

Project plan

Project name	Client / Sponsor	Project manager

1 Executive summary

A short summary of the project plan

2 Background

Description with a clear and defined connection to the goals. It is advisable to connect to any related project in the background description.

3 Purpose

The impact the project is expected to create, i.e. why it is important to execute the project.

4 Goal

The result the project should deliver, i.e. what should be achieved when the project is executed.

5 Scope

What is included as part of the project and must be performed in order to deliver the goal. The scope is described with a WBS at the overarching level – main packages with a brief description of each. The complete WBS should be included as an attachment.

6 Limitations

What the project should not deliver. The purpose is to avoid false expectations among the different stakeholders.

7 Requirement

7.1 Product requirements

The product specification describes the product that is to be delivered. It is a description of the product in terms of its functionality, performance, quality, etc.

7.1.1 Functional requirements

The functional requirements describe the functions that the product must do.

7.1.2 Non-functional requirements

The non-functional requirements describe the technical specifications of the product in details: e.g. colors, performance, quality metrics.

7.2 Project requirements

Requirements on the execution and prioritization between the project's triple constraints.

7.3 Prerequisites

Demands on the project's sponsor/owner or client that have to be achieved to ensure the project's execution and result.

8 Situational analysis and stakeholders

8.1 SWOT-analysis

Mapping and analysing external and internal factors that might affect execution.

Strengths:

- Strong technical team
- Experienced project management
- Patents on unique technologies

Weaknesses:

- Limited funding
- Lack of experience in the autonomous robotics market
- Supply chain uncertainties

Opportunities:

- Emerging autonomous technology market
- Collaboration opportunities with universities and research institutions
- Regulatory changes could open new markets

Threats:

- Fast-paced technology changes
- Strong competition from established tech companies
- Regulatory restrictions in certain markets

Conclusions:

- Need to secure additional funding and focus on continuous skills development
- Strategic alliances should be considered to mitigate weaknesses and threats
- Constant monitoring and lobbying required to navigate regulatory landscape

8.2 Stakeholder mapping

General example Mapping and analysis of individuals, groups, and organizations that might affect or will be affected by the project.

The stakeholder mapping could include the following:

- **Internal Stakeholders:** This might include the project team, other employees, and the company's management.
- **Investors:** Those who have financially backed the project and have an interest in its success.
- **Partners:** Other organizations or entities working with the company on the project.
- **Customers:** The end-users will use the autonomous robot once it is developed.
- **Regulatory Bodies:** Government and non-government agencies that oversee the rules and regulations that the project must adhere to.
- **Suppliers:** Companies that provide the materials and resources needed for the project.
- **Competitors:** Other companies are developing similar technologies or solutions.

9 Planning

9.1 Milestone plan

Stakeholders may want an overarching flow chart or table of the project's most important milestones as a indicator if the project is falling behind.

9.2 Work breakdown structure

The Work Breakdown Structure (WBS) is a fundamental tool in project management, establishing a systematic and structured approach to break down the intricate project into manageable components. This detailed, hierarchical decomposition of the project's scope presents a visual or textual representation that facilitates an understanding of the project's structure, workflow, and tasks necessary for successful project completion. A key note is that the WBS should only include items related to the project and not the project course, this excludes items such as course assignments as they are not a part of the project, only the course.

If you need an anecdote, picture this: "You are a consultant assigned to help clients develop a Work Breakdown Structure (WBS) for their project. This WBS will outline the project's components, giving a clear roadmap of what is to come. You also have internal tasks like time reports, presentations, and assignments for your consulting firm. However, it is important to remember that these internal duties belong outside the client's WBS. They are part of your job, not the client's project. By keeping the WBS client-specific, you maintain clarity and ensure an effective project management strategy."

The WBS dissects the project into multiple layers for ease of management, separating the work into components, work packages, deliverables, and tasks.

- **Components** represent the broad sections or stages of the project.
- **Work Packages** are subsets of these components, which can be further broken down into specific deliverables.
- **Deliverables** are tangible or intangible goods or services produced in the project. They can be handed over physically or digitally to another person or team, forming the backbone of each work package and contributing substantially to the overall project's progression.
- **Tasks** represent the smallest work units necessary to complete each deliverable, providing clear action items for project participants.

By enabling systematic project planning, effective resource allocation, and reliable progress tracking, the WBS becomes an invaluable aid in navigating the complexity of projects and driving them towards successful outcomes.

