IT Delivery Risk Assessment Framework

Sara Maria Esteves Paraty

Master's Dissertation

Supervisor at FEUP: Prof. Dulce Lopes
Supervisor at Deloitte: Eng. João Ferreira



Master in Mechanical Engineering:
Production Engineering & Management

2022-06-30

Abstract

Today's firms strive to maintain a market competitive advantage by comprehending and embracing the need for digital transformation in order to achieve digital maturity and prosper in the Fourth Industrial Revolution. Deloitte is a leading global consulting firm that contributes to the acceleration of this transition by developing customized customer solutions.

Despite the active digitization of the economy and the increasing number of software projects and assigned funds, only a small percentage of these projects is finished according to the specifications that were first established. The majority of project decisions are made based on managers' intuition and professional expertise, prompting the establishment of a data-driven solution for risk assessment to evaluate software development execution quality.

This thesis scope tackles this topic by standardizing risk estimation throughout the project life cycle, enabling a risk-driven strategy, fostering clear communication and alignment within the team. Furthermore, it strives to support the completion of all projects within scope, on schedule, and within budget.

The fundamental axes of this problem-solving include a comprehensive project flow analysis to determine the main and cross stages to be highlighted in the framework, the creation of reliable and rigorous metrics to evaluate the risk during each stage, and, finally, the establishment of assessment criteria and the percentile impact of each metric on the final risk score.

The greatest challenge of this project has been implementing the theoretical model proposed into practice, through the automation of Power BI dashboards. So, this application is made available to teams in order to assess the prevailing risk of projects in development, avoiding further unexpected setbacks. This solution does not entail additional resources and allows for continual monitoring of risk at all stages. The developed Model Validation Pilot based on effective indicators and uniform across projects, displayed in a user-friendly interface, enables the foresight of issues, hence prompting sustained decision-making in a timely manner.

Resumo

Nos dias de hoje, as empresas asseguram uma posição competitiva no mercado através da compreensão e aceitação da magnitude da transformação digital, a fim de atingir a maturidade tecnológica e prosperar na Quarta Revolução Industrial. A Deloitte é uma empresa de consultoria líder mundial que contribui para a aceleração desta transição através do desenvolvimento de soluções personalizadas para os seus clientes.

Apesar da aceleração da digitalização da economia e do número crescente de projectos de *software*, bem como dos investimentos associados, apenas uma pequena percentagem desses projectos é concluída respeitando todas as especificações previamente estabelecidas em fases iniciais. A maioria das decisões relativas a cada projecto são tomadas com base na intuição e experiência dos gestores dos mesmos, o que destaca a importância do desenvolvimento de uma solução sólida baseada em dados capazes de estimar o risco e, assim, prever a qualidade de execução ao longo do desenvolvimento de *software*.

Neste âmbito, a presente tese de mestrado visa o desenvolvimento de uma estratégia de gestão dos projetos uniformizada e orientada para a estimativa do risco ao longo do ciclo de vida do projeto, promovendo a comunicação e consciencialização dos fatores de risco com maior impacto dentro das equipas. Deste modo, é proporcionada a garantia de concretização de todas as funcionalidades previstas no projeto nos prazos inicialmente definidos e com os orçamentos anteriormente estipulados.

Os eixos fundamentais da abordagem a este desfio passam por uma análise detalhada do fluxo de trabalho dos projetos para determinar as fases transversais e com maior relevância a destacar na *framework*, pela definição de métricas eficazes e rigorosas para avaliar o risco durante cada uma das fases e, por último, pela definição de critérios de avaliação e do impacto percentual de cada métrica na pontuação final do risco.

O maior desafio durante a realização deste projeto foi a implementação prática do modelo teórico proposto, através da sua automatização e criação de *dashboards* em Power BI. Desta forma, a aplicação desenvolvida está disponível para todas as equipas que pretendam avaliar o nível de risco atual dos projetos aos quais estão alocadas, por forma a evitar níveis de risco elevados e imprevistos na fase final do projeto. A solução sugerida não implica recursos financeiros adicionais e proporciona a monitorização contínua do risco ao longo da vida de um projeto. O piloto de validação do modelo desenvolvido, baseado num conjunto de indicadores eficientes e visualizado através de uma interface bastante intutiva, permite antecipar a tomada de decisões, prevenindo consequências onerosas no futuro.

Acknowledgments

I would like to begin by expressing my pleasure to Deloitte Porto for the chance of joining their team and delving into the exciting world of technology consulting.

I want to thank my Deloitte supervisor, João Ferreira, for his unmatched perspicacity and straightforwardness while monitoring me; Ruben Peixoto for placing this real challenge in my hands and who is always prompt to help me; and Manuel Almeida for his endless support and for always finding time for my countless questions.

I would like to convey my gratitude to Prof. Dulce Lopes, my supervisor at FEUP, for her great counseling and all the valuable insights throughout the project.

Thank you from the bottom of my heart to my mother, brother, grandparents, godmother and the rest of my family for their unwavering love, unbounded encouragement, and motivation, and for ridiculously believing in me. I could never thank you enough.

I wish to express my sincere gratitude to my father, whose example has become an inspiration for my entire life and who I hope is truly proud of me, even though he cannot celebrate this achievement with me.

Last but not least, I want to convey my appreciation to my friends, with whom I have shared this epic journey for the past 5 years, and for being the greatest pleasure that university has given me.

Thank you all for making this path mesmerizing.

"In the future, talent, more than capital, will represent the critical factor of production."

Klaus Schwab [1]

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Deloitte and the Project . . . 2
- 1.3 Main objectives . . . 3
- 1.4 Methodology . . . 4
- 1.5 Dissertation Structure . . . 5

2 Literature Review **7**
- 2.1 Software delivery approaches . . . 7
  - 2.1.1 Waterfall Approach . . . 7
  - 2.1.2 Agile Approach . . . 8
  - 2.1.3 Hybrid Approach . . . 10
- 2.2 Software Testing . . . 10
  - 2.2.1 Levels of Abstraction . . . 10
  - 2.2.2 Levels of Testing . . . 11
  - 2.2.3 SDLC Testing . . . 11
- 2.3 Quality Assurance . . . 12
  - 2.3.1 ISO/IEC 25000 . . . 13
  - 2.3.2 Quality Model . . . 13
- 2.4 Risk Assessment . . . 14
  - 2.4.1 Issues Management . . . 15
  - 2.4.2 Risk Scoring . . . 15
  - 2.4.3 Risk Indicators . . . 16
  - 2.4.4 Software Metrics . . . 17
- 2.5 Conclusion . . . 18

3 Description **19**
- 3.1 ISO 9001 . . . 19
- 3.2 Process Flow . . . 20
  - 3.2.1 Inception . . . 20
  - 3.2.2 Development . . . 20
  - 3.2.3 Testing . . . 21
  - 3.2.4 Release . . . 22
- 3.3 Problem Identification . . . 23

4 Risk Framework **25**
- 4.1 Structure . . . 25
- 4.2 Score System . . . 26
  - 4.2.1 Output . . . 26

4.2.2 Dimensions Weighting . . . 27
4.2.3 Metrics . . . 28
4.3 Team Structure Matrix . . . 29
4.4 Project Metrics . . . 30
4.5 Inception Metrics . . . 33
4.6 Development Metrics . . . 34
4.7 CAT Metrics . . . 36
4.8 SIT Metrics . . . 38
4.9 UAT Metrics . . . 41
4.10 Risk Evolution Benchmark . . . 43
4.11 Conclusion . . . 44

5 Model Validation Pilot **45**
5.1 Functional Requirements . . . 45
5.2 IT Tool Implementation . . . 46
5.3 Power BI Application . . . 46
5.3.1 Sequence Diagram . . . 46
5.3.2 Data Model . . . 47
5.3.3 Report . . . 47
5.4 Results . . . 49
5.4.1 Project 1 . . . 49
5.4.2 Project 2 . . . 51

6 Conclusion and Future Work **53**
6.1 Conclusion . . . 53
6.2 Future Work . . . 54

A Power BI Data Model: UML Diagram **59**

B Power BI Report **61**
B.1 Project 1 . . . 61
B.2 Project 2 . . . 67

Acronyms and Symbols

CAT	Component Acceptance Testing
CBO	Core Business Operations
FS	Financial Services
FURPS	Functionality, Usability, Reliability, Performance and Supportability
IEC	International Electrotechnical Commission
IS	Industry Solutions
ISO	International Organization for Standardization
IT	Information Technology
KPI	Key Performance Indicator
MS	Microsoft
MVP	Model Validation Pilot
RC	Responsibility Center
SDLC	Software Development Life Cycle
SIT	System Integration Testing
SQA	Software Quality Assurance
SQuaRE	Software Product Quality, Requirements and Evaluation
TC	Technology Consulting
Th	Threat Level
UAT	User Acceptance Testing
u.s.	User Story

List of Figures

1.1 Statistics of IT projects success 2
1.2 Planning: Gantt Chart 5

2.1 Waterfall Framework 8
2.2 Agile Scrum Framework 9
2.3 Cross Discipline Iterative Development 9
2.4 Waterfall V-Model: Testing Phases 12
2.5 Project Management Triangle 13
2.6 ISO/IEC 25000 13
2.7 Generalized Risk Assessment Model 15
2.8 Risk Scoring Matrix 15

3.1 Execution Rate VS Pass Rate 22

4.1 Risk Assessment Framework Structure 26
4.2 Risk Score Scale 27
4.3 Risk Evolution Benchmark 44

5.1 IT Solution Scheme 46
5.2 Sequence Diagram 47

A.1 Power BI Data Model: Data Tables (1) 59
A.2 Power BI Data Model: UML Diagram - Data Tables (2) 60

B.1 Power BI App: Home Page 61
B.2 Project 1: Project Analysis 62
B.3 Project 1: Project Dimension 62
B.4 Project 1: Inception Dimension 63
B.5 Power BI Report: Mousing Over Functionality 63
B.6 Project 1: Development Dimension 64
B.7 Project 1: CAT Dimension 64
B.8 Project 1: SIT Dimension 65
B.9 Project 1: UAT Dimension 65
B.10 Project 1: Closed Project Feedback 66
B.11 Power BI App: Projects Overview 66
B.12 Power BI App: Risk Evolution 67
B.13 Project 2: Project Analysis 67
B.14 Project 2: Project Dimension 68
B.15 Project 2: Inception Dimension 68

B.16 Project 2: Development Dimension 69
B.17 Project 2: CAT Dimension 69

List of Tables

2.1 Quality Model ISO/IEC 25000 14
2.2 Risk Indicators 16

3.1 User Stories Effort Estimation 21

4.1 Score System 27
4.2 Team Structure Matrix: Deviation Analysis 29
4.3 Ideal Teams Structure 29
4.4 Project Metric 1: Type of Service 30
4.5 Project Metric 2: Type of Project 30
4.6 Project Metric 3: Effort Estimation 30
4.7 Project Metric 4: Deloitte Project Team Experience in Similar Projects 31
4.8 Project Metric 5: Project Duration 31
4.9 Project Metric 6: Project Location 31
4.10 Project Metric 7: Customer's C-level Sponsorship 31
4.11 Project Metric 8: Other Parties Involved 32
4.12 Project Metric 9: Customer Experience with the Chosen Delivery Approach . . . 32
4.13 Project Metric 10: IT Projects Client Preparation 32
4.14 Project Metric 11: Scope Complexity 33
4.15 Inception Metric 1: Team Structure 33
4.16 Inception Metric 2: Stage Duration 33
4.17 Inception Metric 3: Client Availability for Workshops 33
4.18 Inception Metric 4: Customer Sign-off on All Functional Designs 34
4.19 Inception Metric 5: Delivery Delay 34
4.20 Development Metric 1: Team Structure 34
4.21 Development Metric 2: Stage Duration 35
4.22 Development Metric 3: Number of User Stories/Requirements 35
4.23 Development Metric 4: All Tools Availability 35
4.24 Development Metric 5: Development Environment Availability 35
4.25 Development Metric 6: Amount of Delivered Scope 36
4.26 Development Metric 7: Delivery Delay 36
4.27 CAT Metric 1: Team Structure 36
4.28 CAT Metric 2: Stage Duration 36
4.29 CAT Metric 3: Number of Tests 37
4.30 CAT Metric 4: Testing Environment Availability 37
4.31 CAT Metric 5: Percentage of Automatic Tests 37
4.32 CAT Metric 6: Execution Rate 37
4.33 CAT Metric 7: Pass Rate 38

4.34 CAT Metric 8: Percentage of Critical Defects 38
4.35 CAT Metric 9: Delivery Delay 38
4.36 SIT Metric 1: Team Structure 38
4.37 SIT Metric 2: Stage Duration 39
4.38 SIT Metric 3: Number of Tests 39
4.39 SIT Metric 4: Testing Environment Availability 39
4.40 SIT Metric 5: Percentage of Automatic Tests 39
4.41 SIT Metric 6: Execution Rate 39
4.42 SIT Metric 7: Pass Rate 40
4.43 SIT Metric 8: Percentage of Critical Defects 40
4.44 SIT Metric 9: Number of Integrated Systems 40
4.45 SIT Metric 10: Percentage of Successful CAT Tests 40
4.46 SIT Metric 11: Delivery Delay 40
4.47 UAT Metric 1: Customer Support From Deloitte Team 41
4.48 UAT Metric 2: Stage Duration 41
4.49 UAT Metric 3: Customer Availability to Perform Tests Plan 41
4.50 UAT Metric 4: Testing Environment Availability 41
4.51 UAT Metric 5: Percentage of Automatic Tests 42
4.52 UAT Metric 6: Execution Rate 42
4.53 UAT Metric 7: Pass Rate 42
4.54 UAT Metric 8: Percentage of Critical Defects 42
4.55 UAT Metric 9: Percentage of Successful CAT Tests 42
4.56 UAT Metric 10: Percentage of Successful SIT Tests 43
4.57 UAT Metric 11: Number of Business Areas Testing 43
4.58 UAT Metric 12: Customer Testing Experience 43
4.59 UAT Metric 13: Delivery Delay 43

Chapter 1

Introduction

1.1 Context and Motivation

The First Industrial Revolution was marked by a shift from hand production to machines using water and steam for propulsion. The Second one was defined by the introduction of electric power to mass production and lastly, the Third one came up with electronics and information technology to automate production. [1]

Now, we stand on the brink of the Fourth Industrial Revolution that conceptualizes a fusion of technologies characterized by blurring the lines between the physical, digital, and biological realms, resulting in a breakthrough at an exponential pace unlike any other in history. It has been disrupting almost every industry, transforming entire production, management, and governance systems. Emerging cyber-physical systems in fields such as Artificial Intelligence, Cloud, Internet of Things, systems integration, Big Data, and others, must be integrated into the response. [1]

The world is getting faster every day. Today's business environment demand is for more value, higher customer satisfaction, shorter time frames, and lower costs. Even though the active digitization of the economy and the rise in both the number of software projects and allocated budgets, only 31% of software development projects are successfully completed regarding the initial set up requirements (scope, functionalities, budget, time frame, and so forth), as reflected in Figure 1.1. [2] According to [3], the most important determinants of project success are effective project management (13.9%) and clear planning (9.6%). The bulk of project decisions is made on the basis of managers' intuition and professional experience, prompting the need of establishing a data-driven solution for risk assessment. [3]

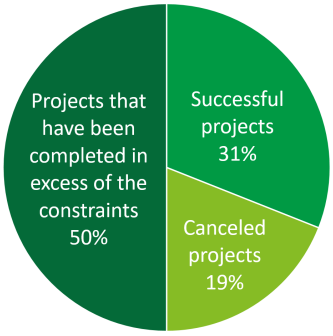


Figure 1.1: Statistics of IT projects success [2]

"A simple definition of risk is a problem that could cause some loss or threaten the success of our project, but which has not happened yet. These potential problems might have an adverse impact on the cost, schedule, or technical success of the project, the quality of our software products, or project team morale." [4] In a nutshell, technology is changing at a breakneck pace, the marketplace is becoming more complex, and there is a global shortage of skilled IT (Information Technology) personnel to meet the high demand, all of which pose unprecedented risks. Nevertheless, risk assessment is still frequently neglected in practice. There is a relatively small number of works in the literature about the process of risk measurement in project environments and thereby, there is a gap in this subject within IT and software project management. [5]

Deloitte creates tailored solutions for customers across all industries for them to sustain a competitive advantage in the marketplace by first understanding each client's specific business requirements. Most of these solutions to achieve digital maturity are concerned with cutting-edge techniques for converting, upgrading, and running existing technological platforms, or with the implementation of new technology. These developments aim to improve product and service quality, boost financial performance, and shorten the time to market, all of which will help the company prosper in the Fourth Industrial Revolution. [6]

This thesis subject was prompted by the accelerating transformation of solutions regarding technological initiatives, as well as their rising complexity. This way, it is evident that the risk inherent in these projects increases in tandem with their complexity, resulting in the need to continuously track the risk level throughout the Software Development Life Cycle (SDLC).

1.2 Deloitte and the Project

Deloitte is a leading global consulting firm that is part of the Big Four, the group of the world's largest professional services firms. There are over 700 offices in more than 150 countries around the world, with a total headcount of nearly 350.000 employees, generating aggregate global revenue of $50.2B in 2021, with the consulting sector accounting for 41% of the revenue. [6]

Deloitte has extensive knowledge and experience concerning consulting projects across all segments of Financial Services, including Banks, Insurers, Fund Managers, and Pension Funds, ranging from strategy development to implementation using the latest technologies. This thesis

is prepared for the Technology Consulting (TC) division of Deloitte Portugal's Financial Services (FS) business unit, Core Business Operations (CBO) unit, and Industry Solutions (IS) responsibility center.

As stated in the preceding section, the increasingly competitive world has been prompting the emergence of more technology implementation projects and more complex approaches to software development. Given that risk management is a key success determinant for delivering successful projects, and since Deloitte manages multiple projects concurrently, optimal resource distribution and utilization is essential, emphasizing the intention of this thesis. [7]

For Deloitte, being at the forefront of promising markets is paramount, as it warrants the timely conditions for greater created value. Indeed, the company has vast experience in implementing several projects, especially in Banking and Insurance areas, allowing to start from a large database on multiple parameters and uncover patterns pertaining to project risk.

1.3 Main objectives

The main scope of this master's thesis proposal is to create a data-driven tool to aid in the project management and decision-making of software implementation initiatives. More specifically, it seeks to analyze historical data from past projects and provide the company with an IT Delivery Risk Assessment Framework, striving to meet the demand for a tool to aid in project risk management using an accurate framework and, by extension, to:

- Develop an efficient algorithm for assessing projects risk throughout its implementation;
- Assist decision-making, which is now mostly dependent on the intuition and past experience of managers;
- Provide risk visibility within the teams;
- Permit senior members of Deloitte to get an overview of the existing risk in the numerous ongoing projects;
- Promote a stronger alignment between the project manager of Deloitte and the customers so that they have a report foundation for eventual mitigation steps in order to better comprehend the workflow;
- Lastly, provide standard data to figure out risk fluctuation patterns.

Based on parameters from previous core projects for Banks and Insurers, it is supposed to distinguish trends in order to develop a framework that can be applied to different types of projects and whose goal is to deliver the risk score during the SDLC attending to the most relevant parameters at each stage.

An algorithm to automate the framework in PowerBI allowing live data flow and projects risk follow-up was developed. This formula must work for any tech consulting project in Financial Services.

This risk-based strategy focus on promoting on-time, within-budget, and completed-scope project deliverables by assigning project quality bottlenecks, which is a firm priority. Therefore, conclusions regarding the identification of the highest risk level areas and the indication of potential mitigation strategies to lower project risk must be reached.

1.4 Methodology

Initial efforts consisted in getting to know the contextualization of the proposed project, as a means to align overall expectations and objectives. Concurrently, stakeholders gave me insights about consulting projects' workflow.

Extensive research was conducted to provide the groundwork for the literature review, where current methodologies for approaching and testing software to be implemented were deeply analyzed. With that in mind, further study to better comprehend quality assurance and risk assessment concepts related to technology projects was carried out.

Furthermore, contextualization of the actual software development phases in Deloitte projects was carried out to gain a better understanding of the process flow, as well as related constraints and prospects to improve performance and to gain risk visibility throughout the SDLC. From this perspective, the most relevant metrics to bear in mind in each stage were gathered.

The development of the framework began with the identification of the pertinent phases to be emphasized, followed by the structuring of its logic and sequence flow to be filled throughout the life cycle of the project. Numerous meetings with stakeholders were held with the objective of identifying the relevant metrics to be considered at each phase, combining them to develop the algorithm, and establishing the criteria for assessing each indicator.

The framework was afterward automated. Microsoft Forms was used to collect the necessary data, while an Excel file was used to model those responses. Finally, a Power BI report was developed to visualize and track risk dashboards.

To end this work, a Model Validation Pilot was conducted to validate the framework's effectiveness in actual projects and for further usage by teams. For the sake of defining the project execution planning, the stages and accompanying time frames were stipulated, as shown in the Gantt Chart in Figure 1.2.

