**Requirements Management Plan**

Document Summary

Provides essential information about requirements management processes to all project stakeholders. Establishes a controlled requirements management process with requirements repository, traceability, requirements attributes, requirements prioritization, and change management. Covers the whole requirements life cycle from initiation to closure.

Definitions

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| BA | Business Analyst |
| PM | Project Manager |
| QC | Quality Assurance Engineer |
| US | User Story |
| AC | Acceptance Criteria |
| SP | Story Points |
| SO | Service Owner |
|  |  |

Distribution (optional)\*

**Client Stakeholders**

|  |  |  |
| --- | --- | --- |
| **Full Name** | **Position (Role)** | **Email** |
| Oleh Herhil | Service Owner | oherh@[softserveinc.com](http://somecompany.com/) |

**SoftServe Stakeholders**

This project is volunteering project, it has a specific point concerning participants: 95% of participants are students of IT Academy, participate in SoftCar project temporary during their course at Academy and develop their skills at the project ~2-3 months. Other team members are mostly SoftServe associates. Project is positioned as opensource, so external participants may join the project.

As Dev and QC Team is constantly changing, it doesn’t make sense to include whole list of participants into the table, but leads should be included.

|  |  |  |
| --- | --- | --- |
| **Full Name** | **Position (Role)** | **Email** |
| Nataliya Revutska | Project Manager/Dev Lead | nrevu@[softserveinc.com](http://somecompany.com/) |
| Ihor Kohut | QC Lead | ikohu@softserveinc.com |
| Olena Bobalo | BA | oboba@softserveinc.com |
| Khrystyna Vashchuk | BA | kvash@softserveinc.com |
| Roman Dovnich | DeOps | rdovn@softserveinc.com |
|  | Designer |  |

**RACI matrix**

RACI Matrix is used for describing the roles:

* R, Responsible — owns the task/area, does the work to achieve it;
* A, Accountable — signs off the task/area, answerable for the correct and thorough completion;
* C, Consultant — subject matter expert, has information necessary to complete the task;
* I, Informed – has to be notified of results or changes, often only on completion of the task/area.

Table uses these definitions (DL - Development Lead, D- Designer, SE - Developer, QC - Tester)

| **Tasks/Area/Process** | **BA** | **SO** | **PM/**  **DL** | **SE** | **QCL** | **QC** | **D** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Backlog RMP Cost-Value Prioritization | A | R | CI |  | CI |  | I |
| US Mockups Creation | C | I | CI | I | I | I | R |
| User Story and Backlog Scope Elicitation | R | I | CI | CI | I | CI | I |
| Situation Analysis (Organization Strategy) | CI | RA | CI |  | I |  |  |
| BA Planning and Monitoring | RA | I | CI |  | I |  |  |
| RMP Requirements Elicitation | R | I | R | I | CI |  | I |
| Requirements RMP Review and Analysis | R | I | R | I | CI | I | I |
| Solution Definition | RCI | RA | CI | CI | I |  | CI |
| Solution Evaluation | CI | RA | CI | I | I |  | CI |
| Functional Requirements Risks Management | RA | RC | RC |  | RCI |  |  |
| Release RMP Delivery Package | R | CI | R A | CI | R | CI |  |

### Stakeholder responsibilities

Only BA may change Epics/US in GitHub. PM approves specifications and the US along with the acceptance criteria for development and accepts the implemented functionality.

| **Role / Areas** | **Project Management** | **Business Analysis** |
| --- | --- | --- |
| **Scope of work** | * Project Scope (how will the solution be created) | * Solution Scope (what is needed by the business) |
| **Outputs** | * Project management plans (e.g. time, budget, resources, procurement) * Task assignments * Management of project resources * Status reports of project progress | * Business and stakeholder requirements * Solution requirements (functional and nonfunctional) * Transition requirements |

### Scope

Two types of scope are critical for the understanding of project management and business analysis role delineation: **project scope** and **solution scope**. Project Scope is the work performed to deliver the result or solution. Solution scope (product scope) is a set of capabilities a solution must deliver in order to meet the business need.

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Project Initiation Process Group** | **Project Planning Process Group** | **Project Executing and Controlling Process Groups** |
| **PM** | * Assess project feasibility * Create measurable objectives * Uncover initial assumptions, risks * Identify stakeholders | * Determine team * Develop activity list, time estimates, budgets, and schedule * Create the project management plans | * Follow processes, facilitate conflict resolution, report progress * Measure performance against the plan |
| **BA** | * Verify completion of business case and business objectives * Elicit high-level business, stakeholder and solution requirements * Determine solution approach and define solution scope | | * Elicit and analyze detailed functional, nonfunctional, transition requirements * Work with technical team members to design the solution based on requirements * Validate solution against original business needs |

References

|  |  |
| --- | --- |
| **Name** | **Link** |
| Docs repository | <https://github.com/ita-social-projects/Car-Front-End> ;  <https://github.com/ita-social-projects/Car-Back-End> |
| App Designs | <https://www.figma.com/file/q2QTrUQLmisr90FzvU7HLm/Request-Mockup?node-id=42%3A53&viewport=246%2C289%2C0.1865042895078659> |
| Work Break Down Structure |  |
| Business Requirements Docs |  |

Project Overview

Project requirements are maintained through different issue management, incorporating information from emails, conference calls, chats, and meetings.

