UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Universidad Distrital Francisco José de Caldas

Computer Engineering Program
School of Engineering

Workshop No. 3 – Kaggle Project Analysis: CIBMTR Equity in Post-HCT Survival Predictions

Sergio Nicolás Mendivelso Martínez – 20231020227
Sergio Leonardo Moreno Granado – 20242020091
Juan Manuel Otálora Hernández – 20242020018
Juan Diego Moreno Ramos – 20242020009

Professor: Eng. Carlos Andrés Sierra, M.Sc.

A report submitted for Workshop No. 3 in Systems Analysis & Design
Semester 2025-III

November 2025, Bogotá D.C.

Contents

List of Figures

Chapter 1

Robust System Design and Project Management

1.1 Incremental Improvements and Lessons Learned

Workshop 1 identified core challenges: high sensitivity to input variables, chaotic and unpredictable behavior, and critical equity requirements Harrington et al. (2025); Kaggle and CIBMTR (2025). Workshop 2 designed a 7-module architectural blueprint. Workshop 3 focuses on making our quality control practical and ensuring fair results.

1.1.1 Evolution Across Workshops

Workshop 1 → Workshop 2:

- *Challenge:* Post-HCT outcomes exhibit chaos where small input variations produce large output changes Kucab et al. (2024).
- *Response:* Designed Uncertainty Quantification Module (M6) with confidence intervals.

Workshop 2 → Workshop 3:

- *Gap:* Had architecture but lacked operational controls.
- *Response:* Added monitoring framework (Table 1.2) with thresholds and protocols CMMI Institute (2018); International Organization for Standardization (2015); Pyzdek and Keller (2018).

Pattern: Workshop 1 (Problem Identification) → Workshop 2 (Solution Architecture) → Workshop 3 (Operational Quality Control).

1.2 Review and Refine System Architecture

The 7-module pipeline from Workshop 2 remains unchanged Eisenstadt, Wagner, Biessmann, and Ustyuzhanin (2020). This design is robust by nature. Here we explain how it handles reliability, scalability, and maintainability.

