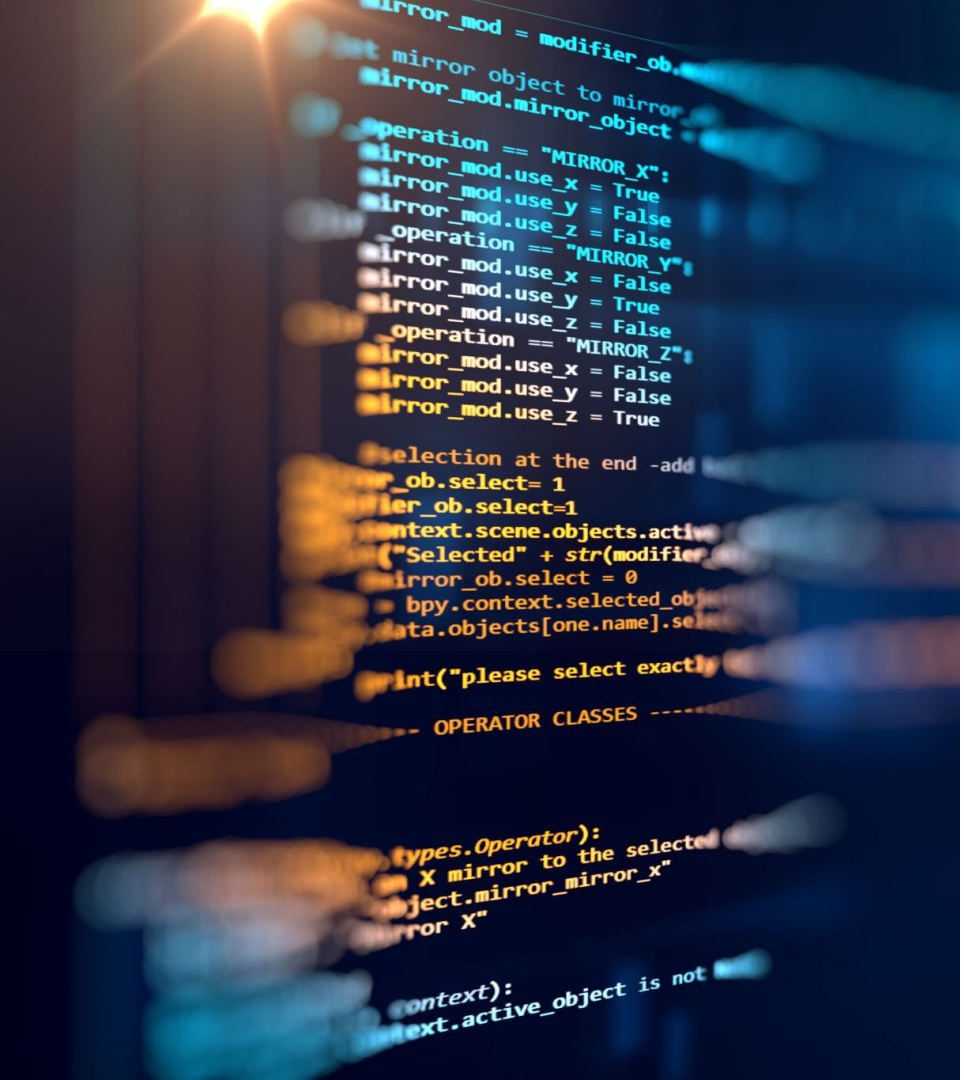


Agile Project:

Core Data & Document Management System Migration

Agile Methodologies: Scrum, BDD (Behavior Driven Development), XP (Extreme Programming)

PRESENTED BY NAN ROSS, PMI-ACP, CSM, SSM, CSPO

Introduction

In this project, a fintech company recognized the need to transition from an outdated platform to a more efficient, scalable CMS to meet growing demands. This migration not only addresses immediate operational challenges but also positions the organization for future growth. The project was complex, involving various stakeholders such as developers, data engineers, and external vendors. The timeline was strict, with the go-live date set for early 2024.

Project Objective & Goals

Objective: Successfully migrate data and document templates from the source platform to a scalable, and target Content Management System (CMS) to enhance operational efficiency, data integrity, and accessibility.

Goals:

- **Reduce Server Processing Time**: Optimize server performance to ensure faster data processing.
- **Enhanced Code Quality**: Less scripting can lead to cleaner, more maintainable code, reducing the likelihood of bugs and errors.
- **Enhance Collaboration**: Easy access to data across the organization fosters collaboration and knowledge sharing among teams.