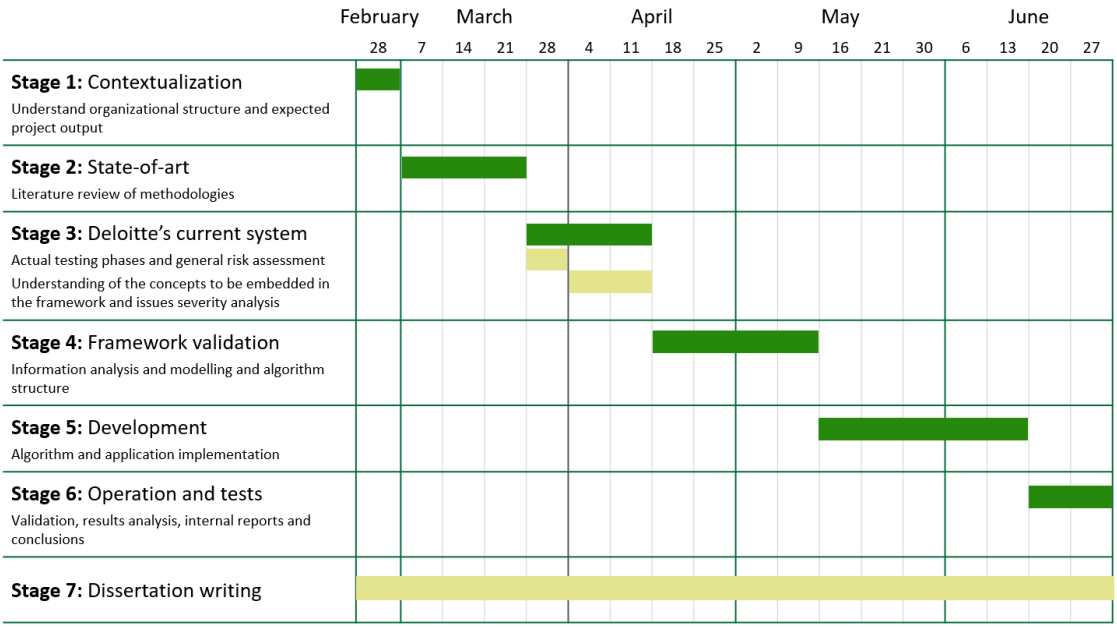


Figure 1.2: Planning: Gantt Chart

1.5 Dissertation Structure

The current chapter presents the project in regards to the firm, the project's motivation and main objectives, as well as the methodology for attaining them. The second chapter covers the literature review on the following topics: software delivery approaches, software testing, quality assurance, risk assessment, and software metrics. In chapter 3, the current situation at Deloitte is given to facilitate comprehension of the subsequent work. The fourth chapter includes all the framework logic, structure, and score system to deliver the risk score. Relevant metrics for risk measurement at each stage are presented and weighted. Chapter 5 depicts the final results of the automated Power BI application and the major steps in order to render it possible. Furthermore, it provides a results analysis based on the framework's applicability to actual projects to prove its reliability. In conclusion, the last chapter is dedicated to closing notes and future work.

Chapter 2

Literature Review

The software development process can be thought of as a set of tasks that must be completed in order to produce high-quality software. According to the literature [8], the quality of the software generated is a reflection of how the process was carried out.

This chapter offers a Software Development Life Cycle bibliography search (SDLC). In the first instance, various software project delivery frameworks are reviewed. The attention then shifts to software testing, which is a cornerstone in its delivery. In addition, bibliographical analysis regarding quality assurance, risk assessment, and software metrics that are used for warning risk is presented.

No comparable framework to the one envisaged for this project was found in the literature, hence this chapter focuses on its theoretical principles and a thorough examination of the workflow execution quality for further structure.

2.1 Software delivery approaches

Before beginning any project, management teams must outline how the workflow will be structured, the resources that will be assigned, how the process will be tracked among other things.

Although there are different software development frameworks, some cross-sectional phases assisting in the production of software stand out. Those generic phases are: Specification (requirements and constraints gathering); Development (design and implementation); Validation (testing) and Evolution (operation and maintenance). [8]

2.1.1 Waterfall Approach

The traditional approach to software development is the Waterfall model which takes a linear approach, relying on specific functional requirements up front and using the outcomes of each stage as the foundation for the next. The progress is regarded as flowing increasingly downwards through a sequential workflow where it is only possible to move to the next phase when the previous one has been completed. [9]

This model is known for segmenting projects into five main activities. The scheme in Figure 2.1 depicts those phases and summarizes their objectives.

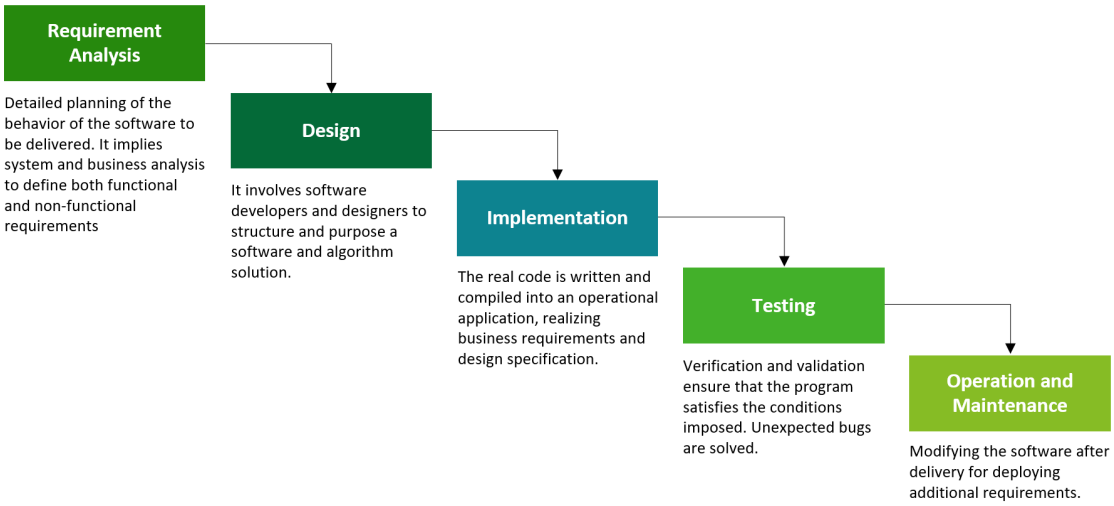


Figure 2.1: Waterfall Framework [10]

The main concern regarding this approach remains on the identification of defects only at the end of the entire process, making it harder to solve them because the whole process must be repeated. It is not easy to maintain a Waterfall-based software up to date due to the lack of integration between components, making it expensive, and amendments due to adjustments in requirement specifications during the implementation are difficult to achieve. These illustrate the shortcomings of the traditional approach. [10]

2.1.2 Agile Approach

Agile is an interactive (successively refined) and incremental (delivered in pieces) development approach that emphasizes flexibility in an integrated environment involving the customer in a technology landscape with the focus of delivering value quickly and providing discipline and transparency to achieve the desired result. Thus, this methodology provides the ability to manage complexity, unpredictability, and change through prioritization, continuous inspection, and adaptation.

Scrum is the most common Agile framework that provides steps to manage and control the development process. The workflow to achieve solutions that meet customer satisfaction involves collaboration between cross-functional teams. The builds are successive and incremental in terms of the features to develop object-oriented software. [11]

Figure 2.2 depicts the cycle of delivering value on a regular basis that starts immediately after the project planning, highlighting the core elements of a Scrum project.

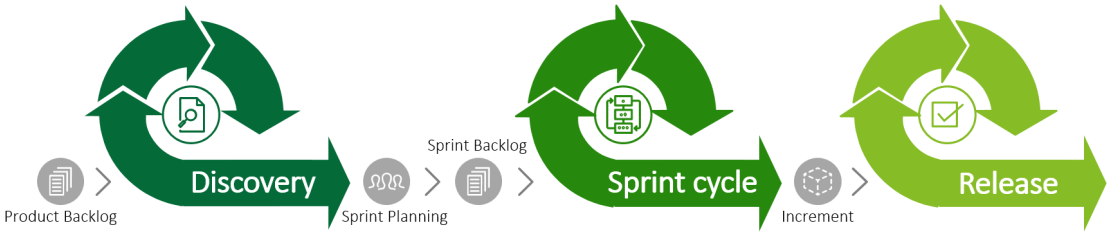


Figure 2.2: Agile Scrum Framework

A sprint is the "heartbeat" of a scrum, with a small team working on assigned tasks to bring the product closer to the product goal. Sprints range from 2 to 4 weeks in duration, with the shorter the better. The milestone of each sprint is a product increment. A sprint backlog is documentation including all the tasks that must be completed in a specific sprint. The product backlog is a roster of requirements called user stories which are determined by the product owner.

With attention to that documentation, each sprint is planned by the entire team. Changes in customers' needs can be taken into consideration during discovery stage where details can be refined.

The combination of product increments delivered over several sprints becomes the solution or the desired outcome of the project. The release cycle aims to test and deploy the developed solution prior to operation and to provide subsequent support. [11]

The gain in efficiency and quality is confirmed during sprint cycles through frequent feedback, process control, and improvement loops. The quality focus, frequent delivery of value, relatively short feedback loop, and quick response to changes in requirements are the major strengths of this strategy. It also denoted the early validation of whether or not the delivered increment meets the requirements and making the necessary corrections with minimal risk and cost as leverage of this SDLC. [8]

The figure below clearly demonstrates the cross-disciplinary iterations during the development process to ensure incremental business value delivery.

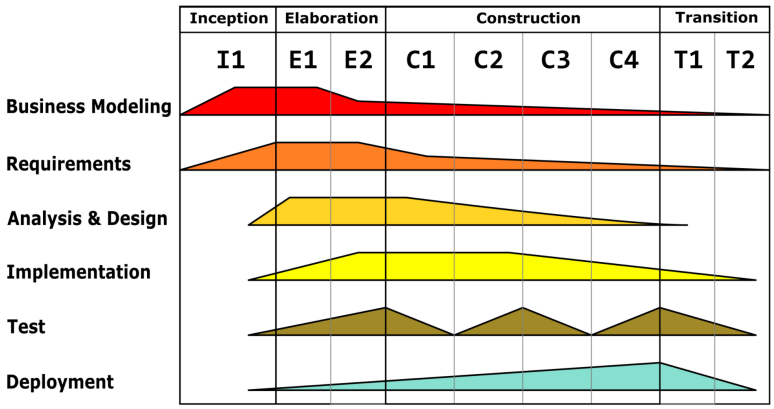


Figure 2.3: Cross-Discipline Iterative Development [12]

2.1.3 Hybrid Approach

Although Agile is sweeping the software business, one can not deny a few strengths of Waterfall. In this approach, the best features of both models Waterfall and Agile will be blended to deal with the rising complexity of handling client requirements and complex projects. This combination retains the demanded flexibility from the Agile model while also keeping the systematic approach from the Waterfall model. It is particularly useful in cases where the products under investigation lack prior information or when are characterized by large variability. [8]

"Water-scrum-fall", as this combination is known, seeks an iterative implementation where development and testing teams adopt Agile methods in order to guarantee strict means for quality assurance, to take a customer-focused approach, and to increase the speed while attempting to reduce time and costs related with issues correction. On the other hand, the Waterfall methodology is used during the planning, requirements analysis, and integration testing (a further testing phase that will be explained in Section 2.2) for the sake of clear traceability of the project. Likewise, budgeting and documenting the project's progress are done in the traditional way to provide a more rigid plan-driven development and more accurate information for decision-making when allocating funds. [13]

To summarize, the adoption of both models typically occurs in long-term projects where fast adaptation to changing requirements, early identification of risks and issues, and continuous customer feedback are needed in combination with transparency and tracking accuracy to gain greater control over the progress.

2.2 Software Testing

Correctness is the minimum requirement of software projects and the essential purpose of testing. The purpose of test phases is to improve quality by running software with the goal of identifying defects and evaluating an attribute or capability to see if it performs as expected. According to the findings of the survey in [14], over half of the projects spend more than 20% of the time testing their system, resulting in an average of 30% across all projects. As the project size grows, this percentage lowers.

2.2.1 Levels of Abstraction

- **Functional Testing**

  This approach relates to a testing method in which any system can be thought of as a function that is meant to return specific output values corresponding to its input domain. Therefore, functional testing can be understood as Black-Box Testing where only the inputs, outputs, and specifications are visible. All test cases are derived from required behaviors, and no code implementation details are taken into account. During testing, different inputs are exercised, and the outputs are compared to the target to validate their correctness.

This testing level is mainly intended to establish confidence. The fundamental drawback of functional testing is to often suffer from twin problems of redundancies and gaps. [15] [16]

- **Structural Testing**

  In contrast with functional testing, this approach is commonly known as White-Box Testing and refers to test cases that are planned accordingly to how the function is actually implemented. This technique pays attention to the structure of the software, being the code its test case basis for the sake of seeking faults.

  A reliable combination of both tests ensures that specified requirements are fulfilled and that structured testing is measured. [15] [16]

2.2.2 Levels of Testing

- **Component Testing**: During this stage, developers test the code just written on the level of each module. It is an evaluation of a function or a single procedure. This type of tests is also referred to as Static Tests and Unit Tests. [16] [17]

- **Integration Testing**: The interaction of different components of the software to be developed is tested as if the program were isolated in order to ensure that it functions as planned. [16] [17]

- **System Testing**: Major modules and components are linked and tested for interoperability. System tests can easily be understood by the customer. [16] [17]

- **User Acceptance Testing**: Prior to the software being released, the client conducts a final round of testing to ensure that all the intended business requirements have been met. [17]

- **Performance Testing**: While the previous stages validated the software code and behavior, performance testing verifies the program's "speed" and "workload" by simulating peak load times. Not every software system has stated performance specifications. [15] [18]

2.2.3 SDLC Testing

- **Waterfall V-Model**

  The Waterfall Model is useful as a testing framework starting point due to the clear identification of distinct testing levels it allows, at the same time it clarifies the objectives that pertain to each one. The Waterfall V-Model given in Figure 2.4 is a variation of the traditional approach in Figure 2.1 emphasizes how the basic testing levels reflect the early development stages. Waterfall top-down development is designated by functional decomposition. On the other hand, the bottom-up testing framework fits well with hierarchical management structures and denotes the logical sequence: test the individual components and then integrate them into subsystems until the entire system performs the desired requirements. [16]

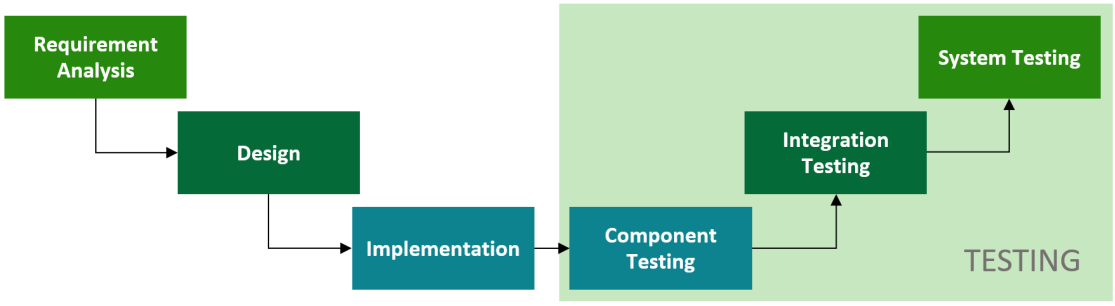


Figure 2.4: Waterfall V-Model: Testing Phases [16]

- **Agile**

  Concerning a project developed under an Agile Scrum approach which was explained in Section 2.1.2, the testing cycle occurs at two levels: the unit level at the end of each day, and the integration level of user stories as well as the system level at the end of each sprint. The increment obtained at the conclusion of each sprint is a potential deliverable product to the customer, indicating that it is a system-level work product. At the end of each iteration, a regression test is performed to ensure that the prior build still works with the added code. Following the program's successful regression test, new user stories are tested to ensure they meet the functional requirements, this stage is also known as progression testing. Keeping in mind that time-to-market is a top priority for this SDLC, the importance of the design-code-test cycle during each sprint is emphasized. [16]

2.3 Quality Assurance

Software quality assurance (SQA) is a technique that aids in the achievement of quality. It is concerned with the measurement and monitoring of software product quality, as well as the optimization of the development process, making it efficient and ensuring product quality at each stage of the software development life cycle. SQA is a formal process for evaluating and documenting the performance throughout the SDLC, providing quality control. [19]

The Triple Constraint Triangle in Figure 2.5, also known as Iron Triangle, has long been a key concept in project management. Each vertex symbolizes one of the three central criteria (time, cost, and scope), all of which have an evident impact on product quality. The balance between these central criteria is represented across the triangle by knowing that any change in any constraint would indicate pressure for changes in other constraints or quality will be affected. [20]



Figure 2.5: Project Management Triangle [20]

2.3.1 ISO/IEC 25000

The International Organization for Standardization (ISO) Software Product Quality, Requirements and Evaluation (SQuaRE) series of standards is the international standard most directly applicable to software product quality control, being its core ISO/IEC 25000 which stands for Quality Management. The main divisions incorporated in this standard are better appreciated in Figure 2.6.



Figure 2.6: ISO/IEC 25000 [21]

ISO/IEC 25010 aims to be a quality model. Additionally, ISO/IEC 25020 is a separate division that provides a reference model for measuring the quality of software throughout metrics. The ability to categorize quality criteria has always been a major benefit of employing quality models. This is mentioned in ISO/IEC 25030, which states that quality requirements are set based on quality characteristics by assigning a target value to the quality measures. The final division of the series (ISO/IEC 25040) is concerned with adding a level of interpretation to the results acquired by the measurement functions in comparison to the settled requirements, and so determining if the software fulfills the required quality rating level. [21]

2.3.2 Quality Model

Quality cannot be assessed solely on the basis of a single criterion. Software metrics are quantitative assessments of indicators in processes, components, and systems that influence development quality.

ISO/IEC 25010 quality model is the most widely used taxonomy for predicting the overall quality of software. This standard specifies two models: Product Quality and Quality in Use, offering a framework that stipulates quality should be split into layers of quality factors that are defined by different criteria. Figure 2.1 depicts the proposed model for addressing product quality, which is the most related to the subject of this dissertation. [21]

Literature also presents further quality models and comparisons of the standardized one with other relevant ones: McCall, Boehm, Dromey, and FURPS (Functionality, Usability, Reliability, Performance, and Supportability). The reviewed models share relevant factors and criteria with the standard quality model, such as efficiency, reliability, functionality, and flexibility. However, some of the models do not cover elements such as maintainability and security, as the ISO standard specifies. On the other hand, the FURPS model includes the factor of supportability, whilst the The Dromey model is the only model that takes reusability and interoperability into account. [19] [22]

The purpose of this review is to go over the key factors of quality that serve as a starting point. The conclusion is that the quality assessment model shall be carried out while attempting to prioritize project scope, budget, and schedule, as well as corporate resources.

Table 2.1: Quality Model ISO/IEC 25000 [21]

Factor	Criterion
Functionability	Functional correctness Functional completeness Functional appropriateness
Reliability	Maturity Availability Fault tolerance Recoverability
Performance efficiency	Time behaviour Resource utilisation Capacity
Usability	Appropriateness recognisability Learnability Operability User error protection User interface aesthetics Accessibility
Maintainability	Modularity Reusability Analysability Modifiability Testability
Security	Confidentiality Integrity Non-repudiation Accountability Authenticity
Compatibility	Co-existence Interoperability
Portability	Adaptability Installability Replaceability

2.4 Risk Assessment

The goal of risk assessment is to avoid issues from happening, and to be proactive rather than reactive by determining a response strategy when weaknesses happen, as we can see in the generalized model in Figure 2.7.

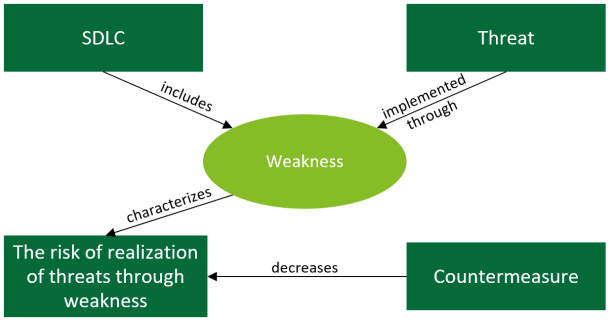


Figure 2.7: Generalized Risk Assessment Model [3]

The key activities for risk management are: identifying and analyzing the risk; planning a response for future occurrences; tracking and controlling every project; communicating the strategy to the team; and finally re-evaluating the risk framework periodically. [4]

2.4.1 Issues Management

Errors/Bugs/Mistakes can occur at any stage of a software development project, resulting in issues/defects/faults which must be detected throughout the testing stages. An issue is an event that has occurred and is negatively affecting the project: its scope, schedule, budget, or quality. When a defect is raised, the following procedures are taken: identifying and reporting as soon as possible; analyzing impacted areas and prioritizing according to severity; monitoring and controlling its resolution before testing again. [16]

2.4.2 Risk Scoring

"A risk can range from catastrophic to negligible". [4] In paper [3], an approach to calculate a threat level of 0 to 1 is proposed. The concept is to determine the threat level (Th) based on the severity of the threat's impact on the resource (ER, specified in percentage), taking into account the probability of it being implemented (P(V) specified in percentage). Thus, Equation 2.1 is obtained. [3]

$$Th = \frac{ER}{100}\frac{P(V)}{100} \tag{2.1}$$

With in mind, the risk scoring matrix is commonly structured as depicted in Figure 2.8.

