Computer Science and Engineering

Good Deed Project

Project Management Plan

Version 1.0

Document Number: SPMP-001

Project Team A14

Project Team Members:

Jonah Sherman, jds854
Kerem Ulcay, ku350
Sofia Machado Lopes, sml856
Timur Gordon, tbg252

REVIEW AND APPROVALS

<Team Members>	Function (Author, Reviewer, Approval)	Date	Signature
Jonah Sherman	Author, Reviewer	11-05-20	JDS
Sofia Machado Lopes	Author, Reviewer	11-05-20	SML
Timur Blair Gordon	Author, Reviewer	11-05-20	TBG
Kerem Ulcay	Author, Reviewer	11-05-20	KU

REVISION LEVEL

Date	Revision Number	Purpose
11/05/2020	Version 1.0	Initial Release

Table of Contents

1 OVERVIEW

1.1 Project Summary

The motivation for the project is to support the development and execution of good deeds, through the use of an online social platform. The need for such a project is evident. There are a lot of

problems that the world faces, and with the collaboration of individuals with a shared motivation, these problems could be tackled through social entrepreneurship. The platform thus aims to empower people who aim to do good in this world, help these people meet others that are willing to contribute to their projects, and provide a tool to get a hold of the resources they may need for the execution of their projects.

The purpose of this software project management plan is to outline plans for management processes, technical processes, and supporting processes that are involved in the implementation of this project. Furthermore, this SPMP aims to deliver information on project organization, schedules for the plans and the overall project, and the budget of the project.

The intended audience of the SPMP is Professor Strauss, a client, or any designer or developer who is to assist in the software delivery lifecycle process, as well as staff and people working on the implementation of the project.

1.2 Purpose, Scope, and Objectives

Within the scope of the project, one should be able to join a specific group under an organization to offer their talents to the organization. One should be able to communicate with others on the platform. One should voice their opinions freely without any fear of authority or punishment but with recognition that they are working towards a common goal. This service will offer good options based on a person’s highlighted skills. This will include serving their interest as well as serving better with their limited time. We recognize the value of time and decreasing online presence costs for NGOs. For this offered decrease in costs and increase in impact we expect donations to cover our costs. We will strive to offer making donations directly through the site to good deed projects. The scope of the project excludes any polarizing projects on controversial and political issues. The scope also excludes any external advertising on the platform.

The release must satisfy business needs such as the capability of registering and logging in to the system, easy project creation, donation transfer to projects, and secure data management.

1.3 Assumptions and Constraints

The project is based on the assumption that a web application is the best software medium to achieve the purpose of the project, assuming that more people can access the platform if it is accessible simply by a browser. It is also assumed that the platform can be used on mobile devices without the necessity of having to create a mobile application tailored to each device. Furthermore, it is assumed that the users have browsers that are up to date and that can support displaying the platform. Another assumption is that there will be enough demand for the platform such that many good deed projects can exist on the platform as well as volunteers and donors, thereby providing users with the content they desire and a successful impact ecosystem.

Externally imposed constraints on the project include primarily scheduling and resource constraints as mentioned in the schedule and budget summary. The constraints limit the functionalities that can be included in the first release, and the platforms that the product can be released on. Therefore, another constraint is that the software product will be constrained to a web application.

1.4 Project Deliverables

The product will roll out in two stages. The initial product / prototype will be designed, implemented and launched prior to the design of the final product. The feedback received from the initial product will be incorporated into the design of the final product. The anticipated milestones and dates for product development and launch are as follows:

Name of Deliverable	*Date*	(Format**)**
Completion of SRS	*17 November 2020*	(PDF)
Project Presentation	*15 December 2020*	(PPT)
Initial product design	*1 January 2021*	(Adobe XD)
Initial product implementation	*1 February 2021*	(Github Repo)
Final product design	*1 March 2021*	(Adobe XD)
Final product implementation	*1 April 2021*	(Web Application)
Final product evolution & maintenance	*1 May 2021*	(Github)

1.5 Schedule and Budget Summary

Completion of SRS *17 November 2020* (PDF)

Regrade request
Reworking on the submitted document, for better understanding of fault
Rendering the feedback in.
Finalizing the new version of SRS.

Project Presentation *15 December 2020* (PPT)

Take the questions and rework everything accordingly.
Finalize the new version of the presentation with feedback provided.
Evaluate support requests if any comes after the request in the presentation

Initial product design *1 January 2021* (Adobe XD)

Rework the design to get the best use out of the project's layers.
Present the initial product design to the NGOs (at least 10)
Render their feedback into the final product
Manage the professor's expectations but prioritize NGO's feedback

Initial product implementation *1 February 2021* (Github Repo)

Submit the document on time and exclude some components from this submission if it is going to be an obstacle against reaching the deadline.
Finish the excluded component before the next deadline.

Regrade request if there is no rubric provided.

Final product design *1 March 2021* (Adobe XD)

Include everything with as much detail as possible to have a easier time with the final product
Take the feedback and rework the initial submission.
Make a critical analysis of the reworked design with the professor.

