Software Project Management Plan

JAMES Project

15-413 Software Engineering

Fall 1997

Carnegie Mellon University

Pittsburgh, PA 15213

Revision History:

Version R0.9 9/25/97 Bernd Bruegge. Created

Version 1.0 12/9/97 Joyce Johnstone. Revised.

Preface:

This document addresses the requirements of the JAMES system. The intended audience for this document are the designers and the clients of the project.

Target Audience:

Client, Developers

JAMES Members:

Gordon Cheng, Li-Lun Cheng, Christopher Chiappa, Arjun Cholkar, Uhyon Chung, Aveek Datta, John Doe, Phillip Ezolt, Eric Farng, William Ferry, Sang Won Ham, Kevin Hamlen, Pradip Hari, Yenni Kwek, Tze Bin Loh, Alexander Lozupone, Christopher Lumb, Vincent Mak, Darren Mauro,

Hoda Moustapha, Venkatesh Natarajan, Stan Pavlik, Michael Poole, Bob Poydence, Kalyana Prattipati, Luis Rico-Gutierrez, Michael Samuel, Michael Scheinholtz, Joel Slovacek, Ann Sluzhevsky, Marc Snyder, Steve Sprang, Paul Stadler, Herbert Stiel, Patrick Toole, Idan Waisman, Aaron Wald, Andrew Wang, Zhongtao Wang, Nathaniel Woods, Jaewoo You, Bin Zhou.

Table of Contents

Table of Figures

Revision History:

Version 0.9 Sep 3, 1997 Bernd Bruegge. Created
Version 1.0 ... Revised

Preface:
This is the controlling document for the JAMES project. It specifies the technical and managerial approaches to develop the software product. As such it is the companion document to Requirements Analysis Document (RAD). Changes in either may imply changes in the other document. All technical and managerial activities required to turnover the deliverables to the client are included. This includes scheduling, identification of tasks, and factors that may impact the project and planning.

Target Audience:
This document is intended for the members of the James project, clients, designers, and project management.

Project Members:
Gordon Cheng, Li-Lun Cheng, Christopher Chiappa, Arjun Cholkar, Uhyon Chung, Aveek Datta, John Doe, Phillip Ezolt, Eric Farng, William Ferry, Sang Won Ham, Kevin Hamlen, Pradip Hari, Yenni Kwek, Tze Bin Loh, Alexander Lozupone, Christopher Lumb, Vincent Mak, Darren Mauro, Hoda Moustapha, Venkatesh Natarajan, Stan Pavlik, Michael Poole, Bob Poydence, Kalyana Prattipati, Luis Rico Gutierrez, Michael Samuel, Michael Scheinholtz, Joel Slovacek, Ann Sluzhevsky, Marc Snyder, Paul Stadler, Herbert Stiel, Patrick Toole, Idan Waisman, Aaron Wald, Andrew Wang, Zhongtao Wang, Nathaniel Woods, Jaewoo You, Bin Zhou.

1. Introduction

Smart card technology has opened up a vast range of applications. Some of the key applications include pay phones, mobile communications, electronic cash, parking, health care and network access. The possibilities of smart cards are endless, because the smart card is a cost effective storage medium that provides high security and portability.

One important application will be the use of the card for the automotive industry as a value added service. Within Mercedes Benz there are ideas and concepts for a broad spectrum of chip card applications to support physical mobility, financial mobility (financial transactions), mental mobility (telecommunication, media services), and financial security (assurances) of the card owner.

The JAMES project described in this document covers the development of prototypical service applications for use with a smart card. This is viewed as an added service to be used by drivers of Mercedes Benz cars and automotive service facilities.

1.1 Project Overview

This document is intended for the members of the project describing the managerial aspects and technical aspects. The document is intended for planning and scheduling purposes, and serves as a summary document of the deliverables expected from each of the teams.

The schedule of project phases and milestones is shown below in Table 1. Each phase results in a document that must be approved by project management and the client liaison before it is baselined. (The baselined document is placed under configuration management).

Table 1: Project Schedule

Date	Project Phases	Project Milestones
Jul 17 - Aug 23	Requirements Elicitation	
Aug 26		Project Presentation by Clients
Aug 26 - Sep 24	Project Planning	
Sep 11- Oct 16	Requirements Analysis	
Oct 16		Analysis Review
Oct 9 - Oct 30	System Design	
Oct 28 - Nov 13	Object Design	
Oct 30		Project Review with Client (via Internet &;video conference)
Nov 8 - Nov 20	Implementation &;Unit Testing	
Nov 13 _		Object Design Review
Nov 17		Project Agreement
Nov 22 - Dec 4	System Integration &;System Testing	
Nov 25		Internal Project Review (functional prototype)
Dec 9		Project Acceptance by Client (via Internet &;video conference)

1.2 Project Deliverables

The project will a produce a running system on a Smart Card that interacts with the car and an external client/server environment in a platform independent way. The system must pass the acceptance test suite as described in the project agreement.

