FBPML
V1.0.0

The Foundation for Best Practices in Machine Learning

Championing ethical and responsible machine learning through open-source best practices

Technical Best Practices

Technical Best Practices
from
The Foundation for Best Practices in Machine Learning

Release:
PDF Version 1.0.0.
19 May 2021

Stichting For Best Practices in Machine Learning
Leiden, the Netherlands
KvK Nummer: 82610363

https://www.fbpml.org/

Notice: This document and its content has been licensed under the Creative Commons Attribution license by the Foundation (Stichting) for Best Practices in Machine Learning (kvk number: 82610363). Any subsequent and/or other use, copying and/or adaptation of this document or its content must abide by the appropriate licensing terms & conditions as reflected thereunder.

Foreword

The promise and value of machine learning is great, but it has been hastily operationalised over the past decade with often little regard for its wider societal impact, sometimes resulting in harmful and unfair consequences.

We, at The Foundation for Best Practices in Machine Learning, want to help data scientists, governance experts, management and other machine learning professionals champion ethical and responsible machine learning. We do this through championing our technical and organisational best practices for machine learning, through the free, open-source guidelines you are currently reading.

The aim of these Best Practices is to be easily accessible to anyone working on or interested in machine learning. This means that they are designed for a large audience who come from a variety of backgrounds and organisations.

At the same time these Best Practices also aim to be complete. Although this means that they can be long at times, please do not be intimidated - read as much or as little as you feel comfortable with and come back later for more. The Best Practices are designed to be adaptable to different organisation sizes, needs, risks, resources, and expected societal impact and so the implementation can be flexible.

Creative Commons Licence

Because we want to lower the barriers to ethical and responsible machine learning, our Best Practices have been licensed under the Creative Commons Attribution license. This means they are freely available for commercial and/or private use and/or adaption, subject to attributing (i.e. referencing) The Foundation of Best Practices of Machine Learning of course.

Who are we?

We are a team of seasoned data scientists, machine learning engineers, AI ethicists and governance experts, who are enthusiastic about lowering the barriers for pragmatic ethical and responsible machine learning.

Best regard, The Board of FBPML
May, 2021

Contents

Introduction

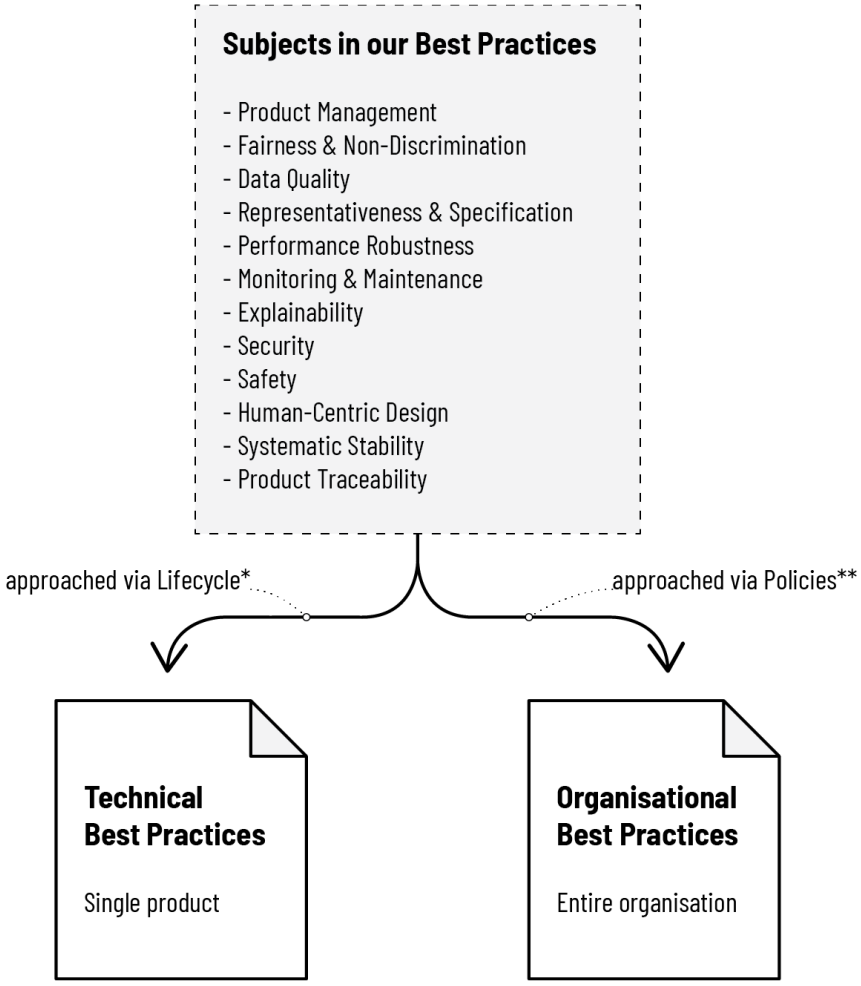
If you are not familiar yet with the Foundation for Best Practices in Machine Learning, and you want to know more about who we are, what we do, and what the philosophy and vision behind the Best Practices are, please visit our website.

On the next pages, you will find an explanation and overview of the structure of our Best Practices. Before diving straight into that, we would like to let you know about the following:

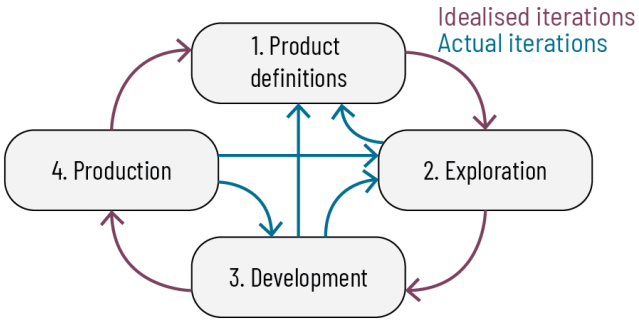
- These Best Practices are available through our Wiki-style portal too, and you'll also be able to find and contribute additional supporting material there.
- These Best Practices are open-source and rely on community contributions for continuous improvements. To find out how to contribute please have a look at our contribution guide;
- For tips on where to get started with implementing the Best Practices please have a look at our User Guides. Come back often, as we will be continuously adding new advice.

How to read the best practices

FBPML has two Best Practices documents. You are currently looking at the Technical Best Practices. The core content of both Best Practices are the subjects you see below.



*** Lifecycle**

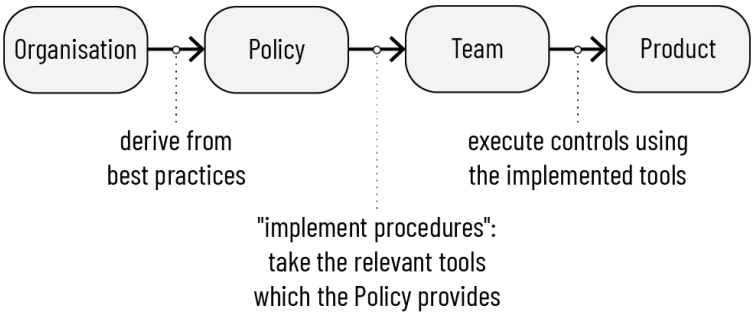


The Technical Best Practices are scoped for a single product (which includes the ML models) and are aimed at helping your team best develop and maintain this product in an ethical and responsible way. The subjects within the Best Practices are approached through Product Lifecycle phases:

Each subject's Best Practices are grouped by phases, so that the Risks and Controls are in the same order as you would typically encounter them during the first iteration of your product. Of course, during the lifecycle of your product, you will revisit each phase very often. Therefore, you will revisit the associated Best Practices too.

The Organisation Best Practices are scoped for the entire organisation. It advises how to effectively support product teams within an organisation. This support is clustered around the core subjects mentioned above. These are approached through Policies. Management and governance aspects that are overarching receive attention as well.

**** Policies**



About the wording

"Controls" and "Aims"

The Best Practices are written in a certain format wherein each "rule" consists of a number, name, Control and Aim.

- The Controls are actions to take and can be understood as the instructions. The "what to do";
- the Aim is why you should do it, sometimes phrased as a goal, but more often as a risk. The "this is what can happen if you do not do it" and/or "this is why it is important".

"Product"

The Product is our word for the technical system around which the Best Practices revolve. It is used to refer to not only the data, the machine learning model and code, but also every component and process from start to finish that is required to produce the desired effect in practice - from UI to the protocols and processes that embed models in the organization and everything in between.

Section 1. Definitions

As used in this Best Practice Guideline, the following terms shall have the following meanings where capitalised. All references to the singular shall include references to the plural, where applicable, and vice versa. Any terms not defined or capitalised in this Best Practice Guideline shall hold their plain text meaning as cited in English and data science.

1.1.	Absolute Reproducibility	means a guarantee that any and all results, outputs, outcomes, artifacts, etc can be exactly reproduced under any circumstances.
1.2.	Best Practice Guideline	means this document.
1.3.	Confidence Value	means a measure of a Model's self-reported certainty that the given Output is correct.
1.4.	Data Generating Process	means the process, through physical and digital means, by which Records of data are created (usually representing events, objects or persons).
1.5.	Data Science	means an interdisciplinary field that uses scientific methods, processes, algorithms and computational systems to extract knowledge and insights from structural and/or unstructured data.
1.6.	Domain	means the societal and/or commercial environment within which the Product will be and/or is operationalised.
1.7.	Edge Case	means an outlier in the space of both input Features and Model Outputs.
1.8.	Error Rate	means the frequency of occurrence of errors in the (Sub)population relative to the size of the (Sub)population
1.9.	Evaluation Error	means the difference between the ground truth and a Model's prediction or output.
1.10.	Fairness & Non-Discrimination	means the property of Models and Model outcomes to be free from bias against Protected Classes.
1.11.	Features	mean the different attributes of datapoints as recorded in the data.
1.12.	Hidden Variable	means an attribute of a datapoint or an attribute of a system that has a causal relation to other attributes, but is itself not measured or unmeasurable.
1.13.	Human-Centric Design & Redress	means orienting Products and/or Models to focus on humans and their environments through promoting human and/or environment centric values and allowing for redress.
1.14.	Implementation	means every aspect of the Product and Model(s) insertion of and/or application to Organisation systems, infrastructure, processes and culture and Domains and Society.
1.15.	Incident	means the occurrence of a technical event that affects the integrity of a Product and/or Model.
1.16.	Label	means the Feature that represents the (supposed) ground-truth values corresponding to the Target Variable.

1.17.	Machine Learning	means the use and development of computer systems and Models that are able to learn and adapt with minimal explicit human instructions by using algorithms and statistical modelling to analyse, draw inferences, and derive outputs from data.
1.18.	Model	means Machine Learning algorithms and data processing designed, developed, trained and implemented to achieve set outputs, inclusive of datasets used for said purposes unless otherwise stated.
1.19.	Organisation	means the concerned juristic entity designing, developing and/or implementing Machine Learning.
1.20.	Outcome	means the resultant effect of applying Models and/or Products.
1.21.	Output	means that which Models produce, typically (but not exclusively) predictions or decisions.
1.22.	Performance Robustness	means the propensity of Products and/or Models to retain their desired performance over diverse and wide operational conditions.
1.23.	Product	means the collective and broad process of design, development, implementation and operationalisation of Models, and associated processes, to execute and achieve Product Definition(s), inclusive of, amongst other things, the integration of such operations and/or Models into organisation products, software and/or systems.
1.24.	Product Manager	means either a Design Owner and/or Run Owner as identified in the Organisation Best Practice Guideline in Sections 3.1.4. & 3.1.7. respectively.
1.25.	Product Team	means the collective group of Organisation employees directly charged with designing, developing and/or implementing the Product.
1.26.	Product Subjects	means the entities and/or objects that are represented as data points in datasets and/or Models, and who may be the subject of Product and/or Model outcomes.
1.27.	Project Lifecycle	means the collective phases of Products from initiation to termination - such as design, exploration, experimentation, development, implementation, operationalisation, and decommissioning - and their mutual iterations.
1.28.	Protected Classes	mean (Sub)populations of Product Subjects, typically persons, that are protected by law, regulation, policy or based on Product Definition(s)
1.29.	Root Cause Analysis	means the activity and/or report of the investigation into the primary causal reasons for the existence of some behaviour (usually an error or deviation).
1.30.	Safety	means real Product Domain based physical harms that result through Products and/or Models applications.
1.31.	Security	means the resilience of Products and/or Models against malicious and/or negligent activities that result in Organisational loss of control over concerned Products and/or Models.

1.32.	Selection Function	means a (where possible mathematical) description of the probability or proportion of all real Subjects that might potentially be recorded in the dataset that are actually recorded in a dataset.
1.33.	Stakeholders	mean the department(s) and/or team(s) within the Organisation who do not conduct data science and/or technical Machine Learning, but have a material interest in Product Machine Learning.
1.34.	(Sub)population	means any group of persons, animals, or any other entities represented by a piece of data , that is part of a larger (potential) dataset and characterized by any (combination of) attributes. The importance of (Sub) populations is particularly high when some (Sub)populations are vulnerable or protected (Protected Classes).
1.35.	Systemic Stability	means the stability of Organisation, Domain, society and environment as a collective ecosystem.
1.36.	Target of Interest	means the fundamental concept that the Product is truly interested in when all is said and done, even if it is something that is not (objectively) measureable.
1.37.	Target Variable	means the Variable which a Model is made to predict and/or output.
1.38.	Traceability	means the ability to trace, recount, and reproduce Product outcomes, reports, intermediate products, and other artifacts, inclusive of Models, datasets and codebases.
1.39.	Variables	mean the different attributes of subjects or systems which may or may not be measured.

Part A
Product Management

What do we mean by Product Management?

We define the roles that engage in Product Management, specifically 'Product Manager' and 'Product Team', but we don't define Product Management. This is out of recognition of the fact that there are varying definitions of the term. Our working definition is as follows -

> *Product Management refers to the process of guiding, governing, and supervising every step of a product's lifecycle - from conception through operationalisation.*

Closely related to this concept is Project Management. Project Management refers to the process of guiding, governing, and supervising each step of a product's lifecycle that is necessary to meet a specific goal and/or success criteria. In other words, Project Management is time and scope-limited; while Product Management implies ownership and responsibility that is neither limited by time or scope. This is an important distinction. However, many organizations and practitioners use these terms interchangeably.

Why is Product Management relevant to this project?

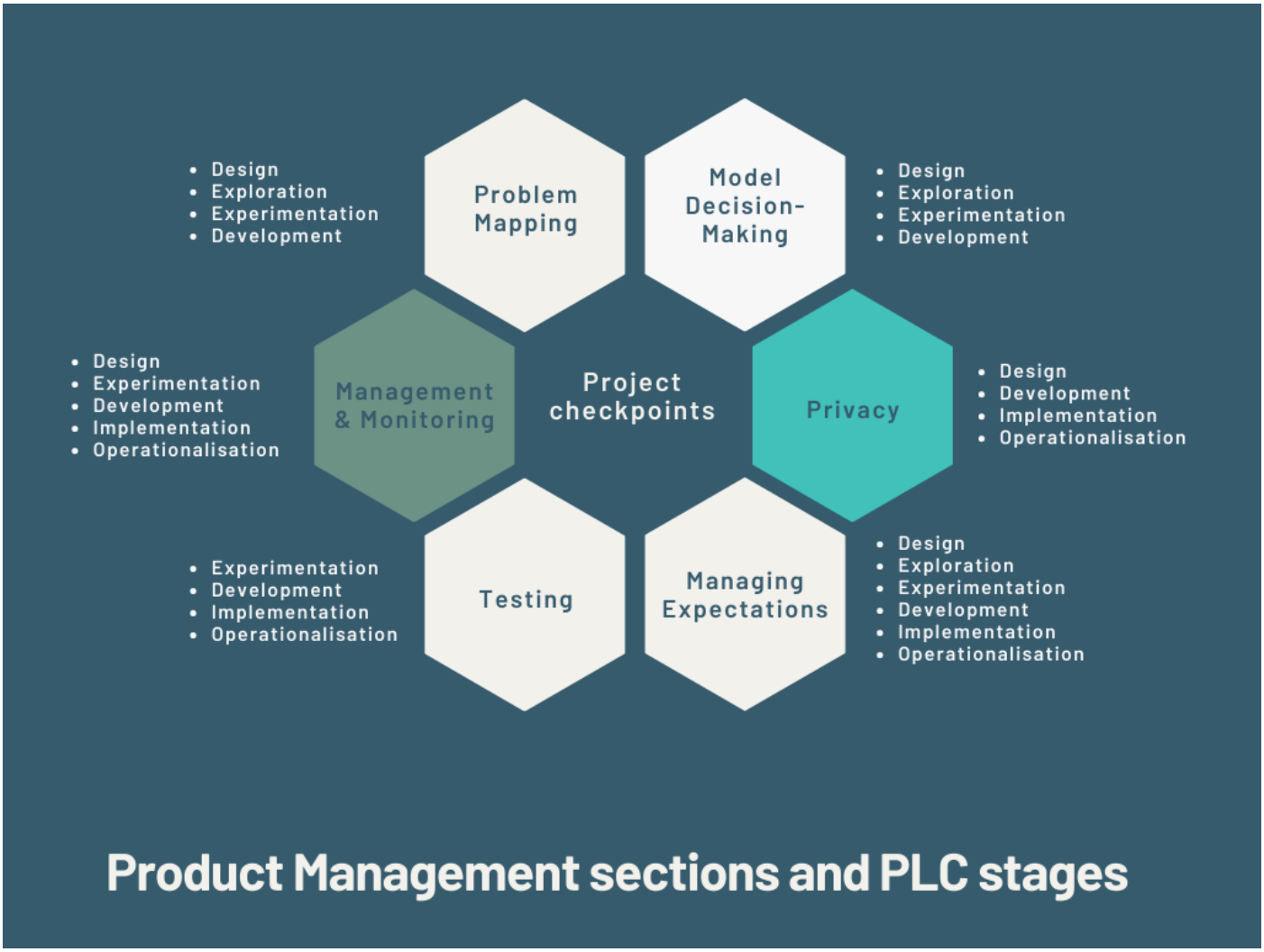
There is an oft-cited statistic that indicates that a staggering percentage of machine learning projects fail. A variety of reasons are given for these failures, but the most cited reasons are in areas that are under the purview of product management e.g. inadequate budget, unreasonable stakeholder expectations, unrealistic time frame, poor market fit, biased or unfair outcomes, inadequate business value, insufficient internal infrastructure, lack of communication and alignment between stakeholders, etc.

