Project Management Plan

for

Egg Flow Communicator

Version 1.0 draft 1

Prepared by Parker, Habegger, Rasler

Egg Flow Communicator Group

10/11/2011

Table of Contents

[1. Overview 1](#_Toc164670634)

[1.1. Project Purpose, Objectives, and Success Criteria 1](#_Toc164670635)

[1.2. Project Deliverables 1](#_Toc164670636)

[1.3. Assumptions, Dependencies, and Constraints 1](#_Toc164670637)

[1.4. References 2](#_Toc164670638)

[1.5. Definitions and Acronyms 2](#_Toc164670639)

[1.6. Evolution of the Plan 2](#_Toc164670640)

[2. Project Organization 2](#_Toc164670641)

[2.1. External Interfaces 2](#_Toc164670642)

[2.2. Internal Structure 2](#_Toc164670643)

[2.3. Roles and Responsibilities 3](#_Toc164670644)

[3. Managerial Process Plans 3](#_Toc164670645)

[3.1. Start-Up Plans 3](#_Toc164670646)

[3.1.1 Estimation Plan 3](#_Toc164670647)

[3.1.2 Staffing Plan 3](#_Toc164670648)

[3.1.3 Staff Training Plan 4](#_Toc164670649)

[3.1.4 Resource Acquisition Plan 4](#_Toc164670650)

[3.1.5 Project Commitments 4](#_Toc164670651)

[3.2. Work Plan 4](#_Toc164670652)

[3.3. Control Plan 5](#_Toc164670653)

[3.3.1 Data Control Plan 5](#_Toc164670654)

[3.3.2 Requirements Control Plan 5](#_Toc164670655)

[3.3.3 Schedule Control Plan 5](#_Toc164670656)

[3.3.4 Budget Control Plan 5](#_Toc164670657)

[3.3.5 Communication, Tracking, and Reporting Plan 6](#_Toc164670658)

[3.3.6 Metrics Collection Plan 6](#_Toc164670659)

[3.4. Risk Management Plan 6](#_Toc164670660)

[3.5. Issue Resolution Plan 7](#_Toc164670661)

[3.6. Project Close-Out Plan 7](#_Toc164670662)

[4. Technical Process Plans 7](#_Toc164670663)

[4.1. Process Model 7](#_Toc164670664)

[4.2. Methods, Tools, and Techniques 7](#_Toc164670665)

[4.3. Configuration Management Plan 7](#_Toc164670666)

[4.4. Quality Assurance Plan 8](#_Toc164670667)

[4.5. Documentation Plan 8](#_Toc164670668)

[4.6. Process Improvement Plan 8](#_Toc164670669)

# Overview

<This section provides an overview of the project’s motivation, objectives, success criteria, major deliverables, and constraints. You might include a top-level summary of major milestones, required resources, schedule, and budget. >

## Project Purpose, Objectives, and Success Criteria

In the Egg Flow Communicator Project, our primary goal is to deliver a working software and hardware solution to the problems that result from abnormal flow of chicken eggs in the collection and packaging process. The solution will offer real time tracking of egg flows on separate conveyors, it will determine when there is a flow problem, and alert a user when and where it occurs in real-time, allowing for a prompt solution and limiting the propogation of the problem over the conveyor.

The product’s success will be determined by the amount of overall labor reduced in the egg collection process. Given the general cost of labor in the chicken egg packaging process, mechanization principles can increase the profit potential of the business. This project is being delivered primarily for the needs of Habegger Poultry, of which, the proprietor, is acting as the sponsor. The company has determined that egg jams occur in such a way that the time required to fix the problem is proportional to the time between the initial problem and the when the problem is noticed. Due in part by the large area that the farm occupies, problems in egg flow are not readily noticed.

This product will be developed and integrated into the preexisting conveyor system of Habegger Poultry with the guidance and assistance of Mr. Tim Habegger, who will also be playing a primary role in the development of the hardware required.

