can be done adhering to Sustainable IT criteria is discussed.
The fifth family is the Front-end family. The criteria of this family are once again essential throughout the entire development cycle. It has a lot of similarities with the criteria of the UX/UI and Contents families. The essential criteria touch upon the restricted usage of external APIs, the importance of backward compatibility, avoiding going beyond the user needs, having a clean version available for prints and sober behaviors of cartography objects, animations, and videos.
Subsequently there is the Architecture family. Architecture is an area which is mostly important in the early stage of development but stays very relevant thereafter as well. One of the essential criteria states that environments other that production should be switched off or decommissioned outside the ranges of usage. Another essential criterion discusses deprovisioning procedures should be documented regarding the lifespan of components. The last essential criterion discussed the importance of using configuration management to identify and characterize technical equipment.

The penultimate family is the Backed family, which is once again vital throughout the entire development of the application. The topics discussed in this family touch upon the following criteria: making sure that the number of requests is kept to a minimum to reduce the impact of data, documenting the functionality to increase maintainability and reuse, compression of data before transmission to limit the volume of exchanges and again the actual usefulness of a feature.
Finally, the last family of criteria is the Hosting family. This family is especially important for selecting a hosting party or for the hosting party itself. The essential criteria of this family firstly focus on reducing the impact of hosting by recovering or recycling the waste of data centers and by measuring the PUE. Secondly, a correct balance needs to be found between physical and virtual resources. Next, it is important to know whether the servers and technical equipment are certified. To monitor the hosting party, it is essential that the supplier displays all its indicators and lastly, end-of-life equipment should be recycled or upcycled.

10.2 Next Steps

As mentioned, this handbook only focused on the most essential criteria of sustainable software engineering. It laid the foundation for more criteria to be added to the handbook. In total, there are 516 criteria of which the 45 essential criteria are in this handbook. Therefore, there are plenty of criteria still to be discussed. The criteria that are left are either recommendations or advice but would still increase the level of sustainable IT of an application if adhered to. Another point of extension of the handbook could be extent the examples of the criteria to not just low-code, but to different kind of programming and programming languages.

10.3 Acknowledgement

We would like to thank everybody that supported the creation of this handbook. Special thanks go to Terryson Plaate, who contributed mostly to the Front-end family and to Jeroen Heikens, who is one of the initiators of this handbook and without him, it would not have been possible to create this handbook. Additionally, we thank Calvin Huynh for his input as an experienced application developer.

11 References

Paul Pacheco (UNSD), Ze Yar Min. (2021, November 11). *https://unstats.un.org/wiki/display/SDGeHandbook/Home*. Retrieved from https://unstats.un.org/wiki/display/SDGeHandbook: https://unstats.un.org/wiki/display/SDGeHandbook/Home

A & A Packaging. (2015). *How is Circular Economy different from Linear Economy*. Retrieved from A & A Packaging: https://www.aandapackaging.co.uk/how-is-a-circular-economy-different-from-a-linear-economy/

Amazon. (2021). *2021 Sustainability Report.* Retrieved from Sustainability, about Amazon: https://sustainability.aboutamazon.com/2021-sustainability-report.pdf

Amazon Web Service, Inc. (2023). *What is an API*. Retrieved from aws.amazon.com: https://aws.amazon.com/what-is/api/

Amazon.com, Inc. or its affiliates. (2023). *Sustainability in the Cloud.* Retrieved from Amazon, Sustainability: https://sustainability.aboutamazon.com/environment/the-cloud?energyType=true

Atlassian, Ian Buchanan. (2022). *Configuration Management*. Retrieved from Atlassian: https://www.atlassian.com/microservices/microservices-architecture/configuration-management

Bertin, E., & Crespi, N. (2014). Urbanization of Information Systems: An Outdated Method? In *Digital Enterprise Design and Management.* Springer International Publishing Switzerland.

Brignull, H. (n.d.). *Deceptive Design*. Retrieved from Deceptive Design: https://www.deceptive.design/

Buckler, C. (2020, January 5). *Printer friendly pages*. Retrieved from sitepoint.com: https://www.sitepoint.com/css-printer-friendly-pages/

Cardello, J. (2021, February 9). *PNG vs JPG*. Retrieved from Webflow: https://webflow.com/blog/png-vs-jpg

Circle, K. (2019, June 13). *print friendly styling*. Retrieved from tbhcreative.com: https://blog.tbhcreative.com/website-print-friendly-styling/

Clevr. (2022, February 25). *Mendix Marketplace*. Retrieved from Mendix: https://marketplace.mendix.com/link/component/114669

Devopedia. (2022, May 20). *Backward Compatibility.* Retrieved from Devopedia.org: https://devopedia.org/backward-compatibility

Ellen MacArthur Foundation; IDEO. (2018). *Circular Design Guide*. Retrieved from Circular Design Guide: https://www.circulardesignguide.com/