While the Product Manager, or the Product Management team, may not be directly responsible for each of these areas of competency, it is the role of Product Management to broker the relationships necessary to ensure all relevant issues to the product are identified and addressed and that information flows freely and as necessary to relevant stakeholders. The Best Practices attempt to highlight the various areas, analyses and decision points have been acknowledged within the industry as critical to the management of machine learning products.

How to navigate the Product Management Section?

The Best Practices are a high level overview of the range of issues that are of interest throughout each stage of the PLC. A watchlist of sorts highlighting, broadly, all of the areas that need to be addressed throughout the process. They are presented as issues to review, consider, analyze, and document. We don't provide specific frameworks or types of analyses to be performed in this document. That information will be provided, along with a granular question set for each section, in the implementation document that will follow the Technical Best Practices. It is our hope that the community will contribute to the implementation discussion in the interim, allowing us to incorporate community input in the implementation document.

Product Management begins with two sections of definitions (Team Composition and Context), before moving into areas that are applicable to various stages of the product lifecycle. The sections outline areas of concern that arise during each of the following stages of the product lifecycle: (a) Design, (b) Exploration, (c) Experimentation, (d) Development, (e) Implementation, and (f) Operationalisation. The infographic on the right illustrates the relationship between the Product Management sections and product lifecycle stages.



Section 2. Team Composition

Objective:
To (a) ensure a balanced Product Team composition that fosters close collaboration and enhances a diversity of skills; and (b) to promote Product Team coordination and understanding through thorough team organization.

		Control:	**Aim:**
2.1.	Product Team Composition	Document and define a clear diversity of Product Team roles and expertises needed for the Product, inclusive of, amongst other things, engineers, data scientists, Product Managers, and user experience experts. Once established, recruit accordingly.	To (a) assemble a robust team for Product and/or Model design, development and deployment; and (b) highlight associated risks that might occur in the Product Lifecycle.
2.2.	Product Team Roles	Document and allocate clear Product Team roles and expectations for Product Team members, including expectations for, and the structure of, intra-Product Team collaboration and overlapping responsibilities.	To (a) ensure that Product Team roles are clearly defined; and (b) highlight associated risks that might occur in the Product Lifecycle.
2.3.	Product Team Strengths and Skills Analysis	Document and assess the range of Product Team member skills and interests. Attempt to match member skills and interests to appropriate Product Team Roles as much as is practically possible.	To (a) ensure Product Team skill alignment and continued interest; and (b) highlight associated risks that might occur in the Product Lifecycle.
2.4.	Product Management	Document and allocate a clear Product Management role and duties to Product Managers, inclusive of ensuring that Product Managers have suitable Product oversight, a clear understanding of Product Team dynamics, and a contextual understanding of the Product and its operationalisation. Please see Section 3 of the Organisation Best Practices Guideline for further context.	To (a) ensure that Product Manager roles are clearly defined; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 3. Context

Objective:
To ensure the Product Team's continual access to a deep understanding of the various external contexts that affect the successful design and deployment of the Product.

		Control:	**Aim:**
3.1.	Industry Context	Incorporate regulations, standards, and norms that reflect industry values, boundaries, and constraints during each phase of Product design and deployment. Document and define clear qualitative metrics and counter-metrics in Product & Outcome Definitions, Data & Model Metrics and Acceptance Criteria Metrics, as relevant, as discussed in Section 4 - Problem Mapping; Section 5 - Model Decision-Making.	To (a) assemble a robust team for Product and/or Model design, development and deployment; and (b) highlight associated risks that might occur in the Product Lifecycle.
3.2.	Deployment Context	Incorporate an understanding of the technical and infrastructure aspects of the deployed Product into the Product design process. Ensure that infrastructure, integration, and scaling requirements and limitations are considered during the Problem Mapping and Planning phases and document and define clear requirements for the Organisation Capacity Analysis, Product Scaling Analysis, Product Integration Strategy, Product Risk Analysis, Testing - Automation Analysis, and POC-to-Production Analysis, as discussed in Section 4 - Problem Mapping; Section 6 - Management & Monitoring; Section 8 - Testing.	
3.3.	Societal Context	Research and consider the on and off platform effects of Product deployment on end users, their communities, and societies during each phase of Product design and deployment. Ensure that behavioral shifts, power balance, and cultural concerns are considered during the Problem Mapping and Planning phases, and that these provide input for the Problem Statement & Solution Mapping, Outcome Definition, Product & Outcome Definitions Data & Model Metrics, Product Risk Analysis, User Experience Mapping, Model Type - Best Fit Analysis, Acceptance Criteria, Privacy, Testing Participants, and Accuracy Perception, as discussed in Section 4 - Problem Mapping; Section 7 - Privacy; Section 8 - Testing; Section 9 - Managing Expectations.	

Section 4. Problem Mapping

Objective:
To determine and define an appropriate, feasible and solvable business problem through consideration of several interacting analyses.

		Control:	**Aim:**
4.1.	Problem Statement & Solution Mapping	Document and define clear problem statements in terms of (i) User needs, (ii) Organisation problem, and/or (iii) Organization opportunity. Subsequently, document and define clear solutions to the problem statements, inclusive of the contextual needs and/or variants of the problem statements and/or their solutions.	To ensure Products have clear scopes to warrant (a) their effective oversight, management and execution, as well as (b) allow for the accurate evaluation of Product risks and controls.
4.2.	Data Capacity Analysis	Map and document the state of the data delivery pipeline and available databases required to support the problem statements and solutions.	To (a) ensure that the data pipeline is sufficient to support Product(s) and enable the desired Outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.
4.3.	Product Definitions	Document and define clear Product definitions, aims, requirements and internal deliverables having regard for the above Problem Statement & Solution Mapping analysis, inclusive of subsequent iterations thereof.	To ensure Product(s) have clear scope to warrant (a) their effective oversight, management and execution, as well as (b) allow for the accurate evaluation of Product risks and controls.
4.4.	Outcomes Definitions	Document, delineate, and define clear Product Outcomes and Outcomes deliveries based on the above Product Definitions and the Problem Statement & Solution Mapping analysis, inclusive of subsequent iterations thereof.	To ensure Product(s) have clear scopes to warrant (a) their effective oversight, management and execution, as well as (b) allow for the accurate evaluation of Product risks and controls.
4.5.	Product & Outcome Definitions Data & Model Metrics	Document and define the above Product and Outcome Definitions in terms of clear Model and data metrics.	To ensure Product(s) have clear scopes to warrant (a) their effective oversight, management and execution, as well as (b) to allow for the accurate evaluation of Product risks and controls.

4.6.	Organisation Capacity Analysis	Document and assess whether the organisation has the requisite capacity to achieve the above Product Outcome and Product Metric Definitions given the Product Team Composition and Product Team Strengths and Skills and Data Capacity Analyses. If constraints detected, reiterate formulations of Product and/or Outcome Definitions to accommodate organisation capacity.	To (a) ensure that the Organization has sufficient capacity to support Product(s) and enable desired Outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.
4.7.	Product Scaling Analysis	Document and assess the estimated degree to which the Product can be feasibly scaled within Product Domains and the Organisation, having consideration for the Organisation Capacity Analysis.	To (a) ensure that the Organization has sufficient capacity to support the Product(s) and enable the desired Outcomes as the Product scales; and (b) highlight associated risks that might occur in the Product Lifecycle.
4.8.	Product Integration Strategy	Document and assess the processes needed to integrate and scale the Product into organisational structures based on the Organisation Capacity and Product Scaling Analyses. If constraints detected and/or integration appears unfeasible, reiterate formulations of Product and/or Outcome Definitions and/or review the Organisation Capacity and/or Product Scaling Analyses to accommodate a practical Product Integration Strategy.	To (a) ensure the Product and Outcome Definitions can be achieved within the bounds of the Organisation Capacity and Product Scaling Analyses; and (b) highlight associated risks that might occur in the Product Lifecycle.
4.9.	Product Risk Analysis	Document and assess the estimated risks associated with Product design, development, implementation, and operation, inclusive of considerations from the Product Scaling Analysis, and the Product Integration Strategy.	To (a) ensure Products have clear risk portfolios to warrant (i) their effective oversight, management and execution, as well as (ii) to allow for the accurate evaluation of Product risks and controls; and (b) highlight associated risks that might occur in the Product Lifecycle.

4.10.	Product Cost Analysis	Collaborate with Finance and purchasing to document and assess the estimated costs associated with Product design, development, implementation, and operation, inclusive of considerations from the Product Scaling Analysis, the Product Integration Strategy, and the Product Risk Analysis.	To (a) ensure a realistic project budget is provided; and (b) highlight associated risks that might occur in the Product Lifecycle.
4.11.	User Experience Mapping	Document and assess the user experience and the desired experience for various user groups, when interacting with the Product (e.g. using Norman's Usability Heuristics). Consider mitigation strategies for possible negative impacts on and off platform. If gaps in user experience are detected or a need for process redesign or behavioral changes are uncovered reiterate formulations of Outcome Definition, as discussed in Section 4 - Problem Mapping, Organisation Capacity Analysis, and Product Integration Strategy to accommodate an effective user experience.	To (a) ensure an effective user experience; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 5. Model Decision-Making

Objective:
To determine the most desirable and feasible model to achieve the desired Product Outcomes through consideration of several interacting analyses.

		Control:	**Aim:**
5.1.	Model Type - Metric Fit Analysis	Document and assess the Model requirements needed to meet the Product Definitions, Outcome Definitions, and Product & Outcome Definitions Data & Model Metrics, as discussed in Section 4 - Problem Mapping.	To ensure that chosen Model(s) meet the requirements of the Product Definitions, Outcome Definitions, and Product & Outcome Definitions Data & Model Metrics.
5.2.	Model Type - Risk Analysis	Document and assess Model requirements needed to meet the Explainability Requirements and Product Risk Analysis, as discussed in Section 16 - Explainability; Section 4 - Problem Mapping.	To ensure that chosen Model(s) meet the requirements of the Explainability Requirements and Product Risk Analysis.
5.3.	Model Type - Organisation Analysis	Document and assess the compatibility of potential Models with the Organisation Capacity Analysis, Product Scaling Analysis, and Product Integration Strategy, as discussed in Section 4 - Problem Mapping, given technical considerations.	To ensure that chosen Model(s) meet the requirements of the Organisation Capacity Analysis, Product Scaling Analysis, and Product Integration Strategy.
5.4.	Model Type - Best Fit Analysis	Document and assess the most appropriate Models that best meet the requirements of, and which produces the most favorable outcome given the trade-offs between, the Model Type - Metric Fit, Risk and Organization Analyses.	To (a) ensure that the most appropriate Model(s) are chosen; and (b) highlight associated risks that might occur in the Product Lifecycle.
5.5.	Acceptance Criteria - Metrics	Document and define the desired performance for an acceptable Model in terms of clear Model and data metrics that are written from the end user's perspective.	To (a) determine the metrics and desired performance for an acceptable Model; and (b) highlight associated risks that might occur in the Product Lifecycle.

5.6.	Acceptance Criteria - Accuracy, Bias, and Fairness	Document and define clear, narrow accuracy goals and metrics that manage the tradeoff of accuracy and explainability. Document and define the Model requirements needed to meet the Fairness & Non-Discrimination goals, as discussed more thoroughly and technically in Section 11 - Fairness & Non-Discrimination.	To (a) ensure appropriate accuracy, bias and fairness metrics for Model(s); and (b) highlight associated risks that might occur in the Product Lifecycle.
5.7.	Acceptance Criteria - Error Rate Analysis	Consider the Societal and Industry Contexts in determining the acceptable method for error measurement, as discussed in Section 4 - Problem Mapping. Document and define the acceptable error types and rates for the Product as required by Representativeness & Specification, as discussed more thoroughly and technically in Section 13 - Representativeness & Specification. Analyze any potential tension between achievable and acceptable error rates and determine whether that tension can be resolved.	To (a) ensure appropriate error type and rate metrics for Model(s); and (b) highlight associated risks that might occur in the Product Lifecycle.
5.8.	Acceptance Criteria - Key Business Metrics / Targeted Metrics	Document and define the key business metrics (KPIs) as determined in Problem Statement & Solution Mapping, as discussed in Section 4 - Problem Mapping, and translate them into metrics that can be tracked within the framework of chosen Model(s), or into proxy metrics if direct tracking is not feasible.	To (a) ensure appropriate business metrics for Model(s); and (b) highlight associated risks that might occur in the Product Lifecycle.
5.9.	Technical Considerations	Document and assess technical issues that should be considered during the Model selection process.	To (a) ensure that technical issues are considered when selecting Models; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 6. Management & Monitoring

Objective:
To ensure an effective and auditable Product Lifecycle.

		Control:	**Aim:**
6.1.	Product Requirements	Draft and document clear Product requirements. Review Model Type - Metrics and Acceptance Criteria, as discussed in Section 4 - Problem Mapping, to ensure alignment. Regularly review Product & Outcome Definitions Data & Model Metrics and User Experience Mapping, as discussed in Section 4 - Problem Mapping, and update as necessary	To (a) ensure current, clear, and actionable Product requirements; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.2.	Product Roadmap and Pipeline	Develop and document a Product roadmap and pipeline that enable the experience envisioned in the User Experience Mapping, as discussed in Section 4 - Problem Mapping, and include the following sections: Schedule and milestones, tasks and deliverables, limitations and exclusions (scope), initial prioritization, and methods for determining future priority.	To (a) ensure a clear, actionable, and prioritized Product roadmap and pipeline; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.3.	Experimentation Constraints	Develop and document a method for evaluating the quality of predictions. Develop and document criteria for determining when to stop the experimentation process.	To (a) develop processes to ensure a balance between effectiveness and efficiency in the experimentation cycle; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.4.	Behavioral Change Analysis - Process Changes	Research and assess the business processes that will be affected by the new Product and/ or the infrastructure changes that enable the Product. Review the User Experience Mapping, Data Capacity Analysis, and Organisation Capacity Analysis, as discussed in Section 4 - Problem Mapping, and reformulate as necessary. Develop and document a plan to retrain affected parties as necessary and mitigate business disruptions as much as feasible.	To (a) determine business processes that may be affected by the project and create a plan to retrain or mitigate impacts as necessary; and (b) highlight associated risks that might occur in the Product Lifecycle.

6.5.	Behavioral Change Analysis - Social (Off-Platform)	Research and document ways in which the Product can be abused or negatively impact customers, end users, or the broader society. Develop and document a plan to mitigate negative impacts as much as feasible. Develop and document counter metrics to assess whether users or the model are 'gaming' the system.	To (a) determine (i) negative product uses, (ii) negative product impacts (iii) negative user or model behaviors and create a plan to counter behaviors or mitigate impacts as necessary; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.6.	Resource Assessment	Document the processes, tools, and staffing that are required for every phase of the project, including the Data Capacity Analysis, Organisation Capacity Analysis, Product Scaling Analysis, and Product Cost Analysis, as discussed in Section 4 - Problem Mapping, before starting each phase of the project and update as necessary.	To (a) ensure adequate resources and funding during every phase of the Product Lifecycle; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.7.	POC-to-Production Checklist	Document and define a POC-to-Production Checklist that details the existing system modifications, and new system builds, required for integrating the Product into Organisation infrastructure and incorporating additional data sources. If gaps in organisational capacity are detected, reiterate formulations of Organisation Capacity Analysis, as discussed in Section 4 - Problem Mapping, as necessary.	To (a) ensure sufficient planning for Product development and production; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.8.	Update Schedule	Document and define a POC-to-Production Checklist that details the existing system modifications, and new system builds, required for integrating the Product into Organisation infrastructure and incorporating additional data sources. If gaps in organisational capacity are detected, reiterate formulations of Organisation Capacity Analysis, as discussed in Section 4 - Problem Mapping, as necessary.	To (a) ensure the Product and its related software are updated and upgraded regularly and that the schedule for said updates are coordinated with information technology department(s); and (b) highlight associated risks that might occur in the Product Lifecycle.

6.9.	Project Records	Develop and document a process for preserving data of the information considered when making significant product decisions. Include any methods for standardized experiment tracking and artifact capturing that are developed by Data Science and Engineering. Develop a continuously maintained and consistently available repository for Product Requirements and any data related to their updates.	To (a) maintain a historical record of Product and data and ensure that all iterations of Product Requirements are continuously available to Stakeholders; and (b) highlight associated risks that might occur in the Product Lifecycle.
6.10.	Project Records - Stakeholder Sign-offs	Develop a standard Stakeholder Sign-off Document to be utilized (i) after the finalization of the following documents and analyses: Problem Statement & Solution Mapping, Outcomes Definition, Product & Outcome Definitions, Product Integration Strategy, Model Type - Best Fit Analysis, Acceptance Criteria - Key Business Metrics/Targeted Metrics, Testing Design and Scheduling Framework, Resource Assessment, as discussed in Section 4 - Problem Mapping, Section 5 - Model Decision-Making; and (ii) at Project Checkpoints, as discussed in Section 10 - Project Checkpoints.	To (a) ensure stakeholder buy-in; and (b) provide an auditable record of project and stakeholder expectations at every major project decision-point.
6.11.	Custody	Develop a system for documenting the chain of custody for Product(s) and the data, microservices, and applications that it is built on and with, that indicates: i) provenance ii) control iii) transfer, iv) analysis, and v) transformation.	To (a) ensure that the building blocks of the Product can be traced back to their origins; (b) allow for undesirable changes to be reverted; and (c) highlight associated risks that might occur in the Product Lifecycle.

Section 7. Privacy

Objective:
To determine the most appropriate and feasible privacy-preserving techniques for the Product.