Final product implementation *1 April 2021* (Web Application)

Invite people to join the beta version
Share the project with NGOs directly to seek their help for development.
Invite EU and UN, UNDP, UNESCO, UN Water, UN Energy for partnerships.
Invite local angel investors to take part in the project

Final product evolution & maintenance *1 May 2021* (Github)

Do not leave the project until someone shuts it down.
Increase partnerships.
Increase operating countries.
Expand Good Deed to support international projects.
Strengthen international relations among the populations by uniting under something greater than politics.

1.6 Evolution of the Plan

Everything is bound to change. Everything is bound to fail with time. Therefore change is valued and encouraged at all phases of this project. Good Deed cannot exist without change because its core idea is to offer change to the resource allocation of NGOs. Every update should be pushed whenever they are established to be better than the previous version. Schedule, especially in developing countries is a concept to aspire. With rapid devaluation of the currency, plans get obsolete overnight. Therefore it is vital that whatever plan is made, it needs to be agile and respond to change. Consequently, change in plan can be suggested by anyone. If the suggestion is accepted, then it will be in production as well. Initial plan should be writing your name on A4's top left corner. As we currently do not have contacts with NGOs and we do not know whether this will be something they want to be a part of, further planning without the “customer” is not sensible. The product has to be developed with the customer. Hence, these ideas will be presented to a group of NGO representatives to flourish partnerships and get their valuable insights on how to make this website run better in relation to their needs.

2 REFERENCES

A14, Project_Proposal.pdf, SRS-000, Version 1.0, 15 September 2020
A14, SRS_A14.pdf, SRS-001, Version 1.1, 22 October 2020

3 DEFINITIONS

Git repository: A git repository is a directory on Github, which team members can access and collaboratively develop software by coding.

4 PROJECT ORGANIZATION

4.1 External Interfaces

While we expect to establish partnerships with the NGOs, we are not responsible for maintaining their systems. As stated in the scope, people would be only responsible for maintaining the Good Deed project when they are developed through the website. Under no circumstances, Good Deed will be the only option for donation. However, NGOs are in charge of their own process's security and integrity. There might be various versions of this project with various languages across the world. However that is a long term plan and not an external interface from today's perspective.

4.2 Internal Structure

Supporting processes may include directing people to the NGO's website to complete a donation. This would mean, as we are the people who are guiding them there, us being responsible for the transaction. Consequently, these kinds of partnerships should be created only if a trust with that establishment has been founded by the government or another notable international organization. Quality of the product is assessed through various means, however, quality of both the projects and the website is expected to be better over time. Within Good Deed, teams have to report to the core team, and the core team will be used as a joker card which can be used to work in a specific group if things are not happening according to the plan and we are falling behind. Furthermore, the core team has to be managing the issues of the developing groups and rebuilding the groups as the time goes by.

4.3 Roles and Responsibilities

Organizational Units:

- Project unit: Refers to all the people involved in the project. Consists of members of Team A14 and any other staff and developers involved in the project.
- Management unit: Consists exclusively of members of Team A14
- Design unit: Product designer and

- Development unit: Consists of the developers involved in the project, and Team A14, led by the lead software developer
- Audit unit: Consists of the lead software engineer and an independent team member who acts as the auditor.
- Quality unit (see section 7.4): quality assurer chosen from team A14

Work activity vs. Organizational Unit Matrix

Major Work Activity (see section 5.2.1**)**	**Organizational Unit Responsible**
Project Management	Management Unit
Requirements Gathering	Management Unit
Design and Analysis	Design Unit
Website Implementation	Development Unit
Testing and Production	Development Unit & Quality Unit

Supporting Process vs. Organizational Unit Matrix

Supporting Process	**Organizational Unit Responsible**
Configuration Management	Management Unit
Qualification	Management Unit & Quality Unit
Quality Assurance	Quality Unit
Reviewing and Auditing	Audit Unit
Problem Resolution	Management Unit, Development Unit, Audit Unit & Quality Unit
Environment Management	Management Unit
Process Improvement	Management Unit

5 MANAGEMENT PROCESSES

5.1 Start-Up Plan

5.1.1 Estimation Plan

See SRS.A14 1.0 Section 13.4 for schedule.

Project development will follow an iterative process involving work windows, likely one to two week long windows in which small deliverables will be completed. Daily morning meetings will cover the work completed the previous day and the work to be completed on the current day. Tools will include relevant development platforms for all team members, as well as a collaboration platform to streamline communication. Costs of the project will include any cost for the necessary tools and compensation for employees. Risks would include technological issues that could delay iterations and therefore drive costs up.

5.1.2 Staffing Plan

A small development team is all that's required to build this project. The team collaboratively works on and decides directional changes of the project. All members of this small team will need skills in either frontend or backend development as well as a good understanding of the goal of the project. The dev team will be needed full time for the duration of the schedule outlined in SRS.A14 Section 13.4. After launch of the project, additional staff may be required to maintain the website/servers/database. Relevant staffing graphics will be provided at a later date.

5.1.3 Resource Acquisition Plan

Resource acquisition should be fairly quick and simple for this project. The majority of the development team are already onboarded and have sufficient training/experience. Any additional member, if necessary, would need to go through a basic application process to ensure proper qualification. All dev team members have necessary hardware in the form of a laptop. Any software resources will likely be easily installed onto the laptops of each team member. The only potential risk to resource acquisition would be the collaborative platform. If free services such as Zoom or Skype prove insufficient, a business platform such as Webex will need to be purchased, driving up costs and potentially delaying the schedule.

