



UPOS Software Development Plan



UPOS-113-01

October 2000

Version 1.0

By
Louis P. Butler
Robert DeMajistre

1	Scope	4
2	Project life cycle	4
3	Development Process	5
3.1	Product Development Process	7
3.1.1	Product Development Planning	7
3.1.2	Product Requirements Analysis	8
3.1.3	Product Design	9
3.1.4	Product Implementation	11
3.1.5	Product Testing	11
3.2	Transition Team Development Process	11
3.2.1	Transition Planning	11
3.2.2	Product Integration	11
3.2.3	Build Integration	12
3.2.4	Build Testing	12
3.2.5	Build delivery	13
3.3	Transition Development Environment	13
3.3.1	AFWA development environment	13
3.3.2	AFSWC Environment	14
3.4	Configuration management	15
3.4.1	Project team configuration management	15
3.4.2	Transition team configuration management	15
3.5	Corrective action	16
3.5.1	Scope	16
3.5.2	Organization	16
3.5.3	Process	16

REVISION LOG

Rev	Date	Description	By
-	Sep 1999	Initial Release	Louis P. Butler Robert DeMajistre
A	Oct 2000	Customer Approved	Louis P. Butler

1 Scope

This document describes the planned software development process for the UPOS project. It is assumed that the reader is familiar with the UPOS project, including its purpose, organization and structure. This plan is applicable to all software developed as a part of the UPOS program.

2 Project life cycle

Deliveries are to be made in three to six month builds. Each build will implement a tailored form of the software process cycle described in this plan. An illustration of the iterative development process is shown in Figure 1. Each UPOS project team will define and develop products for integration by the transition team. During the course of the UPOS program, a project team may develop several versions of the same product. These products will be integrated into builds when they become available to the transition team. The transition team will develop a new build from the available projects every three to six months. After the delivery of a build, it is the transition team's responsibility to provide support to the customer for that build (with help from the project teams when necessary).

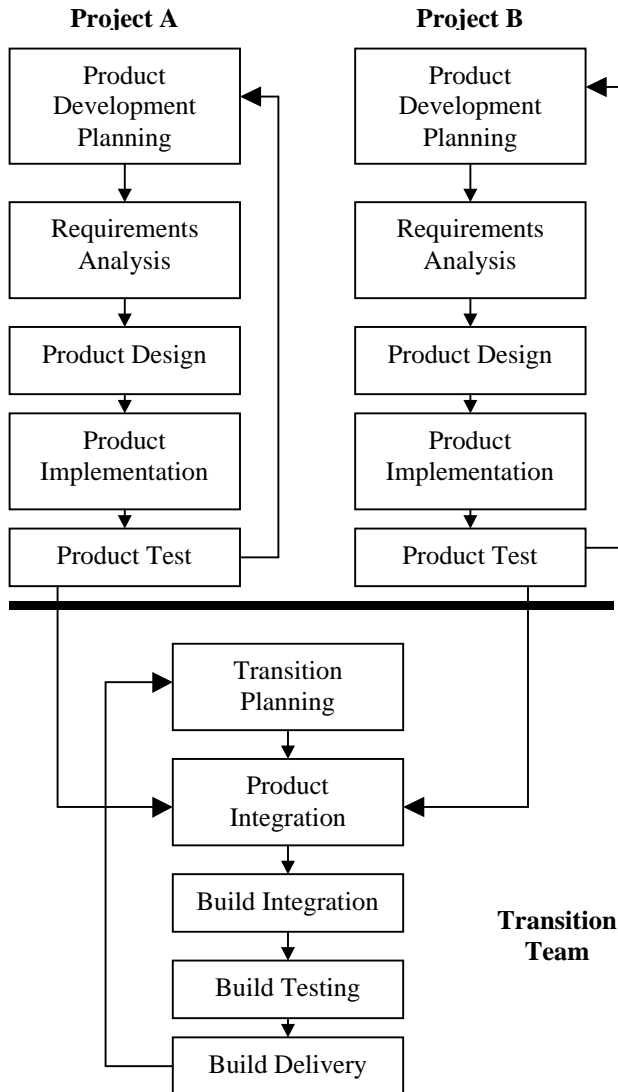


Figure 1. The process for the development of a single build. The figure shows two example projects and the transition from project development to build delivery. Products are developed iteratively and these iterations may not be synchronous with the transition process. The actual number of active project teams is expected to change over the course of the program.