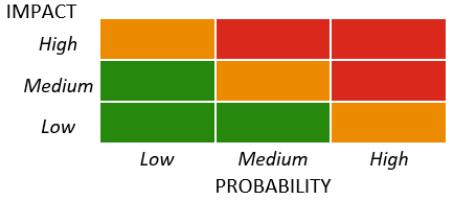


Figure 2.8: Risk Scoring Matrix [23]

2.4.3 Risk Indicators

Indicators are proposed for the purpose of quantifying a project's risk level, aiding in a better perception of risk factors. These can be employed at both the strategic and operational levels. It is critical to precisely identify and monitor indicators during the SDLC in order to avoid project failure and to ensure that the final product attains the required quality, avoiding rework and subsequently schedule overruns. [24]

According to the literature [5], risks are split into three classes: Product Engineering, Development Engineering, and Program Constraints. These are divided into different elements, each of which embeds indicators that, if not fulfilled (in the presence of risk factors), result in an addressed risk level. Figure 2.2 depicts an assortment of indicators, with the elements containing the largest number of occurrences of risk factors highlighted in red. [24]

Table 2.2: Risk Indicators [5]

	Element	Indicator
PRODUCT ENGINEERING	Requirement	Requirement Stability
		Requirement Completeness
		Requirement Clarity
		Software Size
		Requirement Dependence
		Requirement Complexity
		Recoverability
	Design	Reuse Level
		Interfacing Level
	Code and Unit Test	Number of Programming Languages
		Unit tests specificatiuon level
		Number of target devices
	Integration and Test	Product Stability
	Engineering Specialities	Difficult level to implement security requirements
DEVELOPMENT ENGINEERING	Development Process	Experience on the development process
		Geographical distribution level
	Development System	Development infrastucture availability
		Development software availability
	Management Proces	Project manager experience level on managing
		Project dependence level
		Process changes
		Maturity level
	Management Methods	Motivation level
		Role organization effectiveness
	Work Environment	Conflict Level
		Team Focus
		Turnover
PROGRAM CONSTRAINTS	Resources	Team knowledge level
		Team experience level
		Team availability
		Team size
		Project complexity
		Project criticality
		Project size
		Risk points
		Financial feasibility
	Contract	External dependences level
	Program Interfaces	Product business value
		Client participation level
		Client experience

Lastly, these indicators must be continuously assessed to reduce the risk level, assuring a value-driven development process.

2.4.4 Software Metrics

Measurement has always been essential to the development of any engineering discipline. They give helpful information about external quality attributes of software such as its maintainability, usability, and reliability (Figure 2.6: Quality Model ISO/IEC 25000) and are also a method for evaluating the effort required. Software metrics serve as the foundation for:

- Analyzing and enhancing software quality;
- Comparing and evaluating capabilities of different projects;
- Tracking and predicting the process;
- Ultimately, reporting, decision-making, and validation. [25]

Software quality evaluation is a procedure that must be conducted at each key SDLC step. These metrics are broadly categorized into 3 groups, as seen below, with some of the most prevalent metrics assigned to each group:

- **Project Metrics**: used in the early stages of a project to measure potential risks and build a stable plan for development.
  - Software science metrics;
  - Risk metrics.
- **Process Metrics**: used to monitor the development process and team efficiency.
  - Control flow metrics;
  - Information flow metrics;
  - Testing metrics;
  - Performance metrics;
  - Automation metrics.
- **Product Metrics**: the most extensive ones, considering the developed product itself to ensure final product quality.
  - Size and complexity metrics;
  - End-user satisfaction metrics.

A metrics program based on an organization's objectives will aid in communicating, measuring, and ultimately achieving those goals. People will endeavor to achieve what they deem essential, which in this case is to track and minimize risk throughout the SDLC. [22] [25] [26]

2.5 Conclusion

All activities are plagued by dangers, but software projects are especially vulnerable. Projects with abstract deliverables and intangible outcomes are especially susceptible to dangers. Similar to how visibility on the road minimizes accidents, visibility in projects decreases dangers. Process maturity increases transparency and reduces risks. [27]

Risk maturity is the capacity to "reduce noise and concentrate on really high-risk matters more effectively". To realize its benefits, a mature risk management process is necessary, one that takes into consideration the planning and implementation of the process as well as the skills and experience of the people who take part in it. To be mature in the way of managing risk, it is required a framework for assessing risk, allowing to compare against a structured benchmark. Risk management is a fundamental aspect of project management that entails aligning the concerns of stakeholders within a mature framework to enable risk identification, analysis, tracking, prevention, and mitigation. [28]

An estimating model to anticipate and evaluate risks will be developed based on the research conducted in this chapter, including a risk projection within the scope of all estimating models and procedures that are currently in use.

Chapter 3

Description

The current chapter aims to contextualize the reality of Deloitte projects, focusing on the process flow analysis and including a brief discussion regarding certain tracking and planning measures that have already been established. The chapter ends with a summary of the main challenges to be overcome.

The following are the generic traits that best describe technology consulting projects:

- A temporary and finite life cycle with a defined start and end;
- A defined software product with a fixed scope and specified requirements;
- A defined and assigned cross-functional project team handling a set of activities required to achieve the desired outcomes;
- A business-approved budget and schedule;
- Customer involvement throughout all project stages;
- Changes over time generate uncertainties that require risk management.

3.1 ISO 9001

Deloitte is certified by the International Standard ISO 9001:2015 which means that the firm has been acknowledged for its Quality Management System and for delivering high-quality deliverables to clients. This standard improves processes execution flow and customers' satisfaction, promoting new business prospects, and a continuous improvement mindset. This Quality Management System is composed of seven fundamental principles: customer focus; leadership; people commitment; processes approach; continuous improvement; fact-based decision-making; and relationship management.

3.2 Process Flow

At Deloitte, most TC projects in the firm follow an Agile delivery framework largely based on Scrum, previously conceptualized in Section 2.1.2. However, some projects are developed using an approach akin to the Waterfall method to fulfill each project-specific requirements, such as schedule and cost. To summarize, practically every project's delivery approach illustrates a series of stages, each of which follows the most feasible strategy, often culminating in a hybrid framework (Section 2.1.3). The stages are common to all projects, regardless of workflow, which demonstrates the plausibility of this thesis' main objective of building a framework to track and measure the risk of any project throughout its development, incorporating risk factors inherent to the various stages.

The major stages of any software development project are: Inception; Development; Testing and Release. Concerning the risk assessment during each stage, it barely matters whether they are ordered sequentially (Waterfall approach) or cyclically (Agile approach).

3.2.1 Inception

Firstly, the scope of the business proposal is validated by the stakeholders. The customer's high-level requirements are gathered and analyzed for the initial plan to be sketched. With the requirements in mind, the best strategy for approaching the project while adhering to all of the intended constraints is defined. The expected business value is split up into a roster of primary functional requirements (Waterfall approach) or user stories (Agile approach) that will be developed to achieve the desired outcome. At this point, high-level risks and dependencies are identified as well as acceptance criteria are validated.

3.2.2 Development

This stage workflow varies depending on whether the project is delivered using an Agile or Waterfall methodology. Briefly, it includes the functional design of the solution as well as its implementation, which means planning what is committed to being delivered during the current sprint (in the case of an Agile Scrum approach) or the current activity (in the case of a Waterfall approach), writing the code and compiling it into an operational application.

The Agile Scrum method is an iterative process in which requirements can be refined prior to the start of each sprint. Tasks are organized by user stories, which represent minimum viable product sets and can be perceived as incremental process divisions.

Nevertheless, when using the Waterfall method, all planning is done at once and before implementation in a downwards sequence. After the design step, all functionalities have been established and cannot be changed during the entire implementation because it is not cyclical.

3.2.2.1 Development Planning

For this stage's successful planning is essential to consider two significant aspects of the user stories (u.s.) to be developed. First, is the priority of the u.s., this means, the plan must prioritize the ones that are required to be completed before others. Second, the number of developers' work hours necessary to complete a user story is determined by classifying it by their size and providing a time estimate to each group. Table 3.1 provides Deloitte's consideration based on several past projects.

Table 3.1: User Stories Effort Estimation

Size	Hours
XS	12
S	24
M	40
L	80
XL	160
XXL	320
2XXL	640

3.2.3 Testing

The following testing phases apply to all technology projects at Deloitte for ensuring the verification of intended software requirements.

- **Component Acceptance Testing (CAT)**

  This depicts the first stage of testing that occurs at the end of each sprint (Agile) or immediately at the end of the development stage (Waterfall). It is meant to test individual units both structurally and functionally (Section 2.2.1) to ensure that they work properly in isolation and to validate that the acceptance criteria has been met.

- **System Integration Testing (SIT)**

  After CAT for all user stories (Agile) or requirements (Waterfall) is completed, the SIT stage begins, which is a global confirmation that the code is conforming to what is required as well as a validation of the system's end-to-end functionality, integration and interfacing with other IT systems in a fully integrated environment.

- **User Acceptance Testing (UAT)**

  Finally, business users from the client conduct end-to-end functional testing to ensure that the system works as expected and in accordance with the signed-off functional design.

3.2.3.1 Tracking of Testing Stages

During test stages, reports are produced, usually on a daily basis to track progress and defect occurrences. Three rates play a key role in this regard: execution rate, pass rate, and percentage

of critical defects. Commencing with the execution rate, it tracks the actual number of conducted tests in comparison with the overall number of tests to be completed. In contrast, the pass rate reflects the implementation quality by dividing the number of passed tests by the total number of test cases to be completed. Furthermore, the pass rate does not discriminate between defects with a critical high impact (risk scoring matrix in Figure 2.8), hence the percentage of defects with a really high risk within the total number of found defects is addressed.

Noting that as the project goes through the testing phases, the proportion of critical defects must decrease, reaching zero at the UAT level. The quality is expected to rise as the release stage approaches.

The chart depicted in Figure 3.1 is the result of an examination of optimum execution and pass rate rates across testing phases.

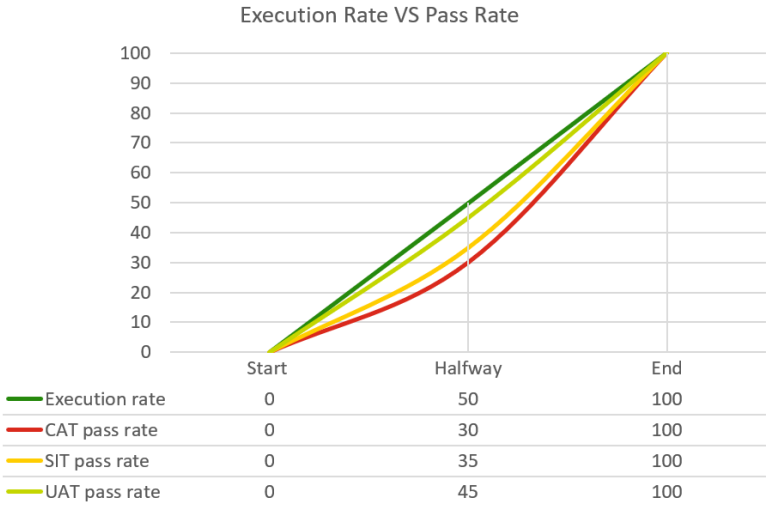


Figure 3.1: Execution Rate VS Pass Rate

The key findings drawn from Figure 3.1 are as follows:

- Execution rate may have a linear growth in each testing stage;
- Pass rate does not vary linearly across testing phases;
- At the end of each testing stage, all the tests must have been conducted and passed;
- Halfway through the stage, the gap between execution rate and pass rate at a given point must drop drastically from CAT to UAT, being 20% in CAT and reaching 5% in UAT.

3.2.4 Release

Before system components are ready to be deployed to the production environment, a Quality Assurance assessment is required to ensure software quality. Afterward, the code is deployed to production and go live.

3.3 Problem Identification

Even if projects have varied scopes, approaches, allotted resources, etc., the most significant risk factors are shared by all of them in regard to the stages of the process flow. With this in mind, it is desirable to have a tool to help project management rather than relying exclusively on the intuition and expertise of the team's senior members. In this manner, two essential issues that must be addressed with this thesis are highlighted:

1. **Risk visibility and awareness**: Having a tool capable of estimating the risk score at any moment based on key inputs enables anyone to keep this number in mind. Once the team realizes that the project is not adhering to the plan and hence is not achieving the intended execution quality, alertness rises.

2. **Quick and consistent decision-making**: Accelerate the decision-making process by assigning a high risk score to the causing criteria. Additionally, decisions for all projects must be based on the same standard metrics.

3. **Up-front plan**: By delving further into metrics that have the greatest impact on risk, mitigation actions may be adopted quickly.

4. **A solid basis for presenting consumers with a variance forecast**: Modifications in the timeline, budget, and needed resources might be foreseen. Attending the projects at hand are consultancy projects, customers may be given a foundation to justify schedule overruns and other adjustments throughout the project.

Chapter 4

Risk Framework

As previously outlined, the final goal of this thesis is to provide a tool to get a quantified risk score across the SDLC, with the purpose of predicting the execution quality for any FS tech consulting project. To accomplish so, the main metrics to assess the quality of each main stage cross to any project are considered, and their relevance is weighted in order to contribute with an accurate weight to the final risk score.

This framework's design focuses on highlighting risk indicators inherent to each stage and, accordingly, on developing a logical formula to deliver the risk score. Its purpose consists in:

- Streamlining the most relevant quality parameters from inception to the last testing stage;
- Providing faster and more informed decision-making to ensure that projects are on track to meet initial requirements (schedule, quality, and so forth);
- Emphasizing the main risk factors contributing to a high risk level during software implementation, assuring that teams are aware of the origin of risks.

4.1 Structure

Metrics are the foundation for any business improvement, prediction, and the cornerstone of assessment. They represent a measurement-based technique used to provide engineering and management information for processes, products, and service quality. It streamlines monitoring by allowing management and business to digest data and drill down, whenever required. [26]

First and foremost, the rationale for structuring the risk assessment framework was to select the most relevant metrics to the Project itself, representing the framework's first dimension. These metrics are input data that can be provided prior to the start of the project and remain nearly static throughout the development life cycle.

Secondly, each major stage of the workflow is associated with a particular dimension of the framework in order to describe the key points that have an impact on the risk score. Relevant process metrics regarding the stages are linked to the respective dimension. Concerning the significance of testing in assessing project risk, it is broken down into three distinct dimensions

corresponding to the three main testing scopes. Thus, in accordance with Section 3.2, Inception, Development, CAT, SIT, and UAT are the dimensions that complement the framework.

Release stage was not included as a dimension in this framework attending to the fact that it covers the final Quality Assurance before the software is produced. The main focus of this work is to quantify execution risk throughout the development life cycle in order to avoid unpredicted high-risk levels at the release stage, as well as to address the source of high-risk scores at each stage level in order to mitigate the impact during the process.

The current framework strives to be cumulative, considering all the data from previous phases. Figure 4.1 shows a flow that illustrates the logic behind the Risk Assessment Framework.

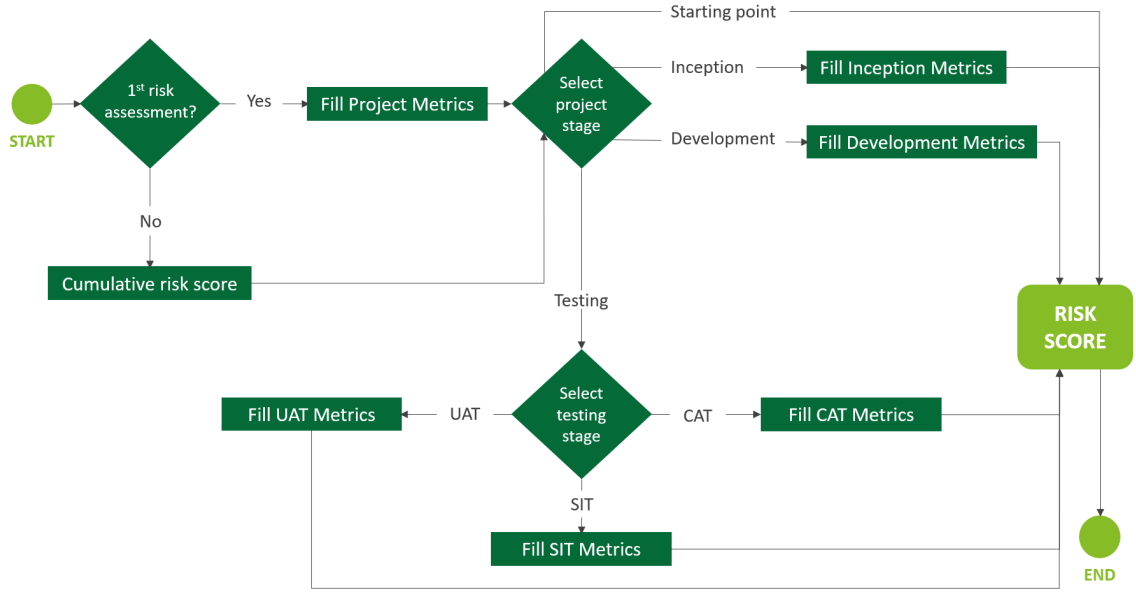


Figure 4.1: Risk Assessment Framework Structure

4.2 Score System

4.2.1 Output

A percentage will be obtained as a result of the combination of metrics weighting. To quickly grasp its significance, a scale ranging from 1 to 5 is employed, as shown in Figure 4.2. Every project has a risk score, even if it is fairly low. Thus, the scale's level 1 corresponds to a risk score ranging between 0% and 20%. Throughout the framework's development, a 15% percentage was parameterized as the optimal risk level to address "no risk factors", which occur when the metric does not contribute to increased risk. Then, there are levels 2 through 4, which increase in severity. Ultimately, when the risk score exceeds 80% (level 5), the project recovery is nearly impossible.

In other words, a project with a risk level of 1 is likely to be completed on time, on budget, on scope, and with high quality. According to the Project Management Triangle in Section 2.5, increasing the risk level jeopardizes one or more of these parameters, implying that mitigation actions will be required to ensure the fulfillment of the initially established plan.

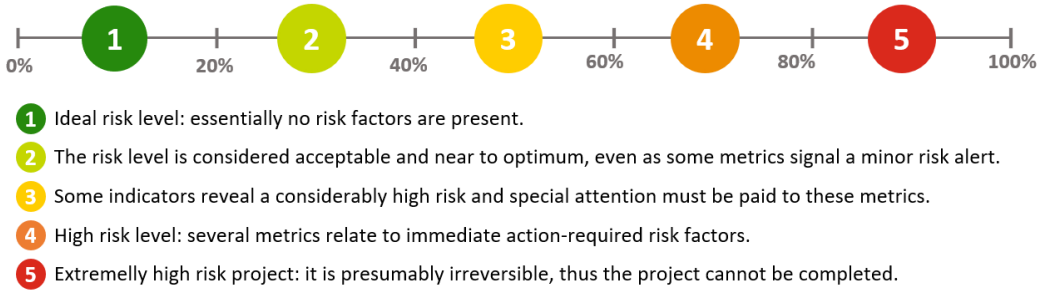


Figure 4.2: Risk Score Scale

4.2.2 Dimensions Weighting

This framework aims to estimate project risk at each stage, regarding the metrics considered for that stage and also taking into account, in a lighter proportion, the project metrics and the cumulative risk score estimated for the previous stages. Project metrics are considered in the assessment at any level, whereas stage metrics are weighted differently depending on the stage at which risk is assessed. When data about a specific project is entered, the cumulative risk score is updated, affecting the risk of that project's subsequent stages, emphasizing the execution quality of previous phases. The cumulative risk score is the result achieved for the last stage assessed previous to the current assessment in which information about subsequent stages was entered. Moreover, it is possible to measure risk prior to the project start using solely project metrics. Also, it is feasible to assess the risk for a given stage when the framework for the entire workflow is incomplete. The framework's score system is better understood in Table 4.1.

Table 4.1: Score System

Stage	Metrics	Weight	Dimension Score
Starting point	Project metrics	100%	1 - 5
Inception	Project metrics	25%	1 - 5
	Inception metrics	75%	
Development	Project metrics	15%	1 - 5
	Cumulative risk score	25%	
	Development metrics	60%	
CAT	Project metrics	15%	1 - 5
	Cumulative risk score	25%	
	CAT metrics	60%	
SIT	Project metrics	15%	1 - 5
	Cumulative risk score	25%	
	SIT metrics	60%	
UAT	Project metrics	15%	1 - 5
	Cumulative risk score	25%	
	UAT metrics	60%	
One stage assessment	Project metrics	25%	1 - 5
	Stage metrics	75%	

4.2.3 Metrics

The subsequent sections go over the metrics that have been raised for each dimension assessment aligned with Section 2.4.4. The most relevant indicators are obtained while prioritizing the risk indicators reported in Table 2.2 from a perspective aligned with Deloitte stakeholders' core concerns. On the other hand, some indicators were included as a result of suggestions from stakeholders in management positions.

Given that the score for each dimension ranges from 0% to 100%, the total contribution of the risk impact of all metrics associated with a dimension must be 100% as shown in Equation 4.1 which applies to any dimension.

$$\sum_{i=1}^{n}(risk\ impact)_i = 100\%, \quad n = no.\ dimension's\ metrics \tag{4.1}$$

Each metric measurement procedure has its own set of criteria for assessing it, also ranging from 0%, which means that the metric has no influence on the risk score, to 100%, which means that the metric significantly contributes to increasing the risk score. The percentage assigned to the specified criterion multiplied by the risk impact of the respective metric is indeed the contribution of that metric to the stage risk score. Thus, Equation 4.2 shows how each dimension risk score is obtained by summing up every metric contribution.

$$Dimension\ risk\ score = \sum_{i=1}^{n}(risk\ impact)_i * (criteria\ score)_i, \quad n = no.\ dimension's\ metrics \tag{4.2}$$

In short, there are 4 key ideas to grasp when it comes to the use of metrics in risk assessment:

- A collection of metrics are allocated to each dimension/stage, and a risk impact percentage is assigned to each of them so that the sum of the risk impacts of all the metrics equals 100%, which is the total risk of the respective dimension (Equation 4.1) and each dimension have a particular weight in the final risk score according to Table 4.1.

