

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA
Ingeniería de Sistemas

Otto Group Product Classification Challenge, A Robust System Design and Planning

Systems Analysis & Design

Group 020-82

Workshop #3

Students

Juan Diego Lozada 20222020014
Juan Pablo Mosquera 20221020026
María Alejandra Ortiz Sánchez 20242020223
Jeison Cuenca: 20242020043

Professor: Carlos Andrés Sierra Virguez

Date: November 2025

Contents

1	Review and Refine System Architecture	3
1.1	Overview	3
1.2	Updated Architectural Design	3
1.3	Component Description and System Quality Support	3
1.4	Design Standards and Quality Frameworks Referenced	4
2	Quality and Risk Analysis	5
2.1	Identification of Potential Risks	5
2.2	Monitoring and Response Plan	5
3	Project Management Plan	6
3.1	Team Roles and Responsibilities	6
3.2	Milestones and Deliverables	6
3.3	Benefits of the Kanban Approach	7
3.4	Project Timeline and Workflow	7
3.4.1	Overview	7
3.4.2	Timeline Summary	8
3.4.3	Iterative Workflow	8
3.4.4	Continuous Feedback and Improvement	8
4	Incremental Improvements	9
4.1	Lessons Learned from Workshops 1 and 2	9
4.2	Evolution of System Design and Management Plan.	10
5	Conclusions	12
6	References	13

1 Review and Refine System Architecture

1.1 Overview

Based on the outcomes of Workshops #1 and #2, the system architecture for the Otto Group Product Classification Challenge has been refined to ensure greater robustness, scalability, and maintainability. The goal is to evolve the initial modular design into a resilient structure that supports continuous operation under varying conditions and aligns with recognized quality and engineering standards such as ISO 9000, CMMI, and Six Sigma.

1.2 Updated Architectural Design

The architecture incorporates robust design principles, including modularity, fault-tolerance, and scalability. Each component is explicitly labeled and structured to contribute to system quality and reliability. Figure 1 presents the updated architecture diagram.

