

**The A Team**

**Image Processing Tool for**

**Leidenfrost-Ratchet Systems**

**Plan Document for Version 2.0 (First Draft)**

February 16, 2014

**Authored By:**

Sanan Aamir

Romando Garcia

Anne Lam

James Rowe

Hieu Tran

Table of Contents

1. Introduction 2
   1. Purpose of Plan Document 2
   2. Project Scope and Objectives 2
      1. Statement of Scope 2
      2. Major Functions 2
   3. Overview of Document 3
2. Project Organization 3
   1. Process Model 3
   2. Team Structure 3-4
   3. Communication 4
   4. Reporting 4
3. Project Resources 4
   1. People 4-5
   2. Hardware and Software 5
4. Risk Management 5
   1. Likely Risks 5
   2. Risk Management Plan 5-6

5. Schedule 6

5.1 Task List and Resource Allocation 6, Appendix A

5.2 Time Line 7, Appendix B

5.3 Task Network Diagram 7, Appendix C

5.4 Objective Grading Sheet 7, Appendix D

6. Tracking and Control Mechanics 8

7. Glossary 8

8. References 8

**1. Introduction/Overview**

The plan document details the project planning for the entire development process.

1.1 Purpose of Plan Document

This plan document will set out the resources available to the project, the work breakdown, and the schedule of work for the second version of the Image Processing Tool.

1.2 Project Scope and Objectives

The scope comprises what we intend to implement and nothing more. The software's current capacity is to efficiently track a drop of liquid through images gathered from a high speed camera as it falls from an injection needle and travels along a ratchet surface. We intend to add to this according to the following sections.

. Systeml is pixels to real world distances tool tips/intuitive usesequencege in the sequencetion of the needle and surface lo1.2.1 Statement of Scope

The main objectives of this edition of the tool are to increase automation, improve the user interface, provide graphical data, and speed up processing.

1.2.2 Major Functions

Specific functions to be brought up in this version include:

* Automatic determination of needle and ratchet location
  + if camera position is constant, determine location once using first image in sequence
  + if camera position is altered, determine location for each image in sequence
* Removal of the base image calibration
* Alteration of drop image manipulation (remove white glare)
* Drop volume measurement for each image
* Graphing of various plots using the extracted data
* Improvements to the user interface, including tool tips/intuitive use
* Increased processing efficiency
* Conversion of pixels to real world distances

1.3 Overview of Document

This document restates the project’s objectives and follows with the project’s organization, resources, potential risks and subsequent plan for management, the schedule and tracking mechanics.

**2. Project Organization**

2.1 Process Model

The team will follow the Waterfall software development life cycle, following the organization laid out in Figure 1. This decision was made due to its simple use and easy to understand style. Also, the requirements are well established and unlikely to be significantly changed.