1.2.1 Updated Architecture Diagram (Robust Principles)

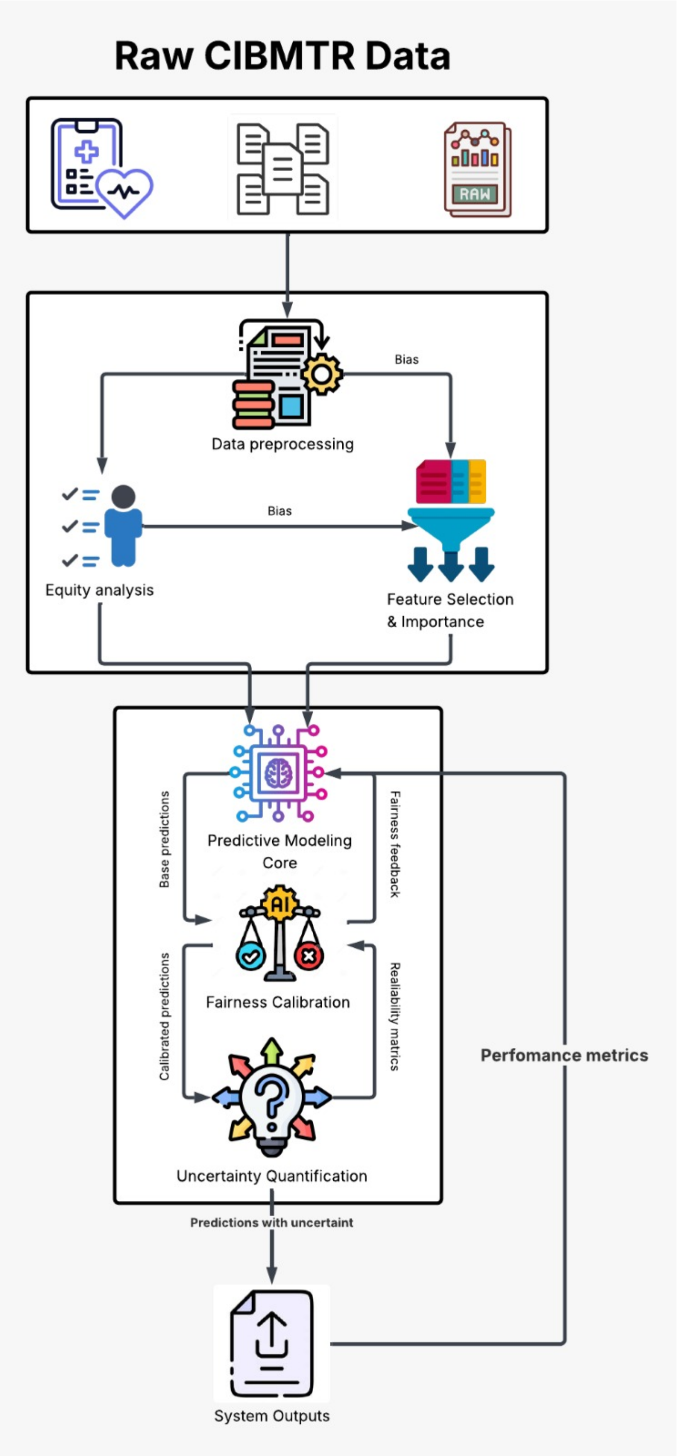

Figure 1.1: System architecture: M1 (Preprocessing), M2 (Equity Analysis), M3 (Feature Selection), M4 (Modeling), M5 (Fairness Calibration), M6 (Uncertainty Quantification), M7 (Outputs).

This architecture supports robust design through four key principles:

- **Modularity:** M1-M7 are discrete components with versioned interfaces. Updating

M5 doesn't touch M4 DataCamp (2023b).

- **Scalability:** M4 scales independently via parallel CV folds without extra resources for M3 Scikit-learn (2023).
- **Maintainability:** Configuration separated from code; MLflow tracks experiments.
- **Fault-Tolerance:** M1 triggers fallback imputation when missingness $> 30\%$; errors don't cascade.

1.2.2 Component Support for System Quality and Reliability

Each component addresses Workshop 1 findings (chaos, sensitivity, equity) Harrington et al. (2025); Kaggle and CIBMTR (2025):

M1: Data Preprocessing

- *Quality:* Equity-aware imputation prevents differential missingness GeeksforGeeks (2023b); Pushkarna (2023).
- *Reliability:* Fallback strategies ensure continuity.

M2: Equity Analysis

- *Quality:* QA component performing stratified analysis and bias detection Microsoft (2023); Trusted AI Team (2023).

M3: Feature Selection

- *Quality:* Validates feature availability across demographics Doherty, Char, Goodman, Shah, and Oberst (2024).
- *Reliability:* Combines domain/statistical/ML strategies to reduce bias.

M4: Predictive Modeling Core

- *Reliability:* Ensembles (Cox GeeksforGeeks (2023a), RSF Pölsterl (2023b), GBM Neptune.ai (2023)) robust against chaotic behavior.
- *Quality:* Demographic-stratified CV ensures consistent evaluation Scikit-learn (2023).

M5: Fairness Calibration

- *Quality:* Primary QC mechanism enforcing stratified C-index standard Dosi (2023); Shelf.io (2023).

M6: Uncertainty Quantification

- *Reliability & Usability:* Manages "chaos" and "sensitivity" via confidence intervals Pölsterl (2023a).

M7: System Outputs

- *Usability:* SHAP interpretability builds clinical trust Auletta, Kou, Chen, and Shaw (2020); DataCamp (2023a); SHAP (2023).

1.2.3 Alignment with Quality Standards

ISO 9000: Customer focus via M7 interpretability; continuous improvement via M2-M5 feedback loop International Organization for Standardization (2015).

CMMI Level 3: 7-module pipeline is a defined process with documented interfaces; MLflow maintains process assets CMMI Institute (2018).

