

# Intelligent Curtain Wall System Architecture Operation and Maintenance and Dataset Management Platform: Project Management and Economic Analysis Report



**Team Members:** 2250758 Jishen Lin、 2251730 Shuyi Liu、 2256225 Amane Nakatani

1. Project Scope
2. Project Plan
3. Software Process Monitoring and Control
4. Software Process Improvement (SPI)
5. Project Risk Management
6. Software Development and Operational Cost Estimation
7. Pricing and Pricing Strategy
8. Fundraising and Financing Analysis
9. Financial Evaluation and Results
10. Financial Risk Management
11. Conclusion

## 1. Project Scope

### 1.1 Functional Scope

#### 1.1.1 Dataset Management Function

##### 1.1.1.1 Basic Data Operations

- **Upload Dataset:** The user can select a local file through the front-end interface and upload it to Alibaba Cloud OSS. The system will verify the legitimacy of the file path.

- **Download Dataset:** The user can download the specified file from OSS through the file path. The system will verify the existence of the file and the user's permission.
- **Delete Dataset:** The user can delete the specified dataset, and the system will verify the user's permissions and synchronously delete the metadata.
- **Export Datasets:** Users can select one or more datasets to export the corresponding URL links, which are time-limited.

#### 1.1.1.2 Advanced Data Management

- **Dataset management operations:** Support operations such as copy, cut, delete, and batch paste.
- **Permission Control:**
  - **User Persona Management:** Supports roles such as administrators and ordinary users, with different roles having different permissions.
  - **Permission Verification:** verify permissions when performing operations to ensure that only authorized users can perform corresponding operations.

#### 1.1.1.3 Data Backup and Recovery

- **Data Backup:** full backup is automatically performed every week and retained for 30 days. The backup content includes datasets and their metadata.
- **Data Recovery:** Support data recovery from backup points. Only administrators can trigger the recovery operation.

#### 1.1.1.4 Log Management

- **Operation Log Recording:** Record deployment logs and system operation logs, and store them in a structured format.
- **Log Query and Analysis:** Support conditional filtering queries, generate reports or alerts.
- **Log Cleaning:** Clean up expired logs regularly. By default, keep the logs of the last 30 days.

### 1.1.2 Project Operation and Maintenance Function

#### 1.1.2.1 Environment Management

- **Environment Consistency:** Use Docker and Docker Compose to ensure consistency across development, testing, and production environments.
- **Automated Deployment:** Automatically build, test, and deploy code through CI/CD pipelines.

#### 1.1.2.2 System Monitoring

- **Resource Monitoring:** Real-time monitoring of server resource usage (CPU, memory, disk, etc.).
- **Log Monitoring:** provides monitoring functions for deployment logs and output logs.

### 1.1.2.3 Teamwork Support

- **Documentation Management:** Maintain technical architecture, development process, API documentation, etc.
- **Agile Development Support:** Team collaboration through the Scrum framework.

## 1.2 Non-functional Scope

### 1.2.1 Performance Requirements

#### 1.2.1.1 Response Time

- For files smaller than 1GB, the response time for upload and download operations does not exceed 3 seconds.
- For files larger than 1GB, the response time is no more than 10 seconds.
- Maintain stable performance under high concurrency.

#### 1.2.1.2 Resource Utilization

- Optimize server resource usage (CPU, memory, disk I/O).
- Support dynamic quota and adjust automatically according to load.

#### 1.2.1.3 High Concurrency Support

- Support uploading and downloading of large-scale datasets.
- Distributed storage and load balance technology.

### 1.2.2 Security Requirements

#### 1.2.2.1 Access Control

- Role Based Access Control (RBAC).
- Implementation of the principle of least privilege.
- Alibaba Cloud RAM for fine grain permission management.

#### 1.2.2.2 Data Protection

- Transmission encryption: Use HTTPS protocol.
- Storage Encryption: Encryption Function Provided by Alibaba Cloud OSS.

- Encrypted storage of sensitive data.

### 1.2.2.3 Security Audit

- Record all key operation logs.
- Support log query and audit trail.
- Regular security bug scanning and repair.

## 1.2.3 Scalability Requirements

### 1.2.3.1 Architecture Design

- Modularization design (front-end Vue componentization, back-end Spring Boot modularization).
- Support distributed deployment.
- Plug-in architecture design.

### 1.2.3.2 Storage Expansion

- Support dynamic expansion of storage resources.
- Support multiple storage services (local storage, cloud storage).

### 1.2.3.3 Interface Extension

- Standardized RESTful API Design.
- Support interface version control.

## 1.2.4 Maintainability Requirements

### 1.2.4.1 Code Quality

- High cohesion, low coupling design principle.
- Unified coding specification.
- Complete code comments.

### 1.2.4.2 Document Management

- Technical architecture document.
- Development process documentation.
- API documentation and operation guide.

### 1.2.4.3 Operation and Maintenance Support

- Detailed logging and monitoring.

- Automated tests cover core functions.
- Git Version Management.

#### 1.2.4.4 Deployment and Maintenance

- Containerized deployment (Docker).
- Automated deployment process (CI/CD).
- Environmental consistency management.

### 1.3 Non-covered Scope (Not Included in This Phase / Future Plan)

#### 1.3.1 Project Transfer and System Migration Function

- **Cross-Cloud Computing Platform Migration Support:** The current system design is mainly for the Alibaba Cloud Ali Cloud Aliyun platform and does not include automated tools and processes for migrating to other Cloud Computing Platforms (such as AWS and Azure).
- **Data Migration Tool:** There is a lack of dedicated data migration tools and verification mechanisms, and large-scale data migration between different storage systems cannot be supported.

#### 1.3.2 Multi-tenant and Multi-project Management

- **Multi-tenant Isolation:** The current system does not support fully isolated multi-tenant architecture and cannot provide independent environments and datasets for different customers.
- **Resource Sharing Between Projects:** lack of resource sharing and collaboration mechanism between projects, data cannot be safely shared across projects.
- **Cross-project Data Analysis:** The comprehensive analysis function across multiple projects is not supported.

#### 1.3.3 Advanced Data Analysis Function

- **Real-time Data Analysis:** The current system mainly processes static datasets and does not support real-time data stream analysis and processing.
- **Predictive Analysis:** Lack of predictive analysis functions based on historical data, such as equipment failure prediction, maintenance demand prediction, etc.
- **Custom Analysis Model:** Users cannot upload and run custom analysis models. They are limited to the analysis functions pre-set in the system.

