**PROJECT MANAGEME NT PLAN (SPMP)**

PRESENTED BY:  
Omer Shay

Shoham Huri

Sharon Shamay

Noam Karasso

PRESENTED TO:

Hadas Hassidim

Yam Biton

LAST UPDATE:

19.6.24

1. **PROJECT NAME & EXECUTIVE SUMMARY/PRODUCT VISION SHORT**

Our project, provides a structured platform for developers who are seeking efficient project management and seamless collaboration in document and content handling. For developers and project teams, our solution aims to centralize essential project documents like initiation documents and requirements, ensuring accessibility and adherence to professional standards. The project focuses on enhancing project quality and efficiency. Unlike Git, which lacks structured document organization and specific requirements, our project mandates the inclusion of key project documents and integrates AI for code bug detection. Our product goals include streamlining project management by offering a comprehensive solution that prioritizes organized documentation and facilitates collaboration among team members, thereby optimizing software development processes.

ראש הטופס

**2. OVERVIEW**

**2.1 Problem Statement, Target, and Objectives:**

**Problem Statement:**

In software development, disorganized codebases and fragmented documentation can severely impact team efficiency and project success. Current tools, such as Git, provide excellent version control but lack structured organization for project documents and code, leading to inefficiencies and communication breakdowns. Without a centralized and structured system for organizing code and documentation, teams struggle to maintain clarity on project requirements, track changes, and ensure that all members are aligned. This disorganization can result in duplicated efforts, overlooked errors, and difficulties in onboarding new team members. Consequently, projects may experience delays, increased costs, and reduced code quality. Our project addresses these issues by offering a platform that enforces structured organization of code and documentation. By centralizing essential project documents and providing tools to maintain organized codebases, our solution ensures that all team members have easy access to the information they need. This approach minimizes confusion, enhances collaboration, and improves overall project efficiency, leading to higher-quality software and timely project delivery.[1]

**Targets and objectives:**

Our project is aimed at software developers, project managers, and development teams who need a comprehensive project management solution. Our platform was designed specifically for teams that value organized documentation, effective collaboration, and adherence to professional standards. By centralizing essential project documents and incorporating AI-driven code quality checks, our project aims to support both small and large development teams in streamlining their workflows and improving project outcomes. We have set the following quantitative goals and objectives to measure our success: improve the organization of project documentation by 25% within the first six months of implementation, as measured by user feedback and audits; achieve a minimum of 20 positive feedbacks per month from users regarding the platform's usability and effectiveness; reduce project completion time by 15% by enhancing collaboration and documentation practices; detect and solve at least 10 bugs in the AI-driven code quality checks per week to maintain high software standards; increase user adoption rates by 30% within the first year of launch through targeted marketing and user engagement strategies; achieve a 90% user satisfaction rate within the first year, as measured by user surveys; facilitate a 20% increase in team productivity by integrating seamless communication and collaboration tools within the platform; and ensure that at least 85% of projects managed through our platform adhere to industry best practices and standards. By achieving these goals, we aim to provide a robust solution that significantly enhances project management and development processes for our users.

**2.2 Customers/Potential Persona-Users:**

**Project Manager:**Alex, age 33, lives in Beer Sheva, Project Manager in the DELL company with over a decade of experience in project management. DELL is a technology company that develops, sells, and supports computers. As a project manager, Alex is responsible for overseeing multiple software development projects, ensuring they are completed on time, within scope, and meet quality standards. His day-to-day duties include planning project timelines, allocating resources, setting milestones, and ensuring effective communication among team members. Alex uses our website to streamline project supervision and enhance team productivity. Alex creates projects and stores all related documents in a central repository. This ensures that all team members have access to the most recent and relevant information, reducing the risk of miscommunication and outdated data. He assigns specific editing and viewing permissions to team members, ensuring that sensitive information is protected.

**Software developer:**Noam, 29 years old, live in Beer Sheva, Software Developer in Nvidia with 5 years of professional experience. Noam holds a Bachelor's degree in Computer Science from Ben-Gurion University of the Negev. As a software developer, Noam is responsible for writing, testing, and maintaining code for various software projects. He collaborates with other developers, project managers, and stakeholders to deliver high-quality software solutions. Noam uses the website as an integral part of his development workflow. Noam works on projects created on the site, collaborating with his team members. Depending on his permissions, he can edit, add, change, and delete content related to the projects he is involved in. Noam accesses project documents stored in the central repository, which helps him stay informed about project requirements, updates, and guidelines.