## Project Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable** | **Recipients** | **Delivery Date** | **Delivery Method** | **Comments** |
| Module Driver Software | Development Team and Sponsor | November, 2012 | Software provided | Includes both on Module PBASIC software and PC communication driver |
| Programmed Hardware Module Prototype | Development Team and Sponsor | November, 2012 | Hardware provided | Pre-Requisite is Module Driver Software |
| Programmed Hardware Modules | Customer/Sponsor | May, 2012 | Integrate into existing components | Pre-Requisites are Module Driver Software and Module Prototype |
| Application Architecture | Project Team/Program Manager | 9/7/2011 | Email/Cmap | Assigned to Andrew Habegger/ Backups: Mark Parker and Matthew Rasler |
| GUI | Customer/Sponsor | May, 2012 | Integrate into existing components | Pre-Requisite is Module Driver Software |
| SRS V1.0 | Project Team/Program Manager | 9/23/2011 | Email/Cmap | Assigned to Matthew Rasler/Mark Parker |
| SRS v2.0 | Project Team/Program Manager | 9/25/2011 | Email/Cmap | Sections Distributed amongst team members by Project Manager |
| FR-DP Decomposition | Project Team/Program Manager | 9/29/2011 | Email/Cmap | Sections Distributed amongst team members by Project Manager |
| Design Matrix | Project Team/Program Manager | 9/29/2011 | Email/Cmap | Assigned to Project Manager |
| FMEA/RISK Report | Project Team/Program Manager | 10/3/2011 | Email/Cmap | Accomplished by group during team meeting |
| PMP V1.0 | Project Team/Program Manager | 10/10/2011 | Email/Cmap | Assigned to Matthew Rasler/Andrew Habegger |
| **Version 2.0** | | | | |
| PMP V2.0(this) | Project Team/Program Manager | 10/19/2011 | Email/Cmap | Assigned to Matthew Rasler/Andrew Habegger |
| IEEE 1016 | Project Team/Program Manager | 10/31/2011 | Email/Cmap | Assigned toMatthew Rasler/Mark Parker |
| UML Diagram | Project Team/Program Manager | 10/25/2011 | Email/Cmap | Assigned to Andrew Habegger/Matthew Rasler |
| House Of Quality | Project Team/Program Manager | 10/25/2011 | Email/Cmap | Assigned to Mark Parker/Matthew Rasler |

## Assumptions, Dependencies, and Constraints

Assumptions:

AS1-Collective software and hardware knowledge base of group is sufficient

Dependencies:

DE1-Hardware resources available, including cables, counters, modules, etc…

DE2-Fabrication of modules from prototype timely

Constraints:

CO1-Hardware supplied by Sponsor regulates software used

CO2-Pre-existing PC requires software created for the Windows OS

## References

Datasheets for Hardware Components

UART:

<http://www.maxim-ic.com/datasheet/index.mvp/id/2052>

BS2P40: <http://www.parallax.com/Portals/0/Downloads/docs/prod/schem/BS2p40SchematicRevD.pdf>

<http://www.parallax.com/Portals/0/Downloads/docs/prod/stamps/web-BSM-v2.2.pdf>

UART-BS2P40 Integration:

<http://www.wd5gnr.com/suart.htm>

Software

BASIC Stamp Windows Editor

<http://www.parallax.com/tabid/441/Default.aspx>

## Definitions and Acronyms

EFC- Egg Flow Communicator (this project)

UART- Universal Asynchronous Receiver Transmitter

BS2P40- BASIC Stamp 40 Pin Microcontroller, developed by Parallax Co.

MAX3110E-CNI- Specific UART used, RS-232 transceiver, developed by Maxim Co.

RS-232- Recommended Standard 232, serial binary communication standard

RS-485- Recommended Standard 485, serial binary communication standard

## Evolution of the Plan

This plan will be updated if constraints change, or if resource dependencies fall outside of expected/necessary timeline.

# Project Organization

## External Interfaces

Andrew Habegger, team member, acts as Customer Interface; he interacts with the Customer, Tim Habegger, to define and communicate the Functional Requirements of the system.

The Project Supervisor, Professor U. John Tanik, communicates the scholarly requirements of the project through class sessions, email, and document interfaces and templates. He directs communication to the Project Manager, Matthew Rasler.

The Project Manager, Matthew Rasler, sets up weekly face-to-face out-of-class meeting times, communicates deadlines, and parses work to team members. Email, phone, weekly face-to-face meetings, and class sessions are the primary interfaces used in communication.

