SOLAR ROUTE PLANNER

Technical Process Report

Janhavi Gaikwad (jg2569)
Systems Engineering, M-Eng'25

Table of Contents

Introduction

Drivers in urban environments are often exposed to varying levels of solar radiation during their commutes, which can lead to health conditions, discomfort, increased fatigue, and reduced overall driving efficiency. Conventional navigation systems prioritize the shortest or fastest routes but do not account for environmental factors such as sunlight exposure. The goal of this project is to develop a solar aware route planner for vehicles that minimizes sun exposure while maintaining efficient travel times. The system integrates real time solar and weather data from external data centers, calculates shadow parameters, and applies optimization techniques to generate personalized driving routes. Using systems engineering principles, the project defines functional and structural requirements, identifies potential failure modes, and ensures that the final solution is reliable, efficient, and aligned with driver comfort and safety needs.

This report documents the lifecycle technical process for development of solar route planner from conceptualization through retirement. It systematically describes each stage from conceptual to retirement and lists key processes in each stages. For each process, the report outline its goals and discuss systems engineering tools and methods applied such as customer affinity process, FMEA etc. It also talks about the possible pitfall of using the tool. Through this structured approach the report demonstrates how robustness, reliability and systems thinking were maintained throughout product lifecycle

Technical Process Report

Concept Stage

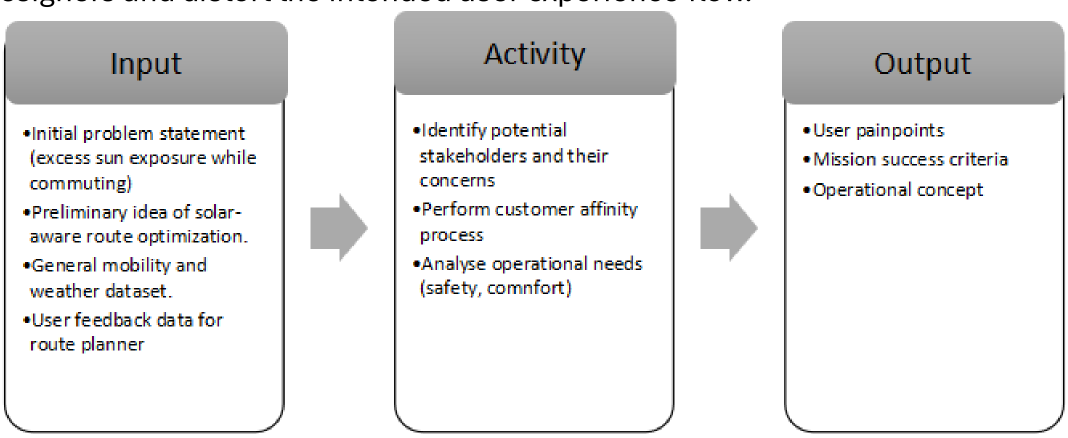
The Concept Stage focuses on establishing a clear concept for the Sun-Exposure-Aware Route Planning system by defining high-level goals, identifying key stakeholders, and outlining the overall system scope and boundaries to deliver user center solution for reducing solar exposure. A critical part of this stage is understanding the domain problem of solar exposure and translating stakeholder needs into preliminary objectives, such as minimizing cumulative sun exposure while maintaining reasonable travel distances. The stage also aims to explore solution alternatives, assess technical feasibility, and identify key risks and assumptions, ensuring that the system concept is both practical and sufficiently detailed to support the subsequent requirements definition and development stages.

Mission Analysis process

The mission analysis process focuses on assessing and understanding the needs of drivers operating under sunny conditions, emphasizing the impact of sun exposure on comfort, safety, and travel preferences. An important goal of this process is to identify and analyze end-user pain points early, enabling the design of user-centered solutions that balance exposure reduction with acceptable travel time. The mission analysis process also aims to uncover trade off criteria, such as the willingness of a user to travel slightly longer to avoid direct sunlight, ensuring that the system concept aligns with real world user priorities.

In our project, the **customer affinity process** aided us in achieving the goal of understanding user needs and environmental constraints. By identifying key pain points, such as discomfort due to prolonged sun exposure, the affinity process helped define early requirements and informed design decisions that balance user comfort with route efficiency. Poorly executed affinity analyses, however, can miss critical needs, potentially leading to suboptimal system performance.

Additionally, **annotated concept sketches** provided a functional visualization of system interactions from user's perspective, helping stakeholders comprehend how different elements, such as route planning, data management, and user interfaces work together to mitigate solar exposure. Incomplete or unclear sketches could mislead designers and distort the intended user experience flow.



Stakeholder needs and requirements process

The Stakeholder Needs and Requirements process focuses on defining the system requirements necessary to satisfy both stakeholder and customer expectations. It translates high-level needs into actionable, quantitative objectives. An important goal of this process is to develop functional requirements aligned with identified customer needs, investigate potential solutions for minimizing sun exposure, and establish methods to collect essential data to quantify performance targets. This process also aims to identify supplementary requirements critical to the operation and integration of primary functions, while generating preliminary functional concepts for feasible solutions to present to stakeholders for validation and alignment.

In our project, the **context diagram** helped define the operational environment and system boundaries for the Sun Exposure Aware Route Planner, clarifying how the system interacts with external entities such as data centers and other data sources. Incorrect context diagrams may misrepresent system boundaries, leading to design errors.

The **activity diagram** provided a visual model of system from user's perspective, including route planning, navigation, and data updates, improving understanding of operational workflows. It is important to make FFBD consistent with activity diagrams to avoid logical ambiguity in systems.