- Regarding process metrics (Inception, Development, CAT, SIT, and UAT), it is necessary to distinguish 3 time periods (start, halfway through, and end) due to the fact that the risk impact of the metrics changes as their significance varies along the stage.

- As a hallmark of testing stages tracking metrics, the evaluation criteria vary throughout the stage since an ideal criterion at the beginning is no longer acceptable at the completion.

- Certain metrics relating to the availability of development and testing environments and tools are deemed a must-have. In this manner, they only have an influence on the risk score when they are sub-optimal and significantly impact the risk. When they are optimal, though, their risk impact is zero percent.

4.3 Team Structure Matrix

Deloitte teams are cross-functional, which means they are made up of analysts, developers, testers, product owners (the voice of the client business), and managers, among others, who work on tasks of distinct scopes simultaneously. In this manner, Deloitte professionals may be categorized according to the role they perform and can be ranked according to their level of experience. Figure 4.2 displays an idea created throughout the course of this thesis research that may be applied to quickly grasp the makeup of various teams and that will be used in the following chapter to evaluate a metric pertaining to team structure in the assessment of Inception, Development, CAT, and SIT stages. Matrix columns are used to check the intended pyramidal shape by looking at the seniority distribution. Lines, on the other hand, are intended to address the different roles performed within the team.

Table 4.2: Team Structure Matrix: Deviation Analysis

	Staff	Senior	Manager	Actual	Ideal	Deviation
Functional resources	2	2	2	40%	60%	33%
Management resources	1	2	1	27%	10%	167%
Technical resources	1	1	0	13%	30%	56%
Testers	2	1	0	20%	0%	0%
Actual	40%	40%	20%		Function deviation ▶	64%
Ideal	20%	50%	30%	Seniority deviation ▼	**Total deviation 58%**	
Deviation	100%	20%	33%	51%		

Comparing the actual team's deviations from the ideal default percentages for the various development stages of each human resource function proportion, as well as their seniority, can be beneficial. This becomes more evident after analyzing Table 4.3 which displays the predefined optimal proportions for each stage. These proportions are the culmination of brainstorming with stakeholders who managed many projects.

Table 4.3: Ideal Teams Structure

	Performed roles					Seniority distribution			
	Functional resources	Management resources	Technical resources	Testers	Sum	Staff	Senior	Manager	Sum
Inception	60%	10%	30%	0%	100%	20%	50%	30%	100%
Development	10%	15%	60%	15%	100%	50%	30%	20%	100%
CAT	10%	15%	60%	15%	100%	50%	30%	20%	100%
SIT	10%	15%	40%	35%	100%	50%	30%	20%	100%

Figure 4.2 matrix is an example of a team's structure assessment. By filling up the white cells, the matrix estimates the actual proportions of each category. The deviation line and column then calculate the difference between the actual and ideal percentages, resulting in a total disparity in seniority and function. Lastly, a total deviation is calculated.

4.4 Project Metrics

This dimension depicts the first framework layer, asserting that it is applied to any project even before it begins. This set of 11 metrics can be used for any project at any time and is tied to the project itself, remaining constant throughout the SDLC.

1. **Type of service**: Needless to say, different types of services provided entail different risk levels. In this regard, it depicts the first metric considered for this dimension assessment. To handle this indicator, Deloitte TC projects are broadly classified into three types listed ahead in Table 4.4 in decreasing risk order. Also, risk impact and criteria for evaluating each option are presented.

Table 4.4: Project Metric 1: Type of Service

	Metric	Risk Impact	Criteria to assess		
1	**Type of service**	20%	3 options	Implement	60%
				Advice	45%
				Operate	15%

2. **Type of project**: The approach chosen to follow each project has an impact on Project dimension risk score, according to the weight and criteria presented in Table 4.5. Deloitte TC activity is broadly represented by the four categories in the table below.

Table 4.5: Project Metric 2: Type of Project

	Metric	Risk Impact	Criteria to assess		
2	**Type of project**	5,0%	4 options	Retire	60%
				Replace	45%
				Resource	35%
				Retool	15%

3. **Effort estimation**: This metric aims to evaluate the effort expected to complete the project. More required hours correspond to a higher risk level as shown in Table 4.6.

Table 4.6: Project Metric 3: Effort Estimation

	Metric	Risk Impact	Criteria to assess		
3	**Effort estimation**	15%	6 options	- 10.000h	15%
				10.000h - 20.000h	35%
				20.000h - 50.000h	45%
				50.000h - 150.000h	55%
				+ 150.000h	65%
				- 10.000h	75%

4. **Deloitte project team experience in similar projects**: Despite Deloitte specialists' extensive experience in a myriad of software projects, some of the TC projects have never been done before by the assigned team, resulting in a dearth of experience. When this occurs, the risk is negatively impacted, and its score rises (see Table 4.7).

Table 4.7: Project Metric 4: Deloitte Project Team Experience in Similar Projects

	Metric	Risk Impact	Criteria to assess		
4	**Deloitte experience in similar projects**	15%	3 options	No previous project experience	75%
				1 similar previous project	45%
				2+ similar previous projects	15%

5. **Project duration**: Some projects are constrained by short time frames, which increases the risk of not finishing on time and failing some quality parameters due to a lack of time. The evaluation of this metric in regards to the first release duration is shown in Table 4.8.

Table 4.8: Project Metric 5: Project Duration

	Metric	Risk Impact	Criteria to assess		
5	**Project duration**	2,5%	3 options	- 6 months	60%
				6 months - 1 year	35%
				+ 1 year	15%

6. **Project location**: Some projects are carried out fully overseas, which may result in a quality trade-off. Table 4.9 shows how the location of a project affects its risk score, whether it is onshore, nearshore, or offshore.

Table 4.9: Project Metric 6: Project Location

	Metric	Risk Impact	Criteria to assess		
6	**Project location**	2,5%	3 options	Offshore	75%
				Nearshore	35%
				Onshore	15%

7. **Customer's C-level sponsorship**: Deloitte stakeholders believe that projects have a higher risk score when they are not sponsored by a customer's C-level. Thus, this metric assessment is depicted in Table 4.10.

Table 4.10: Project Metric 7: Customer's C-level Sponsorship

	Metric	Risk Impact	Criteria to assess		
7	**Customer's C-level sponsorship**	7,5%	Yes/No	Yes	15%
				No	75%

8. **Other parties involved**: When parties collaborate with Deloitte on a project, the risk associated with it increases. Table 4.11 portrays the weight of this indicator in the Project Dimension risk score.

Table 4.11: Project Metric 8: Other Parties Involved

	Metric	Risk Impact	Criteria to assess		
8	**Other parties involved**	7,5%	Yes/No	Yes	75%
				No	15%

9. **Customer experience with the chosen delivery approach**: Deloitte stakeholders have not identified a pattern that correlates project risk with the methodology used in its development (Section 2.1). This indicator is included in the framework to evaluate the customer's practicability for the chosen methodology.

Table 4.12: Project Metric 9: Customer Experience with the Chosen Delivery Approach

	Metric	Risk Impact	Criteria to assess		
9	**Customer experience with the chosen delivery approach**	2,5%	yes/no	Yes	15%
				No	75%

10. **IT projects client preparation**: A sizable proportion of Deloitte customers has partnered on more than one TC project with the firm or has been enrolled in other IT projects. As a result, in addition to raising awareness about the customer's experience with IT systems, an indicator for the risk level can be assigned towards the customer's project preparation by answering 6 questions about actions that the client should have taken previously, which contribute to the risk score, as shown in Table 4.13.

Table 4.13: Project Metric 10: IT Projects Client Preparation

	Metric	Risk Impact	Criteria to assess			
10	**IT projects client preparation**	10,0%	6 yes/no questions	• Establish strategic goals and desired business capabilities • Assess current core system(s) functional and technical capabilities	Answered "Yes" to 1 or 2 questions	90%
				• Assess impacts on other architectural layers • Define the desired target architecture and needed technical capabilities	Answered "Yes" to 3 or 4 questions	60%
				• Assess vendor platforms as potential full or partial core replacement • Elaborate scenarios and the transformation program roadmap	Answered "Yes" to 5 or 6 questions	15%

11. **Scope complexity**: There is a risk assessment classification of the 147 market offerings within tech consulting projects. These market offerings are grouped into 31 offerings, which are then grouped into 8 Responsibility Centers (RC) within Financial Services at Deloitte. Each market offering is internally classified regarding its risk as 'normal,' 'greater than normal' and 'much greater than normal' in the firm, and the criteria contribution to the risk impact of the offerings are 15%, 45%, and 60%, respectively. Table 4.14 outlines scope complexity metric contribution and evaluation. Due to the confidentiality of this assessment, the marketing offerings portfolio risk cannot be included in the thesis.

Table 4.14: Project Metric 11: Scope Complexity

	Metric	Risk Impact	Criteria to assess		
11	Scope complexity	12,5%	8 RC's	31 offerings	147 market offerings	60% 45% 15%

4.5 Inception Metrics

The purpose of this dimension is to assess the risk associated with the Inception stage. To evaluate this process dimension, as well as the remaining ones that portray different stages, it is essential to consider whether the data is inserted at the start, halfway through, or at the end of that phase.

1. **Inception team structure**: This indicator considers the team in charge of Inception, aiming to analyze its deviations to the default ideal matrix (Tables 4.2 and 4.3). The assessment criteria in Table 4.15 is calculated in accordance with Section 4.2.

Table 4.15: Inception Metric 1: Team Structure

	Metric	Risk Impact			Criteria to assess	
		Start	Halfway	End		15%
1	Inception team structure	30,0%	15,0%	5,0%	Matrix deviation	Continuous scoring 90%

2. **Inception stage duration**: This indicator utilizes the same principle as Project Metric 5 (project duration), seeking to address the Inception time frame feasibility. Its risk impact lessens as the stage progresses because other, more essential criteria emerge and gain significance.

Table 4.16: Inception Metric 2: Stage Duration

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		- 1 month	75%
2	Inception stage duration	30,0%	15,0%	5,0%	3 options	1 to 2 months	45%
						+ 2 months	15%

3. **Client availability for workshops**: Throughout the Inception phase, it is imperative to comprehend all consumer requirements and determine the approach to be followed. It is critical to maintaining close contact with the client for this, and if the customer is unable to conduct workshops, a high risk score will emerge.

Table 4.17: Inception Metric 3: Client Availability for Workshops

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		Yes	15%
3	Client availability for workshops	40,0%	30,0%	10,0%	3 options	Partially	60%
						No	90%

4. **Customer sign-off on all functional designs**: Customers' signature on functional designs is required before the Development stage can begin. As a result, they must be constantly signed as long as they are designed to ensure execution quality at this point. From the beginning to the end, obtaining all of these functional design sign-offs becomes increasingly important.

Table 4.18: Inception Metric 4: Customer Sign-off on All Functional Designs

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		Yes	15%
4	Customer sign-off on all functional designs	0,0%	20,0%	50,0%	3 options	Partially	60%
						No	90%

5. **Delivery delay**: A significant delivery delay at each stage increases the future risk of delivery in line with the expected time frame. It was determined, however, that when this metric is reached (deliverables on time), it may not be counted into the score calculation, since it does not represent a risk factor, therefore its risk impact is distributed over the remaining dimension's metrics. This rule applies to similar metrics ahead.

Table 4.19: Inception Metric 5: Delivery Delay

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		On time	-
						- 5%	25%
5	Delivery delay	0,0%	20,0%	30,0%	5 options	5% - 20%	55%
						20% - 40%	70%
						+ 40%	90%

4.6 Development Metrics

The metrics for the development stage are intended to identify the presence of risk factors during development. When it comes to a process, it is necessary to divide it into three time breaks as in the previous Section.

1. **Development team structure**

Table 4.20: Development Metric 1: Team Structure

	Metric	Risk Impact			Criteria to assess	
		Start	Halfway	End		15%
1	Development team structure	15,0%	10,0%	5,0%	Matrix deviation	Continuous scoring
						90%

2. **Development stage duration**

Table 4.21: Development Metric 2: Stage Duration

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
2	Development stage duration	15,0%	10,0%	5,0%	3 options	- 1 month	75%
						1 to 2 months	45%
						+ 2 months	15%

3. **Number of user stories/requirements**: This indicator is intended to assess capacity in terms of the amount of work to be done. It determines whether the actual number of u.s. (Agile approach) or requirements (Waterfall approach) to be developed is reasonable. Criteria in Table 4.22 is based on Section 3.2.2.1, considering medium sized user stories.

Table 4.22: Development Metric 3: Number of User Stories/Requirements

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
3	Number of user stories (Agile) / requirements (Waterfall)	15,0%	10,0%	5,0%	3 options	- 1,5 u.s./developer/week	15%
						1,5 - 2 u.s./developer/week	45%
						+ 2 u.s./developer/week	75%

4. **All tools availability**: Undoubtedly, having all of the necessary tools for software development is crucial. Its importance arises from start to end of the stage because other technical tasks must be completed before the implementation itself can begin. Similar to Inception metric 5, this metric is exceptional in that it does not lower risk when the assessment criterion is satisfied; rather, it is only factored into the risk score and greatly raises it in cases when the answer is either "No" or "Partially".

Table 4.23: Development Metric 4: All Tools Availability

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
4	All tools availability	27,5%	25,0%	5,0%	3 options	Yes	-
						Partially	60%
						No	90%

5. **Development environment availability**: This indicator depicts the conditions that developers must have encountered in order to implement software. This metric follows the same logic as the previous one, as for "must-have" indicators only being considered when they are not completely verified.

Table 4.24: Development Metric 5: Development Environment Availability

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
5	Development environment availability	27,5%	25,0%	5,0%	3 options	+ 80%	-
						70% - 80%	60%
						- 70%	90%

6. **Amount of delivered scope**: The risk increases when the scope is not delivered in its full extent. However, when testing steps begin with all anticipated scope achieved, the risk score for this metric is divided among the others in the development dimension, as it is no longer relevant to impacting the risk score.

Table 4.25: Development Metric 6: Amount of Delivered Scope

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
6	Amount of delivered scope	0,0%	10,0%	45,0%	Halfway 4 options	- 25%	75%
						25% - 50%	45%
						50% - 95%	15%
						+ 95%	0%
					End 5 options	All scope	-
						+ 95%	35%
						75% - 95%	55%
						50% - 75%	75%
						- 50%	90%

7. **Delivery delay**

Table 4.26: Development Metric 7: Delivery Delay

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
7	Delivery delay	0,0%	10,0%	30,0%	5 options	On time	-
						- 5%	25%
						5% - 20%	55%
						20% - 40%	70%
						+ 40%	90%

4.7 CAT Metrics

This set of 9 metrics is intended to examine risk factors that may arise during the Component Acceptance Testing stage.

1. **CAT team structure**

Table 4.27: CAT Metric 1: Team Structure

	Metric	Risk Impact			Criteria to assess	
		Start	Halfway	End		
1	CAT team structure	20,0%	2,5%	2,0%	Matrix deviation	15%
						Continuous scoring
						90%

2. **CAT stage duration**

Table 4.28: CAT Metric 2: Stage Duration

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
2	CAT stage duration	20,0%	2,5%	2,0%	3 options	- 1 month	75%
						1 to 2 months	45%
						+ 2 months	15%

3. **Number of CAT tests**: Its goal is to figure out the amount of work assigned to each tester.

Table 4.29: CAT Metric 3: Number of Tests

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		- 8 tests/day/tester	15%
3	Number of tests	20,0%	2,5%	2,0%	3 options	8 - 10 tests/day/tester	45%
						+ 10 tests/day/tester	75%

4. **Testing environment availability**: This metric emerges as a critical condition for tests to be performed, taking the same approach as Development Metrics 4 and 5 in terms of risk score calculation, only being included in the risk formula when it is not verified.

Table 4.30: CAT Metric 4: Testing Environment Availability

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		+ 80%	-
4	Testing environment availability	25,0%	10,0%	2,0%	3 options	70% - 80%	45%
						- 70%	75%

5. **Percentage of automatic tests**: Automatic tests enhance efficiency during this phase, but they are not strictly necessary, thereby, this metric is only used to reduce risk, and it is never used to increase it for values greater than 15%, which was the previously percentage parameterized to address the optimal risk level.

Table 4.31: CAT Metric 5: Percentage of Automatic Tests

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		+ 30%	0%
5	Percentage of automatic tests	15,0%	5,0%	2,0%	3 options	15% - 30%	7%
						- 15%	15%

6. **Execution rate**: This measure covers testing capacity and efficiency. Assessment criteria for this and the next metric alter during the process in accordance with Section 3.2.3.1. The chart in Figure 3.1 was used to determine intervals for risk evaluation.

Table 4.32: CAT Metric 6: Execution Rate

	Metric	Risk Impact			Criteria to assess		
						- 30%	90%
					Halfway	30% - 45%	60%
		Start	Halfway	End	4 options	45% - 60%	15%
6	Exectution rate					+ 60%	0%
		0,0%	15,0%	20,0%		-85%	90%
					End	85% - 99%	75%
					3 options	100%	15%

7. **Pass rate**: This metric is concerned with implementation quality, hence its risk effect is always greater than that of execution rate.

Table 4.33: CAT Metric 7: Pass Rate

	Metric	Risk Impact			Criteria to assess		
7	Pass rate	Start	Halfway	End	Halfway 4 options	- 10% 10% - 25% 25% - 40% + 40%	90% 75% 15% 0%
		0,0%	25,0%	30,0%	End 3 options	-80% 80% - 99% 100%	90% 75% 15%

8. **Percentage of critical defects**: Distinguishing the number of bugs with a greater danger from the total multitude of issues detected in an effort to improve quality.

Table 4.34: CAT Metric 8: Percentage of Critical Defects

	Metric	Risk Impact			Criteria to assess		
8	Percentage of critical defects	Start	Halfway	End	3 options	- 10% 10% - 20% + 20%	15% 75% 90%
		0,0%	25,0%	30,0%			

9. **Delivery delay**

Table 4.35: CAT Metric 9: Delivery Delay

	Metric	Risk Impact			Criteria to assess		
9	Delivery delay	Start	Halfway	End	5 options	On time - 5% 5% - 20% 20% - 40% + 40%	- 25% 55% 70% 90%
		0,0%	12,5%	10,0%			

4.8 SIT Metrics

This stage contains a total of 11 metrics, the majority of which are identical to those in the previous section because both of them depict testing stages.

1. **SIT team structure**

Table 4.36: SIT Metric 1: Team Structure

	Metric	Risk Impact			Criteria to assess	
1	SIT stage duration	Start	Halfway	End	Matrix deviation	15% Continuous scoring 90%
		10,0%	2,5%	2,0%		

2. SIT stage duration

Table 4.37: SIT Metric 2: Stage Duration

	Metric	Risk Impact			Criteria to assess		
2	SIT stage duration	Start	Halfway	End	3 options	- 1 month	75%
		10,0%	2,5%	2,0%		1 to 2 months	45%
						+ 2 months	15%

3. Number of SIT tests

Table 4.38: SIT Metric 3: Number of Tests

	Metric	Risk Impact			Criteria to assess		
3	Number of tests	Start	Halfway	End	3 options	- 8 tests/day/tester	15%
		10,0%	2,5%	2,0%		8 - 10 tests/day/tester	45%
						+ 10 tests/day/tester	75%

4. Testing environment availability

Table 4.39: SIT Metric 4: Testing Environment Availability

	Metric	Risk Impact			Criteria to assess		
4	Testing environment availability	Start	Halfway	End	3 options	+ 80%	-
		20%	10,0%	2,0%		70% - 80%	45%
						- 70%	90%

5. Percentage of automatic tests

Table 4.40: SIT Metric 5: Percentage of Automatic Tests

	Metric	Risk Impact			Criteria to assess		
5	Percentage of automatic tests	Start	Halfway	End	3 options	+ 30%	0%
		15,0%	5,0%	2,0%		15% - 30%	7%
						- 15%	15%

6. Execution rate

Table 4.41: SIT Metric 6: Execution Rate

	Metric	Risk Impact			Criteria to assess		
6	Exectution rate	Start	Halfway	End	Halfway 4 options	- 30%	90%
						30% - 45%	60%
						45% - 60%	15%
						+ 60%	0%
		0,0%	15,0%	15,0%	End 3 options	-85%	90%
						85% - 99%	75%
						100%	15%

7. **Pass rate**

Table 4.42: SIT Metric 7: Pass Rate

	Metric	Risk Impact			Criteria to assess		
7	Pass rate	Start	Halfway	End	Halfway 4 options	- 10%	90%
						10% - 25%	75%
						25% - 40%	15%
		0,0%	25,0%	30,0%		+ 40%	0%
					End 3 options	-80%	90%
						80% - 99%	75%
						100%	15%

8. **Percentage of critical defects**

Table 4.43: SIT Metric 8: Percentage of Critical Defects

	Metric	Risk Impact			Criteria to assess		
8	Percentage of critical defects	Start	Halfway	End	3 options	- 10%	15%
						10% - 20%	75%
		0,0%	25,0%	30,0%		+ 20%	90%

9. **Number of integrated systems**: The complexity of this testing stage reflects the number of systems that are integrated.