The following items will be produced by the JAMES System:

- A **Software Project Management Plan** defining the technical and managerial processes necessary for the development and delivery of the JAMES system (This document)
- **Agreement** between client and developers, representing a contract between the client and the developers of what is going to be delivered.
- A **Requirements Analysis Document** describing the functional and global requirements of the system as well as 4 models - the use case model, the object model, the functional model and the dynamic model. This document is created in interaction with the application domain experts.
- A **System Design Document** describing the design goals, tradeoffs made between design goals, the high level decomposition of the system, concurrency identification, hardware/software platforms, data management, global resource handling, software control implementation and boundary conditions. This document forms the basis of the object design. This document is read by the analyst as well as the object designer.
- A **Object Design Document** is which is composed of two documents. The first document is an updated RAD. The code related data will be in the form of JavaDoc output from the code from each team.
- A **Test Manual** describing the unit and system tests performed on the JAMES system before delivery along with expected and actual results. This document is used by the developers and maintainers.
- **Source code** for all subsystems of the JAMES System.

The JAMES System documentation will describe the principles of operation. The delivery consists of a presentation of the system, a demonstration of the working system and the successful passing of the acceptance test.The client expects the acceptance test to be successfully demonstrated remotely via the Internet on Dec. 9, 1997 from 8:30 pm to 10:20 pm. All work deliverables will be provided online on a project homepage. The work products will also be delivered on a CD-ROM, Dec 12, 1997.

1.3 Evolution of the Software Project Management Plan

The software project management plan is under version control. Proposed changes and new versions of the plan are announced on the course bulletin board 15-413 announce and are made available to all the project members.

1.4 Reference Materials

The following technical documentation is used by the project:

- [Bruegge-Dutoit 97] Bernd Bruegge, Allan Dutoit: *Model-Based Software Engineering: A Project-Oriented Approach*, Course Manuscript.
- [Gamma 96] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: *Design Patterns*, Addison-Wesley, 1996, ISBN 0-201-63361-2.
- [Orfali-Harkey 97] Robert Orfali, Dan Harkey, *Client/Server Programming with Java and CORBA*, Wiley & Sons, Inc, 1997.
- [Flanagan 96] David Flanagan, *Java in a Nutshell*, O'Reilly & Associates, Inc., 2nd edition, ISBN 1-56592-183-6.
- [JavaCard 97] JavaCard Consortium, *The JavaCard 2.0 API*, Draft Document, Revision July 1, 1997.
- [Rational 97a] *Using Rational Rose 4.0*, Rational Software Corporation, Santa Clara, 1997
- [Rational 97b] *A Rational Approach to Software Development Using Rational Rose*, Rational Software Corporation, Santa Clara, 1997.
- [Rational 97c] *Round-Trip Engineering with Rational Rose/Java*, Rational Software Corporation, Santa Clara, 1997.
- [CodeWarrior 97] Metrowerks, *Inside CodeWarrior Professional*, Release 1 for MacOS, Windows NT and Windows 95, Metrowerks, Inc., 1997.
- [IEEE 828] *IEEE Standard for Software Configuration Management Plans*, ANSI/IEEE Std. 828-199.
- [IEEE 1058] *IEEE Standard for Software Project Management* ANSI/IEEEStd.1058.1-1987.
- [IEEE 1074] *IEEE Standard for Developing Software Life Cycle Processes*, ANSI/IEEE Std. 1074-1991.
- Software Engineering Course Schedule, 15-413 Fall 1997, http://casacade1.se.cs.cmu.edu/JAMES/J_courseDocs /J_schedule.html.
- JAMES System: Problem Statement, 15-413 Fall 1997, Course Handout, http://casacade1.se.cs.cmu.edu/JAMES/J_courseDocs/PS /ProblemS.html.

1.5 Definitions and Acronyms

API - Applications Programming Interface
CASE - Computer Aided Software Engineering
GUI - Graphical User Interface
JAMES - Java Architecture for Mobile Extended Services
JDK - Java Development Kit
ODD - Object Design Document
OMT - Object-Oriented Modeling Technique
RAD - Requirements Analysis Document
ROSE - Visual modeling tool for Java
SDD - System Design Document
SPMP - Software Project Management Plan
UML - Unified Modeling Notation
VIP - Vehicle Initialization and Personalization

2. Project Organization

2.1 Process Model

The project is initiated on Aug 26, 1997 and terminated with the end of the semester on Dec 9, 1997. Major milestones are the Client Project Review on Oct 30, 1997 and the Client Acceptance Test on Dec 9, 1997.

The project uses an object-oriented design methodology based on the Objectory lifecycle process and uses UML for the development of the software. The development process is organized in several activities. The members of the project are organized in teams. At the end of each activity up to and including testing, each team submits documents describing the achievement of the activity. The individual approved documents produced by the teams are considered work products and are part of the software documentation. The team documents are under version control using Perforce running on a PC platform using Free BSE version 2.2. Links to the team documentation are available from the team homepages and the course electronic bulletin boards.The links to the major documents on the Perforce server are also available from the project home page. The activities and milestones are described in the next following sections.

2.1.1 Project Planning

Project planning includes description of project tasks, activities and functions, dependencies, resource requirements and a detailed schedule. This activity results in the software project management plan for the JAMES System. Another output of the planning phase is the project agreement, which is issued after the design activity is completed.