## *C:\Users\mdras\Documents\school\Software\RoleChart.jpg*Internal Structure

## Roles and Responsibilities

Sponsor/

Customer/

Hardware Developer- Tim Habegger, Proprietor of Habegger Poultry

Project Supervisor- Professor U. John Tanik

Project Manager/

Low Level Programmer- Matthew Rasler

System Designer/

Hardware Developer/

Customer Interface- Andrew Habegger

System Analyst/

Programmer- Mark Parker

# Managerial Process Plans

## Start-Up Plans

### Estimation Plan

<elaborate>

### Staffing Plan

Designing Module Prototype:

2 Staff Members @ 2 Weeks

Systems Analyst

Low Level Programmer for semantic concerns

Hardware Developer

Developing Software for Modules:

2 Staff Members @ 2 Weeks

Low Level Programmer

Hardware Developer

Debug and Test Modules:

1 Staff Member @ 1 Week

Low Level Programmer

Designing GUI

2 Staff Members @ 1 Week

High Level Programmers

Develop GUI

2 Staff Members @ 2 Weeks

High Level Programmers

Designing Driver Software

3 Staff Members @ 2 Weeks

High Level Programmer

Low Level Programmer

Hardware Developer

Developing Driver Software

3 Staff Members @ 2 Week

High Level Programmer

Low Level Programmer

Hardware Developer

Test and Debug Driver Software

1 Staff Member @ 1 Week

Programmer

Integrating Components

3 Staff Members @ 1 Week

Hardware Developer

Programmers

### Staff Training Plan

Staff training will be conducted independently as needed, and as assignments and goals are delegated time will be allocated to allow for acclamation and personal training. Time will be allocated for education on hardware standards for communication (RS-242 and RS-485), hardware specifications for the microcontroller (Parallax BS2P40) and the UART (Maxxim MAX3110E-CNI). Also time will need to be allotted for training in low level languages (Parallax BASIC, etc…). As personal training develops for a specific group member, time will be allocated to educate group members on the subject. It will remain however, that individuals who are assigned training time will become the official resource for that topic in the collective.

### Resource Acquisition Plan

* Development resources:

Test modules including microcontrollers and communication chips.

This is a pre-requisite for developing the prototype module, the specifics are directed by the sponsor, and will be acquired by the sponsor.

Development environments.

Microcontroller environments provided online by microcontroller manufacturers, other environments should already be in place.

Fabricated modules.

To be fabricated by hand, after prototyping is complete.

* Test resources:

Acquired.

* Product resources:

Hardware components, including cable, switches, modules, RS-485 to USB convertor, potentially RS-232 to RS-485 convertor, etc…

To be acquired as needed by Sponsor, it is assumed (DE1) that the acquisition time does not interfere with the general timeline.

### Project Commitments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Commitment | Made By | Made To | Due Date | Comments |
| Provide working solution | Group | Customer | May 1, 2012 |  |

## Work Plan

Provided by Gantt Chart, and CMAP:

<http://www.students.ipfw.edu/~raslmd01/RaslerSoftEngCmap.html>

## Control Plan

### Data Control Plan

During the development and collection of data, all documents, timelines, to-do lists, will be stored on a group accessed private server using the tool Basecamp. After data and documents have been finalized, but before the final publication, all data will be housed on a public group website created from the group’s CMap, as listed above. During the completion of the RUP described Elaboration Phase, a web page will be developed to elaborate progress, where all data and documents will be accessible.

### Requirements Control Plan

Minor requirement changes will be incorporated on-the-fly, and informally. If the requirements specified change the feasibility timeline, then a new timeline should be developed.

### Schedule Control Plan

A Gantt chart will be used to set and measure progress at milestones. The progress will be monitored weekly, and weekly objectives will be elaborated at a weekly meeting. Progress will be monitored weekly to minimize the potential of time being underestimated for tasks. Also the phases of the RUP process will be added to the Gantt chart to benchmark the overall progress of the project. Slack time will be added to the transition between these RUP phases for catch up.

### Budget Control Plan

The budget will be monitored and controlled by the Sponsor, who will be supplying the products needed to complete the project. Labor cost is not measured in this project.