Table 4.44: SIT Metric 9: Number of Integrated Systems

	Metric	Risk Impact			Criteria to assess		
9	Number of integrated systems	Start	Halfway	End	3 options	- 5	15%
						5 to 10	45%
		15,0%	2,5%	2,5%		+ 10	60%

10. **Percentage of successful CAT tests**: This metric measures the percentage of Component Acceptance Tests that were completed successfully at the start of the SIT stage.

Table 4.45: SIT Metric 10: Percentage of Successful CAT Tests

	Metric	Risk Impact			Criteria to assess		
10	Percentage of successful CAT tests	Start	Halfway	End	3 options	+ 90%	15%
						80% - 90%	60%
		20,0%	5,0%	2,5%		- 80%	90%

11. **Delivery delay**

Table 4.46: SIT Metric 11: Delivery Delay

	Metric	Risk Impact			Criteria to assess		
11	Delivery delay	Start	Halfway	End	5 options	On time	-
						- 5%	25%
		0,0%	5,0%	10,0%		5% - 20%	55%
						20% - 40%	70%
						+ 40%	90%

4.9 UAT Metrics

Some of the following 13 measures are comparable to those utilized in prior testing phases. The differences come as a result of the customer doing the final testing phase instead of Deloitte team.

1. **Customer support from Deloitte team**: Rather than addressing the Deloitte team as a means of validating the human resources' capabilities to do the task, as in the preceding process dimension, it is only critical at this point to certify that the consulting team is ensuring the customer has the necessary monitoring and assistance.

Table 4.47: UAT Metric 1: Customer Support From Deloitte Team

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
1	Customer support from Deloitte team	10,0%	2,5%	2,0%	3 options	Test plan monitoring	75%
						Business support	45%
						Both of the above options	15%

2. **UAT stage duration**

Table 4.48: UAT Metric 2: Stage Duration

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
2	UAT stage duration	10,0%	2,5%	2,0%	3 options	- 1 month	75%
						1 to 2 months	45%
						+ 2 months	15%

3. **Customer availability to perform tests plan**: It is the customer's commitment to providing the necessary resources to execute the UAT phase.

Table 4.49: UAT Metric 3: Customer Availability to Perform Tests Plan

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
3	Customer availability to perform the tests plan	10,0%	2,5%	2,0%	3 options	No	75%
						Partially	45%
						Yes	15%

4. **Testing environment availability**

Table 4.50: UAT Metric 4: Testing Environment Availability

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End			
4	Testing environment availability	20,0%	10%	2,0%	3 options	+ 80%	-
						70% - 80%	45%
						- 70%	90%

5. Percentage of automatic tests

Table 4.51: UAT Metric 5: Percentage of Automatic Tests

	Metric	Risk Impact			Criteria to assess		
5	Percentage of automatic tests	Start	Halfway	End	3 options	+ 30%	0%
		10,0%	3,0%	2,0%		15% - 30%	7%
						- 15%	15%

6. Execution rate

Table 4.52: UAT Metric 6: Execution Rate

	Metric	Risk Impact			Criteria to assess		
6	Exectution rate	Start	Halfway	End	Halfway 4 options	- 30%	90%
						30% - 45%	60%
						45% - 60%	15%
						+ 60%	0%
		0,0%	15,0%	15,0%	End 3 options	-85%	90%
						85% - 99%	75%
						100%	15%

7. Pass rate

Table 4.53: UAT Metric 7: Pass Rate

	Metric	Risk Impact			Criteria to assess		
7	Pass rate	Start	Halfway	End	Halfway 4 options	- 20%	90%
						20% - 35%	75%
						35% - 50%	15%
						+ 50%	0%
		0,0%	22,5%	30,0%	End 3 options	-85%	90%
						85% - 99%	75%
						100%	15%

8. Percentage of critical defects

Table 4.54: UAT Metric 8: Percentage of Critical Defects

	Metric	Risk Impact			Criteria to assess		
8	Percentage of critical defects	Start	Halfway	End	3 options	- 5%	15%
		0,0%	22,5%	30,0%		5% - 10%	75%
						+ 20%	90%

9. Percentage of successful CAT tests

Table 4.55: UAT Metric 9: Percentage of Successful CAT Tests

	Metric	Risk Impact			Criteria to assess		
9	Percentage of successful CAT tests	Start	Halfway	End	3 options	100%	15%
		10,0%	2,5%	2,0%		90% - 99%	75%
						- 90%	90%

10. Percentage of successful SIT tests

Table 4.56: UAT Metric 10: Percentage of Successful SIT Tests

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		+ 90%	15%
10	Percentage of successful SIT tests	17,5%	7,5%	2,0%	3 options	80% - 90%	75%
						- 80%	90%

11. Number of business areas testing: To ensure the proper operation of the numerous functionalities, it is essential to gain client approval from multiple business areas. Nevertheless, this adds to the complexity, meaning more Deloitte support and, consequently, a higher risk score.

Table 4.57: UAT Metric 11: Number of Business Areas Testing

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		- 3	15%
11	Number of business areas testing	5,0%	2,0%	2,0%	3 options	3 to 6	30%
						+ 6	45%

12. Customer testing experience: At last, the customer's experience conducting these types of tests must be considered to ensure their accuracy.

Table 4.58: UAT Metric 12: Customer Testing Experience

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		No experience	75%
12	Customer testing experience	7,5%	2,5%	2,0%	3 options	1 UAT previously done	45%
						2+ UAT previously done	15%

13. Delivery delay

Table 4.59: UAT Metric 13: Delivery Delay

	Metric	Risk Impact			Criteria to assess		
		Start	Halfway	End		On time	-
						- 5%	25%
13	Delivery delay	0,0%	5,0%	7,0%	5 options	5% - 20%	55%
						20% - 40%	70%
						+ 40%	90%

4.10 Risk Evolution Benchmark

Any project at any time has an intrinsic risk score greater than zero, and certain phases are famously more hazardous. In light of this, it is useful to compare each stage's risk and the overall score progression to what is typical for Deloitte projects, even if the continual mission is to reduce the risk as much as possible in order to assure software quality. For this purpose, a preliminary benchmark for stakeholder perceptions of risk evolution throughout software implementation projects was established and is presented in Figure 4.3.

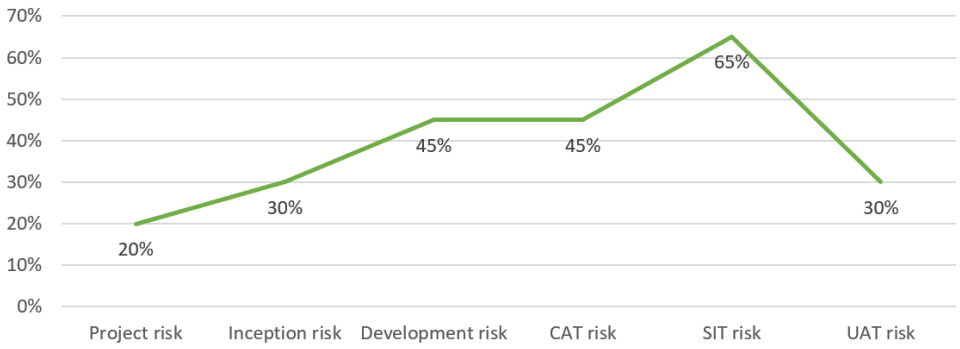


Figure 4.3: Risk Evolution Benchmark

Key perceptions drawn from this benchmark are:

- Risk usually increases from the starting point to the SIT stage;
- Typically, the lowest risk score is attained in the beginning, based only on project metrics;
- SIT may be the most hazardous phase since the most bugs are likely to occur;
- Development and CAT stages have similar risk ratings;
- Between SIT and UAT stages, the greatest risk reduction of the whole SDLC may happen.

4.11 Conclusion

At this point, the strategic part of this work is completed by structuring the framework around the primary risk factors described in Chapter 2 and linking them with the workflow of Deloitte TC projects described in Chapter 3. Also, the quality model ISO/IEC in Table 2.6 was deemed to be met throughout the entire workflow to ensure it is value-driven, using the selected metrics for risk assessment. Chosen metrics largely aim to cover project capacity, progress, quality, and coverage.

In conclusion, the theoretical model has been completed and the first round of validation has been accomplished, as stakeholders ranging from senior consultants to senior managers at Deloitte have validated that the framework aligns with their intuition and experience acquired from past projects.

The next chapter's emphasis remains on executing the first Validation Pilot to put this theoretical model into practice and make it accessible to teams. Although, most importantly, Model Validation Pilot 1 seeks to be presented to the firm's senior members, the partners, for approval with a broader perspective, which is fundamental to the framework's accuracy.

Chapter 5

Model Validation Pilot

As stated earlier, the target of this thesis is also to automate the built theoretical framework in order to provide a useful tool for risk assessment as well as to verify its reliability. This chapter describes all of the steps toward developing the first Model Validation Pilot (MVP).

MVP1 depicts a functional tool accessible to Industry Solutions responsibility center teams within CBO unit, where the internship was held. The objective was to develop an intuitive and straightforward application for project managers to enter data about their projects with the aim of calculating the final risk score and displaying visual dashboards to dig down into the risk elements. Ideally, the project assessment output reports should be representative enough to deliver to consumers in order for them to have a better sense of the project execution quality and main threats.

Furthermore, the purpose of the coded framework included its own validation. The computed risk level and results on the most impactful risk indicators were compared to the managers' risk evaluation and concern over the respective project risk based on their expertise and intuition. This comparison's conclusions depict the final validation of the First MVP designed within the scope of this thesis.

5.1 Functional Requirements

This MVP intends to compute the risk score based on the predefined metrics and accompanying formulae in Chapter 4. In the application, projects' identity-relevant information must be shown. Then, in order to visualize the data, a user-friendly interface needed to be developed to present framework outputs. The Power BI tool attempts to visually display the final score by highlighting:

- The final risk level inherent to each project;
- The risk evolution of risk score throughout the different stages of a project;
- The metrics contributing the most to the project's increased risk within each dimension for the purpose of easing further action decision-making;

- In-depth team structure analysis for each stage both in terms of seniority and skills set (since it depicts the most complex metric in the framework);
- The comparison of different projects' risk scores and risk fluctuation.

5.2 IT Tool Implementation

In the first instance, a Microsoft (MS) Forms was constructed in line with the reasoning presented in Figure 4.1 for the user to address the dimensions to analyze and fill the respective metrics, which is the initial step in the scheme in Figure 5.1 regarding data acquisition. This Microsoft Office program was chosen due to its accessibility to everyone when shared, its ease of use, and its easy interaction with the remaining apps in Figure 5.1.

The questions inquire about each metric, and the answers correspond to the evaluation criteria presented in Tables 4.4 through 4.58. The sole discrepancy between the theoretical framework and the forms pertains to Project Metric 11 responses: only Industry Solutions offers (CBO unit) were included, resulting in 5 distinct options instead of the 147 market options referred to in Table 4.14, as IS is the responsibility center where this tool was launched.

MS Forms responses are automatically collected in an Excel file that is configured to process the data in conformity with the framework criteria. Numerous Excel functions were programmed to determine the weight of each metric, the risk score for each dimension, and the overall risk score for the concerned project, featuring real-time data modeling that allows Power BI to be supplemented with the necessary tables.

As indicated, Power BI Desktop is instantly updated to generate graphical dashboards emphasizing the obtained overall risk level, each dimension risk score, and significant risk factors within the project that make it riskier. This step pertains to the final arrow in Figure 5.1 with respect to data visualization, which is covered in further depth in the next sections.



Figure 5.1: IT Solution Scheme

5.3 Power BI Application

5.3.1 Sequence Diagram

The navigation schemes and wireframe were created with an emphasis on enhancing the user experience and interaction. The filters and dashboards which will be displayed were first selected mostly based on their relevance and prospective interest to the user. Afterward, the sequence diagram that would serve as the cornerstone for the rest of the app programming was created, as shown in Figure 5.2. Pages 1 through 7 are intended to provide an in-depth analysis of a single project. The project's risk assessment is outlined on page 1, and subsequent pages are intended to

delve deeper into its dimensions, whereas Page 8 is only available for completed projects in order to compare the feedback provided at the end by Deloitte project manager and the customer's face to the risk estimated. Pages 9 and 10, conversely, are intended to provide an overview of several projects for the purpose of comparing them and seeking to identify trends.

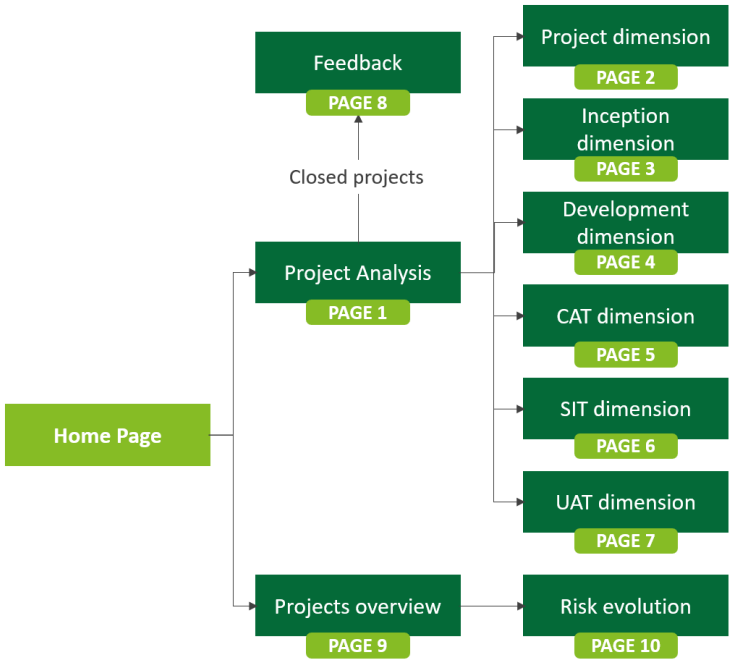


Figure 5.2: Sequence Diagram

5.3.2 Data Model

The first step into Power BI was to set the data source on the Power Query Editor, which allows to connect one or more data sources, shape, and transform the data according to the app's needs. In this context, the data source is the Excel file supplemented with MS Forms responses. This way, 25 out of the 27 tables comprising the data model were imported from Excel.

Appendix A shows the app's data model. The 2 additional tables beyond those imported from Excel are duplicates that had to be built in Power Query Editor to enable some desired features. To accommodate certain chart filtering requirements, the duplicates for the final and project scores were simply transposed and unpivoted from the first imported ones.

The UML diagram in Figure A.2 (Appendix A) illustrates how data relationships between the various tables were established to support the Power BI application. Simultaneously, the tables in Figure A.1 of the same Appendix do not have key data relationships with the remaining tables; instead, queries were coded for this purpose once it was more suitable for their use.

5.3.3 Report

To begin, the developed application's home page is presented in Figure B.1. The project analysis-related pages are meant to analyse a single project. For a better comprehension of the report,

the Power BI Pages supplied in figures in Appendix B regard actual projects' analysis as examples whose findings will be reviewed afterward. To maintain the confidentiality of this project's customer, some data from these dashboards must be hidden and shown generically.

Page 1's slicer (in Figure B.2) only allows one project to be selected. All pertinent information on its characterization is given. In addition to displaying the total project risk level on the tachometer whose colors correspond to the risk scale specified in Figure 4.2 with the respective description, the final score is also calculated in accordance with Table 4.1. Furthermore, each dimension's risk is portrayed in the graph, which acts as a SDLC risk progression schema. Regarding the preliminary Deloitte benchmark for risk evolution presented in Section 4.10, it is illustrated in a smoother yellow on the same chart, accordingly to Figure 4.3, allowing for easy comparison between the standard and actual progression. Also, the current stage is pointed out on the chart when the project is still open. The last assessment date is mentioned on every application page due to its significance in determining up-to-dateness.

The 6 buttons on the page are for navigation between Page 1 and Pages 2 to 7 which allow the detailed analysis of each dimension's metrics. The chat bubble shaped button only displays when the project's engagement status is closed since it permits navigation to Page 8 where Deloitte and the client's final feedback is displayed. Pages 2 through 8 contain the same filtering as Page 1, as they are intended to evaluate the previously selected project's dimensions and feedback once the project has been completed (Page 8).

Figure B.3 depicts Page 2 regarding the first framework layer: Project dimension. Beginning with the tachometer, it presents the dimension score and its corresponding interpretation. The centerpiece of this page is the treemap, giving all of the detailed data on the basis of the Project score calculation. This chart illustrates hierarchical metrics with nested rectangles whose weight corresponds to their risk impact on the score, as explained in Sections 4.2.3 and 4.4. Each rectangle's colors assess the risk level of the evaluation criteria for the metric represented by each rectangle.

For supplementing the acquired outputs, the tool also includes additional observations regarding general comments, major risk areas from the user's perspective, and already implemented mitigation actions. These may be crucial for an outsider's improved impression of the program.

Page 3 adheres to the same logic as the Project Dimension Report. The treemap is presented, although, it is important to underline that in this and the remaining dimensions since they are tied to process metrics, the treemap hierarchy fluctuates based on the stage point when the risk impact changes as outlined in Sections 4.5, 4.6, 4.7, 4.8 and 4.9.

To improve the user experience and make it easier to digest the data, a new feature has been added that allows the user to mouse over each rectangle and read the associated metric, risk impact, description, and assessment criteria. An example of this functionality is shown in Figure B.5.

Moreover, one of the metrics pertaining to stage dimensions relates to the team structure, and two radar charts presenting skills and seniority deviations are provided in the dashboard for a more thorough examination of its assessment based on the deviations calculated as in Section 4.2. With these charts, the shortages or surpluses of human resources for each seniority and functional position are made clear.

The Pages regarding the subsequent dimension's analysis are shown in the same Appendix in Figures B.6, B.7, B.8, and B.9. These pages are identical to the one for the Inception dimension, containing the corresponding metrics, with the exception of the UAT dimension page, where team structure is not analyzed because this testing phase is conducted by client's business areas.

In the next section, the discussion of the validity of this MVP's results will be provided.

Page 8 in Figure B.10 exhibits the customer and Deloitte feedback given at the conclusion of the project. Its purpose is to compare the total risk score at the end of the project to both parties' assessment of the project's success and appraisal regarding the completion on schedule and within budget. To clarify, contrary to the risk color scales, the tachometers below range from red (0%) to green (100%), inferring that what is being evaluated is the success and completion of the project on cost and schedule, so that the ideal score would be 100% in all 4 schemes.

With Figure 5.2 in mind, Pages 9 and 10 in Figures B.11 and B.12 in the same Appendix are for numerous projects overview, having a slicer system that lets the selection of many or even all projects aiming to compare their risk scores (Page 9) and to find patterns and trends regarding risk evolution (Page 10). The projects can be selected based on their unique project codes, as well as those whose overall risk scores fit inside a defined range.

These pages are less sophisticated than those pertaining to a single project analysis since there were insufficient projects to utilize as comparative examples in the dashboards. For this reason, the majority of the projects listed in the figures below are test cases.

5.4 Results

This part focuses on the critical review of the results acquired using the implemented tool. Meetings with project managers of different projects were held to brainstorm, reach conclusions on the actual accuracy of the framework, and inevitably identify opportunities for improvement. For client confidentiality, certain information on the projects under study cannot be disclosed.

5.4.1 Project 1

Project 1 pertains to a Portuguese insurance company's core system transformation service and is the first example of a real project to which the framework was applied and whose findings were compared against the project manager's intuition. This example depicts a project that has already been completed; however, to fill the MS Forms and collect the necessary data to complete the framework, archived historical information at the conclusion of each phase was used to indicate that each stage's evaluation had been done by the end of it. The main page for its analysis in Figure B.2 presents its final risk score: 53%. The chart in the same figure illustrates a rocky SDLC.

1. **Project Dimension**

   Regarding Figure B.3, project risk score was 47%, denoting level 3 on the risk scale. Through the Project Dimension Page analysis, it was anticipated from the beginning that the project would be challenging owing to the type of project (Retire), the involvement of

other parties, and the customer's unfamiliarity with the selected methodology, but mostly due to the client's lack of experience with IT projects. This aligns with the project manager's perception of the project also because of the complexity of the scope (Insurance Core Systems Transformation) and the type of project at hand (Advice). Since the project's initial conception, the mitigation actions described in the respective box of Figure B.3 have been taken, namely, a review of the IT budget for the project and alignment between projects outside the program scope were made a top priority, and product rationalization was delayed due to team unavailability.

2. **Inception Dimension**

   Despite the complexity of the project, the risk appeared to reduce during Inception, as seen in Figure B.4, which accurately depicts what occurred from the stakeholders' perspective. The most critical metric for this stage was the team structure. As shown by the radar chart illustrating the seniority distribution (115%), the seniority of the team was only half of what it should have been, with more staff than needed and a lack of senior members and managers. Once the team realized that their preparation for the development stage did not match expectations, a significant risk was identified in the comments.

3. **Development Dimension**

   Due to the IT team's lack of preparedness for this stage, there was increased alertness and a poor outcome in this stage. In conclusion, the amount of delivered scope and the delivery delay were the most significant risk factors. This phase was approximately 30% behind the planned timeline.

4. **CAT Dimension**

   Regarding the first testing dimension in Figure B.7, the alertness began to increase because CAT risk score attained a score of 62% (level 4). At the conclusion of the phase, execution and pass rates were far from optimal for a "go to the next phase" decision, despite the delivery delay already being noted (orange rectangle). However, the decision to proceed to the SIT phase was based on the low percentage of critical defects and the mention in the comments that the subsequent stage would have a greater percentage of automated tests, notwithstanding the riskiness of this stage. The extremely low execution rate is likely due to the restricted availability of testing environments (red rectangle). The reason for the variance in team skills is that the technical resources did not belong to the team supervised by Deloitte, so its composition was not embedded in the framework.

