FA_07_SE_545: ERAU TEAM BLUE

Software Development Plan
Automatic Production Environment

Christopher Griffis
Steve Harvey
Leonardo Matos
Jason McGuire
Sean Pfeifer
Caylyne Shelton
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/5/07</td>
<td>0.1</td>
<td>Initial document template</td>
<td>Christopher Griffis</td>
</tr>
<tr>
<td>9/19/07</td>
<td>0.2</td>
<td>Risk Plan, Configuration Mgmt Plan Schedule</td>
<td>Steve Harvey, Caylyne Shelton, Sean Pfeifer, Leo Matos, Jason McGuire</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Requirements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Quality Plan</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Size Estimation Process</td>
<td></td>
</tr>
<tr>
<td>9/20/07</td>
<td>1.0</td>
<td>First Release</td>
<td>Christopher Griffis</td>
</tr>
<tr>
<td>12/11/07</td>
<td>2.0</td>
<td>Final post-project Release</td>
<td>TEAM B</td>
</tr>
</tbody>
</table>

## Document Approvals

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Sponsor</td>
<td>A. Kornecki</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Project Review Group</td>
<td>FA_07_SE_545</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Project Lead</td>
<td>Christopher Griffis</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Requirements Lead</td>
<td>Sean Pfeifer</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Hardware Lead</td>
<td>Jason McGuire</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Design Lead</td>
<td>Caylyne Shelton</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Programming Lead</td>
<td>Steve Harvey</td>
<td></td>
<td>12/11/07</td>
</tr>
<tr>
<td>Quality Manager</td>
<td>Leonardo Matos</td>
<td></td>
<td>12/11/07</td>
</tr>
</tbody>
</table>

## Preface

This Software Project Management Plan loosely follows the layout suggested by IEEE Std. 1058-1998, “IEEE STANDARD FOR SOFTWARE PROJECT MANAGEMENT PLANS.” However, certain liberties have been taken with the compliance of the format suggested in IEEE 1058-1998; the material provided herein is presented to best fit the needs of this Software Development Plan as a project communication tool and reference document.
# Table of Contents

**Title Page** ......................................................................................................................................................................ii  
**Signature Page** .....................................................................................................................................................................ii  
**Change History** .....................................................................................................................................................................ii  
**Preface** ..........................................................................................................................................................................ii  

1. **Project Summary** .........................................................................................................................................................1  
   1.1 Background ..............................................................................................................................................................1  
   1.2 Project Purpose ........................................................................................................................................................2  
   1.3 Document Purpose ......................................................................................................................................................1  
   1.4 Stakeholders ...............................................................................................................................................................1  
   1.5 Scope and Objectives ...................................................................................................................................................1  
     1.5.1 Project Scope ......................................................................................................................................................1  
     1.5.2 Project Objective .................................................................................................................................................2  
   1.6 System High-Level Requirements ...............................................................................................................................2  
     1.6.1 Purpose .................................................................................................................................................................2  
     1.6.2 Description ...........................................................................................................................................................2  
     1.6.3 Models .................................................................................................................................................................2  
     1.6.4 Requirements ......................................................................................................................................................3  
     1.6.5 Constraints ...........................................................................................................................................................4  
     1.6.6 Requirements References ...................................................................................................................................4  
   1.7 Assumptions and Constraints .........................................................................................................................................4  
   1.8 Project Deliverables ......................................................................................................................................................5  

2. **References** .................................................................................................................................................................5  

3. **Definitions** .................................................................................................................................................................5  

4. **Project Organization** ......................................................................................................................................................6  
   4.1 Overview ......................................................................................................................................................................6  
   4.2 Human Resources .........................................................................................................................................................6  
     4.2.1 Team Lead (Christopher Griffis) .......................................................................................................................6  
     4.2.2 Requirements Lead (Sean Pfeifer) .....................................................................................................................6  
     4.2.3 Design Lead (Caylyne Shelton) .........................................................................................................................6  
     4.2.4 Programming Lead (Steve Harvey) ...................................................................................................................6  
     4.2.5 Quality Lead (Leonardo Matos) ..........................................................................................................................6  
     4.2.6 Hardware Lead (Jason McGuire) ......................................................................................................................6  
   4.3 Hardware Resources ......................................................................................................................................................6  
   4.4 Software Resources ......................................................................................................................................................6  

5. **Managerial Process Plans** ............................................................................................................................................7  
   5.1 Startup Plan ...............................................................................................................................................................7  
     5.1.1 Size Estimation .....................................................................................................................................................7  
     5.1.2 Effort Estimation ..................................................................................................................................................7  
   5.2 Work Plan ....................................................................................................................................................................7  
     5.2.1 Work Breakdown Structure Summary .............................................................................................................7  
     5.2.2 Project Milestone Summary ................................................................................................................................8
5.2.3 Project Schedule ................................................................. 9
5.3 DOCUMENTATION PLAN .................................................. 9
5.4 RISK MANAGEMENT PLAN ................................................ 9
   5.4.1 Risk Management Process ............................................ 9
   5.4.2 Risk Assessment .......................................................... 9
6. TECHNICAL PROCESS PLAN ............................................... 12
   6.1 PROCESS MODEL SELECTION ....................................... 12
      6.1.1 Software Development Process Selection .................. 12
      6.1.2 Software Lifecycle Model Selection .......................... 12
      6.1.3 Life cycle selection .................................................. 13
   6.2 PROCESS MODEL ............................................................. 13
      6.2.1 Requirements Process Script .................................... 14
      6.2.2 Design Process Script .............................................. 14
      6.2.3 Coding Process Script .............................................. 14
      6.2.4 Testing Process Script .............................................. 15
   6.3 METHODS, TOOLS, AND TECHNIQUES ......................... 15
   6.4 ACCEPTANCE PLAN ....................................................... 16
7. SUPPORTING PROCESS PLANS ............................................. 16
   7.1 CONFIGURATION MANAGEMENT PLAN ......................... 16
      7.1.1 CMP Change Log .................................................... 16
      7.1.2 Scope ...................................................................... 16
      7.1.3 Organization ........................................................... 16
      7.1.4 Configuration Item List ............................................ 16
      7.1.5 Configuration Control .............................................. 17
   7.2 QUALITY ASSURANCE PLAN .......................................... 17
      7.2.1 Overview ............................................................... 17
      7.2.2 Goals ...................................................................... 18
      7.2.3 Data Collection and Analysis Methods ...................... 18
      7.2.4 Team Meetings ........................................................ 18
      7.2.5 Reporting Method .................................................... 18
      7.2.6 Problem Resolution ................................................ 18
   7.3 PROCESS IMPROVEMENT PLAN ...................................... 18
8. APPENDIX A: PROJECT POSTMORTEM ................................ 19
   8.1 TSP PROJECT SUMMARY .............................................. 19
   8.2 TEAM MEET RECORDS...................................................... 19
Table of Figures

FIGURE 1: AUTOMATIC PRODUCTION ENVIRONMENT - LOGICAL COMPONENT LAYOUT ...............3
FIGURE 2: PROJECT SCHEDULE (CALENDAR VERSION) ............................................................................9
FIGURE 4: TSP TEAM EFFORT SUMMARY ................................................................................................19
List of Tables

TABLE 1: PROJECT SCOPE: IS / IS NOT LIST ........................................................................................................2
TABLE 2: HARDWARE RESOURCES ......................................................................................................................6
TABLE 3: SOFTWARE TOOL SPECIFICATIONS....................................................................................................7
TABLE 4: RISK ASSESSMENT TABLE ................................................................................................................10
Software Development Plan
Automatic Production Environment

1. Project Summary

1.1 Background

SE545: Specification and Design of Real-time Systems is a graduate level software engineering course at Embry-Riddle Aeronautical University. As of the fall of 2007, this course requires the production of “software artifacts representing the core operational part of a selected system.”[5] The project Automatic Production Environment aims to fulfill these requirements through a well defined real-time approach to software development.

