Chapter Twenty Three

THE WORKFLOW PROCESS

Introduction

From the concept to the completion of testing, a new idea is managed. This is true in the SuRPAS environment also. Every idea, new feature, service request, or bug fix is tracked from concept to implementation to testing to inclusion in a version release. This is called the SuRPAS workflow process and this chapter steps through each of the components of the process.

In this chapter we will learn about:

- The SuRPAS Intranet and its access to the elements of the workflow process
- The individuals and groups active in the process
- SDMS and the Workflow Documents
- WebFTK
- The FTK Document
- The Development Estimate Document
- Unit Test
- Code Review

Navigating The SuRPAS Intranet

Overview

The SuRPAS intranet provides BAs, software developers, programmers, and QA testers with a full set of tools for managing the workflow process. Although the majority of this information is accessed from the SuRPAS development home page (Berwyn Manufacturing Department) a number of topics are also available directly from the corporate home page. The information in the next sections discusses how to reach the SuRPAS home page and provides an overview of some of the information available once you have arrived.

Browsing the Intranet

PFPC has its own intranet that provides users access to corporate as well as SuRPAS-specific information. Access to the intranet is accomplished by using an internet browser such as Microsoft's Internet Explorer. Internet Explorer is on your desktop and can be opened by double-clicking on the icon. When the browser is open, enter the address or URL in the Address field at the top of the window. The address to enter is:

`http://surpasintranet/surpas/`

or simply

`surpasintranet/surpas`

When you press the enter key or click the GO button the following screen should appear.

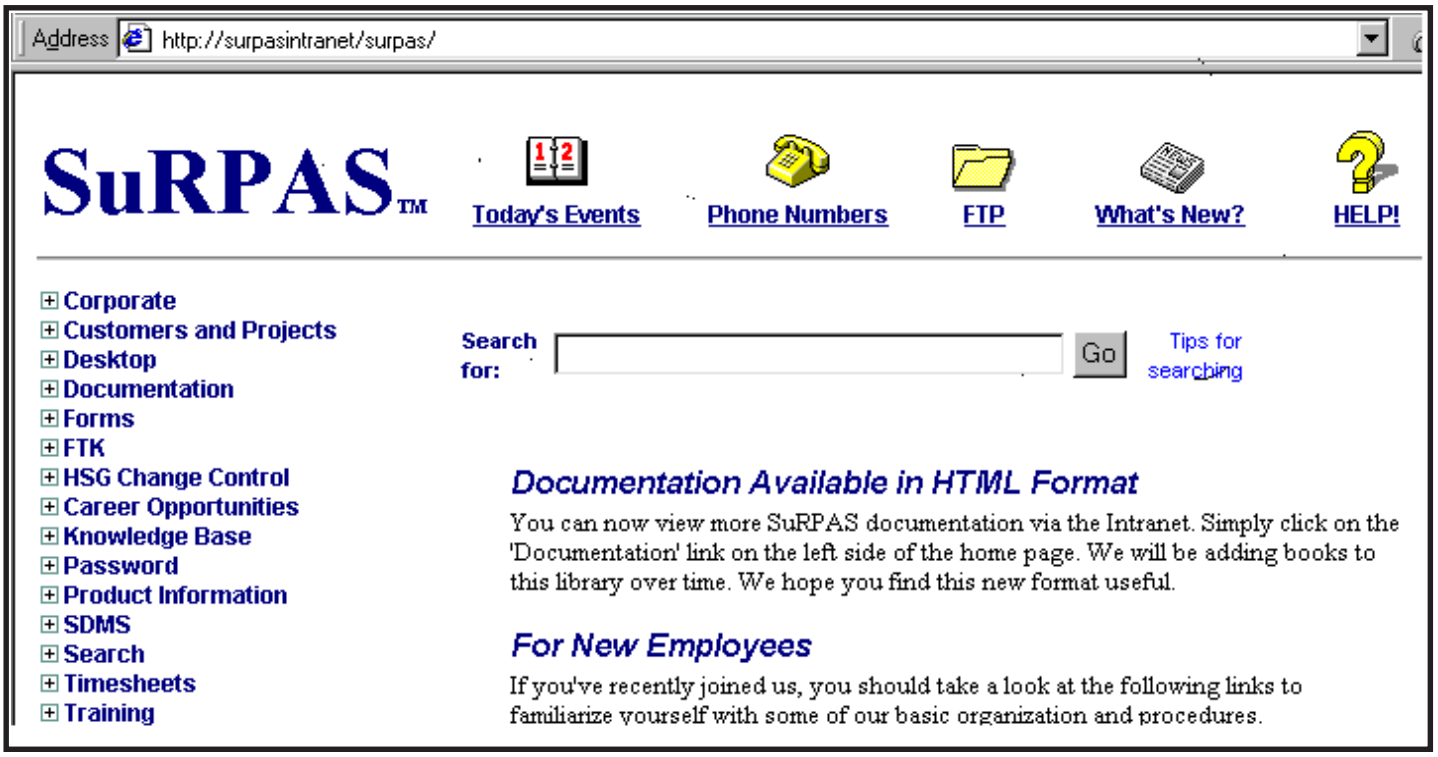

Figure 23-1 - SuRPAS Intranet Home Page

Phone Numbers and FTP - Home Page Options

At the top of the home page there is an icon for Phone Numbers and one for FTP. Use the Phone Numbers icon to get phone numbers and TLAs. Use the FTP icon to get to the Training folder. In the training folder you will find Adobe Acrobat PDF files for each chapter in this student guide. You can then copy and/or print the latest version and updates for this manual as well as some of the PowerPoint presentations used.

