

Computer Science and Engineering

LinkedIncognito

Project Management Plan

Version 1.0

Document Number: SPMP-001

Project Team Number: A27

Project Team Members: Maria Morejon (mjm1175), Tehya Trusty (tt1747), AJ Osborn (ajo331)

REVIEW AND APPROVALS

<Team Members>	Function (Author, Reviewer, Approval)	Date	Signature
M. Morejon	Author	<11-11-21>	Maria Morejon
T. Trusty	Author	<11-11-21>	Tehya Trusty
A. Osborn	Author	<11-11-21>	AJ Osborn

REVISION LEVEL

<u>Date</u>	<u>Revision Number</u>	<u>Purpose</u>
<u><11-11-21></u>	Version 1.0	<u>Initial Release</u>

TABLE OF CONTENTS

1. OVERVIEW	1
1.1 Project Summary	1
1.2 Purpose, Scope, and Objectives	1
1.3 Assumptions and Constraints	2
1.4 Project Deliverables	2
1.5 Schedule and Budget Summary	3
1.6 Evolution of the Plan	3
2. REFERENCES	4
3. DEFINITIONS	4
4. PROJECT ORGANIZATION	4
4.1 External Interfaces	4
4.2 Internal Structure	4
4.3 Roles and Responsibilities	5
5. MANAGEMENT PROCESSES	5
5.1 Start-Up Plan	5
5.1.1 <i>Estimation Plan</i>	5
5.1.2 <i>Staffing Plan</i>	5
5.1.3 <i>Resource Acquisition Plan</i>	6
5.1.4 <i>Training Plan</i>	6
5.2 Work Plan	6
5.2.1 <i>Work Activities</i>	6
5.2.2 <i>Schedule Allocation</i>	6
5.2.3 <i>Resource Allocation</i>	7
5.2.4 <i>Budget Allocation</i>	7
5.3 Control Plan	7
5.3.1 <i>Requirement Control and Traceability</i>	7
5.3.2 <i>Schedule Tracking and Adjustment</i>	8
5.3.3 <i>Budget Tracking and Adjustment</i>	8
5.3.4 <i>Quality Control</i>	8
5.3.5 <i>Reporting Mechanisms</i>	8
5.3.6 <i>Metrics Collection Plan</i>	8
5.4 Risk Management Plan	9
5.5 Post Implementation Plan	9
6. TECHNICAL PROCESSES	9

6.1 Process Model	9
6.2 Methods, Tools, and Techniques	9
6.3 Infrastructure Plan	10
6.4 Product Acceptance and Migration Plan	10
7. SUPPORTING PROCESSES PLANS	10
7.1 Configuration Management Plan	10
7.2 Qualification (Verification and Validation) Plan	11
7.3 Documentation (Library) Plan	11
7.4 Quality Assurance Plan	11
7.5 Reviews and Audits	11
7.6 Problem Resolution Plans	11
7.7 Environment Management Plans	12
7.8 Process Improvement Plan	12
8. ADDITIONAL PLANS	12
9. INDEX	12
10. RATIONALE	12
11. NOTES	12
12. APPENDICES	12
12.1 Schedule Tracking	13
12.2 Defect Tracking	14
12.3 Gantt Chart/Microsoft Project Schedule	16

1. OVERVIEW

1.1 Project Summary

This document is intended for developers, project managers, clients, and the software quality group, and will provide a detailed outline of our development plans for LinkedIncognito. Specifically, we will discuss how our project will be organized and detail the management, technical, and supporting processes that we will be implementing. Information related to budget, staffing, and what will occur after the software is implemented will be discussed in a later release.

The development of LinkedIncognito is motivated by two factors: the inappropriate and biased hiring practices in STEM fields that skew the gender distribution towards men, and a lack of solutions to combat these issues. Currently, underrepresented groups do not have the ability to hide their gender identity from potential employers. This personal information introduces bias, whether subconsciously or not, into the hiring process. Our platform will give people a chance to be judged solely on their qualifications by minimizing the personal information employers have access to during the hiring process. We hope that our project will lead to a more diverse STEM workforce.