5. **SIT Dimension**

   At this point (Figure B.8), the instability has been established: level 5 on the risk scale is justified by critical defects, poor execution and pass rates, and severe delivery delay. Likewise, the large variance in team functionalities is attributable to not considering the developers as being outside Deloitte responsibility. This stage failure is the outcome of

advancing from the CAT stage before fulfilling the minimum quality requirements. The measure used to mitigate this issue was to greatly postpone the UAT stage; hence, the most impacted project triangle vertex was "time".

6. **UAT Dimension**

   The risk score decreases, but is still unsatisfactory as seen in Figure B.9. Due to the instability caused by several components, adjustments to mitigate the risk were taken, including a trial production phase with a small number of users.

7. **Project Analysis and Final Feedback**

   By the analysis of Figure B.2, it could be foreseen that the project would jeopardize some initial requirements. As concluded in the final feedback page (Figure B.10), the project was successful both in Deloitte's and the customer's opinion, however, there were several shortcomings. According to the Project Management Triangle (Figure 2.5), to preserve the bulk of the project's scope with quality, the timeline and budget had to be compromised, resulting in a substantial discrepancy from the initial plan. It was planned to be an 8-month project, but it ended up lasting 1 year and 6 months.

   The framework was in reality matched with the project workflow and outcome, anticipating the most hazardous phases and identifying the most significant risk factors contributing to them in accordance with stakeholders' perspectives.

5.4.2 Project 2

This second example of an actual assessment refers to the implementation of a digital banking solution in Saudi Arabia to provide a more accurate evaluation of the framework by contrasting two distinct engagement sectors, service lines, and locations. As the project follows an Agile delivery approach, both the development and CAT phases are occurring concurrently. Regarding its main page (Figure B.13), the current risk score at the moment is 49% (third level) and it aligns with the project manager's opinion. Nevertheless, this project is enormous, and in addition to Deloitte Portugal, other Deloitte firms, the bank, and even other consultancy companies are involved, making it a challenge. However, this framework was created to assess the role of the Portuguese firm in the project. Thus, the project analysis relates to the deployment of the digital banking core.

1. **Project Dimension**

   Since the project has a large budget and team assigned, stakeholders identify dependencies between teams and in decision-making as major concerns. The risk for Project dimension is level 3 (Figure B.14, which matches the management team's opinion. The hazard is related to the type of service and project itself, as well as the fact that many other parties are involved and that it is located offshore.

2. **Inception Dimension**

   Concerning the first process dimension in Figure B.15, the framework estimated a low Inception risk (level 2) by the end of the stage, and managers agreed. The delivery delay is the most important metric in this assessment, and it is due to the added scope during the stage itself, indicating a poor initial plan.

3. **Development Dimension**

   As shown in Figure B.16, this stage dimension raises awareness of some metrics, reaching level 3 on the risk scale. The amount of scope delivered at the time (development point: halfway) and the number of user stories assigned to each developer is higher than ideal. However, the delivery delay is significant, even if evaluating this metric is difficult due to poor, flexible, and high-level planning when there is no deadline for the first release to go to production. Requirements ambiguity and unprioritized features throughout the development are red flags that should be evaluated as metrics for future MVPs.

4. **CAT Dimension**

   According to the project manager, the bottleneck of this stage is the lack of testing capacity, which restricts testing strategy creation, actual testing of the components, and bug fixing. Component Acceptance Testing is almost non-existent at the moment, and acceptance criteria is roughly verified by developers while implementing. This represents a dangerous factor because, according to the Agile methodology theory, development and CAT sprints may occur concurrently. This way, this dimension risk at level 3, as ascertained by the framework, does not correspond to manager's concern, once this phase should be at the fourth level.

   The following metrics were handled in comparison to the number of features delivered at the time rather than the number of test cases because the testing strategy was not properly described. The red rectangles corresponding to execution rate and delivery delay metrics in Figure B.17 indicate a lack of capacity for testing and delivering on time. However, the pass rate that addresses execution quality is highlighted in orange, indicating that the small number of tests performed was successful. On the other hand, the percentage of critical defects rectangle is green, which is not expected given the lack of tests performed to assess this criterion. Bearing this in mind, an adjustment to the framework was proposed: create a relationship between the percentage of high risk bugs and the execution rate metrics, once the metric for bug criticality is in relation to the total number of found issues. To summarize, even though this dimension's risk score should be higher, it already works in the sense that it indicates the key risk factors and the suggested change would increase the CAT score.

5. **Project Analysis**

   To conclude the analysis in Figure B.13, the evolution of the risk score is not favorable at the time, and there are some indicators that require special attention throughout the stages. Nonetheless, given the inherent riskiness of the project at the starting point, the SDLC risk could not be much lower.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Fundamentally, the purpose of this thesis was to strengthen risk management capabilities, which had been attained by this point. As a means of increasing risk maturity, the work began with risk identification, continued with analysis, tracking, prevention, and concluded with an emphasis on visibility. The approach consists in presenting hard-edged data, as opposed to managers' assumptions, in risk reports, so that issues may be detected earlier on when they are considerably simpler to fix.

Initially, the risk assessment framework was built to exhaustively streamline the execution quality criteria during the software development life cycle in order to highlight the existing risk factors, including coverage, progress, capacity, and quality metrics. Its efficacy was verified by the fact that, throughout the discussion of the results, practically every issue was touched upon by one of the specified metrics.

A score scale was established to quantitatively and qualitatively conceptualize the output. To address metrics severity and weight their impacts in the final score, the algorithm was created. It was validated by the implementation of the Model Validation Pilot in real projects, which led to the conclusion that estimated scores align with managers' intuition and the obtained charts in Power BI emphasize the most critical factors in line with senior people's perceptions.

Therefore, the Power BI application was developed to make the IT delivery risk assessment available to all project managers, as well as to examine the framework's reliability based on a comparison of actual project outputs to managers' concerns regarding the riskiness of the project. With the introduction of this tool, several advantages emerged, including:

- Enhanced risk visibility and awareness of the presence of risk factors and lack in execution quality across all project teams at any phase of the SDLC;

- Alignment between clients and Deloitte by simplifying risk appraisal, no longer relying solely on management team opinion;

- The development of a standard assessment for Deloitte senior members to compare the progression of various ongoing projects;
- Support and provide a foundation for decision-making on required mitigation actions that influence projects' budget, schedule, and scope.

In conclusion, all of the project's established goals were accomplished in practice.

6.2 Future Work

On the one hand, regarding the MVP outcomes, it is essential to note that the more the framework is tested in real-world projects, the more feedback is received and the more adjustments are required to both tune the framework and optimize the algorithm. On the other hand, as a result of the success of this thesis work for the firm, several additional suggestions were made to be added to future MVPs, so expanding the project's value and usefulness, but most importantly its future vision projection.

Further developments of this thesis project would focus on the evolution of the risk score across the SDLC. Any project at any moment has an inherent risk score greater than zero, and there are stages that are notoriously riskier. It would be meaningful to evaluate the fluctuations between and even within each phase in order to develop a Deloitte benchmark for various project parameters, such as: geography, client, and engagement sector, among others. This would be a compilation of real data from past projects that would result in a "typical risk score" for that sort of project over a certain time period. This comparison would provide a more accurate perspective, as the target would be a realistic and data-driven baseline for a specific comparison factor. In essence, project-specific benchmarks might be created by combining benchmarks obtained for the same client, area, engagement industry, and other pertinent project characteristics. This risk baseline might be used to help customers understand that, despite the success of a project, it is not uncommon for problems to arise, particularly during the SIT stage, which has the greatest risk level on average.

With regard to the framework formulation, it would be interesting to develop additional linkages between different metrics, since the assessment criteria of some metrics are generalized and may, in some circumstances, alter in accordance with the values of other metrics. Moreover, the existing MVP only considers the last assessment filled to each dimension, however, it would be insightful to weighted average all the answers to the same dimension to obtain the metrics evaluation based on inputs from members with different roles, not just the project manager. Indeed, this would only be useful to metrics that lack severe concreteness.

Concerning the technological solution developed, it is pivotal that a solid database, more efficient data modeling, and more advanced front-end for data acquisition will be key improvements in forthcoming MVPs, as these were not a primary focus of this thesis validation pilot. These would optimize the whole BI system by reducing the number of tables in Power BI through more effective data modeling. Incorporating machine learning into future pilots would be extremely

insightful, particularly for the creation of the previously mentioned benchmarks and for the identification of possible trends and patterns that have not yet been identified.

Last but not least, to complement the report by employing Artificial Intelligence, mitigation actions at each stage level might be recommended automatically based on the most hazardous metrics. In light of this, variance forecasts for future project stages, particularly in terms of schedule and cost attending to the initial plan, might be predicted as an output of the assessment. Customers might be provided with a strong foundation for project workflow adjustments in this manner.

Bibliography

[1] World Economic Forum. The fourth industrial revolution: what it means, how to respond. https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/. Accessed in 12.04.2022.

[2] Jim Johnson and Hans Mulder. Endless modernization. 2020.

[3] Anna V Lebedeva and Anna I Guseva. Cognitive maps for risk estimation in software development projects. In *Biologically Inspired Cognitive Architectures Meeting*, pages 295–304. Springer, 2019.

[4] John McManus. *Risk management in software development projects*. Routledge, 2012.

[5] Júlio Menezes Jr, Cristine Gusmão, and Hermano Moura. Defining indicators for risk assessment in software development projects. *Clei electronic journal*, 16(1):11–11, 2013.

[6] Deloitte. Deloitte. https://www2.deloitte.com/global/en.html. Accessed in 11.04.2022.

[7] Ibraheem M Alharbi, Adel A Alyoubi, Majid Altuwairiqi, and Mahmoud Abd Ellatif. Analysis of risks assessment in multi software projects development environment using classification techniques. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 845–854. Springer, 2021.

[8] Fernando Almeida, Jorge Simões, and Sérgio Lopes. Exploring the benefits of combining devops and agile. *Future Internet*, 14(2):63, 2022.

[9] Adetokunbo AA Adenowo and Basirat A Adenowo. Software engineering methodologies: a review of the waterfall model and object-oriented approach. *International Journal of Scientific & Engineering Research*, 4(7):427–434, 2013.

[10] Youssef Bassil. A simulation model for the waterfall software development life cycle. *arXiv preprint arXiv:1205.6904*, 2012.

[11] Apoorva Srivastava, Sukriti Bhardwaj, and Shipra Saraswat. Scrum model for agile methodology. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 864–869. IEEE, 2017.

[12] Ralph Hughes. *Agile data warehousing for the enterprise: a guide for solution architects and project leaders*. Newnes, 2015.

[13] Holger Timinger, Markus Schmidtner, and Finn Reiche. A framework for the construction and tailoring of engineering development process models. *IEEE Transactions on Engineering Management*, 2022.

[14] Luyin Zhao and Sebastian Elbaum. Quality assurance under the open source development model. *Journal of Systems and Software*, 66(1):65–75, 2003.

[15] Jiantao Pan. Software testing. *Dependable Embedded Systems*, 5:2006, 1999.

[16] Paul C Jorgensen. *Software testing: a craftsman's approach*. Auerbach Publications, 2013.

[17] Tim A Majchrzak. Best practices for the organizational implementation of software testing. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2010.

[18] Ian Gilchrist. Software testing—testing across the entire software development lifecycle. by gerald d everett & raymond mcleod jr. published by ieee press and wiley interscience, hoboken, nj, usa isbn: 978-0-471-79371-7, 260 pages. price $47.50, 2009.

[19] Ming-Chang Lee. Software quality factors and software quality metrics to enhance software quality assurance. *British Journal of Applied Science & Technology*, 4(21):3069–3095, 2014.

[20] Julien Pollack, Jane Helm, and Daniel Adler. What is the iron triangle, and how has it changed? *International journal of managing projects in business*, 2018.

[21] Stefan Wagner. Software product quality control. 2013.

[22] Tsvetelina Mladenova. Software quality metrics–research, analysis and recommendation. In *2020 International Conference Automatics and Informatics (ICAI)*, pages 1–5. IEEE, 2020.

[23] Yun Fu, Minqiang Li, and Fuzan Chen. Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. *International Journal of Project Management*, 30(3):363–373, 2012.

[24] Júlio Menezes, Cristine Gusmão, and Hermano Moura. Risk factors in software development projects: a systematic literature review. *Software Quality Journal*, 27(3):1149–1174, 2019.

[25] Gurdev Singh, Dilbag Singh, and Vikram Singh. A study of software metrics. *IJCEM International Journal of Computational Engineering & Management*, 11(2011):22–27, 2011.

[26] Premal B Nirpal and KV Kale. A brief overview of software testing metrics. *International Journal on Computer Science and Engineering*, 3(1):204–2011, 2011.

[27] C Ravindranath Pandian. *Applied software risk management: A guide for software project managers*. Auerbach Publications, 2006.

[28] Martin Hopkinson. *The project risk maturity model: Measuring and improving risk management capability*. Routledge, 2017.

Appendix A

Power BI Data Model: UML Diagram

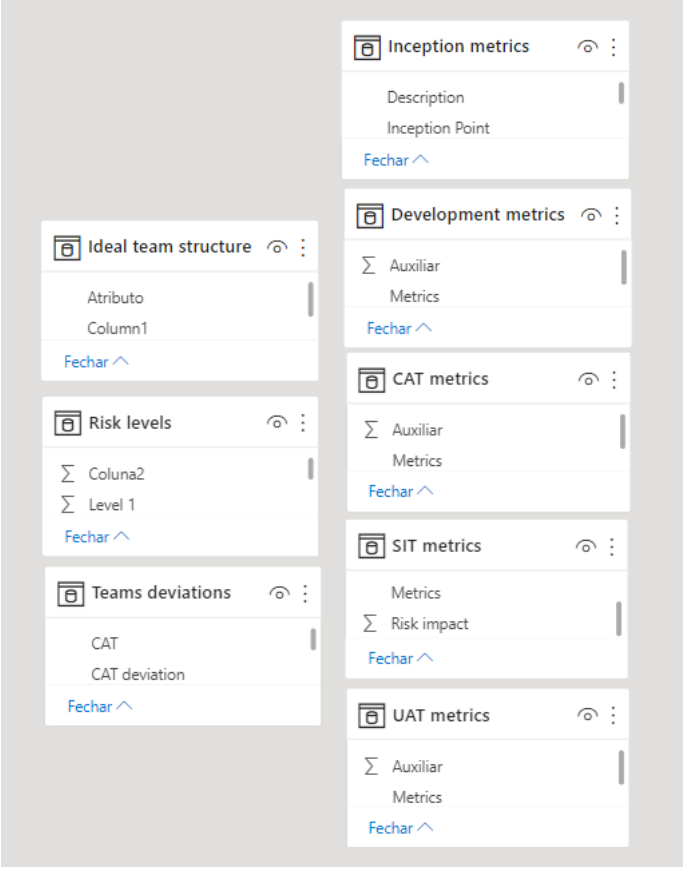


Figure A.1: Power BI Data Model: Data Tables (1)

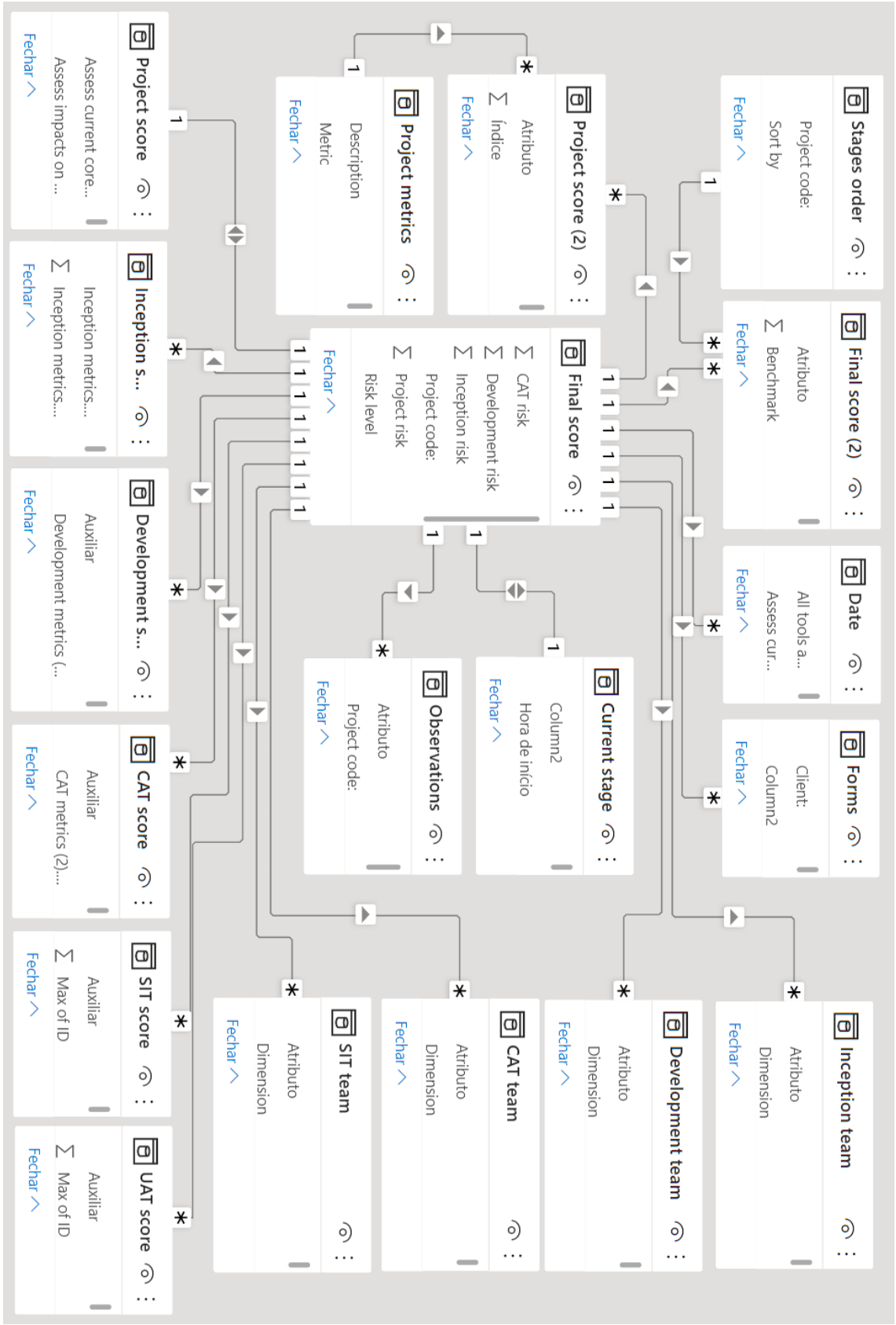


Figure A.2: Power BI Data Model: UML Diagram - Data Tables (2)

