|  |  |  |
| --- | --- | --- |
|  | **CAN THO UNIVERSITY SOFTWARE CENTER** |  |
| **MEKONG DELTA - APTECH** |

**Aptech Certified Computer Professional (ACCP)**

**SEMESTER III**

**Professional Issues in IT**

**[ TBay- a safety critical system ]**

|  |  |
| --- | --- |
| **Instructor:** | **Class: CP2296F02** |
| Mr. Luu Tien Dao | **Students:**  A19084 - Phạm Quốc Anh |
|  |

Can Tho, 4/2023

[Scenario: A safety critical system 4](#_Toc133608527)

[Task 1 – Ethical and Standards 5](#_Toc133608528)

[1. List possible ethical actions 5](#_Toc133608529)

[2. The professional standards 5](#_Toc133608530)

[Task 2 – Project Management Lifecycle and Strategy 7](#_Toc133608531)

[1. The Agile methodology 7](#_Toc133608532)

[2. Five (5) advantages and Five (5) disadvantages of the aglie 8](#_Toc133608533)

[Task 3 – RISK 10](#_Toc133608534)

[1. FIVE (5) methods of identifying risks when developing software 10](#_Toc133608535)

[2. THREE (3) main risks in the Tbay Airline scenario 10](#_Toc133608536)

[3. Diagram shows a simple approach to the Risk Management Process 11](#_Toc133608537)

[4. Diagram shows an risks that i was identified in the scenario in part TWO (2) 12](#_Toc133608538)

[5. International Standards for Risk Management 12](#_Toc133608539)

[Task 4 – Software Deployment 14](#_Toc133608540)

[1. There ways to releasing software 14](#_Toc133608541)

[2. The most feasible method for releasing software 14](#_Toc133608542)

[Task 5 – Service Management within the IT sector 16](#_Toc133608543)

[1. Outsourcing 16](#_Toc133608544)

[2. In-house development 16](#_Toc133608545)

[Task 6 – Software Quality 18](#_Toc133608546)

[1. There are two ways to measure software quality 18](#_Toc133608547)

[2. ISO 9126 18](#_Toc133608548)

[Reference websites 20](#_Toc133608549)

# **Scenario: A safety critical system**

A system has been developed for the scheduling of maintenance at an airline company called TBay Airlines.  
 A trade union spokesperson for the airline maintenance workers argues that the to performing functions in in the new computer system there should be more interactive human judgment if safety is to be ensured. The program is one which schedules maintenance, and can reassign alternative aircraft when emergencies arise. The airline maintenance staff would be able to interact with the system, but only on a basic level.  
 The systems analyst, under whose direction the program was written, is aware that not all operational factors have been taken into consideration in the program, but he had been assured by management that the program conforms to all the requirements of the IATA (International Air Transport Association). In his opinion the program should have contained more interactive human judgement, so that a person is involved in the final decision making, but the company was not prepared to go to the additional expense of a fully interactive system.  
 When testing his program, he could not think of a scenario whereby the system couldn’t meet a safety condition. When he was asked to testify in an inquiry dealing with the trade union's complaint, he did not volunteer his opinion on how the system should have been designed as he could not think of any negative outcomes that the system couldn’t handle.  
 Primary Participants: Systems analyst• designed program knowing that not all operational factors had been taken into account  
• informed management of his concerns  
• failed to volunteer his opinion when needed  
 Airline management• disregarded analyst's concerns  
• indicated ‘decision rules’ used in the program conform to IATA requirements  
• opposed making system fully interactive  
 Implied Participants: Programming staff• implemented systems analyst's design

