Analytics-Driven SEO & SEA Optimization

Project Plan

Spienzer

Fontys University of Applied Sciences Venlo

Svetoslav Stoyanov

26 February 2024

Information Page

Fontys University of Applied Sciences
School of Technology and Logistics
Post Office Box 141, 5900 AC Venlo, the Netherlands

Project Information

Type of report:	Project Plan
Student name:	Svetoslav Stoyanov
Student number:	3793222
Study:	Informatics - Software Engineering
Internship period:	February 2024 – June 2024
Company name:	Spienzer B.V.
Address:	Villafloralaan 1, 5928 SZ Venlo
Country:	the Netherlands
Company supervisor:	Roy Lenders
Supervising lecturer:	Pieter van den Hombergh
Word count:	1393
Date:	26/02/2024
Version:	1.0
Status:	Draft
Author:	Svetoslav Stoyanov

Versions:

Version	Date	Author(s)	Amendments	Status
1.0	26/02/2024	Svetoslav Stoyanov	First version	Draft

Communication:

Version	Date	To
1.0	29/02/2024	Roy Lenders, Pieter van den Hombergh

Contents

1 Project Assignment

1.1 Context

Spienzer B.V. operates at the intersection of SEO (Search Engine Optimization) and SEA (Search Engine Advertising), providing innovative tools for webpage's content workflow automation, position ranking, and advertising optimization. With a focus on leveraging AI for content generation, the company seeks to enhance the visibility and performance of client websites on search engines. Currently, Spienzer B.V. employs a small team, with the majority in the IT department, serving businesses aiming for improved Google search rankings.

1.2 Goal of the Project

The primary goal is to help website owners check which webpages need optimization first, to give them an opportunity to check what the number of visitors per webpage currently is and to compare this value with the numbers of visitors that their page has the potential to reach, or to check if this potential is currently too low and the page therefore needs even deeper optimization.

1.3 The Assignment

The assignment involves creating a robust integration system with Google Analytics to correlate various data points (Search Engine Results Page (SERP) position and search volume) with web traffic data, developing a frontend and backend analytics functionality for overall and per-page analysis, formulating a priority ranking algorithm for pages necessitating modifications and formulating an algorithm for web traffic predictions.

1.4 Initial Research Questions

- How does the position on SERP correlate with web traffic, particularly in terms of click-through rates (CTR) for different positions?

- What methodologies can be developed to prioritize webpages needing modifications based on SEO data?

- What criteria should be used to formulate a priority ranking algorithm for webpages that need optimization?

1.5 Preconditions

- Access to Spienzer's current analytics systems, software development tools, and platforms for the integration and analytics functionality.

- Effective supervision and support from the IT department.

- Workspace equipped with necessary technology and resources for successful project execution.

1.6 Approach

- **Sprint Planning:** Define user stories, tasks, and estimates for each development cycle (sprint).

- **Weekly Scrum Meetings:** Weekly meetings with a duration of at least an hour to discuss progress, identify obstacles, and adapt the plan.

- **Sprint Reviews:** Demonstrate completed work and gather feedback for improvement.

- **Sprint Retrospectives:** Evaluate the sprint, identify areas for improvement, and adapt the process for future iterations.

This Agile Scrum methodology will ensure continuous development, collaboration, and adaptation throughout the project lifecycle.

1.7 Technological Stack

The technological stack that has already been used within the Spienzer project should be leveraged, which is:

- Version control: GitLab

- Project management tool: Jira Software

- Microservices deployment, Database deployment, Cloud computing, Event-driven queueing: Google Cloud

- Backend: Python programming language

- Frontend: Vue.js framework

- Database management system: PostgreSQL

- REST (REpresentational State Transfer) API (Application Programming Interface): FastAPI and Flask frameworks

- Wireframes design tool: Figma Software

Additionally, Google Analytics API is to be utilized and UML (Unified Modeling Language) is to be used for Analysis and Design artefacts.

2 Planning

This chapter focuses on the planning of the project. It goes over important phases and dates.

2.1 Preliminary Analysis

Objective:

- Obtain a high-level understanding of the existing system, user needs, and project objectives.

Activities:

- Review existing documentation and tools related to the current analytics and SEO/SEA strategies.

- Engage with stakeholders to gather initial requirements and goals.

Outcome:

- A foundational grasp of project scope and key requirements that will guide the initial backlog creation.

2.2 Initial Backlog Creation

Objective:

- Develop an initial project backlog that outlines broad epics and user stories based on the preliminary analysis.

Activities:

- Identify major functionalities as epics.

- Draft high-level user stories that capture essential features and outcomes.

Outcome:

- An initial backlog that captures the project's scope and priorities at a high level, ready for further refinement.

2.3 Detailed Analysis

Objective:

- Deepen the understanding of specific requirements, user interactions, and system needs.

Activities:

- Conduct detailed system analysis, including further examination of current integrations and SEO/SEA performance.

- Develop comprehensive use case descriptions, diagrams, identify user groups and create detailed personas and user stories.

Outcome:

- Detailed documentation of system requirements and user needs that will refine and expand the project backlog. (Add sub-tasks to existing epics, add more epics if needed.)

2.4   Backlog Refinement and Prioritization

Objective:

- Refine the project backlog with the insights gained from the detailed analysis, prioritizing epics, and user stories.

Activities:

- Refine existing user stories and add new tasks based on the detailed analysis findings.

- Prioritize the backlog items based on their strategic importance, value to the user, and technical feasibility.

Outcome:

- A prioritized and detailed project backlog ready to guide the design and implementation phases.

2.5   Design Phase

Objective:

- Outline the system's technical architecture, data models, and user interface designs (wireframes).