		Control:	**Aim:**
7.1.	Decentralization Method Analysis	Consider the appropriateness of utilizing methods for distributing data or training across decentralized devices, services, or storage. When analyzing federated learning methods, consider Data Capacity Analysis, Product Integration Strategy, Product Traceability, and Fairness & Non-Discrimination, as discussed more thoroughly in Section 4 - Problem Mapping; Section 21 - Product Traceability; and Section 11 - Fairness & Non-Discrimination. When analyzing differential privacy methods, consider Data Quality - Noise, as discussed more thoroughly in Section 12 - Data Quality.	To (a) ensure appropriate privacy-preserving techniques that are aligned with chosen Models; and (b) highlight associated risks that might occur in the Product Lifecycle.
7.2.	Cryptographic Methods Analysis	Consider the appropriateness of utilizing methods for encrypting all or various parts of the data and/or Model pipeline. When analyzing homomorphic encryption methods, consider Product Integration Strategy and Product Scaling Analysis, as discussed more thoroughly in Section 4 - Problem Mapping. Additionally, consider - (a) whether the types of operations and calculations that can be performed meet the requirements of Model Type - Best Fit Analysis, as discussed more thoroughly in Section 5 - Model Decision-Making; and/or (b) whether the encrypted Model processing speed is acceptable with consideration for real world robustness and direct user interaction, as discussed more thoroughly in Section 14 - Performance Robustness.	To (a) ensure appropriate privacy-preserving techniques that are aligned with chosen Models; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 8. Testing

Objective:
To ensure (a) that robust, effective, and efficient strategies, methodologies and schedules are developed for testing the Product; and (b) clear maintenance metrics and/or phases to warrant continued Product alignment with chosen metrics and performance goals.

		Control:	**Aim:**
8.1.	Testing Design and Scheduling Framework	Document and define a testing design and schedule that does not artificially constrain the testing process. Incorporate the Feedback Loop Analysis in the testing design. Review the Automation Analysis in determining what level of automation is appropriate for each stage of testing. Ensure the individuals chosen through the Testing Participant Identification process are involved at the earliest stages of the testing schedule as practical. Post Product deployment, document and define a framework and process for testing and selecting variations of the production Model.	To (a) ensure a robust and feasible testing design and scheduling framework that allows for effective Product optimization; and (b) highlight associated risks that might occur in the Product Lifecycle.
8.2.	Testing Participant Identification	Document and define a process for identifying test participants as required by User Experience Mapping and Societal Context, as discussed in Section 4 - Problem Mapping; and Section 3 - Context. Determine a framework for ensuring that testing participants are intentionally diverse across use cases, user types and roles, and internal and external Stakeholders.	To (a) ensure testing for user impact and participant pool diversity; and (b) highlight associated risks that might occur in the Product Lifecycle.

8.3.	Automation Analysis	Determine and define data, Model, and component integration validations that can be reasonably automated. Assess and define any processes during the development, deployment, or maintenance phases that could benefit from integrating automation into the testing infrastructure. Be sure to review - (a) the Organisation Capacity Analysis, as discussed in Section 4 - Problem Mapping, while determining the feasibility of automating the identified processes; and/or (b) the Industry, Deployment, and Societal Contexts, as discussed in Section 3 - Context, to uncover any gaps or misalignment raised by the automation of any identified process.	To (a) identify suitable and effective areas for incorporating testing automation within the Product development, deployment, and maintenance phases; and (b) highlight associated risks that might occur in the Product Lifecycle.
8.4.	Feedback Loop	Document and define a feedback loop that enables monitoring of stability, performance, and operations metrics, and counter-metrics, as required by Performance Robustness, Monitoring and Maintenance, and Systemic Stability, as discussed more thoroughly in Section 14 - Performance Robustness; Section 15 - Monitoring & Maintenance; and Section 20 - Systemic Stability. Develop and incorporate a method for flagging bias and for issue reporting. Document and define a process for real-time sharing of testing participant feedback with the development and maintenance teams. Incorporate the Feedback Loop in the Testing Design and Scheduling Framework to ensure that the features the Model is utilizing are acceptable for the application during the development, deployment, and maintenance phases.	To (a) ensure robust and responsive feedback loop measures that enable monitoring of necessary metrics and effectively integrate into the Testing Design and Scheduling Framework; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 9. Managing Expectations

Objective:
To effectively set and communicate realistic Product expectations to Stakeholders and obtain their buy-in.

		Control:	**Aim:**
9.1.	Performance	Product management should attempt to set realistic Product performance expectations for Stakeholders through periodic stakeholder discussions on the following issues: (i) limited industry understanding of what tasks are difficult for the Product; (ii) difficulty of determining what type of modifications - network design, input features, or training data - will create the greatest Product improvement; and (iii) Model improvement can stall significantly while experimenting with different variable modifications.	To (a) effectively communicate realistic Product performance expectations throughout the development process; and (b) highlight associated risks that might occur in the Product Lifecycle.
9.2.	Timeframe	Product management should set expectations for long-term investment in the Product for Stakeholders, specifically focusing on: (i) the unpredictability of Product improvement; (ii) Product difficulties are traditionally hard to diagnose as they are often caused by subtle issues of intersecting inputs; and (iii) it is possible for the Product to completely stall with absolutely no discernible improvement in spite of significant time and effort.	To (a) effectively set Stakeholder expectations regarding the difficulty of locking down Product timelines; and (b) highlight associated risks that might occur in the Product Lifecycle.
9.3.	Accuracy perception	The Product Team should work to ensure that the solution will be accurate enough to meet a variety of different Stakeholders' expectations, recognizing that each group of Stakeholders will have different views on what is 'accurate' based on their interaction with the Product. Product management should set and communicate expectations in-line with the achievable level of accuracy for each user group.	To (a) effectively communicate achievable accuracy levels, considering individual Stakeholder accuracy preferences; and (b) highlight associated risks that might occur in the Product Lifecycle.

9.4.	POC-to-Production	The Product Team should effectively communicate that infrastructure is often the determining factor for the success of the POC-to-Production transition and rely heavily on the POC-to-Production Checklist, as discussed in Section 6 - Management & Monitoring, to set and align Stakeholder expectations of the transition process. The Product Team should set the expectation, before beginning the transition process, that novel problems will likely arise during the transition that may significantly affect the timeline and costs. The Product Team should be on alert for integration issues arising close to the final release of the solution, which the Product Manager should communicate to relevant Stakeholders, along with progress updates, at a progressively more frequent cadence.	To (a) uncover and communicate issues that may delay the transition of the solution from POC-to-Production or make that transition less feasible; and (b) highlight associated risks that might occur in the Product Lifecycle.
9.5.	Production Costs	Review and analyze the finalized POC budget to determine a realistic Product implementation budget. Review the Product Cost Analysis as discussed in Section 4 - Problem Mapping to ensure its continued accuracy and reformulate as necessary. The Product Team should effectively communicate to Stakeholders that the budget for implementation will likely be in-line or more expensive than the cost to get through POC.	To (a) ensure realistic expectations for a sufficient Product implementation budget are communicated; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 10. Project Checkpoints

Objective:
To ensure that necessary factors are considered at key decision points.

		Control:	**Aim:**
10.1.	Machine Learning Appropriate Tool Analysis	The Product Team should work cross-functionally with Stakeholders to define and document a Machine Learning checklist that considers the following areas, amongst other things: a. Is there a different approach that will generate a greater return more quickly; b. Given the results of the Data Capacity Analysis, does the Organisation have enough secure, non-discriminatory, representative, high quality data for every stage of the process; c. Can the problem be solved by simple rules; d. Does the Product solution require emotional intelligence or empathy; e. Does the Product solution need to be fully interpretable or explainable; f. Given the results of the Organisation Capacity Analysis, does the Organization have the people, processes, and tools necessary to productize the end product; g. Can the consequences of Product failure be easily fixed or mitigated; and/or h. What other non-technical solutions can be used to augment the Product and its offering and/or, more directly, whether Machine Learning is the best solution for the Product at hand.	To (a) ensure that Machine Learning is the appropriate method for solving the chosen problem; and (b) highlight associated risks that might occur in the Product Lifecycle.

10.2	Data Buy v. Build Analysis	The Product Team should work cross-functionally with relevant Stakeholders to define and document a Buy v. Build checklist that considers the following areas: a. Does the Organisation have enough data for every stage of the process (training, POC, production) and for every purpose (replacing stale/flawed data, measuring success); b. Does the Organisation have the right type of data for every stage of the process (training, POC, production) and for every purpose (replacing stale/flawed data, measuring success); c. Is bought data secure and free of privacy concerns; d. Is the bias in the bought data limited, mitigatable, or removable; e. Given the results of the Data Quality Analysis, does the Organisation have quality data and are datasets complete; f. Given the Product Team Composition, does the Organisation have the staffing and expertise to clean, prepare, and maintain internal data; and/or g. Given the Data Capacity Analysis, is the necessary data easily and readily available internally.	To (a) ensure that the Organisation's decision to either purchase data or utilize in-house data is appropriate based on Organisation capacity and/or constraints; and (b) highlight associated risks that might occur in the Product Lifecycle.

10.3	Model Buy v. Build Analysis	The Product Team should work cross-functionally with relevant Stakeholders to define and document a Buy v. Build checklist that considers the following areas: a. Is the scope of the Product manageable, given the results of the Organisation Capacity Analysis; b. Can bought Models be used for other Products (eg. transfer learning); c. Does the Organisation have the in-house expertise required to acquire and label the training data, given the Product Team Composition; d. How much would it cost to acquire a properly labeled training dataset; e. Given the Product Team Composition, does the Organisation have the in-house expertise required to retrain Models, if necessary; f. How important is Model customization and, if so, can bought Models be customised; g. Are the Acceptance Criteria - Accuracy, Bias, and Fairness requirements for bought Models feasible given the timeline, Product Team Composition, and Organisation Capacity Analysis; and/or h. What are the usage limits and costs for pre-trained Models.	To (a) ensure that the Organisation's decision to either purchase or build the Models is appropriate based on Organisation capacity and/or constraints; and (b) highlight associated risks that might occur in the Product Lifecycle.
10.4	POC-to-Production Go/No-Go Analysis	The Product Team should work cross-functionally with relevant Stakeholders to define and document a Go/No-Go checklist that considers qualitative and quantitative factors in the following areas: a. Can POC-to-Production Checklist be adequately addressed; b. Is the Product Cost Analysis still feasible; c. Does the Product Team have approval for a Product maintenance budget; d. Are the updates, upgrades, and add-ons to the data infrastructure near completion; e. What is the state of customer process reconstruction and end-user training; f. Has the failsafe, rollback, or emergency shutdown plan been completed and approved; and/or g. Have the communication and mitigation plans in case of failsafe, rollback, or emergency shutdown been completed and approved.	To (a) ensure that the solution should be deployed in production and/or Product Domains; and (b) highlight associated risks that might occur in the Product Lifecycle.

Part B
MODEL DESIGN, DEVELOPMENT AND PRODUCTION

Section 11. Fairness & Non-Discrimination

Objective:

To (a) identify and mitigate risk of disproportionately unfavorable Outcomes for protected (Sub)populations; and (b) minimise the unequal distribution of Product and Model errors to prevent reinforcing and/or deriving social inequalities and/or ills, and (c) promote compliance with existing anti-discrimination laws and statutes.

What do we mean when we refer to Fairness?

Fairness is a complex socio-technical challenge for which there is no single generic definition. Broadly speaking -

> *Fairness is about identifying bias in a machine learning Model or Product and mitigating discrimination with respect to sensitive, and (usually) legally protected attributes such as ethnicity, gender, age, religion, disability, or sexual orientation.*

Algorithmic discrimination can take many forms and may occur unintentionally. Machine learning Products might unfairly allocate opportunities, resources, or information, and they might fail to provide the same quality of service to some people as they do to others.
The conversation about fairness distinguishes between group fairness and individual fairness measures. Group fairness ensures some form of statistical parity (e.g. equal calibration, equal false positive/negative rate) across protected groups. Individual fairness requires that individuals who are similar with respect to the predictive task be assigned similar outcomes regardless of the sensitive attribute.

Why is Fairness relevant?

Machine learning Products are increasingly used to inform high-stakes decisions that impact people's lives.It is therefore important that ML-driven decisions do not reflect discriminatory behavior toward certain populations. It is the responsibility of data science practitioners and business leaders to design machine learning Products that minimizes bias and promotes inclusive representation.

Some business leaders express concerns about a potential increase in the risk of reputational damage and legal allegations in case of discriminatory 'black box' Models. AI fairness can substantially reduce these concerns. Another reason for taking AI fairness seriously is the development of regulatory frameworks for AI. For example, the European Commission published a white paper on AI in 2020, which was followed in 2021 by a regulatory framework proposal for AI in the European Union.

How to apply Fairness?

Fairness should be considered throughout the product lifecycle. Given that AI systems are usually designed to evolve with experience, fairness should be closely monitored during deployment as well as during product development. The Technical Best Practices Guidelines provide detailed guidance into implementing fairness in your AI products.

11.1 Product Definitions

Objective
To (a) identify and mitigate risk of disproportionately unfavorable Outcomes for protected (Sub)populations; and (b) minimise the unequal distribution of Product and Model errors to prevent reinforcing and/or deriving social inequalities and/or ills, and (c) promote compliance with existing anti-discrimination laws and statutes.

		Control:	**Aim:**
11.1.1.	(Sub)populations Definition	Define (Sub)populations that are subject to Fairness concern, with input from Domain and/or legal experts when relevant.	To (a) ensure that vulnerable and affected populations are appropriately identified in all subsequent Fairness testing and Model build; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.1.2.	(Sub)population Data	Gather data on (Sub)population membership. If a proxy approach is used, ensure the performance of the proxy is adequate in this context.	To (a) facilitate Fairness testing pre- and post-Model deployment; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.1.3.	(Sub)population Outcome Perceptions	Document and assess whether scored (Sub)populations would view Model Outcomes as favorable or not, using input from subject matter experts and stakeholders in affected (Sub)populations. Document and assess any divergent views amongst (Sub)populations.	To (a) ensure uniformity in (Sub) population outcome perception, if applicable; (b) highlight Outcome effects for different (Sub)populations; and (c) highlight associated risks that might occur in the Product Lifecycle.
11.1.4.	Erroneous Outcome Consequence Estimation Divergence	Document and assess the results of erroneous (false positive & false negative) outcome consequences, both real and perceived, specifically in terms of divergence between relevant (Sub) populations. If material divergence present, take measures to harmonise Outcome perceptions and/or mitigate erroneous Outcome consequences in Model design, exploration, development, and production.	To (a) ensure uniformity in erroneous Outcomes for (Sub) populations; (b) highlight outcome effects for different (Sub) populations; and (c) highlight associated risks that might occur in the Product Lifecycle.
11.1.5.	Positive Outcome Spread	Document and assess the degree to which Model positive outcomes can be distributed to non-scored (Sub)population, when contextually appropriate. If present, take measures to promote Model Outcome distribution in Model design, exploration, development, and production.	To (a) ensure the non-prejudicial spread of positive Model Outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.

11.1.6.	Enduring Bias Estimation	Document and assess whether exclusions from Product usage might perpetuate pre-existing societal inequalities between (Sub)populations. If present, take measures to mitigate societal inequalities perpetuation in Model design, exploration, development, and production.	To (a) ensure the non-prejudicial spread of Model Outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.1.7.	Appropriate Fairness Metrics	Consult Domain experts to inform which Fairness metrics are contextually most appropriate for the Model when conducting Fairness testing.	To (a) ensure that fairness testing and subsequent Model changes (i) result in outcome changes which are relevant for (Sub)populations; and/or (ii) are consistent with regulatory guidance and context-specific best practices; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.1.8.	Model Implications	Document and assess the downside risks of Model misclassification/inaccuracy for modeled populations. Use the relative severity of these risks to inform the choice of Fairness metrics.	To (a) ensure that improving in the chosen Fairness metrics achieves the greatest Fairness in Model decisioning after deployed; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.1.9.	Fairness Testing Approach	Document and assess the Fairness testing methodologies that will be applied to Model and/or candidate Models, along with any applicable thresholds for statistical/ practical significance, acceptable performance loss tolerance, amongst other metrics.	To (a) prevent Fairness testing methodology and associated thresholds change during Model review; and (b) highlight associated risks that might occur in the Product Lifecycle.

11.2 Exploraition

Objective
To identify and control for Fairness and Non-Discrimination risks based on the available datasets.

		Control:	**Aim:**
11.2.1.	(Sub)population Data Access	Keep separate Model development data and (Sub)population membership data (if applicable Regulations allow the possession and processing of such in the first place), especially if the use of (Sub)population data in the Model is prohibited or would introduce fairness concerns.	To (a) guarantee that (Sub) population membership data does not inadvertently leak into a Model during development.

11.2.2.	Univariate Assessments	Document and perform univariate assessments of relationship between (Sub)populations and Model input Features, including appropriate correlation statistics.	To (a) identify input Feature trends associated with (Sub)populations; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.2.3.	Prohibited Data Sources	Develop and maintain an index of data sources or features that should not be made available or utilized because of the risks of harming (Sub)populations, specifically Protected Classes.	To (a) prohibit the actioning of data sources that will disproportionately prejudice (Sub)populations; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.2.4.	Data Representativeness	Ensure the membership rates of (Sub) populations in Model development data align with expectations and that data is representative of Domain populations.	To (a) guarantee that Model performance and Fairness testing during model development will provide a consistent picture of Model performance after deployment; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.2.5.	(Sub)population Proxies and Relationships	Document and assess the relationship between potential input Features and (membership of) (Sub)populations of interest based on, amongst other things, (i) reviews with diverse Domain experts, (ii) explicit encoding of (Sub) population membership, (iii) correlation analyses, (iv) visualization methods. If relationships exist, the concerned input Features should be excluded from Model datasets, unless a convincing case can be made that an (adapted version of) the input Feature will not adversely affect any (Sub)populations, and document this.	To (a) prevent Model decisions based directly or indirectly on protected attributes or protected class membership; (b) reduce the risk of Model bias against relevant (Sub)populations; (c) understand any differences in data distributions across (Sub) populations before development begins; and (c) highlight associated risks that might occur in the Product Lifecycle.
Associated Controls		Review the following controls with particular attention in the context of bias and fairness with respect to protected (Sub)populations: Section 12.2.2. - Missing and Bad Data Assessment. Section 13.2.4. - Selection Function; which is concerned with accurate representation of (Sub)populations. Section 13.3.1. - 13.3.4.; which are concerned with the choice and definition of the Target Feature.	

