The Team with the Big Guns
Special Weapons, NSWC Crane

# Project Plan


Andrew Houvener
Matthew Jacobs
Kyle Kopacz
Micah Weaver


May 24, 2009

# Contents

# 1 High Level Project Description

The Special Weapons Division at Naval Surface Warfare Center Crane (NSWC Crane) [2] is responsible for testing weapons that are used by the United States Armed Forces, specifically the Navy. In order to test these weapons, it is necessary to be able to quickly and effectively determine how a weapon's reliability, availability, and maintainability (RAM) functions. The weapon's accuracy is taken into account through RAM, as it is dependent upon both the availability (if the barrel is severely worn and must be replaced) and maintainability (as the barrel becomes dirty). Currently, nearly instantaneous results can be determined for super-sonic munitions on specialized ranges with special equipment. For every other situation NSWC Crane encounters, there is a scarcity of automated systems that can handle their weapons testing needs. This causes targets to be collected and held for analysis, and requires tedious hand measurements to be carefully plotted; these results are only as accurate as the person measuring it. Our system will address the need to automate the situations where an acoustic system cannot be used.

# 2 Team Organizational Structure

## 2.1 Team

Andrew Houvener, Team Leader, houvenal@rose-hulman.edu

Matthew Jacobs, Quality Control Analyst, jacobsma@rose-hulman.edu

Micah Weaver, Requirements Analyst, weavermn@rose-hulman.edu

## 2.2 Process Model

An incremental design process model will be used to develop the system. Our clients know many of the features that they want to see happen, but there are more features that we believe the system could incorporate to make for a better product for our client. In discussing these items with our client, items such as a data management system integrated into the software to manage the reports will only be added as time permits. Others items, like auto-guessing hole locations, are things that we know we can do that our clients had not initially wanted. After recommending these features that they did not initially want, the clients and project team have decided to include these into the final product. By using the incremental design, we will be able to incorporate the basic functionality that they want, and later add these features in on top of the previous version.

Another benefit to the incremental design will be the ability to test the accuracy of the system early in the development process. One of our primary risks is the accuracy of the system (see Section 6). By utilizing an incremental design, we will have the ability to test the algorithms used throughout the entire process instead of just at the end. We plan to test various algorithms individually in order to ascertain the feasibility of each one, and ultimately we will choose one or many algorithms to use for the final product. This will allow us to analyze our accuracy and provide many opportunities to document and correct them as we develop the system (see Section 2.3).

### 2.3 Prototypes via Sprints

As part of our incremental design model, we will incorporate sprints into the the fall term of the 2008-2009 school year. These sprints will serve as a manner by which we can prototype the key components of our system, as well as a few of the components that we have the least experience with. Specifically targeted with these sprints is the image recognition portion of the software.

#### 2.3.1 A Sprint

Each sprint will consist of two components: a development phase and a testing phase. During each sprint, an algorithm will be chosen from either a new set of algorithms or an existing algorithm. New algorithms will be implemented in C++ to be evaluated during the second phase of the sprint. Should an algorithm be chosen that has been previously implemented and has shown promise, the sprint must be used to significantly optimize the chosen algorithm.

At the end of development, the testing phase will begin. This phase will last for 3 to 4 days and will follow the guidelines in the following section on evaluation. Once completed, a brief analysis will be recorded, including the feasibility of implementing the algorithm, the estimated effort required to implement the algorithm, and, if applicable, how the algorithm could be improved or optimized.

#### 2.3.2 Evaluation

In evaluating each algorithm, the software will run a pre-defined set of test images five separate times. The tests will be performed on one of the senior Dell laptops, which is approximately the required hardware for the project, and the scores for each category will be recorded. Average, maximum, and minimum scores will also be noted. The following are the items that will be tested for each algorithm:

- Time - The time for processing the image will be recorded. This will not include the time it takes to display the information.
- Holes found - The number of holes found will be given in terms of the number of holes found out of the number of holes possible.
- False positives - The number of holes reported that are improperly recorded will be recorded as the number of misses compared to the number of holes possible.
- Accuracy and precision for each point - for each hole on each target, the accuracy and precision will be recorded. The values that should be noted are the percent error from the actual center of the hole as well as the percent error within the group.

