[In Progress...]

[Company name]

Technical Report and Proposal for Change

Created by:	TechRivo Technologies, Lda.
Date:	DD/MM/YYYY
Description:	Due diligence and Report of [abc]
Version:	0.0
Confidentiality level:	[Undetermined/Low/Medium/High]

Disclaimer

*This Technical Report and associated documentation should be interpreted as guidance to consider when taking further actions. The **Customer** should consult with its own teams and advisors regarding any matters requiring the further use of this Technical Report. Best efforts were made to ensure that the Technical Report was proofread, accurate and error-free. However, the **Customer** is responsible for the final proofreading of all documentation and TechRivo Lda is not responsible for errors, omissions, or typographical errors.*

Table of contents

1 Introduction **4**
1.1 Purpose 4
1.2 Scope 4
1.3 Background 4
2 AS-IS State **4**
2.1 Product Management 5
2.1.1 Roles and Responsibilities 5
2.1.2 Product Methodology 5
2.2 Supporting Tools 5
2.3 Software Component 5
2.3.1 Technical Documentation 6
2.3.2 Architecture 6
2.3.3 Infrastructure 7
2.3.4 Continuous Integration / Continuous Deployment 8
2.3.5 Components 8
2.3.5.1 Component 1 9
2.4 Software Development Life Cycle 9
2.4.1 Requirements Analysis 10
2.4.1.1 Requirement Types 10
2.4.1.2 Tasks 10
2.4.2 Estimating and Planning 11
2.4.2.1 Performance Evaluation 11
2.4.3 Development Cycles 11
2.4.4 Development Workflow 11
2.4.4.1 Branching Strategy 12
2.4.4.2 Deployment 12
2.4.5 Quality Assurance 12
2.4.6 Maintenance and Monitoring 12
2.5 Security Compliance 12
2.5.1 Compliance, Security and Processes MATRIX Evaluation 14
3 TO-BE State **16**
3.1 Gap/Fit Analysis 16
3.1.1 Engineering 17
3.1.2 Project Management 18
3.1.3 Security and Processes 19
3.2 Proposal of Change 20
3.3 Annexes and Supporting Documents 20

1 Introduction

[Introduction of Product/Company and the the subject of analysis/evaluation in this document]

The evaluation focus on the following aspects of the [organisation/application/product]:
(Examples:
- *Features (list of features present in the system and their current state)*
- *Tools and technologies used for product development*
- *Software component maturity*
- *SDLC maturity*
- *Availability and quality of the documentation*
- *Performance (production readiness of the system in terms of performance)*
- *User experience (identification of major UX flaws of the designs))*

The evaluation aims to answer the question about [insert main aim of the evaluation].

1.1 Purpose

The purpose of this document is to [describe purpose of the document]

(Example: The purpose of this document is to summarise and document the evaluation conducted by the [Company] team on the technical assets and processes of the [Product] as well as suggest the next steps that would lead to improvements to the technical component of the organisation).

1.2 Scope

The scope of this document is [describe scope of the document]

(Example: The scope of this document pertains to the evaluation of the technical assets and processes of [Product]. Made on the basis of resources (documentation, tools, code base, process description). In addition, the document contains a set of recommendations for the future of the technical component of [Product].

1.3 Background

This document was produced by [Document Creator], its annexes, and related documentation under the scope of the analysis of the technical assets and processes of [Product name].

2 AS-IS State

The conclusions resulting from this section are then used to develop a desired (TO-BE) state of the solution.

2.1 Product Management

2.1.1 Roles and Responsibilities

Role	Responsibility	Team member
(Ex: Product Marketing)	*(Ex: Marketing the product into the focus persons in the focus markets.)*	
+		
+		

2.1.2 Product Methodology

[Relevant information regarding the methodology]

The maturity of the implemented methodology is represented in the table below:

Layer	Phase	Phase status	Element	Element status	Notes
		Implemented/ Partially implemented/ Not Implemented		Implemented/ Partially implemented/ Not Implemented	
(Ex:Essential Layer (SAFe for teams))	*(Ex: Team and Technical Agility)*	Partially Implemented	*(Ex: Team Scrum)*	Partially Implemented	*(Ex: Planning, review and retrospective in the same meeting on a weekly basis (tech roadmap). Daily was never implemented.)*
+					

2.2 Supporting Tools

...