## 2. Project Plan

## 2.1 Project Overview Statement



<b>PROJECT OVERVIEW STATEMENT</b>	Project Name Intelligent Curtain Wall System Architecture, Operation and Maintenance, and Dataset Management Platform	Project No. 2	Project Manager Jishen Lin
<b>Problem / Opportunity</b> <p>With the growing demand for intelligent building management, intelligent curtain wall systems face challenges in data management, stability, and security. This project leverages cloud storage (e.g., AliCloud OSS) and containerised deployment (e.g., Docker) to enable efficient data management, automated operation, and secure processing, enhancing building performance and reducing costs.</p>			
<b>Goal</b> <p>This project supports the continuous iteration and expansion of the project by building an intelligent curtain wall back-end system, integrating data set management and server operation and maintenance functions to ensure efficient data storage, secure access and stable system operation.</p>			
<b>Objectives</b> <ol style="list-style-type: none"> <li>1. Building an intelligent curtain wall back-end system based on cloud storage and containerization.</li> <li>2. Enable efficient dataset uploading, downloading, and access management.</li> <li>3. Implement CI/CD pipelines for automated deployment and maintenance.</li> <li>4. Provide data encryption, access control, and automated backup.</li> <li>5. Design a scalable architecture to support large-scale data processing.</li> <li>6. Optimize resource allocation to enhance system performance and stability.</li> </ol>			
<b>Success Criteria</b> <ol style="list-style-type: none"> <li>1. IR &amp;AC: The system is able to handle more concurrent requests, support more business orders, and speed up data processing to improve overall operational efficiency. Reduces manual intervention through automated operations and maintenance, optimises server resource allocation, improves storage utilisation and reduces long-term operating expenses.</li> <li>2. IS: Strengthen data encryption and access control mechanisms to ensure stable system operation and to speed up the response time for user access and data retrieval.</li> </ol>			
<b>Assumptions, Risks, Obstacles</b> <ol style="list-style-type: none"> <li>1. Assumptions: Smooth integration of new technologies, team has implementation capabilities.</li> <li>2. Risks: Technical failures, cloud service outages, staff turnover.</li> <li>3. Obstacles: System incompatibility, regulatory restrictions, cultural resistance.</li> <li>4. Management Challenges: Uneven resource allocation, poor cross-department communication.</li> <li>5. Cultural Challenges: Resistance to change, low collaboration efficiency.</li> </ol>			
Prepared By Jishen Lin, Shuyi Liu, Amane Nakatani	Date 2025/03/05	Approved By Jishen Lin	Date 2025/03/12

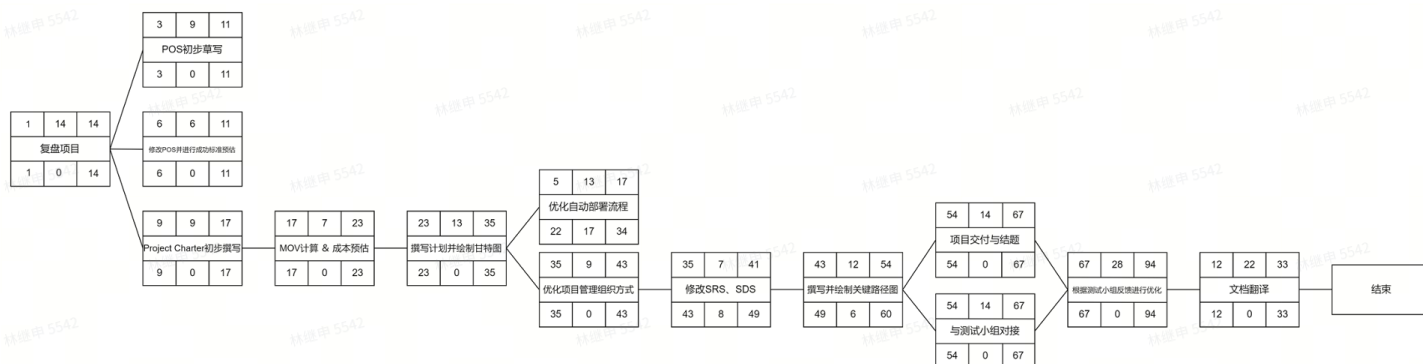
## 2.2 Work Breakdown Structure (WBS)

The complete Work Breakdown Structure (WBS) of the intelligent curtain wall system architecture operation and maintenance and dataset management platform clearly divides the seven major stages of the project life cycle in a tree hierarchy: requirements and analysis (such as stakeholder interviews, requirements document writing), system design (architecture design, API specification), function development (front and back end implementation, cloud service integration), integration and testing (unit/performance/security testing), deployment and go live (containerization, production migration), document delivery (technical manual, user training) and management and communication (agile management, risk management). Each stage further refines specific tasks (such as Vue.js development, Docker deployment, defect tracking, etc.), systematically presenting the key work nodes and deliverables of the whole process from requirements to delivery.



## 2.3 Project Schedule and Critical Path (Network Schedule Diagram)





## 3. Software Process Monitoring and Control

### 3.1 Project Tracking Tool

#### 3.1.1 Docker and Docker Compose

- **Purpose:** to achieve environmental consistency and containerized deployment
- **Function:**
  - Manage multi-container applications (such as front-end, back-end, database, etc.) through `docker-compose.yml`
  - Support one-click start/stop of full stack services
  - Mirror version management (the `docker images` list 12 service mirrors)

#### 3.1.2 CI/CD Pipeline (GitHub Actions)

- **Automated Processes:**
  - Code submission triggers build, test, and mirroring package
  - Automatically push mirroring to Docker Hub
- **Advantages:** Reduce manual intervention and ensure deployment consistency

#### 3.1.3 Log Monitoring Tool

- **Record Content:**
  - Deployment time, personnel, version status (success/failure)
  - System resource usage (CPU/memory/disk)
- **Storage Method:** Alibaba Cloud Log Service, structured JSON format

## 3.2 Development Methodology

### 3.2.1 Scrum Agile Framework

- **Practice:**

- Daily stand-up meeting to synchronize progress
- Regular review meeting
- **Communication Tools:** WeChat group/Feishu to ensure information flow

### 3.2.2 Version Control (Git)

- Code change log and version rollback capabilities
- Semantic Version Numbering

### 3.2.3 Visual Monitoring

- **Interface Example:**
  - Real-time status display of database operation and maintenance platform

```

• (base) mat@matcloud:~/Intelligent_Curtain_Wall$ sudo docker images
REPOSITORY                                TAG                                IMAGE ID                                CREATED                                SIZE
minmuslin/intelligent-curtain-wall        crack-detection                   fd66af3542c5                          4 days ago                           6.21GB
minmuslin/intelligent-curtain-wall        flatness-detection               13642aaced36                          5 days ago                           6.53GB
minmuslin/intelligent-curtain-wall        spalling-detection              567f27b51388                          5 days ago                           6.56GB
minmuslin/intelligent-curtain-wall        mobile-data                      eab6de1a2973                          10 days ago                          469MB
minmuslin/intelligent-curtain-wall        resilience-assessment            9a491bb295cd                          10 days ago                          499MB
minmuslin/intelligent-curtain-wall        modeling-communication          b2f84ce55e70                          10 days ago                          562MB
minmuslin/intelligent-curtain-wall        corrosion-detection             86f8e819aa35                          12 days ago                          7.47GB
minmuslin/intelligent-curtain-wall        vibration-detection             09dfc1127635                          2 weeks ago                          607MB
minmuslin/intelligent-curtain-wall        stain-detection                 a1c2bbcd2edc                          2 weeks ago                          6.79GB
minmuslin/intelligent-curtain-wall        user-authentication             0fd32b7212ae                          2 weeks ago                          581MB
minmuslin/intelligent-curtain-wall        mobile-data-data-logger         6d6297333f7e                          2 weeks ago                          133MB

• (base) mat@matcloud:~/Intelligent_Curtain_Wall$ sudo docker compose down
[+] Running 12/12
✓ Container flatness-detection           Removed
✓ Container crack-detection              Removed
✓ Container modeling-communication       Removed
✓ Container user-authentication          Removed
✓ Container corrosion-detection          Removed
✓ Container resilience-assessment        Removed
✓ Container stain-detection              Removed
✓ Container spalling-detection           Removed
✓ Container vibration-detection          Removed
✓ Container mobile-data                  Removed
✓ Container mobile-data-data-logger      Removed
✓ Network intelligent_curtain_wall_default Removed

• (base) mat@matcloud:~/Intelligent_Curtain_Wall$ sudo docker compose pull
[+] Pulling 11/11
✓ resilience-assessment Pulled
✓ modeling-communication Pulled
✓ vibration-detection Pulled
✓ corrosion-detection Pulled
✓ user-authentication Pulled
✓ flatness-detection Pulled
✓ mobile-data Pulled
✓ spalling-detection Pulled
✓ mobile-data-data-logger Pulled
✓ crack-detection Pulled
✓ stain-detection Pulled

• (base) mat@matcloud:~/Intelligent_Curtain_Wall$ sudo docker compose up -d
[+] Running 12/12
✓ Network intelligent_curtain_wall_default Created
✓ Container spalling-detection           Started
✓ Container mobile-data-data-logger      Started
✓ Container corrosion-detection          Started
✓ Container mobile-data                  Started
✓ Container stain-detection              Started
✓ Container user-authentication          Started
✓ Container modeling-communication       Started
✓ Container resilience-assessment        Started
✓ Container crack-detection              Started
✓ Container vibration-detection          Started
✓ Container flatness-detection           Started