Figure 23-2
SuRPAS Intranet Corporate Sidebar

Drilling Down to the Manufacturing Home page

On the side bar of the SuRPAS home page is a list of topics available. The first entry is Corporate. By clicking on Corporate you will gain access to drop-down list of corporate entries.

The last entry is SuRPAS Departments. By clicking this one you will gain access to the SuRPAS-specific departments. An example of this is shown to the left of this paragraph.

The Manufacturing department entry points to the SuRPAS (Berwyn) home page. Click on this entry to bring up the Manufacturing Home Page. A sample copy of this home page is shown below.

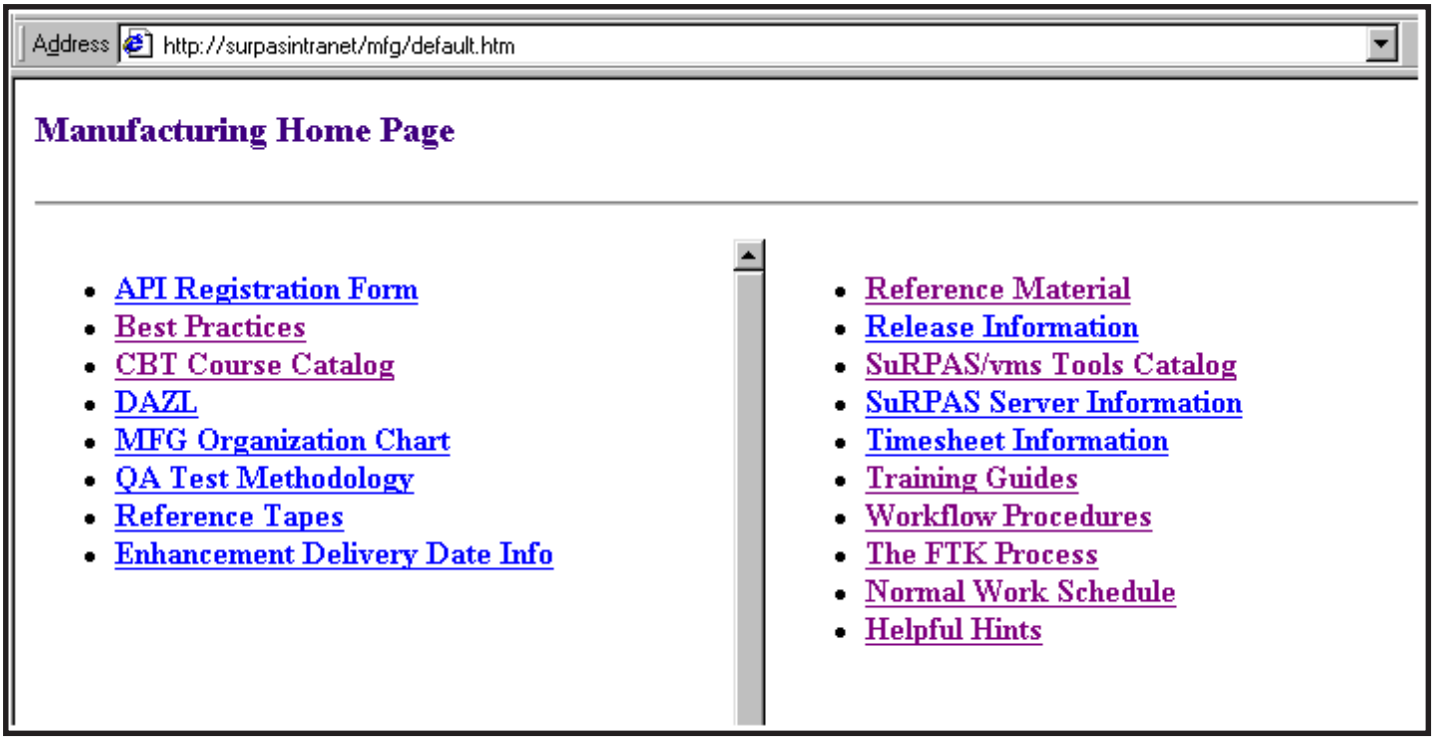

Figure 23-3 - SuRPAS Intranet Manufacturing Home Page

Best Practices

The Best Practices entry on the Manufacturing Home Page provides you with access to Best Practices documents. These documents cover topics related to the OpenVMS operating system, the Fortran programming language, SuRPAS-specific topics, and general Manufacturing department topics. Any of these documents can be selected (check the box to the left of the document title) and then emailed to an address of your choice. You are not limited to one document per email. The documents are in Microsoft Word format.

SuRPAS/VMS Tools Directory

The SuRPAS/VMS Tools Directory contains a list of tools developed internally to support both SuRPAS-specific and general OpenVMS needs. The list is ordered alphabetically and contains a description as well as revision information for each tool. The tools can usually be found in the FAL$TOOLS directory.

Workflow Procedures

The Workflow Procedures entry contains documents and templates pertaining to the SuRPAS Workflow Process. It includes both documents that describe the process as well as templates and poitners to templates for the various support documents, including:

- SuRPAS Workflow
- TAC Procedure
- REQSPEC How To
- Requirements Template
- Spec Template
- Code Review Process

The FTK Process

The FTP Process entry contains documents and templates specific to FTKs and the FTK process. This entry discusses each step in the FTK process and the documents and players that are integral to the process.

The SuRPAS Workflow Diagram and the Players

Overview

The SuRPAS Workflow Diagram shows the steps followed by SuRPAS players when taking a new SuRPAS feature from an idea in a customer or employee's mind through the completion and delivery to SuRPAS customers in a future version of the SuRPAS product. The sections that follow describe each of the steps in the workflow diagram and the players that make this happen.