3 Development Process

As is illustrated in Figure 1, each delivered build will be the result of the effort of both the project teams and the transition team. The development process for the transition team is different than the process applied by the project teams, and we will therefore consider them separately. As part of the development process, there will be standard documentation and reviews. Not all of these documents and reviews will be deliverable to the Air Force. Table 1 and Table 2 provide a summary of the documentation and reviews that will be part of the development process. Details regarding the content of the reviews and documentation are described along with the processes below.

Document	Delivery Type	Author	Process Phase	Brief Description
Functional Requirements	Deliverable	Project Lead	Product Development Planning	List of functions that the product will perform
Product Build Plan	Internal	Project Lead	Product Development Planning	Describes development, tasking and scheduling
Software Requirements	Internal	Project Software Lead	Requirements Development	Informal document or memo describing software requirements and constraints
Software Design	Deliverable	Project Software Lead	Product Design	Description of the software architecture and a module by module description of the software
Source Code and Support Files	Deliverable	Project Lead/ Transition Team	Product Implementation /Build Integration	Source code and support files necessary to install, compile and run
Product Description	Deliverable	Project Lead	Product Implementation	Brief summary of the product including a summary of input, output and algorithms used
Test Plan	Internal	Project Software Lead	Product Testing	Identify tests for each software requirement and resource requirements for each test
Product Test Report	Internal	Project Software Lead	Product Testing	Describes testing procedures and results
Build Plan	Internal	Transition Team	Transition Planning	Describes development, tasking and scheduling for the build
Transition Plan	Internal	Transition Team	Product Integration	For each component product describes requirements for transition
Build Requirements	Internal	Transition Team	Build Integration	Describes integration requirements for the build. References the transition plans
Build Test Plan	Internal	Transition Team	Build Testing	Identify tests for each software requirement and resource requirements for each test
Test Report	Deliverable	Transition Team	Build Testing	Describes testing procedures and results
Operators Manual	Deliverable	Transition Team	Build Integration	Instructions for the operator on how to use the delivered software
Version Description	Deliverable	Transition Team	Build Integration	Brief summary of the build being delivered
Support Plan	Deliverable	Transition Team	Build Delivery	Plan of how the build will be supported at the customer's site
Delivered Build	Deliverable	Transition Team	Build Delivery	Source code and support files necessary to install, compile and run

Table 1. Documentation associated with software development.

Review	External/ Internal	Responsible	Process Phase	Description
Algorithm Interchange Meeting	External	Project Lead	Product Development Planning	Review the basic concept of operations and functional requirements with the customer
Software Requirements Review	Internal	Project Software Lead	Requirements Analysis	Review of software requirements to assess completeness and accuracy
Design Walkthrough	Internal	Project Software Lead	Product Design	Technical discussion and evaluation of the software design
Code Walkthrough	Internal	Project Software Lead	Product Implementation	A review of the code by an independent software developer
Completion Review	Internal	Project Lead	Product Testing	Review to the Project lead and transition team at the point of delivery for transition
Acceptance Review	External	Transition Team	Delivery	Review to the customer introducing the newest build

Table 2. Reviews associated with software development.

3.1 Product Development Process

Each active UPOS project team will deliver one or more products to the transition team for incorporation into a delivered build. The products include software, data and documentation that will be delivered to the customer's site. These products will be developed according to the process described here.

The documentation for this process is summarized in Table 1. Both deliverable and internal documentation will take the form of memos or email messages. Deliverable documentation will be in memo form and tracked according to the memo tracking process at the originator's facility. At minimum, a dated copy of the memo, initialed for receipt and concurrence will be uniquely numbered maintained at the originator's facility. Outlines of some of the deliverable documents will be given in an appendix to this document.

