Computer Science and Engineering

Social Network Analysis with Deep Learning

Project Management Plan

Version 1.0

Document Number: SPMP-001

Project Team Number: A9

Project Team Members: Anish Malhorta (am7161)

Chun Zen Marc Tse (czt210)

Mehmed Mladenov (mm7277)

REVIEW AND APPROVALS

Printed Name and Title	Function (Author, Reviewer, Approval)	Date	Signature
Anish Malhorta	Author	Spring 2018	
Chun Zen Marc Tse	Author	Spring 2018	
Mehmed Mladenov	Author	Spring 2018	

REVISION LEVEL

Date	Revision Number	Purpose
Spring 2018	Version 1.0	Initial Release

Table of Contents

1. OVERVIEW

The project is organized to produce an application for corporations and businesses to utilize data analysis on datasets available on social media and deep learning to analyze and predict trends so that companies can make more informed business decisions and compete in a dynamic market. The early stage of development involves heavy design for most functions of the application, as well as early part of implementation for some functions. The middle stage of development involves heavy implementation and integration of different functions into one application, and the final stage of development involves optimization of application, fixing defects and assuring the quality of application. Relevant documentations such as SPMP (this document), RAS, SDD, will be updated throughout the project cycle to provide an accurate depiction of the progress of project and software deliverables.

1.1. Project Summary

Motivation for the project

The idea for the application emerges as we see the potential of big data analysis and deep learning. With the current availability of large datasets online, myriad of information can be processed to make strategic decisions for businesses and industries. We look to utilize those untapped information to optimize and enhance the effectiveness of businesses, as well as providing meaningful insights to strategies employed by businesses.

Purpose of this document

This document mainly provides an overview of the structure of the management of project, the resources required to manage the project efficiently, as well as to keep track of the project schedule.

Intended Audience of the document

The intended audience for this documents is mainly but not limited to, the client of the software product, the management level personnel involved with the software team, developers of the software, and the testers and end-users of the software.

1.2. Purpose, Scope, and Objectives

Purpose

The purpose of this project is to gauge sentiments and trends on the social media, thereby helping the marketing teams of businesses in identifying possible strategies, as well as analyzing the effectiveness of marketing campaigns. And with the technology of deep learning, the application can help to adjust and find the best fit of solution for a specific client.

Scope

There is a lack of simple, efficient and modularized support for determining the efficacy of marketing campaigns by operating on different data sets and performance metrics. The system will address these concerns by allowing the plugging-in of various data sets and the standards used to determine their success, as well as predict future performance. The system will provide an easy-to-use interface for this purpose, which will show results of big data analytics in visual formats, such as GIS maps, charts and spreadsheets. The system, and the algorithm that supports it, will be fine-tuned to support modularity for a variety of data sets that may be used with it as well as the performance indicators used to measure potential success. Such a system would have tremendous benefit for many organizations, simplifying the process of using Deep Learning greatly while also reducing the cost of implementing such techniques.

Objectives

The highest priority objectives are the optimization of the algorithm necessary to perform analytics on data sets, as well as the actual representation of the analysis. As such, the visual component of the system will be crucial – making certain that the customer has several interchangeable ways to view the analytics they will use the system for. Making the system simple to use will also be a high priority deliverable, with a functional and self-descriptive dashboard being necessary. Another important, but lower priority objective will be modularization of the algorithm itself. Being able to use different data sets is an important long-term goal of the system, but not before the structure has been created for a general use case.

1.3. Assumptions and Constraints

Assumptions

Several factors upon which the project is based are assumed. Legal factor such as the privacy concern regarding data collection of consumers can impact the success of the project and the final software deliverable. The data collected from users of social network is assumed to be made aware by those users, and businesses and clients who use the software is assumed to adhere to the ethical usage of the software. Another factor is the quality of data collected from the social network. The quality can determine how well the software functions, and the data is assumed to be of maximum use to the software.

Constraints

One of the biggest constraints for the project is the technological aspect of how well the AI algorithm of analyzing data from social network and predicting trends function as well as the business implication of it. The project's success or failure rely solely on the algorithm performing well. If businesses rely on the software for determining marketing strategies, and if the software fails to predict the trend, which is completely possible, then the outcome can be detrimental to the software. The legal implication of the project can also be a major constraint. With the negative news coverage that Facebook brought recently to private use of data collection and analysis, the software deliverable of the project can also bring negative attentions and thus result in business failure. Privacy concerns of data usage can potentially annul any use of the software. The software also requires clients and businesses self-compliance to ethical usage (i.e. not using data for sabotage).