1.2 Purpose, Scope, and Objectives

Over the course of this project, we will develop LinkedIncognito, an online platform designed to eliminate implicit bias in the hiring process. Our goal is to have a functioning prototype by early May of 2022 that will demonstrate the key features of our website. Currently, it is unlikely that our project will reach the deployment stage.

The release of this project planning document will establish the roles and responsibilities of each team member, the development timeline for the project, and how the project will be carried out. Since we are still in the beginning stages of our project, budget estimations would most likely be inaccurate. In addition, it is difficult to establish a detailed plan for the conclusion of our project while it's still in its infancy. Future releases will include information on the project's budget and the project's conclusion once we have enough details to determine a more accurate approximation of the two.

LinkedIncognito will allow users to connect with potential employers using an anonymous profile that only features their skills, credentials, and work experience. We hope to combat the implicit bias that our users may face on other hiring platforms and during in-person networking events. To achieve this, our system will not disclose a user's identifying information (gender, race, etc.) until an employer has made an offer. The user may then elect to share their information with their potential employer.

In addition to creating an anonymous profile, our users will also be able to use our anonymous interviewing platform and our embedded email service which will provide another barrier between their personal information and a company. We will also remove identifying information from the resumes they choose to upload to our platform. Our site also caters to potential employers as they can search through a diverse database of profiles to find potential candidates for a position. Companies may also upload their results from a variety of implicit bias tests to attract users.

LinkedIncognito will allow users to perform the following functions: search for employers or potential employees in a database, publish results of implicit bias tests, interview using the Zoom application, send emails through our website, send messages using our on-site messenger, send an introduction including personal information to a potential employer once an offer has been made. Our system is designed to aid users in making connections in a bias-free environment and increase diversity in the technology industry. However, we will not provide interview preparation or resume reviews. It is the user's job to seek out any additional help or resources they'll need over the course of their job hunt.

1.3 Assumptions and Constraints

We began working on this project with the assumption that companies and their potential employees would want to use our platform to connect with each other. It is our hope that the possibility of a more diverse workforce and good publicity will attract businesses. As more businesses join our platform, more applicants will want to make an account and connect with them.

We are constrained in several aspects: budget, time, and external software. Our budget is limited to what the Computer Science department provides for us. However, we don't anticipate needing a large budget for this project. The team will likely remain the same over the course of the project and our platform will not require expensive external resources. We must also conclude our project at the end of the Spring 2022 semester regardless of where we are in the project timeline. We believe that a functioning prototype is possible within that time frame, but it is unlikely that we will be able to deploy our platform. Since our interviewing feature relies on Zoom, we are limited to what Zoom allows us to do.

1.4 Project Deliverables

Deliverables	Date Due
--------------	----------

Project Proposal	9/27/2021
Software Business Specification (SRS - Business Definition)	10/07/2021
Software Requirement and Analysis Specification (SRS - Requirements)	10/19/2021
Software Project Management Plan (SPMP)	11/11/2021
Software Analysis Specification - Final (SRS - Analysis)	11/23/2021
Software Design Document (SDD)	12/09/2021
Final Project Presentation	12/09/2021

1.5 Schedule and Budget Summary

Deliverables	Date Due	Status
Project Proposal	9/27/2021	Delivered

Software Business Specification (SRS - Business Definition)	10/07/2021	Delivered
Software Requirement and Analysis Specification (SRS - Requirements)	10/19/2021	Delivered
Software Project Management Plan (SPMP)	11/11/2021	Delivered
Software Analysis Specification - Final (SRS - Analysis)	11/23/2021	Not Started
Software Design Document (SDD)	12/09/2021	Not Started
Final Project Presentation	12/09/2021	Not Started

The budget will be provided in a later release.