This section discusses:

- The Workflow Diagram
- The Steps of the Process
- The Players

The Workflow Diagram

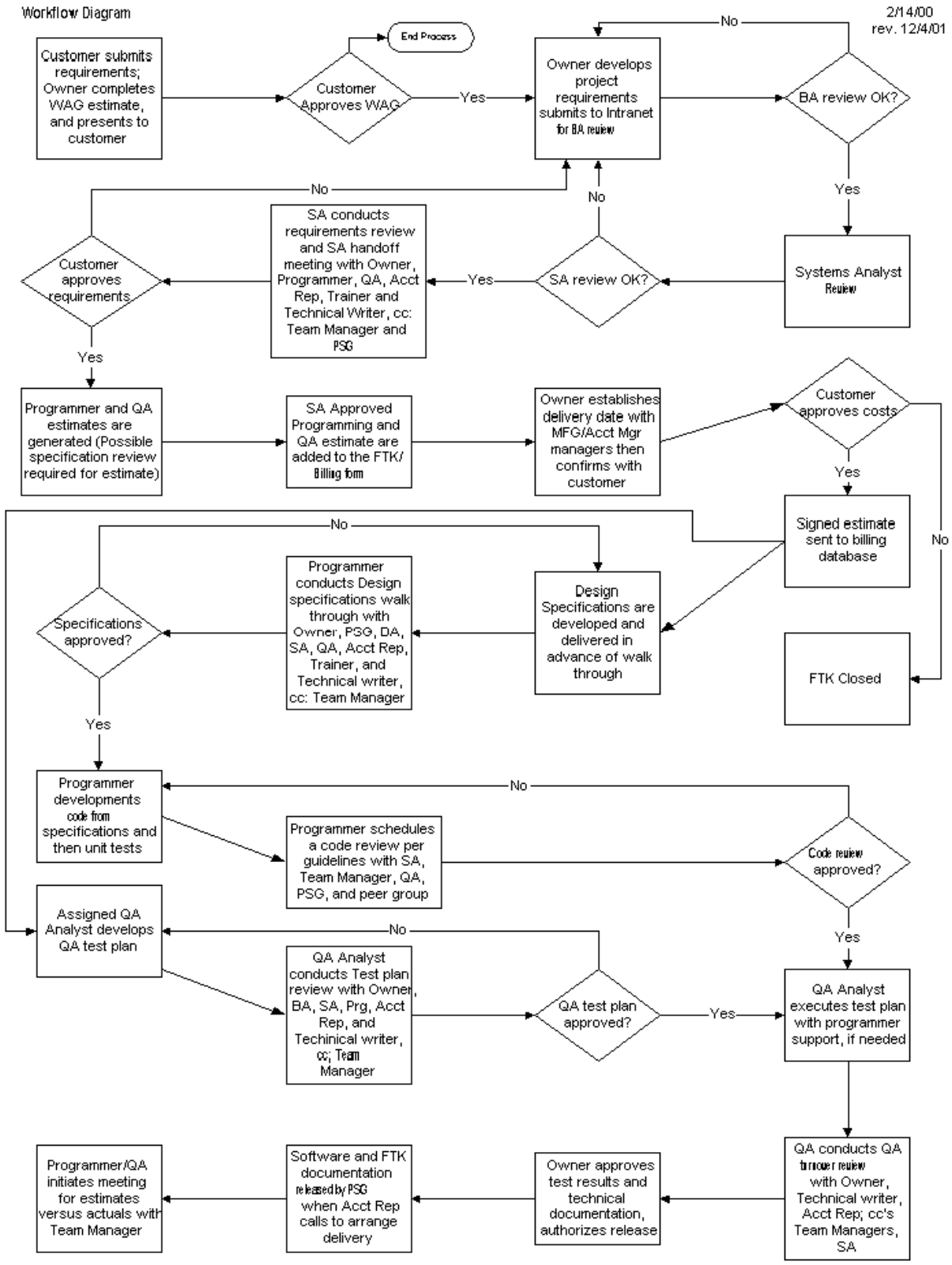

Diagram 23-4 - The SuRPAS Workflow Diagram

The Steps of the Process

Most SuRPAS projects fall into one of three cost categories: billable, non-billable, and dedicated; and one of two project categories: development and production. Depending on the billing and project category the steps may take on a different meaning, or may be skipped altogether. The following is a discussion of each step. Each step includes a code indicating the cost category of project that step applies to (**b**illable, **n**on-billable, **p**roduction). The steps parallel the flow found in the Workflow Diagram on the previous page. The discussion found in the next section, entitled The Players, will help provide an understanding of the roles that the various players take responsibility for.

1a. Requirements Submission (b,n)

On development projects the customer submits their requirements to the Customer Service Representative (CSR). Larger projects are routed by the CSR to the Business Analyst (BA) manager who then assigns the project to a specific BA. Development projects can be billable or non-billable projects. Requirements for non-billable projects can come from many and varied sources.

1b. Production Problem Report (p)

Production projects originate as the result of a problem reported by a client to a Customer Service Representative (CSR). Usually, these projects need immediate attention and involve a development effort of eight hours or less. The CSR reviews the project with the CSR Manager to determine if BA and/or SA involvement is necessary.

2. New FTK Created (b,n,p)

The CSR (p) or owner (b,n) opens a new FTK making sure to fill in all of the required fields. FTKs must be opened using WebFTK to ensure that all required fields are filled.

Non-billable and Production Projects skip steps 3 through 5.

3. WAG Creation (b)

The Owner creates a WAG estimate using the WAG template. The WAG is checked in to The Software Document Management System (SDMS) and them updates the WAG points on the billing form.

4. WAG Pricing (b)

The Owner gives the WAG point amount to the Sales group. A WAG dollar amount or range is provided to the client.

