PM_07 Agile Process Management
Spring 23
Student: Huyen Tran
Assessor: Florian Grote

# GreenCommute: Empowering Sustainable Commute with Impact Visualisation

## I. Introduction

GreenCommute is a product I developed with my team during the Start Hackathon in St.Gallen, Switzerland. The product is an innovative and user-friendly website to help promote the environmental and financial benefits of adopting alternative modes of transportation for daily commutes. By providing data-driven insights, GreenCommute empowers users to make informed decisions that contribute to a greener future for our planet.

The website offers a range of features, including personalized impact reports and interactive comparison tools. Users input their daily commute details to visualize the effects of switching to sustainable transportation options, such as walking, cycling, public transportation, and carpooling. The platform showcases potential savings in CO2 emissions and fuel costs, offering a comprehensive understanding of the advantages of adopting eco-friendly commuting habits.

Key features:

- Personalized impact reports: users can see the immediate benefits of switching to alternative transportation modes, with clear data on CO2 emissions saved and reduced fuel costs. This CO2 impact can be displayed for a single commute and over the course of multiple months.
- Interactive comparison tools: GreenCommute's user-friendly interface allows users to explore various commuting options, comparing the impact of each on their environment and their wallet.
- Gamification and community for educational purposes: The website offers a wealth of information on the science behind sustainable commuting, as well as tips and tricks for making the transaction to eco-friendly transportation options.

## II. Agile process development

### 1. The project idea and user personas:

With less than 2 days ( 36 hours in total from 22-24 of March ) to develop the product, our team faced significant time constraints that required us to adopt a strategic and agile approach to all aspects of product development, including project ideation.

PM_07 Agile Process Management
Spring 23
Student: Huyen Tran
Assessor: Florian Grote

Initially, we were provided interesting datasets from sources such as Google Map API, Google Flight Emission API, Gmail API, and Electricity Maps. Our challenge was to create a product that would help individuals understand their impact on climate change while empowering them to reduce personal emissions. The solution would be judged on potential climate impact, creativity and innovation, and technical sophistication and feasibility.

Our team, consisting of four members, began by allocating one hour for individual research and idea generation. We focused on data source preferences, project idea creativity, potential impact, and technical feasibility within the next 1.5-day timeframe. After this initial brainstorming phase, we focused on a two-hour session to narrow our original ten ideas to the final product concept: Green Commute.

Green Commute emerged as the winning idea because it effectively combined the available datasets, showcased creativity and innovation, and demonstrated significant potential for climate impact. By leveraging the diverse skills of our team members and adopting a highly collaborative and agile approach to product development, we were able to transform the Green Commute concept into a fully-functioning, user-friendly, and impactful product in a remarkably short period of time.

Based on the user story map, we decided to develop the most important features of the product: The Co2 impact comparison dashboard to help users understand their personal impact on climate change and/or enable them to act to reduce their personal emissions.

a. Persona(s)

Claudia

Demographics

- 49 years old ⟶ Generation X
- Female
- Upper middle class
- Married since 1996
- 3 children, all of them moved out by now
- Lives in small town Weida in Thuringia
- Works at a public agency for employment

located in next town

- Owns a car

Motivation

- Save money → gas prices are rising
- Wants to be environmentally conscious → trade-off between comfortability, flexibility, price and time
- Health conscious (?)

Pain points

- Bad public transport
    - By car → 25 minutes
    - By bus → 60 minutes
    - By train → 80 minutes
- Cannot work from home (employer says they do not have the technical capabilities)
- Want to understand climate impact but no resources concrete data
- Likely use multiple devices to check it

**2. Team forming, roles, and responsibilities**

Our team consists of four members:

- Huyen: Product management
- Leon: Interaction design
- Marvin: Frontend development
- Johann: Backend development

To ensure efficient product development, we applied Scrum and Kanban methodologies, dividing the project into four iterations. At the beginning of each iteration, we created a Kanban board of product backlog and engaged in product planning, followed by a retrospective at the end of each planning iteration. As we are not a big team, we decided not to have a scrum master.