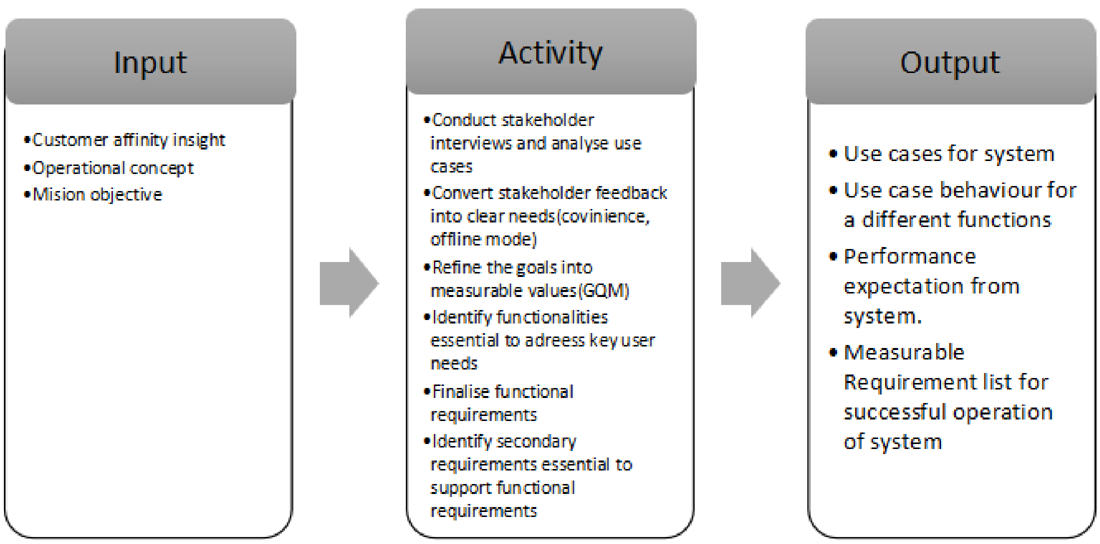
Use cases and user stories captured system functionality from the driver's perspective, ensuring that stakeholder needs were comprehensively represented. Insufficient use cases can lead to functional gaps in the system.

Use case behavioral diagrams modeled system responses to changing conditions such as weather updates or data center failures. It helps uncover the additional systems requirement. Although insufficient details can introduce logic gaps like missing system-to-display communication steps

Originating Requirements consolidates these insights at a single table to simplify planning, tracking and later validation processes. It is important to make sure all requirements are present in the requirement table to ease tracing their progress.

Customer Value Proposition summarized key user expectations, aligning technical development with user goals. Although inaccurate or exaggerated statements may create misalignment in stakeholder expectations.

Finally, the **GQM** approach established measurable performance metrics, such as reductions in cumulative sun exposure, ensuring that system success could be objectively quantified. Poorly executed GQM, however, could increase the cost of execution.



Requirement Definition Process

The Requirement Definition process focuses on evaluating feasible solutions against defined metrics and stakeholder expectations, and functional requirements, ensuring that proposed approaches effectively satisfy user needs and system objectives. Key goals of this process include exploring alternative solutions to determine the best fit for system objectives, translating user-centric requirements into detailed engineering specifications, and establishing a structured framework for evaluating design decisions based on quantifiable performance criteria such as sun exposure reduction and travel time.

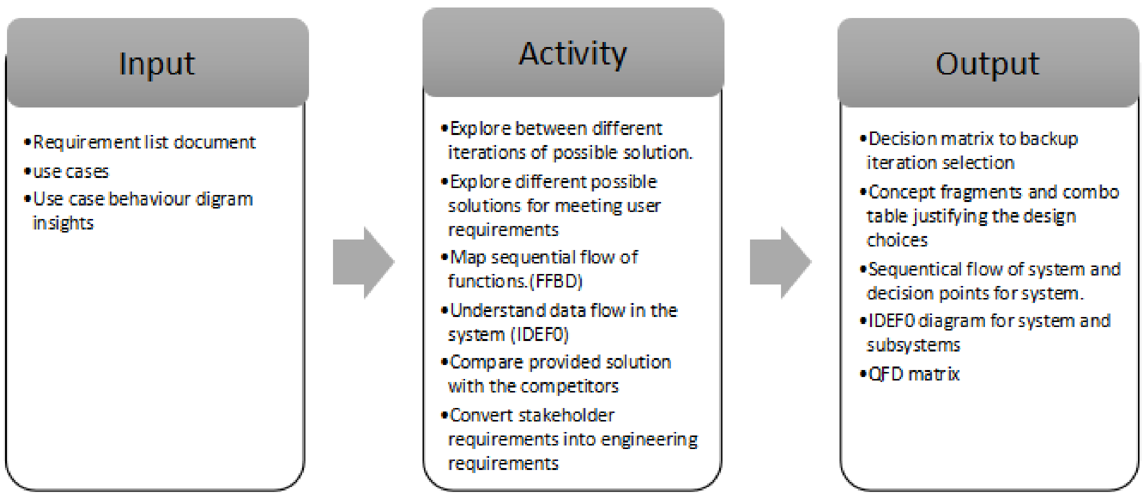
In our project, **concept fragments and combo tables** facilitated exploration of multiple design alternatives, including hybrid data update methods and adaptive recalculation strategies, encouraging creative problem-solving and systematic evaluation. Although, failure to document reasons for rejecting alternatives may cause confusion in later design reviews.

The **Functional Flow Block Diagram** mapped sequential system functions and critical decision points, providing visibility into operational logic. But missing key decision points can result in unrealistic operational assumptions.

IDEF0 diagrams clarified process dependencies and data flow interactions across different stages of the system. But errors or omissions can obscure data handoffs and dependencies.

The **Analytical Hierarchy Process** helped prioritize performance factors using weighted criteria, ensuring transparent and objective decision-making, and decision metrics enabled quantitative comparisons between different route planning iterations to select the most effective approach. Although, assigning weights without stakeholder validation may misalign system priorities.

Finally, the **Quality Function Deployment** converted stakeholder requirements into engineering specifications, ensuring that system design features addressed user priorities and reducing the likelihood of rework in later stages. But incorrect relationship can misallocate resources to less critical feature.