5. WAG Customer Approval (b)

The customer approves the WAG amount and notifies the Sales group. The Sales group then notifies the Owner. If the customer rejects the WAG amount, the process ends and the FTK is closed.

6. Requirements Document Creation (b,n,p)

The BA (b,n) creates a requirements document using the combined Requirements/ Design template, and checks it in to SDMS. The Owner assigns Programmer 1 and Tester 1 to "MFG" (the Manufacturing group), and sets the Documentor field to "DOC". If the SA on the project is known, the Owner will use that SA's initials (or TLA). Otherwise the SA is also assigned to "MFG".

If this is a production project a formal requirements document is not usually created. The CSR/Owner determines the requirements in detail and enters them as FTK notes in WebFTK.

7a. BA Group Requirements Review (b,n,p)

For development projects the Business Analyst (BA) group reviews the requirements document. If changes to the requirements are necessary, the BA checks out the requirements document from SDMS, updates it, and then checks it back in. Once approved, the BA will update the BA Requirement Approval Date field.

For production projects, if a BA is participating, he/she will review the requirements notes and, when approved, will update the BA Requirement Approval Date field.

7b. Requirements Reviewed by Client (b,n,p)

For client-specific projects, the requirements are sent to the client for review.

8. SA Group Requirements Review (b,n,p)

For development projects, the Systems Analyst (SA) group reviews the requirements generated by the Owner. The SA assigned to the project checks the combined requirements document out from SDMS, updates the document with SA review input and notes, and then checks the document back in. Once the document has been approved, the SA will update the SA Requirement Approved Date field.

For production projects, if an SA is participating the systems analyst will review the requirements, and once approved will update the SA Requirement Approved Date field. Production projects skip step 9 and go to step 10.

9. Software Version (b,n)

The Owner and Manufacturing management agree on the software version where the project will be developed. The Owner then updates the Software Version and Release Tape fields in the FTK.

10. Management Review (b,n,p)

For development projects, Manufacturing management reviews the FTK and assigns a Manufacturing contact, the primary programmer, and the primary QA tester. Additional programmers and testers may also be added.

For production projects, if the turnaround time to the customer is 3 days or less, the CSR/Owner assigns the account's TSR to the project as the primary programmer. For production items due to the customer in more than 3 days, the primary programmer assignment is the help desk coordinator. If Manufacturing is needed to test the FTK, "MFG" is assigned as the primary tester. The tester can also be the CSR and in some cases, "N/A" will be used to denote that no testing is required. If documentation is required, "DOC" is assigned to the documentor. "N/A" will be used to denote that no documentation is required. Production projects skip to step 20.

11. Programmer Requirements Review (b,n)

The primary programmer assigned to the project schedules a requirements review meeting.

For development projects the programmer(s) and QA tester(s) will begin working on their design specification and test plan, immediately following the requirements review.

12. Programmer Development Estimate (b,n)

The Programmer creates an estimate using the most recent version of the Development Estimate Template (referred to as DEV).

13. QA Test Estimate (b,n)

The QA Tester creates an estimate using the most recent version of the QA Estimate Template.

14. SA Development and QA Estimate Review (b,n)

The Systems Analyst reviews the DEV and the QA estimate. Once approved, the SA notifies the Programmer and Tester that the estimates are acceptable.

15. Development Estimate Placed in SDMS (b,n)

The Programmer enters the unloaded hours on the FTK Billing Form. The programmer then checks the estimate document in to SDMS.

16. Test Estimate Placed in SDMS (b,n)

The Tester enters the unloaded hours on the FTK Billing Form. The tester then checks the estimate document in to SDMS.

17. Sales and Billing Form Updated (b)

For billable projects only, the Owner sends the estimated hours to the Sales group which then calculates the Quoted Amount. The Owner updates the Quoted Amount field on the Billing form.

18. Customer Approval of Requirements and Amount (b,n)

For billable projects the customer must approve the requirements and the quoted amount for the project. The Owner then notifies the billing administrator via email and provides the approved dollar amount and any customer approval comments. A letter or email directly from the client is required. The owner then updates the Customer Approved Date. If the customer rejects the requirements and quoted amount, the FTK is closed.

For non-billable customer-specific projects, the customer approves the requirements and the Owner updates the Customer Approved Date.

19. Billing Database Updated (b)

For customer-specific, billable projects only, the billing administrator updates the billing database.

20. Owner Finalizes Delivery Date (b,n,p)

For development projects, the Owner finalizes the delivery date with Manufacturing management. Manufacturing management then updates the QA due dates and release-related fields in the FTK. The Priority Code is now changed to “Active”.

For production projects, the CSR/Owner establishes the delivery date in conjunction with the Programming Support Group (PSG) programmer.

21. Owner Updates Customer Due Date (b,n,p)

The Owner updates the Customer Due Date. Production projects skip steps 22 and 23.

22. Programmer Updates ReqSpec (b,n)

The Programmer checks out the combined Requirements/Specification document (often referred to as the ReqSpec) from SDMS, adds the design specifications to the document and then checks it back in.

23. Design Review (b,n)

The primary Programmer sets up a meeting to review the proposed design. Any changes to the design must be reflected in the ReqSpec document. To change the ReqSpec document, the programmer checks it out from SDMS, updates it, and then checks it back in.

24. Development and Unit Test (b,n,p)

The Programmer or programmers involved in the project, develop and then unit test the coded solution. Production projects skip steps 25 and 26.

25. Tester Defines Test Plan (b,n)

The Tester defines the Test Plan for the project and checks the document in to SDMS.

26. Test Plan Review (b,n)

The primary QA Tester sets up a meeting to review the test plan. Any changes to the test plan must be reflected in the document. To change the document, the tester checks it out from SDMS, updates it, and then checks it back in.