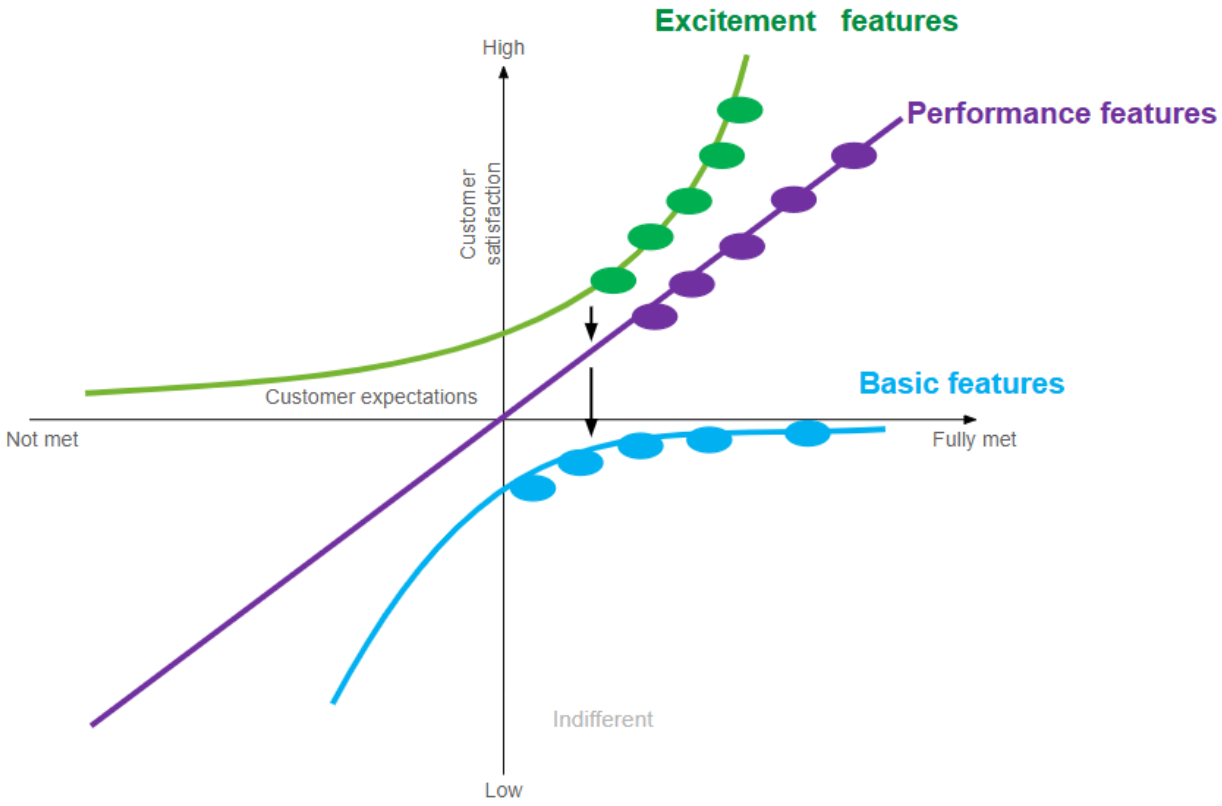
Impacted Stakeholders

1. **Customers** - Receive timely, accurate and personalized financial documents by mail
2. **Print Vendor** - Receive documents for processing, printing and mailing to the customers
3. **(2) Vice President of Software Engineering** - Overseas the strategic direction and ensure alignment with organizational goals

Project Team Members:

Data Engineer	Java Developer
Tech Team Lead/SQL Developer	(2) Front-End Developers
(2) Back-End Developers	DevOps Engineer
Product Owner	Scrum Master

Kano Model



Basic Features:

- Secure document storage and retrieval.
- Compliance with financial regulations.
- Basic search and indexing.
- User authentication and access control.
- Version control and audit logs.
- Document backup and recovery.

Performance Features:

- Speed of document retrieval.
- Scalability for growing data.
- Seamless integration with existing fintech systems.
- Automated approval workflows.
- Advanced search with filters.
- Real-time synchronization and cross-department collaboration tools.

Excitement Features:

- AI-powered document classification.
- Predictive analytics for document trends.
- Voice-enabled document search.
- Instant translation for global teams.
- Blockchain-based document integrity verification.
- Customizable document workflows.

Product Vision Board

VISION	Seamlessly migrate the document management system and data to a cloud-based CMS, ensuring enhanced security, scalability, and accessibility for financial operations, while reducing operational overhead and improving collaboration across teams.

TARGET GROUP	NEEDS	PRODUCT	BUSINESS GOAL
1. Internal employees in the legal, compliance, and operations departments 2. External auditors and regulators 3. IT and security teams responsible for system maintenance 4. Business partners who need access to shared documents	1. Secure, reliable, and scalable storage solution 2. Enhanced collaboration and document sharing capabilities 3. Compliance with financial regulations and data security standards 4. Improved accessibility to documents from multiple locations 5. Efficient search and retrieval functionality	1. Cloud-based CMS with secure data migration 2. Advanced permission controls and auditing features 3. Seamless integration with existing financial tools and applications 4. Automated workflows for document approval and version control 5. Real-time collaboration and document sharing across departments	1. Complete migration of all data with zero data loss or downtime 2. Achieve compliance with financial and legal data storage standards 3. Reduce operational costs associated with legacy systems 4. Increase user adoption and efficiency in document handling processes 5. Improve data accessibility and security across the organization

Porter's Five Force Model

Bargaining Power of Suppliers (Medium)	Threat of Substitute Products (Low)	Bargaining Power of Buyers (Medium)	Threat of New Entrants (Low)	Rivalry Among Existing Competitors (High)
• Dependence on Cloud Providers • Limited Supplier Choices • Potential for Vendor Lock-in	• Few Direct Substitutes • Legacy Systems Are Outdated • Custom-Built In-House Solutions	• Limited Buyer Choice for Complex CMS Systems • Customizable CMS Options • Switching Costs	• High Barriers to Entry • Regulatory Challenges • Established Cloud Providers	• Many Competitors in the Market • Price Sensitivity • Product Differentiation • Switching Costs

Project Risk

Risk Types	Description	Mitigation
Data Security and Privacy	Sensitive data may be exposed or compromised during migration	Ensure compliance with data protection and restrict access to authorized users only
Compatibility Issues	Incompatibility between the source and target systems could cause migration failures	Perform thorough testing to identify and address issues early
Skill Gaps	The developers might lack the necessary skills for the new CMS platform	Provide adequate training to the team and the availability of CMS developers for Q&A
Performance Degradation	The performance of the target system might degrade due to improper data loading	Optimize the data transformation processes and conduct performance testing before deploying to UAT