Appendix B

Power BI Report

B.1 Project 1



Figure B.1: Power BI App: Home Page

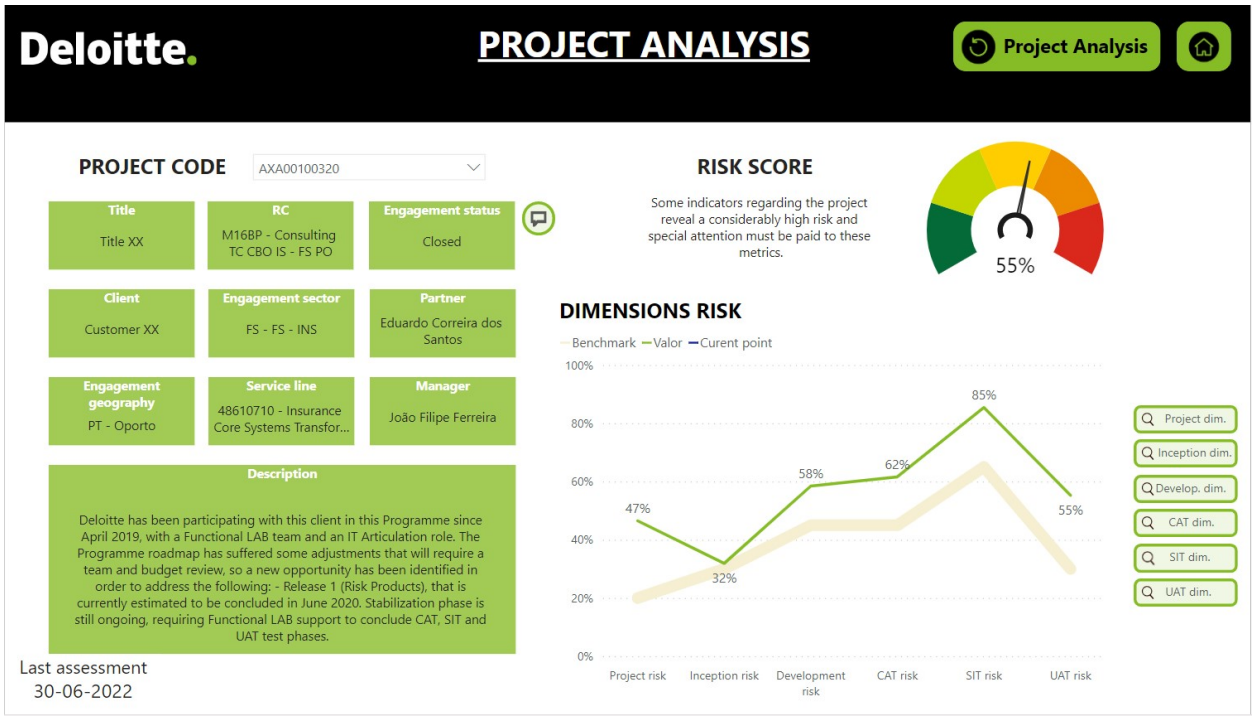


Figure B.2: Project 1: Project Analysis

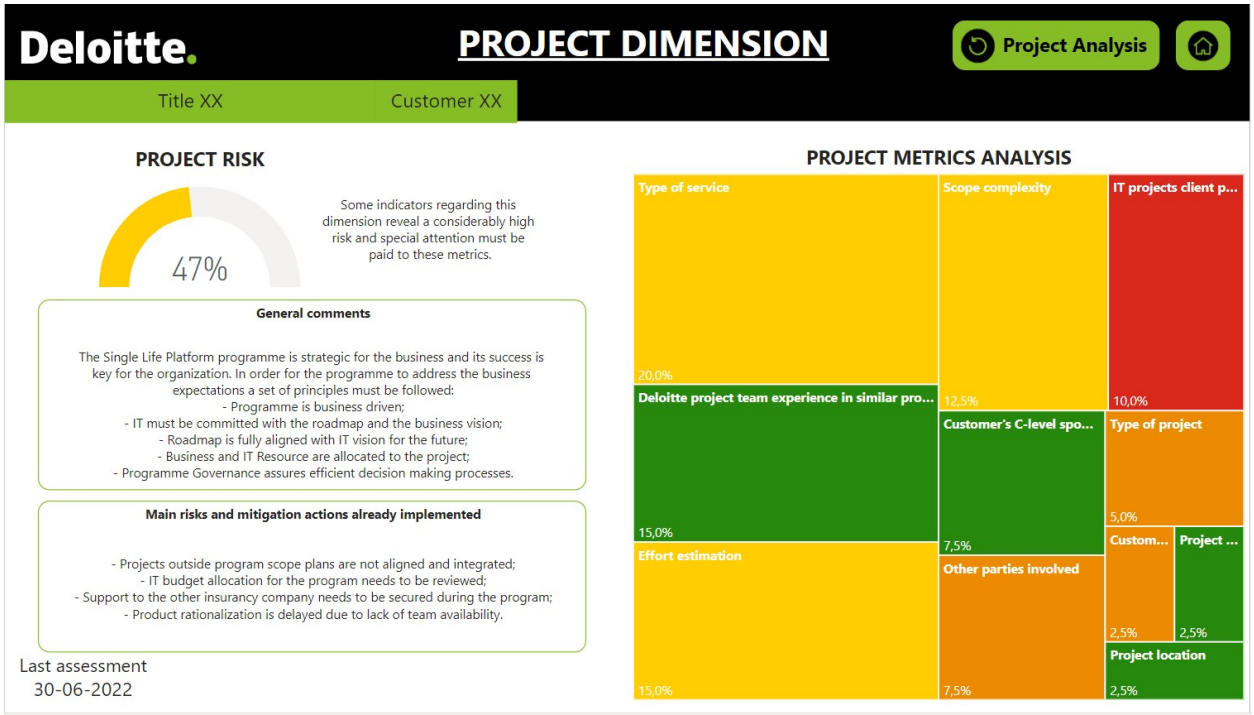


Figure B.3: Project 1: Project Dimension

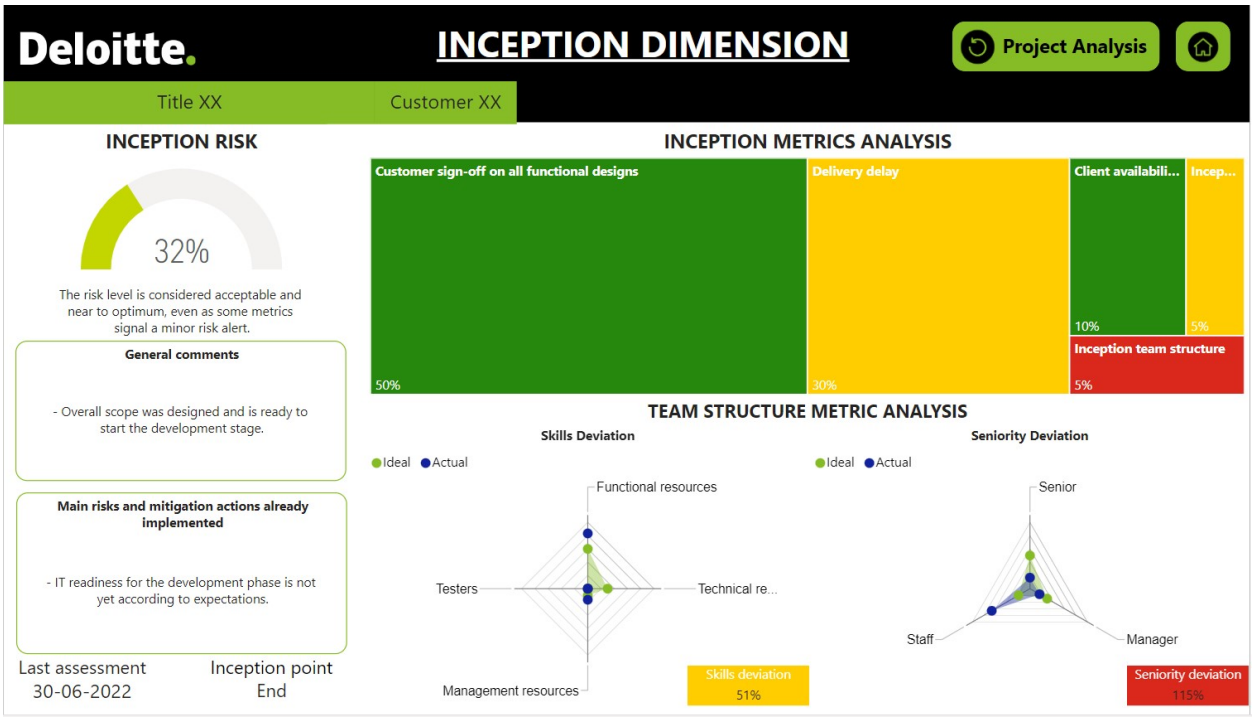


Figure B.4: Project 1: Inception Dimension

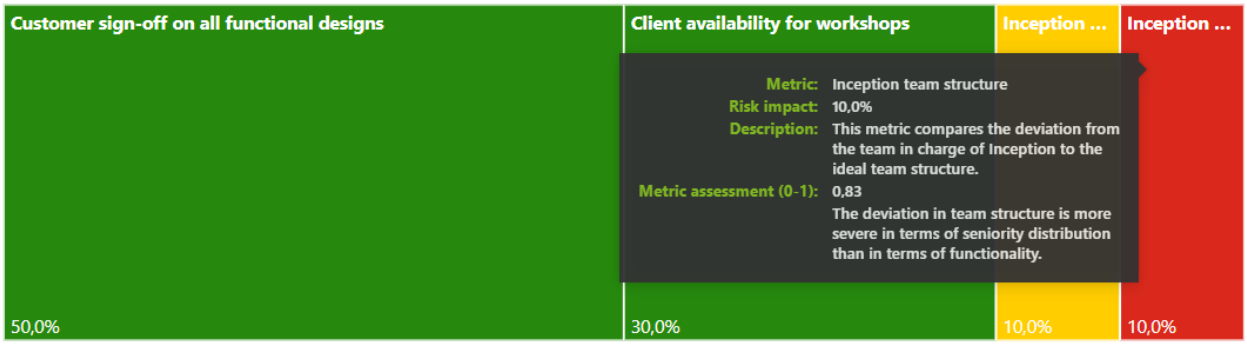


Figure B.5: Power BI Report: Mousing Over Functionality

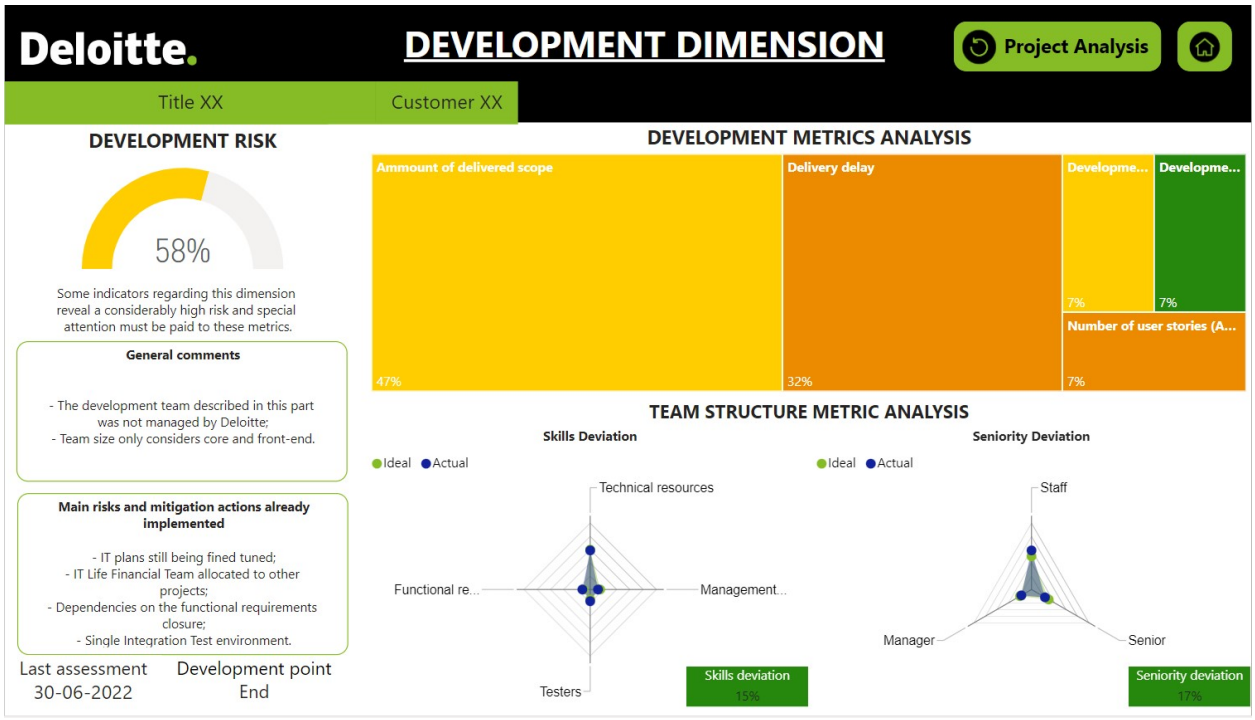


Figure B.6: Project 1: Development Dimension

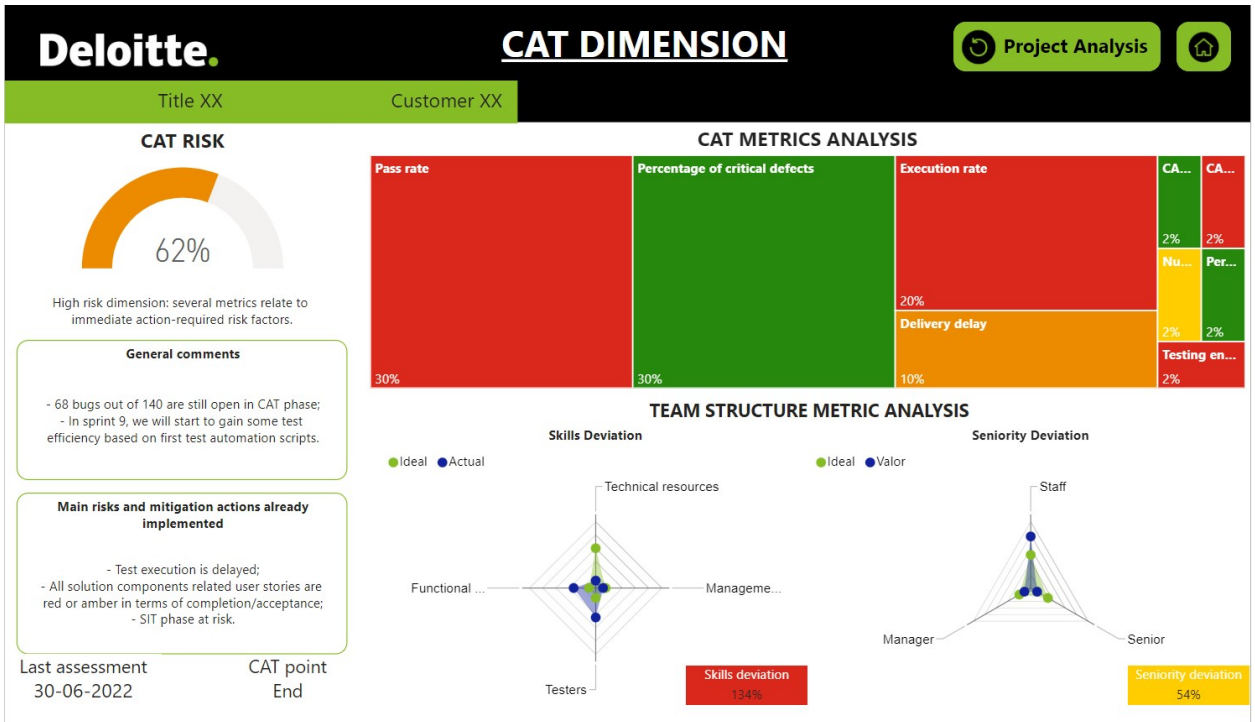


Figure B.7: Project 1: CAT Dimension

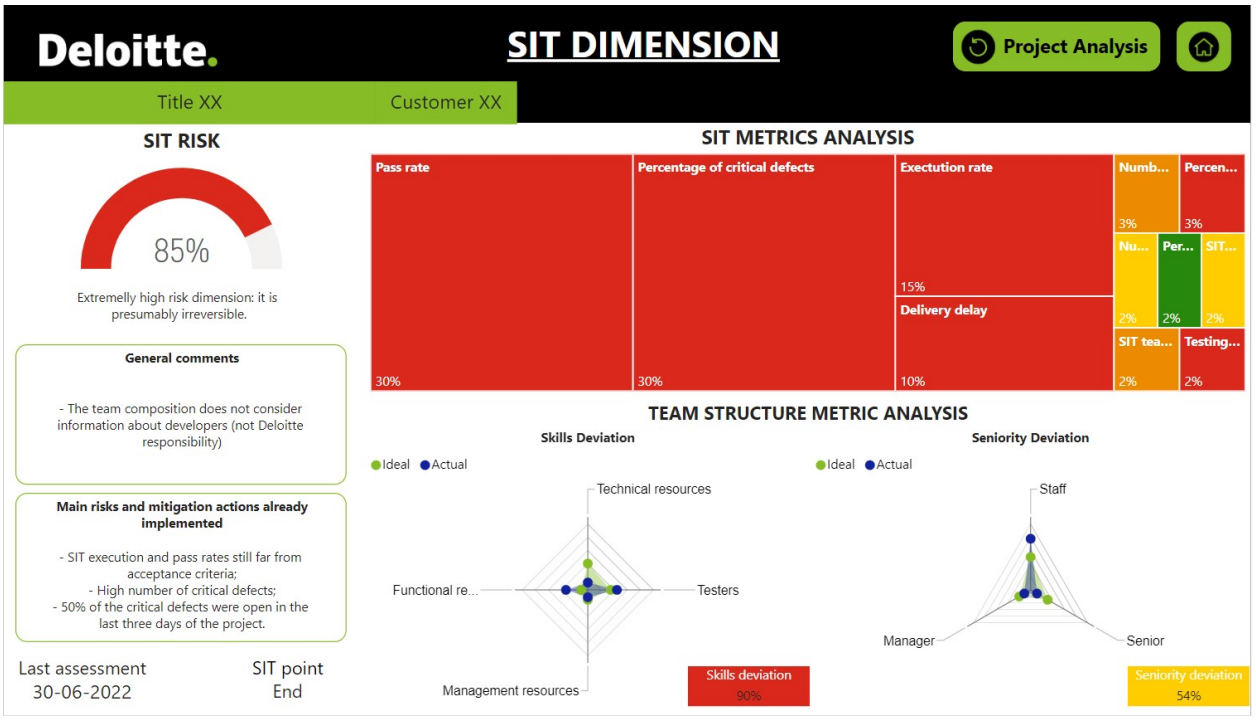


Figure B.8: Project 1: SIT Dimension

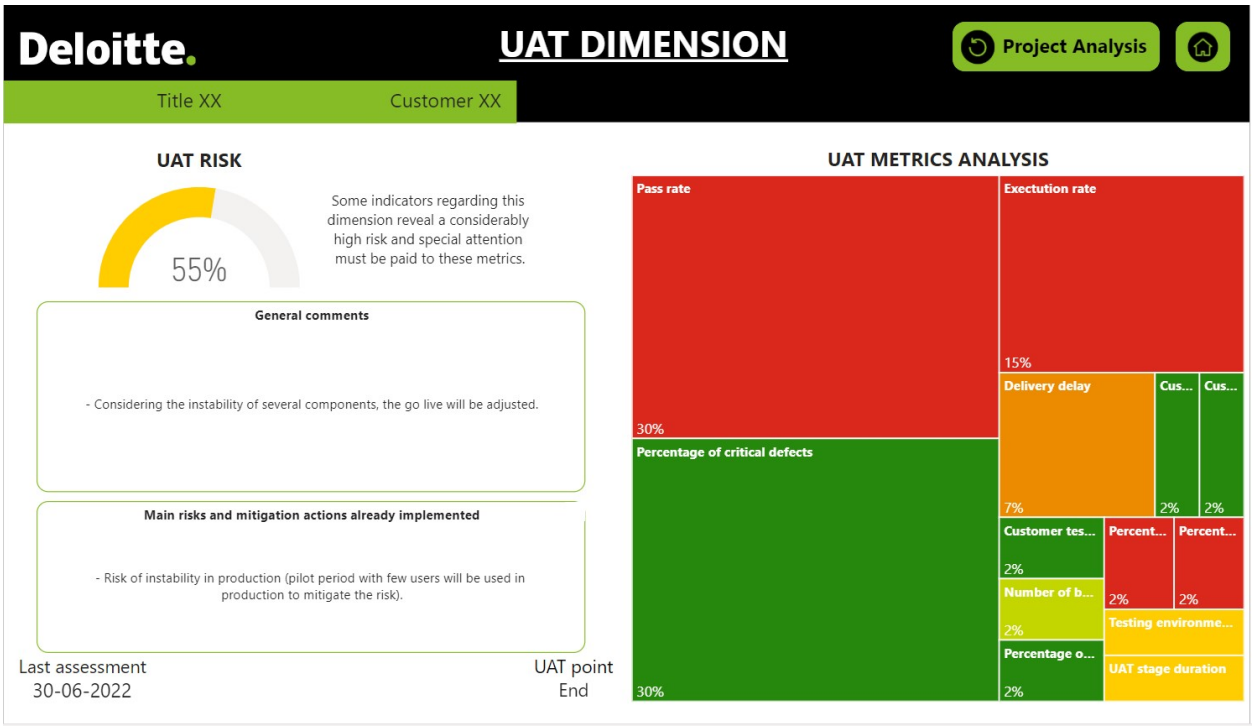