1.4. Project Deliverables

1 Project Proposal: 2/08/18
1 Software Requirements and Analysis Specification (RAS):
Business & Project Definition: 2/20/18
Software Requirements: 3/06/18
Complete: 4/5/18
Software Analysis Specification: 4/17/18
1 Software Project Management Plan (SPMP): 4/05/18
1 Software Design Document: 5/01/18
1 Presentation: 4/26/18 – 5/03/18

1.5. Schedule and Budget Summary

Schedule

All schedule is organized by 2-week sprints. Iterations of development of application/function of application usually lasts for 2-3 sprints (1-1.5 month), when all requirements for the related functions of application is expected to be completed. All project documentations should be updated at the end of each sprint in order to reflect progress of project and any changes made to requirements of application. During the first month of development, most of the first iteration of designing algorithm for data analysis and data prediction as well as the development of UI should occur. Databases related functions of the application such as integration of external social network APIs should also begin development. During the second month of development, the second iteration of algorithm design, UI and databases function should occur. The third month of development should involve the polish of UI, optimization of the application (in terms of memory usage and speed), as well as any optional features that improve user experience should be implemented (if time allows). The forth month of development should be reserved for quality assurance of product and fixing defects of deliverables.

Budget

TBD in future versions of document.

1.6. Evolution of the Plan

After every major milestone is achieved throughout the project, this document and any other relevant documentation should be updated to reflect accurate changes and progress to the project. (Version 1.x -> 1.(x+1))

After each meeting during the week, all discussions should be recorded into meeting minutes. All meeting minutes should be compiled to a single document by the end of the week and should be referenced in this document and any other relevant documentations. If minor changes (do not impact the initial requirements set for the project) are proposed and approved, this document and any other relevant documentation should be updated. (Version 1.x.y -> 1.x.(y+1)).

Each documentation should be saved to the GitHub project repository, and each version should be updated to the repository for convenient version lookup as well as backup recovery or reversion to previous versions of documentations.

2. REFERENCES

“Team A9 Project Proposal” Social Network Analysis with Deep Learning, A9, Version 2.0, February 16, 2018.

“System Requirements and Analysis Specification (RAS)” Social Network Analysis with Deep Learning, Team A9, RAS-001 Version 1.1, March 6, 2018.

3. DEFINITIONS

API – **Application Programming Interface**, layer of clearly defined communication between different systems, applications, databases.

CM – **Configuration Management**, the software engineering methods of management of artifacts generated throughout the project cycle to ensure their quality and integrity.

DKLOC – **Defects/Thousand lines of codes**, a project metric used to measure defects in source code and determine the quality of product.

IEEE – **Institute of Electrical and Electronics Engineers**, professional association related to the field of software engineering that provides the standards for the development of software.

PRC – **Percent of Requirements Completed**, a project metric used to measure the number of requirements completed over the total number of requirements and determine the progress of project.

PRS – **Percent of Requirements Satisfied**, a project metric used to measure the number of requirements satisfied over the number of requirements completed and determine the quality of project.

QA – **Quality Assurance**, the software engineering methods of ensuring and maintaining the quality of deliverables generated throughout the project cycle.

RAS – **System Requirements and Analysis Specification**, a project documentation that details

SDD – **Software Design Document**, a project documentation that details

SPMP – **Software Project Management Plan** (this document), a project documentation that details the management process of the project.

TBD – **To Be Determined**, postponed to later when a concrete decision is then made.

UI – **User Interface**, layer of communication between end-users and the application.

UML – **Unified Modeling Language**, a standardized method for producing visualization for design of systems.