1.2 Project Purpose

The Automatic Production Environment is an industry control system based on Fischertechnik blocks, imitating operation of airport luggage conveyor belt system with a scanning machine. It will simulate the ability of a real-time software based system to transfer a payload/luggage item down a conveyer belt and “scan” it (simulating a security scan). A pneumatic/motor driven pushing device will then transfer the luggage to a second belt, which will carry the luggage to a specified final point. The system will have the ability to meet specified transition, motion, and timing requirements provided in the user needs document. Unlike typical academic software projects, this project looks at the problem from a system perspective, and will serve to emphasize real-time software principles.

1.3 Document Purpose

This document is to act as a project communication tool and reference document, outlining various aspects integral to effective quality real-time software development under time and/or budget constraints. It will establish a baseline for the various standards and procedures to be used while executing this project.

1.4 Stakeholders

- Sponsor: Dr. Andrew J. Kornecki, Embry-Riddle Aeronautical University
- Members of TEAM BLUE
  - Christopher Griffis (Project Lead)
  - Steve Harvey (Programming Lead)
  - Leo Matos (Quality Lead)
  - Jason McGuire (Hardware Lead)
  - Sean Pfeifer (Requirements Lead)
  - Caylyne Shelton (Design Lead)

1.5 Scope and Objectives

1.5.1 Project Scope

The project scope is summarized in the following Is/Is Not list:
Table 1: Project Scope: IS / IS NOT List

<table>
<thead>
<tr>
<th>This Project IS</th>
<th>This Project IS NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>This project is a small scale project consisting of 6 team members.</td>
<td>While it may address this item, this is not a project on software process.</td>
</tr>
<tr>
<td>This project is meant to fulfill the user needs within one academic (ERAU) semester.</td>
<td>While it may address this item, this is not a project on software documentation.</td>
</tr>
<tr>
<td>This project is an experience meant to serve as an academic tool.</td>
<td>While it may address this item, this is not a project on software project management.</td>
</tr>
<tr>
<td>This project is intended to emphasize the principles of real-time system development and operations.</td>
<td>While it may address this item, this is not a project on software requirements.</td>
</tr>
</tbody>
</table>

1.5.2 Project Objective
The project objective is to fulfill the user needs in a timely manner using a real-time approach to software development while adhering to the quality measures set forth in this document.

1.6 System High-Level Requirements

1.6.1 Purpose
This document is meant to overview the requirements and constraints, from the user perspective, of the SE545 Automatic Production Environment. The goal is to describe what the system shall look like and what it shall do in a clear manner, while providing information to refer back to in later phases.

1.6.2 Description
The project is to be built using the Fischertechnik system, using the available components to simulate an airport conveyor belt that will carry an item to and from a scanner. This conveyor belt consists of two belts, each driven by its respective motor. In this system there also shall be a vertically moving gadget that is meant to simulate a scanner, and then a horizontally moving gadget to push the item after it has been scanned. So, in total, there would be four motors, one per belt, and one per gadget, and then four sensors to track the position of the item as it moves through the system.

1.6.3 Models
Figure 1 depicts the system described in a more easy to understand visual manner.
1.6.4 Requirements

1.6.4.1 Functional Requirements

[SYS_FUNC1] When an item at the START PLACE is detected and the SCANNER PLACE is empty, the BELT 1 shall start moving.

[SYS_FUNC2] When the item reaches SCANNER PLACE the BELT 1 shall stop.

[SYS_FUNC3] After 2 seconds and when the item is in the SCANNER PLACE, the scanner shall engage (going down).

[SYS_FUNC4] After 10 seconds of scanning, scanner shall disengage (going up).

[SYS_FUNC5] Five seconds after scanning stops, and if the TRANSITION PLACE is empty, the PUSHER shall engage to move the item into the BELT 2.

[SYS_FUNC6] When the item is detected at the TRANSITION PLACE and there is no item at the END PLACE, the BELT 2 shall start moving.

[SYS_FUNC7] When the item reaches END PLACE, the BELT 2 shall stop.

[SYS_FUNC8] The software shall keep track and archive the processed item with their serial number and a respective time stamps.

[SYS_FUNC9] The software shall display deviations from a normal operation (blocking, sensor breakdown, unusual delay, etc.)

[SYS_FUNC10] The hardware configuration shall allow for placing four additional sensors to provide redundancy (maximum 4 motors and 8 sensors).

1.6.4.2 Nonfunctional Requirements

[SYS_NONF1] The software shall consist of a User Interface (UI) to provide user control over the system operation via keyboard/mouse/joystick/display.

[SYS_NONF2] The software shall consist of a Target Computer (TC) to provide the basic system functionality including the control algorithms and the interface components.

[SYS_NONF4] The development shall use facilities of RTLab (LB131) and Guidant Lab (LB160).
[SYS_NONF5] The primary tools used for development shall be Tornado/VxWorks (Arcom).

1.6.5 Constraints

1.6.5.1 Timing Constraints

[SYS_CNST11] Data acquired by the TC shall conform to specified timing acquisition rate.
[SYS_CNST12] Data generated by the TC shall conform to specified response time.

1.6.5.2 Development Constraints

[SYS_CNST1] The system shall be implemented on Arcom GX-1 target.
[SYS_CNST2] The developers shall keep a log to record the time spent in the project-related activities.
[SYS_CNST3] The developers shall follow guidance from a set of process scripts developed for the project.
[SYS_CNST4] The system shall be developed with the concepts of modularity, strong cohesion within a module, and light coupling between modules.
[SYS_CNST5] Defensive programming shall be used in the design and implementation of the system.
[SYS_CNST6] The system shall check for runtime errors, result reasonableness, and mathematical error.
[SYS_CNST7] The system shall use multiple tasks/processes to handle the data acquisition, processing, storage, timing, and communications (where applicable real-time programming principles will be used).
[SYS_CNST8] Use of development tools (e.g. Rhapsody in C with a code generation capability) shall be encouraged.
[SYS_CNST9] Possible use of alternate hardware interfaces with the Arcom box shall be considered (USB, keypad, display, etc.).
[SYS_CNST10] The system software shall be developed with, at the minimum, the following formal software artifacts compiled as appendices to the Project Report (a document easy to view on internet compiled at the end of the lifecycle containing overall information on the project scope, team organization, observations, effort/defect data, etc.):

- Development Plan
- Requirements Document
- Test Plan
- Design Document
- Configuration Management Document
- Source Code
- Test Results
- User's Manual

1.6.5.3 Standards Compliance

[SYS_STRD1] Manually produced code shall conform to an established C coding standard (e.g. MISRA).
[SYS_STRD2] Requirements/design shall use accepted notations (e.g. UML, structured, data/control-flow).
[SYS_STRD3] PSP/TSP effort and defects data shall be collected and included in the final report.

1.6.6 Requirements References


1.7 Assumptions and constraints

Assumptions and constraints for this project are distributed throughout this document as necessary; they are described in the sections of this document that they apply. This includes assumptions for estimation, scheduling, resource assessment, and risk assessment.
1.8 Project Deliverables

The project deliverables with tentative schedule are outlined below [5]:

- Phase I (Sep 20, 2007): Project Plan with the System Requirements/Design
- Phase II (Oct 11, 2007): Software Requirement Specification and Test Plan
- Phase III (Nov 8, 2007): Software Design Document (High Level and Detailed)
- Phase IV (Nov 29, 2007): Working Prototype Code with Initial Test Results
- Final (Dec TBD, 2007): Project Presentation and Complete Documentation

2. References


3. Definitions

CIL
A Configuration Item List (CIL) is a list of project artifacts.

CMP
Configuration Management Plan (CMP) is a project plan element delineating the procedures and resources in place for effective and consistent project structuring.

Milestone
A milestone is a scheduled event used to measure progress that has zero cost and has no time span. A milestone can serve as a useful point to establish project baselines.

