

Project Management Plan: Software-Tracking Database Project

By Charlee Anne Rodriguez

Project Overview

This project focuses on the implementation of a software-tracking database for a higher education institution, specifically for the University of South Florida. The purpose of this system is to provide a centralized solution for tracking all software assets used throughout the institution. This includes monitoring all software types, developers, versions, licenses, departmental use, installation dates, and device associations that will be utilized with the institution. The system will be primarily used and maintained by the institution's IT department with indirect support to administrative departments and academic faculties that rely on up-to-date and compliant software systems.

Project Scope

The scope of this project will focus on the following: database functionality, user access and roles, reporting and search capabilities, user interface, integration, prototype and usability testing, and documentation and training. The users will include students and staff while the university's IT department serves as the maintenance and support for any and all concerns regarding the software. The software tracking-database will be a full mobile app with advanced AI features and integrations from external vendors and/or license validation APIs. The app will be fully monitored at all times by the IT infrastructure to ensure maintenance and updates. Data migration will also ensue to preserve information from legacy systems.

Initiation Phase

During the initiation phase, the project group will determine whether the project is feasible and worth pursuing. The phase will begin with a preliminary assessment conducted by the IT project lead and cross-functional steering community, which will include university administrators, IT managers, and faculty representatives. Key focus areas will include the current state of the software asset management, budget constraints, existing infrastructure, and staffing capabilities. To fully analyze a successful software-tracking system, there will be a benchmarking study conducted to analyze other institutions' software-tracking systems. Once all research is conducted, a feasibility report will be drafted and a prototype plan will be proposed. The initiation phase will culminate with stakeholder approval and resource allocation for the project.

Definition Phase

The definition phase will formally identify and document all project requirements. This involves gathering detailed input from end users (IT staff, faculty, and students) through surveys, interviews, and workshops. The main requirement will fall under the following categories:

1. Preconditions: Database must operate within the existing university network infrastructure and comply with institutional IT policies and FERPA regulations (IEEE, 1988)
2. Functional Requirements: The system must track all software attributes (versions, license, installation dates, user, device, department) with tagging and categorization.
3. Operational Requirements: Database must support secure user access for IT administrators and provide customizable dashboards, searchable reports, and notification alerts for license expirations.
4. Design Limitations: Integration must be compatible with legacy systems and current vendor APIs. The interface must remain lightweight to ensure mobile responsiveness and minimize server load.

Design Phase

For the design phase, the defined requirements will be turned into a blueprint for development. Design documents will include the following: System architecture diagrams, entity-relationship diagrams for database structure, wireframes and UI mockups, data flowcharts, and API integration plans. This prototype will be reviewed by stakeholders to gather feedback before development. Once approved, no major design changes will be allowed during the development phase to avoid conflicts. The design technologies will include Figma, Adobe XD, Canva, and pseudocode diagrams.

Development Phase

This phase involves setting up the technical infrastructure and initiating the actual database development. Developers will use the design documents to build the system in modular sprints using agile methodology (Butler et al., 2020). AI features and API integrations will be implemented, while IT specialists will prepare for data migration from legacy systems. Security protocols such as encryption, role-based access, least privilege principles, and multi-factor authentication will be coded into the system. A test environment will also be developed for overall internal quality assurance (IEEE, 1988). The user documentation team will also begin

writing manuals and help guides. Any third-party vendors needed for licensing and validation or cloud hosting will be onboarded and briefed on the project's requirements and goals. By the end of this phase, the software will be complete and ready for deployment testing. The technology stack for this project will include the following:

1. React for overall mobile development
2. Firebase for backend services and authentication
3. SQL for data storage
4. Azure Microsoft for cloud hosting
5. Third-Party APIs for AI and other integrations.

Implementation Phase

After internal quality assurance, the system will undergo user acceptance testing (UAT) with faculty and IT staff (IEEE, 1988). Usability testing will also be conducted with select students to ensure intuitive navigation and performance. Following successful testing, the system will be officially deployed. The IT department will provide onboarding and training sessions, distribute user documentation, and offer helpdesk support. Progress will be tracked through software compliance rates, system uptime, and user satisfaction surveys (Butler, 2020). No modifications to requirements or functionality will be permitted during this phase.

Follow-up Phase

In the final phase, the project should already be deployed and readily available for use throughout the institution's system. The focus then shifts to long-term system support and evaluation. The IT department will take over responsibility for system maintenance. To ensure continued success and relevance of the software-tracking system and databases, the following recommendations are proposed:

1. Periodic updates for compatibility with new software
2. Regular license compliance checks
3. Continuous monitoring for security vulnerabilities
4. User training for new staff or faculty
5. Helpdesk support for troubleshooting

A formal post-implementation review will be conducted every six months after launch to evaluate the overall system performance within the original objectives. Feedback will be collected and documented for future improvements.

Conclusion

The development and implementation of a software-tracking database for the University of South Florida is a strategic investment in the institution's digital infrastructure. It showcases the institution's mission to follow technological advancements to improve the overall system of the university. By following the structured six-phase project management methodology, the university can ensure that the system is thoughtfully planned, efficiently built, and effectively maintained. This database will not enhance IT oversight but also improve compliance, reduce software-related costs, and streamline academic operations.

References

- Baars, W., Harmsen, H., Kramer, R., Sesink, L., & Joris Van Zundert. (2008). Project Management Handbook.
https://textbookequity.org/Textbooks/Baars_book_project_management.pdf
- Butler, C. W., Vijayasarathy, L. R., & Roberts, N. (2020). Managing Software Development Projects for Success: Aligning Plan- and Agility-Based Approaches to Project Complexity and Project Dynamism. *Project Management Journal*, 51(3), 262–277.
<https://doi.org/10.1177/8756972819848251>
- IEEE Std 1058.1-1987 : IEEE Standard for Software Project Management Plans. (1988). IEEE.
<https://doi.org/10.1109/IEEESTD.1988.121942>