HIGH POINT UNIVERSITY

Department of Computer Science

CSC 4710 Project Manual Spring, 2026

Table of Contents

Introduction

The software project is a very important part of this course not only because it provides valuable practical experience to reinforce the material covered in lectures and tutorials but also gives you a chance to experience roles and responsibilities that you will face after college graduation. This experience is designed to approximate the situation you may encounter developing software in the "real world". The project is a significant portion of this course.

This document describes in detail the way the project will be managed and assessed. Read it thoroughly so that you know exactly what is required of you.

Groups

The projects will be performed by groups of three to five students each. You have been assigned to groups by me, but the groups have the ability organize themselves. Groups will be organized during the first week of the semester.

In the context of the project, your customer will be able to advise you on the content of your submissions and will mark those submissions. You must communicate with your customer regularly to keep her up-to-date on your progress. This will be achieved via weekly check-in meetings during scheduled class times.

All students are required to participate in every aspect of the project. The group will choose a Project leader/Scrum Master. The role of a project leader should be as a coordinator, not a dictator; each group member should participate equally and take equal responsibility for the work produced by the group. However, each group member will be given a primary position within the group. These roles will determine the "group" leader for that particular function. The best way to demonstrate that you are contributing is to participate fully in group meetings with your customer and your team members. See later in this document for more information on individual assessment. A group leader may be used to make a decision where there is a difference of opinion among group members.

Class Project Team – Positions

Teams of 4 and 5

1. ***Scrum Master/Project Leader***. As Scrum Master: Responsible for communication between the team and the customer. Managing the creation of the software product. As Project Leader: Responsible for managing the Specification documents, advising the overall design, supervising implementation & testing, resource allocation and tracking. Primary responsibilities are cost analysis and control, computer and human resource acquisition and supervision. Chair project management meetings. Collect data and issues weekly cost/manpower consumption reports and a final report, set up meetings, etc

2. ***Configuration/Quality Assurance Manager***: As Configuration Manager: Responsible for change control. Primary responsibilities include tracking change requests and discrepancy reports, calling and conducting change control meetings, archiving, and preparing product releases. As Quality Assurance Manager: Responsible for developing quality standards and maintaining quality assurance. Primary responsibilities are to develop standards for documentation and implementation, calling and conducting reviews and code inspections, evaluating documents and tests.

3. ***Documentation Specialist:*** As Documentation Specialist: Responsible for the final submissions of all Documentations and maintaining standards for appearance and clarity of ALL documentation and for the creation of the Manuals. Maintaining the project minutes and the consistency throughout the project notebook.

4. ***Principal Developer/Trainer***: As Principal Developer: Primary responsibility is to manage the implementation of the individual modules of the design into a prototype and serve as the technical specialist for a particular language or operating system. As Trainer: During the design stages, the programmer will develop tools, learn the case tools and teach other team members how to use them, and experiment with new language constructs expected to be needed in the product and conduct training for other members of the team.

Documents

The software development process used for the projects will follow a model your group chooses from the lectures we have and in the book with some modifications. Each group must produce project documents at various stages of the process. In this class you will perform requirements gathering, documentation creation, implementation, verification and validation.

The documents to be produced are:

Concept Exploration Document (Project Plan): describes the project, its utility, any perceived constraints, the proposed solution strategy in general to be adopted and its feasibility.

Software Requirements Document: describes in detail exactly what the software should do (chapter 4).

Preliminary User's Manual: Paper prototype. Used to visualize the project to Client early. Major functions with sample screens.

Final Design Document: Integration of all the previous documents into a complete whole. Includes in detail how the software meets the requirements. Includes the following:

1. **Test Plan:** A set of documents that defines the acceptance criteria identified in conjunction with their client. The acceptance criteria should include function and performance criteria, start-up and shut-down sequences, system recovery, stress testing. It should also include the criteria for acceptance at the module level and the integration level.
2. **Software Performance Specification**: Defined after the acceptance criteria have been documented. Reliability, Availability, Response time, etc.
3. **Maintenance Requirements Specification**: Detailed agreements regarding maintenance of software during the operational phase

Prototype: Prototype of Design Document.