Project deliverable
A project deliverable is a work product to be delivered to the acquirer.

Software project
The software project is the set of work activities, both technical and managerial, required to satisfy the terms and conditions of a project agreement.

SDP
SDP is an acronym for “software development plan”.

WBS
WBS is an acronym for “work breakdown structure”.
4. Project Organization

4.1 Overview

Human resources will consist of the team of 6 engineers working on this project. The Team Leader will be responsible for all resource allocation and tracking during this project. Due to the relatively small size of the project as well as the team of engineers working on the project, resource allocation and tracking will remain centralized through the Team Leader. Any requests for new resources will be sent to the Team Leader and he shall approve them as needed.

4.2 Human Resources

This category includes the role description and name for each engineer in the team.

4.2.1 Team Lead (Christopher Griffis)

The Team Leader is responsible for maintaining the process and the team discipline. The Team Leader leads the team and ensures that engineers report their process data and complete their work as planned. He tracks and reports the team progress. He facilitates team meetings and keeps record of meeting times and Action Items.

4.2.2 Requirements Lead (Sean Pfeifer)

The Requirements Manager primary role is to lead in the requirements phase. He should have an excellent understanding of requirements analysis and will be leading the effort to create a SRS.

4.2.3 Design Lead (Caylyne Shelton)

The Design Manager primary role is to develop a high/low level design for the development of the software product. She will be leading the effort to produce a SDS.

4.2.4 Programming Lead (Steve Harvey)

The Programming Manager is responsible for all coding of the software product. He will use the SDS as a basis for his code and will lead the effort for producing the code and code review.

4.2.5 Quality Lead (Leonardo Matos)

The Quality Manager is heavily involved in all the inspections through the development process. His responsibility is to track development problems or issues and to report to the Team Leader. The Quality Manager will prepare the test plan and coordinate the testing phases.

4.2.6 Hardware Lead (Jason McGuire)

The Hardware Manager is responsible for maintain the hardware equipment operational and up-to-date. He will perform hardware maintenance tasks when needed and will also help with identify hardware requirements throughout development.

4.3 Hardware Resources

This category includes all hardware resources like computer equipment and all other system components.

<table>
<thead>
<tr>
<th>Table 2: Hardware Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
4.4 Software Resources
This category includes all software resources like development environments, operating systems and tools.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VxWorks Target (Arcom Device)</td>
</tr>
<tr>
<td>1</td>
<td>Fischer Teknik Set of Materials</td>
</tr>
</tbody>
</table>

Table 3: Software Tool Specifications

5. Managerial Process Plans

5.1 Startup Plan

5.1.1 Size Estimation
Section Removed.
See Process Improvement Plan proposal [PROP02] in Table 5, on page 18.

5.1.2 Effort Estimation
Section Removed.
See Process Improvement Plan proposal [PROP02] in Table 5, on page 18.

5.2 Work Plan

5.2.1 Work Breakdown Structure Summary
Based on the results of the Development Plan Process in Section 6.2, summarized below is the WBS that was established and used for this Software Development Plan.

WBS-1 Project Planning
- WBS-1.1 User Needs Review
- WBS-1.2 SDP Artifacts Review
- WBS-1.3 SDP Artifacts Tasking
- WBS-1.4 SDP Documentation
- WBS-1.5 SDP Review
- WBS-M1 SDP Release Milestone

WBS-2 Requirements Development
- WBS-2.1 Plan Review
- WBS-2.2 User Needs Review
- WBS-2.3 SRS Documentation
- WBS-2.4 SRS Inspection/Review
- WBS-M2 SRS Release Milestone

WBS-3 Design Phase
- WBS-3.1 Plan Review
- WBS-3.2 Rapid Analysis
- WBS-3.3 System Architecture Description
- WBS-3.4 System Module Descriptions
WBS-3.5 Software Development Spec Documentation
WBS-3.6 Software Development Spec Review
WBS-M3 SDS Milestone

WBS-4 Coding Phase
WBS-4.1 Coding
WBS-4.2 Build and Debug
WBS-4.3 Code Documentation
WBS-M4 Code Milestone

WBS-5 Testing Phase
WBS-5.1 Test Plan Review
WBS-5.2 Functional Testing
WBS-5.3 Acceptance Testing
WBS-5.4 Testing Documentation
WBS-M5 Test Milestone

WBS-6 Project Wrap-up
WBS-6.1 Documentation Review
WBS-6.2 Presentation Preparation
WBS-6.3 Final Presentation
WBS-M6 Project Complete Milestone

5.2.2 Project Milestone Summary
This Project has the following milestones:

**WBS-M1 SDP Release Milestone**
Scheduled: 9/20/07
When the Software Development Plan is released as version 1, the SDP Release Milestone has been achieved.

**WBS-M2 SRS Release Milestone**
Scheduled: 10/8/07
When the Software Requirements Specification is released as version 1 the SRS Release Milestone has been achieved.

**WBS-M3 SDS Milestone**
Scheduled: 11/6/07
When the Software Development Specification is released as version 1, the SDS Release Milestone has been achieved.

**WBS-M4 Code Milestone**
Scheduled: 11/20/07
When the code has been baselined and documented for the first time, the Code milestone has been achieved.

**WBS-M5 Test Milestone**
Scheduled: 11/27/07
When the product has undergone the first set of testing has been completed and documented, the Test Milestone has been achieved.

**WBS-M5 Project Complete Milestone**
Scheduled: 12/02/07
When the final product has been submitted and presented and no further work is required by any member of the team, the Project complete Milestone has been achieved.
5.2.3 Project Schedule

Figure 2 shows the project schedule as represented in a calendar of events, effort and milestones.

![Figure 2: Project Schedule (Calendar Version)](image)

5.3 Documentation Plan

Our documentation plan follows a basic chain of command approach. Individuals are responsible for following best practice approaches to documentation (basic software documentation conventions, captions, outline numbering, etc.) at the discretion of the author. These efforts are subject to the approval of the responsible role, again at the discretion of the responsible authority. Any tasking of documentation should generally be assigned at least 2 days before the deadline to reduce any undue duress on the documentation contributor. All final documentation should include any items listed in the CIL (Section 7.1.4).

5.4 Risk Management Plan

5.4.1 Risk Management Process

There are two major aspects to the risk management process used in this project, risk assessment and risk control. During risk assessment, risks are first identified according to the risk identification rubric, shown in Table 4. Each risk that is identified is noted with information such as a unique ID, and a description. Risk control involves defining an avoidance strategy for each risk and a response should the risk occur. Each risk is also assigned an impact and likelihood rating, and then prioritized accordingly.

5.4.2 Risk Assessment

TEAM BLUE will use the following risk assessment table as a guideline for assessing, and avoiding risk. Outlined are various strategies for responding to a risk should it manifest as a problem.
### Table 4: Risk Assessment Table