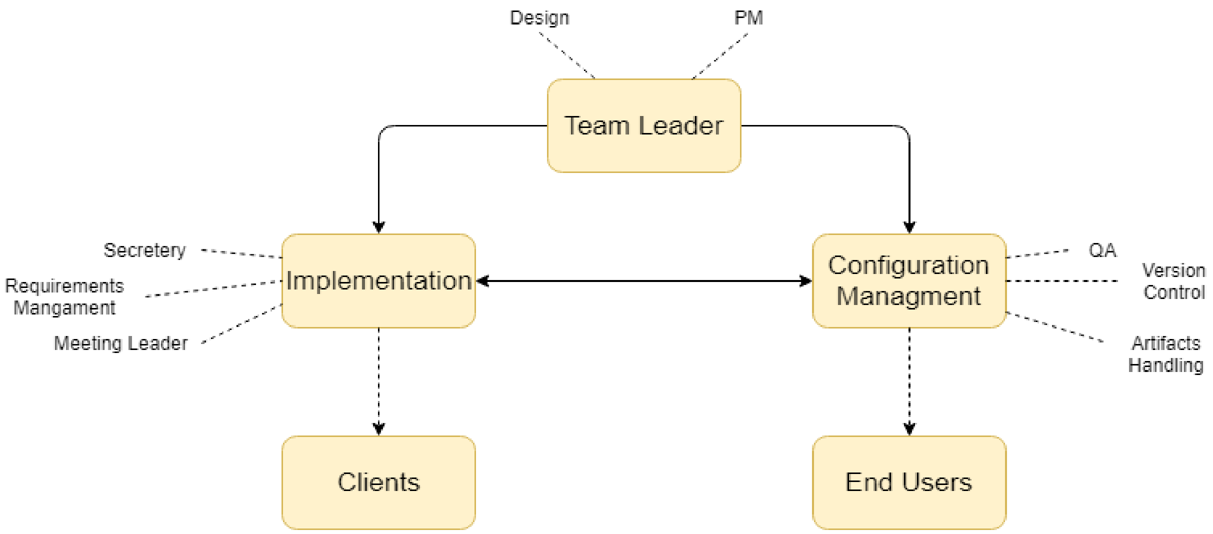
4. PROJECT ORGANIZATION

4.1. External Interfaces

The system boundaries exist between the software and the clients (businesses and corporations), the software and the end-users of the software (employees of clients, data researchers), and software and external APIs form databases of social network companies. Implementation Leader is responsible for communication with clients in acquiring and refining requirements, whereas Configuration Management Leader is responsible for communication with end-users in assuring and improving the quality of software. The Team Leader is responsible for communication with social media companies to ensure external APIs usage fit the project requirement and should change requirements if necessary.

4.2. Internal Structure

Name	Team Leader	CM Leader	QA Leader	Requirements Management	Design Leader	Implementation Leader
Marc	***X***				***X***	
Mehmed		***X***	***X***			
Anish				***X***		***X***



4.3. Roles and Responsibilities

Team Leader

Will oversee communication between other roles and manage changes in project schedule related to changing requirements and defect resolution, make sure project deliverables and milestones of project are achieved in a timely and professional fashion. Will also be responsible for the overall design of the project.

Design Leader

Handles communication with Quality Assurance Leader (QA) and Requirements Manager in order to provide a second level of assurance with regards to verification and validation. The Design Leader will interface with internal roles to determine a design for the project and communicate with those roles in order to build, deploy and present software baselines to the customers.

The Implementation Leader

Responsible for seeing through the entire process of implementing the design into a software deliverable. Will also be responsible for overseeing the daily SCRUM meetings and weekly formal meetings, making sure they are conducted in a productive and efficient manner, as well as taking minutes for the meetings.

Requirements Management

Responsible for acquiring clients wants and needs and shaping them into high-level and detailed requirements. Will maintain consistent communication with clients in order to assure design verification. Will communicate requirements with the Team Leader/Design Leader for design and schedule changes.

Configuration Leader

Responsible for the quality control and management of artifacts generated throughout the project. Primary point of communication with customers with regards to releases of new baselines and approving changes made. Will oversee quality assurance and approve changes made in defect resolution.

Quality Assurance Leader
Responsible for validating that deliverables satisfy requirements with the customers. Maintain the quality of each artifact generated throughout the project by making sure they achieve the desirable metrics set forth in the requirement document.