Development stage

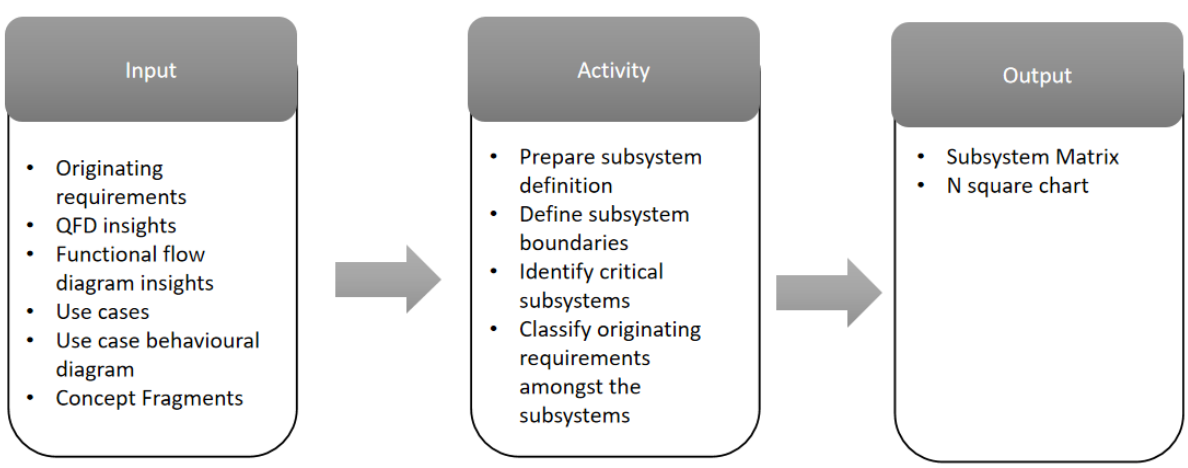
The Development Stage focuses on transforming the system concept into a detailed, operational model by defining subsystems, their interactions, workflows, and interfaces. A critical part of this stage is ensuring that each subsystem, Model Management, Route Planner, Data Management, and Navigation, is well defined, interacts correctly, and supports operational requirements such as computing optimal routes to minimize solar exposure. The stage also aims to validate workflows and data exchanges to achieve a cohesive, robust, and maintainable system that meets both functional and operational goals.

Architecture process

The architecture process focuses on defining the system's high-level structure, establishing subsystem responsibilities, and specifying interfaces to meet functional and operational requirements. An important goal of architecture was organizing subsystems to minimize the connection between subsystems and create more cohesive subsystems. The architecture process also aims to find the critical subsystem based on interactions dependency on particular subsystem.

In our project, the **subsystem diagram** aided us in the goal of dividing given system in multiple clear subsystems. Defining clear subsystems like model management, route planner, Data management and Navigation enabled attaining the goal of maximum cohesion it is achieved by combining online and offline data management into single subsystem data management. Incomplete diagrams can result in ambiguous responsibilities and reduced cohesion of the system. Not being able to define subsystems well might cause a situation where to change single thing in one subsystem needs to make changes in all subsystems increasing time required.

Also, the **N2 chart** aided us in the goal of identifying critical subsystem based on interaction. The N2 chart provides systematic view of all subsystems interfaces showing which subsystem sends and receives data from each other hance identifying the critical subsystem. Poorly constructed N2 charts can lead to misjudgment in identifying critical subsystem.



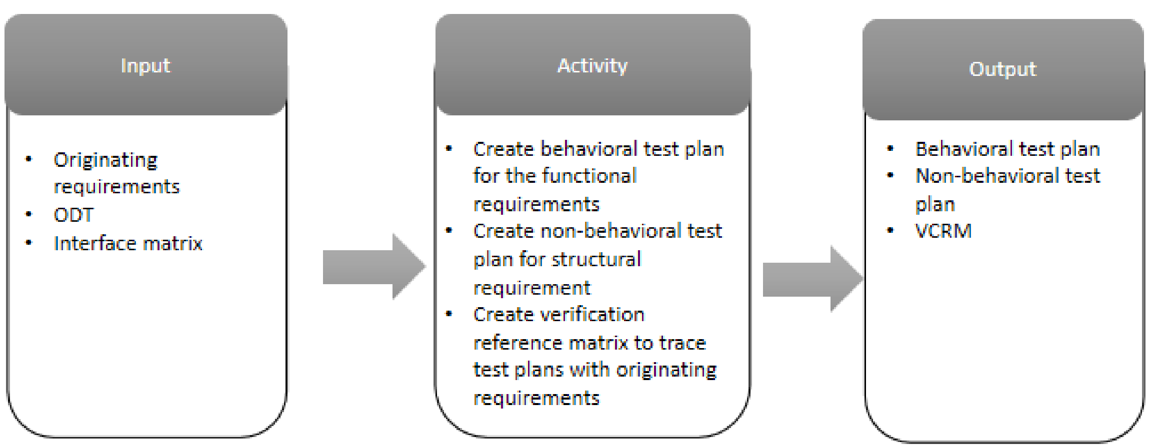
Verification Process

The Verification Process focuses on checking whether the Solar Route Planner system has been *built correctly* according to the requirements, architecture, and design. The goal is to ensure that subsystem interactions, data flows, route computation logic, and triggering mechanisms behave exactly as defined. Verification confirms that every requirement (functional and non-functional) has a corresponding test activity and that the system implementation does not deviate from what was planned. The process checks expected behaviors such as data acquisition method and timing, proper invocation of route recalculation, and consistency in data formats before integration. Essentially, verification ensures correctness of implementation before evaluating real-world usefulness.

In our project, the **Behavioral Test Plan** supported the verification goal of checking whether each subsystem performed its expected actions during route computation. For example, the tests ensured that the Route Planner switches to offline mode when real-time data is not available. These behavioral tests verified that internal workflows matched the designed sequence diagrams. Poorly developed behavioral tests may miss conditional flows, leading to late discovery of logic defects.

