Summer
2024

Clock:uno
Track Time, Lose No Minutes!

UNIVERSITY OF COLOGNE
B.Sc. Information Systems

IS Development Solution

ISD Team Project – Project Clock:uno

Presented by:
Ohad Ragolsky | Sharon Prabakaran | Mike Wallitzek | Benedikt Zwank | Mahta Omidi

Contents

1. Project Overview

In 2019, with the rise of globalisation and digitalisation, and not least because of the COVID-19 epidemic, we witnessed a drastic change in the labour market as we knew it. Companies were initially forced to place employees on unpaid leave and soon after had to quickly find alternative work environments. This was the year when the home office gained popularity.

After the end of the epidemic, many employees preferred working from home, and companies recognised the productivity benefits of combining home office and traditional office work. Employees could manage their working hours, and companies benefited from the fact that their employees could simultaneously work on different projects from anywhere, without being tied to a specific location.

This work style requires time management skills on the part of the employees and their managers, as well as accurate recording of work times. Currently, many companies monitor the working hours of their employees using traditional and ineffective methods such as documentation in Excel sheets or manual time entry into the software This causes errors in the documentation and wasted resources.

With this project, we focus on the digitalisation and automation of tracking working times. Our goal is to equip companies, freelancers, and individuals with a tool that allows them to document work times quickly and efficiently, thus improving their workflows. For instance, instead of manually entering time for each task or program worked on, this tool will automatically identify and record the time spent on each program, displaying the total time worked. Thus, without any additional effort from the employee, their entire working day will be displayed as a timeline.

The highest priority of the development team is to respond to the needs of companies and freelancers, as they represent a larger user base for the product. Therefore, several features must be prioritised by the development team, including manual and automatic entry of times, dashboards for analysis, and the ability to create and export reports and billing pages. All these features should be protected under European data security and privacy laws (GDPR).

To ensure adequate protection, we will operate a variety of security mechanisms. We will also use appropriate tools and experts in the field of cyber security and GDPR, along with accurate requirements management.

We see great potential for the system in the private and public sectors; there is also a wide range of possible applications for students and individuals.

We hope to sustainably improve the coordination processes of time tracking, time analysis, and increasing efficiency with this project, and we would like to provide an overview of how we will implement our project in the following pages.

2. Project Goals

This project deals with the design of an information system that enables advanced time tracking and productivity improvement, helping companies, freelancers, and individuals manage their time efficiently and optimise their workflows.

Tracking work hours is an effective strategy to improve productivity. It provides insight into time utilisation and enables better time management. Therefore, the goal is to make the time entry process as fast and user-friendly as possible while minimising errors associated with manual time entry. This approach saves valuable time and allows managers to analyse employees' working times to implement improvement measures within their organisation.

Another productivity improvement in managing tracked time is to be able to analyse and export reports such as time entries, project time spent, and billing statements based on time tracking to other clients. The product should therefore allow users to export these reports in formats such as PDF, Excel, or CSV making the analysis and sharing of time tracking data easier.

When developing the time tracking system, a primary goal must be to ensure the new process is more secure than the traditional manual time tracking process to protect employee and company data. Therefore, one of the goals will be to create stable and secure data architecture concepts, ensure they comply with the GDPR, and continuously test and improve them. We will collaborate with experts with knowledge in several relevant fields to achieve this.

The purpose of our project requirements analysis will be to provide a comprehensive basis for market requirements at the beginning of the project. In addition, we will prioritise the requirements so that we can plan the development of the product effectively.

Due to the project's scope, many stakeholders and needs must be identified. Therefore, the project planning will include both the establishment of the team entrusted with the development of the product and the bringing in of a group of consultants who will complete the knowledge in several relevant fields.

In addition, a well-thought-out development strategy is required that the developer team will have to work according to. Therefore, the goal is to use system development methodologies adapted to the upcoming tasks and challenges of developing and implementing a new information system.

To achieve all project goals, version 1.0 of the time tracking software Clock:uno should be launched within 10 to 14 months. The project budget of 750,000 euros must be maintained. This estimate is based on the principles set out in the schedule and cost calculation.

3. Project Requirements

The requirements analysis process is divided into 2 parts: Requirements Gathering, Requirements prioritisation according to the MoSCoW principle.

3.1 Requirements Gathering

The requirements-gathering process is divided into several parts to ensure that we collect and understand all product requirements.

First, a thorough understanding of the problem was needed. We analysed why in companies time registration and time documentation are reasons for wasting work time and resources, often leading to mistakes in payments. To this end, we analysed studies indicating the reasons for the result of manual time recording and documented them. The results of these analyses made us better understand the business problem and its causes. It also served as a basis for creating questionnaires and interviews with target groups.
Since we belong to one of our main user groups (workers and project managers), we were first able to gather ideas on how to solve any problems we documented, when we analysed the research. From the insights obtained, we created a use case diagram to enable a model-based analysis where possible and discovered missing or conflicting requirements.
In the second stage of the requirements analysis, we also analysed similar competitors to find out what they particularly value and what functions they offer.
Finally, considering the short project time, we interviewed by phone some of our main user groups (company owners, freelancers, and managers) using the questionnaires we prepared. (See table 1.1 and table 1.2) In addition, we had to put ourselves in their shoes to perform a final validation of the identified requirements with the relevant stakeholders. Let us now turn to the system requirements that the information system must fulfil to meet stakeholder needs and achieve the project goals.

After outlining the requirements of our potential clients, we will briefly explain the product design to meet these needs. This product aims to provide a solution to the business problems mentioned earlier.

For the Clock:uno project we conducted interviews and distributed questionnaires to capture the needs of business owners, company employees, freelancers, project managers, and executives. The requirements that our information system must meet to meet the needs of our targeted customers and achieve project goals are automatic or manual time recording and project and task time management. Reporting and integrations with other project management tools are also important customer requests.
We validated the captured requirements with potential customers/users to ensure all needs were properly understood and documented. Feedback and discussions were used to ensure that important aspects were not overlooked, and adjustments were made to address requirements accordingly. Companies and project managers preferred integration with project management systems as well as manual and automatic time tracking and the ability to see reports, while freelancers preferred the ability to create billing based on time logged. Security and data protection requirements were also considered. This analysis formed the basis for the

development of Clock:uno and ensured that the final product met the needs of the users and offered real added value.

3.2 Requirements prioritisation according to the MoSCoW principle

MoSCoW	Requirement	Description
Must have	Registration	User management through log-in Two-factor authentication
	User - Dashboard	Overview dashboard of all projects and their times
	User - Time Tracking	- Automatic time recording by stopwatch, usable offline - Assignment of tracked time to projects - Automatic app recognition during working hours - Scaling according to the effectiveness of billable hours
	User - Calendar	- Calendar function for planning and overview of working hours - Comment function in the calendar - Manual entry of working hours
	Project Manager - Advanced Dashboard Features	- Daily summaries for project managers of the hours worked - Progress/status display of tasks of different users - App Employee Times - User Management: Distribution of roles or permissions from default user to admin
	Usability	Intuitive and simple user interface
	Security of Data, High Performance, and Scalability	- All data stored and transmitted in encrypted form - Regular security updates and audits
Should have	Integration into Existing Systems	Integration with popular project management and productivity tools: Jira, Asana, Monday
	Privacy - Settings	Users can adjust privacy settings (e.g., what data is collected)
	Adaptability / Customization / Individually Customizable	Users can customize the interface and certain features to suit their needs
Could have	Notifications and Alarms	Real-time notifications about important events (e.g., deadlines)
	Mobile App	Mobile time tracking also via mobile phone
Won't Have	Social Media Integration	No integration with social media platforms for direct publication of working hours or project progress.
	No in-app advertising	There will be no in-app advertising to avoid disturbing the users.

4. Product Description

Now that we have mentioned the requirements of our potential clients, we will briefly explain the product designed to achieve these requirements. This product is designed to offer a solution to the business problems we mentioned earlier.

Clock:uno combines automatic as well as manual time tracking capabilities with comprehensive reporting and analytics solutions. It will be developed as a user-friendly cross-platform application available on desktop, and mobile devices and as a web application. Furthermore, it enables viewing reports and exporting them, creating billings based on tracked time, and able to integrate with popular tools.

Clock:uno ensures that no time is overlooked. It runs quietly in the background, capturing all your activities, including software usage, emails, browser tabs, and calendar events. This data is shown on a timeline next to synchronised calendar events, allowing users to see precisely how their time was spent throughout the day. All tracked data stays on the user's device, ensuring privacy and security.