**Admin:**Sharon, 38 years old live in Beer Sheva, System Administrator with over 15 years of experience. Sharon holds a degree in Information Systems from Tel Aviv University. As a system administrator, Sharon is responsible for managing the technical and security aspects of the development environment at her workplace. Her daily tasks include configuring systems, setting user permissions and addressing technical issues. Sharon also ensures that all systems comply with organizational policies and security standards. Sharon uses the website to manage the technical and security aspects of the development environment efficiently. Sharon configures the platform to integrate with existing tools and systems, ensuring a smooth and cohesive workflow for all users. The user management features allow Sharon to set precise permissions and access controls. She ensures that each team member has the appropriate level of access to documents and tools based on their role and project involvement. Sharon oversees the security of the platform, implementing measures to protect against cyber threats and ensuring compliance with organizational and industry standards.

**2.3 Features (Main High-Level Functions), Scenarios, User Stories IDs:**

* **User management** - Allows project managers, software developers, and admin users to connect, access and disconnect from the system and thus be able to start using the system on their own.

User stories are associated with it: **Project Manager** -4,7,10,2 **Software developer –** 5,8,3,11 **Admin –** 6,9,6,35

* **Access control -** allows project managers and admin assistants to be in control of user permissions on the site and in the project, thus managing in a correct and controlled manner the activities that users can perform on the site and in the project to which the manager is associated.

User stories are associated with it: **Project Manager** – 14  
**Software developer –   
Admin –** 32,33,34

* **Project Organization** - allows project managers and admin users to create and manage the project and thus be able to control it in a good and efficient way

User stories are associated with it:  
**Project Manager** -13,15,17,18  
**Software developer –   
Admin –** 29,30

* **Project items control** - allows project managers, software developers and admins to improve, edit, update and track items that interest them in the project. That way he can be part of the project to know about changes and make them.

User stories are associated with it:  
**Project Manager** – 16,19,20,21,22  
**Software developer –** 23,24,25,26,27,28 **Admin –** 31

**Scenarios:**

**Project Manager:**Alex Thompson, a 38-year-old project manager at SoftSolutions, registers with a project management system to streamline his project management. After registering with the system and logging in securely through two-step matching, he creates a new project called "SmartScheduler Development", enters all the private projects and uploads the necessary documents to the central repository, ensuring that the team has access to the most up-to-date information. Alex invites team members and assigns them customized viewing and editing privileges to ensure adequate information and access control. Sends him automatic notifications about completion dates and upcoming tasks, and helps him track the progress of the project through manual reviews of the status of the tasks and documents. In case of outdated or incorrect content, Alex edits or deletes it to keep the project relevant and up-to-date. At the end of the working day, he performs a secure exit from the system to protect the account and project data without a user system.

**Software Developer:**As a software developer, Noam registers a new account on the project management system and securely logs in using multi-factor authentication to ensure account protection. Once logged in, Noam accesses the tasks assigned and begins by viewing the relevant details. He edits existing content to update and improve code and documentation, adds new content by contributing fresh code, files, or necessary documentation, and deletes outdated or incorrect content to maintain project relevance. To ensure code quality, Noam utilizes AI tools for efficient bug detection and fixing before deployment. Throughout the process, Noam receives notifications about any changes and updates in the projects, staying informed of the project's progress. Finally, Noam logs out securely to protect his account and project data when not using the system.

**Admin:**Sharon, the Admin, starts the day by securely logging into the project management system using multi-factor authentication, ensuring account protection. First, Sharon retrieves updated information on the number of user accounts and projects to monitor usage. Mid-morning, Sharon creates a new project, responding to a request from a project manager by filling in necessary details and adding initial team members. Next, Sharon updates permissions for new team members, assigning appropriate access levels to maintain security. In the afternoon, Sharon blocks a user account flagged for policy violations and restores a critical document accidentally deleted by a project manager. Later, Sharon deletes an outdated project to keep the system organized and lifts a block on a user account after confirming the issue has been resolved. At the end of the day, Sharon securely logs out of the system to protect the admin account and configurations, ensuring the integrity and security of the project management environment.