2.1.2 Requirements Analysis

The requirements analysis activity takes the problem statement and reviews it in terms of consistency, completeness and feasibility. During this activity, a set of models of the proposed system is determined by interacting with the clients resulting in the requirements model. The main part of the requirements model are four models: the use case model describing the complete functionality of the system, the object model, the functional model and the dynamic model.

2.1.3 System Design

The purpose of the system design activity is to devise a system architecture that maps the analysis model to the chosen target environment. The major part of the system design phase is the design of subsystems, that is, the decomposition of the system with respect to the chosen target platform. The system design activity also refines the use cases from the analysis model and describes in terms of interaction diagrams how the objects interact in each specific use case.

2.1.4 Analysis Review

Review of software project management plan, requirements analysis and design. The meetings will take place on Oct 16 from 9:00- 10:20 in WeH 7500. The Analysis Review consists of a set of presentations given by members of the JAMES project. Project Management will review these slides and post their comments on the 15-413 discuss bboard.

2.1.6 GUI Prototype Demonstration

This activity involves the demonstration of the graphical user interface of the JAMES System. The GUI prototype of the JAMES system is expected to be demonstrated on Nov 25 during the Internal Project Review. The prototype must be viewable on the project homepage.

2.1.5 Client Project Review

Review of project plan, requirements analysis and design decisions. The client liaison will be present at the meeting. The meeting will take place on Oct 30 from 9:00-10:20 in WeH 7500. The Client Project Review presentation slides will be made available to the client.

2.1.6 Functional Prototype Demonstration

This activity involves successful execution of a functional prototype of the JAMES System using stubs. The functional prototype of the James system will be presented during the internal review Nov 25 1997.

2.1.7 Object Design Phase

The object design phase specifies the fully typed API for each subsystem. New classes are added to the analysis object model if necessitated by the system architecture. Attributes and methods for each object are fully typed.

2.1.8. System Integration Prototype Demonstration

This activity involves the demonstration of a fully functional system prototype based on the subsystem decomposition. Each subsystem is represented by its service. All service operations can be called by other subsystems using remote method invocation. The implementation of the services can be stubbed out.

2.1.9 Implementation

The focus of this activity is on coding the individual objects described in the object design document.

2.1.10 Unit Testing

During unit testing, test suites are designed and executed for objects or collections of objects in each subsystem. Unit testing enables the individual subsystems to be tested independent from the status of the other subsystems. The result of this activity is part of the test manual that describes how to operate the test suite and how to interpret the test results.

2.1.11 System Integration

During this activity an integration strategy is devised, that specifies the order in which the subsystems of the JAMES system are integrated and tested with respect to the use cases defined in the analysis model. The system integration strategy and the subsystem tests are described in the Test Manual.

2.1.12 System Testing

Structural Testing: This activity tests the major data paths in the complete JAMES System.**Functional Testing:** Tests the major functionality (use cases) with the complete JAMES System. The basis for the functional testing activity is the test manual which is revised according to the results of the system testing phase.**Alpha-test (Client Acceptance Test):** The system is tested to make sure it passes the client acceptance criteria as defined in the project agreement.

2.1.13 Manual Integration

During this activity, the project deliverables are revised. As a result, a complete set of documents consisting of the software project management plan, requirements analysis document, software design document, test manual and source code is made available on the project home page. The system documentation will also be printed on a CD. Each of the students taking the course as well as the clients will receive a CD.

2.1.13 Client Presentation

At the Client Presentation, a slide presentation and software demonstration will be given in the Intelligent Workplace at Carnegie Mellon University. The software developed during the project will be demonstrated for the client acceptance test. The clients will attend the client acceptance test in person or via video conference.

2.2 Organizational Structure

Below is the organizational chart of the people involved in the development of the JAMES system .