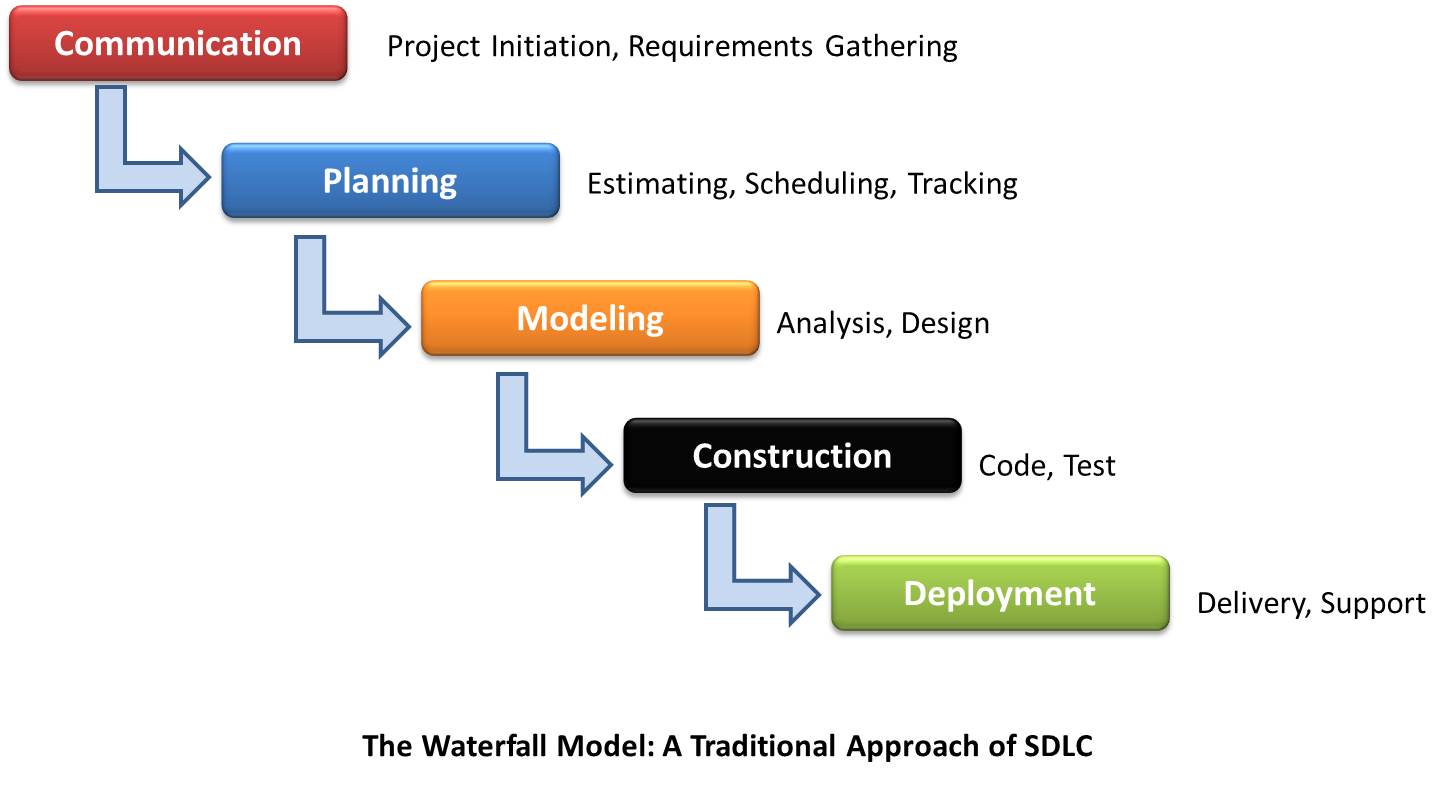


Figure 2. The Waterfall Process model, with communication, planning, modeling construction, and deployment stages.

2.2 Team Structure

Team organization will be controlled and decentralized. The team includes James Rowe as project leader, but executive decision making will not be primary to work flow. Everyone in the team inputs their ideas and opinions and decides best paths of actions together, with James as the final say. Additionally, all members contribute to documentation via primary authoring or editing. Coding will be split based on functionality, with two pairs of partners focusing on a particular set. James/Mando and Anne/Sanan comprise the partners and Hieu will focus on testing and documentation.

2.3 Communication

Communication will be achieved through a variety of pathways. Weekly meet-ups at the library for the entire team are planned to ensure adequate and correct work is being done. Email, Github, and texting are the main secondary avenues for communication. Additionally, quick meetings in class will ensure focus and reminders of tasks to be done.

2.4 Reporting

Reporting of progress will be done during meetings to all members, or to the leader, or through Github issue tracking. Additionally, interim and final reports/presentations given to Dr. Stringfellow and the Software Engineering class will mark our progress as well.

**3. Project Resources**

The resources to the project include the development team members and the hardware and software required to accomplish the development of the tool.