**3. Methods application: scrum, Kanban, Extreme programming**

a. For the Green Commute project, the team adopted the Scrum methodology, which involved creating a product backlog, sprint backlogs, and conducting sprint reviews and retrospectives. Here's a breakdown of each element as applied in the project:

Product Backlog: The product backlog is a prioritized list of features, enhancements, and bug fixes planned for the product. For Green Commute, the product backlog could have included items such as:

1. Implement CO2 emissions calculation for different transportation modes.
2. Integrate Google Maps API for distance and route calculation.
3. Develop personalized impact reports.
4. Create interactive comparison tools.
5. Design a user-friendly interface.
6. Add additional transportation options (e.g., electric cars, public transport).
7. Implement address autocompletion.
8. Display output data per week or month.

Sprint Backlog: The sprint backlog is a list of tasks selected from the product backlog to be completed during a sprint. For Green Commute, each sprint focused on different aspects of the project:

Sprint 1:

- Implement CO2 emissions calculation for different transportation modes.
- Develop a basic Python script for CO2 calculation.

Sprint 2:

- Integrate Google Maps API for distance and route calculation.
- Improve the Python script to incorporate the Google Maps API.

Sprint 3:

- Develop personalized impact reports and interactive comparison tools.
- Design and implement the user interface.

Sprint 4:

- Add additional transportation options and address auto-completion.
- Display output data per week or month.

Sprint Review: At the end of each sprint, the team demonstrates the completed work and gathers feedback from stakeholders ( Google team and the proto user - one of the developers' mothers ). For Green Commute, this involves presenting the progress on CO2 emissions calculation, Google Maps API integration, user interface design, and additional features.

Sprint Retrospective: After the sprint review, the team holds a retrospective to reflect on the sprint and identify areas for improvement. For Green Commute, this involves discussing the efficiency of task completion, the collaboration between team members, and any challenges or roadblocks encountered during the sprint. The team then uses these insights to make adjustments and improvements for the next sprint.

b. The team also incorporates the Kanban methodology, which involves using a Kanban board, setting work-in-progress (WIP) limits, applying a pull system, and focusing on continuous improvement. Here's a breakdown of each element as applied to the project:

Kanban Board: A Kanban board is a visual representation of the workflow and progress of tasks, typically divided into columns representing different stages. For Green Commute, a Kanban board could be organized as follows:

1. To Do: Tasks identified and prioritized but not yet started.
2. In Progress: Tasks currently being worked on by team members.
3. Review: Tasks that have been completed and are awaiting review or feedback from team members.
4. Done: Tasks that have been completed, reviewed, and accepted as meeting the requirements.

Work in Progress (WIP) Limits: WIP limits restrict the number of tasks that can be in progress simultaneously to maintain focus and prevent bottlenecks. For the Green Commute project, the team sets WIP limits based on our capacity and the complexity of tasks. We decide that each team member can work on a maximum of two tasks at a time.

Pull System: In a pull system, tasks are "pulled" from the backlog as team members become available rather than being "pushed" onto them. For the project, team members would start a new task from the "To Do" column when they have completed their current task(s) and are within the established WIP limits.

Continuous Improvement: Continuous improvement in Kanban involves regularly reviewing the process to identify areas for improvement and making adjustments as needed. The team holds regular meetings or retrospectives to discuss their workflow, identify bottlenecks or inefficiencies, and agrees on changes to optimize their process. This involves adjusting WIP limits, refining task prioritization, or implementing new collaboration techniques.

By incorporating these Kanban elements into the Green Commute project, the team can effectively manage tasks, track progress, and continuously improve their workflow to optimize efficiency and productivity.

c. The team also incorporates Extreme Programming (XP) practices to ensure high-quality output and effective collaboration. Here are the details on each practice as applied in the project:

User Stories:

- As a commuter, I want to see the CO2 emissions saved by choosing alternative transportation modes so that I can make informed decisions about my daily commute.
- As a user, I want to input my home and work addresses to calculate my daily commute distance and time for different transportation options.

Continuous Integration: Continuous integration involves frequently integrating code changes into a shared repository and running automated tests to catch issues early. The team uses version control Git and collaboration tools Github and CoPilot for continuous integration, ensuring that all code changes are merged regularly and tested to maintain a stable and functional codebase.

Test-Driven Development (TDD): TDD involves writing tests before writing the actual code, ensuring that the code is correct and reliable. The team creates unit tests for different components, such as CO2 emissions calculation, Google Maps API integration, and the user interface. We then write the code to pass these tests, ensuring each component functions as expected.

Pair Programming: Pair programming involves two developers working together on the same code, one writing the code and the other reviewing it in real time. In our case, the frontend and backend developers work together to develop components such as the user interface or the API integration, improving code quality and enabling faster knowledge transfer between team members.