In cases where existing software is to be used for a UPOS project, the requirements for some of the internal documentation will be waved at the request of the project lead and the concurrence of the customer. All deliverable documentation, however, will be produced. There will be cases where the deliverable documentation assigned in this plan to the project team will be written by the transition team. This tailoring of the process should be documented in the Product Build Plan.

The reviews for this process are summarized in Table 2. Many of the reviews required by this process are internal and do not require the presence of the customer. The attendees for these each review is given below. Internal reviews may be tailored by the project lead and may be conducted via telephone where practical. External reviews will be conducted with the customer in attendance. Review materials for external reviews will be distributed to the customer one week before the review for review and comment.

3.1.1 Product Development Planning

During the product development and planning phase, the project team defines the functions to be implemented, the resources required and the schedule for the development. During this phase, the project team will coordinate with the IPT and the customer to determine the proper scope of the product and the operational concept for its use. In addition to defining the work to be accomplished, the project team must plan for its accomplishment. Much of the project planning may occur before the project is selected by the IPT for implementation.

3.1.1.1 Entry Criteria

- The project is selected by the IPT for activation
- Staff becomes available to work on product development planning activities

3.1.1.2 Exit Criteria

- Documentation and reviews for this phase are successfully delivered
- Project team has a well defined operational concept and functional requirements
- A working schedule has been developed

3.1.1.3 Documentation

Functional Requirements: This document outlines the operational concept and the basic functions that the product is to perform. There are four basic areas that this document should address:

1. Operational Concept: This section should outline the purpose of the product and how it is to be used. Specific issues to be addressed are a) how this product fits into the customer's operational system from a qualitative perspective, b) a list of the data required to produce the product, c) how often the product should be produced, d) a summary of the user interface.
2. Input data: a list of the requirements for the types of data to be input. This includes update frequencies, required data fields, communications methods and general methods for handling of missing data.
3. Output data: A description of the type of output to be generated. This includes a detailed description of any graphical output or output file formats.
4. User interface: General description of the user interface for both input and output.

A sample outline for this document is provided in the appendix. This document is the responsibility of the project lead and requires the concurrence of the IPT.

Product Build Plan: This document describes the plan for implementing the product. This document only needs to cover the most recent build. This document should contain the following sections

1. A list of the major tasks to be completed
2. The resources (people and equipment) required to accomplish each task
3. A schedule for development, including the milestones necessary for tracking and a list of assumptions made during the construction of the schedule
4. A description of the assumed responsibilities of the project and transition teams

This document is the responsibility of the project lead and may be reviewed by the IPT upon its request.

3.1.1.4 Reviews

Algorithm Interchange Meeting: This is a meeting held with the customer to discuss the operational concept and functional requirements of the project. The agenda is negotiated between the project lead, the IPT and the customer. These meetings have already been held for the first four space weather projects.

3.1.2 Product Requirements Analysis

3.1.2.1 Entry Criteria

- Working approval of functional requirements and build plan by the IPT
- Resources available for requirements analysis

3.1.2.2 Exit Criteria

- Agreement between the project lead and project software lead on software requirements

3.1.2.3 Documentation

Software requirements: This is an informal document that may take the form of a memo that contains a description of the software requirements for the project. The following elements should be addressed:

1. A description of the external interfaces to the software under consideration. This includes input and output file formats as well as any other types of interface that the software must support. If an interface needs to be negotiated with the user, a memorandum of agreement or other written communication should be written to document the results of the negotiation. Data format documentation existing at the user's facility should be referenced.
2. The capabilities of the software. This is a list of testable requirements for processing. These capabilities will form the basis for software testing. These requirements may take the form of prototype software. That is, a requirement may be written such that processing be consistent with an existing piece of prototype software.
3. Sizing and timing requirements.
4. Design constraints. List any factors that will influence or limit the design. Examples include operating system limitations, required third party software, security constraints, existing software at the users site, etc.
5. Testing methods. A recommendation on how the software should be tested.

3.1.2.4 Reviews

Software requirements review: This is an informal meeting between the software lead and the project lead. At this meeting, the software lead presents the software requirements for approval by the project lead. It is recommended that this meeting be attended by an independent software engineer to evaluate the suitability of the requirements as a basis for software development.