5.1.4 Training Plan

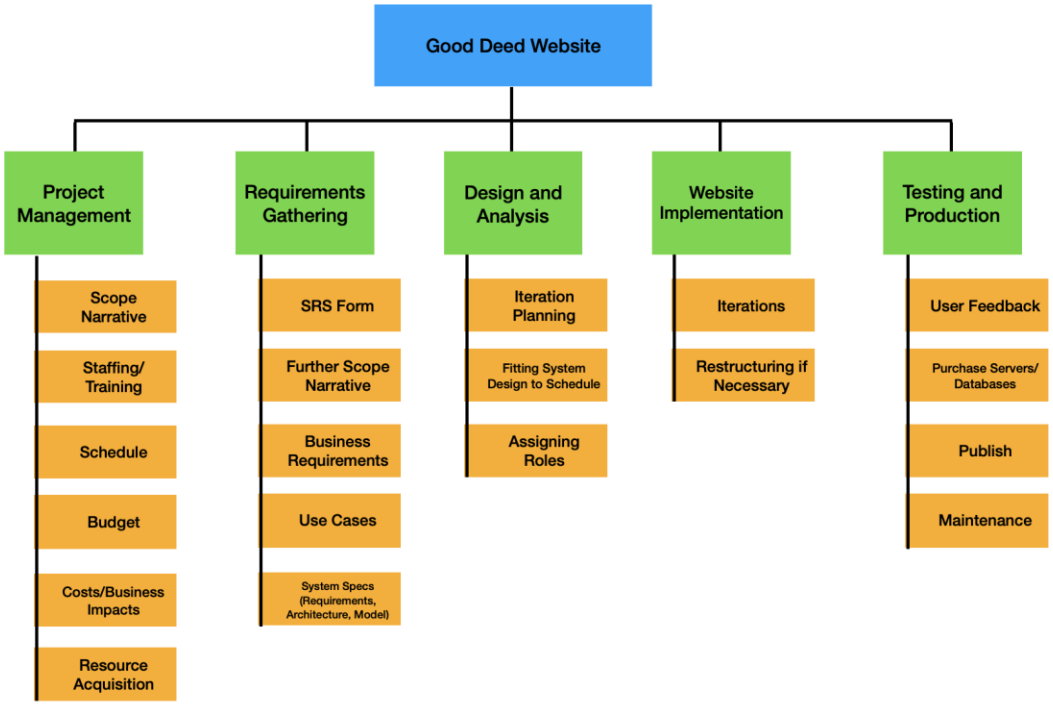
For existing team members, roles will be assigned to strengths; those with backend experience will focus on backend development and the same applies to frontend development. In the case that

additional team members are hired, a basic computer science skill-level will be assumed. Post-hiring training will include familiarization with the goal of the project and platforms that will be used during development. This training is not expected to take excess amounts of time as employees will be expected to have basic knowledge of development platforms and will only need to adapt their knowledge to the specifics of the project. In terms of managerial training, the team works collaboratively and new members will be hired only if they can effectively work in the environment of the team.

5.2 Work Plan

5.2.1 Work Activities

The WBS below gives a general description of the work necessary to complete the project. For a detailed outline of tasks and responsibilities, reference the Gantt chart in section 12.3.RA



5.2.2 Schedule Allocation

All work will be done in two week intervals. Each stage of the WBS in section 5.2.1 will be completed in two weeks barring the Website Implementation phase. In the Website Implementation phase, each iteration will take two weeks; i.e. each piece of software will take two weeks to implement. A

more detailed schedule breakdown can be found in the SRS.A14 1.0 section 13.4. A further visualization of the schedule is shown in section 12.3 of this sheet.

5.2.3 Resource Allocation

Resource allocation will remain very constant throughout the project. For all stages the team will work collaboratively with similar technology to build out the project. The team will decide on a project scope, schedule, and budget during the Project Management phase. The team will all be given the same technology as they work together to develop requirements during the Requirements Gathering phase. The team members will be assigned roles for when implementation starts during the Design and Analysis phase. The team will work via a communications platform with similar development tools during the Implementation phase. Finally, the team will collaboratively work to process user feedback, launch, and maintain the website in the Testing and Production phase.

5.2.4 Budget Allocation

No detailed budget allocation can be given at this time.

5.3 Control Plan

5.3.1 Requirement Control and Traceability

The requirements for this project will be documented in the SRS.A14. The iterative nature of this project's implementation will allow for easier adjustments to the SRS and requirements if necessary at any point. Daily morning meetings with a small team of developers all very involved and aware of the deliverables will give the team the option to suggest changes often and implement it quickly. If at any point during the project requirements need to be modified, the remainder of the two week iteration will be used to plan for said change. The next iteration will then begin the implementation of said change. While this will slow down the project overall, it wastes minimal time in comparison to starting the project all over again.

5.3.2 Schedule Tacking and Adjustment

Major milestones will come at the end of each phase shown in the WBS shown in section 5.2.1; i.e major milestones will come after each successful two week iteration. Minor milestones occur upon completion of a subtask within an iteration, demonstrated by the yellow boxes of the WBS in section 5.2.1. If an iteration is not completed by the end of its two week window, an extra week will be granted to finish the iteration and plan for any necessary schedule changes due to said extension. If an iteration cannot be completed due to changes needing to be made in the requirements or design, the remainder of the two week iteration will be used to rework the project. If this replanning is not complete by the end of the iteration, a one week extension will be granted. At the end of this period of replanning the reworked schedule will begin.