Due to the Agile Scrum process selected, BA and PM are more focused on the frequency and quality of communication than on formal documentation processes. Project communications are conducted according to the referenced team communication plan. Additionally ad-hoc or on-demand communications are held if needed.

All project’s documentation is placed at GitHub. App Designs are placed at Figma.

Main source of team communication is Zoom, for chatting – Discord. Main source of communication with Service Owner is Teams.

SDLC Process

As SoftServe Car App project is a social project with volunteering participation and IT Academy students’ participation, there are no terms of releases dates.

So, dates and terms for each stage of SDLC aren’t defined.

Requirements Format

**Types of Requirements**

The requirement is a usable representation of a problem, opportunity, or constraint with potential value to a stakeholder.

Requirement Types (based on BABOK):

* **Business.** A high-level statement of the goals objectives or needs of the enterprise.
* **Stakeholder.** The statement of needs of the particular stakeholder describes how this stakeholder will interact with the solution.
* **Solution.** Elaborates on business and stakeholder requirements describing the characteristics of the solution.
  + **Functional.** Behaviors or capabilities that the system will perform.
  + **Non-Functional**. Specifies the environmental condition that a solution must operate under or qualities that a solution must have to be successful.
* **Transition.** Capabilities that a solution must have in order to facilitate the transition from the current state to the desired future state.

**Requirements Representation**

On this project, all requirements are represented in the form of User Stories (Stakeholder and Solution requirements) and Epics (Business Requirements). Epics group User Stories into features.  
User Story may contain 9 sections:

* the **User Story** statement,
* **Description/Mockup** (brief description of function described in US/image of the page/screen described in US) (optional)
* **Assumptions & Constraints** (information specific for each story that influences decisions on story criteria) (optional),
* **Acceptance Criteria**,
* **Notes** (useful information which is not acceptance criteria) (optional),
* **TBD** (items without which the story is not ready and finalized; this section is empty when all issues and questions are resolved) (optional).
* **Epic Link** (link to the story with the list of US which belongs to that Epic).
* **Labels** to be added (list of labels which are added to this US by team).
* **Tasks** (list of tasks which are added to this US by QCs).

|  |
| --- |
| As a <role> I want <goal> so that <benefit>.  \*\*Description\*\*  Describe feature functionality  Add mockup if needed  \*Acceptance Criteria\*  A clear and concise description of what you want to happen  1. Testable criteria.  \*Assumptions & Constraints\*  1. Listed assumptions, constraints, and dependencies relevant to this specific  story.  \*Notes\*  1. Non-acceptance items including technical information for developers.    \*TBD\*  1. List all existing blockers, PO review state, estimation needs.  \*\*Epic link\*\*  E.g.: Epic #100 [Epic](https://jira.softserve.academy/browse/100)  \*\*Labels to be added\*\*  "User story" ("story"), Priority ("pri: ")  ### Tasks  1. - [ ] sample task. |

**Requirements Attributes**

| **Attribute** | **Description** |
| --- | --- |
| ID | Unique alphanumeric identifier for each requirement. Not to be altered or re-used if the requirement is moved, changed or deleted. |
| Author | Author of the requirement if there would be any questions later. |
| Estimate | Difficulty in implementing the requirement, indicates how difficult the requirements will be to implement. Indicated through qualitative scale: 1, 2, 3, 5, 8, 13. |
| Assigned To | Indicates the individual that owns the requirement implementation. |
| Priority | **Critical** – the most important and critical features,  **High** – features that will most likely be available in the application, but that aren’t necessarily critical to system function,  **Medium** – a feature that is not critical nor necessary, but that would be “nice to have”,  **Low** – tasks that will not be available or completed in the current version of the application. |
| Status/State categories | **Open** – new US, in the process of development,  **To Do** – story is discussed and ready for development (could be assigned to states associated with newly added work items),  **In Progress** – story is assigned to associate and is in the process of development,  **Ready for Testing** - a solution has been implemented, but isn’t yet verified by QC,  **Done** – work at that item is finished.  **Closed** – work at that item is finished, if any discrepancy with US is found, the bug issue shall be created. |
| Issue type | Task, Subtask, Test, Story |
| Note | **Improvement** – stories which includes comments with improvements,  **On Hold** – US which can’t be developed now by some reason,  **Clarification** – task which isn’t totally clear and need additional discussion/explanation; should include comment with question,  **Release 2** - tasks that will not be available or completed in the current version of the application |

RM processes

This section describes the main processes related to the requirements management area in detail that exist and are utilized in the project. It helps to understand and visualize what each of the processes consists of, its inputs and outputs. This section also serves as a guide for the BA and all relevant stakeholders.

Planning

As it was mention above, as there are no permanent team members, and all works are fulfilled by the temporary students of IT-academy and volunteers from SS side, there are no actual plans of project’s SDLC.