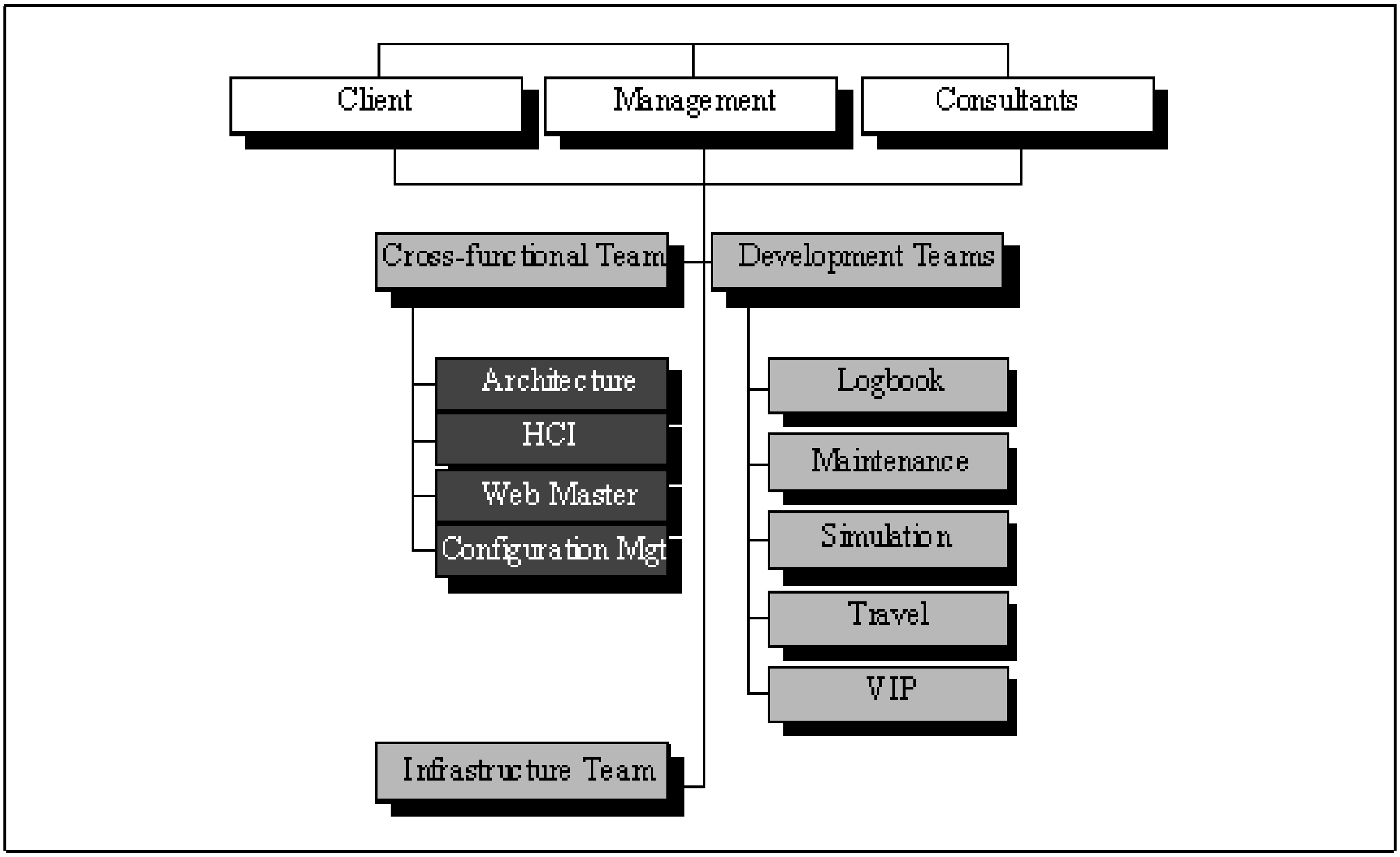

Figure 1: Organization Chart for JAMES Project

The clients of the JAMES System project are:
Dieter Hege, Brigitte Pihulak, Daimler Benz
The project managers are:
Bernd Bruegge, Malcolm Bauer, Brian Cavalier, Allen Dutoit, Alfonso Guerrero-Galan, Sam Perman, Isabel Torres-Yebra
The Infrastructure team consists of:
Joyce Johnstone (Web Master and Lab Management)and Stephan Schoenig (CASE Tool and Java support, Communication Infrastructure)
The project consultants are:
Dieter Hege, Klaus Eitzenberger, Manfred Mueller, Juergen Bortolazzi, Claus Czymmek, Arno Schmackpfeffer, Thorsten Armstroff

DB user group members and list of consultants

2.2.1 Teams and Tasks

This section describes the teams working on JAMES subsystems and their tasks:

- Logbook Assistant (Logbook Team)
    - Distinguish between business and private trips
    - Support disconnect mode while driving the vehicle
    - Make sure every trip is stored on the Smart Card, even if the driver forgets to enter any information at the beginning of the trip.
    - Make it impossible to manipulate the trips in the logbook that have already been taken.

    - Allow multiple drivers to record the use of the same vehicle
    - Allow logbook entries for 50 trips
    - Allow multiple use of the vehicle by different drivers (fleet management, identification with the smart card)
    - Create form to report tax deductible usage of the car.
- Maintenance Assistant (Maintenance Team)
    - Authenticate the customer to the system
    - Maintain maintenance history of the vehicle
    - Communicate with Legacy System
    - Track accumulated Bonus Points
- Travel Assistant (Travel Team)
    - Create the core component of the travel assistant
    - Plan trip functionality
    - Modify trip functionality
    - Locate car on map
    - Guidance on map
    - Also, if possible and time permits, the following:
        - Point out sites of interest
        - Dynamic (on the road) trip alteration
        - Monitor weather conditions and detours
        - Location of car if stolen
- VIP Assistant (VIP Team)
    - Build subsytem to communicate user preferences to vehicle (seat position)
    - drive control adjustments: cockpit control settings, suspension and fuel economy
    - convenience adjustments: position of the seats, mirrors, radio station, audio-level and air-conditioning
    - business adjustments: addresses/telephone book, information from the mailbox, appointments bookand logbook settings
    - general adjustments: settings of navigation system, information systems (such as road conditions, weather information)
- Vehicle (Simulator Team)
    - Access to the functionality of a Mercedes Benz Car simulator provided by the Daimler Benz R&D,; division (F1).
    - Provide platform independent interface to the simulator.
    - Provide web-based access to the simulator.
    - Show that JAMES applications are able to be integrated with the existing electronic systems in the car (mirror/seat position, speedometer).
- User Interface (HCI Team, Cross-Matrix Team)
    - Set standards for common GUI elements
- Software Architecture (Architecture Team, Cross-Matrix Team)
    - Maintain JAMES's system model
    - Ensure consistency of JAMES's system model with the design model
    - Define software architecture
    - Create common class library
    - Create system integration plan
- Documentation (Cross-Functional Team)
    - Integrate the system documentation (SPMP, RAD, SDD, ODD, Test Manual)
    - Review final deliverable documentation for completeness
- Infrastructure Team
    - The purpose of the infrastructure team is to provide smooth operation of the vehicle systems lab and software tools supporting the development. The members are Joyce Johnstone and Stephan Schoenig. Their office is in Building BOM D 154. Their responsibilities are:
        - Web Master, Lab Management: Joyce Johnstone
        - Lotus Notes and Perforce: Stephan Schoenig