```

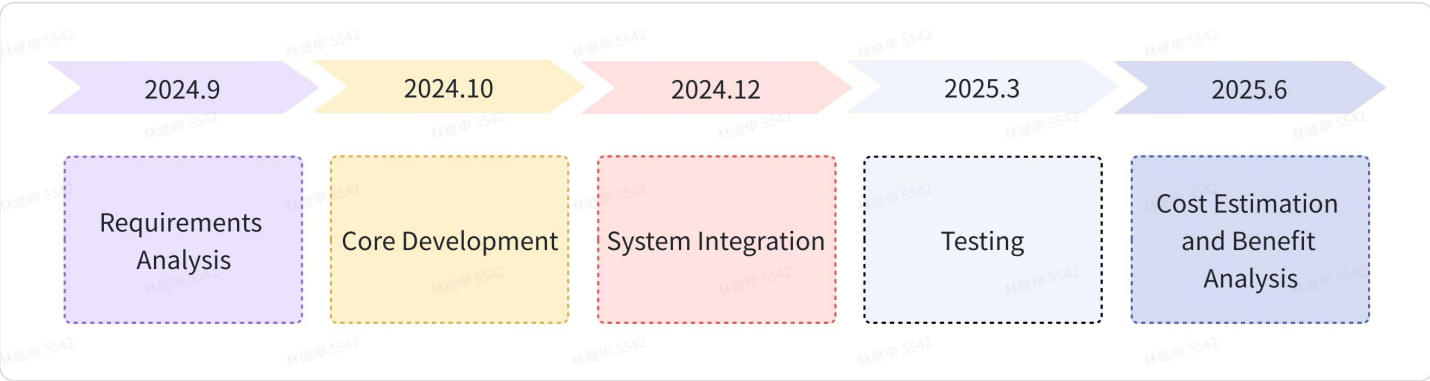
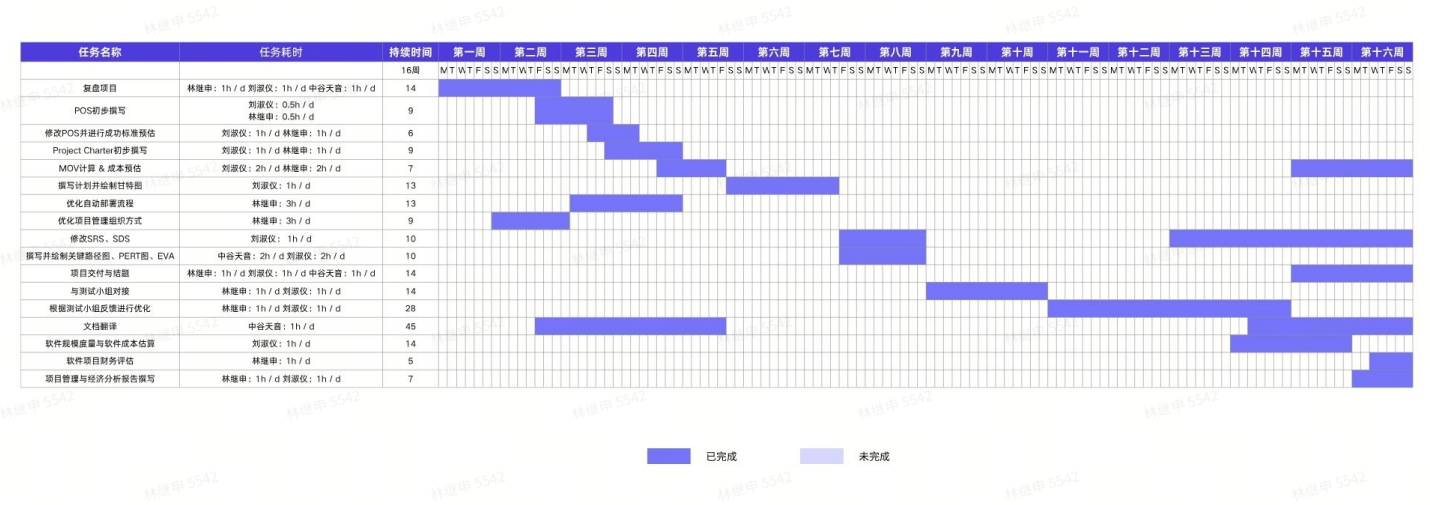
## ◦ Real-time viewing of log output

```

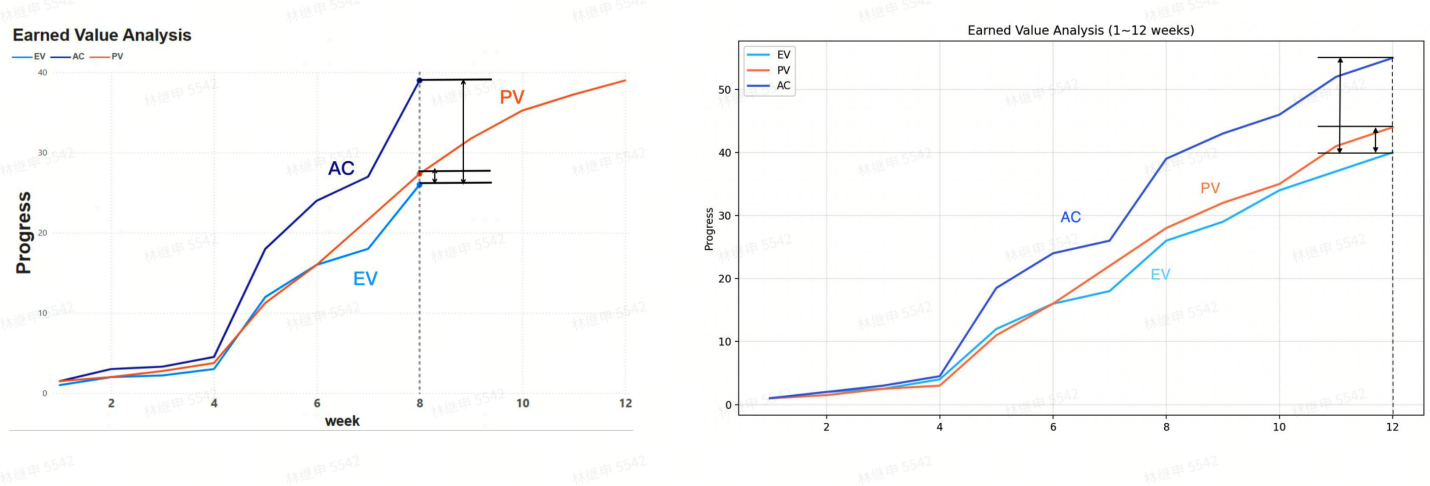
data:View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
data:Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-settings.
data: * Serving Flask app 'app'
data: * Debug mode: off
data:[31m[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
data: * Running on all addresses (0.0.0.0)
data: * Running on http://127.0.0.1:8080
data: * Running on http://172.18.0.10:8080
data:[33mPress CTRL+C to quit[0m
data:192.168.122.1 - - [07/Jan/2025 04:25:41] "[33mGET / HTTP/1.1[0m" 404 -
data:192.168.122.1 - - [07/Jan/2025 08:17:33] "POST /history HTTP/1.1" 200 -