Clock:uno should integrate seamlessly with various project management and productivity software such as Jira, Asana, Monday.com, and many others. Users can create time entries, view and export reports in formats such as PDF, Excel, or CSV, making it easy to analyse and share time tracking data. (See Figure 1: User Interface (UI))

5. Use Case

Based on the requirements and product description, a graphical model of the requirements was prepared as a use case diagram. (See Figure 2)

The use case diagram for Clock:uno includes two main actors: the employee and the manager (or project manager or freelancer). The manager inherits all the use cases of the employee, allowing them to perform manual and automatic time recording, access analyses, and edit task statuses and calendars. This inheritance ensures that managers can engage in basic time tracking and task management, which are essential for understanding individual and team productivity. Additionally, the manager has unique use cases, such as creating and editing projects. This involves defining project parameters, setting deadlines, and assigning tasks, which are crucial for effective project planning and execution. The project manager's tasks extend to customising the dashboard to monitor project progress and team performance effectively, enabling them to have a real-time overview and make informed decisions. The use case for creating a team extends to editing members, as well as creating an analysis report, which includes insight into the analysis. This extension is necessary because managing team composition and financial aspects is important for maintaining project efficiency and transparency. Creating analysis reports provides project managers with valuable insights into project performance, helping them identify areas for improvement and ensure project success.

6. Realisation Form and Methodology

6.1 Realisation Form

The options for implementing an information system range from purchasing standard software to using "software as a service" models to developing a system individually. Since the system represents the product offered to the customer as a subscription, developing a new system is the only logical option. This means that the product will evolve, and new updates and features will be offered.

Given the limited capital at the start and the requirements of the system, outsourcing and end-user development are out of the question. The company has no customers or users at this point. In addition, the product must be constantly updated and expanded with new features, which requires continuous development and maintenance.

In-house development is therefore the most suitable form of realisation for our start-up. This allows the product to be continuously updated and expanded with new functions, which will make our product stand out from competitors, which is crucial for the uniqueness and long-term success of the system.

To use our resources wisely and to avoid legal violations of data protection during development or using automatic time recording by users, we are supported by external experts and follow the General Data Protection Regulation (GDPR).

The system will be implemented as a cross-platform application available on desktops (Windows, macOS), mobile devices (iOS, Android), and as a web application. This approach ensures that users can access and use the software from different devices, which greatly improves flexibility and user experience. The application will operate primarily as a desktop application to take advantage of the ability to track computer activities. Mobile and web versions will focus on manual time tracking, project management, and reporting functionality.

6.2 Scrum Description

Scrum is an agile development method that aims to manage complex projects iteratively and step by step. (See Figure 3)

At the beginning, a product backlog is created, a prioritised list of requirements. The product owner is responsible for processing and prioritising this backlog. The development team then plans the next sprint in a sprint planning meeting, usually a fixed period of one to four weeks, and selects tasks from the product backlog to be completed in the sprint. This result is in Scrum commonly known as an increment, which gains in an iterative approach more and more features after each sprint eventually contributing to the final product. During the sprint, the team meets daily in the Daily Scrum, a short meeting to discuss progress and set the plan for the day. At the end of the sprint, the increment is presented to the stakeholders in a sprint review meeting to obtain feedback. Finally, the team reflects in a sprint retrospective on what went well and what can be optimised to enable improved collaboration among the team members. This cycle is repeated until the project is completed, with a focus on flexibility, communication and continuous improvement.

6.3 Roles and Responsibilities

Product Owner

The Product Owner is responsible for maximizing the value of the product. He is in contact with the stakeholders as well as the development team and always has the economic success of the product as his goal. Additionally, He takes on the viewpoint of the end customer. Unlike the customer, the Product Owner is always available and considers the feasibility of the customer's wishes. In coordination with the stakeholders, he is accountable for the appropriate requirements as well as the release management and documents them in the product backlog.

Scrum Master

He is the scrum methodology expert and supports the team and the Product Owner in optimally applying scrum as well as improving work routines, organisation and communication. He protects the team from harmful influences from the outside and helps with removing any kind of obstacles and makes himself expendable if he does a good job.

Development Team

The development team typically consists of 5-9 team cross-functional members, who work full time. There are no sub-teams, neither formally defined roles. The development teams are self-organising as well as autonomous and should work together in the same space. They are responsible for effort estimation of items in the product backlog and thus, the development of an increment in each sprint.

Stakeholders

The stakeholders are also an important part of the scrum, but they are not part of the development team. Everyone who is affected by the project counts as a stakeholder, who has a consulting role. For example, Company Leadership, Customers, and End-Users.

6.4 Scrum Events

Backlog Refinement

Backlog Refinement occurs on the first day of each sprint. The product backlog, which includes all requirements often in the form of user stories, is prioritised and treated as a living document. The Product Owner is responsible for maintaining this document. During the backlog refinement meeting, the product backlog is updated to reflect any changing requirements, ensuring that the team is always working on the most current and relevant tasks.

Sprint Planning

Sprint Planning is a critical event held at the start of each sprint, following backlog refinement. It involves the Product Owner, Scrum Master, and Development Team, who collaborate to select and transfer items from the product backlog to the sprint backlog. These items are chosen based on priority and readiness for implementation in the upcoming sprint. Importantly, only the Development Team decides how many items or story points they can feasibly commit to completing within the sprint. Once committed, items cannot be added or removed unless agreed upon collectively by the team. Discussions during Sprint Planning also focus on how the selected items will be implemented, ensuring everyone has a clear understanding of the work and the sprint goal is effectively pursued.

Daily Scrum

The Daily Scrum is a brief, daily meeting held at the same time each day, typically timeboxed to 15 minutes and conducted standing up to keep it concise. During this meeting, each team member provides an update on three main points: what they have accomplished since the last meeting, their plans for the current day, and any obstacles hindering their progress. Additionally, there's an update on the status of sprint backlog items to ensure everyone is informed about the progress towards achieving the sprint goal. This daily synchronisation fosters communication, collaboration, and transparency within the team, enabling them to adapt and adjust their plans as needed to meet their sprint objectives effectively.

Sprint Review

The Sprint Review is a pivotal event held on the last day of each sprint, serving as an informal meeting where the team showcases their work. The result, the so-called increment, typically presented in an integrated manner on a single computer, is demonstrated to stakeholders, including the Product Owner. During the review, the Product Owner evaluates whether the demonstrated work meets the acceptance criteria and decides whether to accept it. This meeting also provides an opportunity to discuss any functional or technological problems encountered during the sprint, fostering collaboration and ensuring alignment on project goals and quality standards.

Sprint Retrospective

The Sprint Retrospective, scheduled for the last day of each sprint and following the Sprint Review, is a critical meeting where the team engages in open discussions. They address issues spanning team dynamics, relationships, processes, communication, working style, tools, and the Scrum framework itself. The session is moderated by the Scrum Master, who facilitates discussions on what went well during the sprint and areas that need improvement. This retrospective provides a structured opportunity for the team to collaborate on solutions and decisions aimed at enhancing their performance and effectiveness in future sprints. This while cultivating a culture of continuous improvement and collective סownership within the team.

The development team and the scrum master are an integral part of all phases; the PO either, but his presence is in the Daily Scrum and Sprint Retrospective optionally. The presence of the stakeholders is in the Sprint Review mandatory, to confirm the requirements were met in the sprint.

6.5 Tools and Techniques Used

During our sprints, we will use various tools and techniques to support and enhance our development process. During our whole project, we will use Jira to help create user stories and as our project management tool, confluence as our knowledge database/documentation, and Microsoft Teams for communication.
Draw.io is a free Software that will be used to create diagrams and models of our system.
For the mock-ups, wireframes, and prototypes we will be using Figma as our tool with our technique being prototyping because our Project size is relatively small with a high impact on the user's side.
During our development we plan on using GitLab for our code management, VS-code as an IDE for the code, and Postman to test the APIs combined with the following techniques:

- Backlog grooming to regularly check and prioritize our Backlog.
- Automated testing to make sure that our software's functionality works right.
- Pair programming to ensure knowledge exchange.

- Continuous integration to test our integrations.

After our first initial release, we will use continuous Improvement and automated testing to ensure our software quality and functionality, along with new updates and features.