# Task 1 – Ethical and Standards

1. List possible ethical actions :

* **Airline Management**: They could have invested more in a fully interactive system that allows more human judgment and input in the maintenance scheduling and reassignment decisions. This would have increased the safety and reliability of the system, as well as the trust and satisfaction of the maintenance workers. They could have also consulted with the trade union and the systems analyst to understand their concerns and expectations, and to reach a mutually agreeable solution. They could have followed the ethical principles of beneficence (doing good), non-maleficence (avoiding harm), and justice (treating people fairly) **[1]**.
* **Project Manager**: They could have communicated more openly and honestly with the systems analyst and the airline management about the limitations and assumptions of the program. They could have also ensured that the program was tested thoroughly and rigorously under various scenarios and conditions, and that any potential risks or errors were identified and mitigated. They could have followed the ethical principles of integrity (being honest), accountability (being responsible), and excellence (being competent) **[2]**.
* **Systems Analyst:** They could have expressed their opinion on how the system should have been designed more strongly and persistently to the project manager and the airline management. They could have also documented their concerns and recommendations in a formal report or memo, and shared it with relevant stakeholders. They could have also testified truthfully and completely in the inquiry, and disclosed any information that could affect the safety or quality of the system. They could have followed the ethical principles of autonomy (being independent), fidelity (being loyal), and respect (being courteous) **[2]**.
* **Programming Staff**: They could have followed the specifications and instructions of the systems analyst faithfully, but also raised any questions or issues they encountered during the implementation process. They could have also tested their own work for errors or bugs, and reported any problems or defects to the systems analyst or the project manager. They could have followed the ethical principles of diligence (being careful), competence (being skilled), and professionalism (being respectful) **[2].**

1. Some of the professional standards that are required:

* **Code of Ethics and Professional Conduct**: This document outlines the expectations and obligations of PMI members and credential holders to act ethically and professionally in their work. It covers four main values: responsibility, respect, fairness, and honesty **[3]**.
* **Project Management Body of Knowledge (PMBOK)**: This document provides a comprehensive set of guidelines and best practices for managing projects across various domains and industries. It covers 10 knowledge areas and five process groups that are essential for effective project management **[4]**.
* **Organizational Project Management Maturity Model (OPM3):** This document provides a framework and assessment tool for measuring and improving the project management capabilities of an organization. It helps organizations align their projects with their strategic goals and objectives **[5]**.

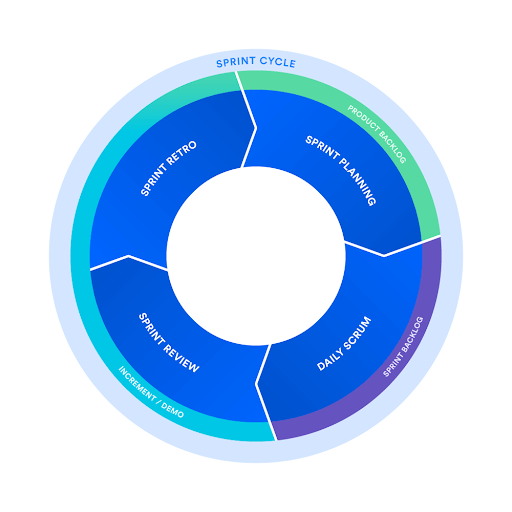
**For example:**

* The Project Manager could have followed the PMBOK guidelines to plan, execute and monitor the project effectively. They could have also used the OPM3 framework to assess and identify areas for improvement. They could have adhered to the Code of Ethics and Professional Conduct by communicating honestly with the Systems Analyst and the Airline Management, and by being accountable for their actions and decisions.
* The Systems Analyst could have followed the PMBOK guidelines to define, analyze, design, test, and implement the system. They could have used the OPM3 framework to evaluate the alignment of their system with the organizational. They could have also adhered to the Code of Ethics and Professional Conduct by expressing their opinion to improve the system.

# Task 2 – Project Management Lifecycle and Strategy

1. The Agile methodology:

Agile methodology aims to deliver value to customers faster and with fewer headaches, by responding to change quickly and collaborating with stakeholders. One of the most common agile frameworks is scrum, which involves a cross-functional team working on a series of sprints, each lasting one to four weeks.