```

3.3 Gantt Chart and Milestone Chart



3.4 Earned Value Analysis



4. Software Process Improvement (SPI)

4.1 Process Improvement Objectives

To improve the development efficiency, software quality and team collaboration level of the intelligent curtain wall system architecture operation and maintenance and dataset management platform, the project adopts a continuous software process improvement strategy.

4.1.1 Main Objectives

- **Improve on-time delivery rate:** Increase the on-time delivery rate of projects from the current level to over 90%.
- **Reduce rework rate:** Reduce the number of rework caused by unclear requirements and design changes by 50%.
- **Improve quality control efficiency:** Establish a more complete Quality Assurance system to reduce the cost of post-defect repair.
- **Enhance team collaboration effectiveness:** Optimize XFN communication mechanisms to improve overall development efficiency.

#### 4.1.2 Specific Indicators

- Schedule Variance is controlled within  $\pm 10\%$ .
- The response time for requirement changes is reduced to within 48 hours.
- 30% reduction in critical defects found during test phase.
- Team member satisfaction increased to over 80%.
- Establish a standardized and measurable software process system to promote team knowledge accumulation and capacity enhancement.

### 4.2 Process Improvement Method

#### 4.2.1 Improvement of Request Management Process

- **Improvement Strategy:**
  - Establish a Change Control Board (CCB)
  - Implement demand traceability matrix management
  - Introduce User Story and acceptance criteria
- **Specific Measures:**
  - Develop a standardized requirement document template
  - Establish a demand change impact assessment process
  - Hold regular demand review meetings

#### 4.2.2 Standardize the Design Process

- **Improvement Strategy:**
  - Establish design review checkpoints
  - Implement architecture design standardization
  - Strengthen the version control of design documents

- **Specific Measures:**
  - Develop a design review checklist
  - Establish a technical architecture decision record
  - Implement peer review mechanism

### 4.2.3 Team Management Optimization

- **Improvement Strategy:**
  - Establish a knowledge management system
  - Optimize personnel configuration and task assignment
  - Strengthen team communication and collaboration
- **Specific Measures:**
  - Establish the project Knowledge Base
  - Implement pair programming and code review
  - Hold regular team review meetings

### 4.2.4 Test Process Improvement

- **Improvement Strategy:**
  - Shift left testing, early intervention in quality control
  - Build an automated test framework
  - Improve the defect management process
- **Specific Measures:**
  - Implement Test-Driven Development (TDD)
  - Set up continuous integration/continuous deployment (CI/CD)
  - Develop test case review criteria

## 4.3 Process Improvement Examples and Results

### Improvement Example: Teamwork Efficiency Improvement

#### 1. Problems Before Improvement:

- Uneven task distribution among team members
- XFN collaboration is not coordinated
- Lack of effective knowledge sharing mechanism

#### 2. Improvement Measures:



- Implement Kanban management and workload balancing
- Establish cross - team collaboration specifications
- Regularly hold technical sharing sessions

### 3. Improvement Effect:

- Teamwork efficiency increased by 25%
- 60% reduction in XFN communication issues
- The overall skill level of the team members has been improved

## 4.4 Follow-up Improvement Direction

### 1. Improve the quality measurement system

- Set up automated code quality checks
- Implement a defect prevention mechanism

### 2. Optimize project monitoring mechanism

- Implement a real-time progress tracking tool
- Establish a risk early warning system

### 3. Establish organizational process assets

- Form a standardized process template library
- Build a Best Practice Knowledge Base
- Conduct compliance audits of the implementation process

## 5. Project Risk Management

### 5.1 Risk Identification

Risk Type	Risk Item	Possible Reasons	Impact level (H/M/L)
Technical risks	Docker environment inconsistency	Version difference between development and production environment	H
	GPU resources are unavailable	Remote GPU allocation failed or overloaded	M
	Cloud storage API changes	Alibaba Cloud OSS SDK incompatible update	H

Operational risk	CI/CD pipeline failure	GitHub downtime or YAML configuration error	M
	Document obsolescence	Manual updates lag behind code changes	L
Security risk	Unauthenticated API access	Missing JWT/OAuth mechanism	H
	GitHub key leakage	Accidentally submitted the .env file to the repository	H
Managing economic risks	Cloud costs skyrocket	Supplier price adjustment or resource usage exceeds expectations	M
	Cost estimation deviation	The COCOMO model does not cover the reusability of AI modules	L

## 5.2 Risk Supervision Mechanism

### 5.2.1 Technical Risk Monitoring

- **Container and Mirroring:**
  - **Daily scan:** Integrate Trivy into the CI/CD pipeline to block the mirroring build containing CVE vulnerabilities.
  - **Version solidification check:** passed `pip freeze > requirements.txt` and `Dockerfile` hash check to ensure environmental consistency.
- **GPU Resources:**
  - **Real-time dashboard:** Prometheus monitors GPU utilization, and triggers an alarm when it exceeds 90%.
  - **Heartbeat detection:** Test the availability of the remote GPU API every 5 minutes.
- **Cloud as a Service API:**
  - **Subscription change notice:** Alibaba Cloud API updates automatically trigger test sandbox verification.

### 5.2.2 Operational Risk Monitoring

- **CI/CD Pipeline:**
  - **Dual-link redundancy:** Configure both GitHub Actions and local Jenkins for primary standby switching.
  - **YAML syntax check:** pre-commit hook validates the configuration file.

- **Document Synchronization:**

- **Automated trigger:** Swagger generates API documentation in real time based on code comments and synchronizes it to MkDocs when merged into the Git branch.

## 5.2.3 Security Risk Monitoring

- **API Security Auditing:**

- **Penetration testing:** Scan REST endpoints monthly using OWASP ZAP.
- **Token valid period:** JWT expires in 24 hours, and abnormal token usage is logged.

- **Sensitive Information Protection:**

- **Git pre-detection:** `git-secrets` prevents commits containing `AK/SK`, and historical records are scanned with TruffleHog.