**2.4 Existing Alternatives, Competitors:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Our Project** | **GitHub** | **Jira** |
| Purpose | Structured project management and document handling | Version control and collaboration | Issue and project tracking |
| Centralized Document Repository | Yes | Limited | Limited |
| Code Version Control | Yes | Yes | No |
| AI-Driven Code Quality Checks | Yes | Limited (GitHub Copilot) | No |
| Project Management Tools | Yes | Basic | Advanced |
| Issue Tracking | Basic | Basic | Advanced |
| User Role Management | Yes | Yes | Yes |
| Permissions Management | Advanced | Yes | Advanced |
| Collaboration Tools | Integrated communication and collaboration features | Pull requests, comments | Advanced (comments, notifications, integrations) |
| Integration with Other Tools | Customizable | Extensive (GitHub Marketplace) | Extensive (Atlassian suite, third-party tools) |
| User Satisfaction | Target: 90% in the first year | Generally high | Generally high |
| Cost | To be determined | Free for basic, paid for advanced features | Paid (various pricing plans) |
| Target Users | Developers, project managers, and system administrators | Developers, project managers | Project managers, teams, and enterprises |

**2.5 Expected Benefits - Why Should You Buy/Use This Product?**

* Improved Documentation Management: Centralizes all essential documents, ensuring easy access and organization.
* Enhanced Team Collaboration: Real-time collaboration tools facilitate seamless teamwork and communication.
* Increased Code Quality: AI-driven code quality checks help detect and resolve bugs efficiently.
* Streamlined Project Management: Integrates task and project management features, reducing administrative overhead and enhancing productivity.
* Consistency and Standards: Provides standardized documentation templates, ensuring adherence to professional standards and reducing errors.

**2.6 Constraints:**

Our project faces several constraints that significantly influence our approach and methodology. These constraints are primarily driven by the limitations of time, resources, and our current level of expertise.

* Utilization of ChatGPT's API:

Given that our project is part of our academic studies, we are constrained by the lack of time and resources to develop a sophisticated language model from scratch. Therefore, we have chosen to leverage the capabilities of ChatGPT's API. This allows us to incorporate advanced AI-driven features into our project without the need for extensive development time and computational resources. By using ChatGPT's API, we can focus on integrating AI functionalities effectively within the scope of our project requirements.

* Integration with GitHub API:

Similarly, due to our project's academic nature and the associated time constraints, we are utilizing the GitHub API rather than developing a custom version control system. The GitHub API provides robust and reliable version control capabilities, enabling us to manage our project's codebase efficiently. This choice allows us to ensure that our project maintains high standards of code organization and collaboration, essential for the success of our project management solution.

* Team Composition and Learning Curve:

Our team comprises four third-year software engineering students. As such, we are still in the process of acquiring professional knowledge and expertise. This necessitates a significant amount of self-learning to understand and implement the various components of our project. The learning curve associated with new technologies and methodologies presents a challenge, but it also offers a valuable opportunity for us to expand our skill sets and gain practical experience in project implementation.

By acknowledging these constraints, we are able to strategically plan our project, ensuring that we make the best use of available tools and resources while continuing to develop our professional competencies.

**3. ORGANIZATION**

**3.1 Internal Structure:**

The internal structure of our project is designed to support efficient development, management and improvement of the platform. The members of the group are:

Noam - has the ability to quickly adapt, collaborates in a group and is creative.

Shoham - responsible, with high self-learning and cooperation.

Sharon - has empathy for team members, self-learning and cooperation.

Omer - creative, good at solving problems and communicative.

The four of us are third-year software engineering students, each team member brings his educational contribution in order for the project to turn out in the best possible way. Thanks to teamwork and time management, the project is organized into several key functional areas, each of which has specific areas of responsibility:

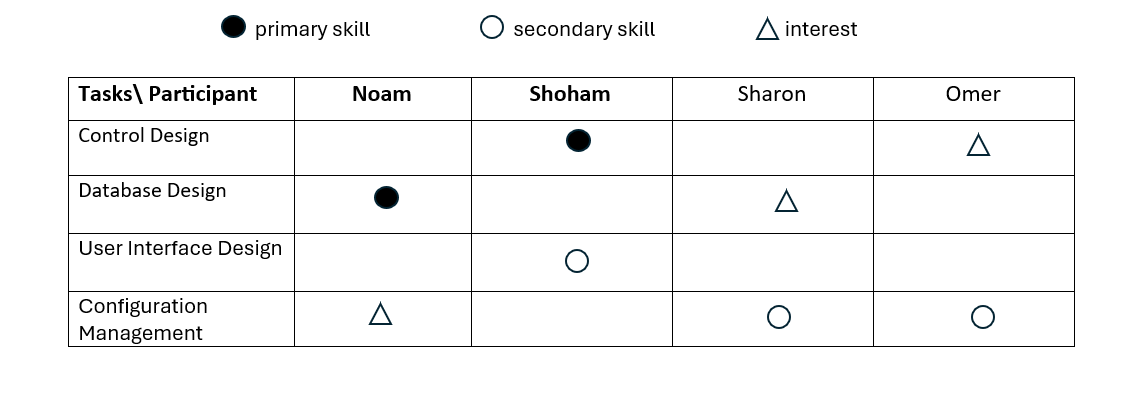
* Core development team (Noam, Shoham, Omer): This team consists of software developers who are responsible for the coding, feature development and maintenance of the platform. They work on implementing new functions, fixing bugs and ensuring the overall robustness of the system.
* Project management team (Sharon, Shoham, Omer): This team handles the organizational aspects of the project. They are responsible for project planning, managing the timeline, allocating resources, and ensuring that all necessary project documents are properly created, maintained, and accessible. They coordinate between different teams to keep the project on track.
* Quality Assurance (QA) team (Noam, Shoham, Sharon): The QA team focuses on testing the platform to identify and resolve bugs and issues. They perform various types of testing, including unit testing, integration testing, and user acceptance testing, to ensure the platform meets high standards of quality and reliability.
* Artificial Intelligence Integration Team: This team specializes in the development and integration of artificial intelligence functionality within our project. Their primary focus is on implementing and refining AI-driven features such as code bug detection, automatic document organization, and other smart tools that enhance the platform's capabilities.

**3.2 External Interfaces:**

The platform integrates with popular version control systems like GitHub, enabling users to synchronize their code repositories effortlessly. Furthermore, our project integrates with ChatGPT's API to enhance collaboration and support within the platform. By leveraging ChatGPT, users can access advanced conversational AI capabilities for tasks such as generating documentation, providing coding assistance. This integration allows team members to query the AI for quick solutions to common problems, receive automated responses to frequently asked questions, and improve overall productivity through intelligent automation. To ensure robust data management and storage, our project interfaces with various external database system SQL Databases: The platform is compatible with major SQL databases - MySQL. This allows users to store, retrieve, and manage relational data efficiently. The integration supports complex queries, transactions, and data integrity features that are essential for maintaining reliable and scalable databases.

This enhanced description effectively combines the project's core objectives with the details of the external interfaces, particularly the database integrations, to present a comprehensive overview of the platform's capabilities.

**3.3 Role & Responsibilities:**

**4. DEVELOPMENT ENVIRONMENT**

**4.1 Development Environment & Tools:**

* Programming Language: Python

Python will be the primary programming language due to its simplicity and extensive support for AI and machine learning libraries.

* IDE: Visual Studio Code

Visual Studio Code will be used for coding, offering powerful extensions and tools for Python development.

* AI Model: OpenAI's ChatGPT API

The ChatGPT API will be employed for the core functionality of code completion and conversion, leveraging its advanced natural language processing capabilities.

* Frameworks and Libraries:

Flask/Django: For developing the web interface or any backend services.

* Requests: For handling API calls to the ChatGPT API.
* Version Control: Git and GitHub

Git will be used for version control, with GitHub hosting the repository for collaboration and project management.