1.6 Evolution of the Plan

There are no plans for scheduled updates at this time. If there are any changes to the project, the software project management plan will be updated, inspected by the team, and redistributed to the necessary parties.

2. REFERENCES

Team A27, *mjm1175_Project Proposal*, PP-01, V 1.0 09-27-21

Team A27, *A27_CS4513_SRS_001*, SRS-001, V 1.0 10-08-21

Team A27, *A27_CS4513_SRS_001*, SRS-002, V 1.1 10-28-21

3. DEFINITIONS

None at this time.

4. PROJECT ORGANIZATION

4.1 External Interfaces

The software quality group is external to our project. This is a separate team that will evaluate the system to ensure that the functionality meets the standards set. To make sure that the system is reaching the benchmarks during the development process, a group to develop the test environment will be outsourced. Other external interfaces include Zoom. We will not be able to control the features that Zoom offers to its clients. We encounter the same situation with Email and the Implicit Bias Test. These are external interfaces that place boundaries on the project. We have to build the system around these.

4.2 Internal Structure

Everyone on the team holds equal weight and responsibility. We will all have complete access to the Github repository and all documentation. All members will be kept up to date on any and all updates related to the project. All decisions made by the team must be unanimous. All members of the team will be involved in each iteration together,

working on their assigned tasks for that activity. This allows for collective ownership of the code and collaboration amongst the development team.

4.3 Roles and Responsibilities

Author	M. Morejon, A. Osborn, T. Trusty
Developer	M. Morejon, A. Osborn, T. Trusty
Inspector	M. Morejon, A. Osborn, T. Trusty
Submit	T. Trusty
Test Developers	M. Morejon, A. Osborn, T. Trusty
Documentor	A. Osborn
Project Manager	M. Morejon

5. MANAGEMENT PROCESSES

5.1 Start-Up Plan

Our team was self-selected and there are no plans to hire additional team members at this time. In order to select a project, we discussed our interests and listed the topics that were most important to us. We all agreed that addressing gender inequality was a worthwhile cause for our project. After deciding on our project, we drafted a project proposal and submitted it for review. Our project was approved.

5.1.1 Estimation Plan

Not specified in this release.

5.1.2 Staffing Plan

Not specified in this release.

5.1.3 Resource Acquisition Plan

Not specified in this release.

5.1.4 Training Plan

No training is required at this time.

5.2 Work Plan

This section specifies the work activities, schedule, resources, and budget details for the System/software project.

5.2.1 Work Activities

Three outside systems that we will be integrating into our's will be: Zoom, Email, and an Implicit Bias test. During the application process, we will link the Implicit Bias test for the employer to take. Once the employer completes the test, they will submit their results for review. The registration function will have an estimated duration of three weeks to develop. It will include, in order:

- The database that stores all the user information from registration for validation during sign-in will be created
- The interface will be designed for user input
- The data that is inputted by the user will be encrypted and stored in the database

Email will be integrated into the system to allow for communication between the applicant and the employer. Since the application process we are providing is void of any identifying information, we will have to process the emails to remove any of this information. That will include: hiding email addresses, ensuring that their signatures along with the rest of the message are absent of names and other identifiable information. The message will then be sent. The messaging process function will allow for the applicant and employer to set up interviews. This development will take approximately two weeks and will include, in order:

- Integrating emails into our system
- Removing email addresses
- Filtering out any identifiable information (key words) before sending the email

- Send the email and validate that it has been sent

Zoom will be integrated into the system to provide the applicant and employer a means through which they can conduct their interview. The duration of this will be one week. It will include, in order:

- Integrating Zoom into the system
- Creating a microphone extension that works with Zoom to alter the voice of the applicant

As for the internal features for the application. We will need to develop the applicant profile. Here, the applicant can provide their resume, connections to different companies, projects they have worked on, the skills they have, any special qualifications they might have, and express jobs that they are interested in. For the profiles of the employers, they will be able to provide a bio for their company, job openings that they have and in what locations, a button for the applicant to apply, and filters for them to select from. This will take approximately two weeks to develop. There will also be the feed, where applicants can search for jobs. They will be able to filter their feed so they can find jobs that better fit their needs. This will take approximately three weeks.

