**Cloud Modernization of CDM and SFDC Workflows for Capital Group**

**Project plan**

**1. Project Overview**

**Client:** Capital Group  
**Scope:** Cloud modernization of Informatica PowerCenter-based CDM and SFDC workflows.  
**Key Stakeholders:** Capital Group, LTIMindtree (LTIM), GlobalLogic (MDM SaaS transformation)

**Systems Involved:**

* **CDM (on-premises):** Informatica PowerCenter-based system.
* **SFDC (on-premises):** Acts as a data source for CDM.
* **MDM (on-premises):** Managed by GlobalLogic and migrating to MDM SaaS.

**2. Current Status**

* CDM job extraction from Autosys: **Completed**
* CDM XML workflow dump: **Received**
* SFDC XML workflow dump: **Pending** (awaiting from source team)
* Dependency identification: **Completed** (GlobalLogic handles MDM publish dependencies)
* SFDC job extraction from Git: **Pending**

**3. Project Phases and Activities**

**Step 1: XML Workflow Collection**

* Receive XML dumps:
  + CDM: Received from Sougato
  + SFDC: Pending (follow-up required)
* Client approval required on **Master Inventory List** to freeze high-level requirements.

**Step 2: XML Inventory Analysis**

* Validate received XML files against the Master Inventory.
* Sort out discrepancies/issues with concerned business stakeholders.
* Ensure XML data quality for downstream engineering.

**Step 3: Reverse Engineering and Metadata Extraction**

* Reverse engineer all XML workflows (CDM + SFDC).
* Design database schema to store custom Informatica workflow metadata.
* Develop Python + AI-based script to parse and store XML details.
* Generate detailed Workflow Deep-Dive Analysis document.
* Identify transformation paths and opportunities for workflow factorization.

**Step 4: Approach Finalization**

**Option Exploration:**

* **Option 1:** LLM API + Manual coding (1:1 mapping to PySpark/Glue) + Manual coding
* **Option 2:** No-Code Tool (DataSwitch) + Manual refinement
* **Option 3:** Custom Conversion Tool (1:1 mapping to PySpark/Glue) + Manual coding
* **Option 4:** LLM-supported Generator Tool for Custom ETL framework + Manual coding
* **Option 5:** Fully Manual Conversion

Select preferred approach based on PoC outcomes, cost, performance, and maintainability.

**Step 5: Infrastructure Setup**

* Finalize cloud architecture based on selected approach.
* Set up dev/sandbox environment.
* Define and assign IAM roles.
* Establish test environments.
* Set up Git/Version Control.
* Implement CI/CD pipeline.
* Conduct initial code deployments with unit testing.

**Step 6: Testing & Integration**

* Develop test cases (manual and automated).
* Build or integrate automation testing frameworks.
* Incorporate testing into CI/CD.
* Record data quality results with versioning.
* Ensure compliance with CG governance and coding standards.
* Conduct performance and load testing.

**Step 7: Downstream System Integration**

* Integrate with downstream systems if required.
* Set up common data hub or publishing APIs.
* Ensure full regression and integration testing.

**Step 8: Production Deployment**

* Replicate production environment infrastructure.
* Deploy all code via CI/CD pipeline.
* Perform end-to-end testing and validation.