Many aspects of the documents in this project manual will be discussed throughout the course. However, you will be doing some research and using concepts learned throughout your time at HPU to complete some aspects of the project. There are also many good examples -*and many bad ones*- available on the web.

Project Assessment Breakdown

There will be 15% dedicated to check-ins with the professor. You will have weekly check-ins with me. These check-ins will be separated by groups. Each group will be with me to explain what has been done, what needs to be completed that week, and who is responsible for what part of the project that week. The group will assess if they are on track to finish in the appropriate timeframe.

Of the 35% of the course assessment allocated to the project deliverables, marks will be split among the documents in the following way:

4%	Project Plan
7%	Software Requirements Document
6%	User's Manual
6%	Specification Document
7%	Final Design Document
5%	Git Hub Repository

Of the 15% of the course assessment allocated to the walkthroughs, marks will be split between each of them. We will have between 3 and 5 walkthroughs depending on the class progress.

Of the 35% of the course assessment allocated to the final product, marks will be split between the Client Demo and Final Product.

NOTE: Reviews will be done for several of these and the grade for the documents will also include the grade for the reviews. Reviews can alter your grade by 50%.

Logic dictates that since all of the assessments for the project derive from your documents, you should attempt to produce readable, attractive documents. **Professional-Level Documents are expected.**

Git Hub Repository

The repository is storage of all information related to the project. It contains the following information:

- **List of Project Goals (created during Launch)**
- **All Deliverables (Final Versions)**
- **All Meeting Minutes – Both Team and Customer**
- **All Bi-Weekly Status Reports**
- **Weekly Work Logs**
- **Sprint Backlogs**
- **All Defect or Change Reports**
- **Issue and Risk Tracking Log**
- **The User Documentation**
- **Code as it is generated**
- **Any Other Information that is generated by the Project**

The Git Hub repository will be assessed at the end of each month – one per team. Keep the repository Up-To-Date so you will not be scrambling when it is due.

Individual Assessment

The nature of group projects is such that it is often difficult to assess the contribution made by each group member to the group submissions. This section describes extra requirements for the subject that will help us assess each individual.

Group Effort Summaries:

When each project document is submitted, each group must also hand in a single page giving an estimate of the percentage effort that was put in on that document by each member of the group. If an agreement cannot be reached within the group about how much effort was put in by each person, an individual estimate must be submitted by each person in the group. Most of you will be professionals working in the workforce within the next year. It is a measure of professional maturity to be able to objectively evaluate peers.

Individual Project Meetings:

During the last week of semester each student may be required to meet individually with me. The student should be prepared to describe the work that has been performed on the project throughout the semester and answer a few questions relating to the project. The meeting should last around 15 minutes, and your goal should be to convince me that you have a good grasp of the project and have done your part with the project.

ALL GROUP MEMBERS ARE EXPECTED TO DO AS MUCH AS EVERY OTHER TEAM MEMBER ON EACH DELIVERABLE. JUST BECAUSE ONE STUDENT IS A BETTER PROGRAMMER DOES NOT ASSUME THAT HE OR SHE IS DOING MOST OF THE PROGRAMMING. INSTEAD THAT PERSON SHOULD BE HELPING OTHER TEAM MEMBERS IMPROVE THEIR SKILLS.

Documents will be marked on a per-group basis. Neither of these latter two requirements will be directly assessed. I will, however, use them to help determine your contribution to the project and hence your individual mark. While I will assign a grade to the overall project, each individual's grade will be determined by weighting that project grade by the results of a confidential peer evaluation. Each team member will be required to assess the contributions of all members of the team with regard to the percentage contributed by each member to the successful completion of all phases of the project, and the cumulative scores for each team member will be averaged.

Your contribution will be judged on the basis of the following criteria: 1) attendance and participation in meetings with your customer and group, 2) your group and individual estimates of each individual contribution, and 3) your understanding of the project as exhibited in the individual meetings. For example, if you don't attend and participate in one or more of the meetings with your customer or group you can expect to lose marks, unless you have a good reason. Similarly, you will lose marks if you have difficulty explaining your project to me at the end of the semester.