Six Sigma: "Defect" = inequity (disparity > 0.10); M5 acts as Control mechanism Kaggle and CIBMTR (2025); Pyzdek and Keller (2018).

Standard	Principle	Evidence
ISO 9000	Customer focus; improvement	M7 SHAP for trust; M2-M5 loop
CMMI (L3)	Defined processes	Module interfaces; stratified CV; MLflow
Six Sigma	Defect reduction	Disparity > 0.10 = defect; M5 control

Table 1.1: Standards mapped to architectural components.

1.3 Quality and Risk Analysis

1.3.1 Potential Risks and Failure Points

We identified six main risks that could affect system performance, fairness, or our ability to deliver on time:

Risk 1: Emergent Behaviors and Prediction Instability

Non-linearities in post-HCT outcomes (immune responses, complications) cause unpredictable model outputs Harrington et al. (2025); Kucab et al. (2024). Small input changes (e.g., 1-year age difference, minor HLA mismatch) may produce disproportionately large prediction variations. Complex variable interactions create "tipping points" where model behavior becomes unstable. The chaotic nature means even deterministic models produce seemingly random results for edge cases.

Risk 2: Feature Selection Bias (M3)

Biased feature selection from indices that don't maintain equity Doherty et al. (2024). Statistical methods may prioritize features more complete for majority demographics. ML-based importance metrics may encode spurious correlations with sensitive attributes (race/ethnicity). Clinical domain knowledge may reflect historical biases. Features may be systematically missing, measured differently, or less accurate across demographic groups.

Risk 3: Input Data Quality ("Garbage In, Garbage Out")

System performance depends critically on data quality CIBMTR (2025). Non-random missingness patterns correlate with demographics (e.g., certain tests less frequently performed for specific populations). Measurement inconsistencies exist across transplant centers and time periods. Data entry errors or outliers can corrupt model training. Temporal quality variations mean earlier records may be less complete than recent ones.

Risk 4: Data Quality Worse Than Expected

Dataset may have more severe quality issues than initial exploration suggests. Missing values may exceed 30% threshold for critical features. Data documentation may be incomplete or inconsistent. Preprocessing complexity may require more time than Week 2 allocation.

Risk 5: Model Accuracy Lower Than Goal

Initial models may not achieve target C-index > 0.70. Fairness constraints may reduce overall accuracy. Ensemble methods may not provide expected performance gains. Hyperparameter tuning may be insufficient within Week 3 timeframe.

Risk 6: Project Schedule and Team Availability

Team member may become unavailable due to illness or other commitments. Tasks may take longer than estimated (especially ML experimentation). Dependencies between tasks may cause cascading delays. Milestone deadlines (Nov 14, 21, 28, Dec 5) may be at risk if issues compound.

1.3.2 Mitigation Strategies

For Risk 1: Emergent Behaviors

- **No Full Automation:** System serves as decision support only; physician must validate all predictions Auletta et al. (2020).
- **Continuous Monitoring:** Compare predictions against actual outcomes as follow-up data becomes available; retrain or recalibrate upon drift detection DataCamp (2023b).
- **Sensitivity Analysis:** Vary multiple input parameters simultaneously to identify unstable regions; document operating envelopes; flag out-of-envelope predictions.
- **Ensemble Diversity:** Use multiple model types (Cox, RSF, Gradient Boosting) in M4 so erratic individual behavior averages out Neptune.ai (2023).

For Risk 2: Feature Selection Bias

- **Fairness as Explicit Objective:** Make equity metrics (demographic parity, equalized odds) first-class objectives alongside accuracy Microsoft (2023); Shelf.io (2023).
- **Fairness-Aware Preprocessing:** Apply reweighing, disparate impact removal, or learning fair representations to eliminate sensitive attribute dependencies Trusted AI Team (2023).
- **Multiple Selection Strategies:** Cross-check domain knowledge, statistical tests, and ML importance; require consensus from at least two methods before including features.
- **Availability Verification:** Perform demographic-stratified completeness analysis; reject features with $> 20\%$ availability disparity across groups.