5. MANAGEMENT PROCESSES

5.1. Start-Up Plan

The estimation of costs and resources should be determined by function points, as well as comparing project to similar applications in the market. The core staff of the team should be the Team Leader, the CM leader, and the Implementation Leader. Because of the small size of team, all members of staff should be equally skilled in all aspects of developing and designing the application. If members of staff are deemed to be unskilled in any aspect of development and designing of application, then members should train themselves to become proficient by any mean necessary.

5.1.1. Estimation Plan

TBD in future versions of document.

5.1.2. Staffing Plan

TBD in future versions of document.

5.1.3. Resource Acquisition Plan

TBD in future versions of document.

5.1.4. Training Plan

The specified Project Leader has the requisite educational and professional managerial background to lead a small project team (management course at NYU Stern Business School, past experiences). Because the size of the team is small, all roles will additionally serve as engineers. One of the major risks in the inception of the project is that the engineers involved in the project are only proficient in some of the required technologies. All are proficient in databases, some are proficient in Artificial Intelligence and GIS Maps data visualization. Development areas will overlap because of the small size of the team. This requires training in Visual Data Representation (D3, etc.), Deep

Learning technology, utilization of Social Network APIs and Big Data technology such as data collection, data cleaning and data analysis. The proposed training methods include books, online courses and video tutorials, as well as referral to documentations of technology used in the software.