- Applicant Profile
 - Take information from the registration process to create a base profile
 - Integrate the database
 - Display their resume on their profile
 - Allow them to edit their profile and update the information in the database
- Employer Profile
 - Take information from the registration process to create a base profile
 - Integrate the database
 - Display job offers and filters on their profile
 - Display company bio on profile
 - Allow them to edit their profile and update the information in the database
- Apply Button
 - Display any questions that the employer submitted for their application process
 - Provide a field for the applicant to input their answers
 - Send the application over to the employer for review with the resume of the applicant
- Job Feed
 - Take jobs from the database and display them on the applicants feed
 - Provide a filter button that allows the applicant to filter out jobs that do not relate to their personal job search
 - Offer related jobs to their filters and their submitted applications

5.2.2 Schedule Allocation

See Gantt Chart in 12.3 for details.

5.2.3 Resource Allocation

Internal Components

- Software and system developers
- Project Manager
- Prototyping

External Components

- Software and system developers
- Project Manager
- Lawyer to create contracts with external parties (Zoom)

Validation

- Test case developers
- Test cases for each iteration of the system
- Project Manager

5.2.4 Budget Allocation

Not required in this release.

5.3 Control Plan

This subsection specifies the metrics, reporting mechanisms, and control procedures necessary to measure, report, and control product requirements. the project work schedule, budget, resources, and the quality of development processes and work products (M).

5.3.1 Requirement Control and Traceability

This subsection specifies the control mechanisms for measuring, reporting, and controlling changes to the product requirements. This subsection will also specify mechanisms used in assessing the impact of requirement changes on schedule, budget, and resources, and risk factors. Configuration Management techniques should include change control procedures and a change control review process. Techniques that are used for requirements control include traceability,

prototyping/modeling, impact analysis, and reviews (M).

5.3.2 Schedule Tracking and Adjustment

This section specifies the control mechanisms used to measure the progress of work completed at the major and minor project milestones, to compare actual progress to planned progress, and to implement corrective action when actual progress does not conform to planned progress. The schedule control plan will specify the methods and tools used to measure and control schedule progress. Achievement of schedule milestones should be assessed using objective criteria to measure the scope and quality of work products at each milestone (M).

5.3.3 Budget Tracking and Adjustment

Not required in this release.

5.3.4 Quality Control

This subsection specifies the mechanisms used to measure and control the quality of the work processes and resulting work products. Quality control mechanisms may include quality assurance of work products, qualification (verification and validation), reviews, audits, and process assessment (M).

5.3.5 Reporting Mechanisms

Not required in this release.

5.3.6 Metrics Collection Plan

This subsection specifies the methods, tools, and techniques used in collecting and retaining project metrics. The metrics collection plan will specify the metrics to be collected, the frequency of collection, and the methods used in validating, analyzing, and reporting the metrics. This plan should be consistent with the SEPG policies and standards for retention of project management data (M).

5.4 Risk Management Plan

For a detailed breakdown of the potential risks associated with this system, see the SRS Document.

Most of the risks will be identified through error handling in software implementation. Errors that are critical to the running or main functionality of the system will be raised this way. This means that the coding/development team will be the first ones notified if any critical risks occur. It is their responsibility to notify the appropriate team members

based on the risk's projected impact.

The on-going identification of risks is a responsibility of all team members. As the product is developed, weekly meetings will include a risk assessment topic so that new risks posed by new features can be determined before the feature's implementation has begun.