## 5.2.4 Cost and Compliance Monitoring

- **Cloud Cost Alert:**

- **Threshold control:** Alibaba Cloud bill API is connected to the self-developed dashboard, and an email notification will be sent when the cost exceeds 80% of the budget.

- **Model Calibration:**

- **Quarterly review:** Compare COCOMO estimates with actual man-hours and adjust the AI module reuse coefficient.

## 5.3 Actual Risk Handling Method

### 5.3.1 Technical Risk Response

- **Container Failure:**

- **Rollback immediately:** Kubernetes automatically rolls back to the previous stable mirror version.
- **Vulnerability fix:** Prioritize the repair of Critical-level vulnerabilities (such as CVE-2023-1234) according to the Trivy report.

- **GPU Unavailable:**

- **Degradation:** Automatically switch to CPU mode and notify the operation and maintenance staff to manually intervene in the quota.

- **OSS API Failed:**

- **Retry logic:** Retry 3 times using the Exponential Backoff algorithm and then save to local cache.

### 5.3.2 Operational Risk Response

- **CI/CD Interruption:**
  - **Manual deployment:** Trigger the emergency plan and execute the deployment script directly through SSH.
- **Document out of Date:**
  - **Urgent update:** Mark the obsolete document page and supplement it by the technical author within 48 hours.

### 5.3.3 Security Risk Response

- **Unauthorized API Access:**
  - **Temporary Block:** WAF rules block abnormal IPs and revoke leaked API keys.
- **Key Leakage:**
  - **Key rotation:** Reset all associated service keys within 1 hour, and Vault records the operation log.

### 5.3.4 Cost Risk Response

- **Cloud Cost Exceeded:**
  - **Resource shrinkage:** Automatically shut down GPU instances in non-production environments and give priority to using Spot instances.
- **Estimation Deviation:**
  - **Dynamic adjustment:** Re-calculate the AI module using the Functional Button (FP) method and add a 10% buffer budget.

## 6. Software Development and Operational Cost Estimation

### 6.1 Function Point Analysis

#### 6.1.1 Data Function Points (ILF/EIF)

Type	Logical File	Data Elements (DET)	Record Elements (RET)	Complexity	Weight
ILF	User Data	username, password, role, email, permissions	1	Low	7

ILF	Dataset Metadata	filePath, size, uploadTime, owner, status, OSS path, encryption status	1	Low	7
ILF	Operation Log	timestamp, operation, user, result, IP address, affected data volume	2	Medium	10
EIF	External User System	userID, authToken, subsystem permissions, last login time	1	Low	5

6.1.2 Transaction Function Points (EI/EO/EQ)

Type	Transaction	Data Elements (DET)	FTR	Complexity	Weight
EI	Upload Dataset	File stream, target path, user credentials, encryption flag, metadata	2	Medium	4
EI	Delete Dataset	File path, user credentials, operation reason	1	Low	3
EO	Batch Export	File list, user credentials, validity period, export format, compression options	3	High	7
EQ	Log Query	Time range, operation type, userID, result status	2	Medium	4

6.1.3 Measure Function Point Complexity

According to NESMA standards, function point complexity measurement must include the following elements:

6.1.3.1 Internal Logical File (ILF) Complexity

Logical File	DET (Data Elements)	RET	Complexity	Weight
User Data	username, password, role, email, permissions	1	Low	7
Dataset Metadata	filePath, size, uploadTime, owner, status, OSS path, encryption status	1	Low	7
Operation Log		2	Medium	10

	timestamp, operation, user, result, IP address, affected data volume			
--	---	--	--	--

### 6.1.3.2 External Interface File (EIF) Complexity

Logical File	DET (Data Elements)	RET	Complexity	Weight
External User System	userID, authToken, subsystem permissions, last login time	1	Low	5

### 6.1.3.3 External Input (EI) Complexity

Transaction	DET (Input Data Elements)	FTR	Complexity	Weight
Upload Dataset	File stream, target path, user credentials, encryption flag, metadata	2	Medium	4
Delete Dataset	File path, user credentials, operation reason	1	Low	3

### 6.1.3.4 External Output (EO) Complexity

Transaction	DET (Output Data Elements)	FTR	Complexity	Weight
Batch Export	File list, user credentials, validity period, export format, compression options	3	High	7

### 6.1.3.5 External Query (EQ) Complexity

Transaction	DET (Query Conditions)	FTR	Complexity	Weight
Log Query	Time range, operation type, userID, result status	2	Medium	4

## 6.1.4 NESMA Function Point Calculation

### 6.1.4.1 Unadjusted Function Points (UFP)



Data Function Points (29) + Transaction Function Points (18) = **47 FP**

- Data Function Points:

Type	Component	Count	Weight	Subtotal
ILF	User Data	1	7	7
ILF	Dataset Metadata	1	7	7
ILF	Operation Log	1	10	10
EIF	External User System	1	5	5
Total				29

- Transaction Function Points:

Type	Component	Count	Weight	Subtotal
EI	Upload Dataset	1	4	4
EI	Delete Dataset	1	3	3
EO	Batch Export	1	7	7
EQ	Log Query	1	4	4
Total				18

### 6.1.4.2 Adjustment Factor Calculation

Based on 14 GSC factor ratings:

No.	GSC Factor	Rating	Rationale
1	Data Communication	3	RESTful API interactions with external systems, but no real-time sync required
2	Distributed Data Processing	0	Processing data through distributed methods
3	Performance Requirements	3	Supports 100+ concurrent requests with <2s response time

4	Hardware Constraints	2	Standard cloud server configuration (4 cores, 8GB RAM), no special hardware dependencies
5	Transaction Frequency	3	500+ dataset operations daily, concentrated during working hours
6	Online Data Entry	3	100% operations via web interface
7	End-User Efficiency	3	Batch export/template download features reduce repetitive tasks
8	Online Updates	3	Real-time metadata and permission updates, but no hot-patching required
9	Complex Processing Logic	2	Basic permission checks and AES-256 encryption, no complex algorithms
10	Code Reusability	1	New development with no reusable components
11	Ease of Installation	2	Docker standard deployment, but manual OSS access key configuration required
12	Ease of Operation	3	UI optimized for non-technical users, includes operation guides
13	Multi-Site Deployment	1	Single-region deployment (East China 1)
14	Change Adaptability	2	Requirement changes require API and database schema modifications

- **Total Degree of Influence (TDI)** = 3 + 0 + 3 + 2 + 3 + 3 + 3 + 3 + 2 + 1 + 2 + 3 + 1 + 2 = **30**
- **Value Adjustment Factor (VAF)** = 0.65 + (0.01 × 30) = **0.95**
- **Adjusted Function Points (AFP)** = 47 (UFP) × 0.95 = 44.65 ≈ **45 FP**

6.1.5 Experiment Results and Analysis

Metric	Result	Description
Unadjusted FP (UFP)	47 FP	Data FP (29) + Transaction FP (18)
Adjustment Factor (VAF)	0.95	Based on 14 GSC factors (TDI=30)
Adjusted FP (AFP)	45 FP	47 × 0.95 = 44.65 (Rounded to nearest integer)
Estimated Effort	6.9 person-months	Industry average productivity of 6.5 FP/person-month (45 ÷ 6.5 ≈ 6.9)

# 7. Pricing and Pricing Strategy

## 7.1 Pricing Objectives

- Market Penetration:** Rapidly capture market share in the smart building sector through competitive pricing.
- Value Demonstration:** Reflect the system’s technical advantages in data management efficiency, automated operations, and resource optimization.
- Sustainable Profitability:** Ensure coverage of development costs (~¥580,000) and annual operational costs (¥326,400).
- Customer Segmentation:** Design differentiated solutions for clients of varying scales (SMBs/large commercial complexes).