<table>
<thead>
<tr>
<th>Risk ID</th>
<th>Risk Name</th>
<th>Risk Description</th>
<th>Avoidance Strategy</th>
<th>Response Strategy</th>
<th>Impact</th>
<th>Likelihood</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RISK-8]</td>
<td>Unforeseen Events</td>
<td>Any Risks not listed elsewhere.</td>
<td>The team has an open policy for discussing issues that occur or may appear in the future. This should allow adequate risk mitigation in advance when possible.</td>
<td>The team will meet to discuss means for mitigation. If mitigation is out of team or student control, the issue will be raised to Prof. Kornecki.</td>
<td>Med</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>[RISK-9]</td>
<td>Version Confusion</td>
<td>Document or software versioning inconsistencies cause confusion, low quality, or time loss</td>
<td>The Configuration Management Plan has mechanisms in place for version control. Following the CMP will avoid this risk.</td>
<td>The team will review the confusion, resolve the discrepancies, and review the configuration management plan and its adherence.</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>[RISK-4]</td>
<td>Loss of Work Products</td>
<td>Artifacts or draft products are lost. Any work effort that has been spent and lost.</td>
<td>Each document author will maintain personal backups of artifacts. If anyone downloads a version for editing or viewing, it should be maintained as another backup. Time buffers are built into the task schedule for any issues with document loss.</td>
<td>All users will bring their backups together at the next meeting and when necessary rework will be assigned to members in the group and project schedule will be updated to accommodate this.</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
</tr>
<tr>
<td>[RISK-5]</td>
<td>Customer Change of Needs</td>
<td>The customer adds or modifies statements in the need statement.</td>
<td>There will be open communication between the customer and the team and effort will be spent ensuring that the needs are thoroughly understood in all phases of development.</td>
<td>A group meeting will be held to discuss major changes, minor changes will be handled by each phases manager to see how it affect their development phase.</td>
<td>High</td>
<td>Low</td>
<td>Med</td>
</tr>
<tr>
<td>RISK</td>
<td>Description</td>
<td>Mitigation</td>
<td>Probability</td>
<td>Impact</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RISK-6</td>
<td>Lacking Member Contribution</td>
<td>A team member is incapable of fulfilling a task, whether in or out of his or her control.</td>
<td>All responsibilities will be distributed amongst several team members so they are not relying on one member. Clear paths of communication and planning are in place so that the team can be notified of potential issues such as travel or other distance issues.</td>
<td>Any responsibilities of the lost member will be delegated to other members of the team. If this causes an issue when we cannot meet project deadlines or requirements this will be expressed to Prof. Kornecki.</td>
<td>Med</td>
<td>Med</td>
<td>Med</td>
</tr>
<tr>
<td>RISK-1</td>
<td>Loss of Member</td>
<td>Team loses a team member for any reason. (Drop course, team swap, etc.)</td>
<td>All manager roles and actions are shared with other team members so each role is not relying on member. All collaboration is tracked in the discussion board for all to reference at any time. If any team member expressed being overwhelmed or feels far behind, they can contact the team for support before due dates and any related tasks may be considered for delegation among the rest of the team until the team member can continue to be an acting member.</td>
<td>Any responsibilities of the lost member will be delegated to other members of the team. If this causes an issue when we cannot meet project deadlines or requirements this will be expressed to Prof. Kornecki.</td>
<td>Med</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>RISK-2</td>
<td>Loss of Hardware</td>
<td>Damage or loss of any course specified hardware including but not limited to the Fischertechnik hardware, Arcom, or host machine.</td>
<td>All hardware will be kept neat and accounted for. All systems will be logged out and shut off when not in use to prevent power issues.</td>
<td>Any hardware issues that cannot be fixed by the team members will be expressed to Prof. Kornecki.</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>RISK-3</td>
<td>Loss of Blackboard</td>
<td>Blackboard goes down temporarily, indefinitely, or access by some members is restricted in any way.</td>
<td>Each document author will maintain personal backups of artifacts. All members shall keep team contact information with them personally.</td>
<td>All users will bring their backups together at the next meeting and a new method for collaboration will be discussed.</td>
<td>Med</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
6. Technical Process Plan

6.1 Process Model Selection

6.1.1 Software Development Process Selection

Selecting the lifecycle that would best fit the project required analysis of several different areas includes a thorough knowledge of the existing lifecycles and their benefits. In order to formalize how this choice was to be made, the following process was followed:

1. Fully understand the scope and size of the project.
2. Research existing lifecycles along with the advantages and disadvantages
3. Select the life cycle that best matches the project and development team’s characteristics.

6.1.2 Software Lifecycle Model Selection

Following the life cycle selection process outlined above, major software life cycles were identified along with their associated strong and weak points. Taking into account the size and scope of the project, the life cycle model selection matrix proposed the questions listed below in order to identify candidate life cycles for this project.


1. Life cycle model selection based on requirements characteristics
   a. Are the requirements easily defined and / or well known?
   b. Can the requirements be defined early in the cycle?
   c. Will the requirements change often in the cycle?
   d. Is there a need to demonstrate the requirements to achieve definition?
   e. Is a proof of concept required to demonstrate capability?
   f. Do the requirements indicate a complex system?
   g. Is early functionality a requirement?

2. Life cycle model selection based on project team characteristics
   a. Are the majority of team member new to the problem domain for the project?
   b. Are the majority of team members new to the technology domain for the product?
   c. Are the majority of team members new to the tools to be used on the project?
   d. Are the team members subject to reassignment during the lifecycle?
   e. Is there training available for the project team, if required?
   f. Is the team more comfortable with structure than flexibility?
   g. Will the project manager closely track the team’s progress?
   h. Is ease of resource allocation important?
i. Does the team accept peer reviews and inspections, management / customer reviews, and milestones?

3. Life cycle model selection based on user community characteristics
   a. Will the availability of the user representatives be restricted or limited during the life cycle?
   b. Are the user representatives new to the system definition?
   c. Are the user representatives experts in the problem domain?
   d. Do the users want to be involved in all phases of the life cycle?
   e. Does the customer want to track project progress?

4. Life cycle model selection based on project type and risk characteristics
   a. Does the project identify a new product direction for the organization?
   b. Is the project a system integration project?
   c. Is the project an enhancement to an existing system?
   d. Is the funding for the project expected to be stable throughout the life cycle?
   e. Is the product expected to have a long life in the organization?
   f. Is high reliability a must?
   g. Is the system expected to be modified, perhaps in ways not anticipated, post deployment?
   h. Is the schedule constrained?
   i. Are reusable components available?
   j. Are resources (time, money, tools, people) scarce?

6.1.3 Life cycle selection
By applying the life cycle model selection process, the Waterfall Lifecycle was chosen.

![Waterfall Model Diagram](image)

Figure 3: Waterfall Development Lifecycle Diagram [3]

6.2 Process Model
TEAM BLUE implementation of the Waterfall process will loosely follow the processes outlined in the following scripts (all process scripts are from [6]).
6.2.1 Requirements Process Script

**Requirements**

**Inputs**
- Customer needs document/interview
- Context diagram
- Detailed system description

**Customer Need Analysis**
After analysis of documents and team brainstorming, interview with the customer to find out the missing details.

**Performance Need Analysis**
Identify the performance requirements related to the real time aspects of the system.

**Functionality Exploration**
Define the functionality of the system based on the Customer’s Need Analysis. Use scenarios and use cases to ensure full coverage of the system functionality. Identify all created use cases and trace them to the requirements.

**Constraint Consideration**
Define the non-functional requirements and constraints including quality and hardware/interface aspects of the system.

**Documentation**
Document all the above findings and edit the Software Requirement Specification document including the above artifacts.

**Outputs**
- Software Description
- List of Requirements
- Test Cases
- Use Cases
- Scenarios

6.2.2 Design Process Script

**Design**

**Inputs**
- Context Diagram
- List of requirements
- Use Cases
- Scenarios

**System Architecture**
Define the structure of the system by expanding the context diagrams and separating common functionality into modules. Define flow of data and events between these modules.

**Module Design**
Expand the modules functionality and define the algorithms, states.

**Documentation**
Document Test Cases based on the algorithms and state transitions.

**Outputs**
- Design Description
- Structural Diagrams
- Dependency Diagrams
- State Chart(s)
- Sequence Diagrams
- Design Test Cases

6.2.3 Coding Process Script

**Coding**

**Inputs**
- Structural Diagrams
- Dependency Diagrams
- Statechart
- Sequence Diagrams

**Code**
Use the inputs to translate the design into a source code equivalent.
6.2.4 Testing Process Script

### Validation

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Requirement Test Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Design Test Cases</td>
</tr>
<tr>
<td></td>
<td>Object Code</td>
</tr>
<tr>
<td></td>
<td>Build Instructions</td>
</tr>
</tbody>
</table>

Use the design test case to ensure the system is built based on the design.