2.3 Software Component

This chapter of the technical report aims to analyse the software component of the solution in the pursuit of answers to the following questions:

- What is the state of the technical documentation?
- Is the system architecture fit for purpose?
- What is the condition of the infrastructure?
- How are Continuous Integration / Continuous Pipelines configured?
- What are the components of the system?
    - Are they fit for purpose?
    - What is their quality?

2.3.1 Technical Documentation

One of the key aspects of the software component is its technical documentation. Records of architecture, processes and maintenance/support procedures enable seamless operation (enhancement and keep-alive) of the system. The analysis of technical documentation of the system aims to provide a comprehensive understanding of the available records together with their level of detail. For the purpose of this analysis, there are three "Levels of details" assumed:

- HIGH (indicates a complete document with a full degree of useful information)
- MEDIUM (indicates a document with a fair amount of information that is missing some important details)
- LOW (indicates a document that is not fit for purpose)

Document	Purpose	Level of detail	Up-To-Date
[Document Title]	[Document Purpose]	[Low/ Medium/ High]	[Yes/No]
(Ex: Solution Architecture Document)	*(Ex: Outline of the system architecture aiming to support developers, DevOps engineers and project managers in product enhancement and maintenance.)*	MEDIUM	Yes
+			

The analysis of the documentation uncovered that its overall quality is [level of quality].

2.3.2 Architecture

Software architecture is a blueprint that enables the building of an efficient, scalable and, most importantly, fit-for-purpose system. Architectural decisions made at the early stages of the project very often impact the ability to adjust the system to business needs. The well-designed system is flexible enough so that it can accommodate ever-changing requirements but stable and efficient enough so it can provide a solid base for the business.

The current architecture of the system is [architecture]. It consists of x main custom-developed components:

- Component 1
- Component 2
- Component 3
- ...
- ...

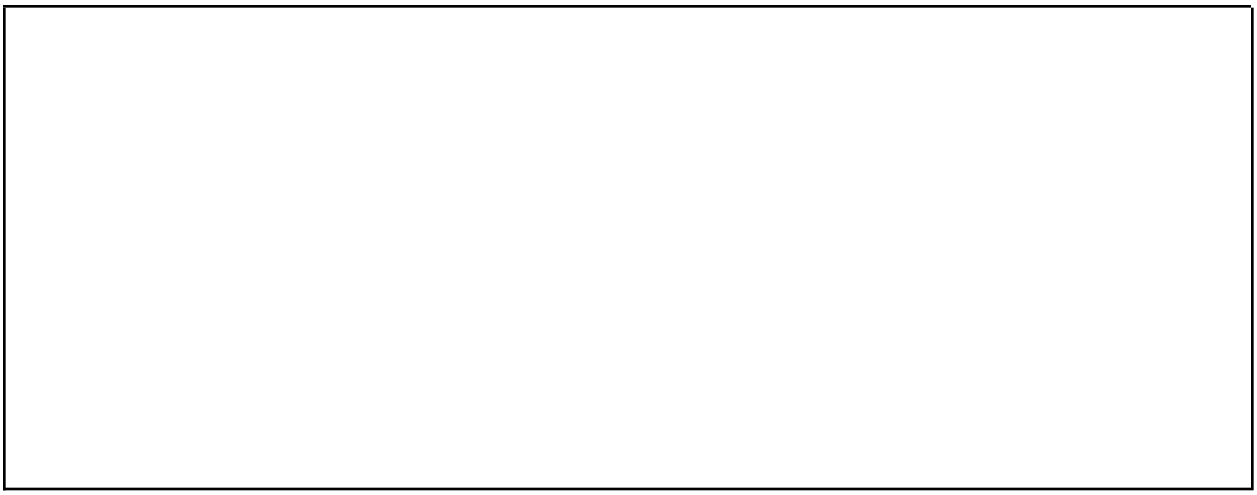
Aforementioned custom developed components are supported by the following third-party technologies:

- Tech 1
- Tech 2
- Tech 3

Third party components [...]. These components are:

- Component 1
- Component 2
- ...