11.3. Development

Objective
To minimise the unequal distribution of Product and Model errors for (Sub)populations during Model development in the most appropriate manner.

		Control:	**Aim:**
11.3.1.	Explainability (xAI) (Sub)population Outcomes	Keep separate Model development data and (Sub)population membership data (if applicable Regulations allow the possession and processing of such in the first place), especially if the use of (Sub) population data in the Model is prohibited or would introduce fairness concerns.	To (a) guarantee that (Sub) population membership data does not inadvertently leak into a Model during development.
11.3.2.	Model Architecture and Interpretability	Choose Model architecture that maximizes interpretability and identification of causes of unfairness. Consider different methodologies within the same Model architecture (ex. monotonic XGBoost, explainable neural networks). Evaluate whether Product Aims can be accomplished with a more interpretable Model.	To (a) provide information that can guide Model-builders; (b) ensure that Model decisions are made in line with expectations; (c) allow Product Subjects and/or End Users to understand why they received corresponding Outcomes; (d) help inform the causes of Fairness issues if issues are detected; and (e) highlight associated risks that might occur in the Product Lifecycle.
11.3.3.	Fairness Testing of Outcomes	Focus fairness testing initially on outcomes that are immediately experienced by (Sub)populations. For example, if a model uses a series of sub-Models to generate a score and a threshold is applied to that score to determine an Outcome, focus on Fairness issues related to that Outcome. If issues are identified, then diagnose the issue by moving "up-the-chain" and testing the Model score and sub-Models.	To (a) ensure that the testing performed best reflects what will happen when Models are deployed in the real world; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.4.	Disparate Impact Testing	If applicable, test Model(s) for disparate impact. Evaluate whether Model(s) predict a Positive Outcome at the same rate across (Sub)populations.	To (a) ensure that (Sub)population members are receiving the Positive Outcome as often as their peers; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.5.	Equalized Opportunity Testing	If applicable, test Model(s) for equalized opportunity. Evaluate whether Model(s) predict a Positive Outcome for (Sub) population members that are actually in the positive class at the same rates as across (Sub)populations.	To (a) ensure that (Sub)population members who should receive the Positive Outcome are receiving the Positive Outcome as often as their peers; and (b) highlight associated risks that might occur in the Product Lifecycle.

11.3.6.	Equalized Odds Testing	If applicable, test Model(s) for equalized odds. Evaluate whether Model(s) predict a Positive & Negative Outcome for (Sub) population members that are actually in the positive & negative class respectively at the same rates across (Sub)populations.	To (a) ensure that (i) protected (Sub)populations who should receive the Positive Outcome are receiving the Positive Outcome as often as other (Sub)populations, and (ii) protected (Sub)populations who should not receive the Positive Outcome are not receiving the Positive Outcome as often as other (Sub)populations; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.7.	Conditional Statistical Parity Testing	If applicable, test Model(s) for conditional statistical parity. Evaluate whether Model(s) predict a Positive Outcome at the same rate across (Sub)populations given some predefined set of "legitimate explanatory factors".	To (a) ensure that (Sub)populations members are receiving the Positive Outcome just as often as (Sub) populations with similar underlying characteristics; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.8.	Calibration Testing Across (Sub) populations	If applicable, test Model(s) for calibration. Evaluate whether (Sub)populations members with the same predicted Outcome have an equal probability of actually being in the positive class.	To (a) ensure that Subpopulations each have the same likelihood of deserving the Positive Outcome for a given Model prediction; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.9.	Differential Validity Testing	If applicable, test Model(s) for differential validity. Evaluate whether Model performance varies meaningfully by (Sub) population, with a special focus on any groups that are underrepresented in modelling data.	To (a) ensure that the Model's predictive abilities aren't isolated in or concentrated to (Sub)population members; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.10.	Feature Selection Fairness Review	Evaluate the impact of removing or modifying potentially problematic input Features on Fairness metrics and Model quality.	To (a) assess whether more fair alternative Models can be made that fulfill Model objectives; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.3.11.	Modeling Methodology Fairness Review	Evaluate the impact of changing Modelling methodology choices (f.e. algorithm, segmentation, hyperparameters, etc.) on Fairness metrics and Model quality.	To (a) assess whether more fair alternative Models can be made that fulfill the Model objectives; and (b) highlight associated risks that might occur in the Product Lifecycle.

11.4 Production

Objective
To maintain operationalised Fairness at the level established during Model Development.

		Control:	**Aim:**
11.4.1.	Domain Population Stability	Continually assess the stability of the Domain population being scored, both in terms of its composition relative to the Model development population, and the quality of the Model by class.	To (a) ensure the continued accuracy of Fairness tests and metrics; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.4.2.	Fairness Testing Schedule	Define a policy for timing of re-assessment of Model fairness that includes re-testing at regular intervals and/or established trigger events (e.g. any modifications to Model inputs or structure, changes to the composition of the modeled population, impactful policy changes).	To (a) detect issues with Model Fairness that may not have existed during pre-deployment of the Model; and (b) highlight associated risks that might occur in the Product Lifecycle.
11.4.3.	Input Data Transparency	Ensure that Product Subjects have the ability to observe attributes relied on in the modeling decision and correct inaccuracy. Collect data around this process and use it to identify issues in the data sourcing/aggregation pipeline.	To (a) ensure that the Model is making decisions on accurate data; (b) learn whether there are problems with Model's data assets; and (c) highlight associated risks that might occur in the Product Lifecycle.
11.4.4.	Feature Attribution	Ensure that Product Subjects can understand why the Model made the decision it did, or how the Model output contributed to the decision. Ideally, an understanding would include which features were most important in the decision and give some guidance as to how the subject could improve in the eyes of the Model. (See Section 13 - Representativeness & Specification for further information.)	To (a) ensure that Product Subjects (i) have some level of trust/ understanding in the Model that affect them and (ii) feel that they have agency over the process and that Model Outcomes are not arbitrary.
11.4.5.	Product Subject Appeal Process	Incorporate a "right of appeal" procedure into the Model's deployment, where Product Subjects can request a human review of the modeling decision. Collect data around this process and use it to inform Model design choices.	To (a) ensure that Product Subjects are, at a minimum, made aware of the results of Model decisions; and (b) allow inaccurate predictions to be corrected.
11.4.6.	Feature attribution Monitoring	As part of regularly scheduled review, or more frequently, monitor any changes in feature attribution or other explainable metric by sub-population. (See Section 15 - Monitoring & Maintenance for further information.	To (a) detect reasons for changes in Model performance, as well as any changes earlier in the data pipeline; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 12. Data Quality

Objective:

To ensure Data Quality and prevent unintentional effects, changes and/or deviations in Product and Model outputs associated with poor Product data.

What is Data Quality?

Like many other concepts in Machine learning and data science, Data quality is something without a single and widely accepted definition. Nonetheless, we think of -

> *Data Quality as data which is fit for use for its intended purpose and satisfies business, system and technical requirements.*

In technical terms, data quality can be a measure of its completeness, accuracy, consistency, reliability and whether it is up-to-date.

Data integrity is sometimes used interchangeably with data quality. However, data integrity is a broader concept than data quality and can encompass data quality, data governance and data protection.

Why is Data quality important?

It is not difficult for all stakeholders involved in a Project to agree that good data quality is of prime importance. Bad data quality means a business or an organization may not have a good grasp on whether they are successful in meeting prior set objectives or not. Bad data quality results in poor analytical solutions, wrong insights and conclusions. This translates into inadequate response to market opportunities, an inability to timely react to customers' requests, increased costs, and last, but not least, potential shortcomings in meeting compliance requirements. In short, poor data results in poor products and poor decisions. This is undesirable.

The How of Data quality

Data quality is something that needs to be addressed throughout the product lifecycle, not only in the early stages of it, and not in any stage in isolation.

12.1 Exploration

Objective
To determine if the quality of the data shall be sufficient, or can be made sufficient, to achieve the Product Definitions.

		Control:	**Aim:**
12.1.1.	Data Definitions	Document and ensure all subtleties of definitions of all data dimensions are clear, inclusive of but not limited to gathering methods, allowed values, collection frequency, etc. If not, acquire such knowledge, or discard the dimension.	To (a) assess and prevent unjustified assumptions about the meaning of a data dimension or its values; and (b) highlight associated risks that might occur in the Product Lifecycle.
12.1.2.	Data Modeling	Document and ensure all relationships between (the fields of) different datasets are clear, in the light of their Data Definitions. (See Section 12.1.1 - Data Definitions for further information.) If this "Data Model" is not clear or available, create it, or discard the datasets.	To (a) prevent the creation and/or combination of invalid datasets; and (b) highlight associated risks that might occur in the Product Lifecycle.
12.1.3.	Missing and Bad Data Assessment	Document and assess (a) the occurrence rates and (b) co-variances of missing values and nonsensical values throughout the Model data. If either is significant, investigate causes and consider discarding affected data dimension(s) or commit dedicated research and development to mitigating measures for affected data dimension(s). (See Section 12.3.1. - Live Data Quality for further information.)	To assess (a) the risk of low quality data introducing bias to Model data and/or Outcomes; and (b) whether Model dataset(s) quality is sufficient for Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle.
12.1.4.	Data Veracity Uncertainty & Precision	Document and assess the veracity and precision of data. If compromised, uncertain and/or unknown, document and assess (i) the causes and sources hereof and (ii) statistical accuracy .Incorporate appropriate statistical handling procedures, such as calibration, and appropriate control mechanisms in Model, or discard the data dimension.	To assess (a) the risk of low quality data introducing bias to Model data and/or outcomes; (b) a priori the plausibly achievable performance; (c) whether the Model dataset(s) quality is sufficient for Product Definitions; and (d) highlight associated risks that might occur in the Product Lifecycle.

12.2 Development

Objective
To determine if Model performance is affected or biased due to data quality issues.

		Control:	Aim:
12.2.1.	Missing and Bad Data Handling	Document and assess how missing and nonsensical data (a) are handled in the Model, through datapoint exclusion or data imputation; (b) affect the Selection Function through datapoint removal; (c) affect Model performance and Fairness for subpopulations through data imputation. If (Sub)populations are unequally affected, take additional measures to increase data quality and/or improve Model resilience. Consult Domain experts during assessment and mitigation.	To (a) prevent introducing bias to Model Outcomes due to low quality data; and (b) highlight associated risks that might occur in the Product Lifecycle.
12.2.2.	Error - Quality Correlation	Document and assess whether low-quality datapoints (those with low-confidence, uncertain, nonsensical, missing and/or imputed attributes) correlate with high (rates of) error, and how this affects (Sub)populations. If so, take additional measures to increase data quality and/or improve Model performance for specific (Sub)populations.	To (a) prevent introducing bias to Model Outcomes due to low quality data; (b) whether the Model dataset(s) quality is sufficient for Product Definition(s); and (c) highlight associated risks that might occur in the Product Lifecycle.

12.3 Production

Objective
To ensure the quality of incoming data to the Product during operations.

		Control:	Aim:
12.3.1.	Live Data Quality	Document and assess whether live incoming data with low quality (low-confidence, uncertain, nonsensical, missing and/or imputed attributes) can be handled appropriately by the Model on the per-Data Subject level. If not, implement additional measures, and/or re-assess validity of Product Definition(s) in view of non-applicability to low quality live subsets.	To (a) assess and control that all Product Subjects can be supported appropriately by the live Product; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 13. Representativeness & Specification

Objective:

To (a) ensure that Product data and Model(s) are representative of, and accurately specified for, Product Domain as far as is reasonably practical; and (b) guard against unintentional Product and Model behaviour and Outcomes as far as is reasonably practical.

What is Representativeness and Specification?

Representativeness is a concept that is often used in statistics and machine learning with regards to the data we use to train a Model. A representative data sample is a set from a larger statistical population that adequately replicates the larger group according to a characteristic or quality under study. Put less metaphorically -

> *Representativeness means the ability of the Model and its data to adequately replicate and represent that characteristics of its operational environment.*

It should not be confused with representation learning (also known as feature learning) in machine learning. The latter refers to a set of techniques for automatically detecting feature patterns and in fact replaces manual feature engineering.

Specification is a less known term. In our context -

> *Specification refers to ensuring the appropriate degrees of freedom in the Model.*

For example, we have selected the appropriate cost function for the problem at hand, the target variable is appropriate and not a proxy for what we are really interested in measuring, etc. It is like representativeness but for the Model, and not the data. Unlike the performance robustness section, many of the controls here will be difficult to precisely measure quantitatively. However, we should still try to consider as many scenarios as possible and minimize all risks stemming from not addressing them rigorously.

Why is Representativeness and Specification important?

If the data is not representative with relation to the goal of the Product, it will not serve us well. It will result in poor performing Models when deployed, and it will inherently contain bias (not in the fairness and non-discrimination sense but in relation to sampling). This can lead to misleading conclusions and unrealistic assumptions and expectations. Correct specifications on the other hand relates to selecting appropriate and rigorous features, selection function, and target, etc. This ensures that the Model we develop is rigorous, robust and has a properly specified number of parameters.

The How of Representativeness and Specification

Representativeness and Specification is something that needs to be addressed throughout the product lifecycle, not only in the early stages of it, and not in any stage in isolation.

13.1 Product Definitions

Objective
To (a) ensure the pragmatic formulation and accurate specification of Product Definition(s); (b) minimise Model simplifications, assumptions and ambiguities; and (c) ensure adequate vigil of the non-reducible ones throughout the Product Lifecycle.

		Control:	**Aim:**
13.1.1.	R&S Product Definition(s) Assessment	Document and assess whether recorded Product Definition(s) are complete, unambiguous and representative of intended Product Outcomes. If they are not, refine them as much as is reasonably practical.	To (a) enable reliable execution of all further research, development and assessments; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.1.2.	Product Assumptions	Document and assess Product assumptions, the likelihood of their appropriateness, their continued validity, and inherent risks.	To (a) detect, mitigate and review Product assumptions and their inherent risks; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.1.3.	Product Simplifications	Document and assess Product simplifications, the likelihood of their appropriateness, and their inherent risks.	To (a) detect, mitigate and review Product simplifications and their inherent risks; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.1.4.	Product Limits	Document and assess the limitations of the Product's application and the applicability of Product Definitions.	To (a) detect and review Model limitations in light of (i) Model assumptions and (ii) Model simplifications; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.1.5.	R&S Problem Definition Review	R&S Product Definition(s) ought to be reviewed continually, specifically when significant Model changes occur.	To ensure that R&S Product Definition(s) are kept up-to-date to ensure their continued effectiveness, suitability, and accuracy; and (b) highlight associated risks that might occur in the Product Lifecycle.

13.2 Exploration

Objective
To (a) ensure that Model dataset(s) correspond to the Product Definition in sufficient detail, completeness and without material unambiguity; and (b) to identify associated risks in order to ensure an adequate vigil throughout the Product Lifecycle.

		Control:	**Aim:**
13.2.1.	Data Subjectivity	Document and assess whether the Model dataset(s) contain subjective components. If subjective components are present, take measures to handle or avoid subjectivity risks in Product and/ or Model design as much as is reasonably practical.	To (a) assess and control for the accuracy of the specification of Model inputs, manipulations, Outcomes, and interpretations to ensure the unambiguous applicability of Model(s) in Product Domain(s); and (b) highlight associated risks that might occur in the Product Lifecycle.
13.2.2.	Heterogeneous Variable Simplification	Document and assess whether Model datasets contain, or Model components produce, simplified input Features that represent inherently heterogeneous concepts in Product Domains. If simplified, take measures to reflect the heterogeneity of Product Domains as much as is reasonably practical.	To (a) detect, review and control for the simplification of heterogeneous input Variables; (b) prevent generalization and spurious correlation; and (c) highlight associated risks that might occur in the Product Lifecycle.
13.2.3.	Hidden Variables	Document and assess whether Model datasets are missing, or Model components hide relevant attributes of Product Subjects or systemic Variables with respect to Product Domains. If hidden, obtain additional data and/ or account for the hidden Variables in modelling as much as is reasonably practical.	To (a) assess and control for hidden correlations and causal relations in Model datasets and Variables and/ or risks of relations being spurious, ambiguous and/or confounding; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.2.4.	Selection Function	Document and assess the propensity of subpopulations and subpopulation members to be (accurately) recorded in Model datasets, with particular care for (i) unrecorded individuals, (ii) Protected Classes, and (iii) survivorship effects. Incorporate the Selection Function in Model development and evaluation in particular during Fairness & Non-Discrimination, Performance Robustness controls.	To (a) assess and control for the accuracy of Model and Model datasets in representing (Sub) populations; and (b) highlight associated risks that might occur in the Product Lifecycle.

13.2.5.	Feature Constraints	Evaluate whether any constraints should be applied to input Features, such as monotonicity or constraints on input Feature interactions in consultation with Domain experts. If determined, utilise identified constraints.	To (a) ensure that (i) Model Outcomes are maximally interpretable and (ii) Model behavior for individual Model Subjects is consistent with Domain experience; and (b) highlight associated risks that might occur in the Product Lifecycle.

13.3 Development

Objective
To (a) ensure that Model design is sufficiently specified to represent Product Domain(s) and the Product Definition(s) as much as is reasonably practical; and (b) minimise the risks of (i) adverse effects from the Model's optimisation leading to unintended loopholes and local optima, and (ii) mis-balancing competing optimisation requirements in Model design and development.