<table>
<thead>
<tr>
<th>Functional Testing</th>
<th>Use the requirement test cases to validate the requirements are met</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation</td>
<td>Create the validation report</td>
</tr>
</tbody>
</table>

### Verification

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Requirements process script</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>o All inputs and outputs</td>
</tr>
<tr>
<td></td>
<td>Design process script</td>
</tr>
<tr>
<td></td>
<td>o All inputs and outputs</td>
</tr>
<tr>
<td></td>
<td>Coding process script</td>
</tr>
<tr>
<td></td>
<td>o All inputs and outputs</td>
</tr>
<tr>
<td></td>
<td>Validation process script</td>
</tr>
<tr>
<td></td>
<td>o All inputs and outputs</td>
</tr>
</tbody>
</table>

Using the requirement process ensure full coverage of the system and that the right system is being built.

<table>
<thead>
<tr>
<th>Requirements Review</th>
<th>Using the design process ensure that the entire system is designed correctly and in the right context.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Review</td>
<td>Code reviews assure that coding standards are followed.</td>
</tr>
<tr>
<td>Code Review</td>
<td>Assurance that the validation process is followed.</td>
</tr>
<tr>
<td>Validation Review</td>
<td>Assurance that the validation process is followed.</td>
</tr>
</tbody>
</table>

### Outputs

- Compliance Report

### 6.3 Methods, Tools, and Techniques

Implementation of this project may use the following methods, tools and technique:

1. Implementation Language: C
2. Scheduling: Microsoft Project
3. Design and product modeling: Pragsoft UMLStudio, OSADE, MS WORD
4. Coding Guidelines: TBD
5. Integrated development Environment: Tornado
6. Requirements Management: TBD
7. Analysis Method chosen: TBD
8. SRS standard chosen: TBD
9. Test plan standard established: TBD
10. Conceptual architecture chosen: TBD
11. SDS standard chosen: TBD
6.4 Acceptance Plan

Time allocations and milestones are set in the schedule to establish baselines for the project (Section 0). The intent is that when these conditions are followed as established, then the result will lead to a timely product delivery. Following quality management outlined procedures (Section 7.2) and ensuring all items in the Configuration Item List (Section 7.1.4) are present and meet quality standards will help insure a successful customer acceptance.

7. Supporting Process Plans

7.1 Configuration Management Plan

7.1.1 CMP Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Editor</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Steve Harvey</td>
<td>Document creation.</td>
<td>9/18/07</td>
</tr>
<tr>
<td>0.2</td>
<td>Christopher Griffis</td>
<td>Formatting changes, unique CIL numbering</td>
<td>9/20/07</td>
</tr>
</tbody>
</table>

7.1.2 Scope

The purpose of this document is to identify all processes and procedures for configuration management of the SE545 TEAM BLUE Project development. The following describes configuration roles, items for control, and control processes. This document is a supplement to the Blue Project Plan.

7.1.3 Organization

This section defines organizational roles and responsibilities for configuration management.

7.1.3.1 Document Authors:

Due to the small size of this development effort, the responsibility of artifact control relies on the author of the each artifact. He or She must follow the specified procedures for base-lining and updating their product. Any changes to a document must be proposed in discussion board communication (preferred method) or verbal discussion (minor changes) and approved by the author. He or she is also required to maintain a personal copy of each artifact as a backup.

7.1.3.2 Configuration Manager:

The Quality Manager is responsible for enforcing the correct procedures for Configuration management. He or she may delegate this responsibility as needed. The Quality Manager will be the mediating authority for any change disputes with authority over author control. The team leader may act as Quality Manager if the need may arise.

7.1.4 Configuration Item List

This section identifies all artifacts to be base-lined and version controlled as changes occur. These were defined by the project needs document and supplemented in group meetings. These items are identified uniquely in a Configuration Item List.

7.1.4.1 Major Artifacts:

[CIL-1] System Development Plan  
[CIL-2] System Requirements Document  
[CIL-3] System Test Plan  
7.1.4.2 Minor Products:
[CIL-9] Member Project Discussion
[CIL-10] Un-finalized Document Collaboration
[CIL-11] Team Software Process Time Logs

7.1.5 Configuration Control
The following section defines the control processes for base-lining all products and artifacts produced in this development project.

7.1.5.1 Baseline and Change Management:
Initial document creation and any related document collaboration and discussion must take place on the Blue team group discussion board, in the subject thread that is most applicable. If an author specifies that they are the editing authority of a specific document all changes must go through that author, otherwise it is assumed that any member can change any document. Any and all versions up to final version must be identified as a "Draft" with author specified version characterization in the document name. (Non-finalized version must not use whole number version notation as this reserve to characterize finalized versioning. EX: Version 1, Version 2, etc.)

All changes between versions must be summarized in the document with applicable date and editor information.*

*Source code should not be renamed to denote draft or version number. This should be specified in posting comments/title and in the comments at the top of the source file.

7.1.5.2 Finalized Artifact Management:
When a version is to be finalized it must submitted to the Team Blue File Exchange either by the Quality Manager, Team Leader or with their approval. Approval methods are at the discrepancy of the Quality Manager or Team Leader. Any changes to finalized version must be edited, tracked in the document, and posted in the discussion board. After changes are approved as a new finalized version the artifact must be posted again on File Exchange with appropriate authority and approval. Finalized Versioning is denoted in the document name with whole number notation starting at 1. All finalized versions must have “Blue Team” or “Blue” identified in the document header and name.

7.1.5.3 Minor Product Management:
Tracking versions of minor product version is at the discrepancy of the author and should only be used when applicable. Time Logs are to be posted in the discussion board unless otherwise specified by the Quality Manager or Team Leader.

7.2 Quality Assurance Plan

7.2.1 Overview
This document and all project artifacts shall follow the standards and practices established by the appropriate IEEE Standards.
7.2.2 Goals
The following are the quality goals set forth for this project:
1. This project plan shall follow the defined standard and process for a Software Project Plan
2. This project shall deliver a product that includes all the requirements specified in the SRS document
3. All documents will be promptly updated throughout the duration of the project
4. All documents will be written precisely and clearly to prevent miscommunication between stakeholders
5. All components of this project shall contain a process which will be adhered to during production
6. Certain components of this project may contain a checklist; in particular for reviewing documents and source code

7.2.3 Data Collection and Analysis Methods
1. All data collection and analysis will be accomplished by following a modified TSP process
2. Due to the relatively small size of this project and team of engineers, data collection will be limited to development time logs
3. If the Team Leader and the Quality Manager agree that further data collection is needed, this document shall be update as such and all additional data collection shall be done using TSP processes

7.2.4 Team Meetings
All team members shall help preserve project quality by attending weekly Team Meetings. Team meetings will serve as a forum for team member accountability and posing quality issues and process issues. Team meetings may serve as soft deliverable deadlines and allow each team member to stay abreast of each others’ activities. This will help preserve quality through team discipline and communication.

7.2.5 Reporting Method
All team members shall report their development time logs to the Team Leader at least once a week before each standing meeting. This will allow the Team Leader to compile and track the data so if issues arise they can be discussed during meetings.

7.2.6 Problem Resolution
The Quality Manger will first attempt to resolve any component problems with the engineer that built the component. If the engineer feels the change or defect is not in fact needed, or if the problem does not get resolved, the Team Leader will be informed of the problem. The Team Leader will resolve any discrepancies between the Quality Manager and the other engineers.

7.3 Process Improvement Plan
The project process improvement plan will incorporate use of the form in Table 5 whenever a process improvement suggestion is made. Each week, project members will be polled for any process improvement suggestions. The process improvement suggestions will then be discussed as a team for consideration.