These requirements are not meant to be hard to satisfy. They are simply designed to ensure that you get the maximum benefit out of your project work. If you participate fully in all aspects of the project and have a good understanding of it at the end of the semester, you will receive the full group mark. For individuals not contributing to the group, the group can vote to release a person. This must be done prior to the last 5 weeks of the semester.

Team Release Protocol

In order to remove a team member from a group the following warning system must be followed:

- Removal from group must be complete prior to the last five weeks of the course. The removal process takes time so this should be taken seriously.
- Group must give notice a week in advance to warn the team member of possible group removal.
- Petition to remove team member must…
    - Be completed by a majority of group members.
    - The petitions must be completed individually.
    - A template for this petition can be found in the files tab on the class Blackboard site.

Once the team member has been removed from the group, they have three options to complete the course:

1. Complete all assignments by themselves without utilizing any group work.
2. Receive an F in the course.

Group Project Grading:

The project is graded on several deliverables. Every group member will complete a peer review where they will objectively quantify each team members participation for the deliverable. You will also assess your participation regarding the deliverable. The professor of the course reserves the right to adjust a team member's grade according to the feedback received from the other team members.

Appendix 1: Deliverables

A1.1: Project Plan

Contents

- Project name
- Team members
- Sponsor/customer
- Roles
- Work breakdown
- Introduction - Complete Description of Problem defined by customer
- Project Goals
- Project Scope
- Hardware and software requirements
- Schedule – Use MS Projects
- Delivery Plan
    - Deliverables to be submitted
- Communication and Reporting
- Project Constraints
- Risk Assessment and Management

A1.2: Requirements Document Format

Contents

- Project name
- Introduction: Project description and goals
- Customer
- Users/Audience
- Functional requirements
- Non-functional requirements
- Operation Scenarios
- Goal hierarchy
- Constraints
- High-Level System model -- CASE tool
- High-Level Cost Estimates
- Glossary
- Appendices (interview notes, market surveys, etc.)

A1.3: Software Specification Document

1. Introduction
    a. Paragraph on Project Description
    b. Purpose of this document
    c. Any changes to previous plan
2. Design Consideration
    a. Assumptions and Dependencies
    b. Goals and Guidelines
    c. Development Methods
3. Architectural Design (use CASE Tools)
    a. Data and Control Flow
    b. Derived Program Structure
    c. Use Cases
4. Interface Design (CASE Tools)
    a. Human Interface Specification
    b. Human Interface Design Rules
    c. External Interface Design
    d. Use Cases
5. Procedural Design (for each module)
    a. Processing Narrative
    b. Interface Description
    c. Design Language (or other) Description
    d. Use cases
6. Updated MS Project Reports - Schedule and Costs
7. Project Meeting Notes

A1.4: Final Design Document

The final design documentation for your project should have the following parts:

- Scope: Overview and Rationale - a textual description of your design approach including the reasons why you decided to design your solution the way you did. In particular, you should explicitly indicate how your design addresses your project's non-functional requirements.

- **Reference Documents: -- inclusion of** Existing software documentation, Vendor (hardware and software) documents (Licenses, contracts, etc.), System Documentation (all code), Technical references

- Architecture - a diagram describing the overall structure of your solution. Boxes in the diagram correspond to *components* or modules in your solution. This may include computational components or data components, which you should distinguish visually. Lines correspond to *connectors*. Different kinds of connectors should be denoted with different styles of lines. Examples of connectors include function/method calls, pipes, inclusion dependencies, data flows, etc. Please provide a legend and explanatory text describing the icons used in your diagram.