Forti, V., Baldé, C., & Kuehr, R. (2018). *E-waste Statistics: Guidelines on Classifications, Reporting and Indicators, second edition.* Retrieved from E-Waste monitor: http://collections.unu.edu/eserv/UNU:6477/RZ_EWaste_Guidelines_LoRes.pdf

Gibbons, S. (2019, March 24). *User Need Statements*. Retrieved from nngroup.com: https://www.nngroup.com/articles/user-need-statements/

Gillis, A. S., & Fontecchio, M. (2022, April). *Power usage effectiveness (PUE)*. Retrieved from Tech Target: https://www.techtarget.com/searchdatacenter/definition/power-usage-effectiveness-PUE#:~:text=What%20is%20power%20usage%20effectiveness,the%20IT%20equipment%20within%20it.

Gispen, J. (2017). *Ethics for Designers*. Retrieved from Ethics for Designers: https://www.ethicsfordesigners.com/

Google. (2022). *Google Sustainability*. Retrieved from Google: https://sustainability.google/

Google Inc. (2018). *The API Product Mindset.* Retrieved from cloud.google.com/files/apigee: https://cloud.google.com/files/apigee/apigee-api-product-mindset-ebook.pdf

Gorbachenko, P. (2023). *functional requirements vs non-functional requirements*. Retrieved from enkonix.com: https://enkonix.com/blog/functional-requirements-vs-non-functional/

Green Revolution Cooling. (2022, May 11). *3-ways to manage e-waste from data centers*. Retrieved from Green Revolution Cooling: https://www.grcooling.com/blog/3-ways-to-manage-e-waste-from-data-centers/

Institute for Sustainable IT. (2021). Retrieved from HANDBOOK OF SUSTAINABLE DESIGN OF DIGITAL SERVICES: https://gr491.isit-europe.org/en/

Koomey, J., & Taylor, J. (2015, June 3). Retrieved from Anthesis Group: https://www.anthesisgroup.com/wp-content/uploads/2019/11/Case-Study_DataSupports30PercentComatoseEstimate-FINAL_06032015.pdf

LeanIX. (2022). *The definitive guide to Application Portfolio Management*. Retrieved from Leanix.net: https://www.leanix.net/en/wiki/ea/application-portfolio-management

Learning for Sustainability - Allen, Will; Kilvington Margaret;. (2009). *Stakeholder Analysis.* Retrieved from Learning for Sustainability: https://learningforsustainability.net/pubs/Allen2009-Stakeholder_analysis.pdf

Learning for Sustainability - Will Allen (PhD). (2022). *Stakeholder Mapping and Analysis*. Retrieved from Learning for Sustainability: https://learningforsustainability.net/stakeholder-analysis/

Mendix. (2022). *Application Portfolio Management with Low-Code*. Retrieved from Mendix.com: https://www.mendix.com/application-portfolio-management/

Mendix Technology BV. (2022). Retrieved from Mendix: https://www.mendix.com/

Mendix Technology BV. (2022). *Mendix Accessibility*. Retrieved from Mendix: https://www.mendix.com/evaluation-guide/app-capabilities/accessibility/

Mendix Technology BV. (2022). *Mendix Cloud Overview*. Retrieved from Mendix: https://www.mendix.com/evaluation-guide/app-lifecycle/mendix-cloud-overview/

Mendix Technology BV. (2022, June 28). *Video Player*. Retrieved from Mendix: https://docs.mendix.com/appstore/widgets/video-player/

Mozilla. (2023, February 24). *Front-end web developer*. Retrieved from https://developer.mozilla.org: https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer

Planet Centric Design. (2022). *Planet Centric Design*. Retrieved from Planet Centric Design: https://planetcentricdesign.com/

Scacca, S. (2020, October 6). *website animation*. Retrieved from editorx.com: https://www.editorx.com/shaping-design/article/website-animation

Standish Group International, Inc. (2010). *Modernization.* Retrieved from standishgroup.com: https://www.standishgroup.com/sample_research_files/Modernization.pdf

TCO Development. (2023). *TCO Certified, New generation TCO certified*. Retrieved from TCO Certified: https://tcocertified.com/new-generation-tco-certified/

TCO Development AB. (2019). *TCO Certified, Generation 8, for network equipment.* Retrieved from TCO Certified: https://tcocertified.com/files/certification/tco-certified-generation-8-for-network-equipment.pdf

TCO Development AB. (2019). *TCO Certified, Generation 8, for servers.* Retrieved from TCO Certified: https://tcocertified.com/files/certification/tco-certified-generation-8-for-servers.pdf

TCO Development AB. (2019). *TCO Certified, TCO Certified, Generation 8, for data storage.* Retrieved from TCO Certified: https://tcocertified.com/files/certification/tco-certified-generation-8-for-data-storage.pdf