		Control:	**Aim:**
13.3.1.	Target Subjectivity	Document and assess whether the Target Feature(s) objectively represent Product Domain(s). If subjective, consider refining Product Definition(s), choosing a different Target Feature, or taking measures to promote the objectivity of Product Outcomes.	To (a) ensure that Product Outcomes are representative of subpopulations and applications, and are not misinterpreted; (b) ensure that Models are optimized only and precisely according to Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle.
13.3.2.	Target Proxies	Document and assess whether the Target Feature(s) are proxies for the true Target(s) of Interest in Product Domain(s). If Target Features are proxies, take measures to ensure and review non-divergence of Product Outcomes with regard to Product Definitions.	To (a) ensure that Product Outcomes are representative of subpopulations and applications, and are not misinterpreted; (b) ensure that Models are optimized only and precisely according to Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle.
13.3.3.	Target Proxy vs. True Target of Interest Contrasting	If the Target Feature is a proxy (i) document and assess whether the true Target(s) of Interest correlate with protected attributes and classes, including through hidden systemic Variables as much as is reasonably practical; and (ii) document and assess whether the true Target(s) of Interest and the proxy Target Feature(s) correlate differently with the Model datasets. If true, take measures to mitigate this as much as is reasonably practical.	To (a) ensure that the Model design is oriented to the true Target(s) of Interest; and (b) highlight associated risks that might occur in the lack thereof in the Product Lifecycle.

13.3.4.	Heterogeneous Target Variable Simplification	Document and assess whether the Target Feature is a simplification of, or contains a subset of, true Target(s) of Interest. If true, consider refining Product Definitions, recovering the heterogeneity, or failing that, take measures to mitigate and review this as much as is reasonably practical.	Idem Section 11.3.1-2; and to (a) detect and control for risks of generalization and spurious correlation creation.
13.3.5.	Cost Function Specification & Optimisation	Document and assess the risk propensity for - (i) optimizing for subset of objectives to the detriment of other Product objectives, (ii) optimizing for Outcomes that are unintended and/or not aligned with any Product objectives, (iii) feedback loops (when containing nested optimization loops), and (iv) Model confinement in adverse or less-than-optimal parameter or solution space - through Model cost function and optimisation procedures during the Product Lifecycle. If risks occur, take measures to mitigate them as much as is reasonably practical.	To (a) ensure the adequate optimisation of Product Definitions through an assessment of the cost function and optimization procedure; (b) to respect the boundary conditions and requirements set by the Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle.
13.3.6.	Importance Weighting	Document and assess whether Model data points are weighted by design or as collateral effect.	To (a) ensure the adequate optimisation of Product Definitions through an assessment of the cost function and optimization procedure; (b) to respect the boundary conditions and requirements set by the Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle
13.3.7.	Asymmetric Error Weights	Document and assess whether Model errors, and error rates, are weighted asymmetrically in the Model.	To (a) ensure the adequate optimisation of Product Definitions through an assessment of the cost function and optimization procedure; (b) to respect the boundary conditions and requirements set by the Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle
13.3.8.	Feature Weighting	Document and assess whether Model features are weighted by design or as collateral effect.	To (a) ensure the adequate optimisation of Product Definitions through an assessment of the cost function and optimization procedure; (b) to respect the boundary conditions and requirements set by the Product Definitions; and (c) highlight associated risks that might occur in the Product Lifecycle

13.3.9.	Output Interpretation(s)	Document and assess whether the interpretation of the Model Outcomes are clearly, completely and unambiguously defined. If not, take measures to promote Outcome interpretation(s) clarity and completeness as much as is reasonably practical.	To (a) guard against the misinterpretation and/or misapplication of Model Outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.3.10.	Time-dependent Data Modeling	Document and assess whether all time-dependent aspects of data generation (including but not limited to gathering, calibration, cleaning, and annotation), data modeling and data usage are specified and incorporated in Model design and Product Definition(s).	To (a) prevent data leakage and other forms of "time traveling" information leading to inaccurate representations of the data and/or Data Subjects; and (b) highlight associated risks that might occur in the Product Lifecycle.

13.4 Production

Objective
To ensure that the Implementation of the Product and Model(s) align with and represent Product Definition(s) and Product Domain(s).

		Control:	**Aim:**
13.4.1.	Asymmetric Error Costs	Document and assess whether Product Domain(s) costs produced by different Model errors types are accounted for in Product implementation and application in software and processes. If not, take measures to ensure that they are.	To (a) ensure that Product Domain(s) and Product Subjects consequences are accurately considered when implementing Product outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.
13.4.2.	Output Interpretation(s)	Document and assess whether Product Outcomes can be clearly, completely and unambiguously interpreted by the non-technical parties and whether these interpretations remain representative of Product Definition(s) and Model inner workings. If not, take measures to ensure that they are as much as is reasonably practical.	To (a) prevent (i) misinterpretation of Product Outcomes, (ii) the application of Products in contexts and/or to Subjects for which their appropriateness and/or quality is unconfirmed, unknown, and/or unsatisfactory, (iii) the intentional and/or unintentional misuse of Product components and Outcomes; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 14. Performance Robustness

Objective:

To warrant Model Outcomes and prevent unintentional Model behaviour a priori under operational conditions as far as is reasonably practical.

What is Performance Robustness?

Model robustness is the property of an algorithm that, when tested on a training sample and on a similar testing sample, the performance is similar. In other words, a robust model is one for which the testing error is similar to the training error. Performance robustness takes into account prospective scenarios where (one of more) inputs or assumptions are (drastically) changed due to unforeseen circumstances and, in light of these, the ability of the Model to to still consistently generate accurate output. Put more holistically -

> *Performance Robustness means the ability of the Model to generate consistent, accurate results across different sampling tests and in light of changes in operational circumstances.*

Why do we need Performance Robustness?

A Model that is not robust will hopefully not end up being used and deployed. Good performance in training, but significantly worse performance when tested on real data, is one of the reasons many proof-of-concepts do not end up being utilized. A Model which is not robust will inevitably deteriorate over time. Its predictions and recommendations will deviate from the ground truth and the end users will lose trust in the Model and may stop utilizing it altogether. This is the optimistic case. More worrisome is when users of the Model continue to use a poor performing Model and are unaware of its poor accuracy or precision, but still take it into account when making (important) judgement calls. In scenarios where there is no human-in-the-loop, detecting poor performance robustness can be even more difficult and time costly. This will result in more unknown harm, which is naturally hard to detect and determine. So, it is clear that it is in everyone's interest to ensure the Model's performance robustness.

How to ensure Performance Robustness?

Though performance robustness needs to be of a certain level to even consider deploying the Model, it is something that needs to be addressed throughout the product lifecycle, not only in the early stages of it, and not in any stage in isolation.

14.1 Product Definitions

Objective
To prevent performance loss due to Product Definition changes.

		Control:	Aim:
14.1.1.	Product Definition(s) Stability	Document and assess the stability of historic and prospective Product Definition(s) and Product Aim(s). If unstable, take measures to redefine or, failing that, to correct for or mitigate as much as is reasonably practical.	To (a) ensure that Product Definition(s) and Models remain stable and up-to-date in light of Product Domain Stability; and (b) highlight associated risks that might occur in the Product Lifecycle.
14.1.2.	Product Domain Stability	Document and assess the stability of historic and prospective Product Domain(s). If unstable, revise Product Definition(s) accordingly to ensure Product consistency and stability.	To (a) ensure that Product Definition(s) and Models remain stable and up-to-date in light of Product Domain Stability; and (b) highlight associated risks that might occur in the Product Lifecycle.

14.2 Exploration

Objective
To prevent performance loss due to (a) data and/or data definition instability; (b) volatile data elements; and/or (c) prospective increases in scale.

		Control:	Aim:
14.2.1.	Data Drift Assessment	Document and assess historic and prospective changes in data distribution, inclusive of missing and nonsensical data. If data drift is apparent and/or expected in the future, implement mitigating measures as much as is reasonably practical.	To (a) assess and promote the stability of data distributions (data drift); (b) determine the need for data distributions monitoring, risk-based mitigation strategies and responses, drift resistance and adaptation simulations and optimization, and data distribution calibration; and (c) highlight associated risks that might occur in the Product Lifecycle.
14.2.2.	Data Definition Temporal Stability	Document and assess - both technically and conceptually - historic and prospective changes of each data dimension definition. If unstable, consider refining Product Definitions and/or limiting usage of unstable data dimensions.	To (a) assess and control for the need for Model design adaptation based on data definition stability; and (b) highlight associated risks that might occur in the Product Lifecycle.

14.2.3.	Outlier Occurrence Rates	Document and assess outliers, their causes, and occurrence rates as a function of their location in data space. If numerous and persistent, include mitigating measures in Model design accordingly.	To (a) identify outliers and assess the need for Model design adaptation; and (b) highlight associated risks that might occur in the Product Lifecycle.
14.2.4.	Selection Function Temporal Stability	Document and assess the historic and prospective behaviour of Selection Function(s) of Model data. (See Section 13.2.4. - Selection Function for more information.) If unstable, take measures to account for past and future changes, and/or promote the consistency and representativeness of Model datasets and data gathering as much as is reasonably practical.	To (a) assess and control for hard-to-measure changes to the relation between Model datasets and Product Domain(s); (b) identify the risk of hard-to-diagnose Model performance degradation and bias throughout Product Lifecycle (to be controlled by 14.3.6); and (c) highlight associated risks that might occur in the Product Lifecycle.
14.2.5.	Data Generating Process Temporal Stability	Document and assess the historic and prospective behaviour of data generating processes, and their influence on the Selection Function. If unstable, take measures to account for past and future changes and/or promote the stability and consistency of data generation processes as much as is reasonably practical.	To (a) assess and control for hard-to-measure changes to the relation between Model datasets and Product Domain(s); (b) identify the risk of hard-to-diagnose Model performance degradation and bias throughout Product Lifecycle (to be controlled by 14.3.6); and (c) highlight associated risks that might occur in the Product Lifecycle

14.3 Development

Objective
To characterize, determine and control for Model performance variation, risks and robustness under live conditions a priori and throughout the Product Lifecycle.

		Control:	**Aim:**
14.3.1.	Target Feature Definition Stability	Document and assess - both technically and conceptually - the historic and prospective stability of the Target Feature definition. If unstable, consider refining Product Definitions and/or choosing a different Target Feature.	To (a) assess the need for Model design and Product Definition adaptation based on Target Feature definition stability; and (b) highlight associated risks that might occur in the Product Lifecycle.

14.3.2.	Blind Performance Validation	Document and validate that Model Performance can always be reproduced on never-before-seen hold-out data-subsets and prove that these hold-out data-subsets are never used to guide Model and Product design choices by comparing Model performance on the hold-out dataset. If performance cannot be reproduced on never-before-seen hold-out data-subset, take measures to improve robustness and Model fitting as much as is reasonably practical.	To (a) ensure Model performance robustness against insufficient generalization capabilities on live data (such as overfitting); and (b) highlight associated risks that might occur in the Product Lifecycle.
14.3.3.	Error Distributions	Document and assess error and/or residual distributions along as many dimensions and/or subsets as is practically feasible. If distributions are too broad and/or too unequal between subsets, improve Model(s).	To (a) assess and control for performance influence of data points and/or groups; (b) assess and control for the distribution of errors to influence - (i) performance robustness as a function of data drift, (ii) the systematic performance of minority data-subsets, and (iii) the risks of unacceptable errors and/or catastrophic failure; and (c) highlight associated risks that might occur in the Product Lifecycle.
14.3.4.	Output Edge Cases	Document and assess the causes, occurrence probabilities, overall performance impact of Edge Cases output by Model(s), inclusive of on Model training and design. If their influence is significant, improve model design. If occurrence is high, increase Model, code and data quality control.	To (a) assess and control for the impact of Output Edge Cases on Model design, bugs and performance; and (b) highlight associated risks that might occur in the Product Lifecycle.
14.3.5.	Performance Root Cause Analysis	Document and assess Model performance Root Cause Analysis as well as its testing method. If Root Cause Analysis is ineffective, simplify Model and/or increase diagnostics like logging and tracking.	To (a) assess and control for Model performance changes and assist in Model design, development, and debugging; (b) highlight associated risks that might occur in the Product Lifecycle.
14.3.6.	Model Drift & Model Robustness Simulations	Document and perform simulations of Model training and retraining cycles, using historic and synthetic data. Document and assess the effects of temporal changes to, amongst other things, the Selection Function, Data Generating Process and Data Drift on the drift in performance and error distributions of said simulations. If Model drift is apparent, document and perform further simulations for Model drift response optimization, and/or consider refining Product Definitions.	To (a) assess and control for Model propensity for Model drift; (b) determine the robustness of Model performance as a function of data changes; (c) determine appropriate Product response to drift; and (d) highlight associated risks that might occur in the Product Lifecycle.

14.3.7.	Catastrophic Failures	Document and assess the prevalence of predictions with High Confidence Values, but large Evaluation Errors. If apparent, improve Model to avoid these, and/or implement processes to mitigate these as much as is reasonably practical.	To (a) assess the propensity of the Model for catastrophic failures; and (b) highlight associated risks that might occur in the Product Lifecycle.
14.3.8.	Performance Uncertainty and Sensitivity Analysis	Document and assess the probability distribution of the model performance using cross-validation, statistical and simulation techniques under - (a) the assumption that the distribution of training and validation data is representative of the distribution of live data; and (b) multiple realistic variations to the Model data due to both statistical and contextual causes. If Model performance variation is high, improve Model and/or take measures to mitigate performance variation impact.	To (a) assess and control for the range of expected values of Model performance under both constant and changing conditions; (b) assess and control for whether trained model performance is consistent with these ranges; (c) identify main sources of uncertainty and variation for further control; and (d) highlight associated risks that might occur in the Product Lifecycle.
14.3.9.	Outlier Handling	Document and assess the effect of various outlier handling procedures on (a) Performance Robustness and (b) Representativeness & Specification. Ensure that only procedures are implemented that positively affect both.	To (a) ensure that outlier removal is not used to heedlessly improve test-time performance only and (b) highlight associated risks that might occur in the Product Lifecycle.

14.4 Production

Objective
To ensure the future satisfaction of Product Definition(s) through the technical and functional implementation of the Product Model(s) and systems.

		Control:	**Aim:**
14.4.1.	Real World Robustness	Document and assess potential future change in the applied effects of the Product, such as through diminishing returns and/ or psychological effects. If significant change or decrease is expected, consider refining Product Definitions and/or develop procedures for mitigation.	To (a) assess and control for the variation in applied effects of the Product on Product Definition(s) and performance; and (b) highlight associated risks that might occur in the Product Lifecycle.
14.4.2.	Performance Stress Testing	Perform and document experiments designed to attempt to induce failures in the Product and/or Model, for example, but not limited to, by supplying large quantities of or unusual data to the training or inferencing phases.	To (a) identify and control for risks associated with operational scenario's outside of regimes encountered during Model development.

Section 15. Monitoring & Maintenance

Objective:

To ensure that Products and Models remain within acceptable operational bounds.

What is Monitoring and maintenance?

Machine learning -

> *Monitoring refers to the processes of tracking and analysing the performance of a model over time and once it is deployed in production*

It provides early warning signals for performance issues. Maintenance is closely related to monitoring but is a more actionable concept.

> *Maintenance relates to the activities we need to perform upon detecting or suspecting possible deterioration in the performance of the model.*

Though it's a process closely related to Models in production, note that maintenance and monitoring steps need to be designed and addressed in early stages of the Product Lifecycle too.

Why is Monitoring and maintenance important?

Monitoring and maintenance is not only important but it is a 'must have' for any Product that is deployed in a production environment. Over time, the 'live' data will differ in small or significant ways from the historical data used to train the Model. Trends and preferences will change too. The way certain data sources are measured and coded will also change over time: new data sources are added, while others become unavailable. Therefore, we need to continuously, real-time monitor the Models that are deployed. A Model that is not maintained or updated over time eventually deteriorates, makes errors and could lead to a loss of trust and varying degrees of harm (if the domain in question is a high-stakes decision domain).

The How of Monitoring and maintenance

Model monitoring and maintenance though most commonly discussed in the deployment phase, is something that needs to be addressed throughout the product lifecycle, not only in the early stages of it, and not in any stage in isolation.

15.1 Product Definitions

Objective
To (a) track Model performance in production; and (b) ensure desired Model performance.

		Control:	**Aim:**
15.1.1.	Monitoring Objectives	Based on Product Definition(s), document and assess Product and Model monitoring objectives, inclusive of which Product and/or Model elements need close monitoring attention, such as Model data and code. Document and assess the associated risks of failing to achieve Model and/or Product Monitoring Objectives.	To (a) define Product and Model monitoring objectives; and (b) highlight associated risks for failed monitoring.
15.1.2.	Monitoring Risks	Document and assess the associated risks of failing to achieve Monitoring Objectives.	To (a) define Product and Model monitoring risks.

15.2 Exploration

Objective
To (a) define robust Product and/or Model monitoring requirements, inclusive of concerns related to Features and skews of the data; and (b) ensure the continued monitoring of Products and/or Models throughout their lifecycles.

		Control:	**Aim:**
15.2.1.	Data Source Mismatch: Training & Production Data	Define and deploy methods to detect the degree to which data sources and Features, in Model training and production data, match one another. If mismatch is detected, take measures to ensure that data sources and Features are adequately matched in both Model training and production data.	To (a) reduce nonsensical predictions of the Model due to (i) missing data, (ii) lack of data incorporated, or (iii) data measurement scaling, encoding and/or meaning; (b) to reduce the discrepancy between training and production data; and (c) highlight associated risks that might occur in the Product Lifecycle.
15.2.2.	Data Definitions and Measurements: Training & Production Data	Define and deploy methods by which to detect the degree to which data sources in Model training and production have the same definitions and measurement scales .	To (a) reduce nonsensical predictions of the Model due to (i) missing data, (ii) lack of data incorporated, or (iii) data measurement scaling, encoding and/or meaning; (b) to reduce the discrepancy between training and production data; and (c) highlight associated risks that might occur in the Product Lifecycle.
15.2.3.	Data Dependencies and Upstream Changes	Derive and implement change assessments for changes in data due to - (i) one or multiple internal or external sources (partial) updates, (ii) substantial source change, and/or (iii) changes in data production and/or delivery.	To (a) reduce nonsensical predictions of the Model due to (i) missing data, (ii) lack of data incorporated, or (iii) data measurement scaling, encoding and/or meaning; (b) to reduce the discrepancy between training and production data; and (c) highlight associated risks that might occur in the Product Lifecycle.