- Module signatures: for each function/method in each computational component, provide a signature. These may be web components which will still have the same information below. Recall that a signature provides the following information:
  - Name - the name of the function/method
  - Description - what role does it play
  - Whether it is a function, procedure, or process
  - If it is a function, its return type
  - For each parameter:
    - Name
    - Purpose - what is its role in the function/method
    - Data type
    - Mode (**input**, **output**, **in/out** (both))
  - Preconditions: what values of the **input** and **in/out** parameters are valid
  - Exceptions: what happens with invalid input (e.g. error messages, return codes, raised exceptions)
  - Post conditions/effect: after the subprogram or web module has completed, in what ways can the rest of the system know it executed: (e. g. how are the return value, any **out,** and **in/out** parameters, and any global data structures altered as a result of executing the function/method, database modified)
- For a data component, provide the following information.
  - Name
  - Purpose
  - Type
- For each connection:
  - Name
  - Purpose

- Type of data it communicates
  - Processing constraints/synchronization
  - Protocol - any transmission rules such as *first-in-first-out*
  - Other filtering or processing it does

- Major algorithms - provide a description and pseudocode
- OMT models
  - Updated static object model
  - A state chart for each object with non-trivial state
  - Data flow diagrams updated with method names
  - Event trace diagrams for use cases
- **File Structure and Global Data:** External File Structure, Logical Structure, Logical Record Description, Access Method, Global Data, File and Data Cross-Reference (Matrix)
- Packaging: Transfer Consideration, Training and Support

A2.1: Weekly Labor/Expense

Log for Week Ending ______

Name ____________________
Team ____________________
Position ____________________

Conversion Table

:01-:06	.1
:07-:12	.2
:13-:18	.3
:19-:24	.4
:25-:30	.5
:31-:36	.6
:37-:42	.7
:43-:48	.8
:49-:54	.9
:55-1:00	1.0

Computer Usage:

Total computer time: ______________**hours**
Wordprocessing: ____________**hours**
Application Time: ____________**hours**
Software Tools: ____________**hours**
Other (Specify): _____________**hours**

Hours Worked (Summary)

	Correspondence	Work on Project	Training/Education	Total
Saturday				
Sunday				
Monday				
Tuesday				
Wednesday				
Thursday				
Friday				

A2.2: Status Report

Software Engineering Project
Team Name: ____________________

High Point University • CSC 4710 • SPRING, 2026

Status Report (to be submitted every two weeks)

Dates: XXX to XXX

Team Members:	**UserID**	**Phone**

(examples of what might be included)

Date
- Name and contact information of five team members formed. and forwarded
- Name of Project Team -XXX.
- Major requirements of project and software were discussed.
- Began brainstorming for project.

Date
- Team Meeting #X.
- Customer has been contacted. Meeting with customer scheduled for 09/06/03
- First draft Project Plan agreed on.
- The customer's problem has been researched
- Questions regarding customer have been selected
- Agree to the workload for Plan
- Start working on Project Plan

Date
-
-
-
-

-

Date

Date

Date

Key issues to be resolved:
General Comments:

A2.3: Discrepancy Report

DISCREPANCY REPORT Fill in bold areas. Report NO:

Originator	Name:	Date Prepared:	Date Needed:
	Position:	Mail Address:	Version:
	Group:	Module/Unit/System Affected:	
Type of Problem	**Design** _____ **Deviation** ______ Compatibility _____ Other ______		**Priority** Emergency _____ Urgent _____ Routine _____
Description			
Analysis			
Correction			
Documentation			Resources: Programmer Hours: _____ Computer Expenses: _____ Other: _____ Total Cost: _____
Accepted for Investigation: YES NO	Signature	Title	Date:
Final Decision	Approved ☐ ☐ Rejected Action Taken or Reason for Rejection		QA: Date
			Monitor: Date
			V & V: Date
			Implementor: Date

A2.4: Request for Change

REQUEST FOR CHANGE Fill in. Request No:

ORIGINATOR	Name:	Date Prepared:	Date Needed:
	Position:	Mail Address:	Version:
	Group	Module/Unit/System Affected:	
TYPE OF CHANGE	Design _____ Compatibility _____ Cost Reduction_____	Deviation ____ New Feature ____ Other ____	Priority: Emergency _____ Urgent _____ Routine _____
DESCTRIPTION OF PROBLEM			
PROPOSED SOLUTION			