Refactoring: Refactoring involves improving the structure and readability of code without changing its functionality. For Green Commute, the team regularly reviews their codebase to identify areas where the code could be simplified or optimized, ensuring it remains maintainable and scalable as new features are added.

Collective Code Ownership: Collective code ownership means that all team members are responsible for the entire codebase and can make changes as needed. For Green Commute, this practice encourages collaboration and knowledge sharing, allowing team members to contribute to different parts of the project and ensuring that the code remains consistent and high quality.

By incorporating these Extreme Programming practices into the Green Commute project, the team maintains a high standard of code quality, improves collaboration, and ensures that the final product is robust and reliable.

**4. Achievements**

Given our tight time constraints, each iteration ( 7 hours ) was treated as a single sprint, typically lasting 2 weeks in a traditional setting. After each iteration, we dedicated an hour to reviewing our progress, re-estimating the product backlog, and adjusting our plans for the next sprint accordingly.

By the end of the 4th iteration, we successfully achieved the MVP as we had initially envisioned. In addition, we developed a product pitch deck and presented it to the Google team.

Overall, our ability to rapidly develop the product can be attributed to our ruthless efficiency and commitment to an agile approach to product development. This allowed us to adapt and respond to changing requirements and priorities, ensuring we delivered a viable solution within the limited timeframe.

**5. Reflection, limitations, and learnings**

Even when the agile process has its clear benefit in software development, there are several weaknesses and limitations to our product that could happen.

- Incomplete requirements or misunderstood priorities: agile methodologies rely on continuous collaboration between team members and stakeholders. If the requirements are not properly communicated or prioritized are misunderstood, the project could easily fail to address key user needs or solve the intended problem.
- Insufficient time for testing and quality assurance: due to tight time constraints, there might not be enough time dedicated to thorough testing and quality assurance processes. This could lead to potential bugs or usability issues in the final product.
- Challenges in scaling: If the Green Commute project needs to scale up and accommodate a larger team or organization, the team may face challenges in adapting and integrating the chosen agile methodologies, such as Large Scale Scrum (LeSS), Nexus, Scrum at Scale, or the Scaled Agile Framework (SAFe).
-

- Lack of technical or domain expertise: If the team members lack expertise in certain technical or domain areas, this could limit the project's success, leading to suboptimal solutions or difficulty implementing more advanced features.
- Ineffective communication and collaboration: Agile methodologies rely heavily on effective communication and collaboration among team members. If the team struggles to collaborate or communicate effectively, this can hinder the project's progress and lead to delays or subpar results.

**Learnings:**

1. The importance of effective communication and collaboration in agile environments.
2. The need for flexibility and adaptability in response to changing requirements or priorities.
3. Balancing speed with quality and ensuring sufficient time is dedicated to testing and quality assurance.

## III. Conclusion

In conclusion, the Green Commute project has successfully demonstrated the power of agile methodologies in developing a solution to promote sustainable commuting habits under tight time constraints. By leveraging methods such as Scrum, Kanban, and Extreme Programming, our team of four managed to create an MVP that empowers users to visualize the environmental and financial impacts of their transportation choices.

Through effective communication, collaboration, and adaptability, we navigated the challenges of rapidly changing requirements and priorities. Although we faced potential limitations, such as scaling and ensuring sufficient testing and quality assurance, the learnings we gained from this experience will undoubtedly benefit future projects.

Green Commute has the potential to inspire a shift in the way people perceive and approach their daily commutes, contributing to a greener future for our planet. By raising awareness about the importance of sustainable commuting and providing tangible benefits for adopting eco-friendly habits, Green Commute demonstrates the power of technology and data-driven insights to drive meaningful change.

PM_07 Agile Process Management
Spring 23
Student: Huyen Tran
Assessor: Florian Grote

**References:**

(n.d.). *What is scrum?* Scrum.org. https://www.scrum.org/resources/what-is-scrum
(n.d.). *What is Kanban?* Kanbanize. https://kanbanize.com/kanban-resources/getting-started/what-is-kanban/
(n.d.). *Glossary*. Agile Alliance.org. https://www.agilealliance.org/glossary/xp/

**Documentation:**
User story map: GreenCommute - requirement engineering
Github: GreenCommute - repo
Product Pitch: GreenCommute - pitch deck
Planning: GreenCommute
Calculation: START - Google usecase
START Hackathon 2023 - Emissions for cars
Google-start23-climateimpact