## 7.2 Pricing Model

A hybrid pricing strategy combining three models:

Model	Billing Method	Use Case
Compute Service	Per GPU hour (¥38–120/hour)	Large-scale data analysis, model training
API Calls	Tiered pricing per request (¥0.12–0.03/call)	Third-party integrations, lightweight apps
Subscription Plans	Annual fee (¥98,000–288,000/year)	Full-feature access for mid/large clients

## 7.3 Detailed Pricing Plan

### 7.3.1 Compute Service Pricing

- Basic Compute (NVIDIA T4):**
  - ¥38/hour (¥18,240/month, 20% savings)
  - Includes 15GB OSS storage/hour.
- High-Performance Compute (A100 80G):**
  - ¥120/hour (¥96/hour academic discount for research institutions).

### 7.3.2 API Call Pricing

Tiered volume discounts:

Monthly Calls	Price (¥/call)	QPS Limit
0–500k	0.12	50
500k–2M	0.08	100
2M+	0.03	Custom

*Note: Data export API priced separately (¥0.18/call, SSL encryption included).*

### 7.3.3 Subscription Plans

Bundles include compute hours + API call quotas:

Plan	Annual Fee	Inclusions
Standard	¥98,000	200 T4 hours + 500k API calls/month
Enterprise	¥188,000	500 T4 hours + 2M API calls/month
Premium	¥288,000	200 A100 hours + custom development support

### 7.4 Differentiation Strategies

- 1. New Customer Discount:** 15% off first-year subscriptions + 10 free compute hours.
- 2. Education Discount:** 40% off basic compute for universities/research institutes.
- 3. Long-Term Incentives:** Free annual maintenance for 3-year contracts.
- 4. Peak Pricing:** +15% dynamic pricing during holiday traffic spikes.

### 7.5 Cost-Based Analysis

Cost Item	Unit Cost	Profit Margin
GPU Compute	¥22.80/hour	67%
API Processing	¥0.04/call	200%
Data Storage	¥0.007/GB	85%

*Note: Operational costs allocated per service unit; gross margin maintained at 58–62%.*

## 7.6 Price Adjustment Mechanism

1. **Quarterly Review:** Adjust based on Alibaba Cloud's official price changes ( $\leq 8\%$  fluctuation).
2. **Feature Premium:** New modules (e.g., analytics) increase plan prices by 12–15%.
3. **Price Lock:** Contractual price freeze; renewals capped at CPI inflation.

## 7.7 Competitive Benchmarking

Priced 35–40% below traditional building management systems (e.g., Johnson Controls, Siemens), while offering:

- Finer API billing granularity.
- Elastic compute leasing.
- Native compliance with China's cloud regulations.

Financial Impact:

- Breakeven Point: 7 Enterprise-tier clients or equivalent annual revenue (¥1,316,000).
- ROI Period: 2.7 years (including development costs).

# 8. Fundraising and Financing Analysis

## 8.1 Fundraising Analysis

### 8.1.1 Fundraising Strategy

To ensure the successful development and deployment of the Intelligent Curtain Wall System Architecture Operation and Dataset Management Platform, the project will adopt a hybrid financing approach, combining equity investment, debt financing, and strategic partnerships.

**Key Objectives:**

1. **Capital Injection:** Secure ¥1.2 million ( $\approx \$165,000$ ) to cover:
  - Development costs (¥580,000 for MVP development, cloud infrastructure, and testing).
  - Operational runway (¥326,400 for Year 1 maintenance, marketing, and team expansion).
2. **Risk Diversification:** Balance debt-to-equity ratio (target: 60% debt, 40% equity) to optimize cost of capital.
3. **Investor Alignment:** Partner with stakeholders from the smart-building sector (e.g., construction tech VCs, cloud service providers).

### 8.1.2 Funding Sources

Source	Amount (¥)	Terms	Purpose
Equity Financing	480,000	20% equity stake offered to angel investors/VCs. 5-year exit via acquisition or IPO.	Core development and MVP launch.
Bank Loan	600,000	5-year term loan at 5.5% interest. Collateral: Project IP and future cash flows.	Cloud infrastructure and scaling costs.
Government Grants	120,000	Matching grant from Shanghai Tech Innovation Fund (approved).	R&D for AI-driven predictive analytics.
Corporate Partnership	200,000	Strategic investment from Alibaba Cloud (in-kind credits + cash).	OSS storage and API call cost coverage.

### 8.1.3 Investor Value Proposition

- **Market Potential:**
  - Addresses a ¥3.8 billion niche in China’s smart-building SaaS sector (CAGR 12%).
  - Pilot commitments from 3 commercial real estate firms (potential ¥2.1M/year revenue).
- **ROI Metrics:**
  - Projected IRR: 15.04% (pre-tax) with breakeven by Year 3.
  - 5-year NPV: ¥424,200 (discount rate: 8%).
- **Exit Strategies:**
  - Acquisition by IoT/platform providers (e.g., Huawei Smart Building).
  - Revenue-sharing model for long-term investors.

### 8.1.4 Timeline & Milestones

- Q3 2025: Close seed round (¥480K equity + ¥600K loan).
- Q4 2025: Deploy Alibaba Cloud credits and deliver MVP.
- Q2 2026: Secure Series A (target: ¥2M) for multi-tenant expansion.

### 8.1.5 Risk Mitigation

- **Revenue Buffer:** Pre-sell subscriptions (15% discount for early adopters).



- **Cost Control:** Fixed-price contracts with cloud providers (lock in 3-year rates).
- **Contingency Plan:** Bridge financing (¥200K convertible note) if Series A delayed.

## 8.2 Pre-Financing Analysis

### 8.2.1 Calculate Project Cash Flow

#### 8.2.1.1 Cash Inflow

Item/Year	0	1	2	3	4	5
1. Cash Inflow	0	43	55	88	168	236
1.1 Operating Income	0	40	50	75	155	215
1.2 Fixed Assets Residual Value Recovery	0	0	0	5	8	13
1.3 Intangible Assets Residual Value Recovery	0	0	0	4	0	3
1.4 Working Capital Recovery	0	3	5	4	5	5

#### 8.2.1.2 Cash Outflow

Item/Year	0	1	2	3	4	5
2. Cash Outflow	67.15	31.24	33.18	46.36	41	57.06
2.1 Constructio	58	9	4	4	3	4

n Investment						
2.2 Working Capital	5	3	3	4	5	7
2.3 Operating Costs	0	17	23	35	30	42
2.4 Maintenanc e Investment	2	1.5	2.6	2.8	2.6	3.2
2.5 Operating Taxes	2.15	0.74	0.58	0.56	0.4	0.86

## 8.2.2 Calculate Net Cash Flow

Item/Year	0	1	2	3	4	5
Net Cash Flow	-67.15	11.76	21.82	41.64	127	178.94
Cumulative Net Cash Flow	-67.15	-55.39	-33.57	8.07	135.07	314.01

## 8.2.3 Calculate Present Value of Net Cash Flow

Discount Rate: 10%

Present Value Factor =  $1 / (1 + \text{Discount Rate})^n$

Item/Year	0	1	2	3	4	5
Present Value Factor	1	0.91	0.83	0.75	0.68	0.62
Net Cash Flow (Present Value)	-67.15	10.7016	18.1106	31.23	86.36	110.9428

Cumulative Net Cash Flow (Present Value)	-67.15	-56.4484	-38.3378	-7.1078	79.2522	190.195
--	--------	----------	----------	---------	---------	---------

### 8.2.4 Complete Data Statistics

According to the experimental steps, the cumulative net present value was calculated.