3.1.3 Product Design

In many cases, project teams will be modifying existing software to meet the UPOS requirements. In these situations, the product design phase will consist of two parts. First, the required modifications to existing code must be identified and designed. Second, the design of the modified software must be documented in the manner described below.

3.1.3.1 Entry Criteria

- A successful software requirements review has been held

3.1.3.2 Exit Criteria

- A successful design walkthrough is held
- The baseline software design document is approved by the IPT and placed under project configuration management

3.1.3.3 Documentation

Software Design: This document is deliverable to the user and is intended to be used as software maintenance documentation. The initial version should be produced by the project team and maintained under project configuration management. In cases where the project team is unwilling or unable to produce this document, it will be produced and maintained by the transition team. If the transition team produces this document, suitable resources to do so will be made available.

The software design document should contain the following information

1. A description and diagram of the structure of the software down to the module level.
2. A detailed description of the input, output and temporary files and global (common) storage used by the software, cross-referenced to the modules that use them.
3. An English language description of the inputs, outputs and logic of each module.

3.1.3.4 Reviews

Design Walkthrough: This is an informal review, led by the project software lead. The project lead and at least one independent software developer should attend. At this meeting, the project software lead will present the design in detail. Detailed designs for existing code need not be presented, however modifications to the existing code should be reviewed at this meeting.

3.1.4 Product Implementation

3.1.4.1 Entry Criteria

- A successful design walkthrough is conducted

3.1.4.2 Exit Criteria

- Code and support files are ready for product level testing and have been placed under project configuration management
- All code walkthroughs have been conducted and action items from the walkthroughs satisfied

3.1.4.3 Documentation

Source code and support files: This includes

- Source code for the product
- Build scripts
- Supporting data files
- Installation scripts
- Test data

Note that software developed in FORTRAN should follow the Air Forces coding guidelines, and software developed in C should follow the APL Space department's coding guidelines for C.

Product Description: Brief summary of the product including the input, output and algorithms used for the product. Many of these descriptions have already been produced.

3.1.4.4 Reviews

Code Walkthrough: This meeting is conducted by the project software lead and should be attended by an independent software developer. During this meeting, the attendees read the source code line by line. Only newly developed source code and existing codes that have undergone substantial modification require walkthroughs.

3.1.5 Product Testing

3.1.5.1 Entry Criteria

- Clean compiling, unit tested software available
- All code walkthroughs complete

3.1.5.2 Exit Criteria

- Successful completion review
- All documentation required for transition is complete

3.1.5.3 Documentation

Test Plan: This is an internal document that can take the form of a memo from the project software lead to the project lead. It should contain:

1. A list of the functions to be tested
2. A list of tests to be conducted, including criteria for pass/fail
3. The data required for each test
4. An outline for the product test report

Product Test Report: The test report describes the final results of testing. The detailed outline of the test report is defined in the Test Plan.

3.1.5.4 Reviews

Completion Review: This is an internal review where the project team describes the software to be transitioned. A representative of the IPT and the transition team should be present. At minimum, the following should be described

1. The basic functions of the product
2. The structure of the software
3. A review of how to operate the software
4. A summary of the testing procedures and results
5. A list of known issues that will effect the transition

At the completion review, the project team must make available to the transition team the following documents:

- Software requirements
- Software design
- Product test plan
- Product test report

3.2 Transition Team Development Process

One cycle of this project occurs for each build. Builds will be delivered to the customer every three to six months.

3.2.1 Transition Planning

During this phase, the transition team identifies the products and product updates that will be contained in the build and allocates resources for build development.

3.2.1.1 Entry Criteria

- At least one product version is available
- Suitable resources are available to the transition team for build planning

3.2.1.2 Exit Criteria

- The build plan accepted is by UPOS management

3.2.1.3 Documentation

Build Plan: This is an internal document that documents the products to be transitioned, the standards to be met for each product and a schedule for the entire build development

3.2.2 Product Integration

During this phase of the transition, the available products are ported to the target environment in a standalone mode. This includes

- running on the correct platform in the required configuration
- accepting input data in the form available at the user's facility
- outputting data in the form required by the user's facility
- meeting, as closely as possible, the users constraints on timing, sizing, stability and maintainability

At the end of this phase, the integrated product will be placed under configuration management.