6.6 Reasoning for the usage of scrum

In the following, we will explain our decision for the scrum methodology by discussing the advantages.
Scrum is the most used methodology for system development. Thus, it is the standard that helps us to find educational sources and skilled workers more easily.
It is very important that our software works faultlessly and that we consider feedback and try to improve the mentioned aspects to keep our users satisfied with our system and to improve the quality of our software. Therefore, a high priority is to be very flexible to solve problems and to improve the suggestions quickly. Due to the quick release cycles (Sprints), Scrum is very flexible and adaptable. Thus, we can react quickly to changes in the market demand and add or modify functions quickly.
Furthermore, transparency and control are very essential. This is achieved through the artifacts and ceremonies of a scrum. By having the ceremonies, we identify and solve problems quickly. In addition, team members and specific stakeholders are well informed about the status and progress of the project, which minimises complications, and misunderstandings and keeps the stakeholders such as investors satisfied.
Moreover, by using Scrum we have clearly defined roles and duties. With the clear structure and allocation of responsibilities, we have fewer misunderstandings and in general a higher efficiency.
Additionally, the transparency and the clearly defined roles create a very good work atmosphere, which increases the motivation of the employees and therefore leads to higher work efficiency and better work results.

7. Project Timeline

The estimated timeline for our project can be divided into six different processes, each working towards distinct goals that build upon each other. These processes, along with further specifications, are outlined below. We have determined checkpoints within these processes to track our progress and make necessary adjustments, ensuring the team stays aligned and updated on the project timeline: (See Figure 3)

1. Initialization of the Project (2 Weeks)
2. Design process (4 sprints; 2 months)
3. Development process (12-16 sprints; 6-8 months)
4. Testing and refinement process (1 month; 2-3 sprints)
5. Release and Rollout (3 sprints; 3 months)
6. Maintenance- and improvement (ongoing)

Important to note is that we plan on using Scrum which is an agile Methodology. This means that we still primarily value the agile principles and manifesto such as responding to change or customer collaboration. That includes feedback gathering, adjustments of the timeline, requirements, and current progress as well as regular meetings for reflection on our progress, cooperation, coordination, and plans. This indicates that the estimated timeline is mainly used as an orientation and not as a fixed plan.

Because we have already done a lot of tasks related to the initialization of the project, we only must do user stories, interviews with potential stakeholders as well as use case and activity diagrams. The goal is to have a full system proposal and ensure that every team member understands the system's functionality and value. The results should also further improve the timeline estimation and worker expectations. The goal is to have a full system proposal and ensure that every team member understands the system's functionality and value. These results should further improve the timeline assessment and employee expectations.

Next is the design process, where we design our system, and the UI and by the end create our first clickable prototype. The goal of this process is to have a design prototype that accurately simulates user experience, without the back-end functionality, and the dashboard so that we can present our stakeholders with the future product. We will use this feedback to finalize the design prototype.

The next process is the Development process, where we implement our core functions and by the end conduct our first external test. The initial tasks include implementing user management, authentication, and dashboard functionality. We then go on to implement the automatic and manual time recording and integrate calendar functions. This is followed by the implementation of summaries, reports, and progress overviews. After this, we implement GDPR-compliant data security features and adjust our system. With everything working comes the integration of PM Systems and APIs. We then finalize our current prototype with adjustments, and an additional messaging feature between project members and prepare for the public tests in the next process.

With that, our development process comes to an end with an estimated amount of around 14 to 18 sprints or approximately 6 to 8 months. After each sprint within the processes, we will review and adjust our estimated times for the following tasks based on the current progress.

With our development process finished, we then go into our test- and refinement process, where we will use the software ourselves and conduct an external beta test. Because it takes some time for users to properly use the software, we think it will take about 1-2 months to receive good and complete feedback. After we have received and analysed the feedback, we then will take another 1 to 2 sprints to refine and optimize our system and adjust, in preparation for our final product. With a total time of about 1,5 to 2 Months for this process.

The next process is the release and rollout, involving the release of the first official version, monitoring and gathering feedback, leading to an estimation of around 3 months.

The last process is the maintenance and improvement process. This ongoing process starts right after we have fixed and updated our software to a sufficient standard, ensuring we don't have regular complaints about bugs or other software problems. This process includes ongoing customer support, regular updates, and long-term feature planning.

Summed up, our estimated timeline ends, with a total estimated minimum time of 10 months up to a maximum of 14 months for the fully developed and deployable software, with 12 months being our most likely development time, not counting the ongoing process.

8. Risk analysis

During the planning phase of our project, we identified numerous potential risks. Below, we will discuss the most significant potential errors and describe the preventive measures we have implemented to address these risks.

Data Privacy
Since our project is a large information and time recording system, a lot of data about the projects and people's work processes as well as some personal data is recorded. Therefore, data protection security measures must be sufficient so that no third party can access the information and possibly lead to data loss or data protection violation. For this reason, we must implement very good security protocols and regularly review, control and update them to ensure that data is stored and transferred correctly and that employees only have access to data that they allow and have been authorised to do so.

Integration
Because this system includes multiple functions and needs to be implemented by multiple people both during implementation and integrated with many other systems in the market in the future, we need a very high level of integration so that it does not lead to problems where inefficient integration occurs. It is also important that the interfaces are as clear as possible and that they are always very well controlled and checked to ensure that they are efficiently connected and fulfil their purposes. Since this system can also be used in large companies with many employees, it is even more important that it is well-integrated and has enough capacity and performance for significantly more users, because these aspects play a significantly larger role when it comes to larger user groups.

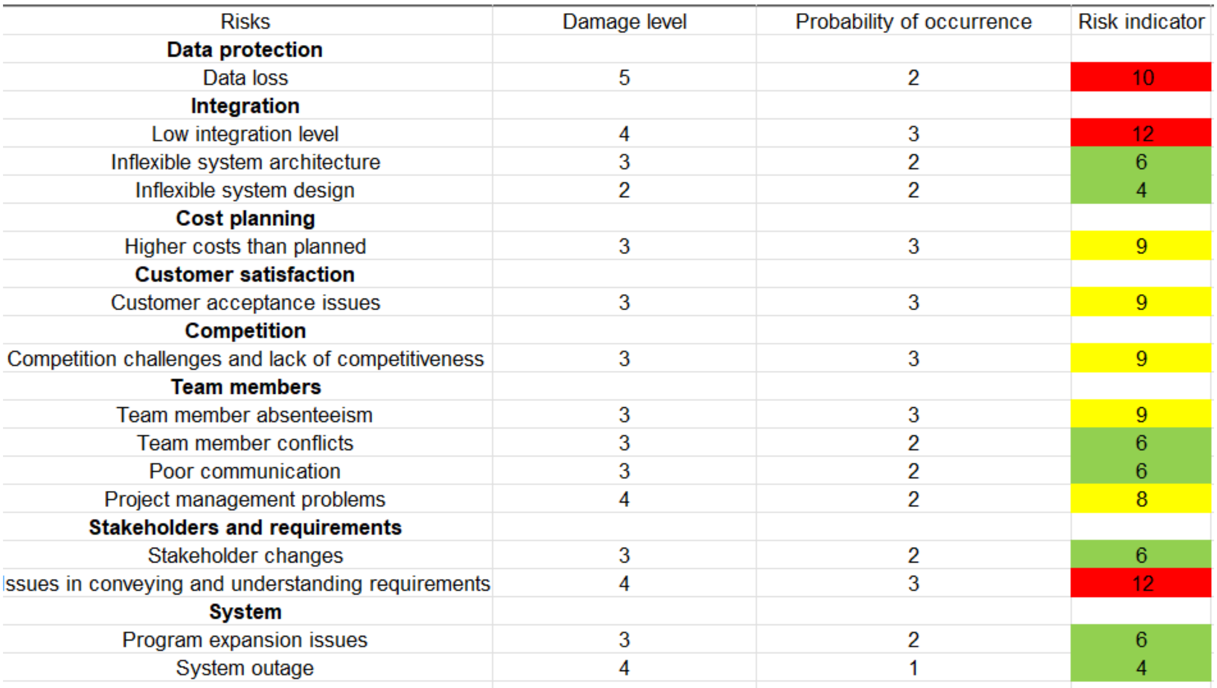
Cost Planning
A very important aspect, which is also very important, is resource and cost planning. Incorrect or incomplete resource planning can lead to huge problems in the entire process, and it could even lead to the whole project failing and making a huge loss. In any case, it must be checked whether the resources match the scope of the process, and one should also expect unplanned or unexpected costs because very often problems arise in the system that one had not expected, and resources are needed to fix these.