Table 5: Process Improvement Form: Form PIP [2]

<table>
<thead>
<tr>
<th>Company</th>
<th>FA_07_SE_545: ERAU TEAM BLUE</th>
<th>Date</th>
<th>9/20/07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Automatic Production Environment</td>
<td>Project #</td>
<td>1</td>
</tr>
<tr>
<td>Customer</td>
<td>A. Kornecki</td>
<td>Artifact</td>
<td>SDP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PIP #</th>
<th>Version</th>
<th>Problem Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PROB01]</td>
<td>1</td>
<td>No Tsp summary section added to document, no place for team notes.</td>
</tr>
</tbody>
</table>
8. Appendix A: Project Postmortem

8.1 TSP Project Summary

Figure 4 shows the TSP team effort summary. This project took 6 people 449.75 man-hours to complete, averaging 74.96 hours per person, give-or-take 25.57 hours. The team meeting notes for the project are shown in section 8.2.

![Figure 4: TSP Team Effort Summary](image)

8.2 Team Meet Records

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial team meet</td>
<td>15</td>
<td>2015</td>
<td>2025</td>
<td>20</td>
</tr>
</tbody>
</table>

# 1 Meeting Notes:

- Objectives:
  - Introductions
  - Schedule standing meet time
  - Team roles
  - Basic expectations and Blackboard usage
  - Contact info, intercommunication practices

Assumptions:
- NA this meet
Established this meeting
- Standing team meeting time proposed
  - 6pm-8pm every Monday night in RT lab
- Roles agreed upon
  - Chris – Team Lead/ indiv support
  - Steve – Hardware
  - Jason – Programmer
  - Caylyne – Designer
  - Sean – reqs and specs
  - Leo – Quality mofo

Tasking and next meet if applicable
- Next meet: Mon 10th is no meeting. Next full head count team meet mon sep 17. Interim summary meet after class Thursday if not sooner.
- Responsibility of all: First lab done during mon night meet, or as practical by at least Wednesday. Goal is to get everyone some hands on time in the lab.
- All: review slides and prepare for in-class pop quiz
- All: eyes on blackboard for further guidance/corresp

Conclude Time: 2025
SE545-0709171800-(Monday) Meet2

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>15</td>
<td>1530</td>
<td>1615</td>
<td>45 min</td>
</tr>
<tr>
<td>Discuss system behavior</td>
<td>20</td>
<td>1820</td>
<td>1830</td>
<td>10</td>
</tr>
<tr>
<td>Discuss SRS and deliverables</td>
<td>10</td>
<td>1810</td>
<td>1820</td>
<td>10</td>
</tr>
<tr>
<td>Discuss Project Plan</td>
<td>30</td>
<td>1800</td>
<td>1810</td>
<td>10</td>
</tr>
<tr>
<td>Discuss team member deliverables</td>
<td>50</td>
<td>1830</td>
<td>1835</td>
<td>5</td>
</tr>
<tr>
<td>Start next lab</td>
<td>50</td>
<td>1845</td>
<td>Indiv. basis</td>
<td>0</td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>10</td>
<td>1835</td>
<td>1845</td>
<td>10</td>
</tr>
</tbody>
</table>

Total

# 2 Meeting Notes:
Objectives:
- Come to common understanding of system behavior
- Identify any hidden requirements
- Review project calendar
- Review project risks
- Review project quality goals
- Task out SRS effort
- Task out Proj Plan effort

Assumptions/Risks/Issues/hidden reqs:
- System doesn’t check s4 before starting belt according to transition diagram v2. diagram needs correction. Will be reflected in v3
- Our system hardware as it stands only has 3 sensors and does not meet the user needs. Jason is working on solution.
- There is an issue with the sensor2 placement. Needs additional hardware or resolution for placement. Jason is working on solution, will discuss w/ Kornecki
- Need config management for versioning, naming conventions and blackboard usage, etc.

Established this meeting
- Jason and Steve swapped roles: Jason is hardware guru and Steve is coder guy.
- Discussed project schedule and deadlines.
  - Everyone understands contents of project plan
  - Leo understands and working on sdp needs for quality goals and resource descriptions
  - Steve understands and working on sdp needs for risk plan, config management
  - Finalize sdp for class on Thursday – Chris G
- Blackboard thread posting vs. file exchange conventions
  - Threads for pre-artifact and exchange for artifacts
- Everyone aware of time record log and using it
- Discussed SRS
  - Need use cases - Caylyne
  - Need high level sys reqs by Thursday – Sean
    - Put user needs into requirements “phraseology” ha! Sean, that is a real word, ms word just auto corrected it for me.
  - Need high level hardware reqs by Friday – Jason

Tasking and next meet if applicable
- Chris (lead)
  - Finalize SDP for class on Thursday (hard deadline) and post in file exchange
  - Fix error identified in state diagram (asap soft deadline)
- Jason (prog)
  - Continue efforts on hardware resolution (asap soft deadline)
- Determine some hardware requirements (Fri soft deadline)
  - Leo (qual)
    - Submit efforts on quality mgmt and resource plan (Wednesday hard deadline)
  - Steve (hware)
    - Risk table as discussed (Wed 8pm hard deadline)
    - Config mgmt plan as discussed (Wed 8pm hard deadline)
  - Sean (reqs)
    - System requirements based on user needs (Wed 8pm hard deadline)
  - Caylyne (dsrg)
    - Use cases (Thursday soft deadline)
- Next meet time:
  - Next Monday 6pm – 8pm

Conclude Time: 1945
SE545-0709241800-(Monday) Meet 3

# 3 Meeting Notes:

Objectives:
- Discuss Requirements analysis models
  - test plan, test cases
  - use cases/scenarios
  - traceability matrix
  - sequence diagrams
  - state transition diagrams (we have these already)
  - context diagram
  - class diagram
- Software requirements
  - Interface requirements
  - Data requirements
  - Derived requirements
- Discuss EV TSP stuff
- Task out and discuss deliverables and due dates.

Assumptions/Risks/Issues/hidden reqs:

Who are the actors?
Remote Conveyor Operator (web server)
Local Operator (puts the item on the conveyor belts etc)

What is "Status" that will be reported by the web server?
  - sensors blocked
  - belts running
  - State (including a unique name) - what part of the route are we on for a piece?

What are the States of Operation? May be more Logical states of the system, rather than design entities.
  - Error State - system has detected something at a sensor, and turns on belts, etc, and it never moves or gets to the next sensor.
    - (x2) A belt is turning and no sensor triggers ever after we initially detect it. Timeout value because sensor is never crossed.
    - The pusher pushes something (X many times), and it is still there
    - At this point it should call for outside assistance
  - Normal Operation State - give precedence to the later belts, and start them, then start what you were doing before.
  - Clearing State?
  - Idle State? Part of Normal Operation State?

Questions to ask customer:
  - Belt timeout?
  - 3 pushes -> error state?

Discussion:
  - Fail safe. If an item is placed in the middle of operation in a certain position (sensed by the sensor)
    Needs error state in case we can't fix something automagically.
  - Assumption: we're not going to ever have an object placed (outside interference) where there isn't a sensor.
  - Assumption: we're never going to insert an item in the middle of the system as its actively operating (belt moving/scanner scanning/pusher pushing.)
For example, if we're pushing an object from sensor 1 to the scanner, we won't put something at sensor 3.

Assumption: we're not going to check to see whether the scanner is actually "hitting" the item.

Four scenarios:
1. Package in last position
2. Package in second-to-last position
3. Package in third-to-last position
4. Package in initial position
   - At the end of the scenarios, we go to the idle state, and begin again.
   - If something is placed at sensor 1 while we're in a later scenario, we ignore it until we're done.
   - At the end of EACH scenario, we go to the SENSOR CHECK state.

Have to manually remove the item from the last sensor when fully done, then we'll be at the idle state after that.