Figure B.9: Project 1: UAT Dimension

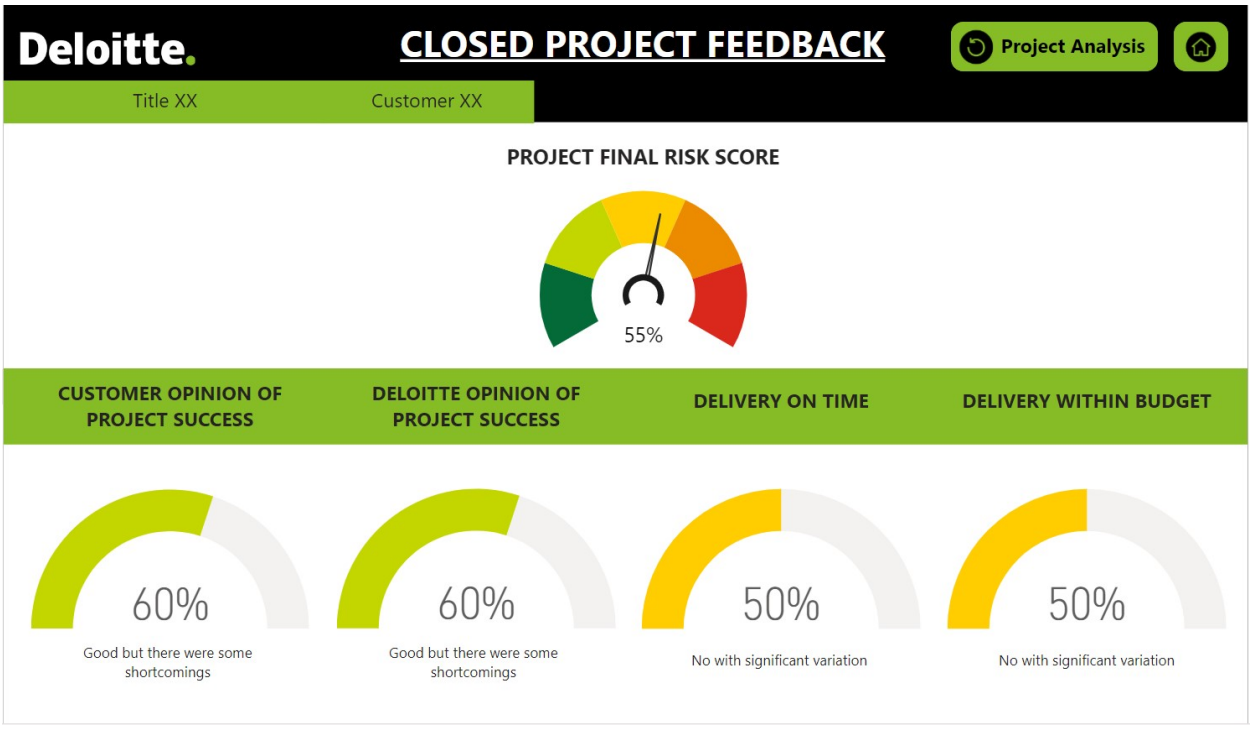


Figure B.10: Project 1: Closed Project Feedback

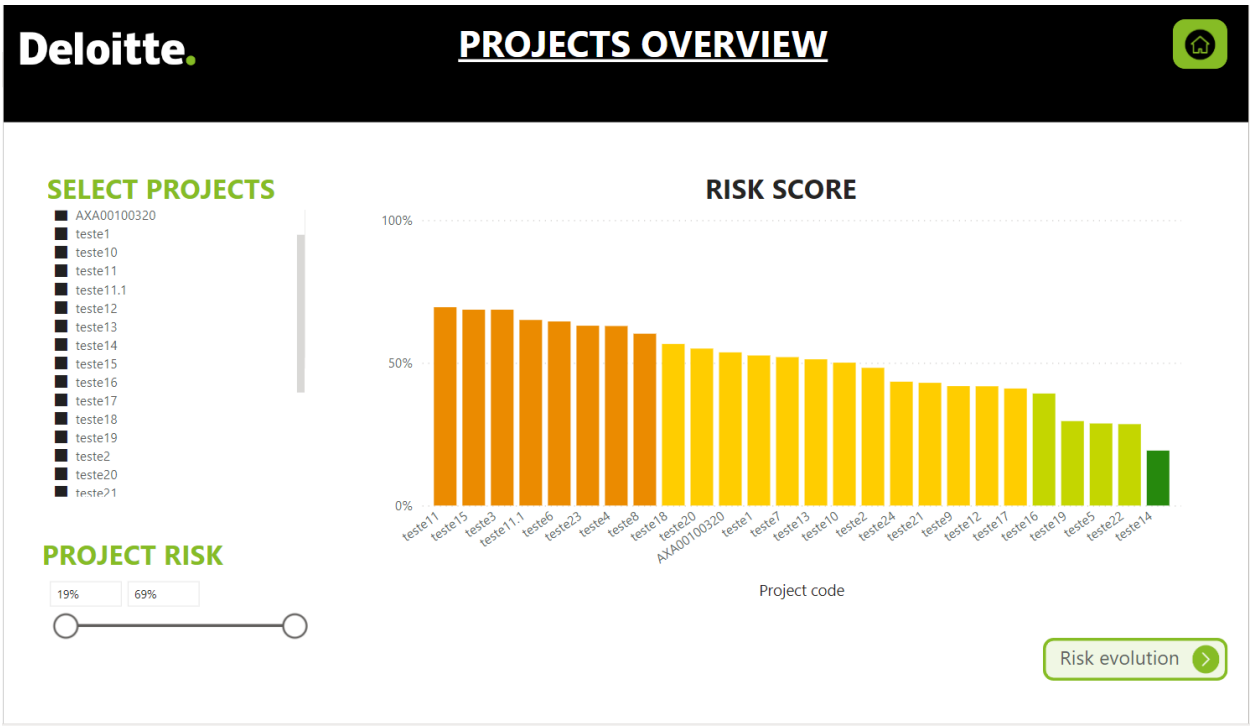


Figure B.11: Power BI App: Projects Overview

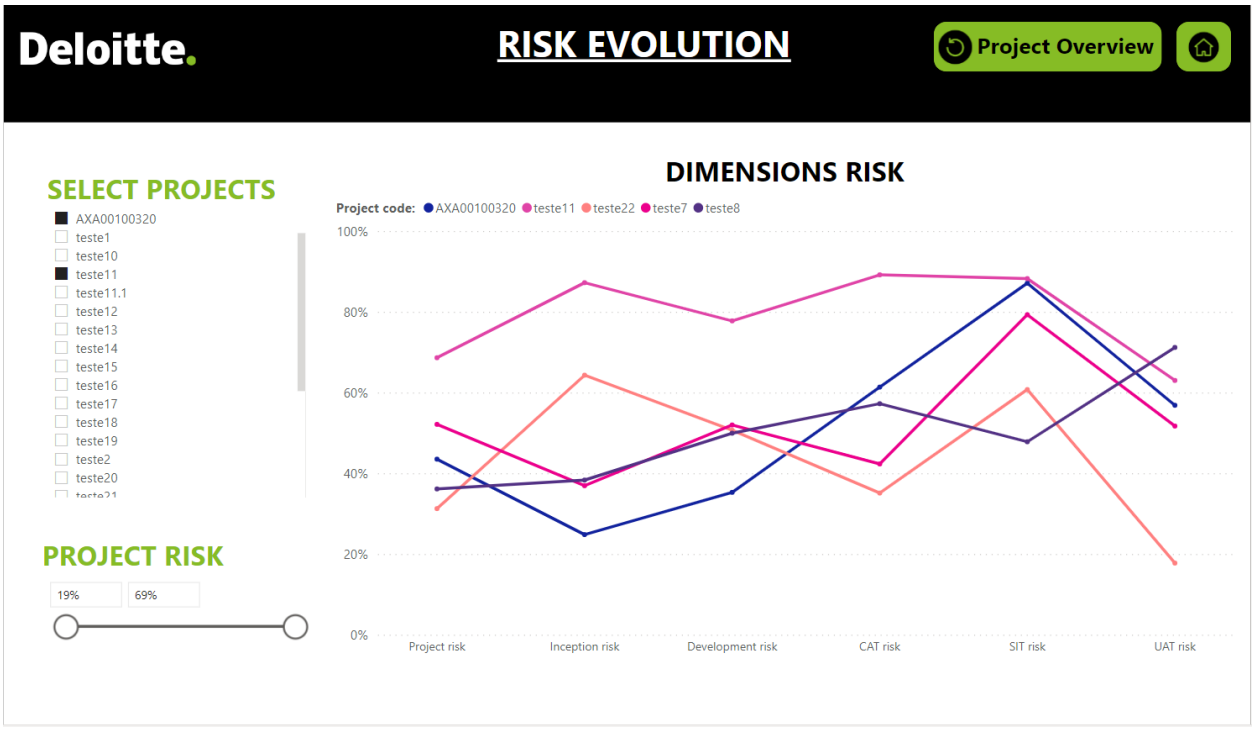


Figure B.12: Power BI App: Risk Evolution

B.2 Project 2

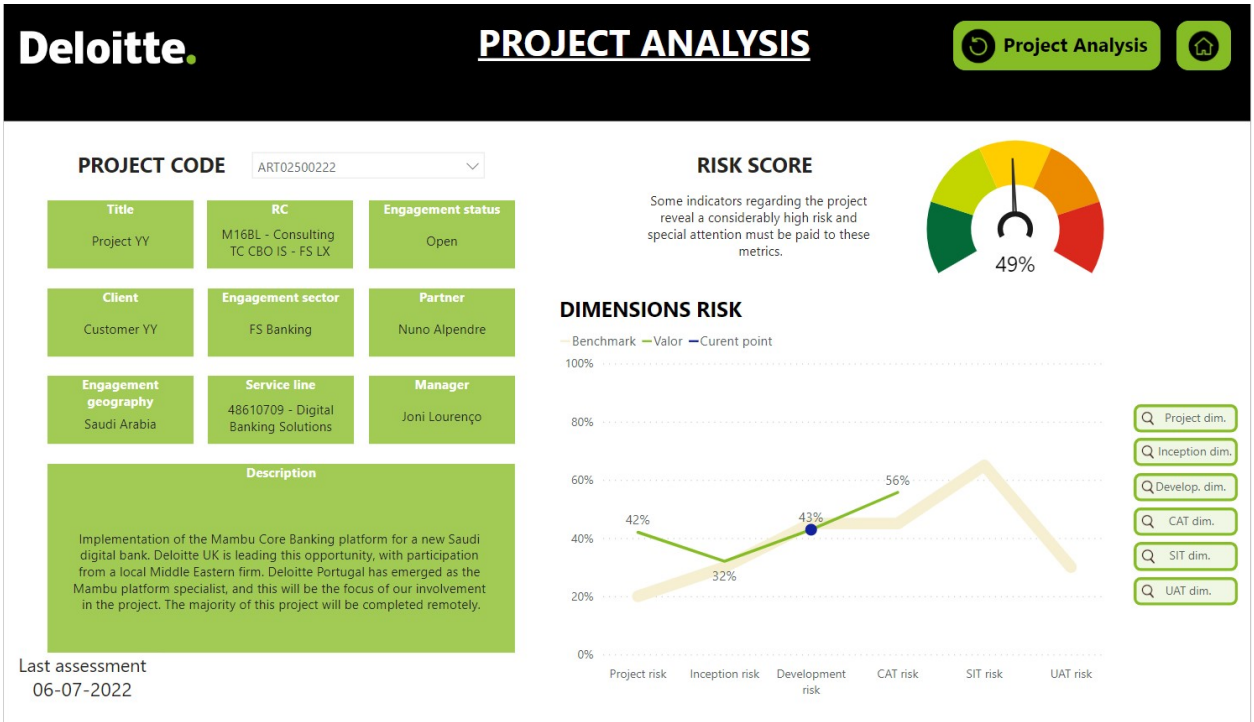


Figure B.13: Project 2: Project Analysis

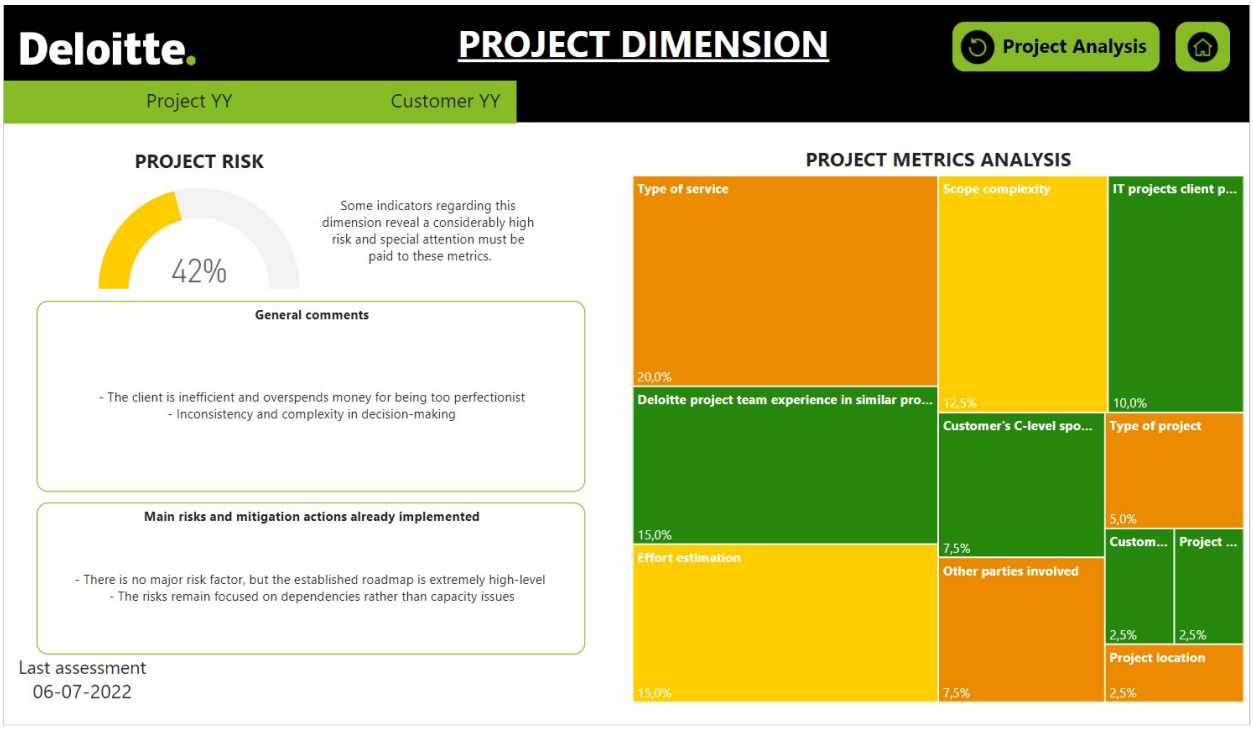


Figure B.14: Project 2: Project Dimension

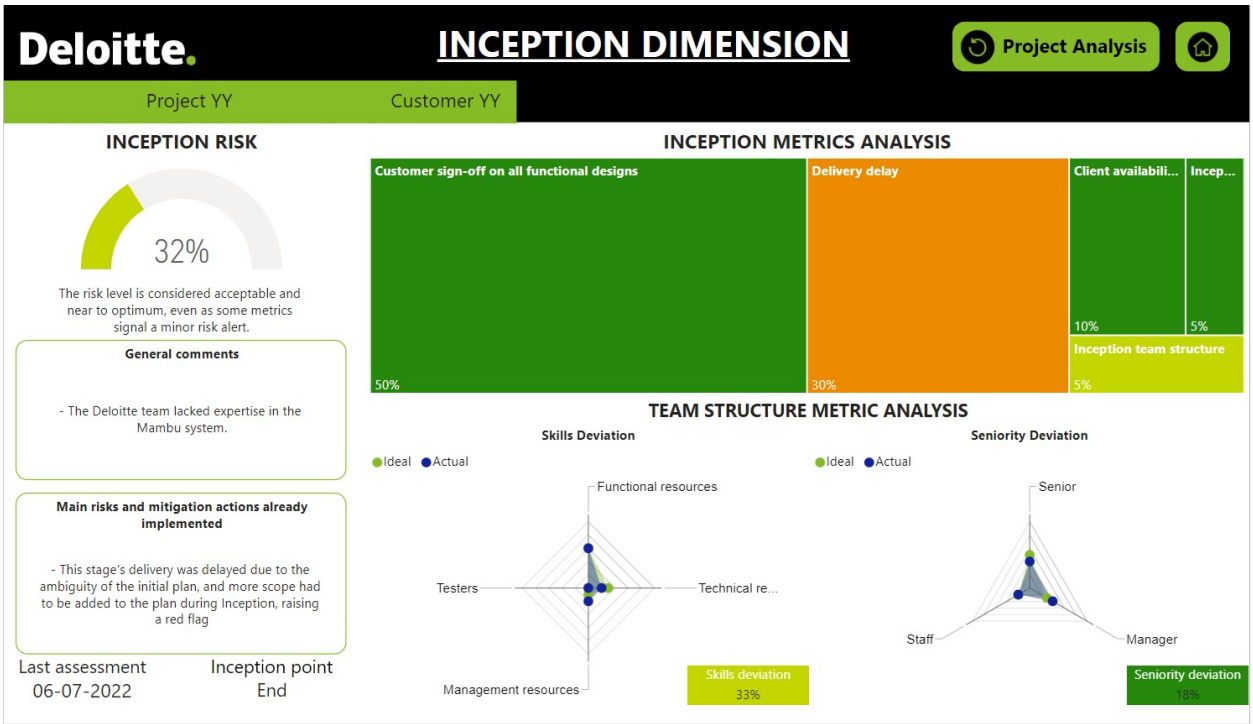


Figure B.15: Project 2: Inception Dimension

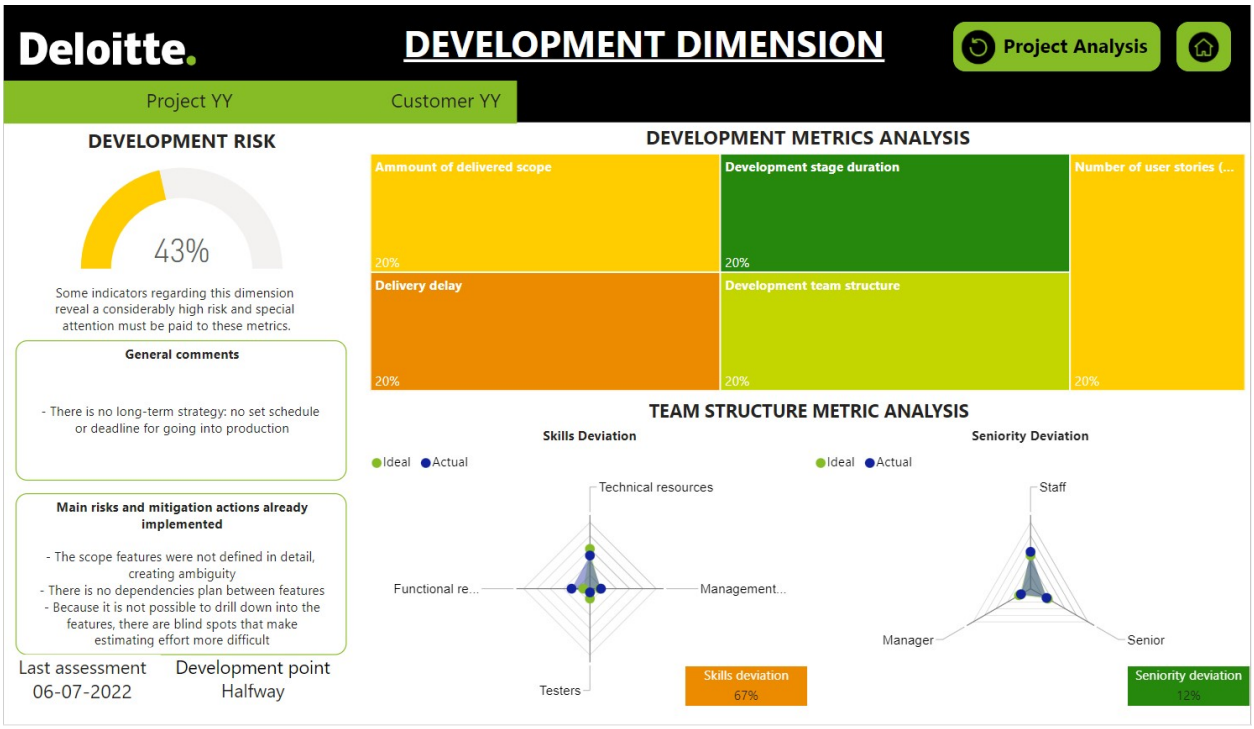


Figure B.16: Project 2: Development Dimension

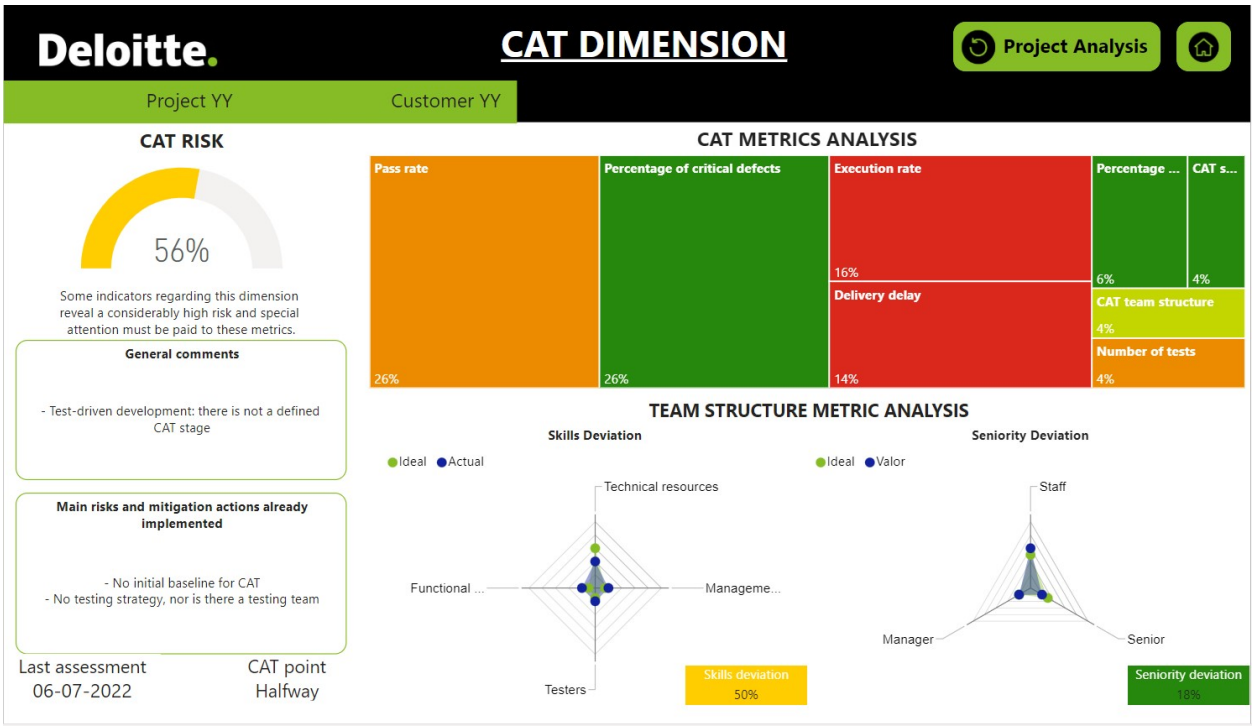


Figure B.17: Project 2: CAT Dimension