The significance of each phase as applied to the scenario is:

* **Product backlog**: This is a list of features or requirements that the product owner wants to deliver to the customers. The product owner prioritizes the backlog based on the value and urgency of each item. In the scenario, the product owner could be the airline management or a representative of them, who defines the scope and vision of the system.
* **Sprint planning**: This is a meeting where the scrum team decides what items from the product backlog they can complete in the next sprint. The product owner clarifies the expectations and acceptance criteria for each item, while the development team estimates the effort and complexity involved. The scrum master facilitates the meeting and ensures that everyone is on the same page. In the scenario, the scrum team could include the project manager, the systems analyst, and the programming staff, who plan how to design, test, and implement the system.
* **Sprint backlog**: This is a subset of the product backlog that contains the items that the development team commits to work on during the sprint. The development team breaks down each item into smaller tasks and assigns them to team members. The sprint backlog is updated daily to reflect the progress and changes made by the team. In the scenario, the sprint backlog could be a list of system features or functions that need to be developed or modified during each sprint.
* **Daily scrum**: This is a short meeting where the development team updates each other on what they did yesterday, what they plan to do today, and what impediments or issues they are facing. The scrum master helps remove any obstacles or dependencies that might hinder the team’s work. The daily scrum helps improve communication, collaboration, and transparency within the team. In the scenario, the daily scrum could be a way for the development team to coordinate their work and report any problems or risks they encounter.
* **Sprint review**: This is a meeting where the development team demonstrates what they have completed during the sprint to the product owner and other stakeholders. The product owner checks if each item meets the acceptance criteria and provides feedback or suggestions for improvement.
* **Sprint retrospective**: This is a meeting where the scrum team reflects on what went well and what could be improved in the sprint. The team identifies the strengths and weaknesses of their process, tools, communication, and collaboration. The team also agrees on some action items to implement in the next sprint. The sprint retrospective helps the team learn from their experience and continuously improve their performance. In the scenario, the sprint retrospective could be a way for the scrum team to share their feedback and suggestions for enhancing the system or the project management.
* **Increment**: This is the potentially shippable product or system that the development team delivers at the end of each sprint. The increment should be functional, usable, and valuable to the customers. The increment should also be integrated with the previous increments to form a coherent whole. The increment represents the progress and outcome of the team’s work. In the scenario, the increment could be a version of the system that meets some of the requirements and expectations of the airline management and the trade union.

1. Five (5) advantages and Five (5) disadvantages of the aglie:

**Advantages:**

* **Flexibility and adaptability**: The agile methodology allows the software development team to respond to changes in requirements, customer feedback, and demand quickly. The team can deliver value to the customers faster and with fewer headaches, by adjusting their strategy and priorities based on the feedback they receive **[6]**.
* **Customer collaboration**: The agile methodology emphasizes customer collaboration over contract negotiation. The team works closely with the customers and other stakeholders to understand their needs and expectations, and to deliver solutions that meet or exceed them. The team also seeks feedback from the customers frequently and incorporates it into their work **[6]**.
* **Continuous** **improvement**: The agile methodology encourages continuous improvement of the product, the process, and the team. The team reflects on what went well and what could be improved after every sprint, and implements action items to enhance their performance. The team also delivers working software in small increments, which allows them to test, validate, and improve their product **[6]**.
* **Team empowerment**: The agile methodology empowers the team to self-organize and make decisions on how to best accomplish their work. The team has a high degree of autonomy, creativity, and ownership of their work. The team also communicates and collaborates effectively within themselves and with other stakeholders **[6]**.
* **Quality assurance**: The agile methodology ensures quality assurance of the product by applying various practices such as test-driven development, pair programming, code reviews, automated testing, etc. The team ensures that the product meets the acceptance criteria and the quality standards before delivering it to the customers. The team also fixes any defects or bugs as soon as they are detected **[7]**.