27. Unit Test Review and Code Review (b,n,p)

For development projects, the Programmer sets up a meeting for Unit Test Review and Code Review. Interested parties and specified inspectors are invited. Code updates are made as necessary. Upon approval of the review group, the code and unit test results are turned over to the QA Tester.

For production projects, the programmer turns over the software and unit test results to the QA tester.

28. Development Finished (b,n,p)

Development is now complete. The Programmer sets the End Programming values.

29. Test Plan Executed (b,n,p)

For development projects, the QA Tester (s) execute the Test Plan and produce documentable results.

For production projects, the tester executes the test as per the Test Plan or FTK test notes. Alternatively, the Test At Client (TAC) process can be used if authorized by the Manufacturing Production Coordinator.

30. Test Results Review (b,n)

The primary QA Tester schedules a meeting with the Owner and other interested parties to review the test results.

31. Testing Finished (b,n,p)

Testing is finished. The QA Tester sets the End Test values in the FTK.

32. Owner Test Results Review (b,n,p)

For development projects, the Owner reviews the project test results and finishes the project by setting the Finished Date in the FTK.

For production projects, the CSR/Owner reviews the project test results and finishes the project by setting the Finished Date in the FTK.

33. Software Packaging (b,n,p)

For billable and non-billable projects, the Programmer packages the software, if necessary. As a general rule, on-release software deliveries do not need to be packaged separately from the release tape.

For production projects, the Programmer packages the software. Production projects are typically off-release and as such must be packaged individually.

34. FTK Released and Delivered (b,n,p)

PSG releases the FTK and delivers the software.

The Players

The players in the workflow process are those people who have a role in making the process successful. The workflow process defines the activities surrounding the addition of a feature or capability into SuRPAS. In some cases these features or capabilities are built as the result of a customer request (which may or may not be billable), in other cases they may be generated either from the customer advisory board or as the result of an internal decision. In either case, if the requested feature or capability makes it through the original requirements and financial feasibility stages, an FTK (FAL Tracking System) number will be assigned and an FTK will be created. In the role discussions below references to service requests, features, capabilities, or projects ultimately result in the building and processing of an FTK to completion or cancellation.

The Customer

The Customer is the SuRPAS client or end-user of the SuRPAS product. The customer originates billable projects by submitting a set of requirements that define a desired service request or feature. If the project is billable the customer must sign off at every step along the workflow process.

The Sales Representative

The Sales or Account Representative is a member of the SuRPAS marketing department and is usually the person who originally sold SuRPAS to the customer and acts as a client interface. Although the sales representative remains in contact with the customer he/she does not usually get actively involved in the workflow process except to review requirements, test plans, and test plan results. The Account Representative may also coordinate delivery schedule, on-site installation, and acceptance testing.

The Customer Service Representative (CSR)

The Customer Service Representative (CSR) is the day to day liaison with the SuRPAS client. Any bug fixes, service requests, or new SuRPAS requirements are submitted to the CSR for handling. It is the CSR that routes all customer-originated requests to the Business Analyst Group.

The Owner

The term Owner refers to the person who takes responsibility for a specific SuRPAS project (FTK) and who has the longest relationship with the project. It is the owner that initiates the project, creates the FTK (assigns an FTK number) and signs off on the FTK when it is complete. The Owner is usually the BA or Account Representative, but can be an Installation team member, a Customer Service Representative (CSR), or a System Analyst (SA). For internal projects the owner is the SuRPAS employee who sponsors the modification or enhancement. The Owner's responsibilities include, but are not limited to:

- A full understanding of the project requirements. This includes: what the desired results are and how they are to be used; the frequency of use; the volume of data; the ramifications on the functionality within the system; the alternative methods, if any, of obtaining these same results; and the value of the results to the user (or client).

- The preparation for the complete documentation of the requirements. This includes: report layouts, screen layouts, detailed processing scenario examples, computations, sort orders, data sources, flowchart of the process, interface layouts, API's, the source of data, etc. The objective is to provide sufficient detail with such clarity that it could be program, tested, and documented, meeting the requirements of the project.

- Outlining of the business reasons for the project. How is the enhancement used? What business needs does it fulfill? How does it fit into the existing client workflow? If there are new events, when they are to be run in relationship to other events in the nightly processing. If there are new reports, how are they used? If the reports show error conditions, how do those conditions get resolved?

Depending on the scope or nature of the project, the development of these requirements may be done with the support and/or collaboration with one or more people, including BAs, SAs, and clients. Having documented the requirements, the owner's continuing role is to keep the project in conformity with these requirements. Any questions raised by other team members about the requirements are promptly addressed by the Owner. Owner activities are listed at the top of the next page.

Owner activities include:

- WAG Development
- Customer Requirements Development
- Provide Requirements via SDMS for BA and SA review
- Get customer approval on requirements and cost
- Manage the FTK system and keep it current
- Attend all project design walk throughs, reviews, and turnovers
- Attend all Quality Assurance (QA) walk throughs, reviews, and turnovers
- Review test results and documentation, and authorize project release (FIN the FTK)

The Business Analyst (BA)

The Business Analyst (BA), in most cases, acts as the Owner of the project. The Business Analyst provides both the business requirements and justification for all service requests, client-generated SuRPAS features, and internally generated enhancements. The Business Analyst Group Manager assigns all projects (FTKs) to a specific BA, who will manage the business requirements portion of that project. The BA provides requirements sections of the FTK document as well as portions of the Requirements Specification. Other roles of the Business Analyst include:

- Interface with or act as the Owner (starting from requirements development through the end of the project).
- Review and approve requirements
- Attend Requirements Review and SA handoff with the SA, Owner, Programmer, QA Tester, Account Representative, Trainer, and Technical Writer.
- Attend design specification walk through with the Owner, Programmer, SA, Account Representative, QA Tester, Trainer, and Technical Writer.
- Attend QA test plan review with the Owner, Technical Writer, Account Representative, Programmer, and QA.
- Review and approve all FTK documentation.