2.3 Organizational Boundaries and Interfaces

2.3.1 Electronic BBoard Communication

The Lotus Notes Databases shown in Table 1 will be used for electronic communication in the JAMES project. Note that these databases are intended to replace Andrew bulletin boards academic.cs.15-413 that have been set up for this course (The Andrew bboards are neither used nor read by project management).

Table 2: Electronic Bboards for JAMES Project	
Announcements	Lecture and project announcements
Discuss	Group discussion
Issues	Structured discussion providing for Issues, Proposals, Arguments, and Resolutions
Client Discuss	Primary forum for interchange with the clients
Handin	For electronic submission of homework
Help	Request for assistance in course material, software applications
Review of Documents	Documentation submittal for review with automatic e-mail notification of the reviewers and inclusion of the comments on the initial post with the document. The sequence of the reviews can be set up in parallel or series.
Logbook Team Discuss	Discussion about the Logbook Assistant
Maintenance Team Discuss	Discussion about the Maintenance Assistant and Bonus Assistant
Simulation Team Discuss	Discussion about the Vehicle Simulator
Travel Team Discuss	Discussion about the Travel Assistant
VIP Team Discuss	Discussion about the Vehicle Initialization and Personalization
Architecture Discuss	Discussion of project architecture and class library
HCI Discuss	Discussion of user interface design
Configuration Discuss	Discussion of the use of the Perforce client and server
Documentation Discuss	Used for informal review of documentation and posting of resource material

Every team member has to:

- Bookmark the Announcements, Discuss, Client, Handin, and Help Bboards
- Bookmark the team specific bboard
- Check these bboards at least twice a day

Communication with the client is primarily via the Client BBoard. As the need arises direct e-mail and/or telephone contact is set up with specific consultants within the client organization.

2.3.2 Meeting Times

There is a weekly project meeting for each group. The initial project meeting times are:

Group	**Day**	**Time**	**Location**
Logbook	Monday	8:00 p.m.	Wean 8220
Maintenance	Wednesday	4:30 p.m	Smith 100
Simulation	Thursday	4:30 p.m.	Smith 101
Travel	Tuesday	6:00 p.m.	Porter 231B
VIP	Thursday	5:30 p.m.	Wean 7220
HCI	Tuesday	4:30 p.m.	Hamburg 1202
Architecture	Tuesday	5:00 p.m.	Smith 100

2.4 Project Responsibilities

Management of the JAMES System is done with the following roles: project management, coach, group leader, Architecture liaison, HCI liaison, document editor, configuration manager, card master and webmaster.

2.4.1 Project Management

The project management function has the following responsibilities:

- Schedule and prepare meetings with clients
- Assign presentations (In-class Project meetings, client review, client acceptance test) to project members
- Listening to gripes from the team members
- Resolve conflicts if they cannot be resolved otherwise

2.4.2 Coach

The coach has the following responsibilities:

- Review weekly team progress
- Attend weekly team meetings
- Insist that guidelines are followed

2.4.3 Group Leader

The group leader leads an individual team. The main responsibility of the group leader is to manage the action items of the group. In addition he or she has the following responsibilities:

- Responsible for intragroup communication
- Run the weekly project meeting
- Define, post and keep track of action items (who, what, when), i.e the agenda
- Measure progress and enforce milestones
- Deliver work packages for the tasks to project management
- Deliver project plan and accomplishment for project phase to project management
- Coordinate and schedule use of resources needed by team (lab, tools,...)
- The group leadership position has to be rotated on a regular basis among the team members.

2.4.4 Architecture Liaison

The liaison interacts with the liaisons of the other teams and with the project management. Each team has a liaison to the Architecture Team. The responsibilities of the liaison are:

- Responsible for intergroup communication
- Make available public definitions of each subsystem service ("API") to the other teams (ensure consistency, etc.)