For Risk 3: Input Data Quality

- **Robust Imputation:** Use MICE/KNN with demographic stratification to prevent amplifying group differences GeeksforGeeks (2023b). Include missingness indicators as features to capture pattern information.
- **Train with Noise:** Inject realistic errors (out-of-range values, empty fields) during training so model handles imperfect real-world inputs gracefully.
- **Metadata Logging:** Track data origin, modifications, and cleaning processes for auditability Yuan et al. (2024). Enable tracing predictions to source transformations.
- **Exhaustive Documentation:** Document collection methodology, population coverage, known limitations, preprocessing steps CIBMTR (2025).

For Risk 4: Data Quality Worse Than Expected

- **Allocate Extra Time in Week 2:** If quality issues discovered, prioritize data cleaning and extend Week 2 by 2-3 days if needed.
- **Request Professor Guidance:** Consult during Friday review session for advice on handling severe quality issues.
- **Adjust Timeline:** If data preparation extends into Week 3, compress modeling tasks by using simpler baseline models first.
- **Team Reallocation:** Juan Diego Moreno (Quality Specialist) assists Juan Otalora (Data Analyst) with cleaning if workload exceeds 20 hours/week.

For Risk 5: Model Accuracy Lower Than Goal

- **Iterative Approach in Week 3:** Start with simple baselines (Cox), progressively add complexity (RSF, GBM); identify which techniques provide gains.
- **Professor Feedback:** Present intermediate results during Week 3 Friday review; get suggestions for alternative approaches.
- **Focus on Key Improvements:** If time-constrained, prioritize most impactful optimizations (feature engineering, calibration) over exhaustive hyperparameter search.
- **Realistic Goal Adjustment:** If C-index > 0.70 proves unattainable, document why and aim for best achievable performance with fairness maintained.

For Risk 6: Schedule and Team Availability

- **Work Redistribution:** If Sergio Mendivelso (Programmer, 20-25h/week) unavailable, Juan Otalora assists with code; Sergio Moreno takes on integration tasks.
- **Task Prioritization:** Focus on critical path items first (data cleaning → feature selection → baseline models → predictions). Defer nice-to-have features (advanced visualizations, extensive documentation polish).

- **Buffer in Week 4:** Week 4 includes documentation and final checks; if behind, compress documentation and prioritize working predictions.csv and technical report.
- **Early Escalation:** Sergio Moreno (Project Coordinator) identifies blockers in Monday meetings and escalates to professor immediately rather than waiting until Friday.

1.3.3 Monitoring and Response

Risk Category	Monitoring Metric	Alert Threshold	Response Protocol
Emergent instability	Prediction std. dev. for similar patient profiles	> 0.15	Run sensitivity analysis; simplify or regularize model; clinical expert review
Selection bias (M3)	Max C-index gap across demographic groups	> 0.10	Apply reweighing Trusted AI Team (2023); revise feature set; recalibrate M5 thresholds
Data quality	Percentage missingness in critical features	$> 30\%$	Escalate to advanced imputation; flag high-uncertainty predictions; consider excluding feature
Model drift	C-index drop vs. validation baseline	$> 5\%$	Retrain on recent data; refresh M5 calibration; investigate population changes
Schedule delay	Tasks behind planned completion	> 2 days	Team meeting to redistribute work; prioritize critical path; escalate to professor
Milestone risk	Success criteria not met	Any red flag	Block progression; focus team on resolving issues; adjust scope if necessary

Table 1.2: Risk monitoring framework with metrics, thresholds, and response protocols.

During Development (Weeks 1-4):

We'll maintain a comprehensive log of project implementation using Git version control and MLflow experiment tracking. In the event of an error, this logging helps us quickly identify the source and find a solution.

We'll keep high-level team communication to effectively share ideas, corrections, and errors, keeping everyone informed through:

- **Automated CI/CD:** Builds fail when Table 1.2 thresholds violated DataCamp (2023b); prevents merging problematic code.
- **Weekly Monday Meetings (15 min):** Review completed work, current plans, blockers; Project Coordinator escalates critical alerts.
- **Milestone Quality Gates:** All success criteria must be met before progressing to next week; QA Specialist has authority to block if unmet.
- **Daily Slack/WhatsApp Updates:** Quick questions and progress sharing maintain continuous visibility.