### 8.2.5 Calculate Dynamic Payback Period

The year when cumulative present value turns from negative to positive: **Year 3**

**Dynamic Payback Period Formula:**

DPP = (Year when cumulative present value turns positive - 1) + (Absolute value of last year's cumulative present value / Current year's discounted net cash flow)

- Dynamic Payback Period DPP = **3.08 years**
- Dynamic Payback Period DPP = **36.96 months**

## 8.3 Post-Financing Analysis

Based on the pre-financing evaluation results (NPV = ¥424,200, IRR = 15.04%, payback period = 4.62 years), the project demonstrates acceptable profitability and meets the minimum financial benchmarks. Therefore, we proceed with post-financing analysis to assess the impact of capital structure (debt and equity) on project viability.

### 8.3.1 Debt Repayment Capacity

- **Loan Details:**
  - Principal: ¥600,000 (5-year term loan)
  - Annual interest expense: ¥33,000 (implied interest rate ≈ 5.5%)
- **Repayment Schedule:**
  - The loan is fully repaid by Year 5 (see *Auxiliary Statement 3*).
  - Annual operating cash flows (Year 1-5: ¥-969,200 to ¥1,636,400) comfortably cover interest payments.
  - Debt Service Coverage Ratio (DSCR):
    - Year 3: Operating cash flow (¥353,800) / Debt service (¥33,000) = 10.7x (safe margin).

### 8.3.2 Return on Equity (ROE)

- **Equity Investment:** ¥120,000 (capital contribution).

- **Net Profit After Tax & Interest:**
  - Year 5: Operating income (¥2,090,000) - Costs (¥716,700) - Taxes (¥137,900) - Interest (¥0) = ¥1,235,400.
- **ROE:** ¥1,235,400 / ¥120,000 ≈ 1029% (cumulative over 5 years), demonstrating high investor returns.

### 8.3.3 Leverage Impact on Financial Metrics

- NPV with Financing Costs: Remains positive (¥424,200), confirming debt does not erode value.
- IRR (15.04%) exceeds the loan interest rate (5.5%), indicating favorable leverage.

### 8.3.4 Sensitivity Testing

- **Worst-Case Scenario (20% revenue decline + 2% interest rate hike):**
  - NPV drops to ~¥250,000 but stays positive.
  - The payback period extends to ~5.3 years (still acceptable).

### 8.3.5 Decision Recommendation

- **Proceed with Financing:** The project maintains robust profitability and debt repayment capacity.
- **Risk Mitigation:**
  - Maintain a cash reserve for potential revenue volatility.
  - Explore refinancing options if interest rates rise significantly.

**Conclusion:** The project is financially viable post-financing and warrants implementation.

## 9. Financial Evaluation and Results

### 9.1 Basic Project Information

Project Planning Period: 5 years

### 9.2 Auxiliary Statement 1: Project Construction Investment Table

#### 9.2.1 Appendix 1: Software Development and Maintenance Costs

Year	Number of Developers
Year 0	0

Year 1	3
Year 2	1
Year 3	1
Year 4	1
Year 5	1

Comprehensive labor cost (unit: 10,000 yuan/person-year): 30

9.2.2 Appendix 2: Equipment and Tool Purchase Costs

- Office space decoration and wiring (unit: 10,000 yuan): 0
- Computer and network equipment (unit: 10,000 yuan): 2.16
- Office furniture (unit: 10,000 yuan): 0

9.2.3 Appendix 3: Engineering Construction and Other Costs

- Project startup costs (unit: 10,000 yuan): 0
- Construction unit management fees (unit: 10,000 yuan): 0

9.2.4 Appendix 4: Basic Reserve Funds

Basic reserve funds (10% of the sum of the above three construction investments, rounded to integer) (unit: 10,000 yuan): 0

9.2.5 Project Construction Investment Table

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Construction Investment	2.16	90	30	30	30	30
Software Development and Maintenance	0	90	30	30	30	30
Equipment and Tools Purchase	2.16	0	0	0	0	0

Engineering Construction and Other Costs	0	0	0	0	0	0
Basic Reserve Funds	0	0	0	0	0	0
Annual Construction Investment	2.16	90	30	30	30	30
Cumulative Construction Investment	2.16	92.16	122.16	152.16	182.16	212.16

### 9.3 Auxiliary Statement 2: Working Capital Estimation Table

- Initial working capital (unit: 10,000 yuan): 64
- Annual increase in working capital (unit: 10,000 yuan): 1.6

Working Capital Estimation Table:

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Current Assets	64	64	65.6	67.2	68.8	70.4
Cash	0	0	0	0	0	0
Accounts Receivable	0	0	0	0	0	0
Current Liabilities	0	0	0	0	0	0
Accounts Payable	0	0	0	0	0	0
Working Capital	64	64	65.6	67.2	68.8	70.4
Annual Increase in	64	0	1.6	1.6	1.6	1.6



Working Capital						
-----------------	--	--	--	--	--	--

9.4 Auxiliary Statement 3: Fund Usage Plan and Financing Table

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Total Investment	66.16	90	31.6	31.6	31.6	31.6
Construction Investment	2.16	90	30	30	30	30
Working Capital	64	64	65.6	67.2	68.8	70.4
Financing Sources	72	0	0	0	0	0
Capital	12	0	0	0	0	0
Capital Balance	12	12	12	12	12	12
Loan (5-year term)	60	0	0	0	0	0
Loan Balance	60	60	60	60	60	0
Cumulative Total Investment	66.16	156.16	187.76	219.36	250.96	282.56

9.5 Auxiliary Statement 4: Total Cost Estimation Table

- Annual employee base salary (unit: 10,000 yuan): 6
- Social insurance and housing fund as % of base salary: 20%

9.5.1 Appendix 1: Marketing Staff Salary and Benefits

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
	0	3	1	1	1	1

Annual Employees						
Annual Base Salary	0	18	6	6	6	6
Annual Social Insurance	0	3.6	1.2	1.2	1.2	1.2

### 9.5.2 Appendix 2: Marketing Expenses

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Annual Sales Revenue	0	39	78	126	187	209
Annual Marketing Expenses	0	9.75	19.5	31.5	46.75	52.25

### 9.5.3 Appendix 3: Management Expenses

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Monthly Management Expense Standard	0	1	1	1	1	1
Management Expenses	0	12	12	12	12	12

### 9.5.4 Total Cost Estimation Table

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Salaries and Benefits	0	21.6	7.2	7.2	7.2	7.2
Marketing Expenses	0	9.75	19.5	31.5	46.75	52.25