5.3.3 Budget Tracking and Adjustment

No detailed budget tracking and adjustment information can be given at this time.

5.3.4 Quality Control

Review of the previous day's code will occur daily in the morning meetings. This is intended to catch issues quickly as they occur so they can be remedied immediately. This should prevent larger problems in the code, but if deep-rooted problems do occur, refactoring will be required.

5.3.5 Reporting Mechanisms

Reporting mechanisms have not been agreed upon at this time.

5.3.6 Metrics Collection Plan

Microsoft Project will be the likely platform used to keep track of progress in relation to the initial schedule. This platform allows for easy time and team member allocations to tasks.

5.4 Risk Management Plan

Any risks assumed via technological errors that can delay project progress will be handled under the guidelines specified in section 5.3.2. These scheduling adjustment guidelines would also broadly apply to any risks that might cause the delaying of any iteration. Other risk factors and their management cannot be identified at this stage in the project.

5.5 Post Implementation Plan

No detailed post implementation plan can be given at this time.

6 TECHNICAL PROCESSES

6.1 Process Model

The website will be developed using the iterative model and the object oriented methodology. Each iteration of the website will be developed and deployed in two week intervals. Code reviews will be conducted every day during the morning team meeting to decrease the likelihood of an issue delaying production. With daily reviews, the dev team stays up to date on all the tasks being completed and is also able to perform peer code review. Feedback from the client will also be collected at each iteration and implemented in the following one.

The initial milestone consists of deploying the skeleton of the website with its basic functionalities so that it is ready to be used. Later milestones consist of additional features to the website. All functionalities of the website will be simple and user friendly with a shallow learning curve, to ensure that the client and any other user that is interacting with the website find it to be straightforward.

The project deliverables, referred to in section 1.4, consist of the following:

Completion of SRS
Project Presentation
Initial product design
Initial product implementation
Final product design
Final product implementation
Final product evolution & maintenance

For the duration of the project, decisions that are impactful to the development of the website will require the approval of all the dev team members.

6.2 Methods, Tools, and Techniques

The work will be carried out with the Agile Methodology. Additionally to this, guidelines used by Boston Consulting Group will be used in order to maximize the benefits of any given hour spent for this project. If we are going to be getting external support in development, software development teams are must. Since, they will be reporting to the core group, rather than making a production code, their code will be more error prone but that is no issue with the testing mechanism planned. Models will be developed as the customer suggests. Internally, however, the Unified Modeling Language (UML) will be used to develop valuable models for the development of the project.

It is absolutely essential to highlight that both versions of the code have to be functional, yet errors in disaster response mode are intolerable. With utmost importance, this version has to be maintained periodically. World is more global than the 1960s. Thus, emergency response systems should not be limited to 911 or TV. Mobilization of the people for a faster response and allocation of resources is vital. Amber alerts should not be limited to a buzzing on my phone only. People should be working collectively to settle this issue. Disaster Response mode will not be used with this case, but it still proves to show how important the performance of the code can be. NGOs are always associated with the worst disasters around the world. Good Deed is responsible for accommodating their needs.

6.3 Infrastructure Plan

Each team member owns the necessary hardware, and any required software will be purchased and distributed to each member’s devices.

Microsoft Project will be used as the scheduling tool. Slack will be used for instant communication and Zoom will be used to carry out both the morning meetings and any other internal / external meetings.

Credentials for tools will be stored securely by the team leader.

The website will be developed in independent environments, with frequent commits to GitHub, where the most recent version of the code can be found. Commits can be made by any team member but will have to be approved by the remaining team members before it is merged with the main branch. Tests will be conducted frequently by designated team members, and other members will be warned to refrain from committing to the repository or cloning it while the tests are ongoing.

6.4 Product Acceptance and Migration Plan

Will be provided in a later release

7 SUPPORTING PROCESSES PLANS

7.1 Configuration Management Plan

The configuration management plan for the software project outlines the plan to be followed for any configuration made on the software project implementation or the software product. The configuration management plan is implemented in instances where it is necessary to devise and modify the software requirements specification and the software project management plan.

All configurations will be identified through the person demanding the configuration notifying the project team. A configuration is defined as any diversion from the predetermined project plan, which includes the project proposal, the SRS and the SPMP. Our team plans on using a project management tool such as Trello or Slack(though this is yet to be determined). Therefore the person demanding the configuration will be responsible for posting details of the configuration, as well as the necessity for it on a “channel” dedicated for configurations in the project management tool used, such that the configuration is clearly labeled and visible to the entire team. If a client or instructor demands the configuration, the team member responsible for external communication will post the configuration on the project management tool.

All code and changes to code for the software will be managed and available over github. This is to ensure that all developers have access to the code even when the infrastructure that the software is implemented on change. Changes to code will be logged through commits on github, that is, any change to code will be documented alongside an explanation and reason for the change. The lead

software engineer of the project will oversee changes to the code, and determine whether changes made to code should be implemented.