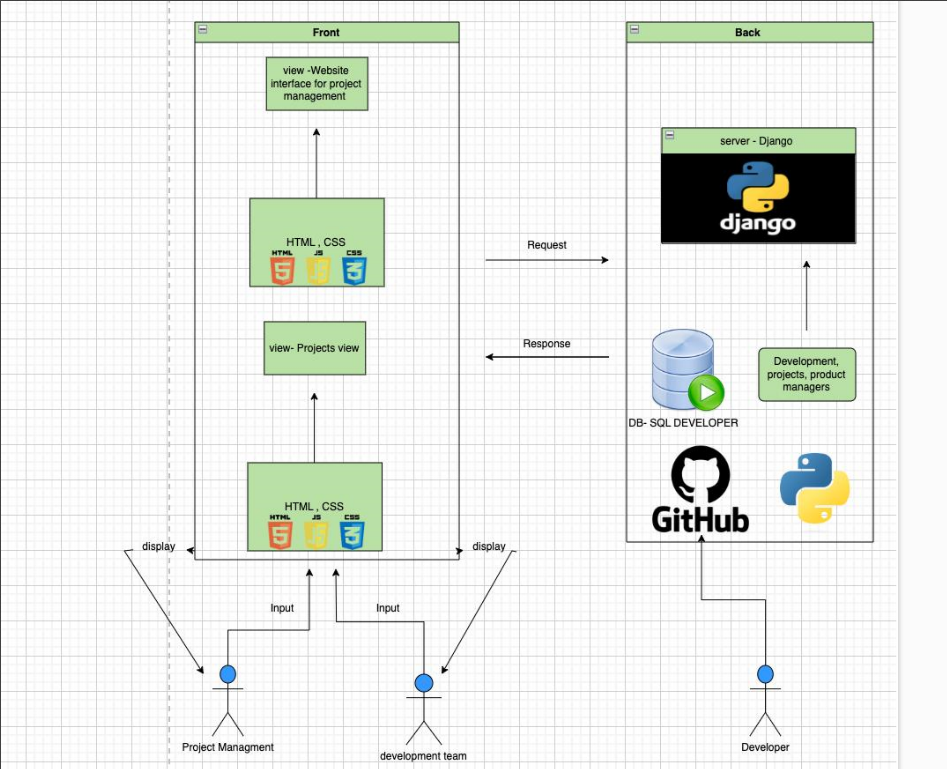
* Project Management: Jira

Jira will be used for task management, sprint planning, and tracking progress.

* Testing Tools: pytest.

pytest will be used for writing and running unit tests to ensure code quality and reliability.