5.2. Work Plan

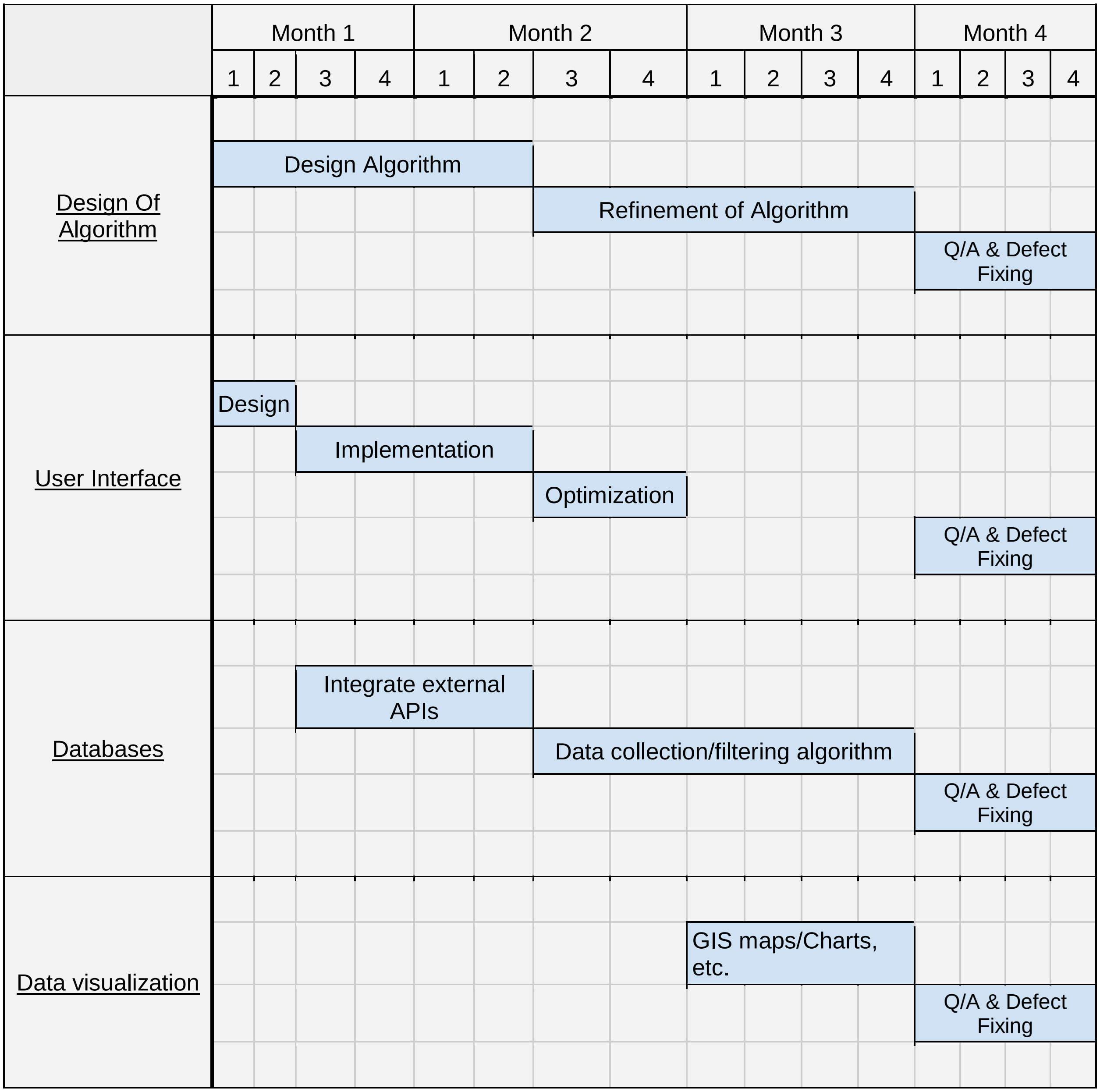
5.2.1. Work Activities

Task	**Subtask**	**Person (time allocated)**
UI		
	Design	AM (20h)
	Implementation	AM, MM (40h)
	Optimization	AM, MM (15h)
Databases		
	API Implementation	AM, MM (40h)
	Data Collection/Selection	MM, MT (45h)
Data Visualization		
	GIS Maps	MM (20h)
	Charts and Graphs	AM, MM (20h)
	Exportable Spreadsheets	MM (5h)
Algorithm		
	Initial Design (Data Cleaning, Analysis, Prediction)	AM, MM, MT (120h)
	Refinement (Speed)	MT (80h)
	Refinement (Modularity)	AM, MT (40h)
Management and Documentation		
	SPMP	AM, MM, MT(10h)
	RAS	AM, MM, MT (25h)
	SDD	AM, MM, MT (10h)
Testing		
	Defect Resolution	AM, MM, MT (40h)
	Verification and Validation	AM, MM, MT (20h)
Overall Hours		
Anish Malhotra	175h	
Chun Zen Marc Tse	185h	
Mehmed Mladenov	180h	
Cumulative	**540h**	

Work Breakdown Structure (person-weeks)																	
		Month 1				Month 2				Month 3				Month 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	AM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	MM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	MT	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Weekly Totals		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Cumulative		3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48

5.2.2. Schedule Allocation



5.2.3. Resource Allocation

Task		Skill Level Required (1 low - 5 high)
UI		
	Design	1
	Implementation	2
	Optimization	1
Database		
	API Implementation	3
	Data Collection/Selection	3
Data Visualization		
	GIS Maps	4
	Charts and Graphs	3
	Exportable Spreadsheets	2
Algorithm		
	Initial Design (Data Cleaning, Analysis, Prediction)	5
	Refinement (Speed)	5
	Refinement (Modularity)	3
Management and Documentation		
	SPMP	1
	RAS	1
	SDD	1
Testing		
	Defect Resolution	3
	Verification and Validation	3

5.2.4. Budget Allocation

TBD in future versions of document.

5.3. Control Plan

5.3.1. Requirement Control and Traceability

All requirements analysis will emphasize requirement traceability in order to prioritize requirements that specify customer wants and needs. This will minimize the impact of later changing requirements. Customer wants and needs will be communicated to the Implementation Leader/Requirements Manager, who will then follow the principles of Agile development in construction of requirements-based configuration items.

Subsequent requirement control should adhere to the principle of Agile requirements analysis, where the initial proposal of requirements are discussed among members of team, and the requirements should be measured against a traceability chart and risk factor chart in order to determine the impact of the proposed requirement. If the proposed requirement is determined to have a positive impact on the overall quality of software and will not significantly delay the project schedule, then the proposed requirement should be updated to the SPMP document and other relevant documents. Otherwise, the proposed requirement should not be implemented and the detail of the proposed requirement as well as the reasoning for requirement non-implementation should be documented.

5.3.2. Schedule Tracking and Adjustment

Project metrics should be used in order to measure progress of work completed at the major and minor project milestones. Such metrics include percent of requirements completed that track the number of requirements already completed over a timeframe versus the total number of requirements, as well as the total number of man-hours used for design, implementation and optimization of functions of software. The metrics should be evaluated against the planned schedule, and if it is determined that the total number of man-hour spent has significantly delayed the schedule (by 1 week or more), or if the percent of requirements completed is not up to par according to the planned schedule (expected 50% completion by week 2 or 1-month long schedule), then the team should discuss about points of pivots: whether to continue with the development of the function, or change the requirements of software to fit the schedule (be it a more simplified function, or completely canceling feature if deemed not impactful to the quality of software). If it is determined to continue with development, all personnel should be directed to the effort of finishing that specific feature of software.