3.2.2.1 Entry Criteria

- At least one product version is available for integration

3.2.2.2 Exit Criteria

- A baselined version of the product software is available for build integration.

3.2.2.3 Documentation

Transition Plan: For each product to be integrated into the build, a transition plan will be drafted. This plan will describe the tasks required to transition the product to the target environment in a way that meets the above requirements.

3.2.3 Build Integration

During this phase, the transitioned products will be integrated into a single build. This will involve the following:

- Development or extensions of integrated user interfaces (if applicable)
- Database integration for projects that share the same data stores
- Configuration of the products into a single area
- Development of build level installation scripts
- Development of Operators Manual and Version Description

3.2.3.1 Entry Criteria

- Product integration complete

3.2.3.2 Exit Criteria

- A baselined version of the build is ready for testing

3.2.3.3 Documentation

Build requirements: Describes the requirements for the development of the build. This includes

- Software requirements for user interface software and installation scripts
- Constraints on the build that must be considered during integration
- Functions to be documented in the operators manual

Operator's manual: Describes in detail the ways in which the build may be operated. Will contain step by step instructions on the generation of products.

Version description: A brief summary of the build being delivered, including a list of products and their capabilities.

3.2.4 Build Testing

3.2.4.1 Entry Criteria

- A baselined version of the build is ready for testing

3.2.4.2 Exit Criteria

- A successful completion review is conducted
- All build software is on media, ready for delivery

3.2.4.3 Documentation

Build Test Plan: This is an internal document that can take the form of a memo. It should contain:

1. A list of the functions to be tested
2. A list of tests to be conducted, including criteria for pass/fail
3. The data required for each test

Product Test Report: The test report describes the final results of testing. This is a deliverable document, thus it must be a comprehensive summary of the build testing. It must include

1. The functions that were tested
2. The methods used to test the functions

3. A description of each test and its procedures and results
4. A description of the test data used (the data itself will be placed under configuration management).

3.2.4.4 Reviews

Completion Review: At this review, the transition team will present the build to the project leads and at least one representative of the IPT. Demonstrations of the system are strongly encouraged.

3.2.5 Build delivery

3.2.5.1 Entry Criteria

- A successful completion review is conducted
- All build software is on media, ready for delivery

3.2.5.2 Exit Criteria

- Successful acceptance review

3.2.5.3 Documentation

Delivered Build: All software and required data installed at the user's site. Installation test scripts will be run.

Support Plan: A description of how the build will be supported by the transition team. Includes the procedures for corrective action and enhancement requests to be followed by the users and the transition team.

3.2.5.4 Reviews

Acceptance Review: Review to be held with user after the build has been installed at the facility. The review should include:

1. A summary of all deliverable documentation for the build (Functional requirements, Software design, Product description, Test report, Operator's manual, Version description and Support plan)
2. A demonstration of the build

3.3 Transition Development Environment

In order to properly test and maintain the products generated by UPOS, the transition team will maintain a hardware and software development configurations that closely matches the configurations of the users. Two transition development environments will be maintained; one environment that simulates the environment at AFWA and one environment that simulates the environment at AFSWC. Each environment will be described in turn below.

It should be noted that development configuration management will be centralized for both projects. The configuration baselines will be maintained with tools running on the AFWA development system (see below).

3.3.1 AFWA development environment

The UPOS program will deliver hardware platform as well as software products to AFWA. The collection of hardware and software delivered to this facility will be called the UPOS/AFWA Delivered System. The transition team will maintain an identical system at APL called the UPOS/AFWA User Support System. The transition team will maintain the configuration of both of these systems. The User Support system will not be used as a development system. It will be used to reproduce problems and perform customer support with the AFWA users. A third system, the UPOS/AFWA Development System will be maintained as a configuration management and test environment for the transition team. Software releases will be tested

and packaged on the development system and placed on the Delivered and Support systems nearly simultaneously.

The configuration of the delivered and user support systems will be as close as possible to the computing environment at AFWA. This includes similar operating systems, third party software, directory structures and data file formats. The transition team will develop the requirements, design and implementation of these platforms and environments.