Accepted for Investigation ☐	Signature	Title	Date

FINAL DECISION	Approved ☐ Rejected ☐ Action Taken or Reason for Rejection	DOCUMENT	Description of changes:
		RESOURCE	Programmer Hours ______ Computer Expenses: ______ Other: ______ TOTAL COST: ________
		QA:	Date
		Monitor:	Date
		V & V	Date
		Implementer	Date

A2.5: Issue and Risk Tracking Log (IRTL form)

Team Number: ___________________________

IRID	Description and Comments	Status
1		
2		
3		
4		
5		
6		

A2.6: Team Meeting Minutes Template

Team Meeting Minutes (MINUTES form)

Team Number:

Date and Time:

MIDs of Team Members Present:

General Notes:
Decisions Made:
IRIDs of Issues and Risks Raised:

Appendix 3: Assessment

A3.1: Peer Evaluation

Peer evaluation is a common practice in the engineering community. Critical evaluation is a necessary component of improving the software engineering profession. It is possible to achieve higher and lower individual project scores that the team averages from the graded assignments. Each of you will submit a Peer Evaluation for each of the team members FOR EACH DELIVERABLE, and a final evaluation at the end of the semester. These will be submitted in the Drop Box confidentially.

Peer Evaluation Form

Your name____ Date _____________________

Deliverable_______________________________ Team ________________________________

Using the scale below, circle the number that best corresponds to how well each team member met the following four criteria. ***(Rate your performance first)*** Disagree Agree --- Strongly Agree;

Team Member's Name _______________________________

1. **Member effectively performed her/his role functions** **1 2 3 4 5**
(e.g., did more than his/her share of work and writing, attended all team meetings, wrote his/her sections of the report).
2. **Member made useful contributions to the team project** **1 2 3 4 5**
(e.g., worked on the hardest tasks, exhibited competence, learned new tasks and explained them to others).
3. **Member contributed to efficient team procedures** **1 2 3 4 5**
(e.g., managed his function as a project leader, formulated goals, provided leadership, gave ideas to the general effort, confronted issues, identified action items).
4. **Member had a constructive, open-minded and cooperative attitude** **1 2 3 4 5**
(e.g., courteous, friendly, enthusiastic, worked to reach consensus, good humored, followed team ground rules, encouraged feedback on self, drew others out).
5**. Member met deadlines identified by the team** **1 2 3 4 5**
6. **Member provided Quality and Quantity of work in preparing the final product.** **1 2 3 4 5**

TOTAL points for this team member ______________________

Comments:____

* * * * *

Team Member's Name _______________________________

1. **Member effectively performed her/his role functions** **1 2 3 4 5**
(e.g., did more than his/her share of work and writing, attended all team meetings, wrote his/her sections of the report).
2. **Member made useful contributions to the team project** **1 2 3 4 5**
(e.g., worked on the hardest tasks, exhibited competence, learned new tasks and explained them to others).
3. **Member contributed to efficient team procedures** **1 2 3 4 5**
(e.g., managed his function as a project leader, formulated goals, provided leadership, gave ideas to the general effort, confronted issues, identified action items).
4. **Member had a constructive, open-minded and cooperative attitude** **1 2 3 4 5**
(e.g., courteous, friendly, enthusiastic, worked to reach consensus, good humored, followed team ground rules, encouraged feedback on self, drew others out).
5**. Member met deadlines identified by the team** **1 2 3 4 5**
6. **Member provided Quality and Quantity of work in preparing the final product.** **1 2 3 4 5**

TOTAL points for this team member ______________________

Comments:____

* * * * *

Team Member's Name ______________________________

1. **Member effectively performed her/his role functions** **1 2 3 4 5**
(e.g., did more than his/her share of work and writing, attended all team meetings, wrote his/her sections of the report).
2. **Member made useful contributions to the team project** **1 2 3 4 5**
(e.g., worked on the hardest tasks, exhibited competence, learned new tasks and explained them to others).
3. **Member contributed to efficient team procedures** **1 2 3 4 5**
(e.g., managed his function as a project leader, formulated goals, provided leadership, gave ideas to the general effort, confronted issues, identified action items).
4. **Member had a constructive, open-minded and cooperative attitude** **1 2 3 4 5**
(e.g., courteous, friendly, enthusiastic, worked to reach consensus, good humored, followed team ground rules, encouraged feedback on self, drew others out).
5**. Member met deadlines identified by the team** **1 2 3 4 5**
6. **Member provided Quality and Quantity of work in preparing the final product.** **1 2 3 4 5**