5.3.3. Budget Tracking and Adjustment

TBD in future versions of document.

5.3.4. Quality Control

Quality control mechanisms such as inspections, prototyping, modeling and testing will be used to ensure quality. Because the components of the project are functional independently, each iteration will consist of a prototyping phase which will minimize risk and reduce the defect rate. Modelling and testing will be used for the algorithm to verify analysis and prediction. Several metrics will be used in keeping quality in the project and staying on schedule, such as the DRR, DKLOC and PRS.

5.3.5. Reporting Mechanisms

TBD in future versions of document.

5.3.6. Metrics Collection Plan

Weekly formal meetings will be conducted as an overview for the progress of the project over the week. Metrics will be collected with this same frequency – weekly recordings of the PRC, number of man-hours spent, DRR, DKLOC, and PRS. These metrics will be reported alongside the weekly compilation of daily minutes from SCRUM meetings, and will be memo'd to all members of the team. This frequency is necessary because the workload of the project is quite high for the specified schedule, so the metrics will assist in maintaining project development and testing on pace

5.4. Risk Management Plan

Risk identification will occur at the beginning of every sprint. Risks will be retired through avoidance using prototypes for each iteration of project development. If change to the project requirements is needed to mitigate a risk, then the impact of the requirements change will be evaluated and discussed by the team. Risks will be evaluated for priority in determining scheduling changes. Risks will be identified by imagining all worst-case scenarios relative to each function of the software. The risk management plan will be enforced by the Project Leader throughout the development of the project. In extreme situations where a significant functionality of the software has to be added or changed in a short time frame, the design of the functionality should take place immediately instead of performing risk analysis.

5.5. Post Implementation Plan

TBD in future versions of document.

6. TECHNICAL PROCESSES

6.1. Process Model

An Agile/SCRUM process will be used for the development of this project. 15-minute daily SCRUM meetings will occur reviewing the previous day and going over the workload before the next meeting. Weekly formal meetings will occur to track the progress of the project from a larger scale as well as performing ongoing risk identification and diligence with regards to changes of requirements and configuration items. On this day, weekly inspections will occur as well to decrease defects present throughout development. Each sprint will last for two weeks, during which there will be two formal meetings and inspections. Each sprint will result in a project deliverable with working software. The Configuration Manager will review and approve these changes at the end of every sprint. Formal risk identification will occur at the beginning of every sprint, starting with prototyping. Each iteration of project deliverable may contain from 1 sprint to 3 sprints (as per the planned schedule in 5.2.2). All requirements of major milestones should be completed within necessary number of sprints.

6.2. Methods, Tools, and Techniques

UML is used as the main method for design of the system of application. Draw.io is used as the tool for created UML formatted diagrams. For all documentations of the project, Google Doc. is used as the tool for collaboration of documentation among team members, while Microsoft Word is used as the tool for publishing the finalized documentation. All charts used in documentations are created using the Microsoft Excel tool. All schedules used in documentation are created using the Microsoft Project tool. All daily minutes and weekly meeting notes should be recorded using Microsoft Word. GitHub is used for storing all artifacts created throughout the project including source codes, documentations, minutes.

6.3. Infrastructure Plan

The IDE will be used for initial debugging and testing of source code. Because there is not testable metric to measure the efficacy and validity of the algorithm's results and analysis, historical data will be sued to determine the effectiveness of running the algorithm on different sets of data. Comparing historical data and results to new datasets found by the application will be the primary method of testing the algorithm. Testing of data visualization will occur similarly, using historical data to validate graphing and representation techniques.

6.4. Product Acceptance and Migration Plan

TBD in future versions of document.

7. SUPPORTING PROCESSES PLANS

7.1. Configuration Management Plan

The Configuration Management plan of the project contains the configuration evaluation, identification and release management. The evaluation is completed with the customers, and team evaluations are done constantly throughout the weekly meetings. The identification is also determined by the customer and their requirements. The releases are done after each milestone is achieved as per the schedule. This schedule is modified throughout the development process and updated according to the updates and change requests. These changes could cause missed deadlines, so the team is observing them frequently during the development of the product.