The **Non-Behavioral Test Plan** aided the goal of verifying system qualities such as reliability and safety in route recalculation, by ensuring appropriate memory utilization during data storage, data latency rate and maintaining data synchronization speed. This helped confirm whether the implementation delivers consistent performance in different conditions. Weak non-behavioral testing can cause hidden performance issues to appear during real-world operation.

The **Verification Cross-Reference Matrix (VCRM)** supported the verification goal by mapping every requirement to its corresponding test activity. This ensured full coverage of requirements and prevented untested features. Without a correct VCRM, some requirements may remain unverified or be incorrectly tested, leading to integration mismatches.

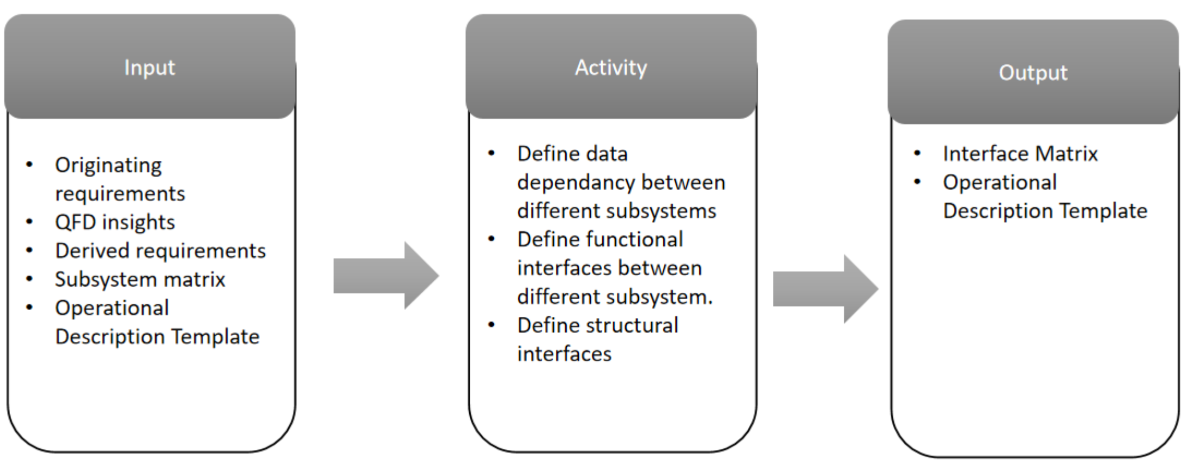


Integration Process

The Integration process focuses on assembling the individual subsystems into a functioning system, verifying that design and operational interfaces work correctly, and ensuring the system meets functional and operational goals. A critical part of integration is identifying what kind of design constraint or design data dependency is amongst the different subsystems produces reliable route predictions under varying solar exposure conditions. Another key goal is identifying functional interfaces and decision-making points for different use cases. The Integration process also aims to ensure system level robustness and modularity, so that changes in one subsystem do not create cascading effects in others.

In our project, the **Interface Matrix** aided in the integration process goal of defining and verifying the design data exchanged between subsystems. For example, the Model Management subsystem team decided the maximum number of intermediates stops the model should handle, and the other subsystems, such as Route Planner and Navigation, were designed to support only that number of stops. This ensured consistent expectations across subsystems and minimized design mismatches. Poorly maintained interface matrices can result in misaligned subsystem designs, incorrect assumptions, or runtime errors, since one subsystem may expect data in a format or quantity that another subsystem does not provide.

The **Operational Description Template** aided the integration process goal of identifying functional interfaces and decision points for different use cases. While creating the ODT, we realized that even though we are using online data, we still need to temporarily store infrastructure and weather data, which in turn highlighted that the RAM size requirement should be captured in the interface matrix. The ODT also helped us define different decision-making points for each use case, ensuring that subsystem interactions support operational requirements under varying conditions. Missing or incomplete ODT documentation can lead to integration errors or unhandled operational scenarios, as teams may misinterpret how subsystems should interact or which decisions need to be made at runtime.

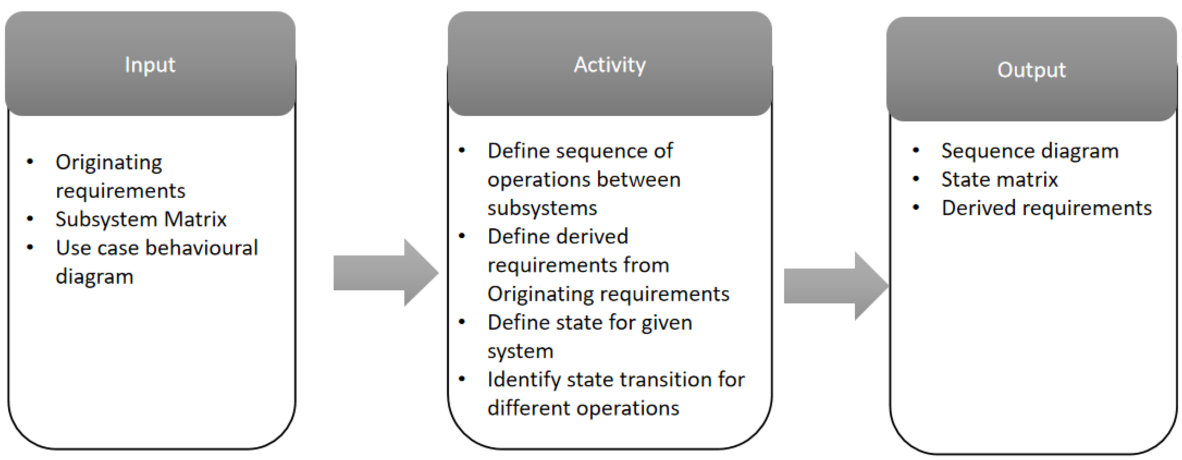


Design Process

The Design process focuses on detailing how different user processes will be implemented including algorithms, workflows and data handling (e.g. before recalculating route based on new data whether system will check the changes in data or not). The important part of the design stage is to define sequence of operations and corresponding interactions between the subsystems so that process can be divided into smaller chunks of work pockets which can be integrate together later. Another crucial goal of design process is to define subsystem state transitions to define the system behavior under different operating conditions.