**Disadvantages:**

* **Lack of predictability**: The agile methodology depends heavily on flexibility and adaptability. This means that it can be difficult to predict efforts such as cost, time, and resources at the beginning of the project. It can also be difficult to measure progress since agile methods deliver in increments rather than in a final product **[8]**.
* **Lack of documentation**: The agile methodology values working software over comprehensive documentation. This means that the team may not produce enough documentation for the product or the process, which can cause problems for future maintenance, support, or integration. Documentation may also be needed for compliance or audit purposes **[8]**.
* **Lack of control**: The agile methodology gives a lot of control to the team and the customers over the direction and scope of the project. This can lead to scope creep or feature creep, where more features or requirements are added without proper planning or prioritization. This can also lead to conflicts or disagreements among stakeholders over expectations or deliverables **[8]**.
* **High dependency**: The agile methodology relies heavily on communication and collaboration among team members and stakeholders. This means that the team needs to have a high level of trust, commitment, and accountability for their work. The team also needs to have a high level of skill, experience, and expertise for their work. If any of these factors are missing or weak, the project may suffer **[8]**.
* **High turnover**: The agile methodology requires a lot of involvement and engagement from the team and the customers throughout the project. This can cause burnout or fatigue for some people who may not be able to cope with the fast pace or frequent changes. This can also cause turnover or attrition for some people who may not be satisfied with their work or role **[9]**.

# Task 3 – RISK

1. FIVE (5) methods of identifying risks when developing software:

* **Checklist analysis**: This method involves using a list of common or potential risks that have been identified from previous projects or ind ustry standards. The list can be customized or modified according to the specific context and characteristics of the current project. The checklist can help the project team to review and assess the likelihood and impact of each risk, and to prioritize them accordingly **[10]**.
* **Brainstorming**: This method involves gathering the project team and other stakeholders to generate and share ideas about possible risks that could affect the project. Brainstorming can be done in a structured or unstructured way, using techniques such as mind mapping, affinity diagrams, nominal group technique, etc. Brainstorming can help the project team to identify risks from different perspectives and sources, and to encourage creativity and participation **[8]**.
* **SWOT analysis**: This method involves analyzing the strengths, weaknesses, opportunities, and threats of the project. Strengths and weaknesses are internal factors that can affect the project positively or negatively, while opportunities and threats are external factors that can create or reduce risks for the project. SWOT analysis can help the project team to identify risks that are related to the project’s objectives, resources, environment, and stakeholders **[11]**.
* **Interviewing**: This method involves conducting interviews with the project team members, customers, users, experts, or other stakeholders who have relevant knowledge or experience about the project. Interviewing can be done in a formal or informal way, using open-ended or closed-ended questions, depending on the purpose and scope of the risk identification. Interviewing can help the project team to collect qualitative and quantitative data about potential risks, and to gain insights and feedback from different viewpoints **[12]**.
* **Delphi technique**: This method involves using a panel of experts who anonymously provide their opinions and judgments about possible risks that could affect the project. The opinions are collected and summarized by a facilitator, who then sends them back to the experts for review and revision. The process is repeated until a consensus is reached or a sufficient level of agreement is achieved. The Delphi technique can help the project team to identify risks that are complex, uncertain, or controversial, and to reduce bias and influence among the experts .

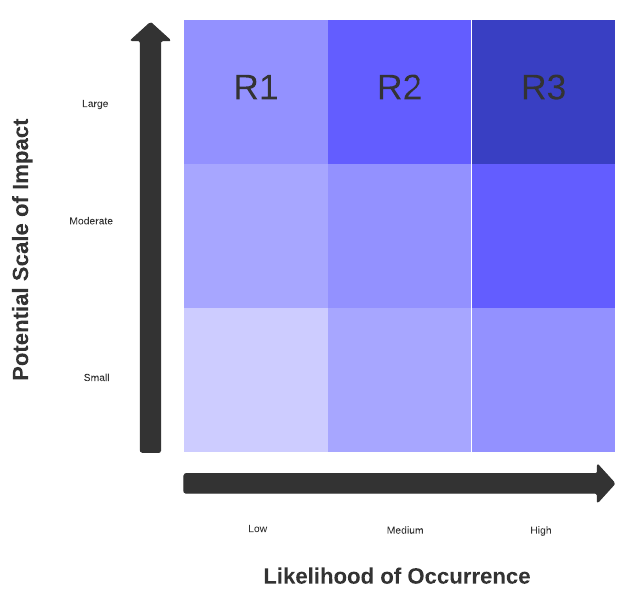
1. THREE (3) main risks in the Tbay Airline scenario:

* **R1**: The risk of developing a system that does not meet the safety or quality standards of the airline industry or the trade union. This risk could result from the lack of interactive human judgment in the system, the omission of some operational factors in the program, or the insufficient testing of the system under various scenarios and conditions. This risk could affect the reliability and usability of the system, as well as the trust and satisfaction of the customers and the maintenance workers. This risk could also lead to legal or regulatory issues or penalties for the airline company.
* **R2**: The risk of losing or alienating key stakeholders due to poor communication or collaboration. This risk could result from the disregard of the systems analyst’s concerns by the airline management, the failure of the systems analyst to volunteer his opinion in the inquiry, or the lack of consultation with the trade union by the airline management. This risk could affect the alignment and agreement of the project objectives, requirements, and expectations among the stakeholders. This risk could also lead to conflicts or disputes among the stakeholders, or loss of support or cooperation from them.
* **R3**: The risk of exceeding the budget or schedule due to inaccurate estimations or scope creep. This risk could result from the unrealistic time and cost estimates by the airline management, the difficulty of predicting the actual effort and resources required for the project, or the addition of more features or requirements without proper planning or prioritization. This risk could affect the feasibility and viability of the project, as well as the profitability and reputation of the airline company. This risk could also lead to reduced quality or functionality of the system, or increased stress or workload for the project team.

1. Diagram shows a simple approach to the Risk Management Process



1. Diagram shows an risks that i was identified in the scenario in part TWO (2)



1. International Standards for Risk Management:

* ISO 31000
* Risk Management Standard
* COSO2

One of the most popular standards is ISO 31000:2018 - Risk management – Guidance **[13]**. It is a standard issued by the International Organization for Standardization (ISO) that provides general principles and guidelines for risk management that any organization can apply to any type of risk, regardless of industry or size **[13] [14]**.

The goal of ISO 31000 is to help organizations increase their ability to achieve their objectives, improve the identification of opportunities and threats, and effectively allocate and use resources for risk management3. In addition, ISO 31000 provides a reassuring level of economic resilience, professional credibility and environmental and safety outcomes **[14]**.

The purpose of ISO 31000 is to provide a general approach to risk management, including risk management principles, frameworks and processes. The principles are guidelines for establishing and maintaining an effective risk management system. The framework is the structure for the design, implementation, monitoring, evaluation and improvement of risk management within the organization. Processes are continuous steps for contextualizing, identifying, analyzing, assessing, processing, monitoring, and reviewing risk **[15]**.

# Task 4 – Software Deployment

1. There are many different ways of releasing software into use, depending on the type, scope, and complexity of the software product. Some of the common methods are:
   * + **Pre-alpha, alpha, beta, and release candidate**: This is a method that involves releasing software in different stages of development and testing, before the final version or “gold” is released to the public. Each stage has a different level of functionality, stability, and feedback from users. This method is often used for open source software or software that requires continuous updates and improvements **[16]**.
     + **Systems development life cycle (SDLC)**: This is a method that follows a structured process of planning, building, testing, preparing, and deploying a software update. Each stage has specific tasks and deliverables that ensure the quality, speed, and efficiency of the software delivery. This method is often used for large-scale or complex software projects that require formal approval and documentation **[17] [18]**.
     + **Agile**: This is a method that emphasizes flexibility, collaboration, and customer satisfaction over rigid processes and documentation. It involves releasing software in small increments or “sprints” that deliver value to the customer and allow for feedback and changes. This method is often used for software projects that have changing requirements or need to adapt to a dynamic market **[18]**.

Each of these methods has its own advantages and disadvantages depending on the context and goals of the software project. For example:

* **The size and complexity of the software update**: A large or complex update may require more planning, testing, and documentation than a small or simple one. Therefore, a method like SDLC may be more suitable than agile for such an update.
* **The urgency and frequency of the software update**: A urgent or frequent update may require more speed, flexibility, and responsiveness than a non-urgent or infrequent one. Therefore, a method like agile may be more suitable than SDLC for such an update.
* **The expectations and feedback of the customers**: A customer-oriented or feedback-driven update may require more collaboration, communication, and satisfaction than a product-oriented or feature-driven one. Therefore, a method like agile may be more suitable than pre-alpha/alpha/beta/release candidate for such an update.