A small text-based example, recommend visual based, of applying a WBS for a mobile autonomous robot development project is illustrated below, and only one component is added to keep it short for the example. This example should provide a more tangible understanding of the WBS and its purpose.

ID	Type	Activity
1	Root	Mobile Autonomous Robot Development
1.1	Component	Navigation and Path Planning
1.1.1	Work Package	Sensor Integration
1.1.1.1	Deliverable	Sensor Selection Report
1.1.1.1.1	Task	Research Suitable Sensors (LiDAR, RADAR, Ultrasonic, etc.)
1.1.1.1.2	Task	Prepare Sensor Selection Report
1.1.1.2	Deliverable	Sensor Installation Manual
1.1.1.2.1	Task	Install Sensors on Robot Chassis
1.1.1.2.2	Task	Prepare Sensor Installation Manual
1.1.1.3	Deliverable	Sensor Data Fusion
1.1.1.3.1	Task	Develop Sensor Data Fusion Algorithm
1.1.1.3.2	Task	Implement and Test Sensor Data Fusion Algorithm
1.1.2	Work Package	Path Planning
1.1.2.1	Deliverable	Path Planning
1.1.2.1.1	Task	Research Suitable Path Planning Algorithms
1.1.2.1.2	Task	Implement Selected Path Planning Algorithm
1.1.2.2	Deliverable	Path Planning Testing Report
1.1.2.2.1	Task	Test Path Planning Algorithm in Simulated Environment
1.1.2.2.2	Task	Prepare Path Planning Testing Report

9.3 Schedule

Activity plan with a time axis where duration and connection between activities and milestones are shown.

10 Staffing

Example of a staffing plan, add references to other plans that might include additional information.

10.1 Roles, responsibilities and authorities

Our project team structure is specifically crafted to leverage the distinctive skills and capabilities of each team member, fostering effective teamwork. The key roles, responsibilities, and authorities in our project include:

- **Software (SW) Team Lead:** Leads the software team, responsible for overseeing all activities related to software development, ensures that software requirements are met and coordinates with the developers on the software team.
- **Hardware (HW) Team Lead:** Leads the hardware team, responsible for all activities related to hardware development, ensures that hardware requirements are met and coordinates with the developers on the hardware team.
- **Software (SW) Developer:** Implements software components, conducts testing and debugging, and addresses software-related issues.
- **Hardware (HW) Developer:** Designs, tests, and debugs hardware components, addresses hardware-related issues.

10.2 Staffing plan

Our staffing plan assigns roles to each member of our project team as follows:

- Stieg Larsson - Software Team Lead
- Astrid Lindgren - Hardware Team Lead
- Henning Mankell - Software Developer
- Selma Lagerlöf - Hardware Developer

Each individual was chosen for their role based on their skills, experience, and interest in the project. Regular meetings will be held to ensure that everyone is on track and to address any concerns that may arise.

11 Project budget

The project's preliminary calculation – a outline of internal and external costs for resources needed to execute the project.

Internal costs	External costs	Other costs	Summary

12 Development Plan

Example of a development plan

This Development Plan provides a comprehensive roadmap for the project, detailing our technical approach, the development process, the methodologies we employ, and each team member's distinct roles and responsibilities. The plan is designed to facilitate seamless integration for new members joining at any project lifecycle stage. By becoming familiar with the plan, you will gain crucial insights into our work processes and project-related information, enabling you to contribute effectively and align with the team's objectives.

12.1 Development Methodology

Our team strictly adheres to [specific methodology - e.g., Agile, Scrum, Waterfall, etc.]. We chose this methodology due to its [specific benefits], encompassing practices such as [describe practices].

12.2 Team Structure and Roles

Please note: The roles listed here might have additional responsibilities outlined in the corresponding plans (Documentation Plan, Communication Plan, Testing Plan).

Our team comprises the following roles:

- **Hardware Team Leader:** Manages the hardware development process, coordinates with the software team, and ensures timely delivery while maintaining quality standards.
- **Software Team Leader:** Oversees the software development process, collaborates with the hardware team, and ensures project deadlines are met with high-quality deliverables.
- **Hardware Developer:** Designs, develops, tests, and troubleshoots hardware components and collaborates with the software team.
- **Software Developer:** Develops, tests, and debugs software modules and coordinates with the hardware team.