In our project, the **sequence diagram** aided the design process goal of define sequence of operations and corresponding interactions between the subsystems for route computation. The sequence diagram defines for a given operation which subsystem needs to perform which actions to get the desired operations. It helps classify originating requirements amongst the defined subsystems. (e.g. data access requests will be handled by data management). Poor sequence diagram can miss some important steps in execution which then later can cause ambiguity and integration problems amongst the subsystems.

Also, the **state matrix** aided the design process goal of defining subsystem state transitions for route optimization model by mapping how system stages are transitioning in response to different originating requirements (e.g. availability of updated weather data will turn route planner model into data acquisition mode). It defines the context for given originating requirement as route updating context needed to be run when user is already in navigation mode. Poorly constructed state metric can lead to unhandled states or incorrect transition leading to unpredictable behavior of system during execution.



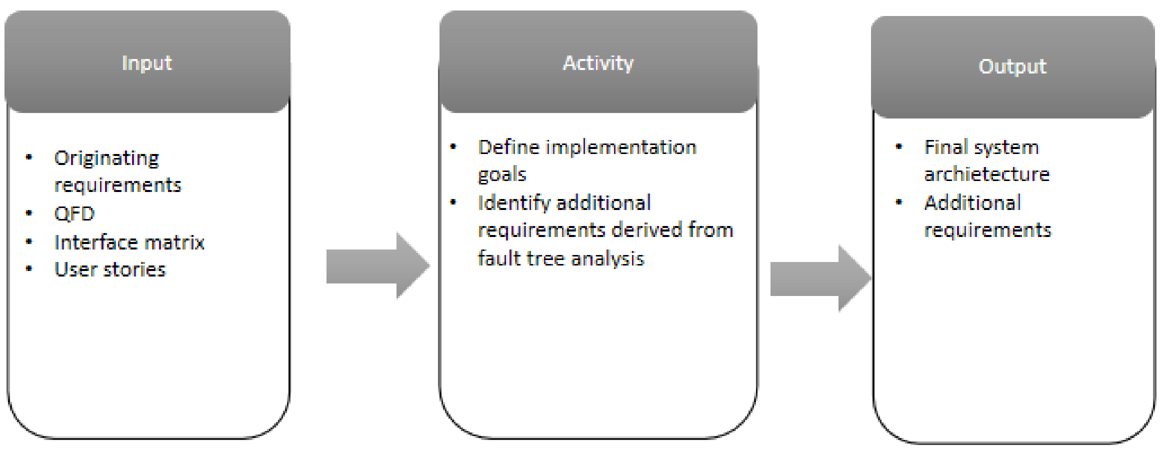
Implementation Process

The Implementation Process focuses on producing the actual working software components from the approved designs and integrating them into a functioning system. The goal is to ensure that code implementation aligns with design decisions, that subsystem interfaces are correctly realized, and that integration does not introduce new faults. The process involves transforming design constraints into working modules, mapping requirements to implementation features through QFD, and iterating on subsystems that show higher failure likelihoods based on fault tree. The implementation stage aims to maintain consistency, repair weak design areas, and deliver a robust system ready for final verification and validation.

In our project, Quality **Function Deployment (QFD)** supported the implementation goal by mapping user needs to technical features. This ensured that implementation decisions such as solar minimization logic, data refresh frequency, or navigation guidance were aligned with user expectations. QFD helped ensure that coding efforts prioritized user-critical features rather than internal assumptions.

The **Interface Matrix** supported the implementation goal by ensuring that the actual code-level interfaces followed the defined data formats, units, and expectations established during design. For example, implementing the correct maximum number of intermediate stops required by Model Management prevented mismatch errors in Route Planner and Navigation. It also helped to maintain the memory usage during calculation. Poorly maintained interface matrices can lead to interface incompatibilities during integration.

The **Fault Tree** also supported implementation by identifying components or interactions that were more failure-prone. During implementation, these areas were redesigned, strengthened, or simplified. For example, in route planner GPS accuracy is increased by adding dead reckoning system which increase robustness of the system. Hence components with higher failure contribution were improved before deployment.

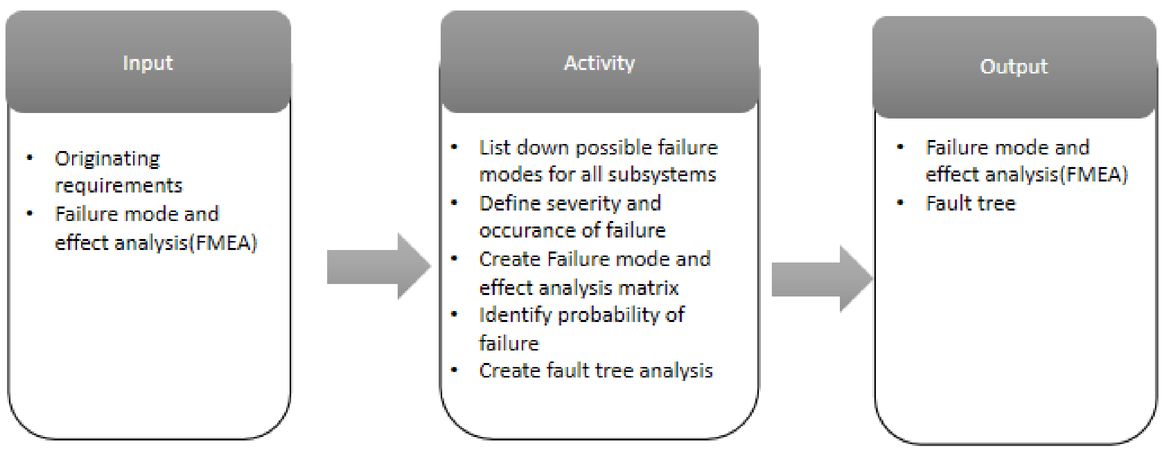


Validation Process

The Validation Process focuses on ensuring that the *right product is built*, meaning that the Solar Route Planner actually performs satisfactorily in realistic operating conditions. The key goal is to confirm that the generated route truly minimizes solar exposure, adapts sensibly to weather changes, and supports user needs during navigation. Validation process also ensures robustness of system in real life scenarios such as inadequate data, changing weather, lost signals, or unexpected traffic. The overall aim is to ensure the system's real-world performance is safe, reliable, and acceptable for end users.