7.2. Qualification (Verification and Validation) Plan

Verification and validation will be done with unit and acceptance tests. Verification is done between team members through daily SCRUM meetings, followed by formal weekly meetings. Prototyping of application and modeling of algorithm used in application will be heavily used for the verification process of development. At the beginning of each iteration of development of application, a prototype of the core priority of development should be created in order to verify the feasibility of the application. The prototyping process should not take longer than twelve hours, and in extreme cases where development is lagging behind schedule, prototyping process should not be used. Instead, a less labor intensive method such as traceability of metrics is applied. Our customers will help in the validation process with validating if requirements and tasks are being satisfied. This can be done in weekly formal meetings, which will include demonstrations of baseline iteration of core function of product and project status.

7.3. Documentation (library) Plan

Internal deliverables such as meeting reports, test plans, risk factor charts, metrics report and review report should be documented during the weekly formal meetings (some only applicable after the release of initial baseline versions of software deliverables). Meeting minutes should be generated during each daily SCRUM meetings, and minutes should be compiled into the weekly meeting reports. The Implementation Leader should generate all meeting reports and minutes, while all other reports should be generated by their respective leaders related to the types of the reports. External deliverables such as RAS, SPMP, SDD, the application (source codes, executables and manuals) should be generated by all members of team. The Configuration Leader should oversee the review and approval of the deliverables. The initial baseline version of deliverables should be in accordance to the expected achievement of milestones as per the schedule.

7.4. Quality Assurance Plan

As stated in previous sections of the documents, acceptance and validation tests such as weekly review and inspection of baseline iterations of application as well as demonstration of project progress and software deliverables to both internal members of team and clients is used to assure the quality of application. All quality assurance test should ensure that the application adhere to the IEEE quality standards (IEEE 730-2014).

7.5. Reviews and Audits

After a baseline version of the development of a core function is published, a thorough review of the version is reviewed in each weekly formal meeting. Developers are also encouraged to spend time reviewing the application on their own, and bring up any issues during the daily SCRUM meetings. The audit of the project is done by the CS Department of NYU Tandon University of Engineering as per its guidelines on quality and specification for the project. The professor of the Software Engineering and Senior Design course should oversee the final approval of the project.

7.6. Problem Resolution Plans

Problem reporting should occur in the daily SCRUM meetings, as well as the weekly formal meetings, where developers should express difficulties encountered during development as well as found defects. All effort should be spent towards resolutions of problems and defects at the end of each week. The last month of development is also reserved entirely for fixing defects. The issues raised during these meetings are organized by their importance and priority level in terms of the impacts of the issues relative to the product with the risk factor chart and schedule in mind. The team scheduler’s tool should be used to keep track of project problems and updates.

7.7. Environment Management Plans

TBD in future versions of document.

7.8. Process Improvement Plan

TBD in future versions of document.

8. ADDITIONAL PLANS

TBD in future versions of document.

9. INDEX

None

10.RATIONALE

None

11. NOTES

None

12. APPENDICES

12.1. Schedule Tracking

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
RAS Business & Project Definition	Anish	5 Hours	6.5 Hours	1.5 Hours
	Mark	3 Hours	6.5 Hours	3.5 Hours

	Mehmed	7 Hours	6.5 Hours	- 0.5 Hours
	Summary (People Hours)	15 Hours	19.5 Hours	4.5 Hours

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
RAS Software Requirements	Anish	7 Hours	5 Hours	2 Hours
	Mark	8 Hours	5 Hours	3 Hours
	Mehmed	7 Hours	5 Hours	2 Hours
	Summary (People Hours)	22 Hours	15 Hours	7 Hours

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
RAS Complete	Anish	12 Hours		
	Mark	10 Hours		
	Mehmed	11.5 Hours		
	Summary (People Hours)	33.5 Hours		

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
Software Analysis Specification	Anish	10 Hours		
	Mark	10.5 Hours		
	Mehmed	9 Hours		
	Summary (People Hours)	29.5 Hours		