The following diagram depicts all major components of the system, the way they align in the architecture and how they are deployed:



[diagram of the system]

Overall, the architecture of the system is [fit/not fit] for purpose. [Explanation of why it is fit or not fit for purpose].

2.3.3 Infrastructure

The infrastructure of the system is tightly coupled with the architecture. Similarly to system design, infrastructure choices impact its performance and ability to adapt to changes. It is essential to select appropriate infrastructure components at an early stage of development because a change to the infrastructure might be costly and time-consuming.

The current infrastructure is based on [description of current infrastructure]

Full list of technologies can be found in the table below:

Component	Description	Provider	Active
[Component name]	[Description of the Component]	[Component provider]	Yes/No
+			
+			

2.3.4 Continuous Integration / Continuous Deployment

[State of current CI/CD processes]

2.3.5 Components

The following chapters describe the identified components of the software - their purpose, focus and quality.

The overall quality of the solution is a very important aspect that has a direct impact on systems maintainability and enhanceability. Identified issues should be addressed before moving towards extensions and modifications. The simplified approach to the quality of the solution can be divided into four categories:

- **Perceived quality** - Number of business and technical issues found.
- **Code readability** - The degree to which code is understandable to external parties e.g. new coming developers.
- **Dynamic quality** - Number of issues found through automated tests created by the developers.
- **Static quality** - Grade of quality measured through third-party tools e.g. Sonar, identifying common coding antipatterns.

2.3.5.1 Component 1

Notable tools and libraries used for the development of this component are as follows:

Tool/Library	Description	Version	Current Version
[Tool/Library name]	[Tool/Library description]	[Version]	[Latest version of the Tool/Library]
+			
+			

As demonstrated in the table above the versions of key technologies of the system are kept [result of the analysis]

Perceived quality [level of the perceived quality assessed]

Code readability [level of code readability assessed]

Dynamic quality [level of dynamic quality assessed]

Static quality [level of static quality assessed]

A representative set of the major issues can be found in the table below:

Issue	Severity
[Issue title]	Blocker Critical Major Minor
+	
+	

The overall quality of the component can be assessed as [good/reasonable/bad].

2.4 Software Development Life Cycle

Software development life cycle (SDLC) is a methodology (set of processes) aiding the translation of business requirements into an enhanceable and maintainable software. Proper SDLC is essential to building software that fulfils business requirements and adds value to the product/service. Typical SDLC is divided into several areas. The following chapters describe the AS-IS state of each of the SDLC areas within [Product/Company].

2.4.1 Requirements Analysis

The analysis of requirements is a critical phase of the SDLC. When properly executed, it enables the effective translation of business requirements into software features. Requirements, at minimum, should consist of a high-level title, comprehensive description and acceptance criteria (answer to the question: what makes the requirement realised?).

2.4.1.1 Requirement Types

Within [Product/Company], there are x kinds of requirements:

Business Requirements usually originate from the business stakeholders (also known as Product Strategists). These requirements mostly define product features that will aid business or compliance procedures.

Technical Requirements usually originate from the technical stakeholders (e.g. Tech Lead). These requirements mostly define activities regarding infrastructure or technical debt elimination.

Bugs and Issues to the currently deployed software components usually originate from the after testing activities. They define issues found within the software that need to be addressed in order to reinstate the functionality required by the business.

2.4.1.2 Tasks

Regardless of the requirement type, once fully understood, they are put on a roadmap and created in the Project Management Tool in the form of Tasks. Tasks can either be used to encapsulate a single action item or as a grouping entity that includes several Subtasks. Tasks usually include:

- Title
- Description
- Due date
- Assignee
- Status
- Product
- Priority

Description field is inconsistently filled, usually containing a brief explanation of the activity (however there are notable exceptions), sometimes accompanied by a screenshot or editorial content (e.g. an email body). Acceptance criteria and/or technical details are rarely present.