Established this meeting
- Discussed issues with existing state transition diagram
  - Perhaps need to simplify; instead of state based transition, have a global sensor check state to monitor and makes transitions based on sensor states.
  - 4 major use cases relates to 4 positions of package
- Jason figured out a way (hi five, jason) to add a 4th sensor to the device.
- Discovered lots of requirements and questions
- Leo established method for team members submitting individual time on a regular basis

Tasking and next meet if applicable
- Chris (lead)
  - Requirements analysis models, context diagram
- Jason (hware)
  - Continue efforts on hardware
- Leo (qual)
  - Maintian team time logs
- Steve (prog)
  - Use case scenarios
- Sean (reqs)
  - State transitions updated?
- Caylyne (dsrn)
  - Updated calendar
- Next meet time: after class on Thursday (briefly)

Conclude Time: 2040
SE545-0709272005-(Thursday) Meet 4

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning (spontaneous meet)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Requirements analysis discussion</td>
<td>0</td>
<td>2005</td>
<td>2045</td>
<td>40</td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>40</strong></td>
</tr>
</tbody>
</table>

# 4 Meeting Notes:

Objectives:
- Discuss team understanding of use cases
- Discuss team understanding of context diagram
- Discuss team understanding of state transition diagram
- Perform a basic status check

Assumptions/Risks/Issues/hidden reqs:
- Don’t assume need a time out on the belt movement if not specified in user statement. Save for implementation but don’t impose it.

Established this meeting
- Restated importance of discussion board activity
- Implied deadline convention: When asked to do something on Thursday, have for Monday. When asked to do something Monday, have for Thursday (unless otherwise agreed)

Tasking and next meet if applicable
- Chris (lead)
  - SRS artifact list
  - Event response table (k-map)
  - Finite state machine table for four sensors
- Jason (hware)
  - Data flow diagram (level-1 of context diagram)
- Leo (qual)
  - Maintain time logs
- Steve (prog)
  - Use cases and state transition diagram
- Sean (reqs)
  - Use cases and state transition diagram
- Caylyne (dsgn)
  - Calendar update
- Next meet time: Monday 10/01

Conclude Time: 2045
SE545-0710011800-(Thursday) Meet 5

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Requirements analysis</td>
<td>110</td>
<td>1800</td>
<td>1930</td>
<td>90</td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>10</td>
<td>1930</td>
<td>1935</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>95</strong></td>
</tr>
</tbody>
</table>

# 5 Meeting Notes:

**Objectives:**
- Come to resolution on the state transition diagram (needs to include web server and direct control)
- Come to resolution on the use case diagram
- Discuss the current data flow diagram
- Clear up issues with blackboard under usage
- Clear up issues with time logs and project interim deliverables not being submitted in a timely manner
- Understand the web server and its role

**Assumptions/Risks/Issues/hidden reqs:**
- Belt must continue for a little bit after the sensor detects to make sure item is placed into scanner place.
- Polling processes must be able to be turned on and off by the controller process
- All requirements must be testable

Established this meeting
- Requirements Analysis Artifacts list
  - Use case diagram
  - Use case scenarios
  - Sequence diagrams
  - Context diagram
  - DFD level 1 and 2
  - P-specs for DFD
  - State transition diagram
- SRS also must have
  - Sys reqs
  - Hware reqs
  - Software reqs
  - Interface reqs

Tasking and next meet if applicable
- Chris (lead)
  - Fix state transition diagram
  - Update scenarios
  - P-specs for level-1 DFD
- Jason (hware)
  - DFD level 2
- Leo (qual)
  - Maintain time logs
- Steve (prog)
  - Go ahead web server research
- Sean (reqs)
  - Respond to use case updates
  - Create sequence diagrams
- Caylyne (dsgn)
  - Submit time logs
  - Calendar update
- Next meet time: Thurs after class 10/4
SE545-0710081800-(weekday) Meet 6

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>discuss SRS and task out for offline review</td>
<td>30</td>
<td>1800</td>
<td>1847</td>
<td>47</td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>5</td>
<td>1847</td>
<td>1907</td>
<td>20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>67</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# 6 Meeting Notes:

Objectives:
- Discuss SRS version 1 pre release
- Assign/Identify inspection roles

Assumptions/Risks/Issues/hidden reqs:
- none

Established this meeting
- Discussed requirements document posted. Reviewed for completeness and consistency.
- Everyone is aware of the doc online with user tracking. Will submit any changes by 8pm Tuesday 10/09
- Discussed roles for inspection
  - Author – chris g
  - Moderator – sean
  - Recorder – Caylyne
  - Inspector – leo
  - Inspector – Jason
  - Inspector – steve

Tasking and next meet if applicable
- Chris (lead)
  - Post any SRS comments/changes
- Jason (hware)
  - Post any SRS comments/changes
- Leo (qual)
  - Post any SRS comments/changes
- Steve (prog)
  - Post any SRS comments/changes
- Sean (reqs)
  - Post any SRS comments/changes
- Caylyne (dsdn)
  - Post any SRS comments/changes

Next meet time: 10/15 monday

Conclude Time: 1906
SE545-0710221800-(Monday) Meet 7

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>15</td>
<td>1530</td>
<td>1550</td>
<td>20</td>
</tr>
<tr>
<td>Discuss project status</td>
<td>10</td>
<td>1800</td>
<td>1810</td>
<td>10</td>
</tr>
<tr>
<td>Work individually</td>
<td>90</td>
<td>1810</td>
<td>1930</td>
<td>80</td>
</tr>
<tr>
<td>Review meet progress and task out</td>
<td>15</td>
<td>1930</td>
<td>1940</td>
<td>10</td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>5</td>
<td>1940</td>
<td>1950</td>
<td>10</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>130</strong></td>
</tr>
</tbody>
</table>

# 7 Meeting Notes:

Objectives:
- Login to team terminal and communicate with arcom device
- Determine os configuration and download setup
- Find fischertechnik manuals and figure out how to read sensors and turn on motors
- Figure out situation with webserver
- Define artifacts list for SDS

Assumptions/Risks/Issues/hidden reqs:
- None this time

Established this meeting:
- Steve explored webserver and got it running at home. Will come in Thursday and get it on arcom
- Jason and sean got a port to open between arcom and the fischer hardware controller over 232. more on Thursday
- Caylyne explored and confirmed software description document layout based on ieee 1016 cant get actualieee format.
- Discussed our requirements versus our design efforts. Thorough requirements puts us ahead for design.
- Leo was not present but has started the test plan. Will make test scripts and cluster covered requirements for the tests as suggested in lecture.
- Chris g is working on updating the SRS. Caylyne provided inspection report for author to put comments in

Tasking and next meet if applicable:
- Chris (lead)
  - Finalize SRS and assist sean and jason in demystifying rs 232 arcom-to-ROBO comm..
- Jason (hware)
  - Figure out how to at a minimum get the motors moving and sensors read
- Leo (qual)
  - Continue efforts on test plan
- Steve (prog)
  - Understand webserver and ASP use
- Sean (reqs)
  - Work with jason on getting an initial behavior out of the motors via arcom communication
- Caylyne (dsgn)
  - Come up with high level architecture for project
- Next meet time: Thursday after class, brief update meeting.