Project Issues

Issues Types	Description	Mitigation
Incorrect or Incomplete Data mapping	Data fields from the source system may not map correctly to the target system	Review and verify data mapping thoroughly and include a subject matter in the mapping process
Technical Failures	Software failures during development can halt the process	Document the failure, including error logs, immediate troubleshooting, or consult with a subject matter for shared knowledge for potential solutions
Post-Deployment Performance Issues	The data files may experience performance issues post-deployment	Monitor the file performance closely, flag data file errors, and send automated notifications to designate personnel and troubleshoot quickly
Manual Testing	Due to the lack of automated testing, developers can easily overlook bugs and defeats in upper environment testing	Collaborated with Software Engineers to implement automated testing to reduce bugs and defects

Planning Stage – Sprint Zero

Setting Up the Environment:

1. **Development Environment**: Configure servers, databases, and software tools (e.g., Jira, version control systems like Git).
2. **Test Environment**: Set up testing frameworks, API curl calls, Continuous Integration (CI)
3. **Deployment Pipelines**: Establish the necessary pipelines for continuous delivery or deployment.

Setting Up for Data Migration:

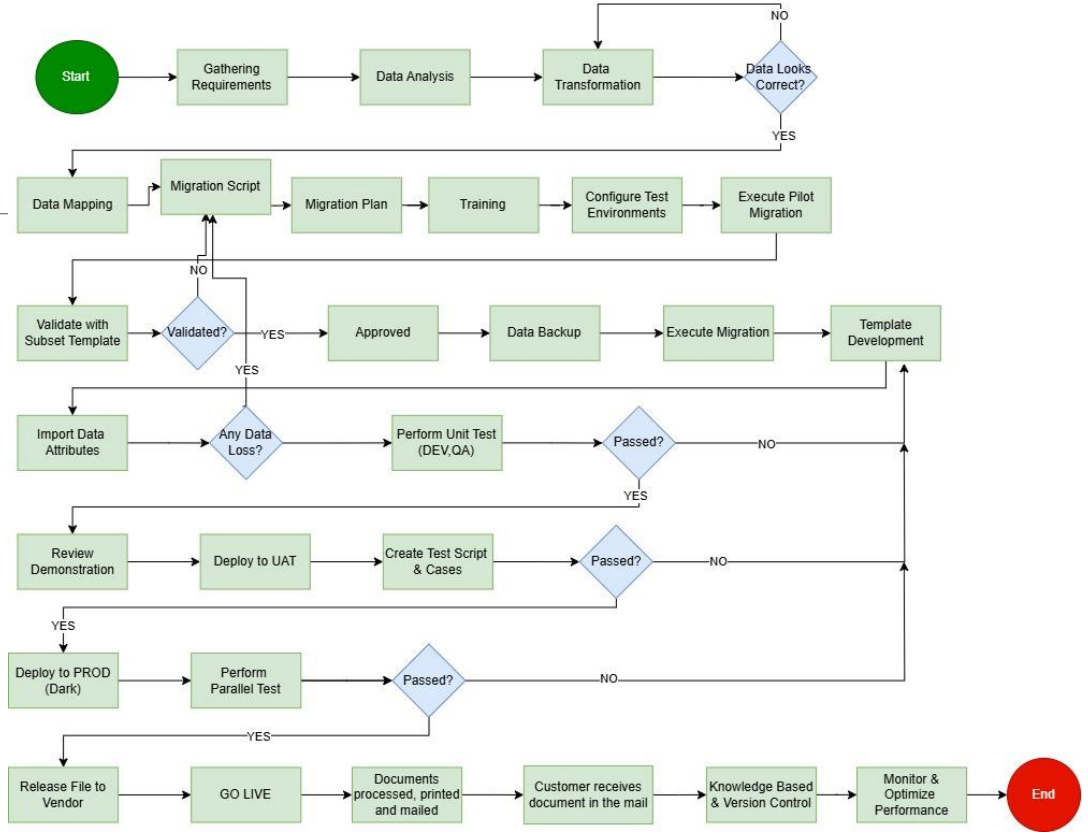
1. **Identify Data Sources**: Determine where the data currently resides (databases, applications, cloud storage, etc.).
2. **Determine Scope**: Decide what data will be migrated, including any exclusions (e.g., outdated or redundant data).
3. **Data Quality Assessment**: Analyze the quality of the existing data, identifying inaccuracies, duplicates, or incomplete records.

Planning Stage – Sprint Zero

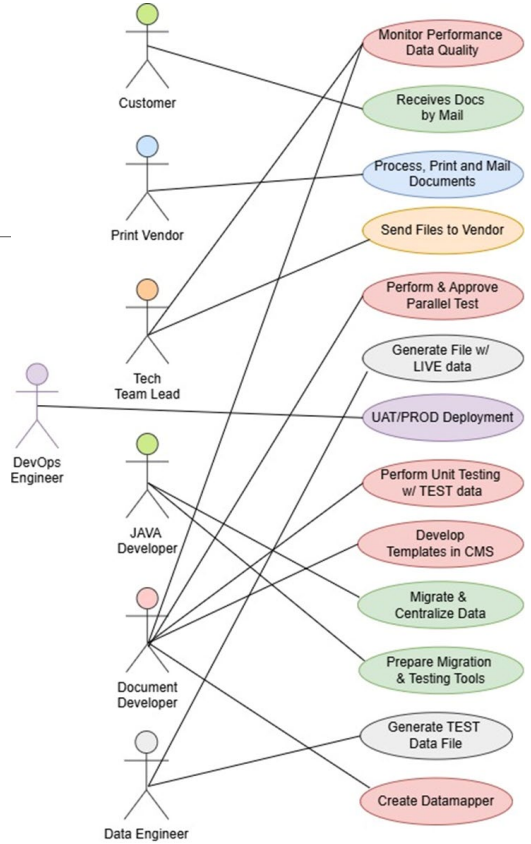
Setting Up for Data Migration:

1. **Data Mapping**: Create a data map to understand how the data will be transformed during migration, including field mappings between the source and target systems.
2. **Data Dependencies**: Identify any dependencies, relationships, or integrations between different data sets.
3. **Created a Data Migration Plan with a Phased Migration Approach**: Include timelines, key milestones, roles, and responsibilities. Define what will happen before, during, and after the migration. Data is migrated in stages or increments to minimize risk and maintain operational continuity.
4. **Coordinate Data Assessment**: Work with the development team and data specialists to conduct a data quality assessment. Facilitate discussions on how data will be mapped, cleansed, and migrated.
5. **Support Data Cleansing Activities**: Ensure that the team includes data cleansing activities in the sprint backlog, with clear acceptance criteria around accuracy, consistency, and completeness.

End-to-End Workflow



Use Case Diagram



Planning Stage

Facilitated Backlog Refinement Sessions:

- Ensured regular backlog refinement sessions were scheduled, held, and focused on refining high-priority items without allowing discussions to go off-track.
- Helped the Product Owner break down features into user stories that may be too vague or too large.
- Ensured the team was aware of the prioritized items and aligned the team's focus on what was most important.
- Ensured that backlog items met the team’s Definition of Ready before sprint consideration.
- Encouraged the team to size or estimate the user stories using techniques like story points.
- Ensured the backlog was continuously refined and enough PBI (Product Backlog Items) were always ready for upcoming sprints.

Feature Name: Data Migration

Story No.: DM101	Story No.: DM102	Story No.: DM103	Story No.: DM104
As a system administrator, I want to identify and categorize all legacy documents and data for migration, so that the data is organized for efficient transfer to the new CMS.	*As a system administrator, I want to create a backup of all legacy documents and data, so that no information is lost during the migration process.*	*As a system administrator, I want to map the legacy data fields to the new CMS data fields, so that all data is correctly formatted and structured in the new system.*	*As a system administrator, I want to test the migrated data for completeness and accuracy, so that we ensure no data loss or corruption during the migration.*
Acceptance criteria: **Use Case #1:** **Given** that the legacy data has been accessed, **When** the system administrator reviews and organizes the documents, **Then** all documents are categorized into predefined groups with complete metadata. **Use Case #2:** **Given** that categorization is complete, **When** a verification check is performed by another team member, **Then** the categorization should be confirmed as complete and documented.	**Acceptance criteria**: **Use Case #1:** **Given** that the legacy system is ready for backup, **When** the backup process is initiated, **Then** a complete backup of all documents, metadata, and settings is created and stored securely. **Use Case #2:** **Given** the backup is completed, **When** a test restoration is conducted, **Then** the system should successfully restore from the backup without errors.	**Acceptance Criteria:** **Use Case #1:** **Given** the legacy data fields have been identified, **When** the mapping process is initiated, **Then** all legacy fields should be accurately mapped to the corresponding CMS fields. **Use Case #2:** **Given** the mapping is complete, **When** the migrated data is tested, **Then** the sample data should show proper field alignment and data integrity.	**Acceptance Criteria:** **Use Case #1:** **Given** the data migration has been completed, **When** the system administrator tests the migrated data, **Then** no data loss or corruption should be detected, and all data should match the original legacy system. **Use Case #2:** **Given** the test is completed, **When** the stakeholders review the test results, **Then** they must approve the data migration as complete and accurate.
Story size: 5	**Story size**: 3	**Story size**: **5**	**Story size**: **8**

Planning Stage – Overall Outcome

- A validated set of requirements tied to use case diagrams and end-to-end workflows.
- **Use Case Diagram**: Clearly defined, accurate, and validated with stakeholders, ensuring that all interactions and processes were captured.
- **End-to-End Workflow**: A seamless workflow that was validated through discussions and refined to ensure that the final implementation reflects the actual business process.
- **User Stories**: Developed detailed, actionable user stories with clear acceptance criteria aligned with the diagrams and workflows.
- Reduced risks, improved development clarity, and increased alignment between stakeholders and the technical team.
- **CMS Training**: Facilitated necessary training, especially for tools, processes, and technologies that the team will be using.
- **Technical Architecture**: Defined the system architecture, technology stack, frameworks, and tools that will be used throughout the project.

Backlog Refinement – Overall Outcome

- Dissected complex features into smaller, actionable user stories
- Prioritized user stories based on business value and technical complexity to ensure that the most critical aspects of the migration were addressed first
- The team assigned story points based on the level of effort.
- Clarified acceptance criteria to prevent miscommunication and rework
- Facilitated estimation to help the team understand the complexity and risk of each story
- Identified risks and blockers such as external dependencies, unclear requirements or technical constraints.

Sprint Planning

Facilitated Sprint Planning Sessions:

- Scheduled and organizing the session keeps the meeting focused on the agenda within the allotted timebox.
- Collaborated with the Product Owner and the team to help define a clear Sprint Goal for one feature that are achievable and align with the overall product vision.
- Ensured the product backlog was refined and prioritized and that the team had enough well-defined stories that met the **Definition of Ready** (**DoR**).
- Ensured user stories were clear and had detailed acceptance criteria so that the team understood what needed to be done to mark them as "Done.“
- Helped the team and Product Owner identify any dependencies, risks, or external factors that might affect the sprint’s scope and ensured they were addressed during the planning.