The Team Leader

The Team Leader has employment/human resource responsibility for the programmers and quality assurance testers. Although the team leader may not be a part of the technical implementation of the project's workflow, he/she is responsible for assessing the quality of work done by the programmers and QA testers that are a part of his/her team. The team leader provides training for new employees through mentoring and allowing these new employees to attend the various reviews (BA review; SA review, code review, QA review, etc.).

The Systems Analyst (SA)

The Systems Analyst oversees the implementation of the project requirements. It is the SA's responsibility to provide top-level design of the project as a part of the FTK, and to work with the programmer(s) and quality assurance (QA) tester(s) in implementing and testing the design. The SA is also responsible for the code review and is a member of the QA review. The SA provides portions of the Requirements Specification and the Design Estimate documents (both FTK documents managed by SDMS). The Systems Analyst's role includes:

- Interface with the Owner (starting with requirements development through project end).
- Review and approve requirements before Programmers and QA testers provide development and QA estimates.
- Use SDMS to update FTK specification and design sections, and technical notes, as appropriate.
- Conduct Requirements review and SA handoff with the Owner, BA, Programmer, QA tester, Account Representative, Trainer, and Technical Writer.
- Review Programmer and QA estimates through SDMS.
- Assist with functional/programming/design specifications, as required.
- Review code, attend code reviews, ensure adherence to standards and requirements
- Review test plans, ensure correct testing occurs at the unit test level.
- Attend walk throughs and reviews where needed.

Technical Lead

The role of the Technical Lead is to assist the programmers during all phases of the project. The Technical Lead works with the programmers at a more detailed level than the Systems Analyst. Projects vary in terms of number/size of modifications and complexity. Programmers on the project team also have a wide range of experience. The Technical Lead role varies to fit the situation. In smaller projects the role of the Technical Lead may be assumed by the Programmer and/or Systems Analyst. The following are some roles assumed by a technical lead:

- Attend requirements reviews and SA hand-offs
- Assist in project task assignments among programmers
- Review development and QA estimates before submission for SA review.
- Attend all specification and code reviews.
- Support Programmers during the coding and unit test phases
- Assist programmers meeting schedules and report schedule risks.
- Review QA test plan and support QA testing needs

The Programmer

The Programmer implements the project top-level design both in the FTK and in code. The programmer is also responsible for unit testing his/her code and gaining approval for the design of the code at the code review. The programmer provides portions of the Requirements Specification and Design Estimate. The Programmer's role includes:

- Schedule and attend Requirements review and SA hand-off with the Owner, BA, SA, QA Analyst, Account Representative, Trainer, and Technical Writer (and notify his/her Team Leader)
- Develop a programming Development Estimate
- Create/update design specification using SDMS
- Conduct design specifications walk through with Owner, BA, SA, Account Representative, Trainer, Technical Writer, and QA Analyst (notify Team Leader and PSG)
- Build program requirements and unit test plan, and then implement both.
- Conduct a Code Review process and review unit test results with SA, QA and peer group (notify Team Leader)
- Attend QA test plan review
- Provide testing tools for use by QA Tester, as needed.
- Create software installation package, if required.
- Conduct an Estimates versus Actuals analysis and review meeting with Team Leader.

The Quality Assurance (QA) Tester

The Quality Assurance (QA) Tester writes and executes the project test plan. Quality Assurance's task is to apply a sufficient variety of testing scenarios to ensure the programming correctness under all practical conditions and to ensure that nothing else in the system was incorrectly effected. The QA tester is also responsible for gaining QA approval for the test results. The QA tester's role is:

- Attend requirements reviews and SA handoff
- Develop a QA estimate for testing
- Attend the design specifications walk through
- Assist the Programmer with data and database setup, when needed
- Attend Code Review and receive/review Programmer's unit test results
- Develop test plan for QA testing and then schedule a QA test plan review.
- Conduct a QA test plan review and receive Owner approval prior to testing.
- Code handoff from Programmer and move software to appropriate tree.
- Supply any Regression Test Notes in FTK Technical Notes.
- Conduct Estimates Versus Actuals Review Meeting with Team Leader.

Technical Writer

The Technical Writer role is to gather information from all team members and develop technical documentation of the project. Responsibilities of the Technical Writer include:

- Attend all Requirements reviews
- Attend all Design specifications walk throughs
- Attend the QA test plan review
- Provide complete documentation of the new FTK requirement
- Attend the QA turnover meeting with the Owner, Account Representative and the QA tester.

Trainer

The Trainer's role is to gather and understand information from all team members and develop a training program for Account Representatives and, if required, the clients. The responsibilities of the Trainer include:

- Attend all Requirements reviews and SA handoffs
- Attend Design specifications walk through
- Develop a training program for Account Representatives and Customers.

SDMS and the Workflow Documents

Overview

The four documents that support the workflow process are the WAG, the Requirements Specification, the Development Estimate, and the QA Estimate. These documents are managed by an intranet enabled application called the SuRPAS Document Management System or SDMS.

This section discusses:

- The SDMS process flow
- The SDMS Interested Parties
- SDMS Usage
- The WAG and the QA Estimate

SDMS

The SuRPAS Document Management System (SDMS) is an intranet enabled application that provides the framework for users to collaborate on FTK project-related documents. SDMS provides restricted access to a centralized repository for documents. In our case, the documents involved are only those related to an FTK. Once a document has been stored in this repository, ongoing documentation modifications are restricted to one user at a time. At all times, the documents stored in the repository are available for viewing and/or retrieval. Users who need to modify a document will "check out" the document. This effectively locks out changes from any other user until the modifications to the document have been completed. At this point the user "checks in" the document. This removes the lock from the document and stores the updated version in the document repository.