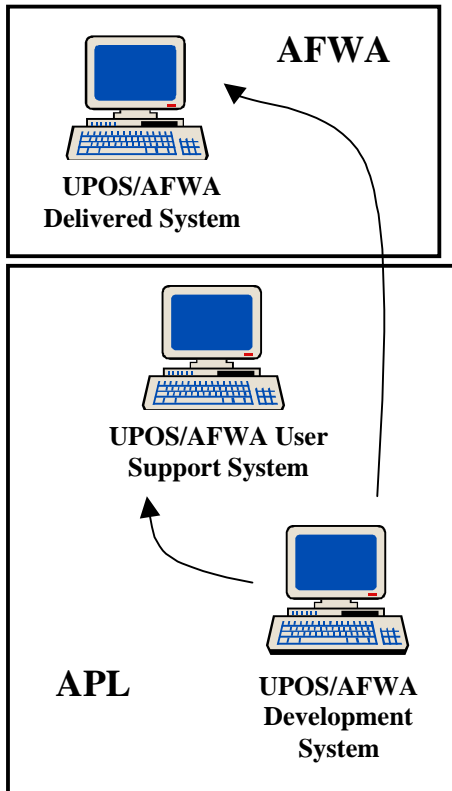


Figure 2. Various components of the UPOS AFWA configuration. The Delivered System and the Support System will be configured identically. Configuration management testing and corrective action will take place on the Development System.

The UPOS/AFWA Development System will be the development environment for the transition team. It will host the software development tools necessary for the construction of the software releases that run on the other two systems. The transition team will develop the requirements, design and implementation of this platform and environment.

3.3.2 AFSWC Environment

The UPOS project will not be delivering a hardware platform to the Air Force Space Weather Center. The transition team will install software at the AFSWC facility. Thus we will maintain two hardware/software environments, the UPOS/AFSWC User Support System and the UPOS/AFSWC Development System. These systems are analogous to their AFWA counterparts. That is, The Support System will contain only the software delivered to AFSWC while the development system will contain all the development tools necessary for the construction of the software delivered to AFSWC. The transition team will develop the requirements, design and implementation of these platforms and environments.

3.4 Configuration management

3.4.1 Project team configuration management

Project teams are responsible for the development configuration management of the software that will be delivered to the transition team. Each project is responsible for defining their own configuration management method, though some suggestions are given here. The essential configuration management responsibility of the project teams is to be able to reproduce, at any time after their software has been delivered to the transition team, to reproduce the delivered configuration. This implies the following

- The project team should be able to identify and produce a copy source code and support files that were submitted to the transition team
- The project team should be able to build and run a version identical to the software submitted for transition
- The project team should be able to account for any changes to individual source code and support files submitted to the transition team between deliveries under the corrective action process below
- The project team should be able to account for all changes to source code and support files made for each new release of product software

It is strongly recommended that the project teams also maintain a development baseline as well, that is, they should keep track of the most current version of the software being developed.

There are several tools available to the project teams to help them with version management. Most notably RCS may be used to track versions of individual files, and CVS for tracking collections of files. RCS is generally supplied with the UNIX operating system and CVS is freeware available from GNU. Both are fairly easy to use for rudimentary version control tasks.

3.4.2 Transition team configuration management

The transition team has the following configuration management responsibilities

- The transition team should be able to identify and produce a copy source code and support files that were submitted to the transition team by the project teams
- The transition team should maintain a development baseline of all software being transitioned. The process of maintaining this baseline is described below
- The transition team be able to identify and produce a copy source code and support files that were delivered to UPOS customer facilities
- The project team should be able to build and run a version identical to the delivered to UPOS customer facilities
- The transition team should be able to account for any changes to individual source code and support files delivered to UPOS customer facilities team between deliveries under the corrective action process below
- The transition team should be able to account for all changes to source code and support files made for each new delivery of the UPOS system

The transition team will use CVS as a tool to manage the various baselines. The directory structure for the UPOS configuration management area is shown in Figure 3. During development, a developer will be designated to manage the configuration for each project. The share area will be managed by developer consensus (the transition team lead will decide matters where no consensus is reached). The entire system for each customer will be baselined prior to integration testing. During integration and system testing, any changes made to any of the areas will be made by developer consensus only. After successful testing, the version of software being tested will be used as the new delivery baseline. The development configuration will be maintained on a single development system. Changes made to the delivery baseline will be managed through the corrective action process.