- Roles and Responsibilities
 - Project Manager:
 - Maintaining the risk management plan
 - Keeping the team up to date with the status of risks
 - Facilitating risk assessment
 - Project team:
 - Identifying, monitoring and managing risks
 - Updating risk assessment when necessary
 - Assigned mitigation tasks
 - Coding/Development team:
 - Implementing risk management tasks
 - Report new risks
 - Assigned mitigation tasks

5.5 Post Implementation Plan

Not specified in this release.

6. TECHNICAL PROCESSES

6.1 Process Model

Our team will be applying the waterfall model to our development process. We began by establishing the requirements for our software and drafting a software requirements specification document. It was reviewed and modified to further develop the information presented in the document. Our next deliverable will be a final analysis of our established requirements. Once that has been distributed and reviewed, we will begin designing our online platform. Following a review of the design, the team will begin coding LinkedIncognito. The system will then be tested for defects. We hope to reach this milestone by early May of 2022. If we achieve this goal earlier than expected, we will begin working on deploying the system and patching any issues that may arise in the client's environment.

6.2 Methods, Tools, and Techniques

In order to collaboratively develop and maintain project deliverables, our team will use Google Docs. The system models within each deliverable will be created using UML. The deliverables will be stored in Google Drive. Once a deliverable is complete, a member of the team will upload a copy under the appropriate assignment name in Brightspace.

To develop LinkedIncognito, we will follow an object-oriented development methodology. Our product will be stored in a Github repository where each team member will be able to upload code, modify the system, and approve changes made by other team members. Changes must be approved by the entire team in order for them to take effect.

6.3 Infrastructure Plan

During the development cycle of LinkedIncognito, we will use our personal computers to design, develop, and test our project. We do not anticipate needing any other hardware for our project. If the need does arise, we will modify our infrastructure plan with the technology we needed and who it was provided by.

Our platform will use a teleconferencing feature provided by Zoom as the basis for our interviewing system. Other software that will be necessary for our platform to function correctly will be developed by our team or will be discovered and reused by our team. Once discovered, we will update our software project management plan with the necessary details.

6.4 Product Acceptance and Migration Plan

Not specified in this release.

7. SUPPORTING PROCESSES PLANS

7.1 Configuration Management Plan

In order for everyone on the team to be able to access the documentation and code for the system, the repository will be accessible through Github. Here, we can find the different components of the system. All changes that will be made to the system must be approved by all members of the team: M. Morejon, A. Osborn, and T. Trusty. If the changes are not approved by all, the discussion will continue until the changes are agreed upon by everyone. Once this is completed and approved by everyone, the changes will be implemented by the team member designated to the task. It will then be uploaded to the Github repository to be accessed by everyone on the team. The component must be completed, tested, and proper documentation must be completed before moving on.

Afterwards, integration of the components follows. The system will be tested by those who developed the test cases to ensure that the functionality of the system is up to the standard of the customer. When all the test cases are passed and the system is deemed up to standard, it will be implemented into the system as a whole. Programmers will work in groups each step to keep everyone in communication during the development process and mitigate errors in the system.

If users are experiencing any issues with the system after implementation, they will be able to report. There will be a Report Problems option on the taskbar. Reports will be sent to the designated team to handle these complaints. They will assess the feedback and determine its importance and what team the report needs to be sent to in order for the changes to be implemented. Once the report is sent over, the team will follow the previously outlined steps for making changes to the component and then integrating that into the system.

7.2 Qualification (Verification and Validation) Plan

Developers will be divided up into teams in order to allow for peer review. This will lower the chances of the developers encountering hurdles that they might not be able to overcome. Different perspectives and interpretations of the code will be helpful in finding the errors. Same with the developmental process, the verification process will also be done in teams. This is to ensure that nothing is missed during the verification process that can result in a false positive verification of the component. The verification process will be completed by those who developed the test cases. Development will be completed at the end of a milestone, which will be set similar to the SCRUM process; in small increments over a period of a few weeks. The code will be reviewed at the end of each milestone where it will then be sent off for testing. The specific checkpoints are outlined in Sections 7.5 and 7.6.

