Workshop 3

Samuel Aljure Bernal - 20202020111
Carlos Alberto Barriga Gámez - 20222020179
David Santiago Aldana Gonzalez - 20222020158
Juan Diego Alvarez Cristancho - 20221020076

November 2025

1 Review and Refine System Architecture

The refined architecture of the *Chocolates 4U Predictive System* has been redesigned to reflect a structured framework that emphasizes reproducibility, traceability, and modular quality control. Unlike the earlier distributed concept, this version focuses on a contained architecture that integrates all processes within a single controlled environment, ensuring stability and compliance with engineering quality standards.

1.1 System Architecture Description

The architecture is divided into three fundamental layers, each with specific modules that interact through a well-defined data flow and feedback control mechanisms (Figure 1). This design ensures modularity, maintainability, and fault isolation, all aligned with ISO 9001 and CMMI Level 2–3 principles for managed and defined processes.

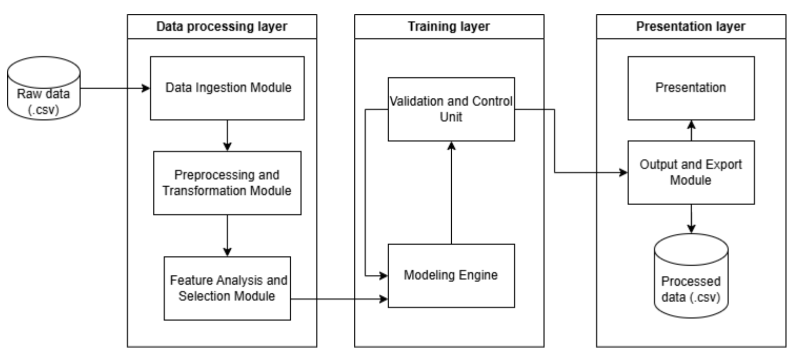

Figure 1: Refined System Architecture Diagram

1.2 Layer Descriptions and Quality Integration

1.2.1 Data Processing Layer

This layer manages the acquisition, cleaning, and preparation of input data. It ensures data integrity and consistency through structured pipelines and validation checkpoints.

- **Data Ingestion Module:** Handles loading of `.csv` files, performing schema validation, type checking, and initial quality assessment before integration into the workflow.
- **Preprocessing and Transformation Module:** Executes normalization, encoding, and missing-value imputation to produce clean, standardized datasets suitable for analysis.
- **Feature Analysis and Selection Module:** Applies correlation analysis and ANOVA-based filtering to select the most relevant variables for modeling, minimizing noise and redundancy.

ISO 9001 alignment: Quality assurance is achieved through documented preprocessing routines and verification steps that ensure input data meet established specifications.

CMMI alignment: The process is traceable and repeatable, supporting continuous improvement through measured data-handling performance indicators.

1.2.2 Training Layer

This layer represents the core analytical process, where predictive models are trained, validated, and managed for consistency and performance reliability.

- **Modeling Engine:** Executes regression algorithms (linear, regularized, and ensemble methods) using controlled parameters to maintain reproducibility across runs.
- **Validation and Control Unit:** Performs model evaluation via cross-validation, monitors performance metrics (e.g., MAE), and ensures parameter consistency across iterations.

ISO 9001 alignment: Incorporates quality control checkpoints for model verification and accuracy validation.

CMMI alignment: Implements controlled experimentation and version management practices, establishing baseline models and maintaining configuration integrity throughout the training cycle.

1.2.3 Presentation Layer

The final layer focuses on the visualization and communication of predictive outcomes, providing end users with interpretable and exportable results.

- **Presentation Module:** Generates visual or tabular summaries of model performance and results, supporting analytical interpretation.
- **Output and Export Module:** Exports predictions and processed datasets to standardized `.csv` files, ensuring reproducible external reporting.

ISO 9001 alignment: Output validation ensures that deliverables conform to the documented reporting format and meet defined accuracy thresholds.

CMMI alignment: The output process is linked to verification and validation activities, maintaining documentation of results, versions, and control records.

1.3 Architectural Principles and Compliance

The refined system architecture embodies key principles of engineering design and quality management:

- **Modularity:** Each layer operates as an independent yet coordinated component, simplifying maintenance and upgrades.
- **Traceability:** All transformations, models, and results are linked through documented metadata, supporting reproducibility and auditability.
- **Fault Tolerance:** Validation checkpoints and controlled data flow minimize the impact of processing errors.
- **Scalability:** The modular structure supports future integration of additional models or preprocessing methods.
- **Quality and Continuous Improvement:** Guided by ISO 9001 principles, each process includes feedback mechanisms for iterative enhancement.

