PROJECT PORTFOLIO for

**EARL: Egg Alert and Real-Time Logistics**

IPFW UNIVERSITY-CS360

FALL 2011

Mark Parker Andrew Habegger Matthew Rasler

Table of Contents

\*Paging is Relative to Portfolio, and does not include Attached Documents which have associated page numbers

Project Proposal - 3

Project Research - 4

Personal Evaluation Sheets - 12

Vision Document – 16

Architectures -17

Software Requirements Specifications – 25

Quality Functional Deployment – 26

Failure Mode and Effects Analysis – 27

Progress Reports – 28

SWEBOK Knowledge Areas – 32

Project Management Portfolio – 35

Gantt Chart – 36

Progress Slides – 37

Software Design Document – 40

Individual Work – 41

Appendix – 42

Key References – 42

Verifying Assumptions Document – 42

Verifying Assumptions Flow Chart – 44

Voice of Customer – 45

CMaps – 46

Conclusion and Future Work - 51

Project Proposal

Creation of a system that mechanizes the process of determining when and where a chicken egg flow problem (egg jam) occurs on a system of conveyors through the chicken egg packaging process. Mechanical units will be installed along separate conveyors to track the flow of eggs down that specific conveyor, these units will report to a software program designed to determine if the flow is normal or abnormal. In the case of abnormal flow, the system should alert the user in real-time as to which specific line the problem has occurred on. In actual use, thousands of feet of conveyor lines would need to searched manually in the instance of a jam, this system would minimize the searching, thus minimizing the labor needed to fix the problem. Also the system eliminates unnecessary loss in performance, by alerting a user even when the packaging system is not in use.

Project Research

**Project Research Summaries**

Research Conducted By Matthew Rasler:

The client had in mind a method of hardware construction that he felt would satisfy the hardware requirements of the system to be made. He purchased the necessary components for prototyping, and handed these components off to the group for verification. These components played a large role in determining design attributes and placing constraints on software methods and language choices. In order to design to these constraints, and to insure that the underlying hardware components would, in deed appropriately interact with determined functional requirements, research was conducted on the following components with the following documents associated with those components:

1. MAX 3110E –UART (Universal Asynchronous Receiver Transmitter)
2. BS2P40 Microcontroller
3. The connections and interactions therein, including RS232 Serial Specification, RS484 Serial Specification, PBASIC (a proprietary subset of the BASIC programming language used on the microcontroller), among others.
4. MAX3110E/MAX3111E Data Sheet
   1. Also researched by Mark Parker and Andrew Habegger.
   2. Attached (1), also referenced by the following web link: <http://cmapspublic.ihmc.us/rid=1K1K85B4Y-76K6DQ-2TLS/MAX3110E-MAX3111E.pdf>
5. BS2P40 Data Sheet
   1. Also researched by Mark Parker and Andrew Habegger.
   2. Attached (2), also referenced by the following web link: <http://datasheet.octopart.com/BS2P40-Parallax-datasheet-135274.pdf>
6. PBASIC Syntax References
   1. Also researched by Mark Parker and Andrew Habegger.
   2. Wiki book for syntax guidelines: <http://en.wikibooks.org/wiki/PBASIC_Programming/Introduction#BasicStamp>
   3. Connecting BS2P40 to UART example
      1. Attached (3), also referenced by the following web link: <http://www.wd5gnr.com/suart.htm>

Research Conducted By Mark Parker:

1. Open CV *Learning OpenCV: Computer Vision with the OpenCV Library* by Bradski and Kaehler. This tome explains the OpenCV API in-depth, from interpreting camera data in real time to machine learning concepts. The book was a primary source for our original project idea, which would have been an extension of an existing computer vision system, and it has little bearing on the current project other than to make for interesting reading.
2. “Iterative Learning Control Applied to a Gantry Robot and Conveyor System,” by Freeman, Lewin, Rogers, and Ratcliffe (*Transactions of the Institute of Measurement and Control*, March 2010). Machine learning concepts applied to a conveyor system robot, which was exactly what our first project was about. Also useless for the current project.
   1. Attached (4).
3. “Talk To Multiple Devices With One UART,” by Kavaiya (*Electronic Design*, 11/15/07). Describes efficient usage of hardware components by connecting multiple UARTs to a single microcontroller and vice-versa. Includes some handy diagrams.
   1. Attached (5).
4. *Sams Teach Yourself UML In 24 Hours, Third Edition* by Schmuller. This is the guide everyone received to learn UML. It is quite informative and an excellent guide for a novice.
5. “Windows Serial Port Programming,” by Bayer ([www.robbayer.com](http://www.robbayer.com)). This paper details how to manage a serial connection using the Windows API in C/C++ along with a number of things to watch out for. Much of what I have in my Serial Communications Driver is based off of this paper.
6. “Serial Communications in Win32,” by Denver (msdn.microsoft.com). This paper was cited in the previous one by Rob Bayer. It mostly relates the same information, although it goes into further detail on a few of the functions and their usage.

Research Conducted By Andrew Habegger:

1. Parallax Basic Syntax and Reference Manual. This document provides a basic working and understanding of the syntax of the proprietary language known as PBASIC, which is required by the microcontroller being used.
   1. Web link: <http://www.parallax.com/dl/docs/prod/stamps/web-BSM-v2.2.pdf>
2. PBasic Programming by Parallax. This document provided code examples and syntactical information for PBasic.
   1. Web link: <http://a.parsons.edu/~traviss/cc/lab/propeller/PBASICProg.pdf>
3. Determining Clock Accuracy Requirements for UART Communications. To integrate the UART and BS2P40, clock pulses had to be examined. This pdf illustrates the requisites for the integration of the BS2P40 and the Maxim Uart.
   1. Web link: <http://pdfserv.maxim-ic.com/en/an/AN2141.pdf>

**Project Research Documents**

1. MAX3110E/MAX3111E Data Sheet

**Project Research Documents**

1. BS2P40 Data Sheet

**Project Research Documents**

1. Connecting BS2P40 to UART example

**Project Research Documents**

1. Iterative Learning in Conveyor Robot

**Project Research Documents**

1. Talk to multiple devices with one UART

Personal Evaluation Sheets

|  |  |
| --- | --- |
| **CS 360: Software Engineering** | |
| Project Evaluation | **Matthew Rasler** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | This is an overview of our project evaluation and my personal details. Details below. | | | | | | **Topic** | Egg Flow Communicator | | **Project Members** | Mark Parker, Matthew Rasler, Andrew Habegger | | **Project Manager** | Matthew Rasler | | **Aspects** | Motion Analysis, Object Tracking, Embedded Software, Hardware Drivers | | **Subjects** | Computer Science, Mechanical Engineering, Electrical Engineering | | **Development Time** | 12 months | | **Objective** | Fully functional, minimal error, mechanical system | | **Sponsor** | Tim Habegger, Proprietor of Habegger Poultry | | **Personal Details** | | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | *Interests* | |  |  | | --- | --- | | • | Uses cutting edge technology | | • | Potentially profitable | | • | Spans multiple Computer Science disciplines to accomplish a singular task | | • | Has tangible and measurable outcomes | | |  | *Qualifications* | |  |  | | --- | --- | | • | Experienced Software Developer, Confident in Java, C/C++, and Object Oriented Paradigm, HTML, CSS, PHP | | • | Mathematics and Physics background, with experience applying Mathematical concepts to real-world projects | | |  | *Expectations* | |  |  | | --- | --- | | • | Success collaborating multiple members and developmental streams in the development of a software based project | | • | Hone software development skills, including C based languages, and learn new languages or programs that help accelerate the development of this concept. | |   Creation of a system that mechanizes the process of determining when and where a chicken egg flow problem (egg jam) occurs on a system of conveyors through the chicken egg packaging process. Mechanical units will be installed along separate conveyors to track the flow of eggs down that specific conveyor, these units will report to a software program designed to determine if the flow is normal or abnormal. In the case of abnormal flow, the system should alert the user in real-time as to which specific line the problem has occurred on. In actual use, thousands of feet of conveyor lines would need to searched manually in the instance of a jam, this system would minimize the searching, thus minimizing the labor needed to fix the problem. Also the system eliminates unnecessary loss in performance, by alerting a user even when the packaging system is not in use.   * Pictures -- <http://ge.tt/87WlRb7?c> * Software -- <http://ge.tt/9oRwRb7?c> | | | | **Project Details** | | | **Resources** | | |  |  |  |  |  | | |