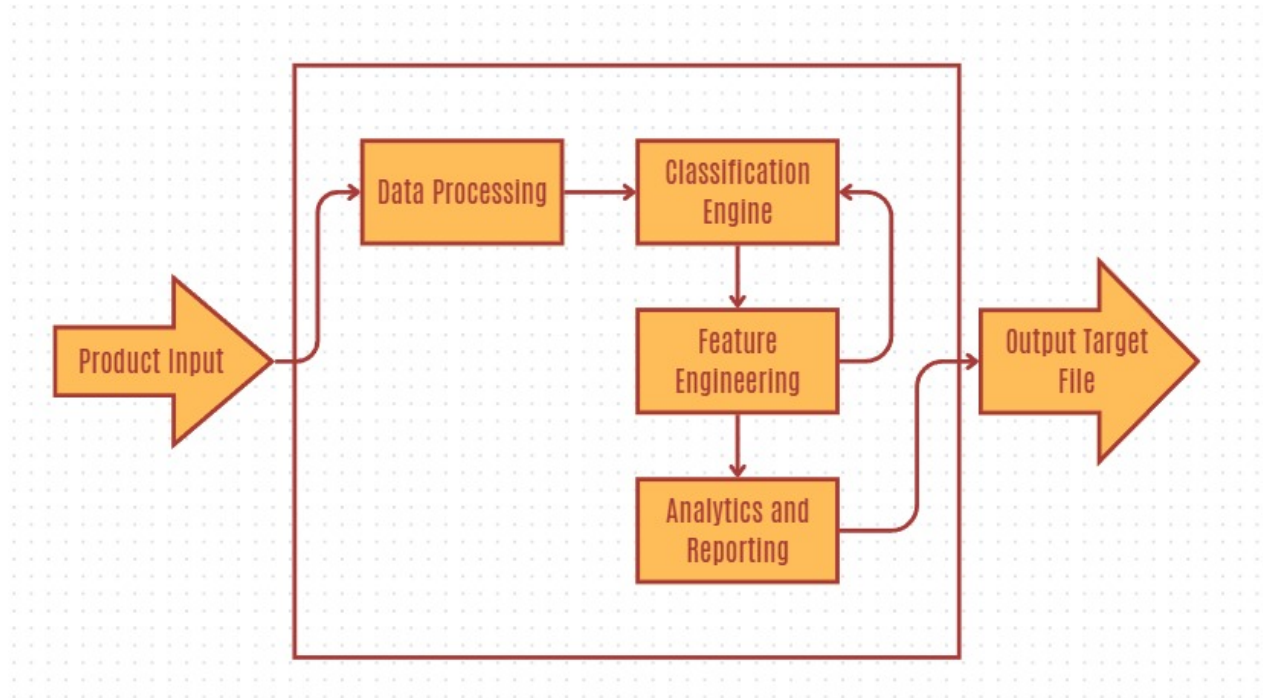


Figure 1: High-Level Architecture

1.3 Component Description and System Quality Support

- **Data Processing:** Cleans, normalizes, and validates product data. Built with validation scripts that ensure consistency and prevent data corruption.
- **Feature Engineering:** Extracts meaningful representations from obfuscated variables. Fault-tolerant pipelines allow automated retries in case of processing errors.

- **Classification Engine:** Implements machine learning models for product categorization. Supports scalability through modular model loading and parallel training.
- **Analytics and Reporting:** Monitors model performance, system metrics, and error rates. Provides visual dashboards that support feedback and iterative improvement.
- **Monitoring and Logging Layer:** Tracks performance indicators, logs all runtime errors, and reports anomalies for diagnostic purposes.

1.4 Design Standards and Quality Frameworks Referenced

- **ISO 9000:** Ensures that data handling, pre-processing, and validation procedures meet standardized quality assurance processes.
- **CMMI:** Provides process maturity guidelines, promoting consistency and repeatability in model training and validation.
- **Six Sigma:** Applies DMAIC principles (Define, Measure, Analyze, Improve, Control) to continuously improve accuracy, reliability, and system performance.

2 Quality and Risk Analysis

This section outlines the primary risks that may affect the stability, performance, or integrity of the system during its development and operation.

2.1 Identification of Potential Risks

Risk	Description	Mitigation Strategy	Standard
Data Loss or Corruption	Incomplete or damaged datasets due to transfer errors or pre-processing issues.	Use Git-based version control, automated data validation, and redundant backups.	ISO 9000
Model Overfitting	Model performs well on training data but poorly on unseen data.	Apply k-fold cross-validation, early stopping, and probability calibration.	CMMI
Pipeline Downtime	Failure during long training or processing sessions.	Implement error recovery, automatic re-execution scripts, and alert systems.	Six Sigma
Security Breach	Unauthorized access or manipulation of sensitive data.	Control repository permissions, apply encryption, and use credential management tools.	ISO 27001
Performance Degradation	Delays caused by growing dataset or resource limits.	Use batch processing and parallel computation for scalability.	ISO 9000

Table 1: Risk assessment and mitigation strategies.

2.2 Monitoring and Response Plan

To ensure continuous system reliability, a risk monitoring plan has been integrated into the project’s workflow and version control structure.

- **Testing and Validation Cycles:** The tester role performs periodic validation through predefined test cases covering data integrity, model stability, and execution time.
- **Version Control Integration:** Using Git and GitHub, all changes in code or data are tracked, allowing rollback and comparison between stable and experimental versions to minimize data loss or corruption.
- **Response Protocols:** If a risk materializes (e.g., downtime or data corruption), the project manager coordinates an immediate rollback to the last stable version and initiates corrective procedures

3 Project Management Plan

The project management plan defines the structure, responsibilities, and methodologies guiding the development of the Otto Group Product Classification System. It establishes the framework for coordination, communication, and incremental progress throughout the course timeline.

3.1 Team Roles and Responsibilities

The project is developed by a multidisciplinary team of four members, each assuming a specific role aligned with their skills and the project’s systemic objectives. Table 3.1 summarizes the assigned responsibilities.

Role	Member	Main Responsibilities
Project Manager	Jeison Cuenca	Oversees project progress, manages deadlines, and ensures communication between team members.
Analyst	Juan Diego Lozada	Defines requirements, analyzes data, ensures consistency, and maintains system documentation
Developer	Juan Pablo Mosquera	Implements and maintains code for preprocessing, simulations, and model training.
Tester / Quality Engineer	María Alejandra Ortiz	Validates functionality, tests system outputs, and monitors risk mitigation strategies.

Table 2: Team Roles and Responsibilities

3.2 Milestones and Deliverables

The project development follows a sequence of milestones that align with the course workshops and the incremental evolution of the system. Each milestone consolidates the progress of the analysis, design, simulation, and reporting of the system.