Planning process is executed according of quantity of available students and their skills, volunteers from SS side and demand from Service Owner.

PM is responsible for weekly planning sessions.

According to MVP tactic product for first release shall include such list of functions: Create a Ride, Search for a Ride, Address Book, Car list, Notifications, Chats.

Scope definition and new functionality requests

**Backlog**

BA and PM are responsible and authorized to add the details to the backlog in the form of US with acceptance criteria, aligned with the service owner's vision and stakeholder's needs. Acceptance criteria represent the functionality usage scenarios for verifying that the changes are complete.

**Epics**

The big chunks of functionality, called Epics, should be defined prior to adding user stories to the backlog. After creation, they should be split into smaller parts, User Stories (US), for more precise estimation and control. Hierarchical representation of Backlog items should be held in order to keep traceability from high-level requirements to low level ones and be able to calculate requirements implementation progress. It is held in a form of linked work items in GitHub tool.

Epics that exist in the product backlog solely for hierarchy (have no estimated US) should have an empty SP estimate field. Epics CAN’T have the “Close” status.

**User Stories**

User Stories should be added to the backlog. If there is a need to create User story for certain functionality in case when mockup is absent and designer position is vacant, BA shall make a detail visual description in AC.

Prioritization

SoftServe BA manages the backlog priority within the specific iteration and releases according to team suggestions, aligns the priorities with SO.

SO is responsible for setting the priorities in the backlog and release prior to starting the development process. Prioritization should be done by SO before the planning so that efforts can be focused on the most important user stories first. Changes in the priorities are reviewed at the planning session. Estimates should be taken into account for prioritization as they affect the cost-value ratio of the work item. If items are not estimated in advance, additional estimation sessions should be initiated by the PM or BA.

The features that have high business value and require low efforts for implementation should have higher priority. All unplanned improvements shall be discussed with SO and priorities shall be indicated according to Business value. The majority of technical complexity should also get higher priority because of the related risk.

Requirements Elicitation

There are a few main elicitation techniques used on the project. Since the product already exists and customers are not available all the time one of the main elicitation techniques is document analysis.

Analysis and Review

BA and PM facilitate the team to identify multiple solution options. They also specify the advantages and disadvantages of the solution and proposes the best solution approach. Service Owner makes the final decision about which the solution option to implement.

Reviews are used to evaluate the content of a work product. (e.g. peer review, inspection, team review).

Traceability

Requirements traceability identifies and documents the relationship between the requirements, including its backward and forward traceability.

Requirements may be traced to other requirements (including goals, objectives, business requirements, stakeholder requirements, solution requirements, and transition requirements), solution components, visuals, business rules, and other work products.

It is important to track and describe dependencies in a form of lined items in GitHub during regular daily standups. BA should identify and address dependent requirements, technical interfaces, and implementation priority conflicts and provide resolution for them by re-prioritizing or creating the new requirements.

Estimating

The SoftCar project team should clarify all the details needed to estimate the relative size of the backlog items. Such details can be gathered in a backlog review meeting, planning meetings, phone calls, Discord chatting, by email, or with the help of a work item tracking tool.

As sprints aren’t used in this project, US estimation sessions could be fulfilled during daily meeting and could be initialized by BA or PM.

Estimation results (SP: 1, 2, 3, 5, 8, 13) should be noted into each estimated US.

If SP for US is higher then 13, US shall be decomposed.

**Definition of Ready**

**Definition of Ready for starting User Story implementation**

* User Stories are created, communicated, and estimated.
* All acceptance criteria are discussed with the team and testable.
* Mockups are attached to the story (where appropriate).

Acceptance

Acceptance is performed by the SoftCar BA, PM, SO, and other stakeholders.

Each story needs to pass internal QC acceptance first. Acceptance testing verifies that stories were developed such that each works exactly the way the customer team expected it to work. BA evaluates the business and stakeholder requirements implementation to conform to the user story description and understanding.

Before the release, a demo is held for all the stakeholders and the PM should accept the result or give feedback on what should be changed. Feedback may produce new backlog work items (tasks and stories). Minor tasks should be fixed during the next week unless the other is agreed by stakeholders during a demo.

Demo notes with appropriate action items are added and stored at the project Confluence.

**Definition of Done**

**Definition of Done for a User Story**

* All tasks (development, unit testing, manual testing, UI forms, documentation) for the user story are defined and completed (closed).
* All bugs related to the changes introduced by the new user story are closed. Exceptions are approved by PM and added to the backlog if needed.
* All acceptance criteria are met, including: UI acceptance criteria, story criteria with listed assumptions.
* Regression tests are completed.
* BA and PM have validated (accepted) the story from the business needs perspective and updated the story with appropriate status or comment.

**Definition of Done for a Release**

* Planned user stories are accepted and closed.
* Documentation is reviewed and accepted.
* Deployment strategy isn’t defined yet.

Change Management

CMP isn’t defined yet.

Tools

Visio

GitHub User Story Editor,

GitHub Projects,

Sonar Cloud

Discord

Zoom

Teams