To summarize, there are different ways of releasing software into use depending on various factors such as the type, scope, complexity, urgency, frequency, and customer expectations of the software project. Some of the common methods are pre-alpha/alpha/beta/release candidate, SDLC, and agile. Each method has its own pros and cons that should be considered before choosing one for a specific software project.

1. Taking to the account into the answer above , the most feasible method for releasing software in the TBay Airlines scenario would depend on the nature and scope of the software update. For example:

* If the software update is a minor bug fix or enhancement that does not affect the core functionality or performance of the system, then a method like pre-alpha/alpha/beta/release candidate may be suitable. This would allow for quick testing and feedback from users without disrupting the normal operations of the airline.
* If the software update is a major overhaul or upgrade that affects the entire system or introduces new features or capabilities, then a method like SDLC may be suitable. This would ensure that the software update is thoroughly planned, tested, prepared, and deployed with minimal risks and errors. It would also provide clear documentation and approval for the stakeholders involved in the project.
* If the software update is a response to changing customer needs or market trends that require frequent adjustments or iterations, then a method like agile may be suitable. This would enable the software update to deliver value to the customers faster and more efficiently while adapting to their feedback and changes.

Therefore, there is no one-size-fits-all solution for releasing software in the TBay Airlines scenario. The release manager should evaluate each software update based on its type, scope, complexity, urgency, frequency, and customer expectations and choose the most appropriate method accordingly.

# Task 5 – Service Management within the IT sector

1. Outsourcing is the process of contracting outside businesses to perform specialized work operations. IT outsourcing is a common practice for many companies that need IT services or solutions but do not have the resources or expertise to do it in-house. Some of the advantages and disadvantages of outsourcing IT for the development of the new computerized system at TBay Airlines are:

**Advantages:**

* **Lower costs**: Outsourcing IT can help TBay Airlines save money. The third-party provider can offer competitive rates . Outsourcing can also help TBay Airlines avoid the costs of hiring and training **[19] [11]**.
* **Access to expertise**: Outsourcing IT can give TBay Airlines access to a larger pool of skilled and experienced IT professionals. It can offer the latest technology and best practices that may not be available for an in-house IT team **[19] [11]**.
* **Focus on core business**: Outsourcing IT can allow TBay Airlines to focus on its core business functions and goals, such as improving customer service, increasing operational efficiency, and expanding market share **[19] [11]**.

**Disadvantages:**

* **Less control**: Outsourcing IT can reduce TBay Airlines’ control over its IT operations and quality. The third-party provider may not follow the same standards and policies of TBay Airlines. Outsourcing can also make TBay Airlines dependent on the third-party provider **[19] [20]**.
* **Security risks**: Outsourcing IT can expose TBay Airlines’ sensitive data and information. The third-party provider may not have adequate security measures or protocols to protect TBay Airlines’ data. Outsourcing can also increase the risk of legal or regulatory compliance issues if the third-party provider does not follow the same laws or regulations as TBay Airlines **[19] [20]**.
* **Cultural differences**: Outsourcing IT can create cultural differences or conflicts between TBay Airlines and the third-party provider. The third-party provider may have different standards and policies than TBay Airlines **[19] [20]**.

1. If TBay Airlines had chosen to use in-house development, it could be seen as the best way to develop the new Airline Maintenance system for several reasons:

* **More control**: More control: In-house development can give TBay Airlines more control. TBay Airlines can set own standards and policies for they team and ensure that they are followed consistently and accurately. In-house development can also make TBay Airlines more independent and self-reliant **[19] [20]**.
* **More security**: TBay Airlines can implement its own security measures and protocols to protect data from external threats or attacks. TBay Airlines will more security for its data and information **[19] [20]**.
* **More alignment**: In-house development can create more alignment, communicate , trust and also foster more collaboration between TBay Airlines and its IT team **[19] [20]**.

# Task 6 – Software Quality

1. There are two ways to measure software quality: internal quality and external quality **[21]**. Internal quality refers to the quality of the software code and architecture, such as its readability, maintainability, modularity, testability, and security. External quality refers to the quality of the software product or service, such as its functionality, reliability, usability, performance, and compatibility.