Github will also be used as a tool for problem tracking. Bugs reported by users, developers, and clients will be logged into github as issues pending to be solved, by the lead software engineer. The lead software engineer is also responsible for the allocation of problems requiring solving to the developers.

Changes in software project management, such as changes in tools being used to develop the project, whether it be project management tools or productivity and design tools, will require the project team to get familiar with the new tools being used. Such will be done through a team wide training of the new utility being used, followed by a brief meeting on how to shift to using these new tools and how to get resources if help is needed. Furthermore, if such configuration in project development occurs, the team will also hold a meeting to transition existing documents etc. to the new tools being used by the team, such that all necessary files, project plans, design documents can be kept in a unified manner. The project team, and the authors of this SPMP will gather in case of such changes to modify the SPMP accordingly.

Configuration of the specifications of software prior to the release of the product will require configuration of the SRS, and a review of the schedule as necessary. The project team will have to update the SRS to facilitate the changes in the end product, and reevaluate the development plan in accordance with the development process.

Version management will be implemented following the release of the first version of the software product. Software components, and the versions of these components used in each version of the product will be logged and tracked. Prior to the development of each new version of the product all components that require updating for the new version will be outlined, and configured accordingly. This will also play a crucial role in system integration, such that once components have been configured to facilitate the integration of the components for the next version of the product.

When new baselines are established, or current baselines are modified, the team will modify the SRS accordingly, and come up with use cases, additional context and UML diagrams, and specifications introduced to aid the developers in the development of the product to meet the baseline requirements. The changes made will be logged and the developers will be modified through the communication tools used internally by the project management team. Then, the team will arrange a meeting with the developers to discuss the changes to the existing baselines or the new baselines, to clarify any questions the developers may have on the topic and to come up with tasks, milestones, and schedules for the developers to meet said baselines.

Automated configuration management and change control tools will be used for support in the configuration management process.

The entire team should be updated on any configuration made through a team-wide briefing, which will be made in the form of a broadcasted message to the team through the team collaboration tool being used. The client will be notified of the configuration following the team briefing, and the project team will communicate with the client in order to meet client expectations and demands with the adjustments being made.

7.2 Qualification (Verification and Validation) Plan

- Scope: In order to stay in the scope every member of the team should be told about the scope. It must be easier to maintain scope since it is the most general definition of the project. New members must also be introduced to the project through these documents.
- Tools: Prior to the submission of a functional tool into the system, it should be carefully tested. Under no condition a donation should be failing to be submitted. The blood flows through the veins of this project as donations and without blood anything built will not be sustainable.
- Techniques: Quality of the techniques used to approach an individual project of the website will be developed by the users. Under no circumstances people who are building this website are expected to find solutions for a better reactor in a NGO owned cloth machine's colouring section. People who are working for Good Deed will only administer these project's development rather than develop it on their own. People who are part of a Good Deed project are responsible for finding better techniques to develop the projects. Assessment of their performance is not a field for us to be accounted for.
- Responsibilities: Responsibilities are determined during the hiring process, a contract should protect Good Deed from bad intended employees. Termination of the job should be a ready option under these cases if a peaceful resolution cannot be achieved.
- Organizational Relations: Through the organization, every Good Deed project will have a responsible person. This person will aim to maximize the benefit of the accumulated support for their project. This person can be a representative of a NGO organization if the project is initiated by them. At the same time, this person can also be a trusted member of the Good Deed community. Ways of determining trust will be through CAMELS rating system.
- Degrees of Independence between sectors: Every good deed project is on its own. They do not have to be under the umbrella of an NGO. A project will not be allowed to gather donations unless a representative is assigned to the project. This will enable protection against false posts to scam people.
- Traceability: Traceability has key importance in the project's development. If there is an error, then the person responsible for pushing that code to production should be accountable. Without accountability there is chaos. Possibly, in order to maintain traceability and honesty in good deed projects, a section of "Shameful People" could be included to pinpoint people who used this system with bad intentions. Turkey specifically has a population who values dignity and carrying a trait like this might be more frightening than being sentenced to jail.

- Milestone review: Reaching milestones on time is critical but not vital. The most vital part of this project is the end product. While no errors can be tolerated with the final product as it will be representing an idea rather than implementation of the idea, beta version will be created only to be crashed. That is a key difference and an important aspect of the project development process.
- Walkthroughs: Walkthroughs should be available both for people who register for the platform and people who are working for the software project. In order to avoid any misallocation of the funds, there should be protocols to protect irrational behaviour. Both parties should be aware of these protocols to ensure a swift transmission. Walkthroughs for the software project staff should cover all information mentioned in the SRS and this document to ensure that the staff is up to date with the project information.
- Peer Reviews: Everything has to be reviewed by everyone. If there is a suggestion made by the users, then we should be as welcoming as possible for that effort. At the same time, reviews should be concise and not take too much time unless otherwise is wanted. Furthermore, the coding process should also be hosting peer reviews. These reviews can be passed to the management only, or openly discussed before the start of the next sprint by the team, as the Boston Consulting Group suggests.
- Prototyping: The beta run of the website will be made in a highschool with students testing the website and actively trying to crash it, without malicious intent but to foster the development of the project further.
- Simulation: Some people were right, the real world is the simulation. As previously stated, the main part of this project is to find a way, fail and find a better way. Innovation comes from iteration of the product in the market with a better quality in certain aspects that is good enough to make our product be sold. Innovation is required in the NGO's because expert opinions are associated with an expense, research and development is a secondary area for NGOs. Since they use their power to alleviate the conditions in critical need. Often, they are not capitalized enough to expand their operations or focus on efficiency in allocation of their workforce. By helping them to come up with better solutions without any additional costs, and even possibly assessing their financials openly, NGOs must be able to perform better than before.
- Modeling: Modeling is used to give an initial image of the process and how this project would be running. We will not need modeling after an introduction to the project is made for the person. Models can be altered as the project pivots or as a better model is proposed by any member.