Daily Standup

- **Removed Blockers & Impediments:**
  - Access to legacy data systems.
  - Data integrity problems (e.g., corrupted data).
  - Performance issues with migration scripts.
  - Infrastructure limitations (e.g., database performance).
- **Resolved Issues Quickly** by coordinating with other teams, stakeholders, or external vendors to ensure that technical or resource-based issues are removed as soon as possible.
- **Escalated Critical Risks** (e.g., system outages, data loss) to the appropriate stakeholders (e.g., product owners or IT management) for resolution.

Design Stage - Outcome

- Ensured the design phase resulted in a **detailed and scalable solution** for both data migration and document templates, meeting all **non-functional requirements**.
- Identified and managed key **technical constraints** early to help reduce **technical debt** and keep the project on track.
- Ensured risks were proactively mitigated through **early testing and validation to prevent** critical issues from emerging later.
- Managed **external dependencies** early to ensure smooth integration with vendors and external systems, reducing delays and ensuring a timely, successful deployment.
- Facilitated structured **design sessions** with cross-functional teams, to ensure the team had a clear and actionable plan for **data transformation and modeling** from the legacy system to the new CMS. This collaboration enabled smooth data migration.

Development Stage – Overall Outcome

- **Escalated Data Compatibility Issue: Outcome**: Prevented sprint delays and kept the development process on track.
- **Prioritized Data Sets for QA Testing: Outcome**: Testing started on schedule, avoiding delays in the testing phase and ensuring that defects were identified early.
- **Resolved API Blocker: Outcome**: Minimized downtime and prevented extended development delays.
- **Implemented Early Testing: Outcome**: Identified bugs early in the process, reducing the cost of fixing issues later and improving code quality.
- **Monitored Progress and Velocity: Outcome**: Provided insights into team performance, enabling better sprint planning and early detection of potential delivery risks.

Testing Stage – Overall Outcome

Coordinated Testing Efforts:
Outcome:

- Testing plans were successfully integrated into sprint planning, ensuring that both **functional and non-functional testing** were covered comprehensively.
- **Functional testing** focused on core CMS features and document templates, ensuring that these key areas met requirements.

Risk Monitored and Resolved:
Outcome:

- Continuous monitoring of risks such as **data corruption** during large uploads helped ensure mitigation strategies were in place.
- The resolution of **CMS performance issues** by optimizing queries ensured the system performed efficiently before the go-live phase.
- Early detection and resolution of these issues ensured the system was stable and ready for production use.

Testing Stage – Overall Outcome

Coordinated Testing Efforts:
Outcome:

- Testing plans were successfully integrated into sprint planning, ensuring that both **functional and non-functional testing** were covered comprehensively.
- **Functional testing** focused on core CMS features and document templates, ensuring that these key areas met requirements.

Risk Monitored and Resolved:
Outcome:

- Continuous monitoring of risks such as **data corruption** during large uploads helped ensure mitigation strategies were in place.
- The resolution of **CMS performance issues** by optimizing queries ensured the system performed efficiently before the go-live phase.
- Early detection and resolution of these issues ensured the system was stable and ready for production use.

Sprint Review – Overall Outcome

Facilitated User Acceptance Testing (UAT):

Outcome:

- The JAVA Developer demos the data migration from the legacy to the new CSM platform.
- The CMS Developer demos and showcases the exact comparison of documents between legacy and new CMS.
- The Team Lead demos the segmentation of the legacy and CMS files to the print vendor for processing. Showcased newly developed dashboard to cover document count validation and processing errors.
- For example, the discovery of **metadata display issues** during document migration was addressed and resolved swiftly before the next release, preventing post-go-live complications.

Pre-Deployment Stage – Overall Outcome

Coordinated the Release Plan:
Outcome:

- This ensured clear timelines, third-party dependency management, and smooth coordination for the go-live phase.
- The choice of a **phased deployment strategy**, starting with a dark launch, minimized risks by allowing performance validation and system integrity checks without affecting the user experience, leading to a smooth transition.

Managed Cutover Activities:
Outcome:

- Executed smoothly, with final data migration, validation checks, and system testing all completed on time.

Deployment Stage – Overall Outcome

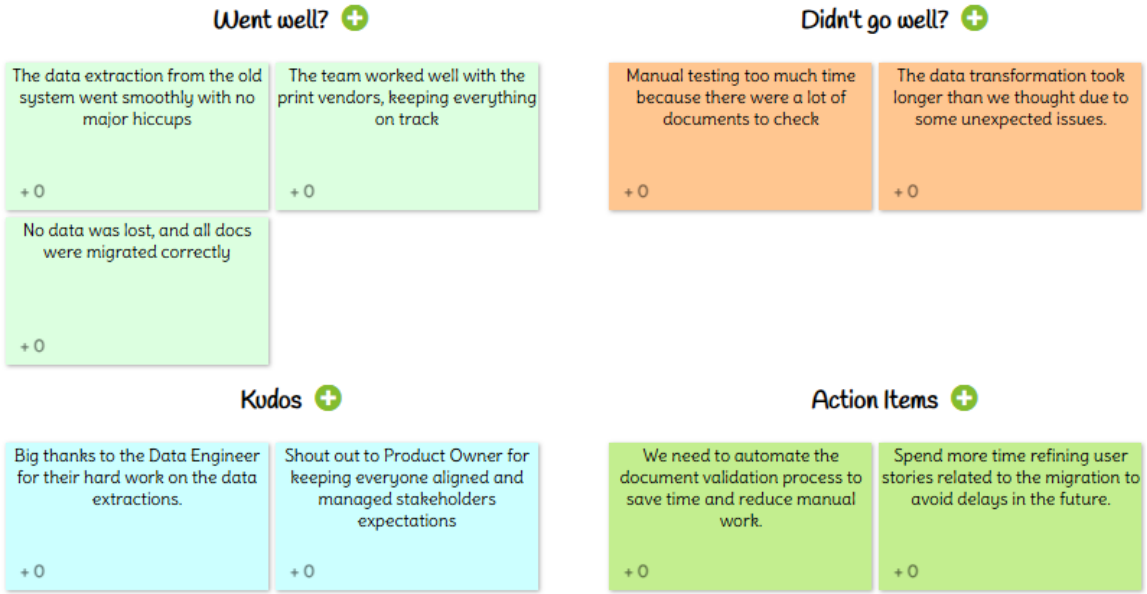
Monitored Post-Deployment Testing:
Outcome:

- Ensured the system's performance was validated and all migrated data was accurate. This helped catch any lingering issues early before they could impact users.
- When the QA team identified a **performance bottleneck** during document uploads, it was promptly addressed through a troubleshooting session with the development team. The issue was resolved before users experienced any disruptions.

Ensured Continuous Monitoring:
Outcome:

- Ensured that any potential issues were detected early.
- A minor **lag in document template loading times** was discovered through monitoring. The issue was resolved by optimizing the front-end code and improving system performance and user experience.

Sprint Retrospective – Board



Disclaimer: The data presented is for illustrative purposes only. These figures do not reflect actual company performance or metrics and are intended solely to demonstrate reporting formats and trends. Additionally, in adherence to confidentiality policies, all proprietary company information has been omitted.

Sprint Retrospective – Process Improvements

Action Items Taken:

- Collaborated with the Software Engineers to create an automated script to significantly reduce the manual effort required for validation, allowing the team to validate more documents in less time.
    - The script accurately flagged discrepancies early, enabling the team to resolve issues before the end of the sprint.
- Allocated more time for **backlog refinement**, ensuring that stories related to data migration are broken down more effectively to reduce complexity

Post – Deployment/Maintenance Stage – Overall Outcome

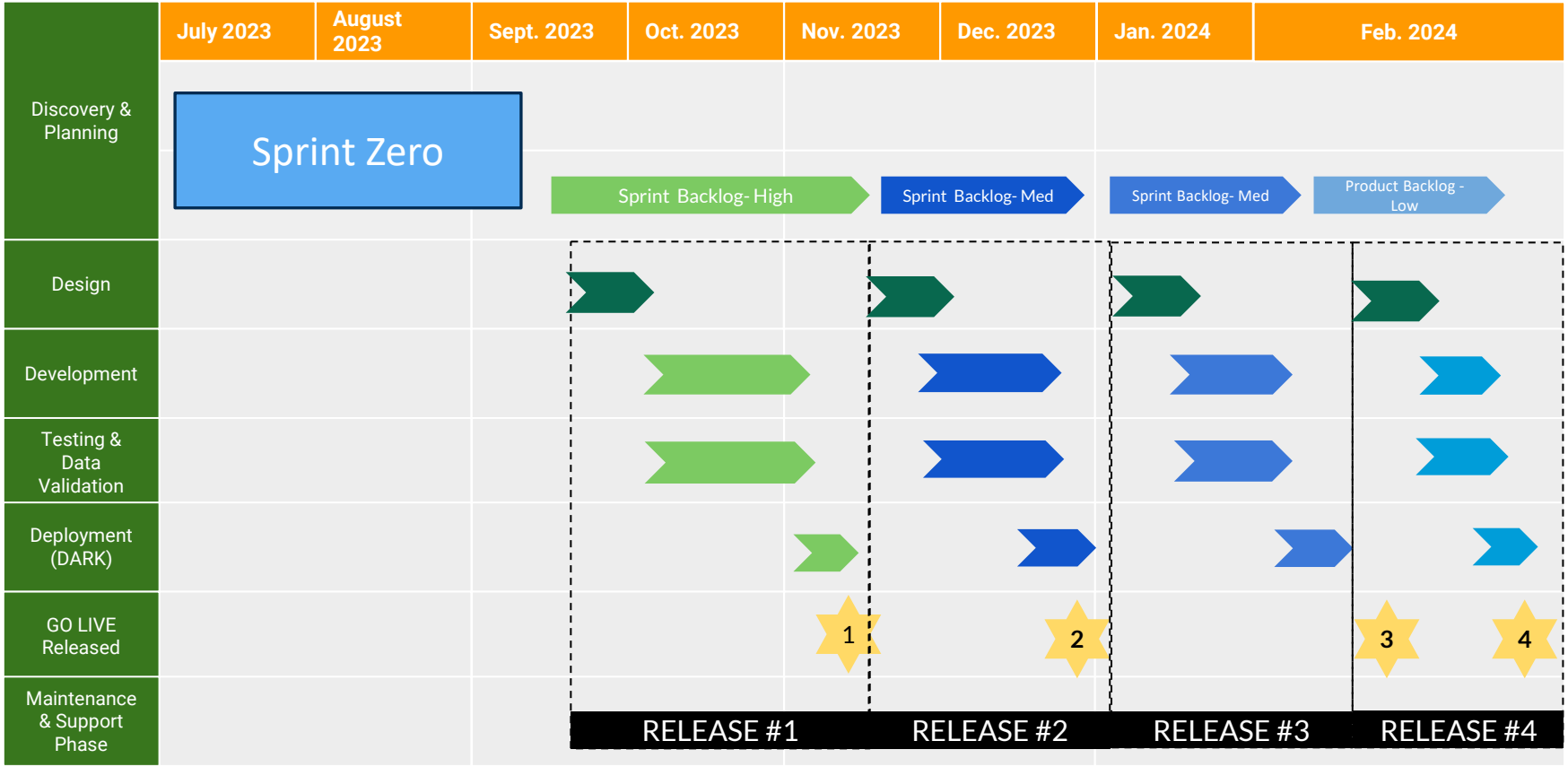
Established a Monitoring Process:

- Facilitated the setup of a **monitoring system** to track key performance metrics such as document load times, system performance, and error logs. This enabled the team to identify and address issues before they affected end users.
- For instance, the monitoring system flagged an increase in **error logs related to document upload failures** during peak usage hours. I coordinated a session with the development and operations teams, leading to server capacity optimizations that resolved the issue quickly.