* **Internal quality** can be measured by using static analysis tools that can detect code smells, bugs, vulnerabilities, duplications, and other issues in the source code. These tools can also calculate various metrics such as cyclomatic complexity, cohesion, coupling, code coverage, and technical debt. Some examples of static analysis tools are SonarQube, Code Climate, and PMD **[21]** **[22]**.
* **External quality** can be measured by using dynamic testing tools that can verify the behavior and performance of the software under different conditions and scenarios. These tools can also collect feedback from users or customers on their satisfaction and expectations. Some examples of dynamic testing tools are Selenium, JMeter, and SurveyMonkey **[21] [23]**.

In my opinion both internal and external quality measurements are beneficial for the TBay Airlines:

* **Internal quality** can ensure the system is well-designed, secure, and easy to maintain and extend.
* **External quality** can ensure the system meets the functional and non-functional requirements for theTBay Airlines.

1. **ISO 9126** is an international standard for software quality evaluation. It defines a quality model that consists of six software quality characteristics: functionality, reliability, usability, efficiency, maintainability, and portability. Each characteristic is further divided into sub-characteristics that can be measured by using specific metrics or criteria.

The ISO 9126 quality model can be applied to the scenario of developing a new Airline Maintenance system for TBay Airlines as follows:

* **Functionality**: This characteristic measures how well the system provides the functions that meet the stated and implied needs of the airline company and its customers. Some sub-characteristics are suitability (how well the system fits the purpose), accuracy (how well the system produces correct results), interoperability (how well the system interacts with other systems), security (how well the system protects data and resources), and compliance (how well the system adheres to standards and regulations).
* **Reliability**: This characteristic measures how well the system performs under specified conditions for a specified period of time. Some sub-characteristics are maturity (how well the system avoids failures), fault tolerance (how well the system recovers from failures), recoverability (how well the system restores data and resources after failures), and compliance (how well the system follows reliability standards and specifications).
* **Usability**: This characteristic measures how easy it is for users or customers to use and understand the system. Some sub-characteristics are understandability (how easy it is to comprehend the system’s functions and concepts), learnability (how easy it is to learn how to use the system), operability (how easy it is to operate and control the system), attractiveness (how appealing the system is to users or customers), and compliance (how well the system conforms to usability standards and guidelines).
* **Efficiency**: This characteristic measures how well the system uses its resources to provide the required functions. Some sub-characteristics are time behavior (how fast the system responds or processes data), resource utilization (how efficiently the system uses memory, CPU, disk space, network bandwidth, etc.), and compliance (how well the system meets efficiency standards and benchmarks).
* **Maintainability**: This characteristic measures how easy it is to modify or extend the system to cope with changing requirements or environments. Some sub-characteristics are analyzability (how easy it is to diagnose or identify defects or causes of failures), changeability (how easy it is to make changes or corrections to the system), stability (how well the system avoids unexpected effects of changes), testability (how easy it is to verify or validate the system after changes), and compliance (how well the system follows maintainability standards and practices).
* **Portability**: This characteristic measures how easy it is to transfer or adapt the system to different hardware, software, or operational environments. Some sub-characteristics are adaptability (how easily the system can be configured or customized for different environments), installability (how easily the system can be installed or uninstalled in different environments), co-existence (how well the system can share resources or co-operate with other systems in the same environment), replaceability (how easily the system can be replaced by another system with similar functionality), and compliance (how well the system follows portability standards and specifications).

# Reference websites

1. faasafety.gov
2. nbaa.org
3. zdnet.com
4. cio.com
5. slideshare.net
6. tutorialspoint.com
7. ccaps.umn.edu
8. geeksforgeeks.org
9. managedagile.com
10. google.com
11. indeed.com
12. codebots.com
13. iso.org
14. ibm.com
15. risk-engineering.org
16. en.wikipedia.org
17. asana.com
18. lucidchart.com
19. executech.com
20. analyticsinsight.net
21. sealights.io
22. linearb.io
23. academia.edu