United Nations. (2022). *https://sdgs.un.org/goals*. Retrieved from https://sdgs.un.org: https://sdgs.un.org/goals

University of Cambridge. (2017). *Inclusive Design Toolkit*. Retrieved from Inclusive Design Toolkit: https://www.inclusivedesigntoolkit.com/

WAI. (2022). *WAI Standards/Guidelines*. Retrieved from Web Accessibility Initiative (WAI): https://www.w3.org/WAI/standards-guidelines/

Walbank, J. (2022, October 07). *Navigating and addressing the data centre e-waste crisis*. Retrieved from Data Center Magazine: https://datacentremagazine.com/articles/navigating-and-addressing-the-data-centre-e-waste-crisis

Westfall, P. (2018, January 23). *Web images png vs jpg vs gif vs svg*. Retrieved from Pagecloud: https://www.pagecloud.com/blog/web-images-png-vs-jpg-vs-gif-vs-svg

Zabeu, S. (2022, September 19). *Five Areas of environmental impact in data centres*. Retrieved from Network-King: https://network-king.net/five-areas-of-environmental-impact-in-data-centres/

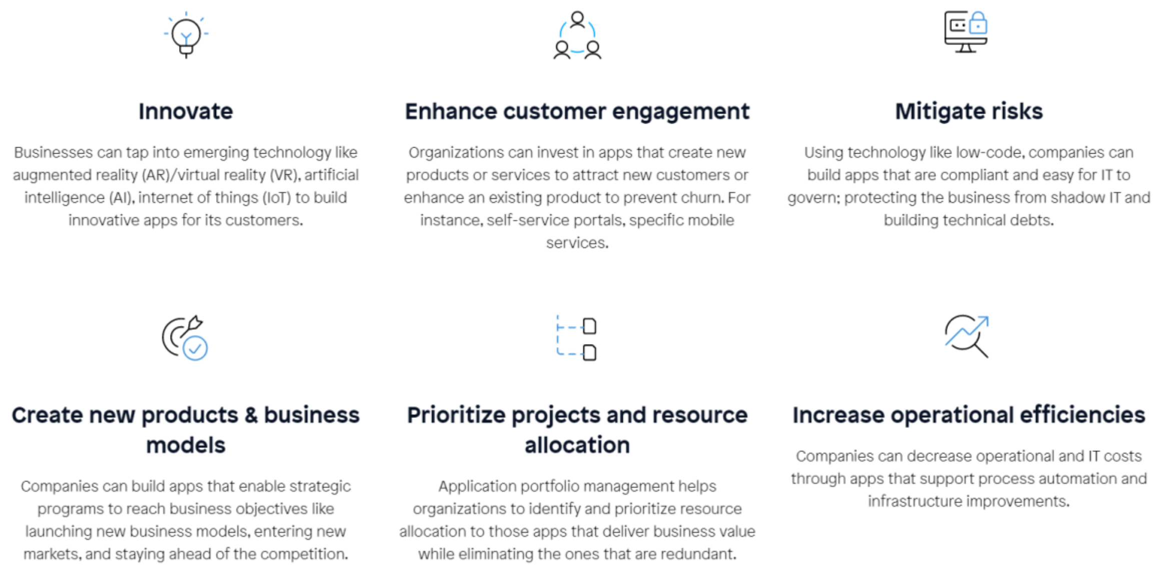
EVIDEN

12 Table of Figures

13 Appendices

13.1 Appendix I

According to Mendix, the following benefits are obtained when using APM:



Innovate

Businesses can tap into emerging technology like augmented reality (AR)/virtual reality (VR), artificial intelligence (AI), internet of things (IoT) to build innovative apps for its customers.

Enhance customer engagement

Organizations can invest in apps that create new products or services to attract new customers or enhance an existing product to prevent churn. For instance, self-service portals, specific mobile services.

Mitigate risks

Using technology like low-code, companies can build apps that are compliant and easy for IT to govern; protecting the business from shadow IT and building technical debts.

Create new products & business models

Companies can build apps that enable strategic programs to reach business objectives like launching new business models, entering new markets, and staying ahead of the competition.

Prioritize projects and resource allocation

Application portfolio management helps organizations to identify and prioritize resource allocation to those apps that deliver business value while eliminating the ones that are redundant.

Increase operational efficiencies

Companies can decrease operational and IT costs through apps that support process automation and infrastructure improvements.

Figure 38: Benefits of APM according to Mendix.

About the author

Abbas Shahim is the Global Head of Sustainability and the co-founder of the sustainability practice at Eviden. In this role, his focus is on strategic partnerships, portfolio development, integration in digital services and go-to-market approach. He is a trusted advisor to many nationals and multi-nationals and has a broad experience in a variety of sectors.

Connect with us

/in/eviden
@EvidenLive
@evidenlive
/EvidenLive

eviden.com