|  |  |
| --- | --- |
| **CS 360: Software Engineering** | |
| Assignment 3: Project Evaluation | **Andrew Habegger** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | This is an overview of our project evaluation and my personal details. Details below. | | | | | | **Topic** | ImpulseTM – Brain-controlled peripheral device | | **Project Members** | Mark Parker, Matthew Rasler, Andrew Habegger | | **Project Manager** | Matthew Rasler | | **Aspects** | Heuristic algorithms, data filters, hardware interface, GUI | | **Subjects** | Computer Science, Biotechnology | | **Development Time** | 12 months | | **Objective** | Fully functional peripheral device | | **Financial Investment** | $800.00 | | **Personal Details** | | |  |  |  | | --- | --- | --- | |  |  |  | |  | *Interests* | |  |  | | --- | --- | | • | Practical application of software development – industrial automation, networked communication, and biomedical research. | | • | Computer graphics – organic modeling and virtual design. | | |  | *Qualifications* | |  |  | | --- | --- | | • | Experienced software developer – C, C++, C#, Java, JavaScript, ActionScript, Visual Basic, PHP, Perl, and Python. | | • | Real-world experience – over two years of experience creating industrial applications involving optical, physical, and environmental control systems | | |  | *Expectations* | |  |  | | --- | --- | | • | Committed development of a practical application – start to finish. | | • | Innovation-driven creativity | |   Source:Rational Software White Paper  TP026B, Rev 11/01 | | | |  |  |  |  |  | | |

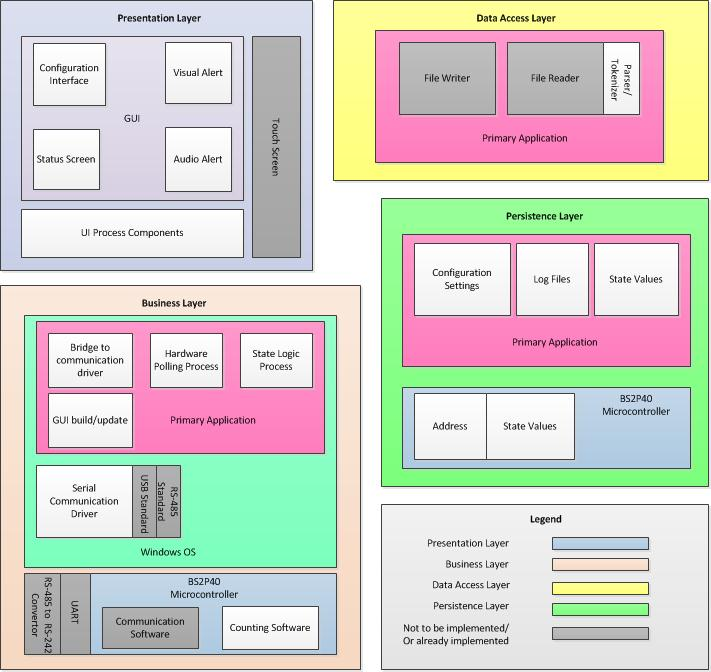
|  |  |
| --- | --- |
| **CS 360: Software Engineering** | |
| Project Evaluation | **Mark Parker** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | This is an overview of our project evaluation and my personal details. Details below. | | | | | | **Topic** | The Claw: A Robotic Egg Size Classifier | **Project Members** | Mark Parker, Matthew Rasler, Andrew Habegger | | **Project Manager** | Matthew Rasler | **Aspects** | Motion Analysis, Object Tracking, Image Recognition, Hardware Drivers, Machine Learning | | **Subjects** | Computer Science, Robotics, Graphics | **Development Time** | 12 months | | **Objective** | Fully functional, minimal error, robotic device. | **Sponsor** | Tim Habegger, Proprietor of Habegger Poultry | | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Personal Details** | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | *Interests* | |  |  | | --- | --- | | • | Novel application | | • | Solution integrates hardware and signal processing | | • | Solution piggybacks on existing technology which performs computer vision interpretation | |  |  | | | *Qualifications* | |  |  | | --- | --- | | • | Languages: Java, C/C++, VBA, HTML, Open GL, GLSL | | • | Chemical Engineering background with an emphasis on mathematical modeling | | | *Expectations* | |  |  | | --- | --- | | • | Expand knowledge of hardware technologies | | • | Learn about computer vision concepts, particularly OpenCV | | | | **Project Details** | Creation of a system that mechanizes the process of egg sorting through a video feed, video recognition, and a mechanized control arm that isolates appropriate sized chicken eggs for packaging along a conveyor belt.  Software will need to be developed that has the ability to recognize with a high degree of accuracy the size of an egg at any orientation as it moves down a conveyor, certain eggs of a size limit would need to be removed from the conveyor via suction cup mechanized control arm controlled by that software that falls into user specified size parameters.  Also via some input (possible the same video feed, or another camera with similar algorithms) should be able to monitor the flow off eggs, determining and notifying (possibly a way to correct the situation, or via notification that prompts user support) when an egg jam, or egg flow problem results. | | **Resources** | * Pictures -- <http://ge.tt/87WlRb7?c> * Software -- <http://ge.tt/9oRwRb7?c> | | |