This architecture consolidates the system within a coherent, quality-oriented framework.

By aligning its design with ISO 9001 (Quality Management Systems) and CMMI (Capability Maturity Model Integration), the Chocolates 4U Predictive System achieves structural robustness, process transparency, and operational traceability. The result is a well-defined system that not only enables accurate predictive modeling but also incorporates engineering principles.

2 Quality and Risk Analysis

This section identifies the main risks and potential failure points that could affect the reliability, reproducibility, and accuracy of the *Chocolates 4U Predictive System*. Given that the system operates in a localized environment (executed from a single workstation) the primary focus of the risk analysis is on data integrity, model robustness, dependency control, and process documentation rather than distributed system concerns. The mitigation plan follows principles from ISO 9001 (Quality Management), CMMI Level 3 (Process and Configuration Management), and Six Sigma (Defect Prevention and Continuous Improvement).

2.1 Risk Identification and Mitigation Strategies

Table 1 summarizes the key technical and procedural risks identified in the system design, along with their root causes, impact levels, and mitigation actions. Each mitigation strategy seeks to maintain reproducibility, consistency, and recoverability through preventive and corrective measures implemented in code and documentation.

Table 1: Risk and Quality Management Plan (Localized Execution Context)

Risk / Failure Point	Root Cause	Impact Level	Mitigation Strategy	Reference Standard
Data Corruption or Missing Values	Manual data handling, inconsistent CSV formatting, or accidental overwriting.	High	Implement validation scripts that verify file integrity, schema structure, and missing values before processing. Maintain backup copies of input datasets and use version-controlled folders for all data sources.	ISO 9001 Quality Assurance
Dependency or Environment Failure	Python or R package conflicts, missing libraries, or version incompatibilities.	Medium–High	Use reproducible environments (`conda`, `renv`) and specify dependencies in `requirements.txt` or environment configuration files. Document versions and include a session summary for traceability.	CMMI Verification and Validation
Model Overfitting or Drift	Limited data diversity or overly complex regression models.	Medium	Apply k-fold cross-validation, regularization, and hold-out testing. Monitor model performance across different random seeds to ensure stability and generalization.	Six Sigma (Control phase)
Loss of Reproducibility	Untracked code changes or undocumented preprocessing steps.	High	Use Git for version control, maintain structured execution logs, and store data about parameters, date, and performance metrics.	CMMI Configuration Management
Execution Error or Process Interruption	Unexpected data types, file path errors, or manual interruption during execution.	Medium	Add `try--except` (Python) and `tryCatch()` (R) blocks to prevent crashes. Save intermediate outputs and include checkpoints that allow recovery without restarting the full process.	ISO 9001 Process Control

2.2 Monitoring and Response Plan

Since the project operates within a single-computer environment, quality monitoring and risk response are based on systematic validation, structured logging, and periodic manual reviews rather than automated alerts. The following mechanisms ensure early detection and recovery from potential issues:

- **Execution Logs:** Each major stage (data ingestion, preprocessing, modeling, validation) generates a time-stamped log file summarizing the inputs, configurations, and results obtained. These logs provide traceability and facilitate debugging when inconsistencies appear.
- **Manual Checkpoints:** Key milestones within the workflow (e.g., after preprocessing or model training) include checkpoint saves that allow the process to be resumed if interrupted.
- **Environment Verification:** Before execution, a diagnostic script verifies package versions, file paths, and variable types, reducing the likelihood of runtime errors.
- **Version Control and Review:** Code commits are reviewed and tagged according to milestones (e.g., *v1.0_data-cleaning*, *v1.1_model-training*) to document the evolution of the project.

3 Project Management Plan

The project management plan establishes the organizational and procedural framework for developing, validating, and maintaining the *Chocolates 4U Predictive System*. Since the project is executed in a local, academic context by a small development team, the management strategy prioritizes clarity, documentation, and iterative progress over complex enterprise workflows. The adopted methodology is based on lightweight **Kanban principles** combined with **incremental milestones**, ensuring visibility, adaptability, and traceable progress throughout the system's lifecycle.

3.1 Roles and Responsibilities

Each role in the project team contributes to the stability, traceability, and quality of the predictive system. Responsibilities are distributed according to the principles of ISO 9001 (Quality Management) and CMMI Level 2 (Project Management and Defined Roles), ensuring clear accountability and controlled collaboration.