- **Friday Professor Reviews (optional):** Opportunity to get guidance and validate approach before weekend work.

During Operation (if deployed post-competition):

- MLflow logs every prediction with timestamp, input features, output probabilities, uncertainty scores, model version.
- Monthly automated equity audits calculate stratified C-index on all recent predictions Shelf.io (2023).
- Outcome-based recalibration: as actual patient outcomes become available, compare against predictions; if drift detected (C-index drop $> 5\%$), trigger retraining and M5 calibration refresh.
- Critical alerts (severe fairness violations, system crashes) trigger immediate investigation; system can be temporarily disabled if patient harm risk identified.

1.4 Project Management Plan

1.4.1 Team Roles and Responsibilities

Our 4-member team has clearly defined roles aligned with the 7-module architecture:

Sergio Mendivelso - Programmer

- *Main Tasks:* Write and test code for data processing (M1); build prediction pipeline (M4); integrate system components (M1-M7); manage GitHub repository; debug and fix technical issues.
- *Weekly Hours:* 20-25 hours
- *Key Deliverables:* Working code for all modules, tested functions, final predictions (predictions.csv)

Sergio Moreno - Project Coordinator

- *Main Tasks:* Organize team meetings and communication; track project progress against timeline; report status to professor; manage deadlines and milestones; identify and solve blockers.
- *Weekly Hours:* 5-8 hours
- *Key Deliverables:* Status reports, meeting notes, timeline updates, final submission coordination

Juan Otalora - Data Analyst

- *Main Tasks:* Explore and understand dataset; clean and prepare data (M1); analyze fairness and equity issues (M2); select important features (M3); validate results.
- *Weekly Hours:* 15-20 hours
- *Key Deliverables:* Clean dataset, fairness analysis reports, feature list, validation results

Juan Diego Moreno - Quality Specialist

- *Main Tasks:* Assist with testing and validation; help with data analysis; support code documentation; quality assurance checks; general project support.
- *Weekly Hours:* 10-15 hours
- *Key Deliverables:* Test reports, documentation, quality checks, final QA report

1.4.2 Project Management Methodology

For this 4-week implementation sprint (November 8 - December 5, 2025), we use a milestone-driven approach with weekly synchronization meetings.

Communication Protocol:

- *Weekly Meeting:* Every Monday at 9:00 AM (15 minutes) covering: (1) completed work, (2) current week plans, (3) blockers
- *Daily Updates:* Slack/WhatsApp for quick questions; GitHub for code updates; shared documents for progress
- *Optional Review:* Friday check-in with professor for guidance

Development Tools:

- *GitHub:* Code storage, version control, collaboration
- *Python:* Primary programming language
- *Data Format:* CSV files for datasets and predictions
- *Communication:* Slack/WhatsApp for coordination; Google Docs for shared notes

1.4.3 Project Timeline and Milestones

The project spans 4 weeks with clearly defined milestones:

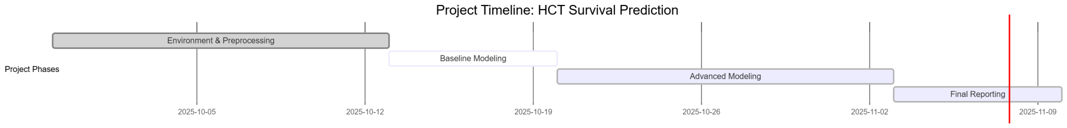

Figure 1.2: 4-week sprint timeline showing milestones, deliverables, and team responsibilities from November 8 to December 5, 2025.