3.1 People

James Rowe is acting project leader and will perform the supervising role and give final say for split decisions. In addition, he and Mando Garcia will handle implementation of including frame rate in the Excel sheet, plotting the data in Excel with automatic graph generation, designing the User Interface, and handling inconsistent image locations. Anne Lam will do the primary authoring of the documentation with assistance from Hieu Tran, and with the rest of the team for editing. Anne and Sanan Aamir will implement the utilization of real world units (opposed to just pixels), calculating and outputting net measurements for velocity and acceleration as well as drop volume, and the fine-tuning of resulting locations of the needle and surface. Hieu is responsible for diagrams, testing and any additional needs as they come up.

3.2. Hardware and Software

Hardware use involves our personal laptops or desktops from where documentation and code will be written. The experiment’s set-up and the hardware associated are not included as the experiment has been completed and put away for the time-being.

Visual Studio and currently undetermined testing software will be used to aid our software development.

**4. Risk Management**

As with any endeavor, there are risks associated that may impede or impact the quality of this product. Recognizing the following potential risks, the team intends to prepare and work as organized and proactive as possible.

4.1 Likely Risks

We believe we may face these common obstacles:

* division of focus because of other responsibilities
* poor communication between team members
* poor time management of specific tasks
* scheduling conflicts between team members and possibly the client
* unforeseen emergencies involving health, family, etc.

4.2 Risk Management Plan

To avoid or alleviate the effects of the previously mentioned risks, the team has identified a few management techniques. These include performing proactive and focused work when given the opportunity in addition to when work is scheduled, and good organization and constant communication. Table 1 below organizes our assessment and plan for the potential risks.

**Table 1.** Potential obstacles that may risk the successful completion of development.

|  |  |  |
| --- | --- | --- |
| **Obstacle** | **Likelihood**  **(1-10 scale, 10 being most likely)** | **Mitigation Techniques** |
| Division of focus because of other responsibilities | 7 | Don’t procrastinate any assignments. Put software engineering first |
| Poor communication between team members | 3 | Use all available means of communication (email, text, face-to-face) |
| Poor time management of specific tasks | 5 | Be organized from the start, equipped with a plan. |
| Scheduling conflicts between team members and possibly the client | 2 | Warn team members of any conflicts that may arise |
| Unforeseen emergencies involving health, family, etc. | 1 | Do as much work while we can. |

1. **Schedule**

5.1 Task Lisk and Resource Allocation

The task lisk enumerates the individual aspects of the project that can be sorted independently and identified as a single function. Doing this allows for precise distribution of work. Additionally, resources (team members) must be allocated to help achieve completion of tasks. This information is given in Appendix A, attached at the end of the document.

5.2 Time Line

The time line illustrates when each task should be started and completed. Some tasks will be done concurrently, others are dependent on the completion of previous tasks. This information is detailed in Appendix B, also attached at the end of the document.

5.3 Task Network Diagram

The task network diagram illustrates the flow of the task schedule, detailing the pertinent dates and durations of each task. Note the critical path here, which outlines the list of tasks from start to finish that influence the most time, and where should any task on the path take longer than prescribed, deployment would be delayed. Find this diagram in Appendix C.

5.4 Objective Grading Sheet

The objective grading sheet organizes the division of labor between the team members in terms of effort distribution among the various aspects of the project. See Appendix D.

**6. Tracking and Control Mechanics**

Tracking of work will be done with regular status meetings in which team members report progress and problems. Additionally, all documents will be scrutinized in review meetings.

Control by the team leader will involve ensuring tasks are completed by their designated due dates. Overall, little control should be required as team members recognize the importance of pulling their weight. Should problems arise, the leader may possible reassign resources and redefine the schedule.

**7. Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Waterfall Process Model** | A sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a **waterfall**) through the phases of Conception, Planning, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. |

**8. References**

This document was completed with the guidance from the Plan Outline and Planning Powerpoints written by Catherine Stringfellow for the Software Engineering course at Midwestern State University.