Customer Satisfaction
Another potential problem, which is the case with many new systems that come onto the market, is that the users, i.e. the employees of the respective companies who will work with our program in the future, prefer to use the old one You want to stick with the time recording method and don't want to completely restructure the entire process. That's why it's important, for example, before or during the implementation of the entire system, to carry out surveys among the employees of potential customers and to ask about their expectations and potential problems that they will have with the system and to also orient themselves accordingly the system will be better accepted by customers in the future due to the added value it brings. A user-friendly user interface implementation is also one of the most important criteria that you must attach great importance to so that the program is better accepted by future employees because it is clear and easy to use.

Competition

It should be noted that there are already similar systems on the market that are our potential competitors, and it is very important that our system brings customers added value that our competing products do not have. These can, for example, user-friendly design of the user interface, better, more efficient, and simpler time recording and transfer, expanded options for different roles in a company (e.g. authorization to enter, change, read, and compare data), more cost-effective subscription offers, continuous improvement and rapid resolution of problems that arise over time and a high level of integration with other programs.

For our project, we use the risk matrix tool to identify the risks of the project.
A risk matrix is a tool used in project management to systematically identify, assess, and prioritise risks. It consists of a table in which the likelihood of occurrence and the severity of impact are recorded for each risk. In our case, the likelihood of occurrence is indicated on a scale from 1 to 5, with 1 representing a low probability that the risk will occur. The severity of impact is also rated on a scale from 1 to 5, with 5 representing the most severe impact. These risk values allow the project team to prioritize risks based on their significance and devise specific mitigation strategies. The cases with the highest total risk are highlighted in red, those with the lowest total risk are marked in green, and the cases with intermediate risk levels are shown in yellow.



Risks	Damage level	Probability of occurrence	Risk indicator
Data protection			
Data loss	5	2	10
Integration			
Low integration level	4	3	12
Inflexible system architecture	3	2	6
Inflexible system design	2	2	4
Cost planning			
Higher costs than planned	3	3	9
Customer satisfaction			
Customer acceptance issues	3	3	9
Competition			
Competition challenges and lack of competitiveness	3	3	9
Team members			
Team member absenteeism	3	3	9
Team member conflicts	3	2	6
Poor communication	3	2	6
Project management problems	4	2	8
Stakeholders and requirements			
Stakeholder changes	3	2	6
Issues in conveying and understanding requirements	4	3	12
System			
Program expansion issues	3	2	6
System outage	4	1	4

9. Staffing Plan

Product Owner

As described in Section 6.3: Roles and Responsibilities, his responsibility is to communicate with the stakeholders and the development team, ensuring the (economic) success of the product is always his primary goal. (1x Project owner)

Scrum Master

The Scrum Master is an expert in Scrum Methodology as described in Section 6.3: Roles and Responsibilities. His responsibility is to support the development team and the product owner throughout the project. (1x Scrum Master)

Data Scientist

The data scientist is responsible for the design implementation and development of the database used to store the information, such as user data and working hours. (1x data scientist)

Development Team

A very important part of the said team consists of the developers who work directly on the development of the program. The team consists of a three-person **backend development** team, which is used to plan and implement the system architecture and ensure that the program works properly, and a two-person **frontend development** team, which is responsible for developing a suitable front-end design to ensure that users can make the best possible use of the program and that it is well accepted by them. (3 backend and 2 frontend developers)

UI/UX specialist

To ensure that the final product is user-friendly and best received by customers, they are directly responsible for the design of the user interface. (2x specialists)

Software Tester

The software tester is responsible for creating and executing test plans to identify program errors and inform the developers about them so that they can be fixed in the best possible way. (2x Software Testers)

external consultants

Our project requires the support of external consultants, for example in the areas of labour law and data protection regulation (GDPR). The team of consultants consisting of lawyers and data protection consultants will support the development team with questions and problems in these areas. The work of external consultants is required only on a part-time basis in this project according to the Service Level Agreements (SLAs) and Consulting contracts.

10. Cost estimation

Since we are founding the company specifically for the Clock:uno product, we must consider various elements for the cost calculation. However, we focus more on development costs and related expenses. This means costs for founding, legal matters, contracts, and support teams after the product launch are excluded. This cost calculation is an estimation based on our project requirements and cost information gathered from the web. (Links to our sources will be added)
We consider the implementation of a workspace, salaries, costs for servers and databases, and expenses incurred by software that supports development. We also estimated the costs for integration and API, cloud services and hosting, and rent. (See Table 2)
This table is a description of activities related to various elements of a project, divided into several categories:

Team
To calculate a team member's cost in the Team section, we identified the annual salary for the role, determined the number of individuals in that role, and multiplied the annual salary by the number of individuals. The total team costs amount to €618,700 annually, covering the salaries of a Product Owner, Scrum Master, Data Scientist, three Backend Developers, two Frontend Developers, two UI/UX Specialists, and two Software Testers. (See References)

External IT Service Providers
The total costs for external IT service providers are €9,560, which include expenses for first consultation, product-network implementation, and external lawyer consultation.

Hardware and Software
Total hardware costs amount to €22,604, which includes laptops, monitors, docking stations, accessories and other network components such as routers, firewalls, switches, and access points. The total software cost per year is €9,620 and covers: Jira Premium, Confluence, GitLab, Figma, MS Teams, MS 365 Business with Vodafone Services, and additional development tools and licenses.

Office and Other Costs
Office costs for one year are €70,000, covering €60,000 for the rent of a 250 sqm office and €10,000 for office equipment and furniture. The total other costs amount to $16,228, which includes expenses for integration and API costs, development and test servers, web servers, application servers, and database servers.

Total project cost: € 746.712,00

11. Stakeholder analysis

In the following, we are going to assess the stakeholders of our project. First, we must distinguish the term stakeholder. A stakeholder can be an individual, organisation or group that is connected in any way by the outcomes of the project. Stakeholders are categorised into internal and external stakeholders. Internal stakeholders are stakeholders that are connected through the inside of the organization and external are those who are affected by the outside of the organization.

Regarding the identification process of the stakeholders, we started by identifying the potential users. Our software aims to help companies, and freelancers manage their time efficiently and optimise their workflows. Hence our external stakeholders are customers.

Furthermore, we need specific staff, freelancers, or external providers to plan and implement our software. For the system development aspects, we need a product owner, scrum master and a development team, which consists of three backend developers and two frontend developers. We also hired 2 UI/UX specialists. These mentioned employees are internal stakeholders.

Moreover, to clarify and certify all data laws and their implementation and to clarify the aspects of the networks, security, and office equipment, we need a lawyer and an IT service provider. However, they are not employed by us and therefore are external stakeholders.

After the implementation, our priority should be to test and evaluate our software and maintain it constantly to ensure that all functions work and to detect and solve errors. Thus, internal stakeholders are software testers and a support team.

We then asked ourselves who else we are connected to or need to consider. We analysed similar projects and their best practices. In this process, we came up with the ethical organizations, regulatory bodies, competitors, and potential investors. These are external stakeholders.

To underline the importance of each stakeholder, we portrayed our stakeholders in a stakeholder matrix. The matrix is divided into four categories. One category is “managed closely”, which stands for high influence and high interest. Then “keep informed”, which represents the stakeholders with a low influence but a high interest. “Keep satisfied” consists of the stakeholders that have high influence and low interest. And finally, “monitor”, which represents the stakeholder with low influence and low interest. (See Figure 4)

12. Ethic challenges

Our software greatly helps companies and individuals to improve their productivity and efficiency through our time tracking features. Despite the many benefits for the user, there are ethical challenges, which we must assess and solve. Especially with our software, where a lot of data is being tracked, this could result in various ethical challenges regarding data. Particularly we are referring to the GDPR, which is the data protection law by the European Union. In the following, we will evaluate the ethical challenges and name our approaches to solving them.

The first aspects are data privacy and data security. The threats are that third parties or even internal employees have unauthorised access and could misuse the data of the user. Therefore, we're implementing end-to-end encryption to protect the data from unauthorised access (Art. 5(1)(f)). (Art. 5 GDPR)

Furthermore, the server should have a zero-knowledge encryption. By that, even the server has no access to the encryption keys and therefore no access to the users' data. As a result, only the users can access their data (Art. 5(1)(f)).