TOTAL points for this team member ______________________

Comments:____

____

____

* * * * *

Team Member's Name ______________________________

1. **Member effectively performed her/his role functions** **1 2 3 4 5**
(e.g., did more than his/her share of work and writing, attended all team meetings, wrote his/her sections of the report).
2. **Member made useful contributions to the team project** **1 2 3 4 5**
(e.g., worked on the hardest tasks, exhibited competence, learned new tasks and explained them to others).
3. **Member contributed to efficient team procedures** **1 2 3 4 5**
(e.g., managed his function as a project leader, formulated goals, provided leadership, gave ideas to the general effort, confronted issues, identified action items).
4. **Member had a constructive, open-minded and cooperative attitude** **1 2 3 4 5**
(e.g., courteous, friendly, enthusiastic, worked to reach consensus, good humored, followed team ground rules, encouraged feedback on self, drew others out).
5**. Member met deadlines identified by the team** **1 2 3 4 5**
6. **Member provided Quality and Quantity of work in preparing the final product.** **1 2 3 4 5**

TOTAL points for this team member ______________________

Comments:____

____

____

* * * * *

Team Member's Name ______________________________

1. **Member effectively performed her/his role functions** **1 2 3 4 5**
(e.g., did more than his/her share of work and writing, attended all team meetings, wrote his/her sections of the report).
2. **Member made useful contributions to the team project** **1 2 3 4 5**
(e.g., worked on the hardest tasks, exhibited competence, learned new tasks and explained them to others).
3. **Member contributed to efficient team procedures** **1 2 3 4 5**
(e.g., managed his function as a project leader, formulated goals, provided leadership, gave ideas to the general effort, confronted issues, identified action items).
4. **Member had a constructive, open-minded and cooperative attitude** **1 2 3 4 5**
(e.g., courteous, friendly, enthusiastic, worked to reach consensus, good humored, followed team ground rules, encouraged feedback on self, drew others out).
5**. Member met deadlines identified by the team** **1 2 3 4 5**
6. **Member provided Quality and Quantity of work in preparing the final product.** **1 2 3 4 5**

TOTAL points for this team member ______________________

Comments:____

____

____

* * * * *

A3.2: Design Reviews and Walkthroughs

Why we conduct project reviews:

- Find errors early.
- Improve quality.
- Increase productivity.
- Shorten schedule.
- Mitigate risk.
- Supplement testing.
- Avoid preventable defects.

Checklist

1. Traceability: Is each piece of the software traceable to a specific requirement specification?

2. Risk: Are there issues here?

3. Practical: Is it an appropriate solution?

4. Maintainability: Once developed, can it be easily maintained?

5. Quality: How is this being addressed?

6. Interfaces: Criteria for this

7. Technical clarity: Can design be easily translated into code?

8. Alternatives: What if something isn't workable? What are alternative plans?

9. Limitations: Defined and livable?

A3.3: Review Check sheets

1. **Project Planning Checklist**

The review of the *Software Project Plan* establishes the degree of risk. The following checklist is applicable (some questions may not relate to your project).

1. Is software scope unambiguously defined and bounded?
2. Is terminology clear?
3. Are resources adequate for scope?
4. Are resources readily available?
5. Have risks in all important categories been defined.
6. Is a risk management plan in place?
7. Are tasks properly defined and sequenced? Is parallelism reasonable given available resources?
8. Is the basis for cost estimation reasonable? Has the cost estimate been developed using two independent methods?
9. Have historical productivity and quality data been used?
10. Have differences in estimates been reconciled?
11. Are pre-established budgets and deadlines realistic?
12. Is the schedule consistent?