In our project, the **Failure Mode and Effects Analysis (FMEA)** supported the validation goal by identifying where the system may fail during real-world operation. For example, failure modes like “outdated data leading to unreliable routing” or “incorrect GPS signal” were evaluated for severity and detectability. This allowed the team to check whether the system remained operationally acceptable despite possible failures. Without FMEA, validation may ignore critical real-world vulnerabilities.

The **Fault Tree** supported the validation goal of identifying which subsystem or interface contributes most to user-level failures. For example, in our scenario the most probability of failure is because of the incorrect GPS signal which originates from route planner subsystem. The given analysis helped team to identify focus areas for improvement to build more robust system. A poorly constructed fault tree can misidentify fault sources and lead to ineffective fixes.



Production Stage

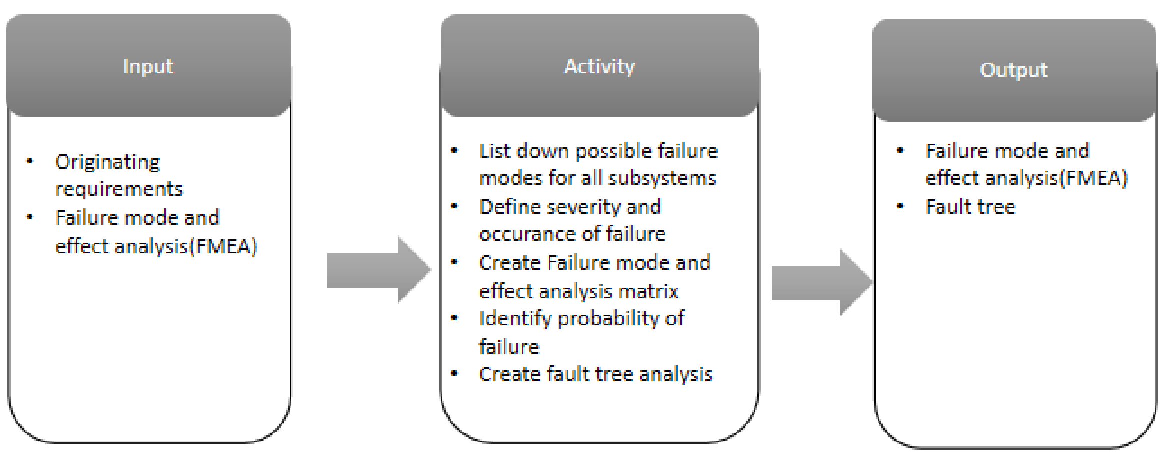
The production stage focuses on deploying the solar route planner with high reliability, efficiency, and consistency while minimizing defects and resulting waste. The key goals of this stage are to manufacture the required quantity of product on time, which meets specified standards and customer requirements.

Validation Process

The validation process confirms that the manufactured solar route planner product meets all the functional and non functional requirements in realistic operating conditions. Key goals for this process in production stage include verifying accurate solar exposure calculation , real-time route updates, offline functionality and integration with vehicle hardware. Validation process ensures the system operates safely, reliably and consistently across every manufactured model and unexpected conditions like GPS signal loss or incomplete data.

In our project, the **Failure Mode and Effects Analysis** supported the validation goal by identifying where the system may fail during real-world operation. For example, failure modes like "outdated data leading to unreliable routing" or "incorrect GPS signal" were evaluated for severity and detectability. This allowed the team to check whether the system remained operationally acceptable despite possible failures. Without FMEA, validation may ignore critical real-world vulnerabilities.

The **Fault Tree** supported the validation goal of identifying which subsystem or interface contributes most to user-level failures. For example, in our scenario the most probability of failure is because of the incorrect GPS signal which originates from route planner subsystem. The given analysis helped team to identify focus area for improvement to build more robust system. A poorly constructed fault tree can misidentify fault sources and lead to ineffective fixes.



Utilization Stage

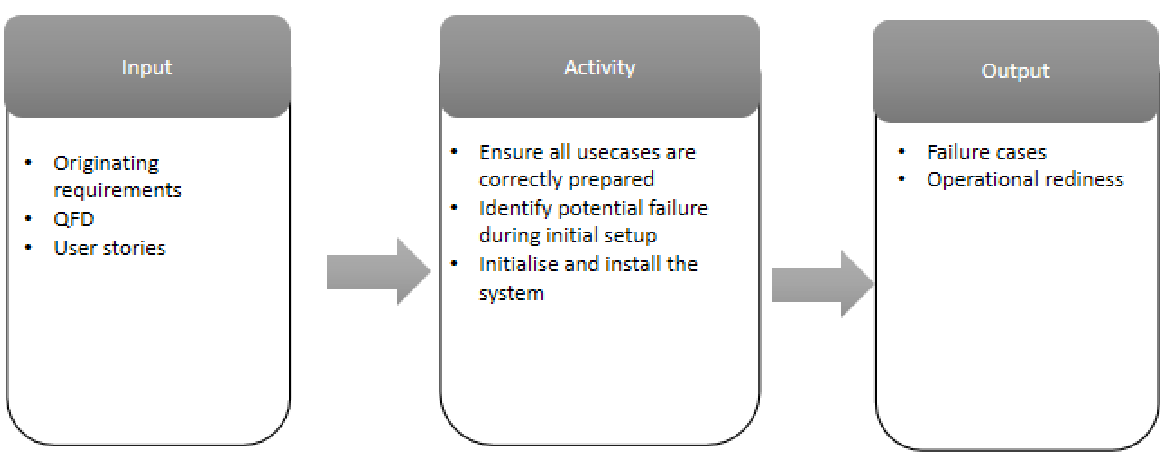
The Utilization Stage focuses on deploying the Solar Route Planner into its operational environment and ensuring it performs effectively for end-users. The primary goal is to ensure smooth transition from development to active use, with all subsystems, Model Management, Route Planner, Data Management, and Navigation, functioning correctly in real-time conditions. The stage emphasizes maximizing system availability, reliability, and accuracy of route predictions while minimizing downtime or operational errors.