It is important to note that SDMS replaces the Web submittal process for FTK-related document. Non-FTK-related documents are still submitted using the old Submittal Form or its new browser enabled twin, the Non-FTK Submittal Form.

The SDMS Process Flow

The SDMS process flow is basically the same for each type of document supported. The Requirements Specification, Development Estimate, and QA Estimate all follow this process.

Document Creation and Check-in

The author of a managed document creates the document by providing an eight character name. The name is made up of three characters indicating the type of document it is (e.g. REQ for Requirement Specification), followed by the five character FTK number.

When the author has completed the document, it is checked in to SDMS as a "NEW" document with its appropriate class (in the case of a ReqSpec the class is requirements). SDMS will require the author to add appropriate comments and, if the author knows which projects the FTK belongs to, they can be added as well.

There is a list of ***interested parties*** defined in each FTK. Once the document is successfully checked in, it is immediately available to all interested parties by way of the "GET" or Checkout functionality of SDMS. Each time a document is successfully checked in, all interested parties are notified by email.

Document Modification

When another party needs to update the document, that person can use the Check Out capability in SDMS to lock the document and retrieve the latest version of the document by way of email. At this point, everyone can still retrieve the document for viewing or printing, but only the person who checked the document out of SDMS can make modifications to it and then update the version in the SDMS repository. Document retrieval by other interested parties can be accomplished either by use of the "GET" option in SDMS or through the FTK Document Lookup utility.

All changes to the document must be made to the version of the document emailed by SDMS to the person who checked out the document. This guarantees than no other changes are lost.

When the changes are complete, the modifier checks the document back in to SDMS but does not designate the document as "NEW". For existing documents like this modification, the previous version is archived. Archived copies can then be retrieved using the Get Archived Document option. The successful Check In unlocks the document and it can now be checked out by other users, as necessary.

SDMS Interested Parties

There is a list of interested parties built into each FTK. These persons receive an email notice when a significant event occurs in the life cycle of an FTK. Among these significant events is the Checking In of a document (new or modified) to the SDMS document repository. The Interested Parties list is obtained from the entries in the following FTK fields:

- Owner
- Approved By
- Programmers (1 through 3)
- Testers (1 through 3)
- Documentor
- Business Analyst
- Systems Analyst
- MFG Contact
- User Who FINs the FTK

In addition, anyone who has registered to receive emails based on the document class will also be part of the Interested Parties list. The Document class can be Requirements, Dev Estimates, QA Estimates, Test Plans, or WAGs. To register to receive email for a particular document class, send an email to the WebMaster, indicating the class of documents that you want to be notified about.

SDMS Usage

SDMS Login

SDMS uses the same Three-Letter Acronym (TLA) that you use for other Intranet applications including WebFTK and the Web Timesheet. If you have not previously used your intranet password, you will need to change the assigned password to something else the first time through. Ongoing password maintenance is handled by the Timesheets - Password Maint Link from the SuRPAS Home Page.

When you enter SDMS from the SuRPAS home page or from within the WebFTK application the SDMS Login Screen will appear, as seen below in Figure 23-5 and you will be required to enter your user TLA (three letter acronym) and the appropriate password. Press the Go button to proceed to the SDMS Main Menu.