- Coordinate tasks that overlap subsystems with the teams
- Responsible for team negotiations, that is, resolve technical issues spanning more than one subsystem
- Defines the software architecture for JAMES
- Defines the class library for JAMES

2.4.5. HCI Liaison

The liaison interacts with the liaisons of the other teams and with the project management. Each team has a liaison to the HCI Team. The responsibilities of the liaison are:

- Responsible for inter-group communication
- Make available public descriptions of each subsystem interface to the other teams (ensure consistency, etc.)
- Coordinate interfaces for overlapping subsystems with the teams
- Resolve technical interface issues spanning more than one subsystem
- Defines the hardware and software elements of interface for JAMES
- Develops the style ("look & feel") for JAMES

2.4.6 Documentation Editor

The editor in each team is responsible for producing the documentation of the current project phase and:

- Collect, proofread and distribute team documentation to the Architecture team
- Interaction with the Architecture team
- Writes minutes and provides them to team Webmaster

2.4.7 Configuration Manager

The responsibilities of the configuration manager in each team are:

- Coordinate change requests
- Provide version control for group's working directory
- Coordinates configuration management issues with other groups
- Installation of group specific software and hardware

2.4.8 WebMaster

The responsibilities of the webmaster in each team are:

- Maintain the Team Homepage
- Coordinate team page with course Webmaster
- Link Meeting Agendas, Minutes, Action Items and Issues to the team homepage
- Maintain links to team documentation under version control

2.4.9 Card Master

The responsibilities of the card master in each team are:

- Maintain Cyberflex environment
- Keep track of updates from Schlumberger
- Understand Java Card API
- Be able to download of applets (applets to cardlets)
- Manage team's JavaCard

Table 2 describes the group leader assignments and Table 3 indicates the other team assignments.

Team Leader Assignments

	Logbook	Maintenance	Travel	VIP	Vehicle
Requirements Analysis	Uhyon Chong	Arjun Cholkar	Bin Zhou		Andrew/Jay
System Design	Pradip Hari, Michael Poole	Joel Slovacek	John Doe		Robin
Object Design	Michael Stienholtz, Aaron Wald	Yenni Kwek	Kalyana Prattipati		Hoda Moustapha
Implementation	Herb Stiel	Vincent Mak	Christofer Chiappa		Will
Testing	Nate Woods	Darren Mauro	Michael Samuel, Ann Sluzhevsky		Paul/Ogi

Team Role Assignments

	Logbook	Maintenance	Travel	VIP	Vehicle
Card Master	Michael Scheinholtz	Arjun Cholkar	Bin Zhou	Philip G Ezolt	Jaewoo You
CASE Modeler	Herbert Stiel	Joel Slovacek	Michael Samuel	Christopher Lumb	Hoda Moustapha
HCI Liaison	Nathaniel Woods	Yenni Kwek	Gordon Cheng	Idan Waisman	Paul Stadler
Architecture Liaison	Pradip Hari	Vincent Mak	Ann Sluzhevsky	Venkatesh Natarajan	Andrew Wang
Configuration Manager	Michael Poole	Darren Mauro	Chris Chiappa	Eric Farng	William Ferry
WebMaster	Uhyon Chung	Aveek Datta	Kalyana Prattipati	Li-Lun Cheng	Tze Bin Loh
Documentation Editor	Aaron Wald	Stanislav Pavlik	John Doe	Patrick Toole	Bob Poydence
Coach	Alfonso Guerrero-Galan	Sam Perman	Isabel Torres-Yebra	Brian Cavalier	Bernd Bruegge

3. Managerial Process

3.1 Management Objectives and Priorities

The philosophy of this project is to provide a vehicle for students to get hands-on experience with the technical and managerial aspects of a complex software problem. The emphasis is on team work and encouraging individual teams to work together towards the goal of implementing the JAMES system completely.

3.2 Assumptions, Dependencies and Constraints

The functionality of the JAMES System is achieved when the client acceptance test can be executed.

Each software development activity results in one or more documents to be submitted to the project management before the deadline. Each document is reviewed at least once by the project management before it is accepted and becomes a baseline document.

The following documents will be graded: SPMP, RAD, SDD, ODD, TM and are worth each 10 points. The agenda, minutes, action items and issues for each weekly team meeting have to be posted. The complete set of these is also worth 10 points. We will give a project A to everybody who participates in the project if all the project deliverables are delivered and the JAMES system passes the client acceptance test as defined in the requirements analysis document.

The JAMES System is a project that puts emphasis on collaboration, not competition between the students. We will not accept a system that is done by one team alone.

3.2.1 Assumptions

3.2.1.1 Logbook:

- That the card is used by the driver.
- We will be able to lock data on the card so that it can not be tampered with.
- We can determine what information tax collection agencies need in order to stipulate that a trip is tax deductible.

3.2.1.2 Maintenance:

- All of the dealers will have access to a smart card reader and to the Legacy System (via the Web).
- We will be in communication with the client throughout the duration of the project.
- The functionality and information on the card does not get corrupted.

3.2.1.3 Simulation:

- System wide architecture will be able to meet constraints of simulation team resources.
- Card Master keeps up with the latest news about the changes of Cyberflex.
- That simulator documentation will be obtainable.
- That simulator hardware will be reliable.
- The Perforce software will function as expected and will not be difficult to learn or use.
- Space will be provided by the CMU AFS servers to store the web page files.
- Resources will be made available expeditiously to the web master for posting upon request.
- AIM web interface can be integrated fully into the team web page.
- There are no bugs in the CASE tool ROSE. The learning curve of case tool is not very high.
- We have full license to create and add things that we see fit to the hardware architecture.
- Communication via CORBA to the rest of the application is attainable on the linux platform.
- We can all learn the programming tools without difficulty.
- There are enough computing resources.
- The smart card technology is sufficient for the needs of the project.
- All software development tools will perform to specifications.