Moreover, it is important for us not to use a central database but instead to use distributed databases to save the data. Thus, the threats of data breaches are minimised, because we do not have just one point of attack. To further prevent the risk of being attacked by hackers, we could use additional methods. One method could be to use multi-layer security, which includes firewalls and intrusion detection systems, to defend against the attacks of hackers (Art. 5(1)(f)). Another method could be to execute regular security checks, penetration tests, and security updates (Art. 5(2)). The aim is to find potential threats and vulnerabilities in the system and to solve them. Ultimately, we have a plan with protocols on how we must act and react in the event of security events (Art. 5(1)(f)).
Another ethical aspect that we must consider is that users may not be aware of what data is being tracked or cannot limit or decide about the data tracking. Therefore, our priority should be transparency. We must communicate to the user which data is being tracked and if it is going to be used or not. Hence, we provide a privacy policy and terms of use for the user, which every user must accept before using our software (Art. 5(1)(a)). We are also introducing a control management option, where the user can decide about tracking features and can customise them. Besides that, the user also has the option to start and stop tracking, to manage and delete his data (Art.5(1)(c)).

The last ethical aspect is fair use and non-discrimination. Our software, especially our user interface, should be designed so that it can be used by everyone, including people with disabilities. It is important that everyone has the same conditions of use and that no one is neglected. To ensure this, we prioritise user testing. The results should help us to solve potential problems and to guarantee equal fair use.

List of Figures

Figure 1: User Interface (UI)

Figure 1.1: Clock:uno timeline of time entries

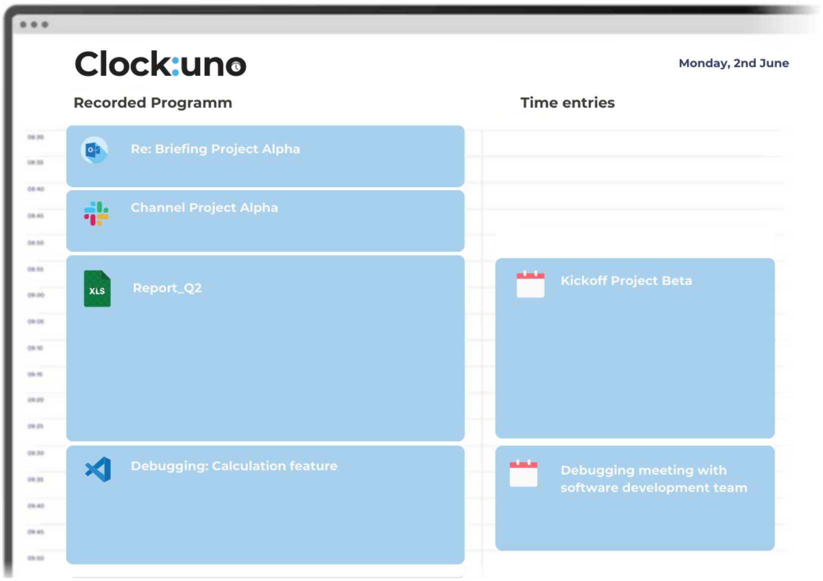
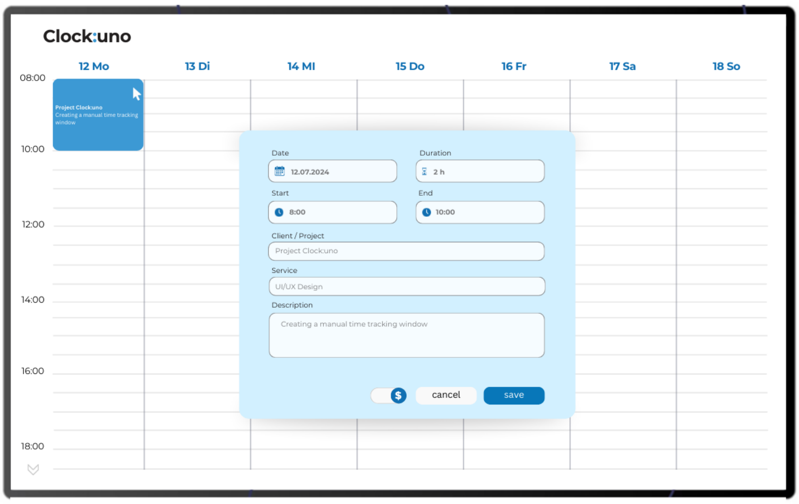


Figure 1.2: Manually time entry with the activated billing option



Date: 12.07.2024

Duration: 2 h

Start: 8:00

End: 10:00

Client / Project: Project Clock:uno

Service: UI/UX Design

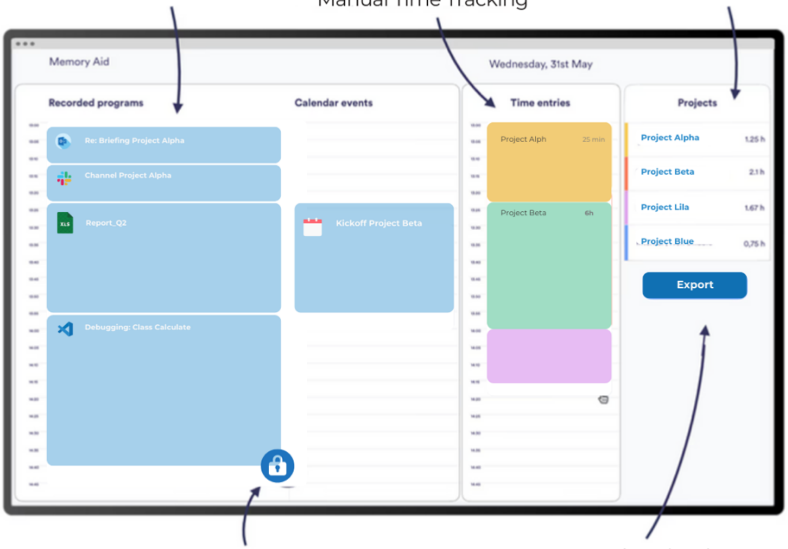
Description: Creating a manual time tracking window

Figure 1.3: Clock:uno's functions in the timeline

Automatic Time Tracking

Manual Time Tracking

Allow users to link tracked time directly to specific projects



Memory Aid

Wednesday, 31st May

Recorded programs

- Re: Briefing Project Alpha
- Channel Project Alpha
- Report_Q2
- Debugging: Class Calculate

Calendar events

- Kickoff Project Beta

Time entries

- Project Alph 25 min
- Project Beta 6h

Projects

Project Alpha	1,25 h
Project Beta	2,1 h
Project Lila	1,67 h
Project Blue	0,75 h

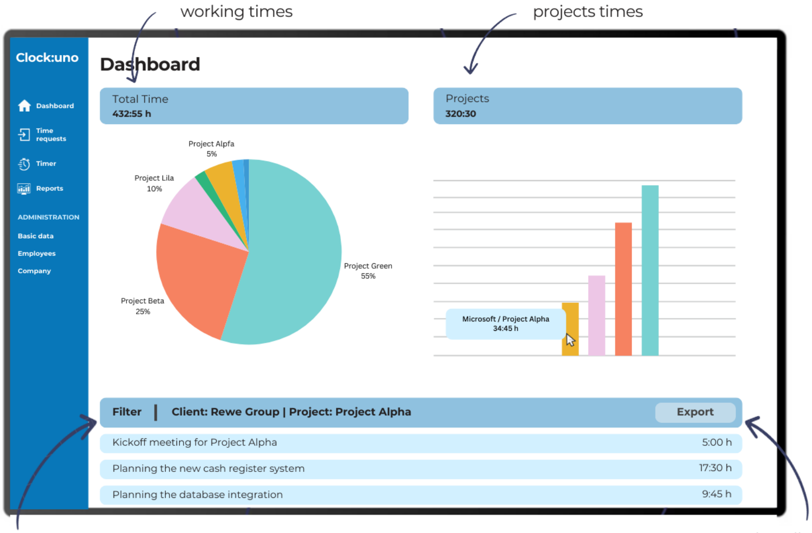
Ensure all tracked data remains on the user's device

Synchronize the project times with the PM software

Figure 1.4: Interactive time tracking Dashboard

Overview of all recorded working times

Overview of all recorded projects times



Clock:uno

Dashboard

Total Time
432:55 h

Projects
320:30

Filter | Client: Rewe Group | Project: Project Alpha

Kickoff meeting for Project Alpha	5:00 h
Planning the new cash register system	17:30 h
Planning the database integration	9:45 h