Vision Document



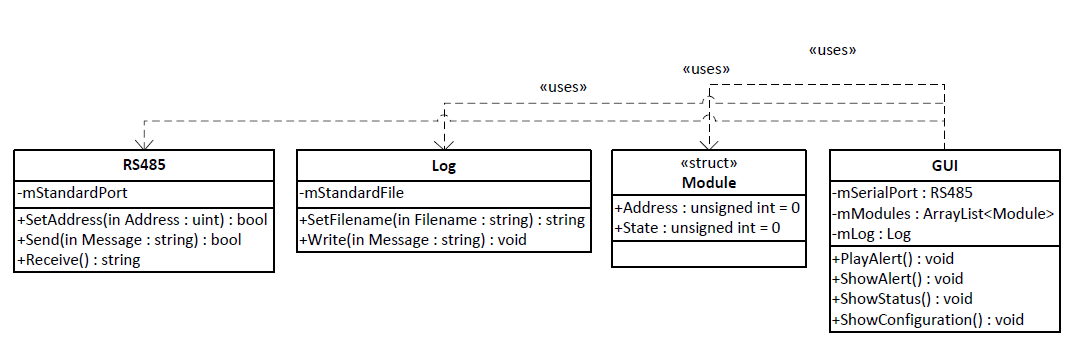
Architectures

**Application Architecture**

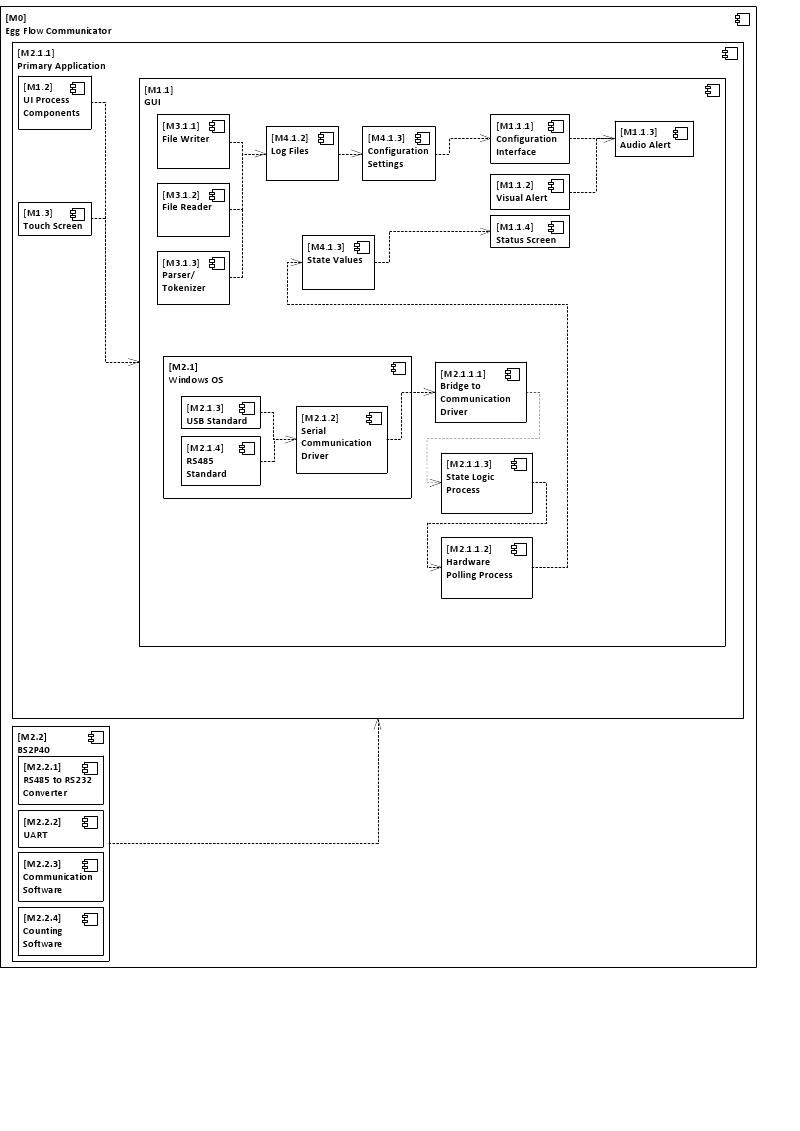
****

**Information Architecture**

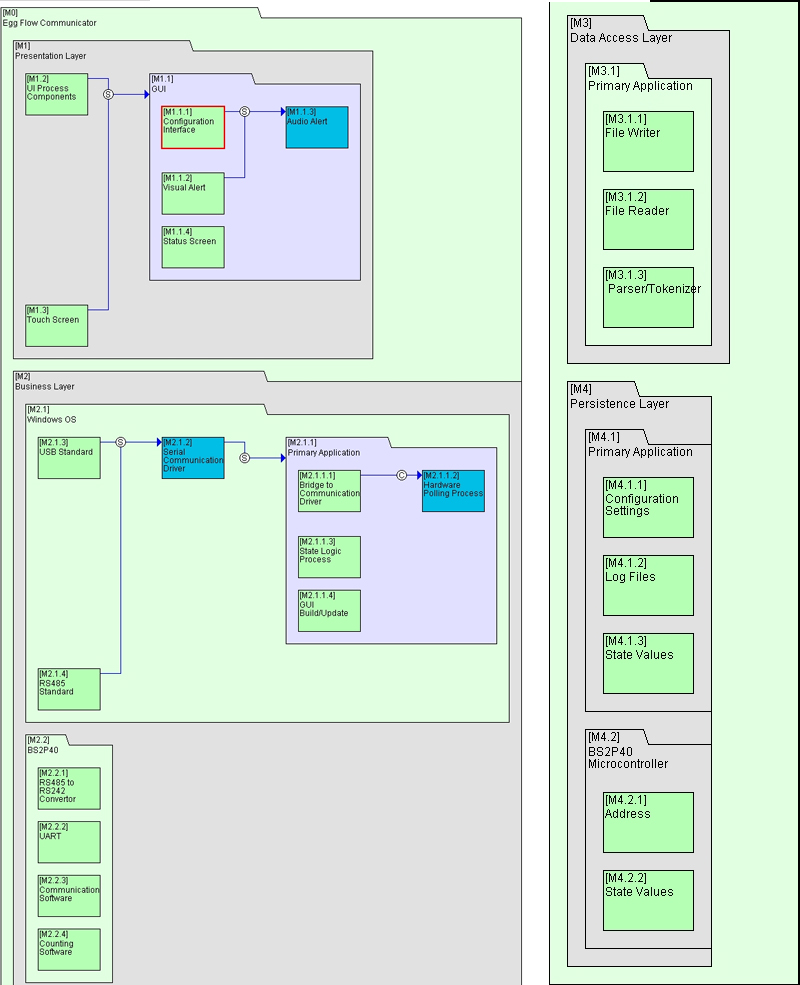
**UML Class Diagram**



**UML Component Diagram**

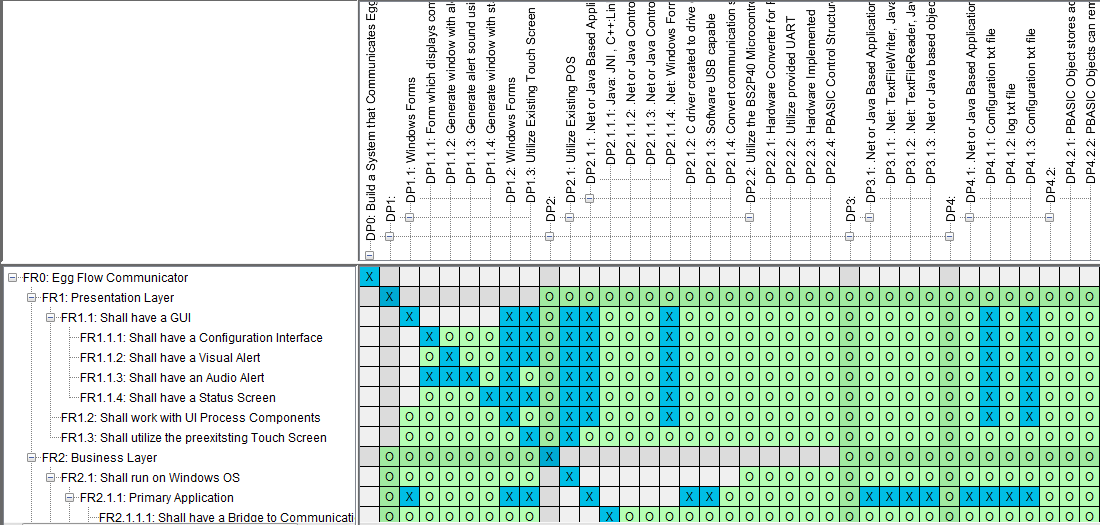


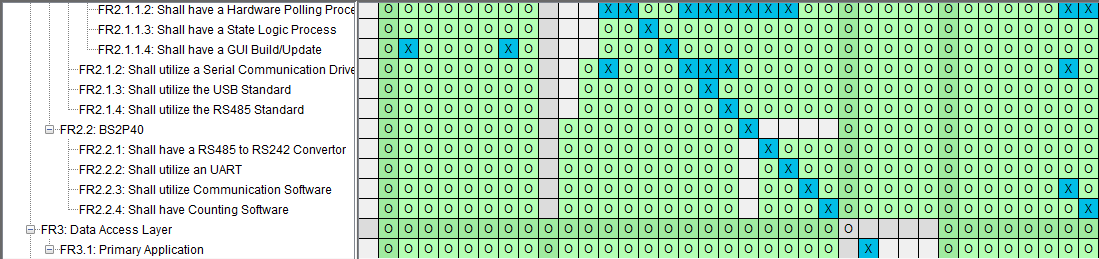
**System Architecture**

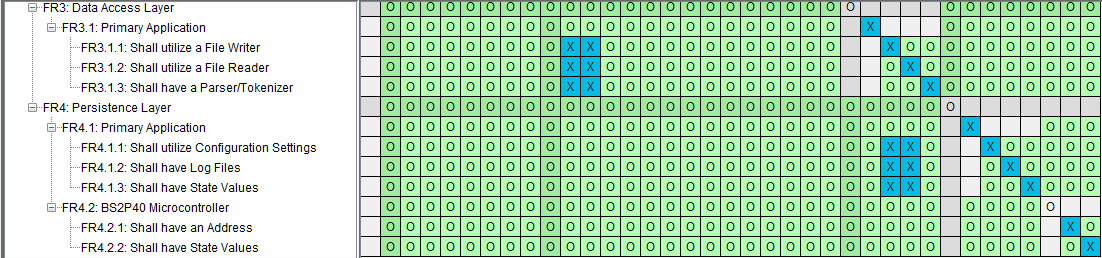
****

**Information Architecture**

**Design Matrix**

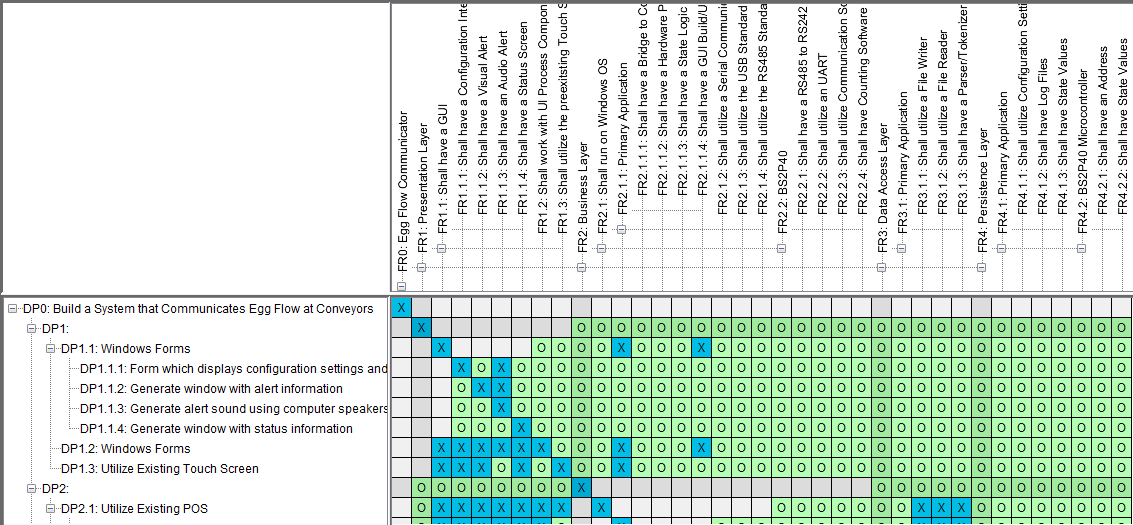


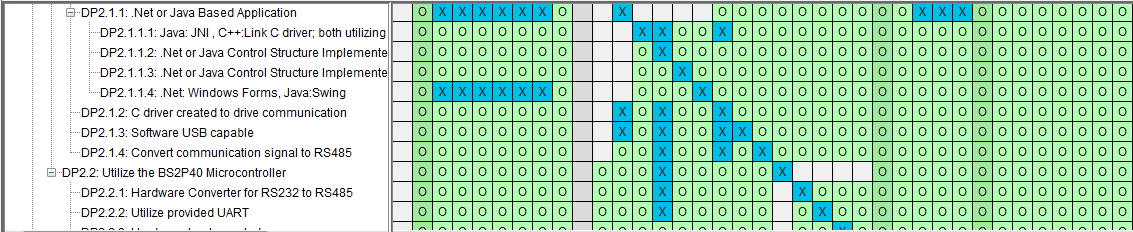


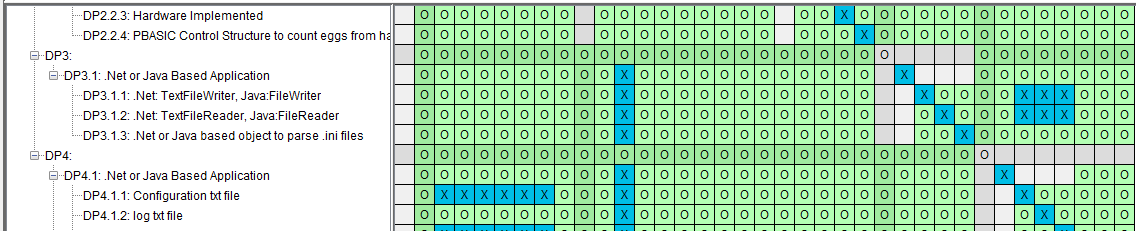


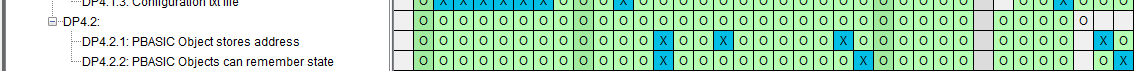
**Information Architecture**

**DSM**



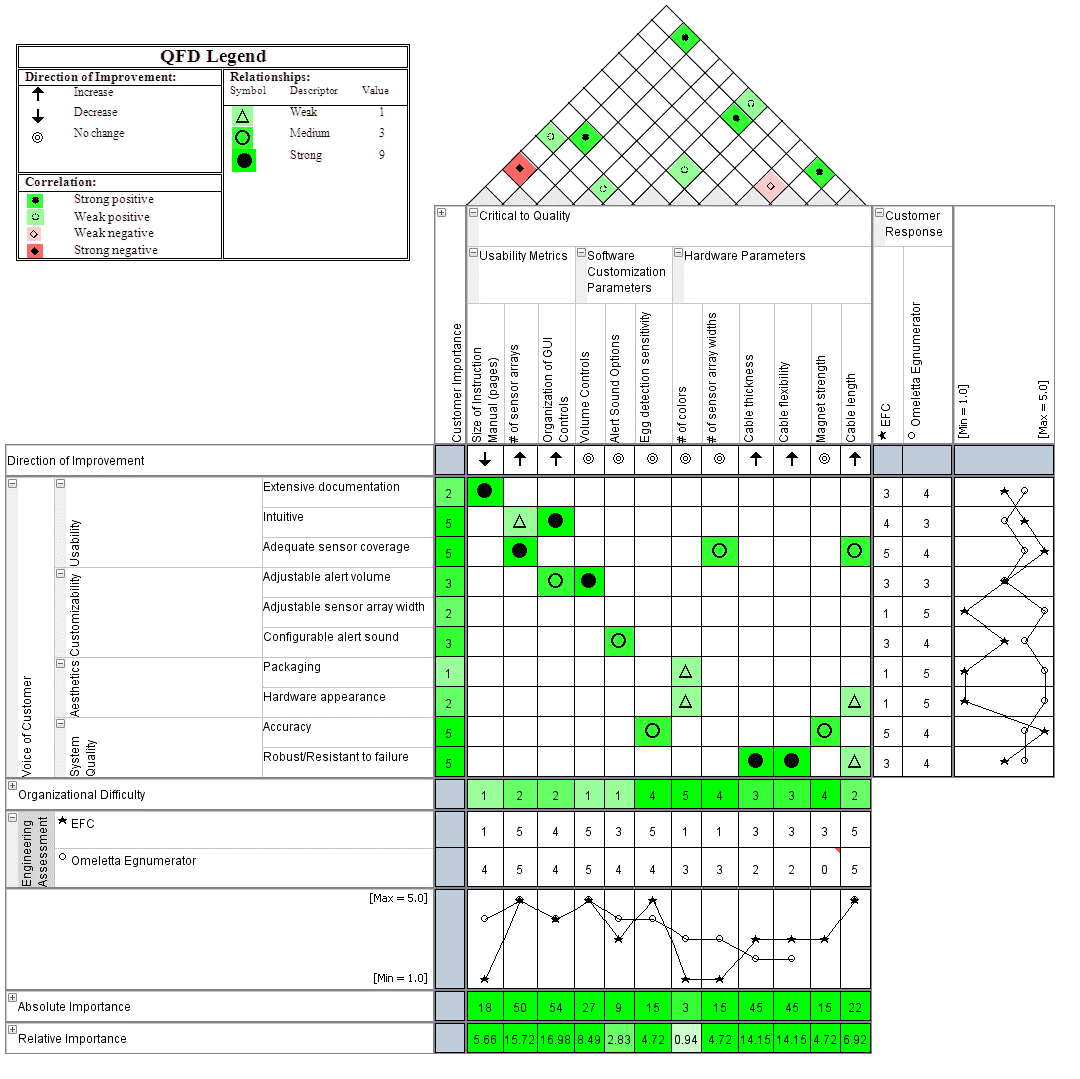