12.3 Tools, Technologies, and Systems

We leverage the following tools, technologies, and systems in our project:

- **Software Development Tools:** Include various programming languages and 3D modelling software. For instance, [Programming Language] is used for [purpose] and can be downloaded from [source]. For 3D modelling, we use [Software name], which can be downloaded from [source].
- **Project Management Tools:** We utilize [Tool/Platform Name, version] for [purpose], and it can be downloaded from [source].
- **Communication Tools:** For effective communication, we use [Tool/Platform Name, version], which can be downloaded from [source].
- **Hardware Tools and Software:** We utilize tools like a 3D printer ([specify model]) and PCB Design Software ([Software Name]) which can be downloaded from [source].
- **Operating Systems:** We primarily operate on [Operating System name and version]. Please align your system with the same for consistency and compatibility. If your system operates differently, notify the team for assistance with potential compatibility issues.
- **Hardware:** We primarily use [Computer model/brand] with [specify processor, RAM, storage]. Please let us know if you use a different model for compatibility checks.

Please install the correct version numbers as specified for consistency and compatibility.

12.4 Coding Standards and Guidelines

To ensure high code quality and readability, we follow these coding standards and guidelines: [Outline specific standards and guidelines]. For more comprehensive information, refer to our detailed coding standards document [link to document].

12.5 Version Control

We utilize [Version Control System Name, e.g., Git] to manage changes to our project files and maintain different versions. For detailed guidance on using this system, please refer to our [link to version control usage guide].

12.6 Testing and Quality Assurance

We employ [describe testing methodologies] for our testing and QA. For detailed information on our testing process, refer to our [link to Testing and QA document or section].

12.7 Integration and Deployment

Our integration strategy involves [describe strategy, e.g., Continuous Integration]. The deployment process includes [describe process], facilitated by specific tools. For comprehensive information on these processes, refer to our integration and deployment guide [link to guide].

13 Communication plan

Plan for spreading information in the purpose of guaranteeing the right target group gets the right information at the right time and through the right channels, simple example listed below.

The Communication Plan is essential to any project, serving as a guide for sharing information throughout the project lifecycle. It optimizes the distribution of project-related data among team members and stakeholders, fostering cooperation, ensuring transparency, and promoting mutual understanding.

This strategic document lays out various facets of communication, such as the type of information that needs to be disseminated, the audience for each type of information, the timing or frequency of communication, and the mediums or channels for transmitting the information.

- **Who (Target Audience):** This determines who needs to receive specific information. This could include project team members such as the hardware and software developers, hardware and software team leaders, stakeholders, the examiner, the course responsible, and the supervisor.
- **Why (Purpose):** This refers to the reason behind the communication. For example, project updates inform stakeholders about progress, and task assignments may be needed to guide the developers. At the same time, risk alerts might be necessary to manage potential issues.
- **What (Type of Information):** This refers to the specific data or information to be shared. This could include project updates, task assignments, meeting agendas, change requests, and risk alerts.
- **When (Timing):** This indicates when and how often the communication should occur. This could be daily, weekly, biweekly, monthly, or at specific project milestones, depending on the nature of the information.
- **How (Communication Channels):** This determines how the information will be delivered. Communication methods include email, direct communication, meetings, project management software, or the course management system.
- **Responsible:** This identifies who is responsible for ensuring the communication happens. This can vary depending on the type of information and the target audience.

By implementing a well-structured communication plan, potential misunderstandings can be minimized, efficient decision-making can be facilitated, and a conducive environment for project success can be established. This is achieved through setting clear communication protocols and procedures that are known and understood by everyone involved in the project.

The following example shows a simple communication plan for a project. This table is just a template and can be modified to fit your project's needs, try to be more specific and who it may concern:

Who	Why	What	When	How	Responsible
HW Team Leader	To keep updated on project status	Project Updates	Weekly	Email, Project Management Software	Project Team
SW Team Leader	To keep updated on project status	Project Updates	Weekly	Email, Project Management Software	Project Team
HW Developer	To know their assignments	Task Assignments	As needed	Project Management Software	HW and SW Team Leaders
SW Developer	To know their assignments	Task Assignments	As needed	Project Management Software	HW and SW Team Leaders
HW Team Leader	To prepare for meetings	Meeting Agendas	Before each meeting	Email	Meeting Organiser
SW Team Leader	To prepare for meetings	Meeting Agendas	Before each meeting	Email	Meeting Organiser
HW Developer	To adapt to changes	Change Requests	As needed	Meetings, Email	Person Requesting Change
SW Developer	To adapt to changes	Change Requests	As needed	Meetings, Email	Person Requesting Change
HW Team Leader	To manage risks	Risk Alerts	As needed	Direct Communication, Email	Risk Manager
SW Team Leader	To manage risks	Risk Alerts	As needed	Direct Communication, Email	Risk Manager

14 Risk analysis and response planning

Example section

We conduct a rigorous risk analysis and response planning as part of our project management. This involves identifying potential risks and assigning them a probability (from 1, very unlikely, to 5, very likely) and an impact score (from 1, minimal, to 5, critical). We then calculate a risk score by multiplying the probability by the impact.