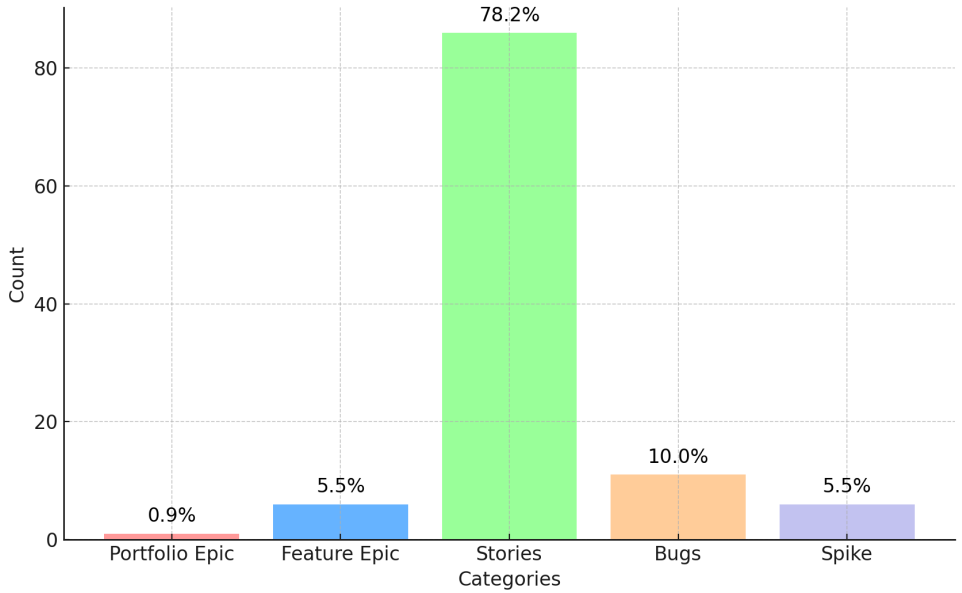
Managed Stakeholder Requests:

- Worked closely with the Product Owner to manage and **prioritize new stakeholder requests** that arose post-launch. I ensured that these requests were integrated into the backlog, balancing urgent bug fixes with less critical feature enhancements.
- For example, stakeholders requested a feature to **generate automated reports from the CMS**. I facilitated a backlog refinement session where the team discussed this request and planned it for future sprints, ensuring that the current **maintenance workload** was not disrupted.