Filter by Client / Project

Exporting all working hours recorded accurately to clients

Figure 2: Use Case Diagram

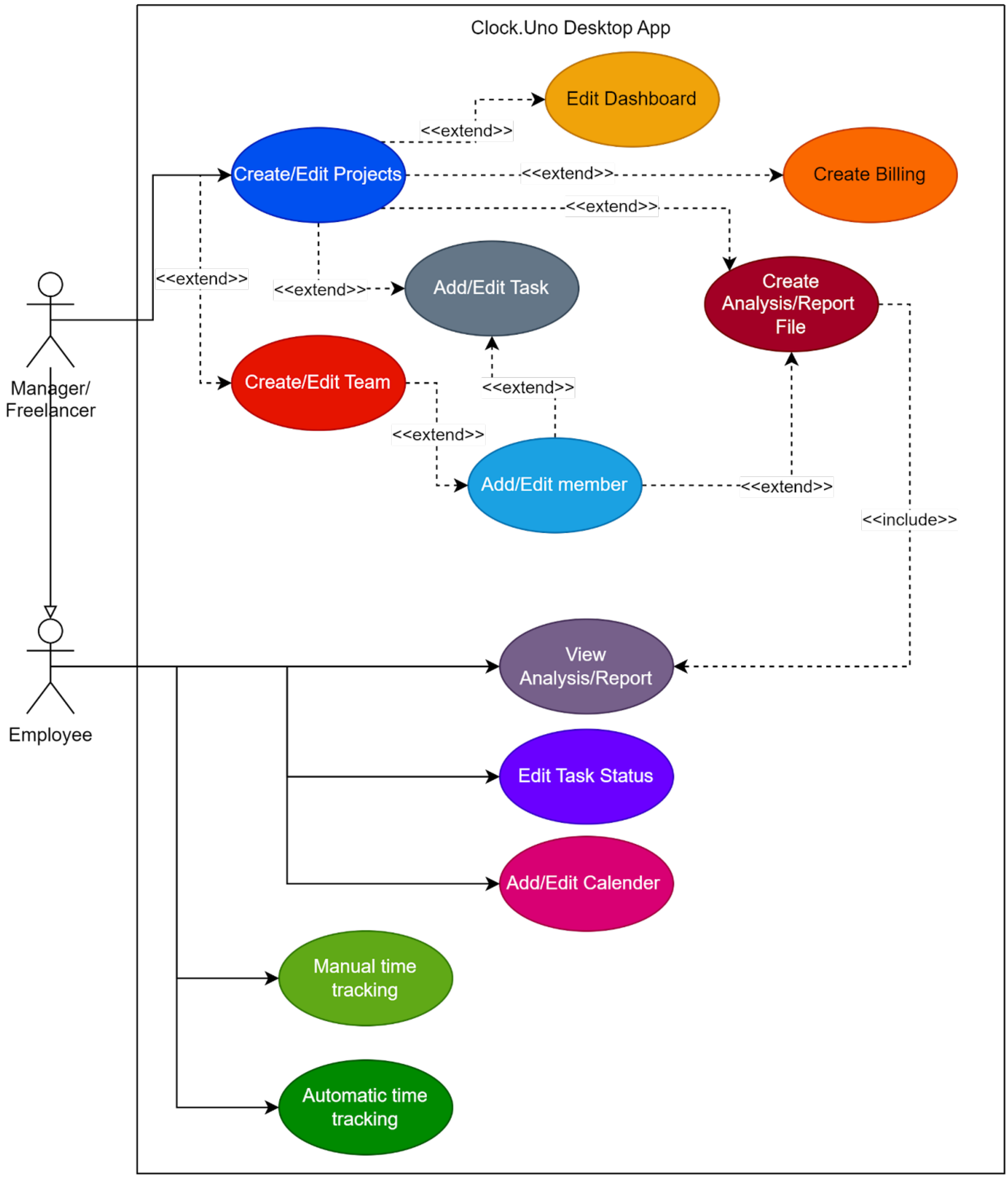


Figure 3: Scrum Process

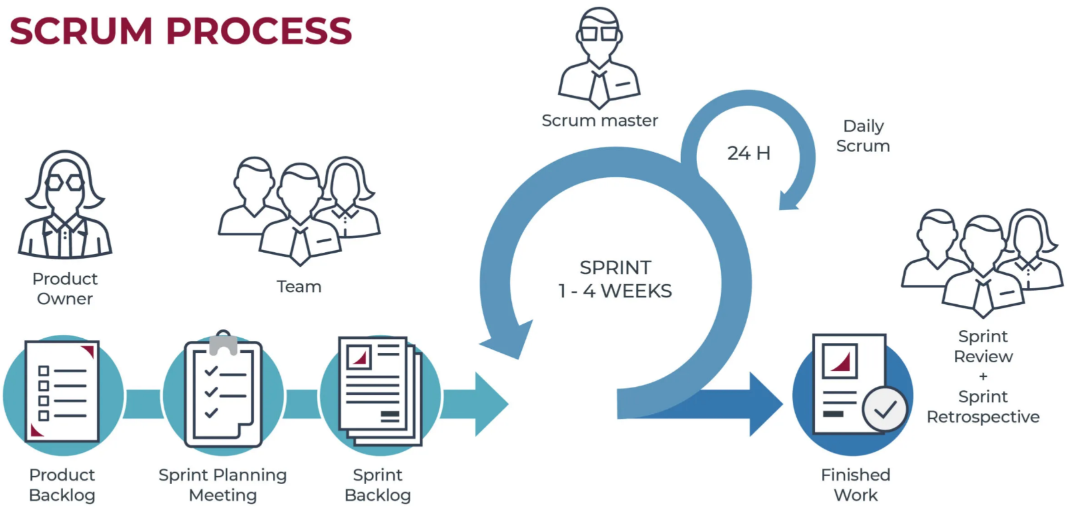
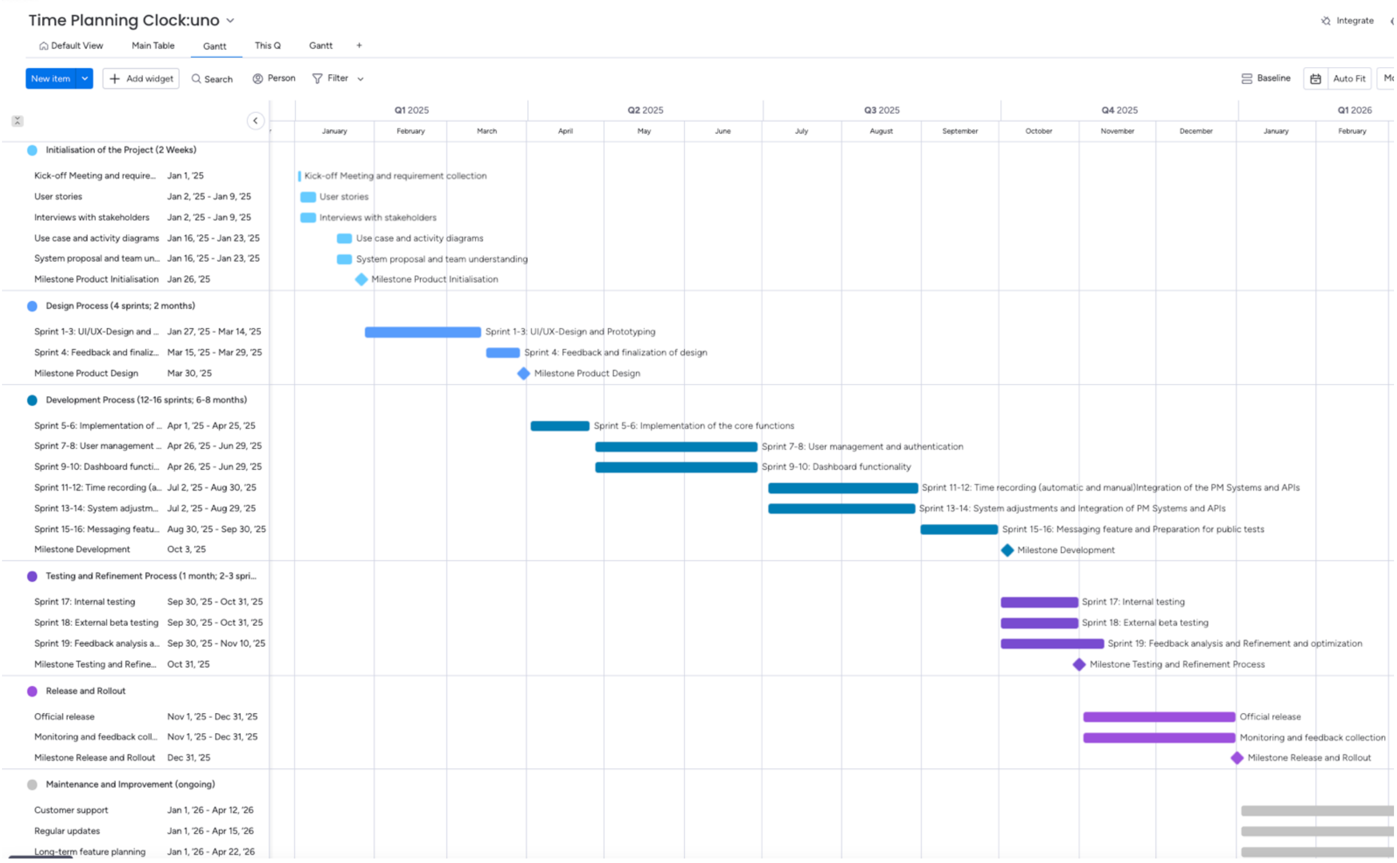
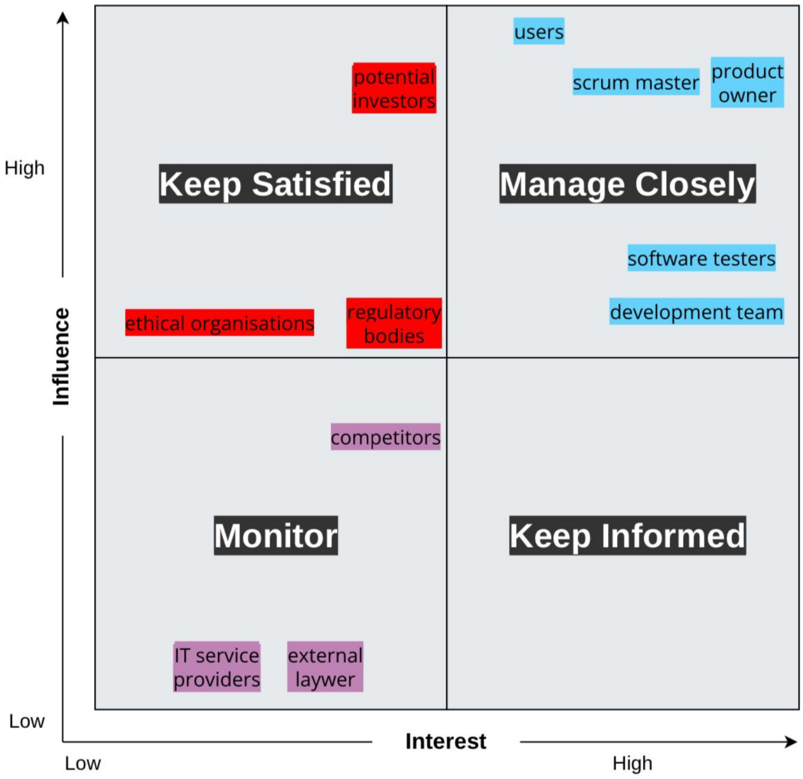