**Step 9: Documentation**

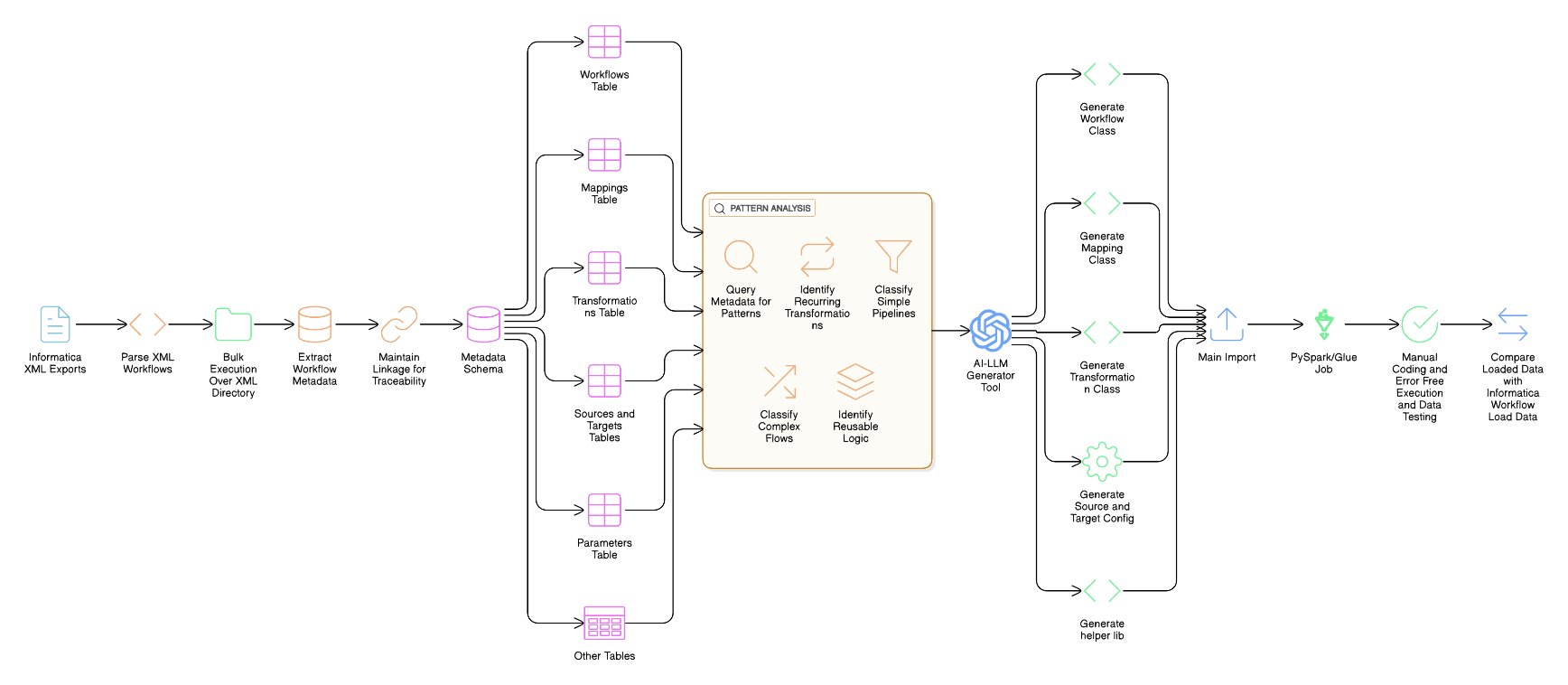
* Maintain Confluence pages for:
  + Infrastructure details
  + Testing plans and approvals
  + Technical architecture and transformation approach