15.2.4.	Data Drift Detection	Define and deploy monitoring metrics and thresholds for detecting sudden and/or gradual, short term and/or long term changes in data distributions, giving priority to those that can detect past observed changes. (See Section 12.2.1- Missing and Bad Data Handling for further information). Document and assess distribution families, statistical moments, similarity measures, trends and seasonalities.	To (a) prevent predictions from diverging from training data and/ or Product Definitions by assessing whether production data is representative of older data; and (b) highlight associated risks that might occur in the Product Lifecycle.
15.2.5.	Product and/or Product Domain Changes: Trends and Preferences	Define and deploy (a) monitoring methods for detecting changes in Product Domain(s) and/or Product Definition(s); and (b) timeframes and/ or contextual triggers for reassessment of Product Domain(s) and Product Definition(s) continued stability.	To (a) ensure Models capture accurate, relevant, and current trends and preferences in Product Domain(s); (b) reduce Model 'blind spots' and better capture malicious events/attempts; and (c) highlight associated risks that might occur in the Product Lifecycle.

15.3 Development

Objective
To (a) create metrics for (i) Model performance and (ii) Model performance deterioration; and (b) ensure the continued monitoring of Products and/or Models throughout their lifecycles.

		Control:	**Aim:**
15.3.1.	Model Performance Deterioration Thresholds	Document, assess, and set thresholds for Model performance deterioration in consultation with Stakeholders.	To (a) ensure clear guidelines and indices of Model failure and performance deterioration; (b) reduce the risk of unacknowledged Model failure and performance deterioration; (c) reduce the likelihood of Model decay, ensure robustness and good performance in terms of selected metrics and scenarios; and (d) highlight associated risks that might occur in the Product Lifecycle.
15.3.2.	Product Contextual Indicators: Model Performance Deterioration	Document, assess, and set Product and Product Domain specific indicators of Model performance deterioration, inclusive of technical and non-technical indicators.	To (a) ensure clear guidelines and indices of Model failure and performance deterioration; (b) reduce the risk of unacknowledged Model failure and performance deterioration; (c) reduce the likelihood of Model decay, ensure robustness and good performance in terms of selected metrics and scenarios; and (d) highlight associated risks that might occur in the Product Lifecycle.

15.3.3.	Reactive Model Maintenance Indicators	Document, assess, and set thresholds for Model failure and reactive maintenance	To (a) ensure clear guidelines and indices of Model failure and performance deterioration; (b) reduce the risk of unacknowledged Model failure and performance deterioration; (c) reduce the likelihood of Model decay, ensure robustness and good performance in terms of selected metrics and scenarios; and (d) highlight associated risks that might occur in the Product Lifecycle.
15.3.4.	Awareness of feedback loops	Define and deploy as far as is reasonably practical (a) methods to detect whether feedback loops are occuring, and/or (b) technical and non-technical warning indicators for increased risk of the same.	As per Section 17 - Security: to prevent (in)direct adverse social and environmental effects as a consequence of self-reinforcing interactions with the Model(s); and (b) highlight associated risks that might occur in the Product Lifecycle.

15.4 Production

Objective
To (a) identify operational maintenance metrics; and (b) ensure timely update, re-train and re-deployment of Model(s).

		Control:	**Aim:**
15.4.1.	Operational Performance Thresholds	Define and set metrics and tolerance intervals for operational performance of Models and Products, such as, amongst other things, latencies, memory size, CPU and GPU usage.	To (a) prevent unavailable and unreliable service; (b) enable quick detection of bugs in the code; (c) ensure smooth integration of the Model with the rest of the systems; and (d) highlight associated risks that might occur in the Product Lifecycle.
15.4.2.	Continuous Delivery of Metrics: Model Performance	Continuously report on and record metrics about Model performance, predictions, errors, Features, and associated performance metrics to relevant Stakeholders (as decided upon in Section 13.2 _- Representativeness & Specification: Exploration and Section 13.3 - Representativeness & Specification: Development).	To (a) enable rapid identification of Model decay, and/or red flags and bugs in Model and/or data pipelines; and (b) highlight associated risks that might occur in the Product Lifecycle.

15.4.3.	Model Decay & Data Updates	Operationalise procedures to mitigate Data Drift and/or Model decay (as described in Section 14.2 _- Performance Robustness: Exploration and Section 14.3 - Performance Robustness: Development).	To (a) ensure timely implementation of any changes required in data and/or Modelling pipelines; and (b) highlight associated risks that might occur in the Product Lifecycle.
15.4.4.	Model Re-training	Operationalise procedures on how Model re-training ought to be conducted as well as approached, inclusive of, amongst other things, - (1) when will (i) a new Model be deployed, and/or (ii) a Model with the same hyperparameters but trained on new data; and/or (2) when operationalizing re-trained Models ought they be run in parallel with older Models and/or do to gracefully decommission older Models.	To (a) ensure timely implementation of any changes required in data and/or Modelling pipelines; and (b) highlight associated risks that might occur in the Product Lifecycle.
15.4.5.	Create Contingency Plans	Develop and put in place contingency plans in case of technical failures and out-of-bounds behaviour based on (a) bounds and threshold set in other controls; and (b) risk assessment of failure modes.	To (a) prevent adverse effects from failures and unexpected behaviour by providing clear instructions on roll-back, mitigation and remediation; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 16. Explainability

Objective:

To ensure Model functions and outputs are explainable and justifiable as far as is practically reasonable in order to (a) foster explainability for Stakeholders, (b) promote Model trust, (c) facilitate Model debugging and understanding, and (d) promote compliance with existing laws and statutes.

What do we mean when we refer to Explainability?

There is not one agreed-upon definition of explainability but the working definition we have adopted is that

> *Explainability refers to making the behavior and decisions of a complex machine learning model understandable to humans.*

Closely related to the concept of explainability is interpretability. Interpretability refers to the degree to which a human can inherently understand the cause of a Model's decision. In other words, interpretability relates to using Models that are transparent and can be inherently understood by humans; while explainability concerns making complex, non-transparent models understandable to humans. Many researchers and practitioners use the terms interchangeably.

Transparency is another closely related concept to explainability and interpretability. It is the broadest of the three. Transparency refers to the openness of the workings and/or processes and/or features of data, Models and the overall project (the Product). Transparency can be both comprehensible or incomprehensible depending on its content. Transparency does not necessarily mean comprehension: this is important and why it differs from explainability and interpretability. Again, transparency just refers to the openness of the workings and/or processes and/or features of data, Models and the overall project - whether technical or not.

Why is Explainability relevant?

When we talk about machine learning used for high-stakes decisions, there is a strong agreement that it is extremely important for the public and for machine learning practitioners to understand the inner workings and decision-making of Models. This is because through such understandings, we can ensure that machine learning is done fairly or, rather, that it does not generate unfair or harmful consequences. To put it more simply, we can ensure human oversight and correction over machine learning operations. Explainability also is very important for promoting trust and social acceptance of machine learning. People do not often trust and accept things they do not understand. Through explainability, we can help people understand machine learning and, in turn, trust it.

How to apply Explainability?

In order to generate thorough and thoughtful explainability, it must be considered continuously throughout all stages of the product lifecycle. This means that explainability must be addressed at the (a) Product Definition(s), (b) Exploration, (c) Development and (d) Production stages of machine learning operations.

16.1 Product Definitions

Objective
To (a) ensure the transparency of Product Definitions; (b) foster multi-stakeholder buy-in through explanations; and (c) reduce ethical risks in Product Definition(s) decision-making and Model Runs.

		Control:	**Aim:**
16.1.1.	Explainability Aims	Having consideration for (a) Product Definition(s), (b) the explanations and/or transparency sought, (c) the Model adopted, and (d) datasets used, document and assess the explainability aims of the Model.	To (a) clearly document the explainability and transparency aims of the Model; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.1.2.	Explainability Stakeholder	Document and assess the internal and external Stakeholders affected by the Model.	To identify the Model explainability Stakeholders; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.1.3.	Explainability Risks Assessment	Document and assess the individual risks of failing to provide model explainability, inclusive of a legal liability and Explainability Stakeholders mistrust.	To identify the risks of failing to provide Model explainability; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.1.4.	Legal Requirements for Interpretability	Document and assess any specific legal requirements for Explainability in consultation with legal experts.	To (a) ensure that minimum standards for explainability are met and legal risk is addressed; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.1.5.	Explainability Requirements	Document and assess the explainability and transparency requirements and levels in light of (a) Explainability Aims, (b) Explainability Stakeholders, and (c) Explainability Risks, taking care that the elicitation of said requirements involves appropriate guidance, education and understanding of Stakeholders.	To (a) clearly document the explainability requirements of the Model; and (b) highlight associated risks that might occur in the Product Lifecycle.

16.2 Exploration

Objective
To identify and document Model explainability and transparency requirements, inclusive of Stakeholder needs.

		Control:	**Aim:**
16.2.1.	Stakeholder Appraisal	Document and conduct (a) ad-hoc interviews, (b) structured surveys and/or (c) workshops with Explainability Stakeholders about their Model and Product concerns and literacy.	To (a) generate Explainability Stakeholders analytics in order to map Model explainability requirements and demands; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.2.2.	Stakeholder Appraisal Analysis	Document, analyse and map the outcomes of the Stakeholder Appraisal against the Explainability Aims and Explainability Risks.	To (a) map and analyse Model explainability requirements and demands in light of the needs of Explainability Stakeholders; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.2.3.	Explainability Matrix	Document, assess, and derive a matrix evaluating and ranking the metrics and/or criteria of explanations needed for based on the (a) Stakeholder Appraisal Analyse, (b) Explainability Aims, (c) Explainability Risks, and (d) Explainability Requirements, inclusive of explanations accuracy, fidelity, consistency, stability, comprehensibility, certainty, and representativeness.	To (a) derive a clear matrix from which to assess Model explainability requirements; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.2.4.	Explainability Feature Selection	Document and analyse the degree of Feature explainability needed in light of the Explainability Matrix.	To (a) identify the requisite degree of Feature explainability needed; and (b) highlight associated risks that might occur in the Product Lifecycle, such as later stage Model retraining due to feature ambiguity.
16.2.5.	Explainability Modelling Mapping & Analysis	Document and analyse the technical needs of Model explainability in light of the Explainability Matrix, inclusive of considerations of global vs. local explainability and/or pre-modelling explainability, modelling explainability, and post-hoc modelling explainability	To (a) identify the technical needs of the Explainability Matrix; and (b) highlight associated risks that might occur in the Product Lifecycle.
16.2.6.	Explanation Frequency & Delivery Assessment	Document and assess the frequency, most suitable and practically reasonable methods of communicating Model explainability in light of the Explainability Matrix and Stakeholder Appraisal Analysis.	To (a) identify the most appropriate method of communicating Model explainability in order to promote explainability comprehension; and (b) highlight associated risks that might occur in the Product Lifecycle.

16.3 Development

Objective
To ensure that Model design represents the explainability requirements and demands of transparency aims as much as is reasonably practical.

		Control:	**Aim:**
16.3.1.	Explainability Feature Selection Assessment	Conduct a Feature analysis of the Explainability Feature Selection in order to remove correlated and dependent Features.	To (a) interrogate the assumption of zero Feature dependency in explainability modelling; (b) prevent misleading Model explainability and transparency; and (c) highlight associated risks that might occur in the Product Lifecycle.
16.3.2.	Global Explainability Model Run	Document and run as many types of global explainability Models as is reasonably practical, such as Feature importances, Feature interactions, global surrogate Models, perturbation-based techniques or gradient-based techniques. When there is doubt about the stability of the techniques being used, test their quality through alternative parameterizations or by comparing across techniques.	To (a) generate global explainability of the model; (b) help promote model debugging; (c) ensure explainability fidelity and stability through numerous explainability model runs; and (d) highlight associated risks that might occur in the Product Lifecycle.
16.3.3.	Local Explainability Model Run	Document and run as many types of local explainability Models as is reasonably practical, such as perturbation-based techniques or gradient-based techniques or, for more specific examples, Local Interpretable Model-Agnostic Explanations (LIME), SHAP values, Anchor explanations amongst others. When there is doubt about the stability of the techniques being used, test their quality through alternative parameterizations or by comparing across techniques.	To (a) generate global explainability of the model; (b) help promote model debugging; (c) ensure explainability fidelity and stability through numerous explainability model runs; and (d) highlight associated risks that might occur in the Product Lifecycle.
16.3.4.	Visual Explanations Assessment	Develop visual aids to present and represent Model explainability and transparency insights, such as Tabular Graphics, Partial Dependency Plots, Individual Conditional Expectations, and/or Accumulated Local Effects plot.	To promote explainability comprehension.
16.3.5.	Example-based and Contrastive Explanations Assessment	Develop example-based and contrastive explanations to present and represent Model explainability insights, such as the underlying distribution of the data or select particular instances.	To promote explainability comprehension, such as of complex data distributions and/or datasets for Explainability audiences.

16.4 Production

Objective
To monitor and track the performance of the explanations and trigger when any of the explainability approaches need to be re-trained.

		Control:	**Aim:**
16.4.1.	Explainability Model Thresholds	Set clear performance thresholds and limitations for explainability Model(s).	To (a) define parameters for the continued suitability and performance of explainability Model(s); and (b) highlight associated risks.
16.4.2.	Explainability Model Review & Monitoring	Periodically, or when significant Model changes occur, review implemented explainability Model(s) in light of Explainability Model Thresholds.	To (a) ensure the continued suitability and performance of explainability Model(s) and their explanations; and (b) highlight associated risks.
16.4.3.	Explanation Tracking & Monitoring	Document and conduct (a) ad-hoc interviews, (b) structured surveys, and/or (c) workshops with Explainability Stakeholders on explanations provided and adjust outcomes in Section 14 - Performance Robustness accordingly.	To ensure the continued effectiveness and suitability of provided Model explanations.

Section 17. Security

Objective:

To (a) prevent adversarial actions against, and encourage graceful failures for, Products and/or Models; (b) avert malicious extraction of Models, data and/or intellectual property; (c) prevent Model based physical and/or irreparable harms; and (d) prevent erosion of trust in Outputs or methods.

What do we mean when we refer to Security?

Security is broadly defined as the state of being free from danger or threat. Building on this definition, within the context of machine learning -

> *Security refers to the state of ensuring that machine learning Products and/or Models are free from adversarial danger, threat or attacks.*

Adversarial danger, threat or attacks are understood as the malicious intent to negatively impact machine learning Products' and/or Models' functionality and/or metrics without organisation consent, whether threatened or actualised. If an organisation does consent to any such activity, this is - rather - a form of penetration testing and/or security analysis, as opposed to an adversarial danger, threat or attack.

Why is Security relevant?

Machine learning Product and/or Model security is imperative to ensure operational robust performance. Without the ability to secure the Product's and/or Model's integrity from adversarial danger, threat or attack, malicious third parties can use an organisation's Products and Models to either unlawfully enrich themselves or, more seriously, cause operational environment harms, including death and/or destruction. These are intolerable risks as they undermine organisation, societal and machine learning trust and confidence.

How to apply Security?

In order to generate thorough and thoughtful security, it must be considered continuously throughout all stages of the product lifecycle. This means that security must be addressed at the (a) Product Definition(s), (b) Exploration & Development, (c) Production and (d) Confidence & Trust stages of machine learning operations.

17.1 Product Definitions

Objective
To identify and control for Adversarial risks and motives based on Product Definition, characterized by adversary goals.

		Control:	**Aim:**
17.1.1.	Exfiltration Attacks	Document and assess whether the data employed and gathered by the Product, and the intellectual property generated possess value for potential adversarial actors.	To (a) identify the risks associated with (i) Product Subject physical, financial, social and psychological wellbeing, and (ii) Organization financial wellbeing; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.1.2.	Evasion Attacks	Document and assess whether Product Subjects gain advantage from evading and/or manipulating the Product Outputs. Document and assess whether adversarial actors stand to gain advantage in manipulating Product Subject by evading and/or manipulating Product Output.	To (a) identify the risks associated with Product Output manipulation in regard to malicious and nefarious motives; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.1.3.	Targeted Sabotage	Document and assess whether adversarial actors can cause harm to specific targeted Product Subjects by manipulating Product Outputs.	Document and assess whether adversarial actors can cause harm to specific targeted Product Subjects by manipulating Product Outputs.
17.1.4.	Performance Degradation Attack	Document and assess whether a malicious performance degradation for a specific (Sub) population can cause harm to that (Sub) population. Document and assess whether general performance degradation can cause harm to society, Product Subjects, the Organization, the Domain and/or the field of Machine Learning.	To (a) identify the risks in regard to (i) Product Subjects' physical, financial, social and psychological wellbeing, (ii) the Organization's financial and reputational wellbeing, (iii) society-wide environmental, social and economic wellbeing, and (iv) the Domains' reputational wellbeing; and (b) highlight associated risks that might occur in the Product Lifecycle.

17.2. Exploration & Development

Objective
To identify and control for Adversarial Risks based on and originating in Model properties and/or Model data properties.

		Control:	**Aim:**
17.2.1.	Data Poisoning Assessment	Document and assess the ease and extent with which adversarial actors may influence training data through manipulating and/or introducing - (i) raw data; (ii) annotation processes; (iii) new data points; (iv) data gathering systems (like sensors); (v) metadata; and/or (vi) multiple components thereof simultaneously. If this constitutes an elevated risk, document, assess and implement measurements that can be taken to detect and/or prevent the above manipulation of training data.	To (a) prevent adversarial actors from seeding susceptibility to Evasion Attacks, Targeted Sabotage and Performance Degradation Attacks by way of (i) introducing hard to detect triggers, (ii) increasing noise, and/or (iii) occluding or otherwise degrading information content; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.2.	Public Datasets	Employ public datasets whose characteristics and Error Rates are well known as a benchmark and/or make the Product evaluation results public.	To (a) increase the probability of detection adversarial attacks, such as Data Poisoning, by enabling comparison with and by public resources; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.3.	Data Exfiltration Susceptibility	Document and assess the susceptibility of the Model to data Exfiltration Attacks through - (i) the leakage of (parts of) input data through Model Output; (ii) Model memorization of training data that may be exposed through Model output; (iii) the inclusion by design of (some) training data in stored Model artifacts; and/or (iv) repeated querying of the Model.	To (a) warrant and control the risk of Model data theft; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.4.	Model Exfiltration Susceptibility	Document and assess the susceptibility of Models to Exfiltration Attacks with the aim of obtaining a copy, or approximation of, the Model or other Organization intellectual property, through repeated querying of the Model and analysing the obtained results and confidence scores.	To (a) warrant and control the risk of Model and intellectual property theft; and (b) highlight associated risks that might occur in the Product Lifecycle.