The risk score thresholds are interpreted as follows:

- 1-5: Low priority risks. These are monitored but may not require immediate action.
- 6-10: Medium priority risks. These require a mitigation plan and should be addressed as resources allow.
- 11-15: High priority risks. These require a detailed mitigation plan, and resources should be allocated to address these risks immediately.
- 16-25: Critical risks. These must be addressed immediately with a detailed and efficient mitigation plan to avoid severe project disruption.

The table below outlines the identified risks, their evaluation, and planned responses:

Risk	Probability(1 to 5)	Impact(1 to 5)	Risk(P*I)	Risk Response
Critical team member leaves during project	2	5	10	Develop contingency plan, cross-train team members
Scope creep leading to project delay	4	4	16	Regular scope review, robust change management process
Technology becomes obsolete	1	4	4	Regularly update technological skillsets, maintain flexibility in tech stack
Software bugs detected after deployment	3	5	15	Rigorous testing process, allocate resources for post-deployment bug fixes
Unexpected budget cuts	2	5	10	Prepare a flexible budget that can accommodate cuts, regular financial reviews

The mitigation strategies and responses to these risks form an integral part of our project plan, ensuring that we are prepared for uncertainties and can effectively manage them should they arise.

15 Documentation plan

Plan for how the project's documentation is to be managed. Example listed below.

The purpose of a Documentation Plan is to provide a structured framework for creating, storing, reviewing, and disseminating project-related information. It ensures systematic recording and retention of all details, facilitating efficient communication, transparency, accountability, and continuity throughout the project. Comprehensive documentation is a critical tool in project management, aiding in tracking progress, making informed decisions, and providing references for future initiatives. Furthermore, it promotes knowledge sharing and learning during the project and beyond.

The following sections present a detailed example of a Documentation Plan. It outlines what needs to be documented, how and when to do so, who is responsible for various documentation tasks, where the documents will be stored, and the review and approval process for these documents. This example serves as a guide and should be tailored to fit your project's specific needs and circumstances.

15.1 What to Document

Define the types of documentation that will be required. This could include design documents, technical specifications, meeting minutes, code comments, testing reports, user manuals, troubleshooting guides, etc.

15.2 How to Document

Specify the format and structure of each document. For example, technical specifications might require diagrams and detailed descriptions, while meeting minutes may be more informal. Develop templates for consistency.

15.3 When to Document

Define when each document should be updated. For instance, design documents should be updated whenever a design decision is made, testing reports after every test run, and meeting minutes following each meeting.

15.4 Who is Responsible

Assign documentation duties to specific roles:

- **Hardware Team Leader:** Oversees and approves hardware-related documentation and assigns hardware-related documentation tasks.
- **Software Team Leader:** Oversees and approves, and assigns software-related documentation tasks.
- **Hardware Developer:** Creates and updates hardware-related documentation as assigned.
- **Software Developer:** Creates and updates software-related documentation as assigned.

15.5 Where to Store Documents

Define where documents will be stored. This could be on a shared drive, a project management tool, or a version control system. Ensure it is accessible to all team members and has appropriate backup and security measures.

15.6 Document Review Process

Specify a process for reviewing and approving documents. This could involve peer reviews, leader reviews, or formal approval processes.

15.7 Training

Ensure everyone involved in the project is aware of the documentation plan and trained on how to use the tools and templates. This can be part of the initial project orientation or documentation training session.

16 Testing Plan

General example This Testing Plan provides a detailed outline of the testing strategy, approach, and procedures we employ in our project. It establishes a clear pathway for validating and verifying our project's software and hardware components, ensuring they meet the specified requirements. The plan is designed to offer clarity for any team members joining at any project lifecycle stage. Understanding this plan will provide a solid understanding of our testing practices and expectations, aligning with our overall project goals.

16.1 Testing Methodology

Our team strictly follows the [specific testing methodology - e.g., Manual, Automated, Unit, Integration, etc.]. This methodology has been chosen due to its [specific benefits] and involves procedures such as [describe procedures].

16.2 Testing Team Structure and Roles

Please note: The roles listed here might have additional responsibilities outlined in the corresponding plans (Development Plan, Documentation Plan, Communication Plan).