- **Workshop 1 – System Analysis:** Identification of requirements, data constraints, and chaos-related sensitivities in the Otto Group data set.
- **Workshop 2 – System Design:** Definition of modular architecture, technical stack, and systemic relationships between core components.
- **Workshop 3 – Incremental Improvements:** Refinement of system design, adoption of Kanban methodology, and implementation of quality and risk management frameworks.

- **Workshop 4 – System Simulation:** Development of data-driven and event-based simulations to validate system behavior, performance stability, and sensitivity under varying conditions.
- **Final Project Delivery:** Integration of all workshops into a comprehensive technical report and functional prototype demonstrating full system workflow, stability, and feedback mechanisms.

3.3 Benefits of the Kanban Approach

We chose the Kanban methodology because it is practical and makes it easy to see our progress. It also helps us avoid bottlenecks, since it shows how many tasks are in progress in a very visual way. This makes it easier to spot when something is slowing down and to keep the workflow running smoothly.

- Enhanced visibility and accountability of tasks.
- Steady and continuous project flow.
- Real-time adaptability to risks or feedback.
- Balanced workload between team members.



Figure 2: Example Kanban Board

3.4 Project Timeline and Workflow

3.4.1 Overview

This section outlines a feasible project timeline for the *Otto Group Product Classification System*, following software engineering standards within the **System Design** phase. The schedule integrates the iterative and adaptive nature of the **Kanban methodology**, emphasizing continuous feedback, quality assurance, and flexibility to handle unforeseen events.

3.4.2 Timeline Summary

Table 3: Simplified Project Timeline

Phase	Weeks	Main Goal	Key Deliverables
System Analysis	1–2	Identify requirements, data constraints, and systemic relationships.	Analysis report and system map.
System Design	3–5	Define modular architecture and control mechanisms.	Architecture document and design diagram.
Risk & Quality Control	6–7	Integrate ISO/CMMI standards and Kanban workflow.	Risk plan and active Kanban board.
Simulation & Testing	8–10	Validate model behavior and performance stability.	Simulation report and test metrics.
Integration & Delivery	11–12	Consolidate all modules into a final system prototype.	Final report and functional prototype.

3.4.3 Iterative Workflow

The Kanban process ensures visibility and flexibility through stages: **To Do** → **In Progress** → **Review / Testing** → **Done**. Each week includes feedback sessions, validation cycles, and small task refinements. 3

3.4.4 Continuous Feedback and Improvement

System components evolve through iterative control loops linking development, testing, and analytics. When anomalies or performance drops occur, new Kanban tasks are created immediately, ensuring adaptability and ongoing quality assurance. 4

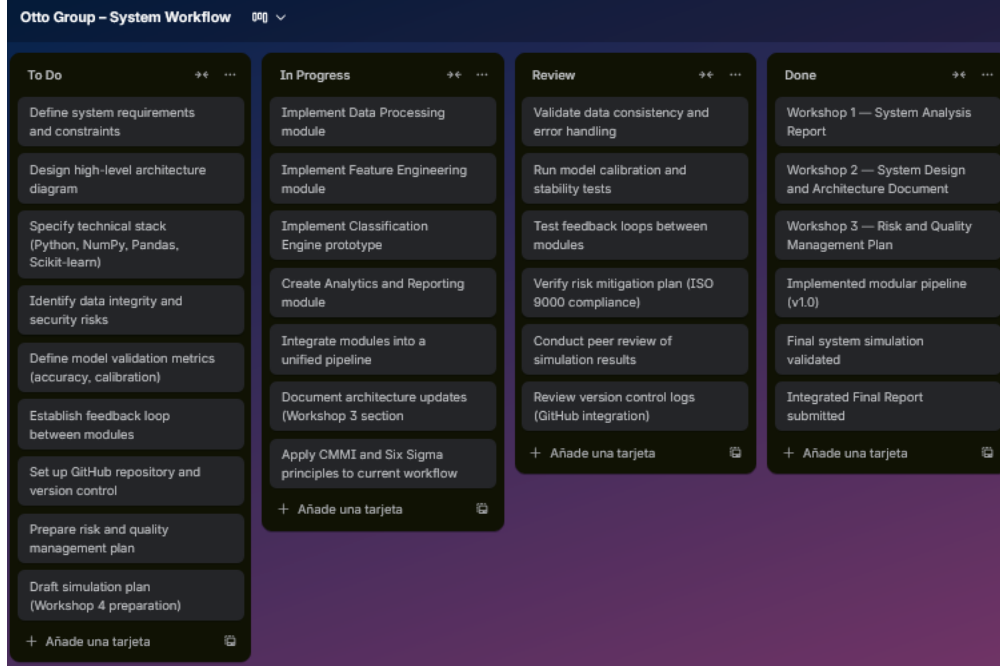


Figure 3: Kanban workflow for the Otto Group project.