Week 1: Setup and Planning (Nov 8-14) - Milestone M1: Kickoff Complete

- *Goals:* Plan project scope; prepare dataset; setup development environment; establish workflows
- *Deliverables:*
  - Project Charter (1 page) - Sergio Moreno
  - Data Overview Report (2 pages) - Juan Otalora
  - GitHub setup with README - Sergio Mendivelso

  - File structure outline - Juan Diego Moreno

- *Success Criteria:* Team understands goals; dataset accessible; environment works for all; repository organized

Week 2: Data Analysis and Design (Nov 15-21) - Milestone M2: Data Ready

- *Goals:* Deep dataset understanding; identify data quality issues; plan architecture; select features
- *Deliverables:*
  - Cleaned Dataset (CSV) - Juan Otalora
  - Fairness Analysis Report (2-3 pages) - Juan Otalora
  - Data processing functions with tests - Sergio Mendivelso
  - Feature selection report - Juan Otalora
  - Professor approval - Sergio Moreno
- *Success Criteria:* Data clean and ready; missing values handled; fairness issues identified; features selected and validated

Week 3: System Building (Nov 22-28) - Milestone M3: Models Working

- *Goals:* Build prediction models; test accuracy; ensure fairness across groups; validate results
- *Deliverables:*
  - Working model code - Sergio Mendivelso
  - Accuracy and fairness results (CSV) - Juan Otalora
  - Fairness validation report - Juan Otalora
  - Optimized and debugged code - Sergio Mendivelso
  - Test report - Juan Diego Moreno
- *Success Criteria:* Models trained and working; accuracy acceptable; fairness checked across groups; code clean and documented; all tests pass

Week 4: Documentation and Delivery (Nov 29 - Dec 5) - Milestone M4: Project Delivered

- *Goals:* Prepare final predictions; write documentation; create presentation; submit deliverables
- *Deliverables:*
  - Final predictions.csv - Sergio Mendivelso
  - Code documentation - Sergio Mendivelso
  - Technical Report (8-10 pages) - Juan Otalora
  - Presentation slides (10-15) - Sergio Moreno
  - Quality assurance report - Juan Diego Moreno
  - Final submission package - Sergio Moreno
- *Success Criteria:* Predictions formatted correctly; all code documented; technical report complete; presentation ready; files organized on GitHub; submitted on time (Dec 5, 2025)

1.4.4 GitHub Repository Structure

Our repository follows this organization:

```
HCT-Survival-Prediction/
|-- data/
|   |-- raw/               # Original dataset
|   |-- processed/         # Cleaned data
|   +-- predictions/       # Final predictions.csv
|-- src/
|   |-- preprocessing.py   # M1: Data Preprocessing
|   |-- equity.py          # M2: Equity Analysis
|   |-- features.py        # M3: Feature Selection
|   |-- models.py          # M4: Predictive Modeling
|   |-- calibration.py     # M5: Fairness Calibration
|   |-- uncertainty.py     # M6: Uncertainty Quantification
|   +-- outputs.py         # M7: System Outputs
|-- tests/                 # Unit and integration tests
|-- docs/
|   |-- reports/           # Analysis and technical reports
|   +-- presentation/      # Final presentation
|-- requirements.txt       # Python dependencies
+-- README.md             # Setup and usage instructions
```

1.4.5 Success Criteria

Technical Goals:

- Prediction model works accurately (stratified C-index > 0.70)
- Fairness similar across all demographic groups (gap < 0.10)
- Code clean and well-documented with passing tests

Project Goals:

- All 4 milestones (M1-M4) completed on time
- Team collaboration effective with clear communication
- Deliverables professional and clear
- Final submission by December 5, 2025

Learning Goals:

- Team learns about fairness in machine learning
- Team practices data analysis and prediction techniques
- Team develops teamwork and project management skills
- Team completes project successfully with quality outcomes

1.5 Conclusion

This workshop transitions from problem identification (W1) and solution design (W2) to operational frameworks for quality (W3). The 4-week implementation plan (November 8 - December 5, 2025) makes these frameworks practical with concrete milestones and deliverables.

Challenges Faced in This Phase: The main challenge was making our theoretical design work with real-world constraints. Translating abstract concepts like “robustness” and “fairness” into concrete, measurable criteria took careful thought about what thresholds would be both achievable and meaningful. Setting the C-index gap threshold at > 0.10 balances our equity goals with realistic model capabilities given complex medical data.