Table 2: Team Roles and Responsibilities

Team Member and Role	Responsibilities
Samuel Aljure Bernal — Analyst	Responsible for defining system requirements, analyzing data characteristics, and maintaining alignment between the analytical goals and system architecture. Ensures that functional and non-functional requirements are traceable across design artifacts, following ISO 9001 documentation control practices.
David Santiago Aldana González — Developer	Implements data processing and modeling modules using Python and R. Ensures code maintainability, version control consistency, and adherence to defined architecture. Contributes to modular integration and supports reproducibility verification according to CMMI Level 2 process maturity standards.
Carlos Alberto Barriga Gámez — Tester	Validates model performance, verifies preprocessing correctness, and executes reproducibility and fault-tolerance tests. Documents test cases, maintains verification logs, and confirms compliance with internal quality benchmarks. Applies ISO 9001 quality control and verification procedures to ensure system reliability.
Juan Diego Álvarez Cristancho — Manager	Coordinates task planning and execution following a Kanban-based workflow. Supervises progress tracking, milestone completion, and documentation updates. Ensures that the team adheres to CMMI project management practices, maintaining consistent reporting and stakeholder communication.

3.2 Project Workflow and Methodology

The project follows an **iterative Kanban workflow** designed for continuous improvement and traceability. Each task is tracked through the stages *To Do*, *In Progress*, *Under Review*, and *Completed*, allowing transparent management and feedback-driven progress.

- **Management Tool:** Trello or GitHub Projects for visual task management and traceability.
- **Version Control:** Git repositories with milestone tags ensuring reproducibility and controlled updates.
- **Collaboration:** Centralized documentation using Markdown and LaTeX, compiled into final PDF deliverables.

- **Review Frequency:** Weekly team reviews focused on progress validation, documentation quality, and consistency with ISO 9001 continuous improvement principles.

3.3 Milestones and Deliverables

The system development process is structured around four main milestones that represent the progressive evolution of the project from conceptual design to implementation and validation.

Table 3: Project Milestones and Deliverables

Milestone	**Description and Deliverables**	**Responsible Role**	**Estimated Date**
M1: System Analysis Completed	Delivery of the systemic analysis (Workshop #1), identifying system inputs, processes, and constraints.	**Analyst, manager**	Completed (Sep 2025)
M2: Architecture and Requirements Defined	Documentation of the refined system architecture, functional and non-functional requirements (Workshop #2).	**Analyst, manager, tester, developer**	Completed (Oct 2025)
M3: Robust Design and Risk Management	Integration of robust design principles, risk identification, and mitigation plan aligned with ISO 9001 and CMMI guidelines (Workshop #3).	**Analyst, manager**	Nov 2025
M4: System Implementation and Validation	Execution of the predictive workflow, model validation, and report generation for final submission (Workshop #4).	**Analyst, manager, tester, developer**	Nov 2025

3.4 Project Workflow Diagram

Figure 2 illustrates the overall workflow, showing the iterative nature of development, validation, and documentation. Each stage feeds back into the next through review and version control checkpoints, ensuring continuous improvement and traceability.