7.3 Documentation (library) Plan

Organizational entities responsible for providing information, generating and reviewing documents will be specified in this documentation plan.

Software is developed to answer the needs of NGOs and make them function more efficiently. Hence, the customer might seem to be the donators only at the first sight, but if the end product is

not delivering the initial requirement, then partnerships will not be established as planned or swiftly which will hamper the process towards success. Interface should be fostering the feelings of the donators. That means the colour schema should be targeting empathy and trust. A combination of blue and red, or orange in order to represent innovation.

Documents related to a specific project will be reviewed by the project representative who is only appointed by the Good Deed project and is independent in his/her operations, both in securing where the donations are heading and how the people are offering possible solutions. Always peaceful, empathetic and goal oriented. Google docs will be used for documentation.

Traceability is required at all phases in order to ensure security and avoid fatal errors for the project. Transparency is the key factor in development of any component of this project. If there is more information, then the solutions would be tailored accordingly to the needs of the consumer. Therefore, server costs associated with this component, if nobody offers covering the costs of the server, should be carefully managed. Furthermore, the software should cease its operations running with the regular UI if a natural disaster occurs. This software component will simplify the existing UI in order to function faster with a damaged cellular network.

Source code will have two functional versions, one minimal and the other the regular day version. Both of these versions should run without errors preferably but errors in the minimal version are absolutely intolerable. Since, the regular day version is constantly worked on, it can have a mistake on the production code.

Business layer can run anywhere remotely but the data layer, especially as it contains valuable information, should be carefully protected. Card information, donation spreadsheets and personal information or a private post in a project group should be kept integral, and private from the people who did not share this information.

Online assistance should be done as the hierarchy suggests. First connection is the project representative, then the sector representative then the top. If the issue regarding the software cannot be resolved through this path, then possibly walkthroughs or manuals have to be recrafted to guide people better on how to use the end-product of this process.

Software should not include complex libraries. It should focus on using components from open-source projects. Therefore, documentation should be matching the requirements proposed by the NGOs and besides that as long as the system is efficiently functioning, any pivot is possible.

One should be reminded that this whole website is based on testing. The aim is to increase efficiency of solutions developed by NGOs through testing repeatedly and noting the changes. Therefore, documentation should be profound enough if it is going to be presented to the NGOs who are potentially less informed about software. All of these and checking the accuracy of these will be done by Kerem Ulcay at all phases of development.

7.4 Quality Assurance Plan

In order to ensure that the software project fulfills its commitment to the software process and the software product as specified in the requirements specification, the software project plan, and supporting plans, the project team will delegate a member to be responsible for quality assurance. The member responsible for quality assurance (referred from hereon as “quality assurer”) will be within the core project team, and will attend every meeting held by the team.

The quality assurer will also organize quality assurance meetings to be held with the client or a supervisor of the project, as well as with internal team members. In these meetings, the software project management and the current stage of the software product will be reviewed. The aim of these meetings will be to go over the SRS and the SPMP to ensure that the plans, requirements, and proposed organizational structure are being implemented as foreseen. The current implementation of the software project alongside the software product will be analyzed and inspected in accordance with the proposed SRS and SPMP. At the end of each quality assurance meeting, a software project implementation quality review and assessment will be created by the quality assurer, and the review will be shared with the entire project team.

If the review indicates that the software project does not fulfill its commitment to the software process and the software product as specified in the requirements specification, the software project plan, and supporting plans, then additional reviews and audits will be held to diagnose the root of the problem and the reason behind the below par quality the software project has demonstrated, and necessary steps will be taken according to the problem resolution plan to ensure that the desired quality is being achieved.

7.5 Reviews and Audits

General audits and reviews will be conducted on a monthly basis. General audits will be concerned with scheduling, resources, and meeting client demand, as these are the three most critical concerns for a software project. These general audits will ensure that all team members and project components are on schedule, that these processes do not exceed the budget allocated, and that they are executed in accordance with the client demand and the SRS. The results from this general audit, alongside results from all other sub-audits mentioned below will be shared and reviewed by a pre-selected audit team on a monthly basis in a meeting open to all team members.

A project management audit will be conducted throughout the development of the software project. This audit will be done through organizational management reviews, developer peer reviews, technical reviews, walkthroughs, inspections. Organizational management reviews will be conducted through one on one interviews with team members responsible for project management, software development and quality assurance. These interviews will be scheduled on a monthly basis. Developer peer reviews will be conducted through github, as each developer adds a new code file to the repository. Other developers will be responsible for peer reviewing the added code

and providing feedback. Walkthroughs will take place frequently, and in an unscheduled manner. During these auditing walkthrough sessions, one team member will be assigned with simply talking to another team member about what they are working on, and how the process is proceeding. If the team member conducting the walkthrough audit recognizes any discrepancy or practice that deviates from the standard set forth in the SRS and this document, such will be added to the review to be shared with the audit team's aforementioned monthly meeting.