3.2.1.4 Travel:

- CD-ROM based map application will be available and APIs can be obtained.
- Speech input/output resources will be available.
- Maps are downloadable from web within acceptable timeframe.
- GPS system can be obtained with APIs.
- The digital screen resolution planned for the vehicle is adequate for graphic display of the maps.

3.2.1.5 VIP:

- That the simulator works and we can interface to it.
- That we have some way to interface with the vehicle itself (i.e. seats, etc.), or have the means to create some interface.
- Sufficient number of machines will be available so that we can write the needed code.
- Enough space exists on the card or that there is some way for us to store needed information there, or access said information by other means.
- We also assume that all tools to be used work properly.

3.2.2 Dependencies

3.2.2.1 Logbook:

- The HCI team will come up with a system to interact with the logbook that would allow us to meet some of our goals. Such as distinguishing between drivers, and allowing additional information about a trip to be entered.
- There is a way to distinguish between drivers, get the location of the car, mileage of the car, etc, from the car systems.

3.2.2.2 Maintenance:

- We depend on the stability of the hardware and software involved in the development of the project.
- We depend on other groups in the project to provide us with objects we need.

3.2.2.3 Simulation:

- Other subsystems depend on the simulation team to test their products.
- Other members depend on card master to provide information on javacard and Cyberflex.
- Depend on vendors and client for certain documentation and simulator hardware.
- Other team members depend on a version control policy that will not slow down their ability to make and implement changes to the code.
- Daimler Benz's provision of the AIM system . Group leader's submission of agendas before each meeting and the documentation editor's submission of minutes one hour after each meeting. Chief web master of JAMES project to provide mirror site of the team web page.
- Other member depends on case modeler to "publish" their design decisions.
- AIM Simulator from Daimler Benz. Needs of other teams on the project. Good communication among team and non-team members (via liaisons).
- Proper timing of project phases.

3.2.2.4 Travel:

- We will receive inputs from the car (e.g. Speed). API's are available for external systems.
- Simulation team is able give us usable system.

3.2.2.5 VIP:

- We will be able to send and receive data from the simulator.
- A bridge pattern will work to interface with either the simulator or vehicle.
- The JavaCard has adequate space for data storage.
- Hardware and software resources continue to function properly.

3.2.3 Constraints

3.2.3.1 Logbook:

- We will be able to come up with a scheme to store 50 trips on a single java card.

3.2.3.2 Maintenance:

- Quantity of memory storage space on the card.
- Deadline and time constraints.
- All the code has to be developed in Java.

3.2.3.3 Simulation:

- Simulation system must identify and meet the needs of the other subsystems.
- Our source code has to comply with the limits of javacard.
- Must rely on other people and teams to supply certain documentation.
- Must rely on team members to follow version control policy and take care to verify that any code that is committed does not cause the system to unexpectedly break.

- The ROSE tool automatically generates code in Java JDK 1.02.
- Must provide an acceptable interface to the automobile which is easily and readily accessible to the other project teams. Restricted to features feasibly applicable in an automotive environment.
- Must be CORBA architecture compliant (or something else that fits into the grand scheme of things).

3.2.3.4 Travel:

- Java language specifications and the Java virtual machine.
- Card will have enough useful space.

3.2.3.5 VIP:

- Limited by the space/functionality provided by the card.
- Quality of the simulator (so that we can perform sophisticated tests).
- Short delivery time, something that can be done by December 9.

3.3 Risk Management

We assume that the hardware/software configuration on top of which the JAMES System is built, is performing reliably.

The individual team risk management assessments are as follows:

3.3.1 Logbook Team

Risk: If we do not have a solid understanding of the JavaCard and how it works we risk designing a system that will not work with the actual card.

> Contingency: Allot time to experiment with small prototypes on the JavaCard.

Risk: The storage constraints on the card might limit the number of trips that can be stored.

> Contingency: Reduce the number of trips westore locally on the card, perhaps develop a system that allows for transfer of trips from card to PC.

Risk: The interface to the car might limit what sort of data can actually be collected, and this may or may not be enough for meeting the goal of creating a tax deductible form.

> Contingency: Create a partial form.

3.3.2 Maintenance Team

Risk: We may get over ambitious, try to implement more features than allowed for by our time and resource constraints.

> Contingency: We will try to implement all of the required features first and leave room for additional features if time permits. We will break up the requirements into smaller tasks so that they are more flexible and easier to manage. The tasks will be prioritized. The worst case scenario would require us to start cutting features with low priority.

Risk: Team members drop the course.

> Contingency: We must then redistribute the tasks among the remaining members. By keeping all of the other members informed on what we are working on, it will help to lessen the learning curved involved in taking over a project for someone else. By breaking the project into smaller, more manageable tasks, the extra work would be easier to handle.

Risk: We miss an important feature in the software system to be built.

Contingency: Develop a functional prototype and request a review by the client liaison.

Risk: The client is unable to respond to questions about the system in a reasonable time frame.

Contingency: Post the question to project management and otherwise use the team's concensus.

3.3.3 Simulation Team

Risk: Other subsystems decide that they need a different architecture or functionality from simulation team.

Contingency: Keep in constant communication with other subsystems until we can identify a solution that will satisfy their needs and our constraints.

Risks: If we don't have a clear understanding of the limits of javacard and Cyberflex, we may come up with solution that cannot be fit into the card.