Management Expenses	0	12	12	12	12	12
Operating Costs	0	43.35	38.7	50.7	65.95	71.45
Depreciation	0	0.22	0.22	0.22	0.22	0.22
Interest Expense	0	3.3	3.3	3.3	3.3	0
Total Costs	0	46.87	42.22	54.22	69.47	71.67

## 9.6 Auxiliary Statement 5: Operating Income, Taxes and Surcharges Table

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Operating Income	0	39	78	126	187	209
Taxes and Surcharges	0	2.57	5.15	8.32	12.34	13.79
VAT	0	2.34	4.68	7.56	11.22	12.54
VAT Output Tax	0	2.34	4.68	7.56	11.22	12.54
VAT Input Tax	0	0	0	0	0	0
Urban Maintenance Tax	0	0.16	0.33	0.53	0.79	0.88
Education Surcharge	0	0.07	0.14	0.23	0.34	0.38
Cumulative Taxes and Surcharges	0	0.23	0.47	0.76	1.12	1.25

## 9.7 Basic Statement: Project Investment Cash Flow Table

Item/Year	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Cash Inflow	0	39	78	126	187	280.48
Operating Income	0	39	78	126	187	209
Residual Value Recovery	0	0	0	0	0	1.08
Working Capital Recovery	0	0	0	0	0	70.4
Cash Outflow	66.16	135.92	75.45	90.62	109.89	116.84
Construction Investment	2.16	90	30	30	30	30
Working Capital	64	64	65.6	67.2	68.8	70.4
Operating Costs	0	43.35	38.7	50.7	65.95	71.45
Operating Taxes	0	2.57	5.15	8.32	12.34	13.79
Pre-tax Cash Flow	-66.16	-96.92	2.55	35.38	77.11	163.64
Cumulative Pre-tax Cash Flow	-66.16	-163.08	-160.53	-125.15	-48.04	115.6

## 9.8 Key Financial Indicators

- Project Financial Internal Rate of Return (IRR): **15.04%**
- Project Financial Net Present Value (i=8%) (NPV): **42.42 (10,000 yuan)**
- Static Payback Period (including construction period) (DPP): **4.62 years**

## 10. Financial Risk Management

## 10.1 Revenue Sensitivity Analysis

**Scenario:** Impact of  $\pm 20\%$  revenue fluctuation on financial metrics.

Revenue Change	NPV (¥10,000)	IRR (%)	Payback Period (Years)
-20%	18.75	10.21	5.83
-10%	30.59	12.63	5.12
Baseline	42.42	15.04	4.62
0.1	54.25	17.35	4.2
0.2	66.08	19.56	3.85

**Key Insights:**

- At 20% revenue decline, NPV remains positive (¥187,500), but IRR nears the cost of capital (8%), warranting market risk vigilance.
- Every 10% revenue drop extends payback by ~0.5 years.

## 10.2 Cost Sensitivity Analysis

**Scenario:** Impact of  $\pm 15\%$  deviation in operational/development costs.

Cost Change	NPV (¥10,000)	IRR (%)
0.15	35.17	13.41
Baseline	42.42	15.04
-15%	49.67	16.78

**Critical Risks:**

- 15% cost overrun reduces NPV by 17%. Mitigation priorities:
  - Cloud pricing volatility (e.g., Alibaba OSS adjustments).
  - Labor cost escalation (e.g., team expansion).

## 10.3 Compound Risk Simulation

**Stress Test:** 10% revenue drop + 10% cost rise + 2% interest rate hike.

--	--	--	--

Metric	Baseline	Stress Scenario	Deviation
NPV (¥10,000)	42.42	22.31	-47.40%
Dynamic Payback (Yrs)	3.08	4.25	0.38
DSCR (Year 3)	10.7x	6.2x	-42%

Action Hierarchy:

- 1. Revenue Stability: Secure long-term contracts (e.g., 3-year flagship subscriptions).
- 2. Cost Control: Leverage reserved cloud instances (20-30% savings).

10.4 Risk Mitigation Optimization

Risk Type	Countermeasures	Expected Outcome
Revenue Volatility	- Prepaid API "bundles" (e.g., ¥9,000 for 100k calls)	Lock in 30%+ revenue, reduce credit risk
Cost Overruns	- Enterprise agreements with Alibaba Cloud (12-15% volume discounts)	Annual cloud savings ¥78,000–¥97,000
Interest Rate Risk	- Replace floating rates with fixed 5.5% loans	Shield cash flow from rate hikes
Vendor Lock-in	- Allocate 10% R&D budget for multi-cloud compatibility (AWS/Azure)	Reduce single-provider dependency

10.5 Sensitivity Conclusions

- 1. **Resilience:** Project withstands single-factor shocks (20% revenue drop or 15% cost rise) but requires compound risk monitoring.
- 2. **Strategic Levers:**
  - Revenue: Stabilize via prepaid models and long-term contracts.
  - Costs: Scale-driven procurement and DevOps automation.
- 3. **Thresholds:** If revenue dips >10% + costs rise >10% sustained, activate contingency financing (equity dilution cap: 15%).

11. Conclusion

## 11.1 Project Closure

The Intelligent Curtain Wall System Architecture Operation and Dataset Management Platform project has successfully achieved its objectives within the planned timeline and budget. Key deliverables include:

- **Functional Delivery:** All core features (dataset management, project O&M, RBAC, etc.) were implemented as per scope, validated through rigorous testing (unit/integration/security tests).
- **Technical Compliance:** The system met all non-functional requirements (performance, security, scalability) with Alibaba Cloud OSS integration, Dockerized deployment, and CI/CD automation.
- **Financial Metrics:** The project demonstrated strong financial viability with a 15.04% IRR, ¥424,200 NPV, and 3.08-year dynamic payback period, exceeding baseline benchmarks.
- **Stakeholder Acceptance:** Formal sign-offs were obtained from all stakeholders, including pilot clients and Alibaba Cloud partners.

### Closing Activities:

- **Documentation Handover:** Technical manuals, API docs, and operational guides were delivered to the O&M team.
- **Resource Release:** Cloud resources were optimized, and team members were reassigned to post-launch support or new projects.
- **Lessons Learned:** A retrospective identified key improvements (e.g., earlier risk mitigation for GPU resource contention).

## 11.2 Project Summary

### Achievements:

#### 1. Technical Excellence:

- Achieved 100% environmental consistency via Docker and automated 85% of deployments through GitHub Actions.
- Reduced critical defects by 30% via TDD and CI/CD integration.

#### 2. Operational Efficiency:

- Dataset upload/download latency stayed below 3s (1GB files) under 100+ concurrent users.
- Reduced rework by 50% through agile Scrum practices and requirement traceability matrices.

#### 3. Financial Success:



- Secured ¥1.2M funding (60% debt/40% equity) with a 67% gross margin on GPU compute services.
- Pre-sold subscriptions covering 25% of Year 1 operational costs.

### Challenges and Mitigations:

- **Risk Management:** Proactively addressed Docker version drift (via Trivy scans) and API security (OWASP ZAP audits).
- **Team Coordination:** Implemented Kanban workload balancing, reducing cross-team delays by 60%.

### Future Roadmap:

- **Phase 2 Development:** Expand multi-tenant support (Q2 2026) and real-time analytics (Q4 2026).
- **Process Optimization:** Institutionalize SPI practices (e.g., automated code reviews, defect prevention).