Transition Process

The Transition Process focuses on moving the Solar Route Planner system from development into actual operational use. The main goal is to ensure that the system can be successfully deployed, installed, configured, and made usable for end users or operational teams without introducing failures or unexpected behavior. This includes preparing deployment, migrating required offline datasets, ensuring compatibility with target vehicle hardware, and validating installation procedures. Another critical goal is training or orienting users so they understand how route updates, how to set customizable parameters , and data synchronization function. The transition stage ensures the system leaves the development environment smoothly and enters an operational environment with clear stability and reliability.

In our project, the **Operational Description Template** supported the transition goal of ensuring that each use case was correctly prepared for deployment. The ODT documented how the system should behave when first launched, how data needs to be preloaded, and what the system must check before computing routes. This helped define what permissions and data preparation steps must be performed during deployment. Missing ODT entries can cause incorrect assumptions about operational readiness, leading to incomplete installations.

The **Fault Tree** aided the transition goal of identifying potential points of failure during deployment or initial setup. For example, a failure such as “unable to plan route” was traced to improper data initialization or missing offline infrastructure data. Using fault tree ensured that deployment instructions included checks to prevent such failures. Without Fault tree, failures might only appear after the system is already live.

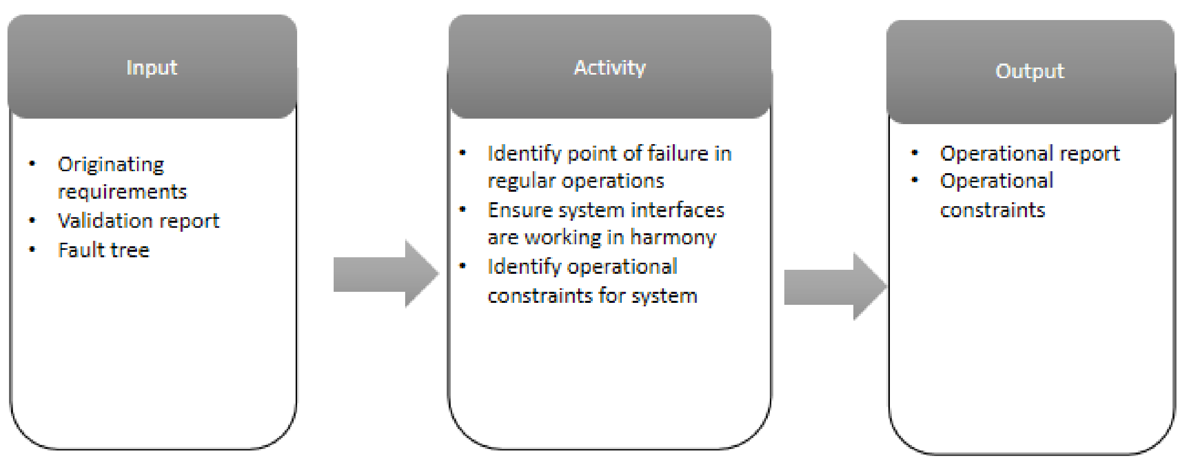


Operation Process

The Operation Process focuses on ensuring that the Solar Route Planner continues to perform safely, effectively, and consistently during everyday use. This includes monitoring system performance under real conditions, handling real-time weather changes, ensuring updated route computations are meaningful, and supporting continuous access to necessary data sources. A key goal is to maintain system usability and robustness while minimizing failure. The operation process also ensures that emerging issues are detected early, routed to maintenance teams, and do not degrade the user's experience. The aim is to deliver stable performance, uninterrupted route guidance, and reliable solar-exposure optimization in the field.

In our project, the **Failure Mode and Effects Analysis** assisted the operation goal by identifying failures that might occur during regular system use and assessing their impact. Examples include inaccurate weather data causing incorrect solar predictions, slow GPS updates affecting position accuracy, or a sudden delay in model computation. FMEA ensured that high-risk operational failures were monitored and that fallback mechanisms were defined. Inadequate FMEA increases the chance of the system failing in ways users cannot recover from.

The **Interface Matrix** also supported operation by ensuring that ongoing data exchanges (e.g., weather data, infrastructure updates, GPS orientation) continued to match the formats and constraints defined during design. During operation, mismatched or corrupted data interfaces are one of the most common sources of failure; maintaining updated interface definitions reduces this risk.



Support Stage

The primary goal of the Support Stage is to ensure the Solar Route Planning system continues to operate reliably, efficiently, and safely throughout its operational life. This stage focuses on updating components to adapt to evolving user needs or environmental conditions, and maintaining alignment with stakeholder expectations. Additionally, the Support Stage aims to implement improvements to maximize system availability, user satisfaction, and overall lifecycle value.