Contingency: Our source code has to comply with the limits of javacard.

Risk: Simulation documentation will be overwhelming, time consuming, or not in English.

Contingency: Streamline documents, get uniform formats, and delegate some work.

Risk: Team member may be working on test code that they do not want to commit to the archive because they are not sure if the code will break the system.

Contingency: Need to determine a protocol for maintaining test branches of version tree, or some other method to allow such testing code to be produced without affecting main project.

Risk: Team member may be unable to work on a file from the depot because somebody else currently has the file open.

Contingency: Need to establish a system to contact team members so that files can be unlocked. Need to ensure that only files being actively edited are held open by any member.

Risk: Failure of servers containing the web files.

Contingency: Reduce this risk by coordinating with chief web master to update mirror site on a constant basis.

Risk: Miscommunication between webmaster and dependencies leading to inaccuracy and delay of new information.

Contingency: Duplex communication between web master and other team members will allow quick recovery and correction of errors.

Risk: Redundancy of web information due to repetitive information found in JAMES web page.

Contingency: Have to achieve a good balance between convenience (through providence of links) and excessive repetition of easily available web resources.

Risk: The ROSE case tool will not support java 1.1. The ROSE case tool has bugs or is not efficient.

Contingency: We use ROSE for making diagrams and manually generate our own java 1.1 code or restrict ourselves to using java 1.02.

3.3.5 VIP Team

Risk: Porting the system to multiple vehicles. The hope is that there will be some translation device that will adjust data so that even if you switch models/makes of cars there will be little to no need to change the settings.

> Contigency: If this option is not possible then we will need to discuss with the client which vehicles need to be implemented so that a maximal number of platforms will be supported within the time allotted.

Risk: the code will not be scalable to allow for new features and cars with ease.

> Contigency: The only real contingency for this is to inform the client of the functionality available and to decide whether there is time enough to alter the code so that it is scalable with a fair amount of ease.

Risk: Dependency on the Simulation and Architecture teams.

> Contingencies are we have our representative encourage them to speed up or help. Failing that we will discuss ways to get them on schedule with the staff. Failing there we will write the code we need provided there is sufficient time. If there is not enough time then we will discuss with the client to maximize desired functioning features within the remaining time.

Risk: Smart Card is supposed to work on every Mercedes Benz car. But because of the fact that different cars have different stuffs, compatibility is a risk. Some functionality may be supported by a certain kind of cars but may not be supported by some cars else. For example, some cheap car may not have air condition, the card we are going to make shall not only work on the cars with air condition (means user can control the air condition through car) but also work on the cars without air condition. If the shape of the car is new and fancy, the card may not support it at all because we don't know what new features will be in it.

> Contingency: In order to make product be compatible, we need to build object models of cars. This also has risk that the current object tool only supports Java 1.0, we may not do as many thing as we want.

Risk: The size of the card is limited. If we don't design it carefully, it may be possible that the final code won't fit on the card.

> Contingency: Carefully design and implement the system.

3.4 Monitoring and Controlling Mechanisms

For each project meeting each team produces an agenda and the minutes of the meeting. The minutes have to contain explicitly the action items assigned during the meeting. The agenda and minutes are posted on team specific bulletin boards by the minute taker of the meeting.

The baseline documents are reviewed by the project management. It is expected that each document undergoes one or more iterations.

4. Technical Process

4.1 Methods, Tools and Techniques

Our development methodology is based on a combination of use cases (from the OOSE methodology) [Jacobsen 92] combined with the OMT methodology [Rumbaugh 1991].The following tools are available to support the management and development of the JAMES project:

Code Warrior: A set of Java tools for the "back end" of software development, compilation, editing and debugging of Java programs.

Netscape Communicator Internet browser

Adobe PageMill 2.0, **Claris Home Page 2.0, FrontPage** HTML editors

Rose/Java CASE tool for UML by Rational Software Corporation.

Adobe Acrobat 3.0 Portable Document Format Software Reader

Powerpoint 4.0 Slide Presentation program

4.2 Software Documentation

The following activities result in a project deliverable: -
Project Planning: Software Project Management Plan (SPMP) -
Requirements Analysis: Requirements Analysis Document (RAD) -
Analysis Review: Analysis Review Slides -
System Design: System Design Document (SDD) -
Client Review: Client Review Slides -
Object Design: Object Design Document (ODD) -
Reviews: Review Presentation Slides -
Implementation and Unit Testing: Code -
System Integration and System Testing: Test Manual -
Delivery: Client Acceptance Test Slides

4.4 Work Elements, Schedule and Budget

4.4.1 Overall Project Plan

The overall project plan follows the sawtooth model, a modified waterfall model. 3 prototypes have to be delivered: A graphical user interface, a functional prototype and a system integration prototype. Analysis is started before Project Planning is finished. System Design is followed by Object Design. Important Milestones are the Analysis Review Oct 16, the Project Status on Oct 23, the Project Review on Oct 30 and the Object Design Review on Nov 13 and Nov 15. Implementation and Unit Testing are scheduled to overlap significantly. System Integration is scheduled to immediately follow Unit Testing. System Testing starts immediately after system integration and leads to the Client Acceptance Test on Dec 9.