### Communication, Tracking, and Reporting Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of Communication | Communication Schedule | Typical Communication Mechanism | Who Initiates | Recipient |
| Status Report | every Friday | team meeting face to face | Project Manager | Project Team |
| Schedule and Effort Tracking Report | weekly | Email/CMap/Basecamp | Project Manager | Program Manager |
| Project Review | At infrequent intervals | In class and via email | Program Manager/Project Sponsor | Project Team |
| Requirement Changes | as changes are approved | Via client interface | Project Sponsor | Project Team |
| Information or Knowledge Collected | When information or knowledge is collected | Team meeting face to face and email | Team Member | Other Team Members |

### Metrics Collection Plan

Microsoft Gantt Chart will elaborate the time consumed by each sub-project or task, as well as the time utilized for the project as a whole. It will also operate as a communication tool to describe overall project status. Also, a tablical form will be used to describe each week’s goals, deadlines, and task status, initiated by the Project Manager, it will be discussed at weekly project meetings. This form will be available to the public via a web interface (CMap), in particular it will be available to the Project Team Members, Program Manager, and the Project Sponsor.

## Risk Management Plan

The Axiomatic Design Software Tool: Acclaro provides capabilities to elaborate and analyze risk areas of the project. This information can be found elaborated on the project teams CMap. During the RUP elaboration phase, as new design parameters are defined, new FMEA reports will be created to identify potential risks.

The software will also use logging to help identify erroneous conditions after deployment.

## Issue Resolution Plan

As requirements change, the Client Interface will communicate these changes to the team, and a team decision will be made whether the requirements change will be sufficient enough to create new tools including but not limited to: a new SRS document, a new Application Architecture, new FMEA reports, a new design matrix, etc… The Axiomatic Design Software tool: Acclaro will help streamline this process.

As objects of the project are elaborated, they will be pre-emptively validated with the Sponsor, allowing for the low overhead resolution of changes prior to development.

All documents will be created by with an iterative version number, allowing for a paper trail of decision. Prior documents will be maintained for this sake.

## Project Close-Out Plan

During the completion of the project, a binder will be created to house hard copies of documents to date, as well as progress reports, images, and other materials necessary to convey the pourpose, scope, and development of the project for scholastic means. Also a binder will be created with similar documentation to be kept by the company loosely created by the joint effort of the team member as a portfolio of the engeneering process, and as a tool for recreation of the project for future clients. Finally, a binder will be created with all information pertinent to the client with documentation necessary to understand, manipulate, or repair components or software at will.

# Technical Process Plans

## Process Model

The Rational Unified Process (trademarked by Rational Software) will be used as the process model for this project. Both the CMap and Gantt Chart tools used further demonstrate the commitment to this process. The white pages for the model can be found here:

<http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf>

## Methods, Tools, and Techniques

Software Tools:

Acclaro Axiomatic Design Software

- Used for Requirements Management

- Used to elaborate the Functional Requirements of the Client

- Generate FMEA reports

- Compare Design Parameters to each other with a Design Matrix Analysis

Microsoft Visio

- Create Application Architecture

- UML modeling

Basic Stamp Editor for Windows

- Creating BASIC applications for Module

- Loading application on module

- Testing and Debugging module

Microsoft Windows

- Targeted platform and development platform

Microsoft Visual Studio

- Development Environment for Software

- Testing and Debugging Software

Microsoft Gantt Chart

- Resource allocation

- Scheduling

- Progress reporting

IHMC Cmap Tools

- Concept mapping

- Website creation

Document Templates:

Software Requirements Specification

[www.processimpact.com/process\_assets/srs\_template.doc](http://www.processimpact.com/process_assets/srs_template.doc)

Project Management Plan

<http://www.projectinitiation.com/index.html>

Development Methodologies:

Rational Software – Rational Unified Process

Acclaro Software – Axiomatic Design Philosophy

## Configuration Management Plan

To be implemented.

## Quality Assurance Plan

A Quality Functional Deployment examination will be conducted using the Acclaro Software Tool. Also a FMEA will be conducted to pre-emptively discern possible risk areas that could affect the assurance of quality of the product. The various methodologies including: Axiomatic Design, Rational Unified Process, and utilizing standardized documents (IEEE 1016, SRS, Vision Document and PMP) assist in assuring quality of the product.

## Documentation Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Document | Template or Standard | Created By | Reviewed By | Target Date | Distribution |
| User Guide | To be decided | Project Team | Program Manager | May, 2012 | By hand, hardcopy |
| Project Portfolio | To be decided | Project Team | Program Manager | Dec, 2011 & May, 2012 | By hand, hardcopy |

## Process Improvement Plan

Not Applicable

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
| Matthew Rasler | 10/18/2011 | initial draft | 1.0 draft 1 |
| Matthew Rasler | 10/26/2011 | Subsequent draft to accommodate elaboration phase | 2.0 draft 1 |