Figure 3: Project Timeline


Time Planning Clock:uno
Integrate
Default View
Main Table
Gantt
This Q
Gantt
+
New item
Add widget
Search
Person
Filter
Baseline
Auto Fit
Q1 2025
Q2 2025
Q3 2025
Q4 2025
Q1 2026
January
February
March
April
May
June
July
August
September
October
November
December
January
February
Kick-off Meeting and requirement collection
User stories
Interviews with stakeholders
Use case and activity diagrams
System proposal and team understanding
Milestone Product Initialisation
Sprint 1-3: UI/UX-Design and Prototyping
Sprint 4: Feedback and finalization of design
Milestone Product Design
Sprint 5-6: Implementation of the core functions
Sprint 7-8: User management and authentication
Sprint 9-10: Dashboard functionality
Sprint 11-12: Time recording (automatic and manual)Integration of the PM Systems and APIs
Sprint 13-14: System adjustments and Integration of PM Systems and APIs
Sprint 15-16: Messaging feature and Preparation for public tests
Milestone Development
Sprint 17: Internal testing
Sprint 18: External beta testing
Sprint 19: Feedback analysis and Refinement and optimization
Milestone Testing and Refinement Process
Official release
Monitoring and feedback collection
Milestone Release and Rollout

Initialisation of the Project (2 Weeks)	
Kick-off Meeting and require...	Jan 1, '25
User stories	Jan 2, '25 - Jan 9, '25
Interviews with stakeholders	Jan 2, '25 - Jan 9, '25
Use case and activity diagrams	Jan 16, '25 - Jan 23, '25
System proposal and team un...	Jan 16, '25 - Jan 23, '25
Milestone Product Initialisation	Jan 26, '25
Design Process (4 sprints; 2 months)	
Sprint 1-3: UI/UX-Design and ...	Jan 27, '25 - Mar 14, '25
Sprint 4: Feedback and finaliz...	Mar 15, '25 - Mar 29, '25
Milestone Product Design	Mar 30, '25
Development Process (12-16 sprints; 6-8 months)	
Sprint 5-6: Implementation of ...	Apr 1, '25 - Apr 25, '25
Sprint 7-8: User management ...	Apr 26, '25 - Jun 29, '25
Sprint 9-10: Dashboard functi...	Apr 26, '25 - Jun 29, '25
Sprint 11-12: Time recording (a...	Jul 2, '25 - Aug 30, '25
Sprint 13-14: System adjustm...	Jul 2, '25 - Aug 29, '25
Sprint 15-16: Messaging featu...	Aug 30, '25 - Sep 30, '25
Milestone Development	Oct 3, '25
Testing and Refinement Process (1 month; 2-3 spri...	
Sprint 17: Internal testing	Sep 30, '25 - Oct 31, '25
Sprint 18: External beta testing	Sep 30, '25 - Oct 31, '25
Sprint 19: Feedback analysis a...	Sep 30, '25 - Nov 10, '25
Milestone Testing and Refine...	Oct 31, '25
Release and Rollout	
Official release	Nov 1, '25 - Dec 31, '25
Monitoring and feedback coll...	Nov 1, '25 - Dec 31, '25
Milestone Release and Rollout	Dec 31, '25
Maintenance and Improvement (ongoing)	
Customer support	Jan 1, '26 - Apr 12, '26
Regular updates	Jan 1, '26 - Apr 15, '26
Long-term feature planning	Jan 1, '26 - Apr 22, '26

Figure 4: Stakeholder Matrix



List of Tables

Table 1.1: Questionnaire Form

General Information

1. What is your primary role?
   - ○ Freelancer
   - ○ Company Employee
   - ○ Project Manager
   - ○ Executive
   - ○ Other ____________
2. How many hours per week do you typically spend on time management activities (e.g., tracking time, generating reports, invoicing)?