4 Incremental Improvements

This stage of this project focuses on refining the system architecture and project management plan based on insights gathered during Workshops 1 and 2. The iterative approach adopted in this process follows the principles of systems analysis

4.1 Lessons Learned from Workshops 1 and 2

The initial workshops provided valuable understanding of both the technical and organizational aspects of system development. Workshop 1 established the analytical foundation by exploring the Otto Group dataset, identifying data structures, and defining the problem boundaries.

Workshop 2 expanded this foundation through the design of a modular system architecture. The focus shifted from isolated data exploration to systemic interaction between subsystems, including data processing, feature engineering, classification, and feedback control.

From these stages, three key lessons emerged:

- **Data-driven systems require structured design.** A clear modular framework ensures that analytical components like data, features, models and remain interoperable and testable.
- **Feedback mechanisms are essential for stability.** Incorporating feedback loops prevents uncontrolled variance and allows adaptation to new data or performance fluctuations.

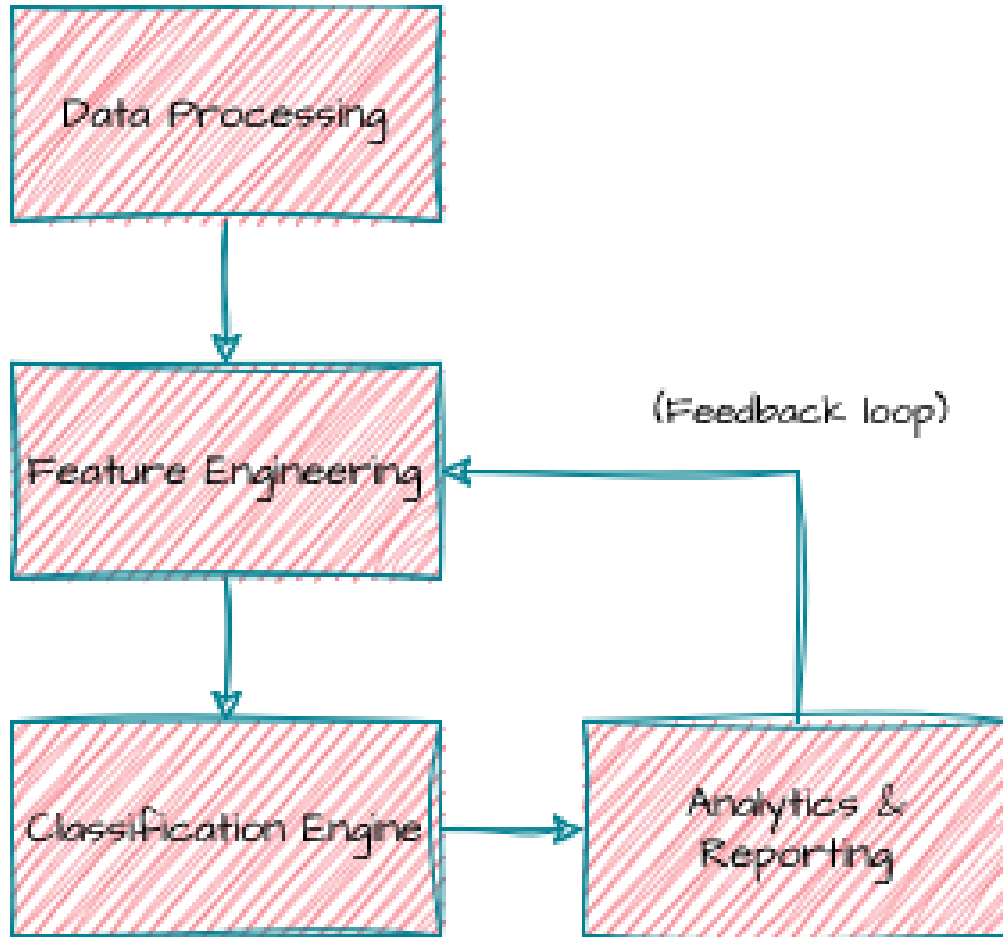


Figure 4: Feedback loop

- **System documentation and control are critical.** Effective configuration management, naming conventions, and repository organization improve team coordination and system maintainability.