17.2.5.	Exfiltration Defence	To reduce susceptibility of Exfiltration Attacks, (a) make Exfiltration Attacks computationally expensive; (b) remove as much as possible information from Model Output; (c) add noise to Model Outputs through techniques such as differential privacy; (d) limit querying possibilities in volume and/or scope; and/or (e) change Model architecture.	To (a) warrant and control the risk of Exfiltration Attacks; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.6.	Adversarial Input Susceptibility	Document and assess the susceptibility of Models to be effectively influenced by manipulated (inferencing) input. Reduce this susceptibility by (a) increasing the representational robustness (f.e. through more complete embeddings or latent space representation); and/or (b) applying robust transformations (possibly cryptographic) and cleaning.	To (a) warrant the control of the risk of Evasion and Sabotage Attacks, including Adversarial Examples; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.7.	Filtering Susceptibility	If sufficient potential motive has been determined for adversarial attack, document and assess the specific susceptibility of the pre-processing filtering procedures of Models being evaded by tailored inputs, based on the information available to an adversarial attacker about these procedures; in addition to the general Susceptibility Assessment. Increase the robustness of this filtering as far as practically feasible.	To (a) warrant the control of the risk of Evasion and Sabotage Attacks, including Adversarial Examples; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.8.	Training Susceptibility	If sufficient potential motives have been determined for adversarial attack, document and assess the specific susceptibility of Model training to attack through the manipulation of (a) the partitioning of train, validation and test sets, and/or (b) Models' hyperparameters; in addition to the general Susceptibility Assessment. Implement more strict access control on production-grade training and hyperparameter optimization procedures.	To (a) warrant the control of the risk of Evasion, Sabotage and Performance Degradation Attacks; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.9.	Adversarial Example Susceptibility	If sufficient potential motives have been determined for adversarial attack, document and assess the specific susceptibility of Models to Adversarial Examples by considering - (a) sparse or empty regions of the input space, and/or (b) Model architectures; in addition to the general Susceptibility Assessment. Document and implement specific protective measures, such as but not limited to adversarial training.	To (a) warrant the control of the risk of Evasion Attacks, specifically Adversarial Examples; and (b) highlight associated risks that might occur in the Product Lifecycle.

17.2.10.	Adversarial Defence	If sufficient potential motive and susceptibility to adversarial attacks have been determined, implement as far as reasonably practical - (a) data testing methods for detection of outside influence on input and Output Data; (b) reproducibility; (c) increase redundancy by incorporating multimodal input; and/or (d) periodic resets or cleaning of Models and data.	To (a) warrant and control the risk of Adversarial Attacks in general; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.11.	General Susceptibility - Information	Document, assess and control the general susceptibility to attack due to information obtainable by attackers, by considering (a) sensitivity to input noise and/or noise as a protective measure; (b) the amount of information an adversarial actor may obtain from over-extensive logging; and/or (c) whether providing confidence scores as Output is beneficial to adversarial actors.	To (a) warrant and control the risk of Adversarial Attacks in general; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.12.	General Susceptibility - Exploitability	Document, assess and control the general Model susceptibility to attack due to exploitable properties of Models, considering (a) overfit or highly sensitivity Models and Model hyperparameters are easier to attack; (b) an over-reliance on gradient methods that make Models more predictable and inspectable; (c) Models may be pushed past their applicability boundaries if input is not validated; and (d) non-random random number generators might be replaced by cryptographically secure random number generators.	To (a) warrant and control the risk of Adversarial Attacks in general; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.13.	General Susceptibility - Detection	Document, assess and control the capability to detect attacks through the ability to understand when Model behaviour is anomalous by (a) decreasing Model opaqueness, and/or (b) increasing Model robustness.	To (a) warrant and control the risk of Adversarial Attacks in general; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.2.14.	Open Source and Transfer Learning Vulnerability	Document the correspondence between potential attack motives and attack susceptibility posed by using, re-using or employing for transfer learning open source Models, Model weights, and/or Model parameters through - (a) maliciously inserted behaviour and/or code ("trojans"), (b) the ability of an adversarial actor to investigate and attack open source Models unhindered; and (c) improper (re-)use. Consider using non-open source Models or making significant changes aimed at reducing susceptibility.	To (a) warrant and control the risk of Adversarial Attacks in general; and (b) highlight associated risks that might occur in the Product Lifecycle.

17.3 Production

Objective
To identify and control for Adversarial Risks based on and/or originating in Models production.

		Control:	**Aim:**
17.3.1.	IT Security	Traditional IT security practices are referred to. Areas of particular importance to ML-based systems include - (a) backdoor access to the Product, in particular the components vulnerable to attack risk as identified in other controls; (b) remote host servers vulnerability; (c) hardened and isolated systems; (d) malicious insiders (e)man-in-the-middle attacks; and/or (f) denial-of-service.	Traditional IT security practices are referred to. Areas of particular importance to ML-based systems include - (a) backdoor access to the Product, in particular the components vulnerable to attack risk as identified in other controls; (b) remote host servers vulnerability; (c) hardened and isolated systems; (d) malicious insiders (e)man-in-the-middle attacks; and/or (f) denial-of-service.
17.3.2.	Periodic Review and Validation	If motive and risk for Adversarial Attack is high, perform more stringent and frequent review and validation activities.	To (a) warrant and control the risk of Adversarial Attacks in general by increasing detection probability and fixing vulnerabilities quickly; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.3.3.	input and Output Vulnerability	Document and assess the vulnerability of the Product and related systems to direct manipulation of inputs and Outputs. Direct Output manipulation if possible is the most straightforward, simplest, cheapest and hardest to detect attack	To (a) create redundancy with input and inferencing hyperparameter susceptibility; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.3.4.	Defense Strength Assessment	Document and assess the strength and weaknesses of all layers of defense against attacks and identify the weakest links.	To (a) build defense in depth; and (b) highlight associated risks that might occur in the Product Lifecycle.

17.4 Confidence & Trust

Objective
To identify and control for Adversarial Risks based on and/or originating in Product trust and confidence.

		Control:	**Aim:**
17.4.1.	Trust Erosion	Document and assess the potential impact on trust from adversarial and defacement attacks, and establish a strategy to mitigate trust erosion in case of successful attacks.	To (a) prevent erosion of trust in Product Outputs, the Product, the Organization, and/or Domains from preventing beneficial Products and technologies to be employed; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.4.2.	Confidence	Document and assess the degree of over- and under-confidence in the Product output by Product Team, Stakeholder(s) and End Users. Encourage an appropriate level of confidence through education and self-reflection. Note: Underconfidence will lead to users over-ruling the Product in unexpected ways. Overconfidence leads to lower scrutiny and therefore lowers the chance of detection and prevention of attacks.	To (a) balance the risk of compromising Product effects against reduced vigilance; and (b) highlight associated risks that might occur in the Product Lifecycle.
17.4.3.	Warning Fatigue	Document and assess the frequency of warnings and alerts provided to Product operators, maintainers, and Product Subjects, and refine the thresholds and processes such that no over-exposure to alerts can lead to systematic ignoring of alerts.	To (a) prevent an overexposure to alerts that can lead to ignoring serious defects and incidents, causing harm; and (b) highlight associated risks that might occur in the Product Lifecycle.

Section 18. Safety

Objective:

To (a) prevent Model-based physical and/or irreparable harms; and (b) identify and mitigate risks due to Product failures, including Model failures, IT failures, and process failures.

What do we mean when we refer to Safety?

When referring to safety in the context of machine learning we mean-

> *Safety means the protection of the operational environment - and its subjects - from negative physical and/or other harms that might result from machine learning Products and/or Models, either directly or indirectly.*

Put slightly differently, when we discuss safety we are not talking about the safety of machine learning Products and Models (called, rather, security), but, instead, the operational environment within which machine learning Products and Models operate. Specifically, the harms and risks that machine learning Products and Models might cause for these environments and their subjects. For example, an autonomous vehicle crashing and causing injury, death, or destruction.

Why is Safety important?

Machine learning Product and Model safety is imperative to ensure the integrity of the operational environment. Without such safety, machine learning Products and Models can cause grave operational environment harms, such as physical injury or, at worst, death. These are intolerable risks as they undermine organisation, societal and machine learning trust and confidence. Moreover, they cause irreparable real damage in the real world.

How to apply Safety?

In order to generate thorough and thoughtful safety, it must be considered continuously throughout all stages of the product lifecycle. This means that safety must be addressed at the (a) Product Definition(s), (b) Exploration, (c) Development and (d) Production stages of machine learning operations.

18.1 Product Definitions

Objective
To establish the appropriate safety-oriented attitudes based on first principles and Product Definitions.

		Control:	Aim:
18.1.1.	Physical and Irreparable Harm Risk	Document and assess whether any likely failure modes can cause physical and/or irreparable harm, based on the Product Definitions. If such is the case, warrant increased oversight and attention throughout the Product Lifecycle to risks and controls in general and from this section in particular.	To warrant the necessary amount of control and resources throughout the Product Lifecycle with regard to preventing and mitigating significant threats to individuals' physical, financial, social, and psychological well being.
18.1.2.	Domain-first Humble Culture	Document and establish tenents for Product Team culture to promote risk awareness and prevent blind spots, inclusive of (a) put Domain expertise central; (b) never assume only positive effects; (c) admit uncertainty when assessing impacts.	To promote risk awareness and prevent blindspots in analysing failure modes and other safety related controls and (b) highlight associated risks that might occur in the Product Lifecycle.

18.2 Exploration

Objective
To start the process of identifying,specifying and controlling (potential) risks and failures modes of the Model(s) and Product based on research and exploration, and sustain this process throughout the Product Lifecycle.

		Control:	Aim:
18.2.1.	Forecast Failure Modes	(a) Document and assess continuously throughout the Product Lifecycle all potential failure modes that can be identified through - (i) researching past failures; and/or (ii) interrogating all components or product and context with an open mind; (b) rank identified failure modes according to likelihood and severity; (c) prepare for and mitigate these risks as far as is reasonably practical in order of risk throughout the Product Lifecycle.	To (a) reduce harmful consequences of failures through anticipation and preparation; and (b) highlight associated risks that might occur in the Product Lifecycle.
18.2.2.	Prediction Limits	Document and assess with a diversity of Stakeholders the real limitations on the Product and Model Outcomes that ought be strictly enforced in order to prevent physical and/or irreparable harm and/or other Failure Modes.	To prevent Model and Product Outcomes from violating clear, fixed and safe operating bounds.
18.2.3.	Surprise Diary	Document continuously throughout the Product Lifecycle all surprising findings and occurrences.	To discover and subsequently control for previously unknown or unanticipated failures modes.

18.3 Development

Objective
To control Safety risks and failure modes based on testing and controlling Model and Product technical components.

		Control:	**Aim:**
18.3.1.	General IT Testing Practices	Adhere to all traditional IT/Software Testing best practices.	To warrant and control the risk of failures due to code, software and other IT mistakes in general.
18.3.2.	Testing by Domain Experts	Document and perform testing of the Model(s) and Product by Domain experts.	To warrant that Product and Product Safety are tested against the most relevant requirements and prevent blind spots caused by lack of multidisciplinarity.
18.3.3.	Algorithm Benchmarking	Document and perform benchmark testing of Models and Model code against well-known, trusted and/or simpler Models/code.	To warrant the correct implementation of Models and code, and safeguard reproducibility in general.

18.4 Production

Objective
To control for Safety risks and failure modes and prevent physical and/or irreparable harm by performing assessments and implementing measures at the systemic and organisational level.

		Control:	**Aim:**
18.4.1.	System Failure Propagation	Document and assess how failures in Models and Product components propagate to other components and other systems, and what damage they may cause there. Incorporate such information in Failure Mode risk assessments and implementation of Graceful Failures and Kill Switches.	To (a) prevent blind spots and cascading failures and (b) provide essential input for creating mitigation measures with a minimum of uncontrolled side-effects.
18.4.2.	Graceful Failure	Document and assess whether (i) Model errors, (ii) Model failures, (iii) Product failures, (iv) IT failures, and/or (v) process and implementation failures - whether caused by attack or not - can result in physical or irreparable harm to humans, society and/or the environment. If present, mitigate these risks by implementing technological and/or process measures that make these failures graceful.	To identify risks and mitigating measures throughout the Product Lifecycle with regard to significant threats to individuals' physical, financial, social and psychological wellbeing.

18.4.3.	Kill Switch	Document and implement Kill Switches according to the findings of all previous controls, taking into account (a) instructions and procedures for engaging the Kill Switch; (b) who is/are responsible for engaging the Kill Switch; (c) what impacts the engagement of the Kill Switch has on users, other parts of the Product and other systems.	Document and implement Kill Switches according to the findings of all previous controls, taking into account (a) instructions and procedures for engaging the Kill Switch; (b) who is/are responsible for engaging the Kill Switch; (c) what impacts the engagement of the Kill Switch has on users, other parts of the Product and other systems.
18.4.4.	Safety Stress Testing	Document and perform scenario-based stress testing of Product in Domain, Society and Environmental contexts, for realistic but rare high-impact scenarios, recording the Product's reaction to and influence on the Domain, Society and Environment.	To control and prepare for worst-case scenarios in the context of rapid and/or large changes in Domain, in Society or in Environment.
18.4.5.	Product Incident Response	Document and prepare Product-specific implementation of the Organisation Incident Response Plan insofar as that does not cover the Product's specific risks, if appropriate.	To (a) control for and contain Product Incidents; (b) minimize harms stemming from Product Incidents; (c) repair harms caused by Product Incidents; and (d) incorporate lessons learned.

Section 19. Human-Centred Design

Objective:

To ensure (a) building desirable solutions; (b) human control over Products and Models; and (c) that individuals affected by Product and Model outputs can obtain redress.

What is Human-centric Design?

Human-centric (or also called human-centered) Design is a creative approach to problem-solving, by involving the human perspective in all steps of problem solving.

In the context of machine learning and the Model framework, Human-centric Design makes you stay focused on the user when designing with ML, therefore building desirable solutions for your target users. Moreover, it also ensures that the right stakeholders are involved throughout the whole design and development process and helps to properly identify the right opportunity areas. Lastly, human-centric design encompasses the extent to which humans have control over the Model and its output as well as the degree to which humans can obtain redress, if they are affected by the Model.

Why Human-centric Design?

Incorporating Human-centric Design in the Product is vital. It ensures the Model is not built in isolation but is integrated with other problems and, most of all, that it helps solve the right questions. Having the right Stakeholders (beyond the technical teams) involved in the whole Model lifecycle translates to higher trust levels in the Model, increases the rate of adoption, as well as results in more human-friendly and creative solutions. Not having the human-centric part of the Model will inevitably result in an inferior Model - and one which very likely end up on a 'shelf' and, therefore, not be applied in practice.

How to ensure Human-centric Design?

Human-centric Design is something that needs to be addressed throughout the product lifecycle, not only in the early stages of it, and not in any stage in isolation.

19.1 Product Definitions

Objective
To discover and gain insight so that the Product and Model(s) will solve the right problems, designed for human needs and values, before building it.

		Control:	**Aim:**
19.1.1.	Human Centered Machine Learning	Incorporate the human (non-technical) perspective in your (technical) process of exploration, development and production by applying user research, design thinking, prototyping and rapid feedback, and human factors when defining a usable product or model.	To (a) ensure that Product(s) and Model(s) are not only feasible and viable, but also align with a human needs; and (b) highlight associated risks failing such.
19.1.2.	UX (or user) research	Focus on understanding user behaviours, needs, and motivations through observation techniques, task analysis, user interviews, and other research methodologies.	To prevent (a) a focus on technology from overshadowing a focus on problem solving; and (b) cognitive biases from adverse influence Product and Model design.
19.1.3.	Design for Human values	Include activities for (a) the identification of societal values, (b) deciding on a moral deliberation approach (e.g. through algorithms, user control or regulation), and (c) methods to link values to formal system requirements (e.g. value sensitive design (VSD) mapping).	To reflect societal concerns about the ethics of AI, and ensure that AI systems are developed responsibly, incorporating social, ethical values and ensuring that systems will uphold human values. The moral quality of a technology depends on its consequences.

19.2 Exploration

Objective
To (a) cluster, (b) find insights and (c) define the right opportunity area, ensuring to focus on the right questions to solve in preparation for the development and production phase.

		Control:	**Aim:**
19.2.1.	Design Thinking	Ensure an iterative development process by (a) empathize: research your users' needs, (b) define: state your users' most important needs and problems to solve, (c) ideate: challenge assumptions and create ideas, (d) prototype: start to create solutions and (e) test: gather user feedback early and often.	To let data scientists organise and strategise their next steps in the exploratory phase.

19.2.2.	Ethical assessment	Discuss with your team to what extend (a) the AI product actively or passively discriminates against groups of people in a harmful way; (b) everyone involved in the development and use of the AI product understands, accepts and is able to exercise their rights and responsibilities; (c) the intended users of an AI product can meaningfully understand the purpose of the product, how it works, and (where applicable) how specific decisions were made.	Discuss with your team to what extend (a) the AI product actively or passively discriminates against groups of people in a harmful way; (b) everyone involved in the development and use of the AI product understands, accepts and is able to exercise their rights and responsibilities; (c) the intended users of an AI product can meaningfully understand the purpose of the product, how it works, and (where applicable) how specific decisions were made.
19.2.3.	Estimating the value vs effort of possible opportunity areas	Explore the details of what mental Models and expectations people might bring when interacting with an ML system as well as what data would be needed for that system. E.g. an Impact Matrix.	To reveal the automatic assumptions people will bring to an ML-powered product, to be used as prompts for a product team discussion or as stimuli in user research. (See also Section 4.11 - User Experience Mapping.)

19.3 Development

Objective
To (a) ensure rapid iteration and targeted feedback from relevant Stakeholders, allowing a larger range of possible solutions to be considered in the selection process. (b) Increase the creativity and options considered, while avoiding avoiding personal biases and/or pigeon-holing a solution.