   Answer:____
3. What project management tools do you currently use?
   - ○ Jira
   - ○ Asana
   - ○ Monday.com
   - ○ Trello
   - ○ Other____________

Current Time Management Practices

4. How do you currently track your time?
   (Manual entry, Automatic tools, Both, Other__________)
5. What challenges do you face with your current time tracking method?
   Answer:____
6. How important is the accuracy of time tracking to your work?
   - ○ Very Important
   - ○ Important
   - ○ Neutral
   - ○ Not Important

Needs and Priorities

7. Which features are most important to you in a time tracking tool?
   (Please rank in order of importance:
   - ○ Automatic Time Recording
   - ○ Manual Time Recording
   - ○ Exporting billing statements
   - ○ Reporting
   - ○ Integrations with Other Tools
8. For freelancers: How critical is the ability to accurately track billable hours for your business?
   (Very Critical, Critical, Neutral, Not Critical)
9. For company employees and project managers: How important is the integration of time tracking with existing project management systems?
   (Very Important, Important, Neutral, Not Important)
10. How concerned are you about the privacy and security of your time tracking data?
    (Very Concerned, Concerned, Neutral, Not Concerned)

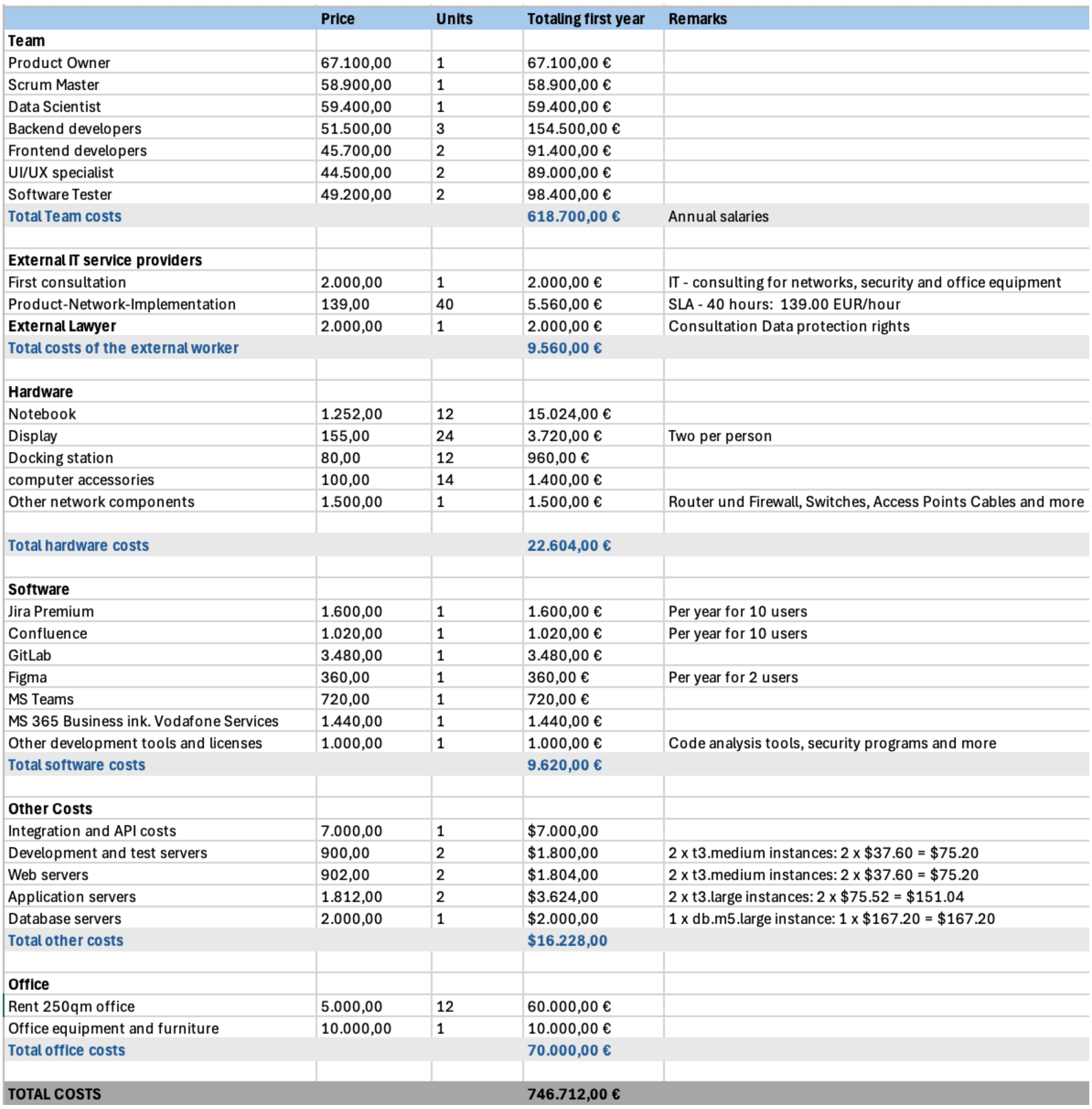
Reporting and Analysis

11. What types of reports do you currently generate?
    (Time Usage Reports, Productivity Trends, Project Progress, Billing Reports, Other)
12. How often do you need to generate these reports?
    (Daily, Weekly, Monthly, Quarterly, Annually)
13. What formats do you prefer for exporting reports?
    (PDF, Excel, CSV, Other__________)

Table 1.2: Interview Guide

Closed ended questions
Do you want an overview of the working time of your team? - Do you know how long your team is working on a project? - Do you think a system can track your time better? - Do you think a time tracking system is helping project management? - Do you primarily work from a desktop computer? - Would you use two-factor authentication for logging in? - Do you need the ability to export reports in PDF, Excel, or CSV formats?
Open ended questions
- Can you describe how you currently track your work hours? - How important are privacy settings? - What does your workspace look like? - What features would you like to see in a new time management software? Probing questions: - How long does it take to track your working time? - Why do you prefer automatic time tracking over manual entry? - Can you provide a specific example of a situation where your current time tracking method fell short? - What privacy concerns are particularly important to you, and why? - How often do you need to share your work hours and project progress with others, and what formats do you prefer? - What expectations do you have for the calendar functionality in a time management software?

Table 2: Cost estimation



	Price	Units	Totaling first year	Remarks
Team				
Product Owner	67.100,00	1	67.100,00 €	
Scrum Master	58.900,00	1	58.900,00 €	
Data Scientist	59.400,00	1	59.400,00 €	
Backend developers	51.500,00	3	154.500,00 €	
Frontend developers	45.700,00	2	91.400,00 €	
UI/UX specialist	44.500,00	2	89.000,00 €	
Software Tester	49.200,00	2	98.400,00 €	
Total Team costs			**618.700,00 €**	Annual salaries
External IT service providers				
First consultation	2.000,00	1	2.000,00 €	IT - consulting for networks, security and office equipment
Product-Network-Implementation	139,00	40	5.560,00 €	SLA - 40 hours: 139.00 EUR/hour
External Lawyer	2.000,00	1	2.000,00 €	Consultation Data protection rights
Total costs of the external worker			**9.560,00 €**	
Hardware				
Notebook	1.252,00	12	15.024,00 €	
Display	155,00	24	3.720,00 €	Two per person
Docking station	80,00	12	960,00 €	
computer accessories	100,00	14	1.400,00 €	
Other network components	1.500,00	1	1.500,00 €	Router und Firewall, Switches, Access Points Cables and more
Total hardware costs			**22.604,00 €**	
Software				
Jira Premium	1.600,00	1	1.600,00 €	Per year for 10 users
Confluence	1.020,00	1	1.020,00 €	Per year for 10 users
GitLab	3.480,00	1	3.480,00 €	
Figma	360,00	1	360,00 €	Per year for 2 users
MS Teams	720,00	1	720,00 €	
MS 365 Business ink. Vodafone Services	1.440,00	1	1.440,00 €	
Other development tools and licenses	1.000,00	1	1.000,00 €	Code analysis tools, security programs and more
Total software costs			**9.620,00 €**	
Other Costs				
Integration and API costs	7.000,00	1	$7.000,00	
Development and test servers	900,00	2	$1.800,00	2 x t3.medium instances: 2 x $37.60 = $75.20
Web servers	902,00	2	$1.804,00	2 x t3.medium instances: 2 x $37.60 = $75.20
Application servers	1.812,00	2	$3.624,00	2 x t3.large instances: 2 x $75.52 = $151.04
Database servers	2.000,00	1	$2.000,00	1 x db.m5.large instance: 1 x $167.20 = $167.20
Total other costs			**$16.228,00**	
Office				
Rent 250qm office	5.000,00	12	60.000,00 €	
Office equipment and furniture	10.000,00	1	10.000,00 €	
Total office costs			**70.000,00 €**	
TOTAL COSTS			**746.712,00 €**	

References

Salaries:

- https://www.stepstone.de/gehalt/Product-Owner-in.html
- https://www.stepstone.de/gehalt/Scrum-Master-in.html
- https://www.stepstone.de/gehalt/Data-Scientist.html
- https://www.stepstone.de/gehalt/Backend-Developer.html
- https://www.stepstone.de/gehalt/Frontend-Entwickler-in.html
- https://www.stepstone.de/gehalt/UIUX-Designer.html
- IT Service Provider

Hardware:

- Notebook
- Display

Software:

- Jira
- Confluence
- GitLab
- Figma
- MS Teams Business
- MS 365 Business ink. Vodafone Services

Other:

- Server
- Scrum process figure
- Art. 5 GDPR