7.3 Documentation (Library) Plan

The documents will be accessible through the Github repository. The documentation will occur alongside the development of the different components of the system. Extensive documentation will prevent the duration of development from extending. Along with the documentation, which will provide a detailed account of the functionality of the component/system in development, the code will also be heavily commented on. The documentation will also include:

- Architectural descriptions
- Specification of the design of the backend and the interface
- Traceability

This alongside the documentation will allow for a better reading experience and understanding of the component/system. It will be generated and reviewed by the developers working the allocated segment of the system. It will then be sent to the other members of the development team as a whole for approval. They will check that each part of the component/system being documented is detailed enough for them to

understand its complete functionality since they did not contribute to the development process of this specific component.

7.4 Quality Assurance Plan

Prior to analysis of the quality assurance, the documentation of the requirements specification and the software project plan must be heavily detailed, covering exactly what the system will do, so we can more accurately assess the quality of the system at any stage in the project. The requirements specification and project plan will be used for analysis of the system's functionality. The review and audit process that will be used for quality assurance is outlined in the next section.

7.5 Reviews and Audits

- List of internal reviews/audits:
 - Organizational Management Reviews
 - Scope: The managerial organization of the project, the project documentation, the project requirements and stakeholder needs
 - Auditors: Independent consultants
 - Checklist:
 - Do requirements exhibit a clear distinction among functions and data?
 - Do requirements exactly define all the information that is needed to be displayed to users?
 - Do requirements address system and user response to all conditions of error?
 - Is each of the requirements stated clearly, concisely, and unambiguously?
 - Is each of the requirements testable?
 - Are there any ambiguous or implicit requirements?
 - Are there any conflicting requirements?
 - Are all non-functional requirements identified?
 - Timeline: Once monthly
 - Developer Peer Reviews
 - Scope: Current sprint task; two groups of pair programmers will swap tasks and review each other's work.
 - Auditors: Fellow code/development team members
 - Checklist:
 - Is the code understandable?
 - Is there duplicate or redundant code?
 - Is the unit testing and debugging system well-set-up?
 - Is there any refactoring that needs to be done for more maintainable code?
 - Timeline: Bi-weekly
 - Walkthroughs
 - Scope: Current sprint task
 - Auditors: Development team members
 - No checklist.

- Timeline: Frequently, at author's discretion. At least once per sprint.
- Inspections
 - Scope: Current iteration/system component
 - Auditors: Managers, stakeholders, internal auditing team
 - Checklist:
 - Overview of completed tasks since last inspection
 - Preparation
 - Code inspection
 - Suggested rework
 - Discussion and findings meeting
 - Timeline: After each increment has been developed, before testing begins, i.e. after a few sprints.
- Audits
 - Scope: The entire system, the project, and the documentation
 - Auditors: Independent consultants
 - Checklist:
 - Compliance with specifications
 - Compliance with organizational standards
 - Compliance with contractual agreements
 - E.g. Timeliness, budgetary, documentation
 - Timeline: Infrequently; once per release

7.6 Problem Resolution Plans

Some examples of problem resolution plan organizations are as follows.

Product Problems:

- Security Failure: Malicious users hack into the system and gain user information

Resolution Plan:

- Identifying Problem:
 - Raised `AccessLeak` will notify the coding/development team.
- Defining and Implementing Resolution:
 - Coding/development team will temporarily stop site deployment.
 - They will determine the source of the leak, and attempt to identify the responsible party.
 - They will then begin work to repair the leak and gather the compromised information.
 - They will notify affected users and re-deploy the system.
- Resolution Timeline:
 - System deployment stops immediately after leak is detected.
 - The problem resolution team will determine the source and gather information on responsible party within the first day after the leak is revealed.
 - The problem resolution team will repair the leak and re-deploy the system within one week of the leak's discovery.
- Verifying Problem Solution:

- The head engineer and the project manager will verify that the leak has been resolved and contact the authorities with information about responsible party.