		Control:	**Aim:**
19.3.1.	Prototyping	1: Focus on quick and minimum viable prototypes that offer enough tangibility to find out whether they solve the initial problem or answer the initial question. Document how test participants react and what assumptions they make when they "use" your mockup. 2: Design a so-called 'Wizard of Oz' test; have participants interact with what they believe to be an autonomous system, but which is actually being controlled by a human (usually a team member)	To gain early feedback (without having to actually have build an ML product) needed to (a) adjust or pivot your Products(s) and/or Model(s) thus ensuring business viability; and/or (b) assess the cost and benefits of potential features with more validity than using dummy examples or conceptual descriptions.
19.3.2.	Cost weighing of false positives and false negatives	While all errors are equal to an ML system, not all errors are equal to all people. Discuss with your team how mistakes of your ML model might affect the user's experience of the product.	to avoid sensitive decisions being taken (a) autonomously; or (b) without human consideration.

19.3.3.	Visual Storytelling	Focus on explanatory analysis over exploratory analysis, taking the mental models of your target audience in account.	To avoid uninformed decisions about your product or model by non-technical stakeholders, when presenting complex analysis, models, and findings.
19.3.4.	Preventative Process Design	Document and assess whether high-risk and/or high-impact Model (sub)problems or dilemmas that are present in the Product (as determined from following the Best Practices Framework) can be mitigated or avoided by applying non-Model process and implementation solutions. If non-Model solutions are not applied, document the reasons for this, document the sustained presence of these risks and implement appropriate incident response measures.	To (a) prevent high-risk and/or high-impact Model (sub)problems or dilemmas through non-Model process and implementation solutions; and (b) highlight associated risks that might occur in the Product Lifecycle.

19.4 Production

Objective
To ensure (a) delivering a user-friendly product, (b) increasing the adoption rate of your product or model, focussing on (dis-)trust as main fundamental risk of ML models with (non-technical) end users

		Control:	**Aim:**
19.4.1.	Trust; increased by design	Allow for users to develop systems heuristics (ease of use) via design patterns while at the same time facilitate a detailed understanding to those who value the 'intelligent' technology used. (See Section 19.4.2 -Design for Human Error; Section 19.4.3 - Algorithmic transparency; and Section 19.4.4 - Progressive disclosure for further information.)	To avoid (a) that the user does not trust the outcome, and will act counter to the design, causing at best inefficiencies and at worst serious harms, or (b) that -trusting an application will do what we think it will do- an user can confirm their trust is justified.
19.4.2.	Design for Human Error	(a) Understand the causes of error and design to minimise those causes; (b) Do sensibility checks. Does the action pass the "common sense" test (e.g. is the number is correct? - 10.000g or 10.000kg) (c) Make it possible to reverse actions - to "undo" them - or make it harder to do what cannot be reversed (eg. add constraints to block errors - either change the color to red or mention "Do you want to delete this file? Are you sure?"). (d) make it easier for people to discover the errors that do occur, and make them easier to correct	To (a) increase trust between the end user and the model; (b) minimize the opportunities for errors while also mitigating the consequences. Increase the trust users have with your product by design for deliberate mis-use of your model (making your model or product "idiot-proof") so users are (a) able to insert data to compare the model outcome with their own expected outcome which will increase their trust, or (b) users able to test the limitations of your product or model -via fake or highly unlikely data- without breaking your product or model.

19.4.3.	Algorithmic transparency	Assess the appropriate system heuristics (eg. ease of use), document all factors that influence the algorithmic decisions, and use them as a design tool to make them visible, or transparent, to users who use or are affected by the ML systems.	To (a) increase trust between the end user and the model; (b) increase end-user control; (c) improve acceptance rate of tool; (d) promote user learning with complex data; and (e) enable oversight by developers.
19.4.4.	Progressive disclosure	At the point where the end-user interacts with the Product outcomes, show them only the initial features and/or information necessary at that point in the interaction (thus initially hiding more advanced interface controls). Show the secondary features and/or information only when the user requests it (show less, provide more-principle).	To greatly reduce unwanted complexity for the end-user and thus preventing (a) end-user non-adoption or misunderstanding and (b) ensuring an increased feeling of trust by the users.
19.4.5.	Human in the loop (HITL)	Embed human interaction with machine learning systems to be able to label or correct inaccuracies in machine predictions.	To avoid the risk of the Product applying a materially detrimental or catastrophic Product Outcome to a Product Subject without human intervention.
19.4.6.	Remediation	Document, assess and implement in the Model(s), Product and Organization processes, requirements for enabling Product Subjects to challenge and obtain redress for Product Outcomes applied to them.	To ensure detrimental Product Outcomes are easily reverted when appropriate.

Section 20. System Stability

Objective:

To prevent (in)direct adverse social and environmental effects as a consequence of interactions amongst Products, Models, the Organisation, and the Public.

What is Systemic Stability?

Model stability is a relatively popular notion. It is usually centered at putting a bound at the Model's generalization error.

> *Systemic Stability refers to the robustness of the Model (or lack thereof) stemming from the interaction between the Model, Organization, environment and the broader public (society at large).*

There are numerous potential risks that can emerge in this interaction. Many of them can impact the stability of the Model - beyond the context of traditional performance robustness or deterioration of the Model over time. Another way to think of it is as the extent to which the Model and/or its building blocks are susceptible to chain effects and self-reinforcing interactions between the Model, Organization, environment and society.

Why Systemic stability?

Systemic stability forces one to think beyond the traditional definitions of Model stability and its potential causes and consequences. Systemic stability ensures that we consider the effect on the Model and society due to the interaction between the Model, the Organization, the environment and society. This means thinking about susceptibility to feedback loops, self-fulfilling prophecies and how either of them may impact the data or the Model and its output. It, therefore, reduces risks related to deteriorated performance and minimises the propagation of undesirable biases.

How to ensure Systemic stability?

In order to ensure systemic stability, it must be considered continuously throughout all stages of the product lifecycle. This means that systemic stability must be addressed at the (a) Product Definition(s), (b) Exploration, (c) Development and (d) Production stages of machine learning operations.

20.1 Product Definitions

Objective
To investigate and mitigate unforeseen social and environmental chain effects and/or risks caused through Product Definition(s).

		Control:	**Aim:**
20.1.1.	Product Assumption Susceptibility	Document and assess whether applying Product Outputs will result in invalidating Product Assumptions. If so, attempt to redefine Product Assumptions to warrant their longevity.	To (a) prevent unpredictable social and/or environmental behaviour through Product Outcomes; and (b) highlight associated risks in the Product Lifecycle.

20.2 Exploration

Objective
To investigate and mitigate unforeseen social and environmental chain effects and/or risks caused through Product exploration.

		Control:	**Aim:**
20.2.1.	Selection Function Susceptibility	Document and assess whether applying Product Outputs will result in invalidating Product Assumptions. If so, attempt to redefine Product Assumptions to warrant their longevity.	To (a) prevent unpredictable social and/or environmental behaviour through Product Outcomes; and (b) highlight associated risks in the Product Lifecycle.
20.2.2.	Data Definition Susceptibility	Document and assess whether applying Product Outputs will result in changes to the Selection Function, and whether this is a self-reinforcing interaction. If true, attempt to mitigate or stabilize associated effects through refining Product Definition(s) and/or improving Model design and/or Product and process implementation.	To (a) determine and prevent Product and/or Model risk in - (i) progressively strengthening biases (from encoded assumptions and definitions to datasets to algorithms chosen); (ii) progressively reinforcing Model errors and/or Product generalizations; (iii) progressively losing sensitivity to data and/or Domain changes; (iv) suffering from self-reinforcing and/or exponential run-away effects; (b) determine and prevent risks of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.

20.2.3.	Data Generating Process Susceptibility	Document and assess whether applying Product Outputs will result in changes to the Product data definitions, and whether this is a self-reinforcing interaction. If true, attempt to mitigate or stabilize associated effects through refining Product Definition(s) and/or improving Model design and/or Product and process implementation.	To (a) determine and prevent Product and/or Model risk in - (i) progressively strengthening biases (from encoded assumptions and definitions to datasets to algorithms chosen); (ii) progressively reinforcing Model errors and/or Product generalizations; (iii) progressively losing sensitivity to data and/or Domain changes; (iv) suffering from self-reinforcing and/ or exponential run-away effects; (b) determine and prevent risks of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.
20.2.4.	Data Distributions Susceptibility	Document and assess whether applying Product Outputs will result in changes to the data generating process, and whether this is a self-reinforcing interaction. If true, attempt to mitigate or stabilize associated effects through refining Product Definition(s) and/or improving Model design and/or Product and process implementation.	To (a) determine and prevent Product and/or Model risk in - (i) progressively strengthening biases (from encoded assumptions and definitions to datasets to algorithms chosen); (ii) progressively reinforcing Model errors and/or Product generalizations; (iii) progressively losing sensitivity to data and/or Domain changes; (iv) suffering from self-reinforcing and/ or exponential run-away effects; (b) determine and prevent risks of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.
20.2.5.	Hidden Variable Susceptibility	Document and assess whether applying Product Outputs will result in the creation of new hidden Variables in the system. If true, record the new Variable during data gathering, or prevent the creation of the new Variable through improved Product Definition(s) and implementation.	To (a) determine and prevent Product and/or Model risk in - (i) progressively strengthening biases (from encoded assumptions and definitions to datasets to algorithms chosen); (ii) progressively reinforcing Model errors and/or Product generalizations; (iii) progressively losing sensitivity to data and/or Domain changes; (iv) suffering from self-reinforcing and/ or exponential run-away effects; (b) determine and prevent risks of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.

20.3 Development

Objective
To investigate and mitigate unforeseen social and environmental chain effects and/or risks caused through Product development.

		Control:	Aim:
20.3.1.	Target Feature Definition Susceptibility	Document and assess whether applying Product Outputs will result in changes to the Target Feature definition. If true, attempt to mitigate associated effects through refining Product Output and/or Model design and/or development.	To (a) determine and prevent risk of unpredictable behaviour once the Product outcomes are applied; and (b) highlight associated risks in the Product Lifecycle.
20.3.2.	Optimization Feedback Loop Susceptibility	Document and assess whether the cost function and/or optimization algorithm exhibits a feedback loop behaviour that includes the gathering of data that has been influenced by previous Model iterations, and whether this behaviour is self-reinforcing or self-limiting. If true, attempt to mitigate associated effects through refining Product Output and/or Model design and/or development.	Idem Section 20.2.1- Selection Function Susceptibility

20.4 Production

Objective
To investigate and mitigate unforeseen social and environmental chain effects and/or risks caused through Product application.

		Control:	Aim:
20.4.1.	Self-fulfilling Prophecies	Document and assess whether applying Product Outputs will result in change to Product inputs, dependencies and/or Domain(s) (other than those mentioned in controls elsewhere) and whether this is a self-reinforcing interaction. If true, attempt to mitigate associated effects through refining Product Output and/or Model design and/or development.	Idem Section 20.2.1- Selection Function Susceptibility
20.4.2.	Hidden Variable Dependencies	Document and assess whether the effect of applying Product Outputs depends on Hidden Variables. If true, control for Hidden Variables, for example through marginalization and/or by deriving indicators for Hidden Variables influence.	To (a) prevent diverging effects on seemingly similar individuals or datapoints; (b) prevent or detect high-risk interactions; and (c) highlight associated risks in the Product Lifecycle.

20.4.3.	Society Susceptibility	Document and assess whether applying Product Outputs results in potentially harmful societal or environmental changes, and research the possible knock-on effects as far as reasonably practical.	To (a) identify and prevent both direct and indirect adverse effects on society and the environment; (b) determine if there is a risk of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.
20.4.4.	Domain Susceptibility	Document and assess whether applying Product Outputs results in changes to application Domain(s), and research the possible knock-on effects as far as reasonably practical.	To (a) identify and prevent both direct and indirect adverse effects on Product Domain(s); (b) determine if there is a risk of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.
20.4.5.	Other Organisation Products Susceptibility	Document and assess whether applying Product Outputs result in changes to inputs, dependencies and/or context for other Organisation Products. If true, attempt to mitigate associated effects through refining Product Output and/or Model design and/or development.	To (a) identify and prevent both direct and indirect adverse effects on the Organisation or other Organisation Products; (b) determine if there is a risk of unpredictable behaviour once the Product Outcomes are applied; and (c) highlight associated risks in the Product Lifecycle.

Section 21. Product Traceability

Objective:

To ensure the clear and complete Traceability of Products, Models and their assets (inclusive of, amongst other things, data, code, artifacts, output, and documentation) for as long as is reasonably practical.

What do we mean when we refer to Product Traceability?

Product Traceability refers to the ability to identify, track and trace elements of the Product as it is designed, developed, and implemented.

Alternatively put, Product Traceability, is the ability to trace and track all Product elements and decisions throughout the Product Lifecyle. It is the identification, indexing, storage, and management of each unique Product element.

Why is Product Traceability relevant?

Through Product Traceability, each element of the Product can be easily identified and, thereafter, re-examined, and amended. This allows for greater Product accountability and transparency as, through this process, each Product element and its developers can be identified.

How to apply Product Traceability?

In order to generate thorough and thoughtful Product Traceability, it must be considered continuously throughout all stages of the Product Lifecycle. This means that Product Traceability must be addressed at the (a) Product Definition(s), (b) Exploration, (c) Development and (d) Production stages of Machine Learning Operations.

21.1 Product Definitions

Objective
To document and maintain an overview of the requirements necessary to complete the Product and the interdependencies in the Product design phase.

		Control:	**Aim:**
21.3.1.	Document Storage	Define a single fixed storage solution for all reports, documents, and other traceability files.	To (a) prevent the usage and dissemination of outdated and/ or incorrect files; (b) prevent the haphazards storage of Product reports, documents and/or files; and (c) highlight associated risks in the Product Lifecycle.
21.3.2.	Version Control of Documents	Ensure that document changes are tracked when changes are made. Subsequent versions ought to list version number, author, date of change, and short description of the changes made.	To (a) track changes to any and all documents; (b) ensure everyone is using the same and latest document version; and (c) highlight associated risks in the Product Lifecycle.
21.3.3.	Architectural Requirements Document	Document which information technology resources are necessary for each element of the Product to provide a necessary overview of system requirements and cost distribution. Document the reasons each resource was chosen along with justifications.	To (a) provide clear documentation of which system resources are used, where they are used, why they are used, and costs; and (b) highlight associated risks in the Product Lifecycle.

21.2 Exploration

Objective
To document the impact analysis of each requirement.

		Control:	**Aim:**
21.2.1.	Document Impact Analysis of Requirements	Document and complete an impact analysis on the resources and design of the Product that can result in technical debt.	To (a) avoid Product failures due to unresolved technical debt by documenting potential sources of friction and the solutions; and (b) highlight associated risks in the Product Lifecycle.
21.2.2.	Resource Traceability Matrix	Provide and keep up to date a clear view of the relationships and interdependencies between resources in a documented matrix.	To (a) document and show resource coverage for each use case; and (b) highlight associated risks in the Product Lifecycle.

21.2.3.	Design Traceability Matrix	Provide and keep up to date a clear view of the relationships and interdependencies between designs and interactions thereof in a documented matrix.	To (a) document design and execution status; (b) clearly trace current work and what can be pursued next; and (c) highlight associated risks in the Product Lifecycle.
21.2.4.	Results Reproducibility Logs	Throughout the entire Product Lifecycle, whenever a Product component - inclusive of Models, experiments, analyses, transformation, and evaluations - are run, all parameters, hyperparameters and results ought to be logged and/or tracked, including unique identifier(s) for runs, artifacts, code and environments.	To (a) enable Absolute Reproducibility; (b) validate Models and Outcomes through enablement of analysis of logs, run comparisons and reproducibility.

21.3 Development

Objective
To document and maintain the status of each product and the testing results. Ensure 100% test coverage. Prevent inconsistencies between project elements and prevent feature creep.

		Control:	**Aim:**
21.3.1.	Backlog	Ensure that an effective backlog is maintained to track work items and serve as a historical representation and timeline of completed features and velocity.	To (a) ensure a comprehensive breakdown of Features and tasks necessary to achieve full product functionality; (b) provide highly readable coarse-grained versioning; and (c) highlight associated risks in the Product Lifecycle.
21.3.2.	Documentation for Technical Contributors	Maintain technical documentation that enables all current and future contributors to efficaciously and safely develop and maintain the Product, including such information as description of each file, the workflow, author, environments, accrued technical debt.	To (a) maintain Product technical integrity by ensuring safe contribution and maintenance practices; and (b) highlight associated risks in the Product Lifecycle.
21.3.3.	Version Control of Code	Maintain uninterrupted version control systems and practices of all code used by, in and during the Product and its Lifecycle.	To (a) maintain Product technical integrity by ensuring safe contribution and maintenance practices; and (b) highlight associated risks in the Product Lifecycle.

21.3.4.	Docstrings and Code Comments	Document in each function the author of code, purpose of code, input, Output, and improvements to be made. Document the source of inputs and potentially a short business description of data used.	To (a) ensure Model clarity as to technical progress; and (b) highlight associated risks in the Product Lifecycle.
21.3.5.	Project Status Reports	Ensure that all status reports and similar communications to Management and Stakeholders are stored and maintained, inclusive of team updates, reports to the Product Manager, and Stakeholder reports by request.	To (a) maintain a formal written record of decisions, progress and context evolution; and (b) highlight associated risks in the Product Lifecycle.

21.4 Production

Objective
To document the observed impact of updates to the product. Document product runs and their input for reproducibility.

		Control:	**Aim:**
21.4.1.	Version control through CI/CD	Maintain distinct production versions to easily revert or roll back to a working previous Product, if production issues arise. Properly set up CI/CD enables easy redeploy of any artifact and version.	To (a) provide functional Product to users at all times; (b) seamlessly redeploy Product versions if needed; and (c) highlight associated risks in the Product Lifecycle.
21.4.2.	Data Lineage Manifest	Utilise a data lake for production data, intermediate results, and end results. Each step should be documented in a manifest that is passed from one step of the process to the next and always accompanies stored data and results.	To (a) create a structured way for tracing where data has been, what was done to it, and results; and (b) highlight associated risks in the Product Lifecycle.