Our testing team includes the following roles:

- **Testing Team Leader:** Oversees the overall testing strategy, coordinates with the development team, and ensures thorough and timely testing.
- **Test Engineer:** Designs, develops, and executes test cases. They are also responsible for documenting and communicating the results to the team.

16.3 Testing Tools, Technologies, and Systems

We leverage the following tools, technologies, and systems in our testing process:

- **Testing Tools:** We utilize [Testing Tool/Platform Name, version] for [purpose], and it can be downloaded from [source].
- **Bug Tracking Tools:** For tracking and managing defects, we use [Tool/Platform Name, version], which can be downloaded from [source].
- **Test Management Tools:** We utilize [Tool/Platform Name, version] for [purpose], and it can be downloaded from [source].
- **Operating Systems:** We primarily operate on [Operating System name and version]. Please align your system with the same for consistency and compatibility. If your system operates differently, notify the team for assistance with potential compatibility issues.

Please install the correct version numbers as specified for consistency and compatibility.

16.4 Testing Standards and Guidelines

To ensure consistent and high-quality testing, we follow these testing standards and guidelines: [Outline specific standards and guidelines]. For more detailed information, please refer to our detailed testing standards document [link to document].

16.5 Version Control for Test Artifacts

We utilize [Version Control System Name, e.g., Git] to manage changes to our test artefacts and maintain different versions. For detailed guidance on using this system, please refer to our [link to version control usage guide].

16.6 Bug Reporting and Tracking

We use [Bug Tracking System Name] to log and track identified defects. Each testing team member is responsible for documenting and reporting bugs according to our bug reporting guidelines, outlined in our [link to Bug Reporting Guide].

16.7 Test Schedule

Our testing schedule aligns with the overall project timeline. The detailed test schedule can be found in our project timeline document [link to document].

16.8 Integration and Regression Testing

Our integration and regression testing strategy involves [describe strategy, e.g., Continuous Integration]. For comprehensive information on these processes, refer to our integration and regression testing guide [link to guide].

17 Handover plan

How to deliver the product to the client and implement it into the environment it is meant for. The following subsections is an general simple example

This Handover Plan outlines the systematic process of transferring all project deliverables, documentation, tools, and knowledge to the project owner at the end of the project or a project phase. The objective is to ensure a smooth transition, preserving all the hard work done on the project and providing a firm basis for future work. Understanding this plan will ensure that all team members know what is required in the final stages of the project.

17.1 Handover Methodology

Our handover methodology includes [describe methods and steps such as inventory check, final documentation, presentation, and sign-off]. These steps have been chosen to ensure a comprehensive and efficient handover process.

17.2 Handover Team Structure and Roles

Our handover team includes the following roles:

- **Handover Team Leader:** Coordinates the handover process, ensuring that all project deliverables and documentation are transferred wholly and accurately.
- **Handover Specialist:** Conducts presentations (if required), answers queries about the project, and ensures that the project owner is comfortable with understanding the project's results and operation.

17.3 Handover Tools, Technologies, and Systems

We leverage the following tools, technologies, and systems during our handover process:

- **Inventory Management Tools:** We utilize [Tool/Platform Name, version] to catalogue and track all project deliverables for handover, which can be downloaded from [source].
- **Presentation Tools:** If necessary, we use [Tool/Platform Name, version] to conduct presentations or demonstrations, and it can be downloaded from [source].

Please install the correct version numbers as specified for consistency and compatibility.

17.4 Handover Schedule

Our handover schedule aligns with the overall project timeline. The detailed handover schedule can be found in our project timeline [link to document].

17.5 Documentation

Comprehensive project documentation, including user manuals, technical documentation, and project reports, will be provided during the handover process. These documents can be accessed from [link to Document Storage/Management System].

17.6 Presentation and Demonstration

Our team will conduct presentations and demonstrations where necessary to familiarize the project owner with the project's operation and results.

17.7 Final Sign-off

The project owner will sign off, indicating successful handover and completion. This will be done using [describe sign-off method/process].

17.8 Version Control for Handover Artifacts

We utilize [Version Control System Name, e.g., Git] to manage changes to our handover artefacts and maintain different versions. For detailed guidance on using this system, please refer to our [link to version control usage guide].

17.9 Follow-up and Feedback

We highly value feedback from the project owner and have a process for follow-up [describe follow-up process] to ensure that all aspects of the handover were completed satisfactorily and address any potential concerns post-hand-over.

18 Other

Anything else?

References