Process Problems:

- Zoom Application goes out of service/System server crashes: Unable to connect the applicant and the employer for an interview and/or to the system itself

Resolution Plan:

- Identifying Problem:
 - Raised `FatalRedirect` will notify the coding/development team.
 - Coding/development team will notify management.
- Defining and Implementing Resolution:
 - Management will determine the severity of the outage -- has Zoom gone out of business, or is a server temporarily out of service.
 - An automated email is sent to all users informing them of the outage and the expected resolution timeline.
 - If an alternate video conferencing solution must be deployed, the management team will determine the best option and hand the refactoring task to the coding/development team.
 - When the server is confirmed to be working again, or the new solution has been implemented, another email will be sent to users informing them that the feature is back online.
- Resolution Timeline:
 - The severity of the problem should be determined on the first day.
 - The automated email is sent as soon as the error is raised.
 - If a new solution is required, the solution proposal and system refactoring should take no longer than one week after the error was raised.
- Verifying Problem Solution:
 - The auditing team will be responsible for system checks to ensure that the functionality has returned to normal, working state.

In summary, the coding/development team is the first one to hear about any (high priority) runtime errors. Some issues can be fixed without immediately alerting management, but most resolution checks will require someone from management or someone from the auditing team to verify that the solution is working.

There will also be a Report a Problem page where users can report any non-fatal system errors (which are lower priority) . In that case, it will go to the customer service department who will decide on the problem's legitimacy and which team should be deployed to resolve it.

7.7 Environment Management Plans

Not required in this release.

7.8 Process Improvement Plan

Not required in this release.

8. ADDITIONAL PLANS

None on this release.

9. INDEX

None on this release.

10. RATIONALE

Not required on this release.

11. NOTES

None on this release.

12. APPENDICES

12.1 Schedule Tracking

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS - Domain	For each team member			
	M. Morejon	3	2	1

	T. Trusty	4	2	2
	A. Osborn	4	2	2
	Summary for entire team	11	6	5

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS - Requirements	For each team member			
	M. Morejon	2	4	-2
	T. Trusty	3	3	0
	A. Osborn	3	2	1
	Summary for entire team	8	9	-1

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS -- Analysis - Complete	Entry for each team member			
	Summary for entire team			

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SPMP	Entry for each			

	team member			
	M. Morejon	3	3	0
	T. Trusty	3	4	-1
	A. Osborn	4	5	-1
	Summary for entire team	10	12	-2

Cumulative

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
Cumulative - Summary	For each team member			
	M. Morejon	8	9	-1
	T. Trusty	10	9	1
	A. Osborn	11	9	2
	Summary for entire team	29	26	2

12.2 Defect Tracking

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
-------------------------	--------------------------	-----------	--------	------------

SRS - Domain	For each team member			
	M. Morejon	0	0	0
	T. Trusty	0	0	0
	A. Osborn	0	0	0
	Summary for entire team	0	0	0

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
SRS - Requirements	For each team member			
	M. Morejon	0	0	0
	T. Trusty	0	0	0
	A. Osborn	0	0	0
	Summary for entire team	0	0	0

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS -- Analysis - Complete	Entry for each team member			
	Summary for entire team			

Artifact or Deliverable	Who (individual and team)	Estimated	Actual	Difference
SRS -- SPMP	Entry for each team member			
	M. Morejon	0	0	0
	T. Trusty	0	0	0
	A. Osborn	0	0	0
	Summary for entire team	0	0	0

Cumulative

Artifact or Deliverable	Who (individual or Team)	Estimated	Actual	Difference
Cumulative - Summary	For each team member			
	M. Morejon	0	0	0
	T. Trusty	0	0	0
	A. Osborn	0	0	0
	Summary for entire team	0	0	0

12.3 Gantt Chart/Microsoft Project Schedule