2. **Requirement Document Checklist**

The following topics are considered during review of Requirements analysis (some questions may not relate to your project).:

1. Is information domain analysis complete, consistent and accurate?
2. Is problem partitioning complete?
3. Are external and internal interfaces properly defined?
4. Does the data model properly reflect data objects, their attributes and relationships.
5. Are all requirements traceable to system level?
6. Has prototyping been conducted for the user/customer?
7. Is performance achievable within the constraints imposed by other system elements?
8. Are requirements consistent with schedule, resources and budget?
9. Are validation criteria complete?

3. <u>Design Checklist</u>

The following checklists are useful for review of the Design Document (some questions may not relate to your project):

1. Are software requirements reflected in the software architecture?
2. Is effective modularity achieved? Are modules functionally independent?
3. Is the program architecture factored?
4. Are interfaces defined for modules and external system elements?
5. Is the data structure consistent with information domain?
6. Is data structure consistent with software requirements?
7. Has maintainability considered?
8. Have quality factors been explicitly assessed?

4. <u>Coding Checklist</u>

The checklist that follows assumes that a design walkthrough has been conducted and that algorithm correctness has been established as part of the design FTR.

1. Has the design properly been translated into code? [The results of the procedural design should be available during this review.]
2. Are there misspellings and typos?
3. Has proper use of language conventions been made?
4. Is there compliance with coding standards for language style, comments, module prologue?
5. Are there incorrect or ambiguous comments?
6. Are data types and data declaration prope (if appropriate)?
7. Have all items on the design walkthrough checklist been re-applied (as required)?

5. <u>Test Plan Checklist</u>

1. Have major test phases properly been identified and sequenced?
2. Has traceability to validation criteria/requirements been established as part of software requirements analysis?
3. Are major functions demonstrated early?
4. Is the test plan consistent with overall project plan?
5. Has a test schedule been explicitly defined?
6. Are test resources and tools identified and available?
7. Has a test record keeping mechanism been established?
8. Have test *drivers* and *stubs* been identified and has work to develop them been scheduled?
9. Has *stress testing* for software been specified?

A3.4: Activities for Successful Walkthrough

Step	Action	Performed By
1	Prepare for and schedule walkthrough a. Identify walkthrough participants. d. Prepare for software product presentation.	
2	Conduct walkthrough meeting a. Designate a recorder (one of team members). b. The author provides an overview of the work product and then begins leading the class through the product step by step. c. Questions are answered, issues raised and recorded on a walkthrough log. d. The group avoids solution discussions; only defects/issues are discussed. Covering all material should be emphasized. e. Open issues are translated into tasks or action items with assignees and due dates.	
3	Perform rework a. The author corrects the product according to the issues list. b. Any issues that remain open are identified. c. The walkthrough Memorandum for Record (MFR) is prepared to reflect the results of the walkthrough, including attendees, metrics and issues or action items. d. The product is verified to fulfill the users' requirement and to be technically sound.	
4	Prepare walkthrough Inspection Report as a cover sheet to the walkthrough MFR.	Project Manager

A3.5: Guidelines for Formal Technical Reviews and Walkthroughs

1. Review the product, not the producer.
2. Set an agenda and maintain it.
3. Limit debate and rebuttal.
4. Enunciate problem areas, but don't attempt to solve every problem noted.
5. Take written notes.
6. Limit the number of participants and insist upon advance preparation.
7. Develop a checklist of potential faults for each work product that is likely to be reviewed.
8. Allocate resources and time schedule.
9. Practice.
10. Review your earlier reviews.

A3.6: Peer Feedback - Walkthroughs and Technical Reviews

1. What did you like best about the interface?

2. Which part of the program confused you the most?

3. What part of using the program seemed the most tedious?

4. Did any parts of the program appear unnatural for a user?

5. What part of the program would you change?

6. Please share any other suggestions you might have:

A3.7: Petition to Remove a Group Member

Name:

Date:

Member in Question:

Wish to remove from group?

Reasons why or why not.

Concerns about removing said member.

<Signature><Date>