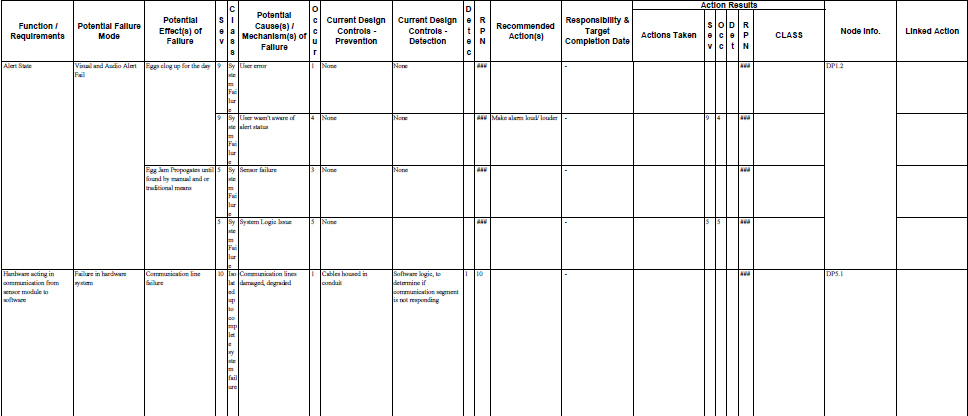
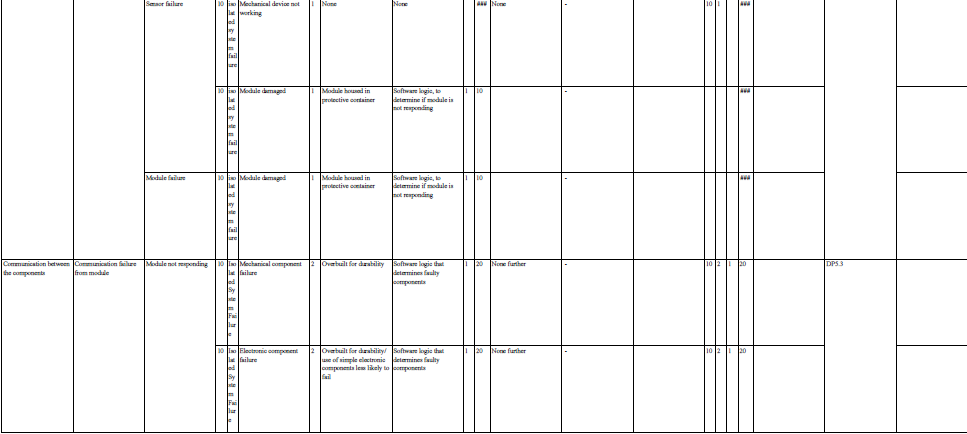


**Software Requirements Specification Document**

**Quality Functional Deployment: House of Quality**



**Failure Mode and Effects Analysis: Risk Analysis**

**Progress Reports**

|  |
| --- |
| Week 5 Sept 19-23, 2011 |

*Scheduled Weekly Meeting: Friday 11:00, ET 118*

Primary Considerations:

-Assure the following are implemented in entirety (highlight complete)-

1. Proposal & research (min 3 papers briefly reviewed and min 3 annotated links listed in MS word)  
2. Personal evaluation sheets  
3. Vision document

*Complete last section*4. Application architecture, color-coded (v1, v2)

*Add Color legend*  
5. SRS (v1) with top-level FR list from App Arch (v2)  
6. Axiomatic design tool (Acclaro download)  
a. Design Matrix - FR/DP (v1)  
b. DSM - DP/DP (v1)  
7. Progress reports (in MS Word, table format, weekly)

Determined DPs, Finished Design Matrix, Done with SRS section 1 and 2, and 3.1

|  |
| --- |
| Week 6 Sept 26-30, 2011 |

*Schedule Weekly Meeting: Friday 11:00, ET118*

To be accomplished this week:

1. App Architecture color legend added, color those which will be accomplished this semester green.
   1. At meeting
2. PMP v1 started
3. Finish SRS v2
   1. Mark sections 3.6
   2. Andrew sections 3.3, 3.4, 3.
   3. Me others
      1. By Wednesday
4. Finish Vision Last Section