Project Timeline

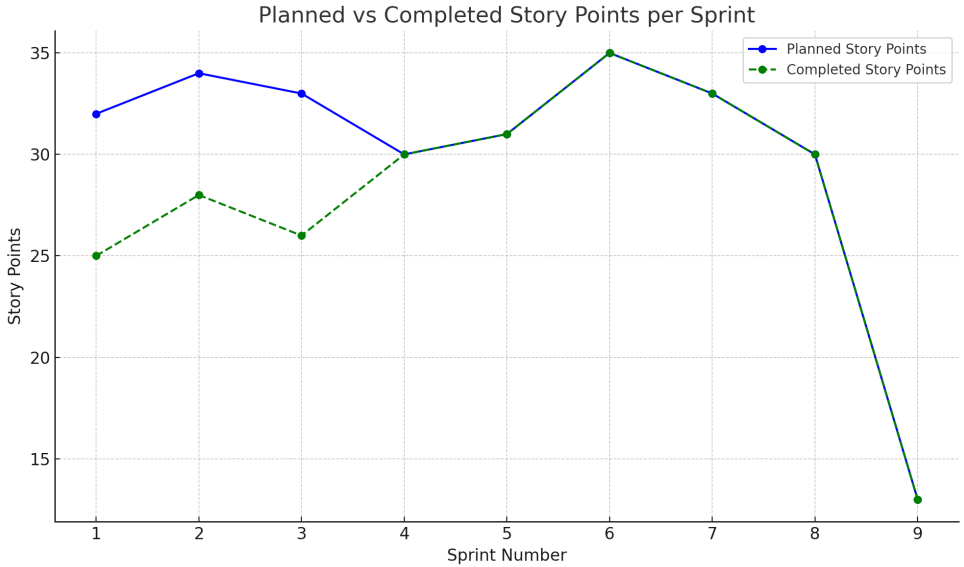


Portfolio Breakdown



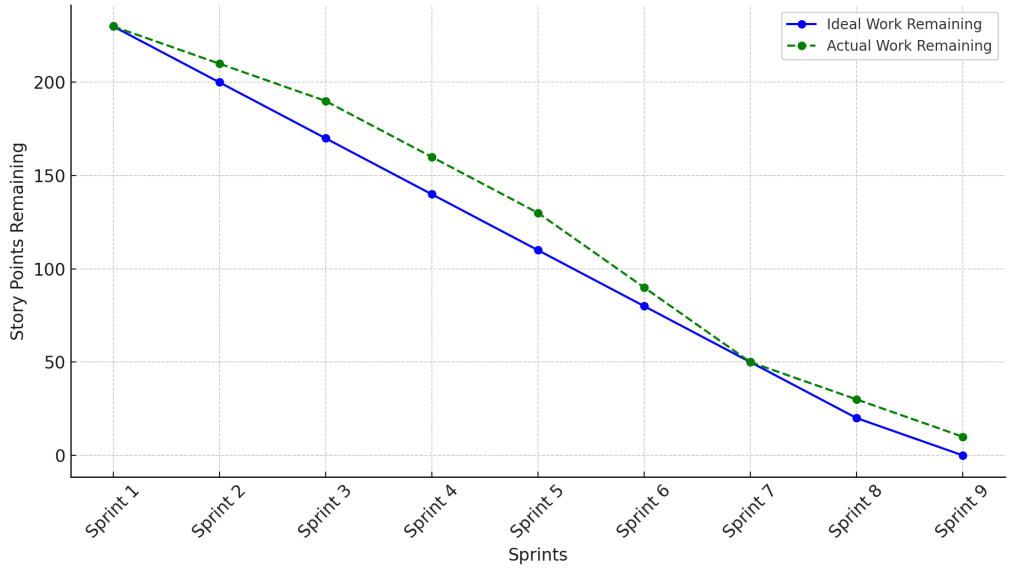
Disclaimer: The data presented is for illustrative purposes only. These figures do not reflect actual company performance or metrics and are intended solely to demonstrate reporting formats and trends. Additionally, in adherence to confidentiality policies, all proprietary company information has been omitted.

Planned Vs Completed Story Points Per Sprint



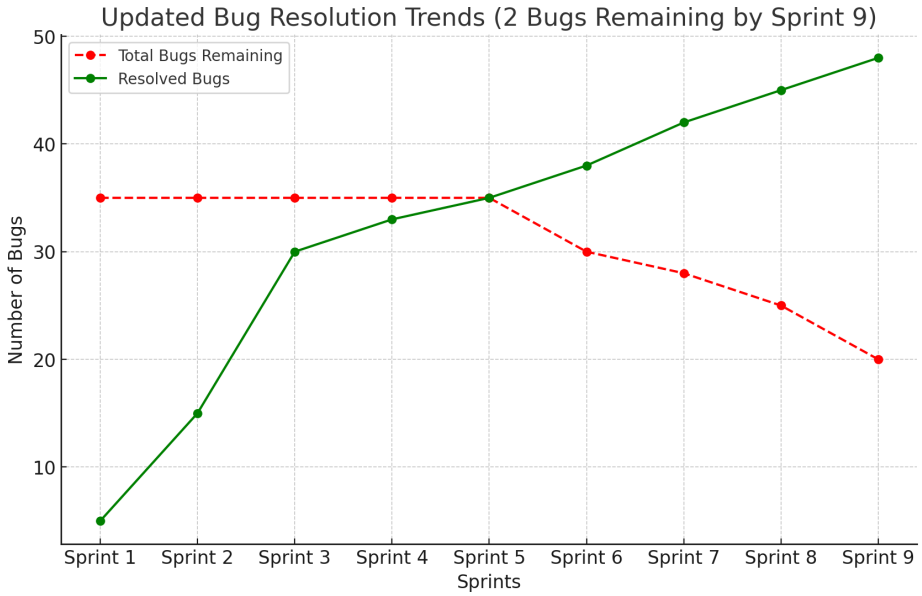
Disclaimer: The data presented is for illustrative purposes only. These figures do not reflect actual company performance or metrics and are intended solely to demonstrate reporting formats and trends. Additionally, in adherence to confidentiality policies, all proprietary company information has been omitted.

Burndown Chart



Disclaimer: The data presented is for illustrative purposes only. These figures do not reflect actual company performance or metrics and are intended solely to demonstrate reporting formats and trends. Additionally, in adherence to confidentiality policies, all proprietary company information has been omitted.

Bug Resolution



Disclaimer: The data presented is for illustrative purposes only. These figures do not reflect actual company performance or metrics and are intended solely to demonstrate reporting formats and trends. Additionally, in adherence to confidentiality policies, all proprietary company information has been omitted.

Project Success Summary

Successful Migration:

- **100% of targeted data and documents** successfully migrated from the legacy system to the new CMS.
- All data and documents were validated to **match exactly** with the legacy system in terms of content, metadata, and file integrity.

Automated Validation:

- Implemented automated scripts to compare documents between the legacy system and the new CMS.
- Reduced manual validation effort by **80%**, increasing accuracy and efficiency.

Cross-Departmental Access:

- Developers from other departments now have **immediate access** to documents, with the ability to **update, enhance, and collaborate** on the latest versions within the new CMS.

Project Success Summary

Stakeholder Confidence

- Comprehensive validation reports provided real-time insights and boosted stakeholder confidence in the migration process.

Improved Template Management

- The new CMS utilizes **less scripting** for document templates compared to the legacy system, leading to enhanced **code quality** and reduced **server processing time**, optimizing overall **server performance**.

Conclusion

As the **Scrum Master**, my role in this data migration project was to facilitate collaboration between the development team and stakeholders, remove impediments, and ensure adherence to Agile principles. By fostering continuous communication, promoting a culture of accountability, and ensuring that each sprint was focused on delivering high-value outcomes, we were able to achieve significant milestones. The successful migration of documents, the implementation of automated validation scripts, and the optimization of template management reflect the collective effort of the team in delivering a high-quality, efficient, and future-proof CMS system.

Thank You!

WEBSITE: WWW.NANROSS.COM | EMAIL: NANDALEROSS@ICLOUD.COM | LINKEDIN: WWW.LINKEDIN/IN/NANROSS | CELL: 404.263.2646