The images to test will allow for testing a variety of features. The images will test images with matching backgrounds (white target with white behind the holes), high contrast with the background, 5mm holes, 10mm holes, distinct holes, and overlayed holes.

#### 2.3.3 Sprint Schedule

- 22 October 2008–2 November 2008: Develop Algorithms for Sprint Number 1.

- 3 November 2008–5 November 2008: Test and write-up results for first sprint.
- 6 November 2008–12 November 2008: Develop Algorithms for Sprint Number 2.
- 13 November 2008–15 November 2008: Test and write-up results for the second sprint.

# 3 Client Overview

Our clients are Charles Zeller, US Navy Technical Warrant Holder, Small Arms & Weapons Crane Division and Jeff Johnson, Joint Weapons Engineering Branch Manager, Crane Division. Our clients are responsible for small arms testing at NSWC Crane. They are average computer users that are primarily users of software and not developers. They deal mostly with web applications and Microsoft Office [12].

# 4 Team Communication Structure

Our teams main communications are in the form of impromptu meetings, briefly commenting on the project. This has been somewhat successful with coordination. On average, one formal meeting is held each week to help the team assess critical documents and plan for major events.

After fall break, as more collaboration is required, impromptu meetings will occur more frequently, and more formal meetings will be required. A division of labor to minimize meetings will still occur.

# 5 Schedule

- 10 October 2008: Drafts of documentation submitted to Sriram and Clients.
- 24 October 2008: Design for Prototype, Test plan for prototype completed
- 24 October 2008: Requirements approved by Charles Zeller, Jeff Johnson
- 12 November 2008: Algorithm analysis completed.
- 17 November 2008: Final documentation submitted.
- 20 February 2009: Submit Version 1 to clients.
- 4 May 2009: Final Acceptance Test.
- 25 May 2009: Submit final version to clients.
- 25 May 2009: Submit final documentation to clients.

# 6 Risks

In this section we outline the main risks that our project faces. In describing these risks, the risk of each occurring is listed, indicating either a low, medium, or high probability of it occurring. We define a low probability to be a risk that, if not accounted for, will most likely not have a noticeable impact on the system. A medium risk is one where it is likely to happen if unaccounted for, but designing to prevent it will mitigate the risk. A high risk we define as one where even if planned and accounted for, there is still a high probability that it will occur, and this level requires significant testing to prevent.

For the impact of each risk, they are categorized into low, medium, high, or critical. A low impact means that the user will likely not notice if it occurs, as other equal options will be available. Medium means that the risk occurring will be a nuisance, but the user could find a way to work around it. High impact means that the risk occurring will cause the program to not work correctly without a work around. A critical impact indicates a risk where, if it occurs, will most likely crash the program or make it unusable.

## 6.1 Accuracy

Risk: Medium

Impact: High

A primary risk of this project is the level of accuracy that can be attained from a single photograph of a target board. The accuracy that can be achieved will be contingent upon the resolution of the camera, the size of the target, the distance of the photographer from the target, and the angle at which the photo is taken.

Solution: In order to accommodate the client, the limits of the software will be tested and reported back to the client. These figures will include:

- A maximum angle at which the software will be able to perform the computations and maximum accuracies defined.
- A minimum resolution and approximate accuracies at various resolutions.
- Other documentation to otherwise indicate and document other parameters that may affect the accuracy of the system.

## 6.2 Distinguishing Shots

Risk: Low

Impact: Low

A minor risk that may be encountered is the program not being able to correctly auto guess individual bullet holes; specifically, the system may not be able to distinguish between two seemingly overlapping holes.

Solution: When the software cannot correctly determine the bullet holes automatically, or when the software incorrectly labels the holes, the user will be able to manually mark and unmark bullet holes (see Problem Statement, Section 1.2.1, Key Business Feature #5).