Status field, representing the current state of the task in the development cycle. The status field can take the following values:

- In Progress
- Additional Input Required
- Deferred
- In Code Review
- Testing
- Ready To Deploy
- Deployed Test Env
- Test Failed
- Deployed Prod Env
- Done
- On-Hold

- Not Started

Product field, indicating a logical area to which this task is assigned, can take the following values:
- CMS
- API
- WebApp
- MobileApp
- Website
- Infrastructure
- Sharepoint / Mail boxes
- New developments
- Web and Mobile App

Tasks are the basic building blocks of Estimating and Planning and Development Cycles.

[Create diagram to depict life cycle of Task]

2.4.2 Estimating and Planning

Estimating and planning are critical phases from the perspective of project control and expectation management. Proper planning is a prerequisite for roadmap creation and business enhancement.

Currently, estimating activities happen on a weekly basis during a synchronisation meeting between the Product Manager and the team (represented by the Tech Lead). The outcome of an estimation session is a set of due dates for specific tasks. The efforts are not denoted in a measurable form (e.g. hours or story points).

Planning activities are performed (on a high level) for Product Increments, and (low-level) Development Cycles are explained in the following chapter. The activities usually take the following aspect into account:
- Business priorities
- Technical priorities
- Collected estimates

2.4.2.1 Performance Evaluation

Development performance evaluation is a key to understanding if the SDLC is efficient and effective. The evaluation is usually achieved through tracking various metrics, among which time estimated vs time elapsed is one of the most basic metrics.

2.4.3 Development Cycles

Development cycles bound software creation efforts into measurable and foreseeable timeframes. Having such cycles allows for putting the focus on the items of highest priority and understanding the progress. Usually, it should be possible to complete tasks, planned for a given cycle, within the boundaries of a single cycle.

Development efforts within [Product/Company] are [Describe development cycles]

2.4.4 Development Workflow

[Create a diagram with workflow]

2.4.4.1 Branching Strategy

The development team follows a [describe the branching strategy]

2.4.4.2 Deployment

The deployment of the solution to the environment is divided into x phases:

[Phase 1] Deployment - [description of deployment phase]

[Phase 2] Deployment - [description of deployment phase]

[Phase 3] Deployment - [description of deployment phase]

2.4.5 Quality Assurance

The quality Assurance phase of the SDLC is [Description of Quality Assurance process]

2.4.6 Maintenance and Monitoring

[Description of maintenance and monitoring activities]

2.5 Security Compliance

This Section covers the current state of the Internal Processes and Policies. It serves to evaluate the maturity of technical processes only.
Although some business processes are also taken into account, their scope is strictly observed under the context of supporting technical implementations.

The goal of this AS-IS evaluation is to identify the gaps in security and technical processes based on well-known and adopted frameworks and certification processes.

The goal is to simply understand the effort needed to implement a remediation process. This is not an audit or an exhaustive assessment and should be seen as a broad indication of effort only.

Technical Assessment Matrix Legend:

N/A	Not applicable to [Product/Company] or out of the scope of this report
-	Currently not available in [Product/Company], might be required
⚫	Present and fully covers the framework or requirement
⚪	Present but does not fully covers the framework or requirement

Methodology Assessment Levels		
0	Inexistent	No evidence exists of a process in place
1	Initial	Processes are seen as unpredictable, poorly controlled, and reactive. Businesses in this stage have an unpredictable environment that leads to increased risks and inefficiency.
2	Managed	Processes are characterised by projects and are frequently reactive.
3	Defined	Processes are well-characterised and well-understood. The organisation is more proactive than reactive, and there are organisation-wide standards that provide guidance.
4	Quantitatively Managed	Processes are measured and controlled. The organisation is using quantitative data to implement predictable processes that meet organisational goals.
5	Optimising	Processes are stable and flexible. The organisational focus is on continued improvement and responding to changes.