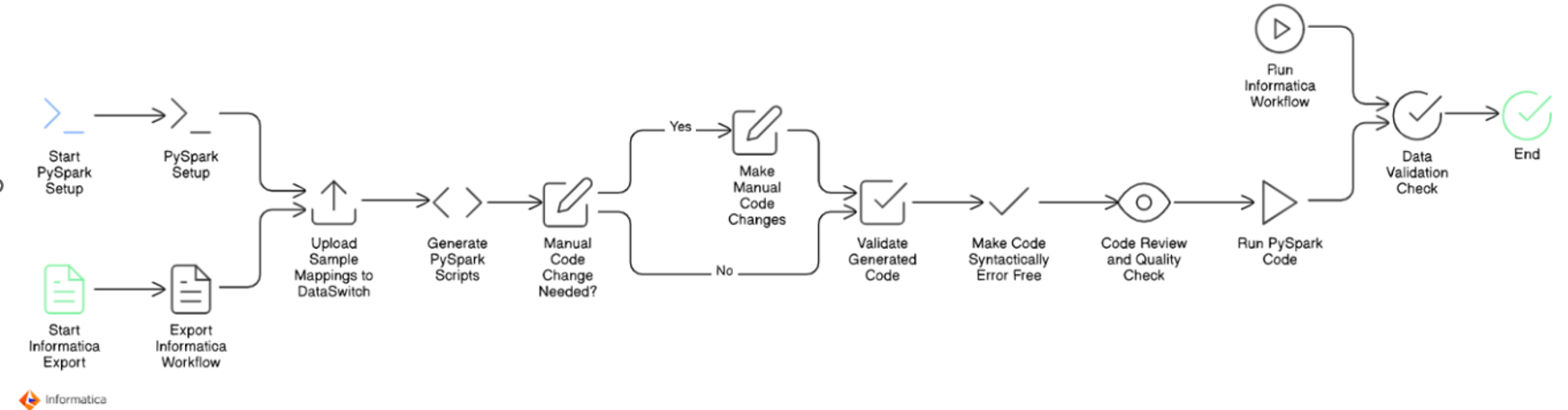
**Step 10: Final Delivery**

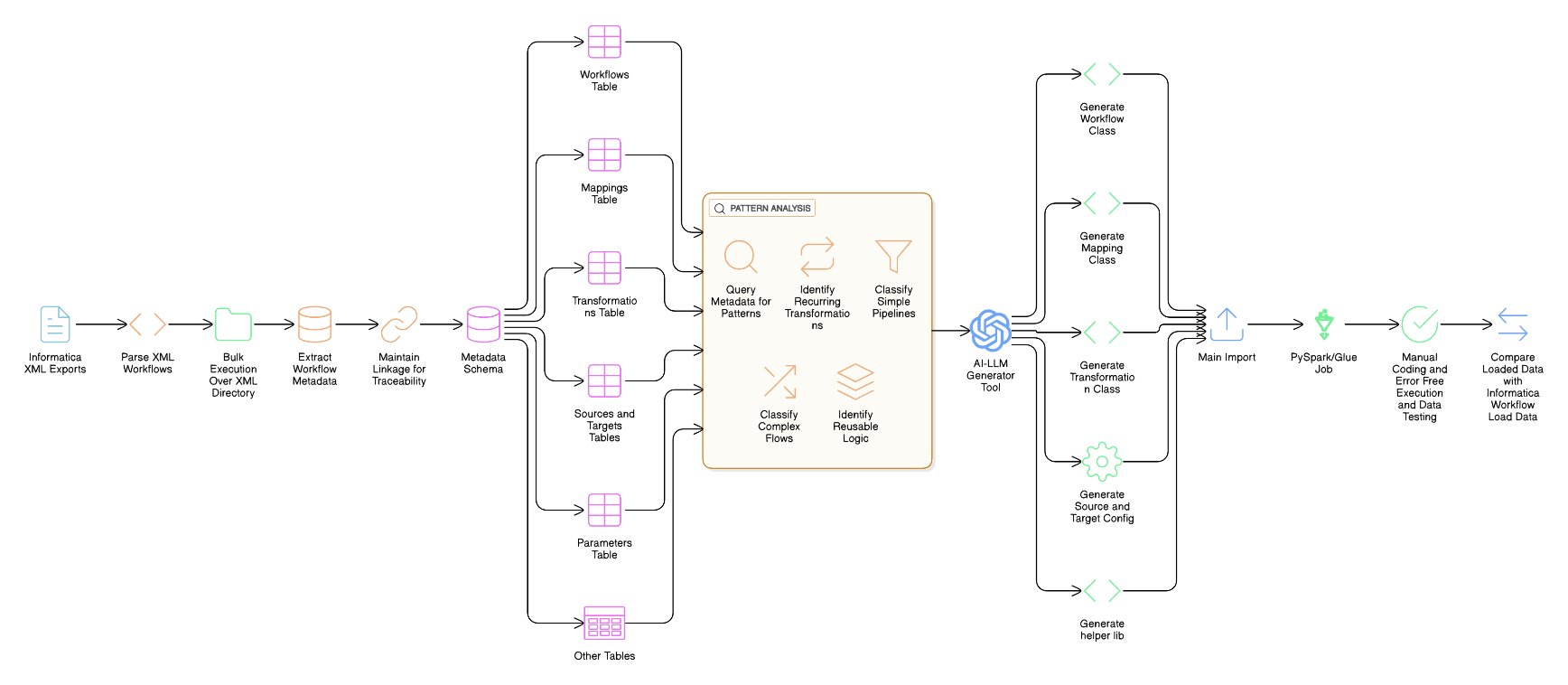
* Ensure production stability and support readiness.
* Handover to operations and support team.
* Close project with client sign-off and feedback.

**Appendices (To Be Developed)**

* Master Workflow Inventory
* Workflow Deep-Dive Analysis Document
* Approach Comparison Matrix
* Custom Metadata Schema
* Test Case Repository
* CI/CD Design Architecture
* Deployment Checklist

**Process Diagram Option 1 , 2 , 3**



**Option 4**