Matthew

1. Risk/FMEA v1
   1. At meeting

|  |
| --- |
| Week 7 October 3-7, 2011 |

*Schedule Weekly Meeting: Friday 11:00, ET118*

To Be Accomplished This Week:

House Of Quality-Mark

|  |
| --- |
| Week 8 October 10-14, 2011 |

*Schedule Weekly Meeting: Friday 8:00, ET118*

To Be Accomplished This Week:

Version 3.0 of Application Architecture

Consequential FR-DP Matrix Update

Setup and Utilize BaseCamp-Matthew

In Meeting:

Discuss and problem solve Hardware module wiring and software requisites

|  |
| --- |
| Week 9 October 17-21, 2011 |

*~~Schedule Weekly Meeting: Friday 9:00, ET Lounge~~*

To Be Accomplished This Week:

Beta test module wiring-Mark building C driver to assist

Uncouple coupled sections of DSM and FR-DP Matrix – team

Upload TECHSPECS and Tech documentations to CMap

Update GANTT Chart

|  |
| --- |
| Week 10 October 24-28, 2011 |

*Schedule Weekly Meeting: Friday 11:00, ET118*

To Be Accomplished This Week:

House Of Quality-Mark

UML-Andrew

Update DSM with new FR-DP-Matthew

New System Architecture-Matthew

3 Slides Per person for Wednesday Presentation

Andrew-Architectures

Mark- House of Quality and Hardware

Matthew- PMP info

At Meeting:

Finish Up UML, update CMAP, Send Tanik link

|  |
| --- |
| Week 11 October 31-November 4, 2011 |

*~~Schedule Weekly Meeting: Friday 11:00, ET118~~*

To Be Accomplished This Week:

Verification of Assumptions document completed and sent to Tanik- Matthew and Mark

|  |
| --- |
| Week 12 November 7-11, 2011 |

*Schedule Weekly Meeting: Friday 11:00, ET118*

To Be Accomplished This Week:

Redo Verification of Assumption document and flow chart- Matthew

Update CMAP for reporting-Mark, Andrew, Matthew

Uplink Version 4 of FR-DP, DSM, Design Matrix, Systems Architecture

Uplink Verification Documents and send link to Tanik

Add content to Information Architecture

|  |
| --- |
| Week 13 November 14-18, 2011 |

|  |  |  |
| --- | --- | --- |
| **Name** | Role | Progress |
| Matthew Rasler | Project Manager | * 3.2 hours : Exam 1 * 3.7 hours : Exam 2 * 52 min : Reviewing COMET slideshow and homework 1 * 38 min : Reviewing CPS executive summary by U.S. Government and homework 2 * 1.2 hours : Writing this progress report, writing reminder email to group, collecting reports, attaching to CMAP, uploading new CMAP to server, emailing you that it its completed. * 1.5 hours : Uploaded content management solution for webpage |
| Andrew Habegger | (2nd) Project Manager  System Designer | * 2 hours : SRS §3.1, 3.2, 3.3 * 5 hours : Exam 1 * 3 hours : Prototyping * 1 hours : Homework 2 |
| Mark Parker | (2nd) System Designer | * 15 hours : Exam 1 * 2 hours : Design decisions about the driver * 1 hours : Homework 2 |

|  |
| --- |
| Week 14 November 21-25, 2011 |

Holiday Week, No reported work

|  |
| --- |
| Week 15 November 28-December 2, 2011 |

Matthew Rasler

SWEBOK KAs 4-11 (SWEBOK completed) 3 hours

Module Software: Learning Syntax, implementation, verification 22 hours

Module Hardware: Learning Syntax, implementation, verification 6 hours

SRS update, additional information 2 hours

Mark Parker

Revamped Vision Document 2 hours

Revamped PMP 1 hour

SWEBOK KAs 4-11 (SWEBOK completed) 4 hour

Andrew Habegger

Exam 2 5 hours

|  |
| --- |
| Week 16 December 5-9, 2011 |

Matthew Rasler

Project Portfolio Started 3 hours

Collaborated Research and Project Proposal 3 hours

Andrew Habegger

SWEBOK KAs 2 hours

Component UML 3 hours

Mark Parker

HOQ Legend 1 hour

COMET homework 3 hours

SDD subsection 1-3 2 hours

|  |
| --- |
| Week 1 7 December 12-16, 2011 |

Matthew Rasler

Project Portfolio Collaborate and Edit 10 hours

SDD subsection 6-8 3 hours

User Interface Mock and Design 2 hours

Andrew Habegger

SDD subsection 4-5 2 hours

Project Portfolio Research summation .5 hours

Project Portfolio Collaborate and Edit 10 hours

Mark Parker

Project Portfolio Collaborate and Edit 10 hours

Project Research Summation 1 hour

**SWEBOK KAs**

Matthew Rasler

**SWEBOK KAs**

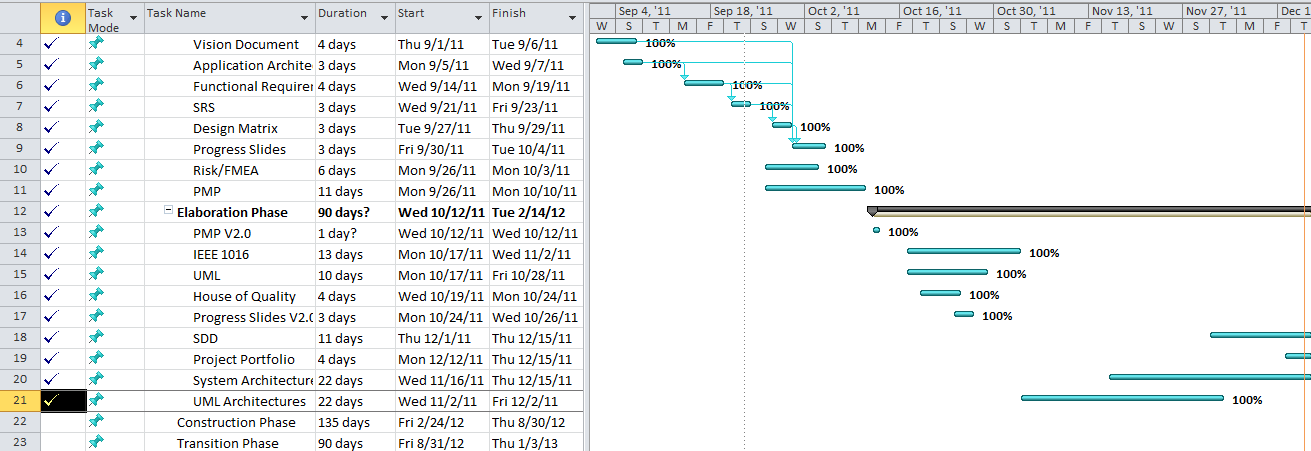
Andrew Habegger

**SWEBOK KAs**

Mark Parker

**Project Management Portfolio**

**Gantt Chart**



**Progress Slides**

9/21/2011

**Progress Slides**

11/16/2011

**Progress Slides**

12/15/2011

**Software Design Document**

**Individual Work**

**Appendix**

1. Key References
   1. Group CMAP Webpage link: <http://cmapspublic.ihmc.us/rid=1K1K884SC-X1XFD0-2TWB/RaslerSoftEngCmap.cmap>
2. Verifying Assumptions Document

In defining the elaboration phase in RUP, Rational states the following:

“While the process must always accommodate changes, the elaboration phase activities ensure that the architecture, requirements and plans are stable enough, and the risks are sufficiently mitigated, so you can predictably determine the cost and schedule for the completion of the development.”