Stakeholder relevance	
Blue	Business stakeholders were involved in answering the relevant section
Green	Technical stakeholders were involved in answering the relevant section

2.5.1 Compliance, Security and Processes MATRIX Evaluation

Assessments / Appraisal Guidance				Notes
... (Technical Domain)				
1. Security				
System Access Controls				
...				
2. Availability				
Incident Report Planning				
...				
3. Processing Integrity				
GDPR Compliance				
...				
4. Confidentiality				
Levels of Protected Information				
...				
5. Privacy				
Sector / Industry Privacy Rules				
...				
Non-Mandatory but relevant to the entity				
...				
Project Management Processes				

Project Planning				
...				
Project Monitoring and Control				
...				
Integrated project Management				
...				
Risk management				
...				
Engineering Processes				
Requirements Management				
...				
Requirements Development				
...				
Technical Solution				
...				
Product Integration				
...				
Verification				
...				
Validation				
...				
Process Management				
Organisational Process Focus				
...				

Organisational Process Definition				
...				
Support Processes				
Configuration Management				
...				
Processes and Product Quality Assurance				
...				
Measurement and Analysis				
...				
Decision Analysis and Resolution				
...				

The Matrix above allows us to establish the current maturity level of [Product/Company] as being [Low/Medium/High].

[Explanation for the established maturity level].

How are the technical processes integrated and aligned with the business processes and the organisation

[Alignment description]

3 TO-BE State

3.1 Gap/Fit Analysis

The following chapters describe Gaps and Fits of the current processes and technical components based on the analysis of the AS-IS state. The tables and paragraphs visualise areas that are working well and the ones that require alteration. The following chapters assume the following colour coding:

- Red gaps - areas where current and desired states are misaligned and require urgent modifications

- Yellow gaps/fits - areas where current and desired states are either partially consistent or filling of the gaps is not critical
- Green fits - areas where there is a fit between current and desired state, thus does not require any changes

3.1.1 Engineering

The following table represents gaps and fits in the engineering (technical) processes and components.

Category	Item/Process	Current State	Desired State	Gap/Fit Level
Infrastructure	**Infrastructure ownership**			
	Deployment strategy			
	Standardised infrastructure deployment			
	Storage of keys and certificates			
	Deployment abstraction			
	Environments			
Development Workflow	**Code workflow**			
	CI/CD pipelines			
	Automatic deployment			
	Team engagement			
	Releases			
Quality Assurance	**Unit tests**			
	Integration tests			
	Static code analysis			
	Vulnerability checks			
	Code reviews			
	Manual testing			
	Perceived quality			

	Static code quality			
	Dynamic code quality			
	Code readability			
Maintenance and Monitoring	Infrastructure status monitoring			
	Infrastructure cost monitoring			
	Performance monitoring			
	Back-end debugging			
	Front-end debugging			
Source Code	Code ownership			
	Version control			
	Domain Driven Design			
	Tools			
	System design			
	Buy instead of build			
	Technological stack			
Documentation	Documentation			

3.1.2 Project Management

The following table represents gaps and fits in the project management processes.

Category	Item/Process	Current State	Desired State	Gap/Fit Level
Requirement Analysis	Project management tool			
	Requirement types			
	Tasks			
	Workflow			

Estimating and Planning	**Involved stakeholders**			
	Decision base			
	Unit of measure			
	Performance tracking			
Development Cycles	**Development cycles**			
	Roadmap			
	Workload			

3.1.3 Security and Processes

Domain / Control	Item/Process	Current State	Desired State (Implementation and Controls)	Gap/Fit Level
Information Security Management	**System Accesses and Controls**			
	Preventive Measures (Intrusion Detections, Anti-Virus, Anti-Malware, Extortion policies, overall Cyber Security, Firewalls, Fraud, Security threats measures)			
	Cryptography			
	Information Management			
	Environmental Security			
	Technical HR Security			
Operations Management	**Assets Management**			
	Access Control			
	Communications			
Risk Management	**Incident, Risk Management and Reporting**			
	Disaster Recovery and Business Continuity			
	Compliance, Confidentiality and Auditing			

Project and Product	**Product and Project budgeting, estimation, monitoring and control**			
	Engineering Process and Procedures			

3.2 Proposal of Change

[Small conclusion of analysis conducted before details of the areas to approach]

The [Product/Company] could benefit from changing its approach in the following key areas:

(Examples:
1) *Engineering and Software Development*
2) *Product and Project Management*
3) *IT and Security Processes)*

3.3 Annexes and Supporting Documents

[List of all annexes and supporting documents]