## 6.3 Usability

Risk: Low

Impact: Medium

As the end users of our system will not be computer experts, the system must be easily learned and usable. Should it not be usable, it will very likely not be incorporated into the routine weapons tests performed by the Navy.

Solution: This risk will be mediated via many usability tests with our client and potential end users of the system. Because we will be using an incremental development method, we will be able to correct any items that present issues with our end users.

### 6.4 Portability

Risk: Low

Impact: Medium

If the system cannot be used at any off site test range, the system may not be used by the Navy. It must be able to be used by testers at both permanent and impromptu firing ranges. Thus, creating a system that can be used as a standalone application is critical.

Solution: As the system is developed, we will incorporate languages and frameworks that allow for the systems portability.

### 6.5 Performance

Risk: Medium

Impact: Medium

The system will not be used if the performance of the software causes it to take longer to use than the current by hand method. Furthermore, if the system takes longer to perform than users are accustomed to base on their experiences with other programs, the system will not received the desired use.

Solution: In order to address the problem of performance, two strategies will be used. First, many revisions of image recognition algorithms will be used. Prototypes of this system will be implemented and accessed to determine the speed and efficiency of the algorithm (see Section 2.3).

## 7 Configuration Management

In order to facilitate the organization of our documents, the following items are the guidelines that we will use.

### 7.1 General Items

- Dates will be written following the military style DD Month YYYY (ex. 10 October 2008)
- All times will be recorded with a 24 clock (military time).

## 7.2 SVN Repository

All of the requirements documents for this project, code, supplementary materials, and any other course related presentation materials will be saved to a SVN repository hosted by team member Matthew Jacobs [13].

Requirements documents stored on the repository should be written in LaTeX. For circumstances where the client is required to comment on a document and must do this from home, Word documents may be temporarily required.

Documents will be uploaded by the responsible party as soon as possible in the respective folders. No documents critical to the project may exist outside of the repository (with the exception of being on the Trac wiki [14]).

The repository will be maintained by Matthew Jacobs. Any issues that occur, organizational structures, and version control issues will be handled by him.

## 7.3 SVN Layout

The subversion server will be organized in such a way to keep all relevant items clustered into logical groups. The details of each segment of the system are detailed below.

### 7.3.1 Code

This folder will contain any code developed by the team for this project. Relevant code may include sprint code from image recognition, development code, testing algorithms, and any other code used in the course of developing this project. The main project will be maintained in the trunk folder, branches will house any code in development, and tags will hold versions of code that exist at a specific milestone.

### 7.3.2 Documents

The documents folder contains all of the documents developed over the course of the project. Grouped by term, these folders contain the present architecture, requirements, project plan, and testing documents from the project.

### 7.3.3 Minutes

The minutes folder contains all of the minutes from fall quarter. To make access and creation simpler, winter and spring terms will have their minutes put on the trac system. Links to the word documents from fall quarter are accessible from here as well for the sake of legacy.

### 7.3.4 Presentations

For the midterm and final presentations each quarter, this folder houses both the presentation as well as the resources.

#### 7.3.5 Release Builds

As development in the Winder and Spring requires, weekly builds will be added to this folder. A script will copy the contents of this folder to a publicly accessible web directory.

#### 7.3.6 Supplementary

This section houses any miscellaneous items that may not otherwise have a home. This includes client questionnaires, summer phone calls, and mostly sample items from Crane and created by the project team.

#### 7.3.7 TEX

This section houses blank TEXfiles for use by the team, templates, and a TEXhelp pdf. These items should not change often over the course of the year.

### 7.4 Trac System

For questions for our client, status reports, bug tracking, tasking, and engineering journals, a trac system will be used. The track system is also hosted by team member Matthew Jacobs [14].

Tickets will typically be assigned to match the status reports by whoever writes the report. In the event of new items being added at a meeting, it will be the responsibility of whoever uploads the minutes to add the tickets.

## 8 Budget

The current budget for this project is the time available in the 2008-2009 school year to complete the software. As students, the developers will not require compensation, and as the software will be run on machines provided by the clients, there will be no need for money to purchase hardware.