7.6 Problem Resolution Plans

As software problems occur during the project, these problems will be reported to on a channel dedicated for software problems on the project management tool used by the project team and staff. The software problems will be ranked in terms of urgency on three levels: critical, intermediate, and low urgency. Critical problems are those that prevent the developer or other developers from completing the development tasks and require immediate resolution. Therefore, each morning, the channel for software problems will be reviewed, and if there are any critical software problems, the entire development team and the project management team will convene for an online meeting to resolve these issues. If the software issue that arises is a bug that can be solved through programming, the development team will collaborate throughout the day to solve the problem using debugger tools if necessary. If the problem cannot be resolved within the duration of the day, the project team will reconvene with the lead software engineer the next morning to discuss alternative routes to circumvent the software problem, and develop a new strategy that will be discussed in the next weekly briefing session organized with the project team.

Intermediate and low urgency software problems will be discussed in weekly developer briefings, and a team of developers will be assigned to solve the software problem. The management team will be briefed on the software problem that has occurred, and will discuss strategies to mitigate any additional delays and resource spending.

If the software problem requires a configuration to the software project, the configuration management plan will be implemented to accommodate said configuration. Following the implementation of the configuration management plan, the quality assurance plan will be implemented to ensure that the configuration made to the software project is within the standards set forth for the software product in the SRS and the SPMP. Once the configuration has been implemented for the arisen software problem, an audit of the configuration will be implemented as a part of the general audit mentioned in section 7.6 to ensure that the project remains within schedule, budget, and client demands. Change control will be implemented throughout the process of problem resolution as specified by the configuration management plan.

Any additional work on problem reporting, analysis, and resolution will be individually reported in the work breakdown system so that work can be tracked and process improvement accomplished.

7.7 Environment Management Plans

The developing environment used by developers will be visual studio code. Github will be used to ensure management access to the development process. Further environment management plans will be discussed and developed, as the project team proceeds with the programming for the project and explores options for the software development environment.

7.8 Process Improvement Plan

The process improvement plan has not yet been discussed and created by the project team. As the project proceeds and the team gathers more information on the process and gathers insight on how the process might be improved, an elaborate process improvement plan will be created.

8 ADDITIONAL PLANS

None for the time being.

9 INDEX

None

10 RATIONALE

None

11 NOTES

None

12 APPENDICES

12.1 Schedule Tracking

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference

SRS – Business Domain	Jonah Sherman	2 hours	1 hour	1 hour
SRS – Business Domain	Sofia Machado Lopes	2 hours	½ hour	1.5 hour
SRS – Business Domain	Timur Blair Gordon	1.5 hours	½ hour	1 hour
SRS – Business Domain	Kerem Ulcay	2 hours	1.5 hours	½ hour
	Summary for entire team	8 hours	3.5 hours	4 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Requirements	Jonah Sherman	1.5 hours	1 hour	½ hour
SRS – Requirements	Sofia Machado Lopes	1.5 hours	1.5 hours	0 hour
SRS – Requirements	Timur Blair Gordon	1.5 hours	1.5 hours	0 hour
SRS – Requirements	Kerem Ulcay	1.5 hours	1 hour	½ hour
	Summary for entire team	6 hours	5 hours	1 hour

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Analysis - Complete	Jonah Sherman			
	Sofia Machado Lopes			
	Timur Blair Gordon			
	Kerem Ulcay			

	Summary for entire team			

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Jonah Sherman	2 hours	2 hour	0 hour
SPMP	Sofia Machado Lopes	2 hours	2 hours	0 hour
SPMP	Timur Blair Gordon	2 hours	1.5 hours	½ hour
SPMP	Kerem Ulcay	2 hours	1.5 hour	½ hour
	Summary for entire team	8 hours	7 hours	1 hour

Cumulative

Who (individual and Team)	Estimated	Actual	Difference
Jonah Sherman	5.5 hours	4 hours	1.5 hours
Sofia Machado Lopes	5.5 hours	4 hours	1.5 hours
Timur Blair Gordon	5 hours	3.5 hours	1.5 hours
Kerem Ulcay	5.5 hours	4 hours	1.5 hours
Summary for entire team	21.5 hours	15.5 hours	6 hours

12.2 Defect Tracking

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Business Domain	Jonah Sherman	1 hours	1 hour	0 hour
SRS – Business Domain	Sofia Machado Lopes	1 hours	1 hours	0 hour
SRS – Business Domain	Timur Blair Gordon	1 hours	0.5 hours	½ hour
SRS – Business Domain	Kerem Ulcay	1 hours	0.5 hour	½ hour
	Summary for entire team	4 hours	3 hours	1 hours

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS – Requirements	Jonah Sherman	1.5 hours	1.5 hours	0 hour
SRS – Requirements	Sofia Machado Lopes	1.5 hours	1 hour	½ hour
SRS – Requirements	Timur Blair Gordon	1 hours	1 hour	0 hour
SRS – Requirements	Kerem Ulcay	1 hour	1 hour	0 hour
	Summary for entire team	5 hours	4.5 hours	½ hour