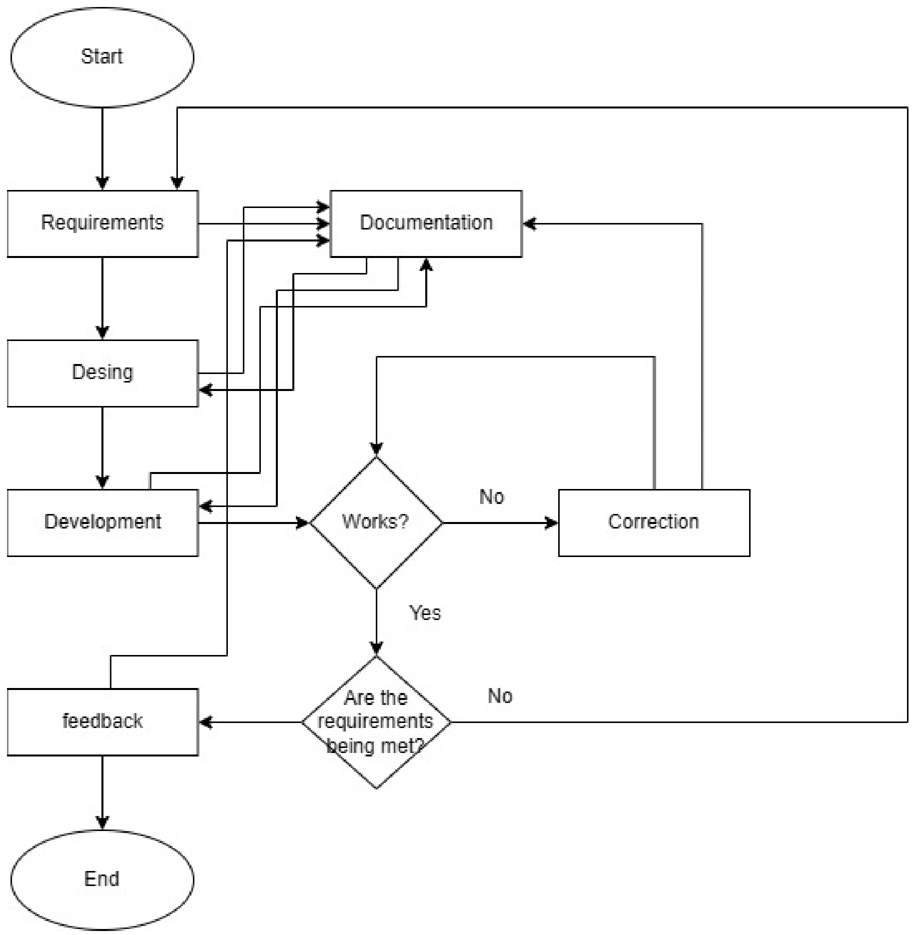

Figure 2: Project Workflow Overview – Chocolates 4U Predictive System

4 Incremental Improvements

The *Chocolates 4U Predictive System* has evolved through an incremental and iterative process across the three workshops. Each stage contributed to strengthening the system's conceptual foundation, modular design, and quality management practices. This section summarizes the main improvements and lessons learned that have guided the project from its initial systemic analysis toward a robust and traceable architecture ready for implementation.

4.1 Progress from Workshop #1: Systemic Analysis

In the first stage, the project focused on understanding the predictive problem through the lens of systems engineering. The primary outcome was a complete **systemic analysis** identifying inputs, processes, outputs, and constraints, as well as sources of instability such as data variability, algorithmic sensitivity, and human decision factors.

- The system was defined as a complex entity composed of interacting subsystems.
- Key factors of chaos and sensitivity were identified, demonstrating the need for reproducibility and controlled experimentation.
- The analysis established the theoretical foundation for later architectural modularization.

4.2 Progress from Workshop #2: System Design and Requirements

The second stage formalized the system's structure through an **architectural design** and a detailed set of **functional and non-functional requirements**. The focus shifted from theoretical understanding to practical design and traceability.

- Developed a modular architecture divided into six main modules (data ingestion, preprocessing, analysis, modeling, validation, reporting).
- Defined system requirements using user stories and formal specifications.
- Introduced version control, validation logic, and the concept of controlled feedback loops.
- Established reproducibility and scalability as core design objectives.

4.3 Advancements in Workshop #3: Robust Design and Management Integration

The current workshop integrates robust design principles and project management strategies, transforming the system into a more mature and sustainable framework.

- Incorporated reliability, fault tolerance, and maintainability into the architecture.

- Defined quality and risk management mechanisms adapted to a local execution context, ensuring stability, reproducibility, and data integrity.
- Established a project management plan based on Kanban principles, defining clear roles, milestones, and traceable deliverables.
- Linked the design with recognized standards (ISO 9001, CMMI, Six Sigma) to ensure compliance with engineering best practices.

4.4 Overall Evolution and Lessons Learned

Over the three workshops, the project transitioned from conceptual analysis to a robust and structured system design. This evolution reflects not only technical improvements but also advancements in process control and systemic understanding.

1. Early conceptual models evolved into a well-defined modular architecture with clear interfaces and control mechanisms.
2. Mechanisms such as validation, fault isolation, and versioning mitigated the sources of chaos identified in Workshop #1.
3. Each component is now designed to interact coherently within the larger architecture, promoting systemic integrity.
4. The integration of project management tools and quality standards ensures sustainability and traceability as the project moves toward implementation.

References

[1] Fonseca, Luis Domingues, Pedro. (2017). ISO 9001:2015 edition- management, quality and value. International Journal for Quality Research. 11. 149-158. 10.18421/IJQR11.01-09.

[2] F. S. Silva, F. S. F. Soares, A. L. Peres, I. M. de Azevedo, A. P. L. F. Vasconcelos, F. K. Kamei, and S. R. de L. Meira, "Using CMMI together with agile software development: A systematic review," Information and Software Technology, vol. 58, pp. 20–43, 2015, doi: 10.1016/j.infsof.2014.09.012.

[3] Zare Mehrjerdi Y (2011), "Six-Sigma: methodology, tools and its future". Assembly Automation, Vol. 31 No. 1 pp. 79–88, doi: https://doi.org/10.1108/01445151111104209