## 9 Metrics

### 9.1 Process Metrics

The earned value for the project will be completed the week following fall break and included in this document.

### 9.2 Product Metrics

The finished system should perform faster than measuring a target and recording the coordinates of each shot.

Metrics like the bugs/kloc and coding standards can be found at
http://www.army.mil/ArmyBTKC/focus/sa/erp_standards.htm#3.

### 9.3 Acceptance Test

To be considered a success, the software must reliably be able to match the results yielded from an acoustic test and hand calculated test with an error less than or equal to 1%.

# References


[1] “Windows.” Microsoft Corporation, 2008, http://www.microsoft.com/WINDOWS/.

[2] “NSWC Crane Main Internet Page.” United States Navy, http://www.crane.navy.mil/defaulthome.asp.

[3] “Excel Home Page - Microsoft Office Online.” Microsoft Corporation, 2008, http://office.microsoft.com/en-us/excel/default.aspx?ofcresset=1.

[4] “Access Home Page - Microsoft Office Online.” Microsoft Corporation, 2008, http://office.microsoft.com/en-us/access/default.aspx.

[5] “JPEG JFIF.” Chris Lilley, World Wide Web Consortium, 09 Jan 2003, http://www.w3.org/Graphics/JPEG/.

[6] “Portable Network Graphics.” Chris Lilley, World Wide Web Consortium, 17 Mar 2006, http://www.w3.org/Graphics/PNG/.

[7] “USA.gov: The U.S. Government’s Official Web Portal.” United States Government, 08 Oct 2008, http://www.usa.gov/.

[8] “U.S. Department of Defense Official Website.” United States Department of Defense, 09 Oct 2008, http://www.defense.gov/.

[9] “NAVSEAHQ.” United States Navy, 10 Oct 2008, http://www.navsea.navy.mil/default.aspx.

[10] “Navy/Marine Corps Intranet Information Website.” United States Marine Corps, 19 Oct 2008, http://www.nmciinfo.usmc.mil/nmci2/nmci.nsf/HomePage?openform.

[11] “GIMP - The GNU Image Manipulation Program.” 3 Nov 2008, http://www.gimp.org/.

[12] “Office Online Home - Microsoft Office Online.” 10 Nov 2008, http://office.microsoft.com/en-us/.

[13] “Senior Project SVN.” 20 Oct 2008, http://dev.qlunq.com/svn/seniorproject/.

[14] “Senior Project - Trac.” 20 Oct 2008, http://dev.qlunq.com/trac/seniorproject/.

# Glossary

The **Joint Photographics Experts Group (JPEG)** [5] is a committee for image standards. JPEG is more commonly associated with an image file format. It is used for compression, but the compression causes some loss of data.

**Mach number** is the velocity of an object divided by the velocity of sound.

**Microsoft Excel** [3] is a spreadsheet program created by Microsoft Corporation.

**Microsoft Windows** [1] is a family of operating systems created by Microsoft Corporation. The operating systems provide a layer of abstraction between computer users and the underlying hardware.

**NAVSEA** [9] stands for Naval Sea Systems Command. This branch of the Department of Defense's mission is to develop, deliver, and maintain ships and systems on time, on cost for the United States Navy. It exists to serve the seaman.

**Naval Surface Warfare Center Crane (NSWC Crane)** [2] is the third largest naval base in the world. NSWC Crane's mission is to "provide engineering and technical services with a product focus in sensors, electronics, electronic warfare and special mission weapons..." [2].

**NMCI**, also known as Navy Marine Corps Intranet, supports all of the software and network solutions for the United States Navy. This is the department that will have to be consulted for final product distribution, as well as for database development.

**NSWC** stands for Naval Surface Warfare Centers. The NSWC centers are divisions of NAVSEA that perform specific duties within the NAVSEA organization. NSWC is the primary weapons depot for the United States Navy.

**Portable Network Graphics (PNG)** [6] is an image file format. It has no loss of image data.

**super-sonic** is defined to be any object traveling at or above 1.15 Mach.