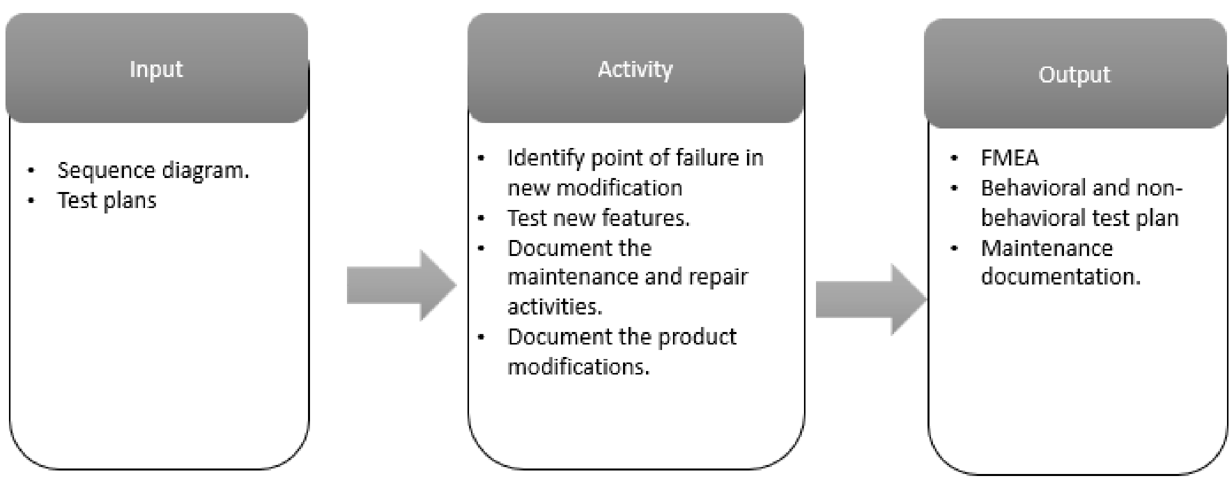
Maintenance Process

The Maintenance process focuses on ensuring the long term reliability, operability, and performance of the Solar Route Planning system. Key goals include monitoring system performance while in operation, identifying and resolving defects, updating components to address evolving environmental or user requirements, and maintaining alignment with stakeholder expectations. This process ensures that the system continues to deliver value over its operational life while minimizing downtime and maximizing user satisfaction.

In our project, **FMEA** can be used to assess the risk of new changes ensuring systems robustness and reliability. Although poorly executed FMEA might not be able to capture risks.

Behavioral Test Plans and Test Methodologies for Non-Behavioral Requirements provided a structured approach for verifying updated system functionality and performance during maintenance, enabling early detection of deviations or failures. Although poorly executed test can lead to the failure of system during operation affecting reliability of product

Verification Cross Reference Matrices ensured that any updates or corrective actions remained aligned with original requirements, preventing regressions or unmet user needs.



Retirement Stage

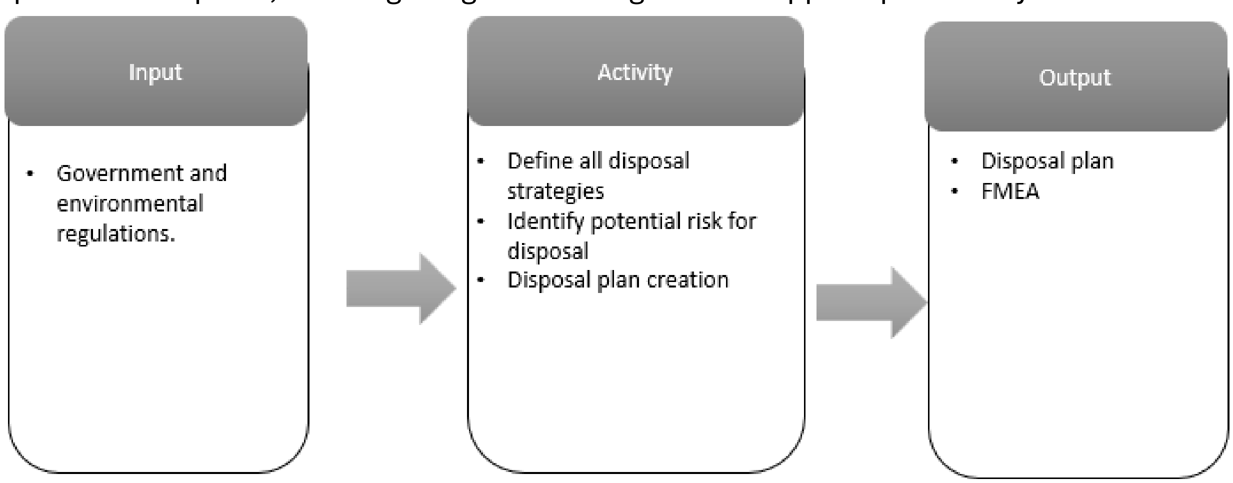
The primary goal of the Retirement Stage is to ensure the systematic, safe, and traceable decommissioning of the Solar Route Planning system at the end of its operational life. This stage focuses on preserving critical data for future reference, safely disposing of or repurposing system components, and mitigating environmental, operational, and security risks. Additionally, the Retirement Stage aims to capture lessons learned, document system performance over its lifecycle, and provide insights to inform the design and deployment of future systems.

Disposal Process

The Disposal process focuses on systematically decommissioning the Solar Route Planning system at the end of its operational life while ensuring minimal environmental, operational, and data risks. Key goals include safely retiring system components, preserving critical data for future reference, complying with regulatory and organizational guidelines, and capturing lessons learned to inform future system developments. This process ensures that disposal activities are conducted in a controlled, traceable, and safe manner, mitigating potential negative impacts.

In our project, **Decision Matrices** were used to evaluate alternative disposal strategies, considering factors such as environmental impact, cost, regulatory compliance, and operational continuity, ensuring an objective selection of the most appropriate method.

Failure Mode and Effect Analysis supported identification of potential risks associated with decommissioning, including data loss, hardware failures, or unintended operational impacts, allowing mitigation strategies to be applied proactively.



Closing Remarks

Through the application of systems engineering methodologies through project lifecycle, we systematically made sure that technical decisions remain aligned with the customer expectations, stakeholder needs and practical operational environment. The structured use of tools, models and lists enables traceability, robustness and informed decision making.

The project demonstrated the value of structured and methodological approach in systems engineering resulting in solutions which is technically sound and user centric. This report can build strong foundation for future enhancements and potential real-world implementations.