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
SPMP	Anish	12 Hours	13 Hours	1 Hour
	Mark	13 Hours	14 Hours	1 Hour
	Mehmed	11 Hours	11.5 Hours	.5 Hour
	Summary (People Hours)	36 Hours	38.5 Hours	2.5 Hours

Cumulative

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
	Anish	32 Hours		
	Mark	32 Hours		
	Mehmed	32 Hours		
	Summary (People Hours)	96 Hours		

12.2. Defect Tracking

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
RAS Business & Project Definition	Anish	7	12	5
	Mark	8	12	4
	Mehmed	6	12	6
	Summary (Avg.)	7	12	4

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference

RAS Software Requirements	Anish	6	12	6
	Mark	12	12	0
	Mehmed	5	12	7
	Summary (Avg.)	8	12	6.5

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
RAS Complete	Anish	16		
	Mark	16		
	Mehmed	14		
	Summary (Avg.)	15		

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
Software Analysis Specification	Anish	8		
	Mark	12		
	Mehmed	11		
	Summary (Avg.)	10		

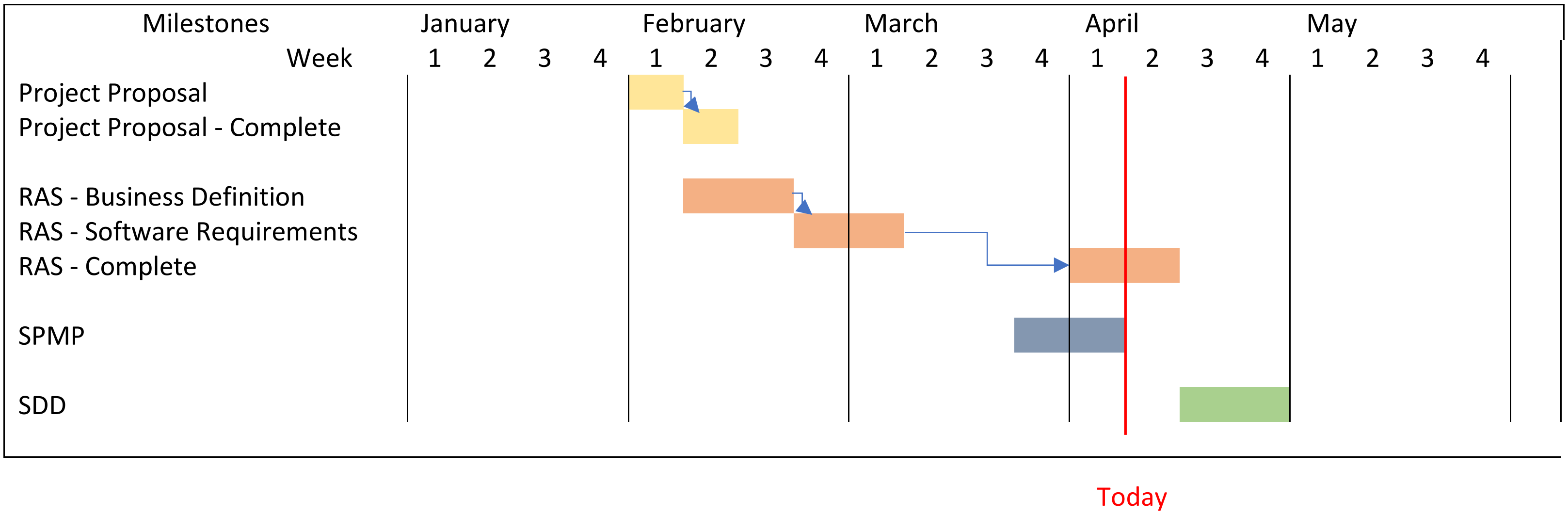
Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
SPMP	Anish	11		
	Mark	5		

	Mehmed	9		
	Summary (Avg.)	8		

Cumulative

Artifact or Deliverable	Who (Individual or Team)	Estimated	Actual	Difference
	Anish	48		
	Mark	53		
	Mehmed	45		
	Summary (Avg.)	49		

12.3. Gantt Chart/Microsoft Project Schedule


Milestones
Week
January
February
March
April
May
Project Proposal
Project Proposal - Complete
RAS - Business Definition
RAS - Software Requirements
RAS - Complete
SPMP
SDD
Today