Activities:

- Create ER diagrams and database schemas if needed, and other necessary design documents.

- Design the system architecture and plan the technical implementation.

- Create wireframes to design the user interface.

Outcome:

- A comprehensive set of design documents that provide a blueprint for development.

2.6 Preparation for Implementation

Objective:

- Ensure all prerequisites for development are met, including environment setup and setting up needed accounts and permissions.

Activities:

- Finalize the refined and prioritized project backlog.

- Set up development, testing, and staging environments.

- Update the project backlog as needed.

Outcome:

- The project is fully prepared for the development phase, with a clear roadmap and the necessary tools and environments in place.

2.7 Implementation Phase

Objective:

- To develop, test, and deploy the system functionalities as defined in the design and planning phases, ensuring the system meets the project goals and user requirements.

2.7.1 Experimenting and Prototyping

Activities:

- Develop initial prototypes for key features, particularly the integration with Google Analytics and the algorithms for webpage prioritization and traffic prediction.

- Experiment with different models and approaches to find the most effective solutions.

Outcome:

- Insights into the most viable solutions for analytics integration and algorithms, ready for full-scale development.

2.7.2 Development

Activities:

- Implement the functionalities defined in the project backlog, including backend and frontend development, database management, and API integration.

- Conduct continuous integration and code reviews to maintain code quality and consistency.

Outcome:

- A fully developed system with all necessary functionalities in place.

2.7.3 Testing

Activities:

- Perform unit testing to ensure individual components work as expected.

- Conduct integration testing to ensure different system components work together seamlessly.

- Execute system testing to validate the complete system's functionality, performance, and security.

- Engage in user acceptance testing (UAT) with the stakeholders to ensure the system meets user needs and expectations.

Outcome:

- A thoroughly tested system with documented test cases and results, ensuring reliability, performance, and security.

2.7.4 Deployment

Activities:

- Deploy the system to a production environment, ensuring all components are properly configured and optimized.

- Monitor the system's performance and address any immediate issues.

Outcome:

- The system is fully operational and accessible to (test) users, with ongoing monitoring in place.

2.8 Closing Phase

Objective:

- To formally close the project, ensuring all objectives have been met, gathering feedback and giving advice for future improvements, and ensuring the client has all the necessary documentation and training.

2.9 Project Documentation Summary

Before concluding the project, it is crucial to highlight the comprehensive documentation efforts undertaken to ensure clear communication and understanding of the project's scope, requirements, and specifications. Among these documents, the Software Requirements Specification (SRS) stands out as a cornerstone of our project documentation. The SRS meticulously outlines all functional and non-functional requirements, system constraints, and detailed specifications crucial for the development, implementation, and testing phases of the project. This document is instrumental in guiding the project team and stakeholders through the project's lifecycle, ensuring that all project deliverables meet the established criteria and stakeholder expectations.

2.9.1 Documentation and Handover

Activities:

- Compile comprehensive documentation, including system architecture, codebase, user manuals, and maintenance guides, as needed.

- Conduct handover sessions with Spienzer's team to ensure they are fully equipped to manage and maintain the system.

Outcome:

- Spienzer has all necessary documentation and knowledge to operate and maintain the system effectively.

2.9.2 Feedback Collection and Evaluation

Activities:

- Gather feedback from stakeholders on the system's performance and usability.

- Evaluate the project's success against initial goals and objectives.

Outcome:

- Valuable insights into the project's success and areas for future improvement.

2.9.3 Future Advice and Recommendations

Activities:

- Give specific advice for future improvement strategies, work left (if any).

Outcome:

- Spienzer's team has a better understanding of the continuation of the project, as needed.

2.9.4 Final Closure

Activities:

- Release of project resources and acknowledgment of team efforts.

Outcome:

- Official closure of the project, with all contractual and administrative tasks completed.

Event	Date
Project start	19.02.2024
Project Plan and Software Requirements Specification Document deadline	04.03.2024
Interim Report deadline	02.04.2024
Interim Presentation	Between 08.04.2024 and 26.04.2024
Final Report and Reflection Report deadline	18.06.2024
Final Presentation	Between 24.06.2024 and 12.07.2024
Project End	30.06.2024

Table 1: Important Dates

3 Project Organization

3.1 Team Members

Name	Role/Tasks
Monika Dobreva	Full-stack developer
Maurice Douben	General manager of Spienzer
Jorrit Deschaux	Backend developer intern building local NLP model
Moussa Adoum Moustapha	UI/UX design intern
Roy Lenders	CTO, Company supervisor, Shareholder
Svetoslav Stoyanov	Full-stack developer intern working on analytics functionality
Lewis Wiggins	Working on Spienzer's marketing

Table 2: Project Organization - Team Members and Roles

3.2 Communication

The project has established weekly progress meetings lasting one hour, primarily held in-person at the company's location in Venlo. When circumstances demand, meetings are conducted online via Zoom. Team members schedule additional meetings as needed, tailored to specific case requirements. Day-to-day communication during work hours occurs at the workplace, ensuring continuous and effective collaboration.

For written online communication, team members use email for formal exchanges and Google Chat for instant messaging, facilitating a seamless flow of information. The project management and organization are supported by Jira Software, serving as the central tool for Scrum practices and tracking progress.

Appendix

A. Software Requirements Specification (SRS)

The Software Requirements Specification document is detailed in this appendix. It serves as a comprehensive guide to the functional and non-functional requirements of the project, including system features, user interactions, and technical specifications necessary for successful development and deployment. This document is essential for developers, project managers, and stakeholders to ensure alignment and clarity throughout the project's lifecycle. For the sake of brevity and focus, the full SRS document is attached as a separate document. Please refer to the project repository or contact the project manager for access to the complete SRS document.