SUSTAINABILITY KNOWLEDGE AWARENESS

Introduction

The Sustainability-Aware Agile Scrum (SusAF Integration) project was developed as part of the hackathon challenge to incorporate sustainability principles into the Agile Scrum task management process. The goal was to use the Sustainability Assessment Framework (SusAF) to guide backlog prioritization based on sustainability impact. Given the strict time constraints of the hackathon, the MoSCoW method was applied to determine the most essential tasks that needed to be completed first.

This project aimed to embed sustainability awareness into Agile workflows, ensuring that development teams consider environmental, social, and economic factors when planning and executing sprints.

MoSCoW Prioritization of Tasks

To manage the scope of the hackathon project effectively, tasks were categorized using the MoSCoW method, which divides them into Must-Have, Should-Have, Could-Have, and Won't-Have priorities.

The Must-Have features included defining a clear "Definition of Done" with sustainability criteria, integrating sustainability-focused user stories into the product backlog, enabling CSV uploads for SusAF reports, and embedding sustainability in sprint planning, execution, and retrospectives.

The Should-Have features included automated sprint goals based on sustainability objectives, a self-assessment checklist for teams, comprehensive user documentation, and access to an educational knowledge hub.

The Could-Have features included a prioritization dashboard displaying backlog tasks ranked by sustainability impact, enhanced sprint planning visibility, a real-time execution phase tracker, and a sustainability dashboard offering insights on performance.

The Won't-Have category included trade-off recommendations between sustainability and other sprint goals, as well as full-scale integrations with external Agile tools like Jira or Azure DevOps due to time constraints.

Data Flow Description

Step	Process	Description
Input SusAF Token	From Susaf tool (Web Tool)	A **token** generated by the SusAF API is required. The system retrieves the token and attaches it to the request for secure API communication.
Report Sent to SusAF API	Web Tool → API Request	The system sends the uploaded report to the SusAF-to-Backlog Converter via API for processing.
Generate Agile User Stories	ChatGPT API → User Story Generation	The ChatGPT API generates user stories based on the extracted sustainability metrics.

SusAF API Processes the Report	SusAF-to-Backlog Converter	The API extracts key sustainability metrics (e.g., carbon footprint, waste reduction, energy efficiency) and converts them into actionable insights.
Convert Sustainability Metrics to Backlog Items	SusAF-to-Backlog Converter → Product Backlog	The processed user stories are added as backlog items in the Agile system for sprint planning.
Display Backlog Items on Work Item Dashboard	Work Item Dashboard	The new backlog items are displayed in the Agile dashboard for sprint planning.
Sprint Planning - Selecting User Stories	Sprint Planning	Developers select sustainability-related user stories by clicking the leaf button. The user can click the generate sustainability sprint goal button and get AI based suggestions on goals for the current sprint that align with the sustainability criteria.
Sprint Execution - Tracking Progress	Sprint Execution	The system continuously fetches updated sustainability metrics to track progress.
Sprint Retrospective - Evaluating Progress	Sprint Retrospective	At the end of the sprint, the system evaluates sustainability progress and generates action plans against the checklist

Sustainability Knowledge Hub

Section	Details
Introduction	The **Sustainability Knowledge Hub** is designed to educate, engage, and empower users on sustainability topics such as climate change, waste management, and energy conservation. It provides a knowledge-driven community to promote eco-friendly actions.
Key Features	**Education:** Covers topics like climate change, waste management, renewable energy, circular economy, and sustainable products. **Engagement:** Interactive quizzes, challenges, and real-life case studies. **Action:** Personalized sustainability recommendations. **Community:** Users can share ideas, learn from experts, and collaborate on green initiatives.

System Requirements	✅ **Web Browser:** Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (latest versions). ✅ **Internet Connection:** Stable broadband or mobile data connection. ✅ **Device Compatibility:** Desktop, tablet, or smartphone.
How to Set Up the Project	**System Requirements** ✅ Node.js (v18 or newer) ✅ npm or yarn ✅ Git (optional but recommended)
Installation Steps	1️⃣ **Clone the Repository** sh git clone https://github.com/your-repo/sustainability-awareness.git cd sustainability-awareness 2️⃣ **Install Dependencies** sh npm install 3️⃣ **Configure the Environment** Create a .env.local file and add: ini NEXT_PUBLIC_API_URL=http://localhost:5000/api DATABASE_URL=mongodb+srv://your-db-url 4️⃣ **Start the Development Server** sh npm run dev Open your browser and go to http://localhost:3000 to view the project.
How to Deploy the Project	**Local Deployment** To deploy the project on your local machine for testing: sh npm run build npm start **Deploying to Vercel** Vercel is an easy way to deploy Next.js applications: sh vercel deploy Alternatively, you can deploy it to **Netlify or DigitalOcean** with a proper backend setup.
Troubleshooting & Common Errors	💡 **Issue:** @headlessui/react Module Not Found ✔ **Solution:** Install it manually using: sh npm install @headlessui/react --legacy-peer-deps 💡 **Issue:** Conflicting Dependency Errors with react-apexcharts ✔ **Solution:** Update packages: sh npm install apexcharts@latest react-apexcharts@latest 💡 **Issue:** Next.js Version Warning ✔ **Solution:** sh npm install next@latest

Conclusion

The integration of the SusAF framework into Agile Scrum processes represents a significant step toward sustainability-aware software development. This project successfully demonstrates how sustainability principles can be embedded into Agile workflows while maintaining efficiency and flexibility.

By using the MoSCoW prioritization method, essential sustainability-driven backlog management features were developed within the hackathon timeframe, laying the groundwork for future

refinements. As Agile methodologies continue to evolve, integrating sustainability considerations into software development will become an increasingly important industry standard