Another challenge was coordinating our 4-member team with varying time commitments (5-25 hours/week) across a tight 4-week schedule. We solved this by defining clear role boundaries (programmer, coordinator, analyst, quality specialist) and establishing weekly sync points (Monday meetings, Friday professor reviews) to maximize parallel work while keeping everyone coordinated.

Key Decisions:

Operationalizing Quality: Rather than vague “ensuring fairness,” we defined concrete criteria: instability > 0.15, bias > 0.10, missingness $> 30\%$, schedule delay > 2 days. These thresholds (Table 1.2) enforced via monitoring transform quality into engineering constraint.

Milestone-Driven Sprints: The 4-week timeline with weekly milestones (M1-M4) provides clear checkpoints. Each milestone has defined deliverables and success criteria, helping us catch issues early and make corrections.

Architecture as Fairness: Robustness and fairness are built into the architecture (M2, M5, M6 modules), not added after the fact. The implementation plan assigns specific team members to each module, ensuring accountability.

Standards-Based Defensibility: Grounding design in ISO 9000, CMMI, Six Sigma (Table 1.1) creates a defensible architecture. The weekly Monday meetings and professor reviews ensure continuous alignment with quality standards.

Implementation Readiness: Our team of 4 members with complementary skills (programming, coordination, data analysis, quality assurance) is ready to execute the plan. The clear role definitions prevent overlap and ensure all technical and managerial needs are covered.

The GitHub repository structure maps directly to the 7-module architecture, making parallel development and integration easier. Weekly communication protocols (Monday meetings, daily Slack updates, GitHub commits) help maintain team cohesion and visibility. Risk management strategies address both technical challenges (emergent behaviors, feature bias, data quality) and implementation challenges (schedule delays, team availability, accuracy targets) with predefined responses, reducing time spent making decisions during crises.

Path to Success: As we move from planning (this workshop) to execution (4-week sprint), the structures we’ve established—monitoring thresholds, quality gates, role responsibilities, milestone criteria, communication protocols—will help us deliver a reliable, equitable system that serves clinical needs. By December 5, 2025, we’ll have transformed our architectural vision into working implementation, validated against both technical metrics (C-index > 0.70) and equity standards (gap < 0.10), making sure clinicians can trust our system for these life-critical HCT survival predictions.

References

Auletta, J. J., Kou, J., Chen, M., & Shaw, B. E. (2020). Indications for hematopoietic cell transplantation and immune effector cell therapy: Guidelines from the american society for transplantation and cellular therapy. *Biology of Blood and Marrow Transplantation*, *26*(7), 1247–1261. Retrieved from https://www.astctjournal.org/article/S1083-8791(20)30114-2/fulltext doi: 10.1016/j.bbmt.2020.03.002

CIBMTR. (2025). *Publicly available datasets.* Retrieved from https://cibmtr.org/CIBMTR/Resources/Publicly-Available-Datasets (Accessed: 2025-09-26)

CMMI Institute. (2018). *A guide to CMMI version 2.0* (Tech. Rep.). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. (Available at the CMMI Institute website)

DataCamp. (2023a). *Introduction to shap values: Machine learning interpretability.* Retrieved 2025-10-18, from https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability

DataCamp. (2023b). *Tutorial: Machine learning pipelines, mlops & deployment.* Retrieved 2025-10-18, from https://www.datacamp.com/tutorial/tutorial-machine-learning-pipelines-mlops-deployment

Doherty, T. S., Char, D. S., Goodman, S. N., Shah, N. H., & Oberst, M. (2024). Addressing ai algorithmic bias in health care. *JAMA*. Retrieved from https://jamanetwork.com/journals/jama/fullarticle/2823006 doi: 10.1001/jama.2024.16564

Dosi, J. (2023). *Implementing fairness and transparency in ai projects: A technical guide.* Retrieved 2025-10-18, from https://www.linkedin.com/pulse/implementing-fairness-transparency-ai-projects-technical-jaydeep-dosi-sstjc