4.2 Evolution of System Design and Management Plan.

Kanban-Based Management Approach. The project team adopted the Kanban methodology due to its practical and visual nature, which facilitates the monitoring of workflow progress and identification of potential bottlenecks. The system development process was mapped into stages such as *To Do*, *In Progress*, *Review*, and *Completed*. This approach improves collaboration and ensures continuous project flow.

This project management framework directly aligns with systems analysis principles—particularly the notions of feedback and continuous improvement—by making system evolution observable and controllable in real time.

Risk and Quality Management. In parallel, a risk mitigation and quality control framework was developed to ensure system reliability and data integrity. The key risks identified

include data loss, model overfitting, pipeline failures, security breaches, and performance degradation. Each risk was paired with a mitigation strategy and a corresponding industry standard, as summarized in Table 1.

Systematic and Technical Enhancements. In addition to methodological and management improvements, several systematic refinements were introduced to strengthen the project’s technical coherence and support the transition toward partial implementation.

First, the system architecture was refined to achieve higher modular independence. Each subsystem *Data Processing*, *Feature Engineering*, *Classification Engine*, and *Analytics and Reporting* was clearly defined with input output interfaces. This modular separation allows each component to be tested, replaced, or scaled independently, facilitating maintainability and reducing interdependency risks.

The team also established a version control strategy using `Git` and GitHub repositories to maintain code history, manage changes collaboratively, and ensure rollback capabilities in case of errors. This also aligns with the quality management framework described previously, promoting reproducibility and documentation of system evolution.

5 Conclusions

1. The refinement of the Otto Group Product Classification System demonstrated the importance of integrating robust design principles such as modularity, scalability, and tolerance. Referencing international quality frameworks such as ISO 9000, CMMI, and Six Sigma, the project established a solid base for sustainable development and continuous improvement.
2. Incorporating quality and risk management processes strengthened the reliability of the system, ensure proactive control of potential failures. The identification of risks, along with clearly defined mitigation strategies and monitoring procedures, contributed to the building of a reliable and adaptive classification environment.
3. The implementation of Kanban methodology improved team coordination and workflow visibility, enabling a dynamic and iterative project management process. This approach encouraged collaboration between analyst, developer, tester and manager roles, ensuring continuous feedback, timely deliverables, and alignment of technical progress with the project's strategic objectives, avoiding bottlenecks.

6 References

References

- [1] Otto Group. (2015). *Otto Group Product Classification Challenge*. Kaggle. Recuperado de <https://www.kaggle.com/competitions/otto-group-product-classification-challenge/>
- [2] Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. New York: McGraw-Hill.
- [3] Forrester, J. W. (1968). *Principles of systems*. Portland, OR: Productivity Press.
- [4] Gleick, J. (1987). *Chaos: Making a new science*. New York: Viking.
- [5] Meadows, D. H. (2008). *Thinking in systems: A primer*. White River Junction, VT: Chelsea Green Publishing.
- [6] Senge, P. M. (1990). *The fifth discipline: The art and practice of the learning organization*. New York: Doubleday.
- [7] Sterman, J. D. (2002). All models are wrong: Reflections on becoming a systems scientist. *System Dynamics Review*, 18(4), 501–531.
- [8] Holland, J. H. (1998). *Emergence: From chaos to order*. Oxford: Oxford University Press.
- [9] Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.
- [10] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., et al. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- [11] McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 56–61. <https://doi.org/10.25080/Majora-92bf1922-00a>
- [12] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [13] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- [14] Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.
- [15] Checkland, P. (1999). *Systems thinking, systems practice: Includes a 30-year retrospective*. Chichester: Wiley.

- [16] Jackson, M. C. (2003). *Systems thinking: Creative holism for managers*. Chichester: Wiley.
- [17] Beer, S. (1979). *The heart of enterprise*. Chichester: Wiley.
- [18] Von Bertalanffy, L. (1968). *General system theory: Foundations, development, applications*. New York: George Braziller.