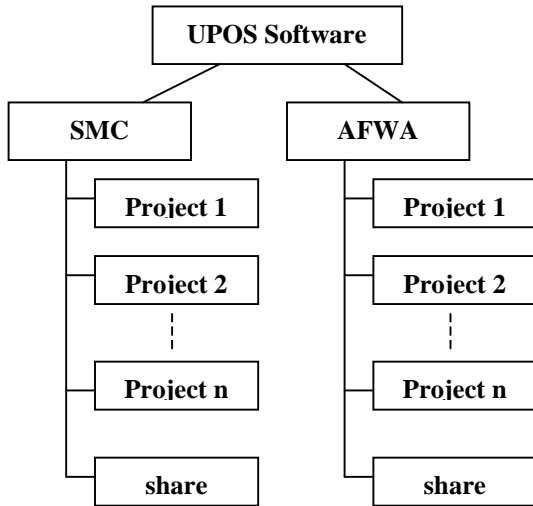


Figure 3. The structure of the UPOS software under transition. Separate areas are maintained for each customer, and for each project. The share area contains software common to all projects and any system level software needed for integration.

3.5 Corrective action

3.5.1 Scope

The corrective action process is used to correct defects to the software that has been delivered to the UPOS customers. It should not be confused with the normal, scheduled development of the UPOS software. It should not be used to add new capabilities or to initiate new development. Its use is limited to correction of software and documentation that is found to be in error. Any member of the UPOS project can initiate corrective action for all software and documentation delivered to the UPOS customers. Internal documentation need not go through this process.

3.5.2 Organization

The corrective action process will be managed by the transition team lead. Because of the small size of the effort, there will be no configuration control board. The transition team lead will, however, seek advice from IPT members regarding resources and priorities and consensus among the developers regarding technical solutions.

3.5.3 Process

The following are the steps of the corrective action process:

- 1) The person that identifies a problem with the UPOS system should contact the transition team lead via email with a Corrective Action Request (CAR). Any member of the UPOS project can submit a CAR. The email message should contain, at minimum
 - The name of the person submitting the CAR
 - The name of the software or documentation that contains the problem
 - The nature of the problem
 - The severity of the problem
 - An email address where status should be mailed

- 2) Upon receipt of the CAR, the transition team lead will choose to submit a Change Order (CO), an Impact Analysis Request (IAR) or choose to do nothing. The transition team lead will advise the requestor of the decision. If a CO or IAR is submitted, a unique number will be issued for the CR. A CO will be submitted only when the consequences of the change are well known. In most cases, an IAR will be submitted. The IAR will be submitted to the appropriate development team member and will contain a copy of the original CAR, its assigned number and the expected date for the completion of the analysis. The contents of the CO are discussed below
- 3) After the IAR is submitted, the developer is responsible for determining the impact and advisability of the change. The results of this analysis should be submitted in writing (via email) to the transition team lead.
- 4) Based on the results of the developer's analysis (or the CAR itself), the transition team lead may initiate a CO. This is a document with a unique number issued by the transition team lead detailing the change that should be made, the schedule for making the change, and identifying a developer to make the change.
- 5) The developer then designs and makes the change. The change is considered complete after the modified software passes regression testing.
- 6) After the change is complete, the transition team lead will update the delivery baseline and arrange for the modified software to be installed at the user's site.

The transition team lead shall maintain tracking information for CARs and COs on a web page available to the UPOS project team. For each numbered CAR this will include

- The number and text of the CAR
- The date submitted
- Whether there is an outstanding IAR, date of IAR
- The text of the analysis resulting from an IAR, date that the analysis was completed
- Whether there is an outstanding CO, its number and the date the CO was closed
- Date Of completed change

For each CO this will include

- The number and text of each CO
- Date submitted
- Date closed