We have identified several assumptions that must be evaluated in order to convey an assurance level of stability, anticipate scheduling and cost requisites (especially for education of staff in proprietary languages and specific communication protocols), and to assist in formulating an accurate BOM. These assumptions are as follows.

1. The terminal can communicate sufficiently to and from the UART.
2. The UART can communicate sufficiently to and from the BS2P40 Microcontroller.
3. The Microcontroller program can be created sufficient enough to accomplish the task of counting eggs.
4. The communication protocol RS232 can be converted to RS485 and back sufficiently.
5. The communication protocol RS485 can be driven sufficiently by the high-level language chosen for the primary application, housed on the existing POS (the terminal).

These have been identified as contingent of the following DP’s associated with their corresponding FR’s, which is an excerpt from the group’s FR-DP decomposition.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FR –DP Decomposition   |  |  |  |  | | --- | --- | --- | --- | | FR2.1 | Shall run on Windows OS | DP2.1 | Utilize Existing POS | | FR2.1.1 | Primary Application | DP2.1.1 | .Net or Java Based Application | | FR2.1.1.1 | Shall have a Bridge to Communication Driver | DP2.1.1.1 | Java: JNI , C++:Link C driver; both utilizing windows.h | | FR2.1.1.2 | Shall have a Hardware Polling Process | DP2.1.1.2 | .Net or Java Control Structure Implemented to Poll Modules | | FR2.1.1.3 | Shall have a State Logic Process | DP2.1.1.3 | .Net or Java Control Structure Implemented as a State Machine | | FR2.1.1.4 | Shall have a GUI Build/Update | DP2.1.1.4 | .Net: Windows Forms, Java:Swing | | FR2.1.2 | Shall utilize a Serial Communication Driver | DP2.1.2 | C driver created to drive communication | | FR2.1.3 | Shall utilize the USB Standard | DP2.1.3 | Software USB capable | | FR2.1.4 | Shall utilize the RS485 Standard | DP2.1.4 | Convert communication signal to RS485 | | FR2.2 | BS2P40 | DP2.2 | Utilize the BS2P40 Microcontroller | | FR2.2.1 | Shall have a RS485 to RS242 Convertor | DP2.2.1 | Hardware Converter for RS232 to RS485 | | FR2.2.2 | Shall utilize an UART | DP2.2.2 | Utilize provided UART | | FR2.2.3 | Shall utilize Communication Software | DP2.2.3 | Hardware Implemented | | FR2.2.4 | Shall have Counting Software | DP2.2.4 | PBASIC Control Structure to count eggs from hardware device | | FR3 | Data Access Layer | DP3 |  | | FR3.1 | Primary Application | DP3.1 | .Net or Java Based Application | | FR3.1.1 | Shall utilize a File Writer | DP3.1.1 | .Net: TextFileWriter, Java:FileWriter | | FR3.1.2 | Shall utilize a File Reader | DP3.1.2 | .Net: TextFileReader, Java:FileReader | | FR3.1.3 | Shall have a Parser/Tokenizer | DP3.1.3 | .Net or Java based object to parse .ini files | | FR4 | Persistence Layer | DP4 |  | | FR4.1 | Primary Application | DP4.1 | .Net or Java Based Application | | FR4.1.1 | Shall utilize Configuration Settings | DP4.1.1 | Configuration txt file | | FR4.1.2 | Shall have Log Files | DP4.1.2 | log txt file | | FR4.1.3 | Shall have State Values | DP4.1.3 | Configuration txt file | | FR4.2 | BS2P40 Microcontroller | DP4.2 |  | | FR4.2.1 | Shall have an Address | DP4.2.1 | PBASIC Object stores address | | FR4.2.2 | Shall have State Values | DP4.2.2 | PBASIC Objects can remember state | |

Please note that colors are arbitrarily chosen purely for aesthetics and should not be considered to have deeper meaning.

Though the following information was requested in table form, I believe that the attached flow chart will more specifically help identify how our group, as per each individual, is verifying these assumptions, and in turn identifying individual progress of individuals (as per requested).

These assumptions have been broken down further into segments in the flow chart to assist in verifying these assumptions, and which group members are tackling which segment. Note that other tasks are being accomplished in parallel, so this in no way identifies total labors of all members.

Also, note that some functional requirements, such as DP3.1.1, DP3.1.2, and DP2.1.1.1, will be implemented naturally through the developmental environment chosen, and to add clarity to the DSM and Design Matrix, these cells have been left gray to illustrate that they are not applicable to the coupling concerns of the respective matrices. These functional requirements will be met through the Design Parameters listed, and as listed in the Design Parameters they will be implemented by the structure listed included in the language chosen. This is also reflected at a high level view in the Application Architecture V3.0. It should be easily understood through the Design Parameters which structure native to the language will be utilized.

**Appendix**

1. Verifying Assumptions Flow Chart

**Appendix**

1. Voice of Customer

**EARL**

Egg Alert and Real-Time Logistics

Habegger Poultry

1357 W 100 S

Bluffton IN 46714

**Overview**

Eggs coming from 70 different lanes are brought together onto a common conveyor for transport to the packer room. Each of these lanes needs to be monitored for possible egg jams, and the number of eggs passing a given point on these lanes need to be counted. When the eggs on a particular lane jam the operator must be notified in an audible manner and the location of the jam displayed on a computer monitor. Eggs counted on each lane should be displayed in real-time on this same monitor.