Eisenstadt, M., Wagner, T., Biessmann, F., & Ustyuzhanin, A. (2020). Designing a machine learning pipeline architecture for production. *arXiv preprint*. Retrieved 2025-10-18, from https://arxiv.org/pdf/1911.09638.pdf

GeeksforGeeks. (2023a). *Cox proportional hazards model.* Retrieved 2025-10-18, from https://www.geeksforgeeks.org/data-science/cox-proportional-hazards-model/

GeeksforGeeks. (2023b). *Data preprocessing in machine learning with python.* Retrieved 2025-10-18, from https://www.geeksforgeeks.org/machine-learning/data-preprocessing-machine-learning-python/

Harrington, P., de Lima Zuchner, C., McGowan, R., Gormley, N., Hill, Q. A., & Snowden, J. A. (2025). Editorial: Improving stem cell transplantation delivery using computational modelling. *Frontiers in Immunology*, *16*. Retrieved from https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2025.1579353/full doi: 10.3389/fimmu.2025.1579353

International Organization for Standardization. (2015). ISO 9000 family - quality management [Computer software manual]. Retrieved from https://www.iso.org/iso-9001-quality-management.html (Geneva, Switzerland)

Kaggle and CIBMTR. (2025). *Cibmtr - equity in post-hct survival predictions.* Retrieved from https://www.kaggle.com/competitions/equity-post-HCT-survival-predictions (Accessed: September 2025)

Kucab, M., Mitchell, T. J., Ariffin, H., Campbell, P. J., Rahman, N. A., Scelo, G., . . . Prokunina-Olsson, L. (2024). Characterization of clonal dynamics using duplex sequencing in diagnoses and relapses of acute myeloid leukemias. *Science Translational Medicine*, *16*(770), eado5108. Retrieved from https://www.science.org/doi/10.1126/scitranslmed.ado5108 doi: 10.1126/scitranslmed.ado5108

Microsoft. (2023). *Fairlearn.* Retrieved 2025-10-18, from https://fairlearn.org/

Neptune.ai. (2023). *When to choose catboost over xgboost or lightgbm.* Retrieved 2025-10-18, from https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm

Pushkarna, C. (2023). *Data preprocessing with pandas, numpy, scikit-learn.* Retrieved 2025-10-18, from https://www.kaggle.com/code/chetalipushkarna/data-preprocessing-with-pandas-numpy-scikit-learn

Pyzdek, T., & Keller, P. A. (2018). *The six sigma handbook* (5th ed.). McGraw-Hill Education.

Pölsterl, S. (2023a). *Introduction to scikit-survival.* Retrieved 2025-10-18, from https://scikit-survival.readthedocs.io/en/stable/user_guide/00-introduction.html

Pölsterl, S. (2023b). *Random survival forest.* Retrieved 2025-10-18, from https://scikit-survival.readthedocs.io/en/stable/user_guide/random-survival-forest.html

Scikit-learn. (2023). *Cross-validation: evaluating estimator performance.* Retrieved 2025-10-18, from https://scikit-learn.org/stable/modules/cross_validation.html

SHAP. (2023). *Shap documentation.* Retrieved 2025-10-18, from https://shap.readthedocs.io/

Shelf.io. (2023). *Fairness metrics in ai.* Retrieved 2025-10-18, from https://shelf.io/blog/fairness-metrics-in-ai/

Trusted AI Team. (2023). *AI Fairness 360 (AIF360).* Retrieved 2025-10-18, from https://github.com/Trusted-AI/AIF360

Yuan, S., Chen, G., Matsumoto, G. I., Chiu, S.-T., Clarke, P., Wakefield, D. L., . . . Warner, J. J. W. (2024). Nih data translators: Developing translatable insights from cross-disciplinary biomedical data. *Journal of Biomedical Informatics*, *151*, 104610. Retrieved 2025-10-18, from https://pmc.ncbi.nlm.nih.gov/articles/PMC11430446/