**4.2 Architecture Diagram**

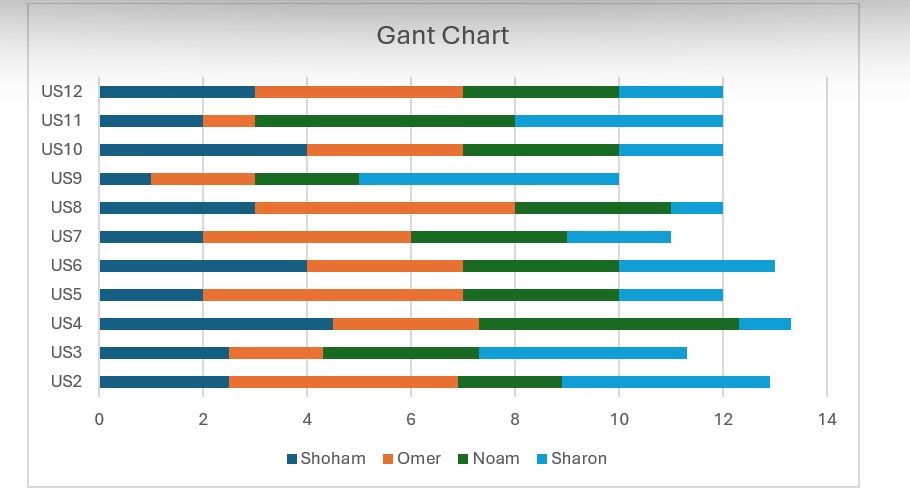


**5. WORK PLAN**

**5.1 WBS presented in a table or a tree**

|  |  |
| --- | --- |
| **Use Case** | **Work Breakdown Structure** |
| **Centralized Document Repository** | 1. Implement centralized document repository |
|  | 1.1 Develop user interface for document management |
|  | 1.2 Implement document version control |
|  | 1.3 Set up secure access and permissions |
| **AI-Driven Code Quality Checks** | 2. Develop AI-driven code quality checks |
|  | 2.1 Integrate AI algorithms for bug detection |
|  | 2.2 Implement real-time code analysis |
|  | 2.3 Develop reporting and feedback system for detected issues |
| **Project Management Tools** | 3. Implement project management tools |
|  | 3.1 Develop user interface for project creation and management |
|  | 3.2 Integrate task assignment and tracking |
|  | 3.3 Implement milestone and deadline tracking |
| **User Role and Permissions Management** | 4. Develop user role and permissions management |
|  | 4.1 Implement user authentication and role assignment |
|  | 4.2 Set up detailed permissions settings |
|  | 4.3 Develop administrative interface for managing roles and permissions |
| **Collaboration Tools** | 5. Integrate collaboration tools |
|  | 5.1 Develop real-time communication features (alerts) |
|  | 5.2 Implement file sharing and collaborative editing |
|  | 5.3 Integrate notifications and activity tracking |
| **Integration with Other Tools** | 6. Implement integrations with existing tools |
|  | 6.1 Develop APIs for integration with version control systems (e.g., GitHub) |
|  | 6.2 Integrate with project management tools (e.g., Jira) |
|  | 6.3 Ensure compatibility with popular IDEs and development environments |
| **User Feedback and Continuous Improvement** | 7. Implement user feedback system |
|  | 7.1 Develop user feedback collection interface |
|  | 7.2 Analyze feedback for continuous improvement |
|  | 7.3 Implement regular updates based on user feedback |

**5.2 Gant (based on the WBS) a high-level plan (Only for sprint 1)**:



**5.3 Detail work assignment for sprint 1:**

Team manager - Noam

frontend engineering - Omer, Shoham

backend engineering – Sharon

Planning Sprint 1:

Full realization of user stories 2-12:

1. Registration interfaces for developer and project manager

2. Login interfaces for all users

3. Creating validations

4. Logout interface for all users

5. Connecting and setting up a database

6. Creating a home page

**5.4 burndown chart:**

**6. RISK MANAGEMENT PLAN**

**6.1 Classification Strategy**

Our project's risk management strategy focuses on proactive identification, assessment, and mitigation of potential risks to ensure successful completion. High critical risks include internal risks like team member absence due to military reserve duty, technical risks associated with integrating new technologies such as ChatGPT API and GitHub API, and operational risks involving project timeline delays from misaligned schedules. To mitigate these, we implement cross-training, thorough research, pilot testing, staggered work schedules, and regular synchronization meetings. We also address execution risks like scope creep with strict change control procedures and scope reviews, and post-project risks from regulatory changes by monitoring updates and engaging legal counsel. This comprehensive approach ensures minimized impact on project objectives and timely delivery.

**6.2 Description, Probability, Severity and Mitigation**Each identified risk is described, and its probability and severity are assessed to prioritize risk management efforts, and for each identified risk, mitigation strategies are developed to minimize their impact. Below is a table format for documenting these elements:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Risk Description | Likelihood | Severity | Mitigations |
| Internal Risks | Team members called to military reserve duty, causing potential shifts in team roles. | Very likely for 2 members to be called to reserve duty | High | Cross-training team members to cover critical roles and responsibilities. Maintain clear communication channels. |
| Technical Risks | Use of new technologies like ChatGPT API, GitHub's API, and Django for the first time may affect project workflow. | Moderate likelihood | Medium | Conduct thorough research and pilot testing before full-scale integration. Provide training and support for team members. |
| Operational Risks | Project timelines and deliveries delayed due to misaligned schedules among team members with varying study commitments. | Moderate likelihood | Medium | Implement staggered work schedules and regular synchronization meetings. Prioritize task dependencies and critical path analysis. |
| Execution Risks | Scope creep leading to increased project timeline as new features and requirements are added during development. | Low likelihood | Medium | Implement strict change control procedures. Conduct regular scope reviews with stakeholders to prioritize and validate changes. |
| Post-Project Risks | Regulatory changes in the use of AI in projects requiring adaptation of project outcomes to comply with new regulations. | Not likely | High | Monitor regulatory updates and engage legal counsel to interpret and implement necessary changes. Plan for contingency measures and compliance audits. |

תחתית הטופס

(FROM SPRINT 2)

Refers to the design and flow of the screens.

8. SUPPORTING PROCESSES

9. SUMMARY AND RETROSPECTIVE

9.1 What are the achievements for this sprint,

compared to the planning, problems occurred

9.2 Lessons learned (end of each sprint)

9.3 Short demo (1-2 min for the sprints artifacts

from sprint 2)

8.1 Quality assurance plan (e.g., measurements, testing audits, bugs

management.)

8.2 Reviews and audits (e.g., internal with the team or course staff).

**Appendices:**

[1] Bob Violino. Contributing writer. "Software project management challenges — and how to handle them."

Publication framework: Feature article

Publication date: February 08, 2023