Conclude Time: 1945
# 8 Meeting Notes:

**Objectives:**
- Prepare for the SDD documentation
- Get some operation out of the APE system.

**Assumptions/Risks/Issues/hidden reqs:**
- Status reporting storage should occur on webserver.

Established this meeting
- Sat down with Steve and Caylyne to review exact details and behavior based on requirements
  - Discussion to provide an understanding of the main system processes.
  - Discussed the format and structure of the software description document
  - Developed a template for recording the process descriptions, including algorithms and pseudo code
  - Steve looked further into the webserver and the ASP aspects.
- Jason and Sean made huge leaps forward in some basic hard coded operation of system
  - Figured out how to get the motor running from arcom and got the sensors to read
  - Got a dummy test run of the conveyor belts to operate (videod and posted)
  - Able to get a reading from the sensors and have motors turn on and off based on sensor activations

Tasking and next meet if applicable
- Chris (lead)
  - Oversee preparation of SDD documentation and test plan.
- Jason (hware)
  - Continue practicing on basic operation code
- Leo (qual)
  - FINISH THE TEST PLAN
- Steve (prog)
  - Co-author the SDD
- Sean (reqs)
  - Continue practicing on basic operation code
- Caylyne (dsgn)
  - Co-author the SDD
- Next meet time: mon 11/5/07
SE545-0711121800-(Monday) Meet 10

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>15</td>
<td>1700</td>
<td>1720</td>
<td>20</td>
</tr>
<tr>
<td>Discuss SDS next steps</td>
<td>30</td>
<td>1815</td>
<td>1845</td>
<td>30</td>
</tr>
<tr>
<td>Discuss test plan</td>
<td>30</td>
<td>1845</td>
<td>1915</td>
<td>30</td>
</tr>
<tr>
<td>Discuss code strategy</td>
<td>30</td>
<td>1915</td>
<td>1945</td>
<td>30</td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>30</td>
<td>1945</td>
<td>2000</td>
<td>15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>125</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 10 Meeting Notes:

**Objectives:**
- Get a grip on total amount of time left with everyone’s holiday schedules
- Sit down and review inspection defects and response
- Review current test cases and strategy for finalizing test plan
- Orchestrate interaction between C code and webserver code.
- Discuss final steps for project after coding and road to completion
  - TIMELOGS everyone needs to catch up
  - Code quality analysis
  - Project postmortem
  - Compilation and writeup with analysis

**Assumptions/Risks/Issues/hidden reqs:**
- **none**

Established this meeting
- Caylyne and Steve will implement corrections to SDS and sign off inspection log checklist
- Steve will fill out defect log according to same template used for the SRS. He can get this from Caylyne.
- Leo and I discussed existing test cases, and reconciled our test case approach with our setup of the original use cases. This included:
  - Coverage of test cases
  - Strategy for test gaps after filling out requirements traceability matrix (checklist for non functional reqs)
  - Step by step through IEEE 829 on how we will address each heading.
  - Format of test plan and narrative content.
- Sean, Jason, and Steve had a conference synchronizing their understanding of the SRS and SDS, and how they will partition the coding responsibilities.
- For coding, the schedule for finalizing the coding in time was established.

**Tasking and next meet if applicable**
- **Chris (lead)**
  - Make sure updated SDS, project test plan, and inspection report are posted on time.
  - Provide feedback and guidance for completion of documents to quality standards
- **Jason (hardware)**
  - Code away on APE internal processes according to SDS
- **Leo (quality)**
  - Finalize Testplan. Have completed for Wednesday in time for peer review and revisions for posting on Thursday.
- **Steve (programming)**
  - Have inspection report for Chris to look at by Wednesday night, ready for posting on Thursday
  - Assist Caylyne in updating SDS
  - Code away on webserver stuff and webpage interface according to SDS
- **Sean (requirements)**
  - Code away on APE internal processes according to SDS
- **Caylyne (design)**
- Implement inspection report changes. Have final v2 SDS ready wed night for Chris to review and have ready to post on Thursday morning.
  - Next meet time: Monday Nov 19
SE545-0711261800-(Monday) Meet 11

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>15</td>
<td>1620</td>
<td>1640</td>
<td>20</td>
</tr>
<tr>
<td>Discuss project issues</td>
<td>30</td>
<td>1800</td>
<td>1830</td>
<td></td>
</tr>
<tr>
<td>Discuss coding issues</td>
<td>30</td>
<td>1830</td>
<td>1900</td>
<td></td>
</tr>
<tr>
<td>Task out and plan for next meeting</td>
<td>15</td>
<td>1900</td>
<td>1915</td>
<td></td>
</tr>
</tbody>
</table>

Total

# 11 Meeting Notes:

Objectives:

- Understand steps from here to complete, talking about deliverables
- Get status of coding
- Schedule testing of system
- Task out or understand timing of completion of the following:
  - The user manual: how to scan a package as a field user, how to operate user interface
  - The final documentation (as annex to the SDP. final doc to include postmortem, meet notes, size evaluation, mccabe complexity and discussion, and description of quality aspects.)
  - The structure, contents, and preparation of the final presentation: discussion of process, our artifacts, a presentation of product, user roles
- NEED TO FINALIZE TIME LOGS

Assumptions/Risks/Issues/hidden reqs:

- Any caveats go here

Established this meeting:

- Leo has scheduling issues with an out of area interview next week, will work out with Kornecki. Has to understands needs for wrapping up time logs and testing documentation. Test schedule has dependency issues on completion of coding. Someone else may end up doing testing, which is okay.
- Caylyne is going to monitor progress of coding and when complete will update design doc v3 to reflect discrepancies. Discussed confusion with tasking diagram in SDD v2. Caylyne will also work on the user guide.
- Sean, Jason and Steve are diligently working on the coding.
- Alert everyone to test next Thursday.
- 610 final project next Wednesday.

Tasking and next meet if applicable:

- Chris (lead)
  - Prepare the final documentation outline as described above by Monday.
  - Bring usability book for Caylyne on wednesday
- Jason (hware)
  - Continue coding for pres on Thursday
- Leo (qual)
  - Compile time logs and begin tsp documentation
  - Be ready for testing
- Steve (prog)
  - Continue coding for pres on Thursday
- Sean (reqs)
  - Continue coding for pres on Thursday
- Caylyne (dsrn)
  - Start the user guide
  - Maintain sdd for v3 final release
- Next meet time: Monday 3rd 6pm

Conclude Time: hhnn
## SE545-0712101800-(Monday) Meet 12

<table>
<thead>
<tr>
<th>Description</th>
<th>Estimate</th>
<th>Start</th>
<th>Stop</th>
<th>Delta/Interrupt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting planning</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discuss status and individual deliverables</td>
<td>30</td>
<td>1800</td>
<td>1820</td>
<td>20</td>
</tr>
<tr>
<td>Task out and plan for final pres</td>
<td>10</td>
<td>1820</td>
<td>1830</td>
<td>10</td>
</tr>
<tr>
<td>Individual work in real-time lab</td>
<td>0</td>
<td>1830</td>
<td>2000</td>
<td>90</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>180</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### # 12 Meeting Notes:

**Objectives:**
- Understand what needs to be done for project completion
- Decide who will do what by when

**Assumptions/Risks/Issues/hidden reqs:**
- Any caveats go here

Established this meeting
caylyne is working on the user manual. caylyne, you are going to talk about the SDS and the user manual for the final presentation.

steve is wrapping up efforts on webserver. he got it to work after countless hours of unrelenting persistence and experimental approaches to debugging. steve, you will have to prepare a ppt portion on how the webserver works.

sean and jason need to clean up and finalize (coding, quality aspects, commenting, and partitioning into .c files) all the code for the project. you will have to make and perform the ppt of your part. you guys can also work on our web setup. i dont really know what we're supposed to do for that. i don know we also need to compile all c code and webserver code into a single code document.

chris g is working on the standalone project summary documentation, and will make the frame story ppt. this includes our process and a discussion on each of our deliverables.

leo NEEDS everyones final timelogs ASAP so he can wrap up the tsp parts for me to include in the updated SDP. crap, i forgot i have to update that too.
i sent out an email requesting a final meet at our normal time. this last stretch is gonna be tight. sorry, i've been held completely underwater with some personal crises that have come up, as well as this gluch project, which i'll be done with as of monday AM.

**Tasking and next meet if applicable**
- Chris (lead)
  - standalone project summary documentation, and will make the frame story ppt. this includes our process and a discussion on each of our deliverables
- Jason (hware)
  - clean up and finalize (coding, quality aspects, commenting, and partitioning into .c files) all the code for the project.
- Leo (qual)
  - wrap up the tsp parts
- Steve (prog)
  - steve is wrapping up efforts on webserver.
- Sean (reqs)
  - clean up and finalize (coding, quality aspects, commenting, and partitioning into .c files) all the code for the project.
- Caylyne (dsbn)
  - caylyne is working on the user manual. caylyne, you are going to talk about the SDS and the user manual for the final presentation.

**Next meet time:** Tuesday 6pm

**Conclude Time:** 2000