**Architecture**

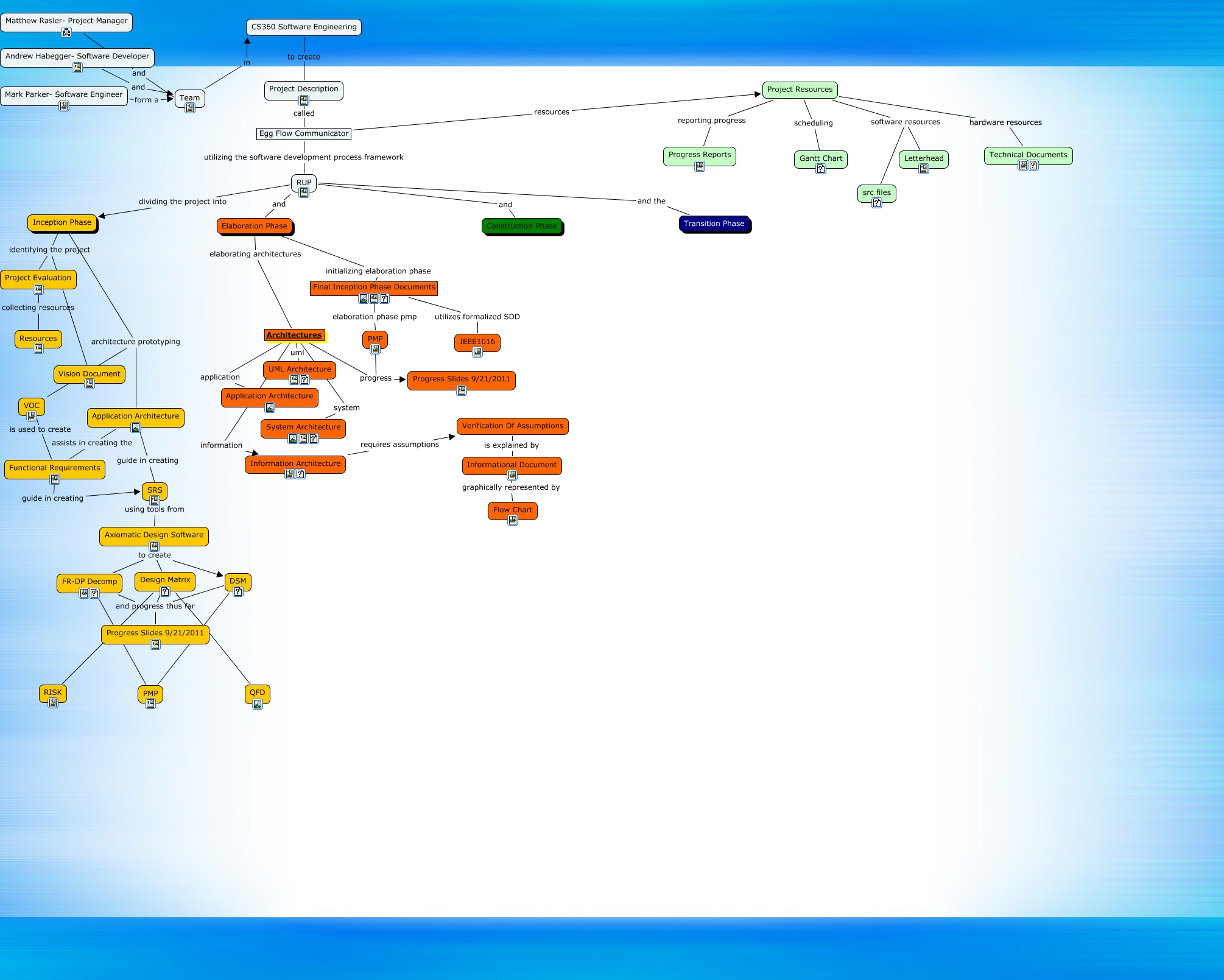
The poultry house collecting system is modular in design with 10 lanes making up a single collector. Collectors are 10 feet apart, and the group of all collectors is approximately 80 feet from the packing room where the display computer and audible alarm are located. In order the accommodate possible expansion in the future which would place additional collectors at distances up to several hundred feet a module design for EARL is requested.

1. Each collector should have it's own microprocessor to monitor 10 switch inputs. This microprocessor should seek it's identity from an attached dip switch on it's motherboard
2. Microprocessor should communicate via RS485 2 wire interface in a multi-drop configuration to the host (display) computer
3. Microprocessor configuration should include a UART to facilitate asynchronous communication to the host (display) computer.
4. In the current 70 lane configuration there should be 7 microprocessors needed to facilitate monitoring and counting the eggs.
5. Periodic queries from the host computer to each microprocessor should provide real-time counting. Further analysis of the counts should allow egg jams to be detected
6. If an egg jam is detected an audible alarm should be triggered. Current hardware configuration on the farm has the sound card of the monitoring computer wired to the stereo which can be used as the audible alarm.

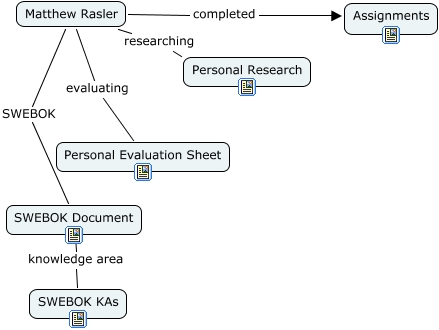
**Appendix**

1. CMaps

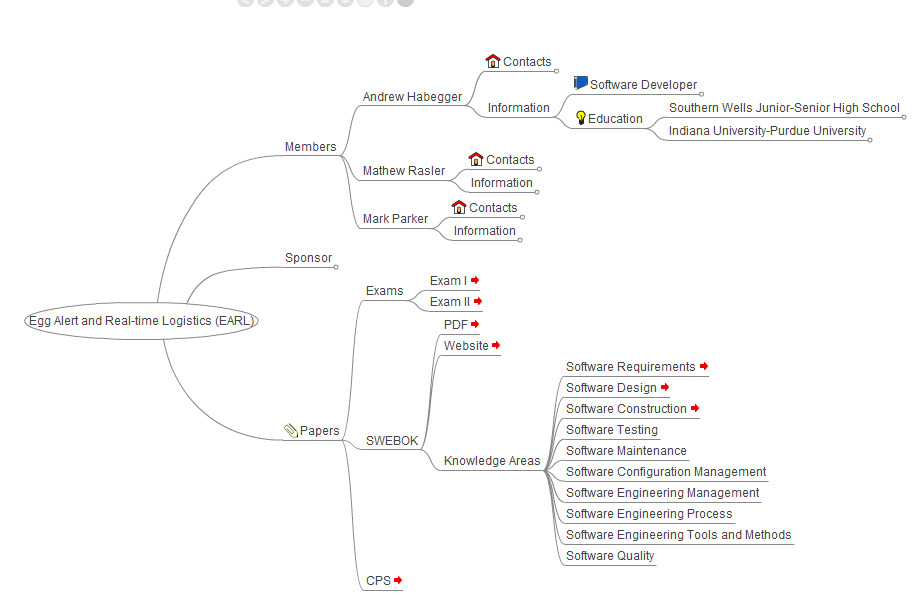
**Group CMap**



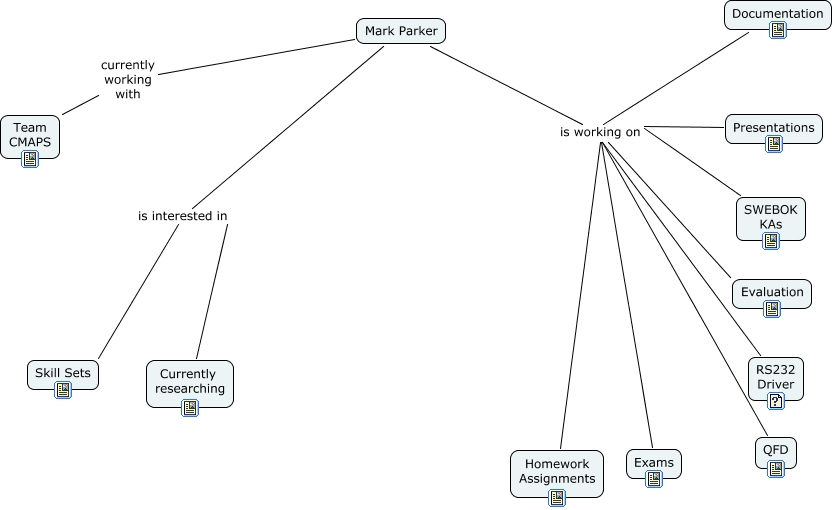
**Matthew Rasler’s CMap**



**Andrew Habegger’s CMap**



**Mark Parker’s CMap**



**Appendix**

1. Conclusion and Future Work

At the end of this semester we, as a group, are in a good position to begin the construction phase of the project. The design has stabilized, pertinent risk factors mitigated, and requirements are met in design. We plan to take our working prototype and fabricate the necessary hardware over the semester break, and with this and the coursework from this previous semester, dive headlong into next semester with the intent on finishing the project.