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference

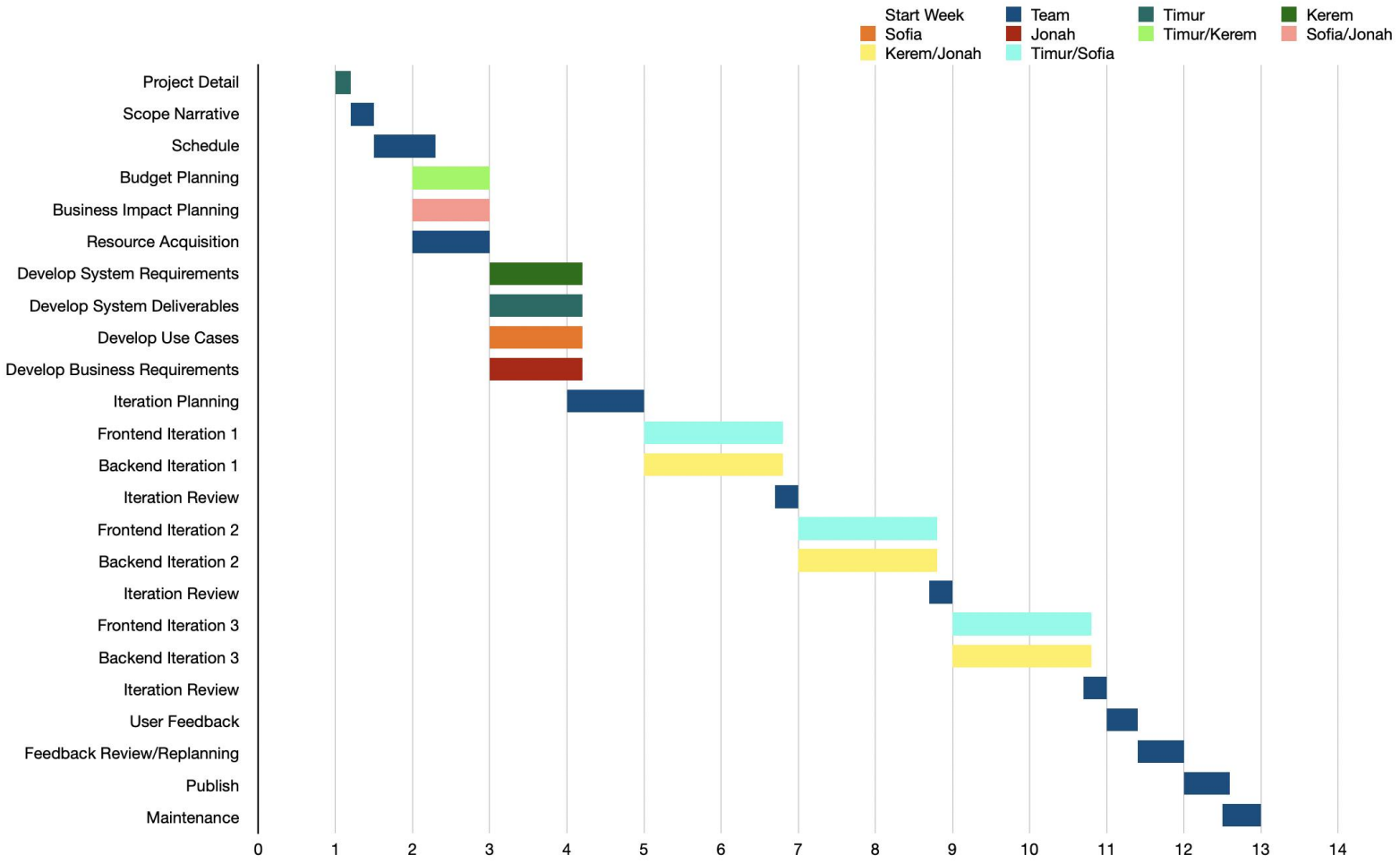
SRS – Analysis - Complete	Jonah Sherman			
SRS – Analysis - Complete	Sofia Machado Lopes			
SRS – Analysis - Complete	Timur Blair Gordon			
SRS – Analysis - Complete	Kerem Ulcay			
	Summary for entire team			

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Jonah Sherman	2 hours	1.5 hour	½ hour
SPMP	Sofia Machado Lopes	2 hours	2 hours	0 hour
SPMP	Timur Blair Gordon	2 hours	1.5 hours	½ hour
SPMP	Kerem Ulcay	2 hours	1 hour	1 hour
	Summary for entire team	8 hours	6 hours	2 hour

Cumulative

Who (individual and Team)	Estimated	Actual	Difference
Jonah Sherman	4.5	4	½ hour
Sofia Machado Lopes	4.5	4	½ hour
Timur Blair Gordon	4	3	1 hour
Kerem Ulcay	4	2.5	1.5 hours

Summary for entire team	17 hours	13.5 hours	3.5 hours



12.3 Gantt Chart/Microsoft Project/Spreadsheet Schedule

Note that this schedule fits within the 14 week NYU semester length.