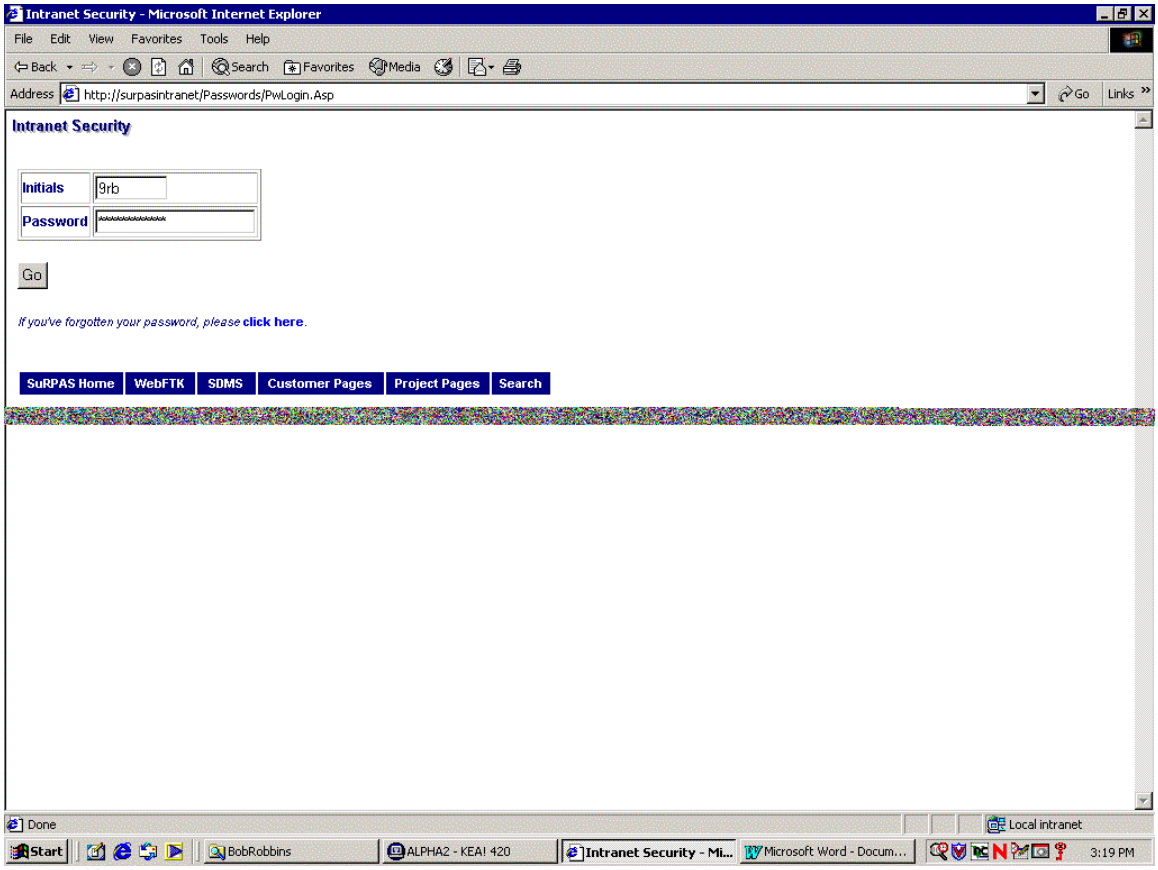

Figure 23-5 - SDMS Login Screen

The SDMS Main Menu

Once you have logged in correctly, you will be directed to the SDMS Main Menu. On this screen, as shown below in Figure 23-6, you can enter your FTK number and then select the appropriate menu choice from the buttons on the left side of the screen. Pressing the ENTER key after typing your FTK number will take you directly to the Get Documents screen. If you are using SDMS to access non-project related documents, you may select the appropriate menu choice from the buttons on the left side of the screen without providing an FTK number.

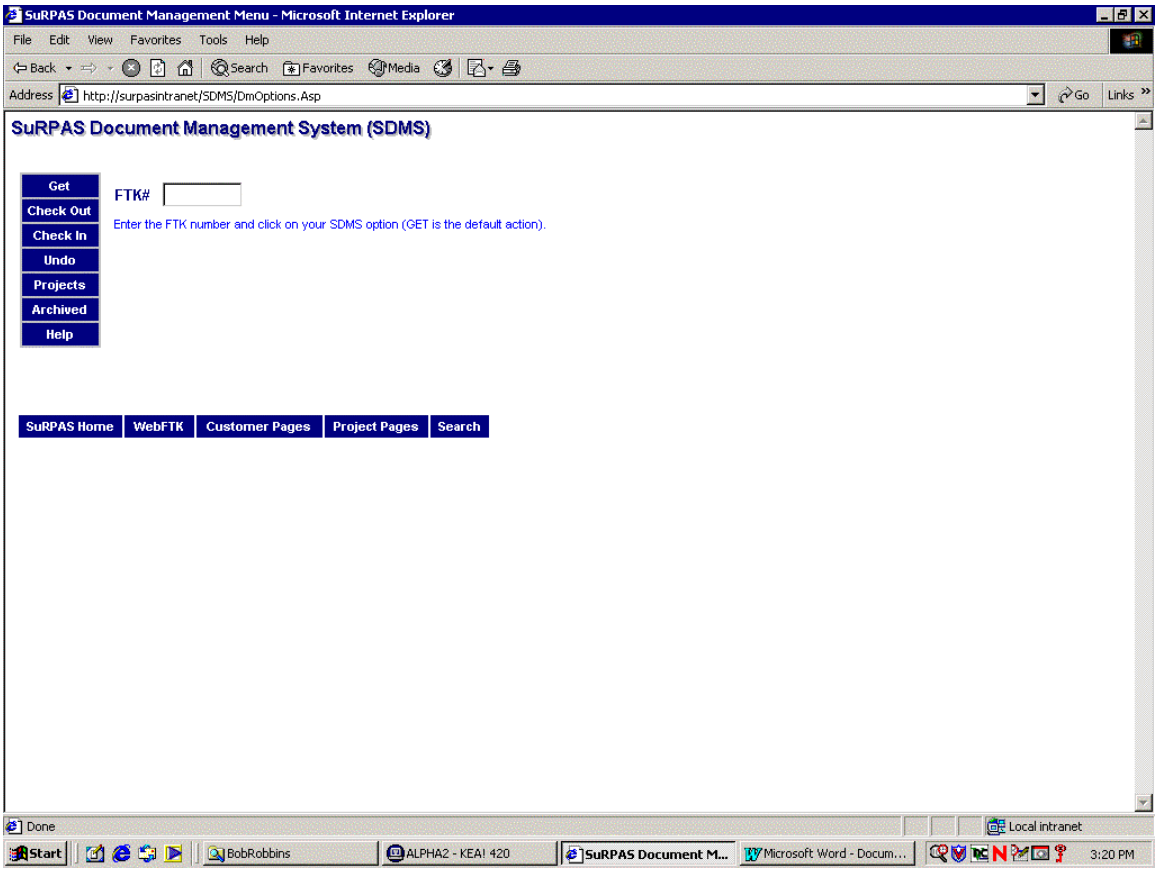

Figure 23-6 - SDMS Main Menu Screen

GETting Documents

The GET Documents screen as seen below in Figure 23-7, is used to retrieve a copy of FTK related documents by email. Select the documents that you would like to retrieve by clicking on the checkbox at the left of each document listed. When you have selected all of the documents desired, click on the GO button to process the form and email the documents back to you.

The Get Documents Screen will expand to support the emailing of documents. Two additional fields will be displayed to allow you to input your TLA and any other TLAs (cc:) you wish to copy (send a copy of these documents to). Notice that unlike other intranet functions that allow you to supply external email addresses, the SDMS function only allows the emailing of SDMS documents to the default email address supported by the TLA (usually the <*user.name*>@PFPC.COM).

The document that you receive is only a copy and other users will not be prohibited from checking the document out.

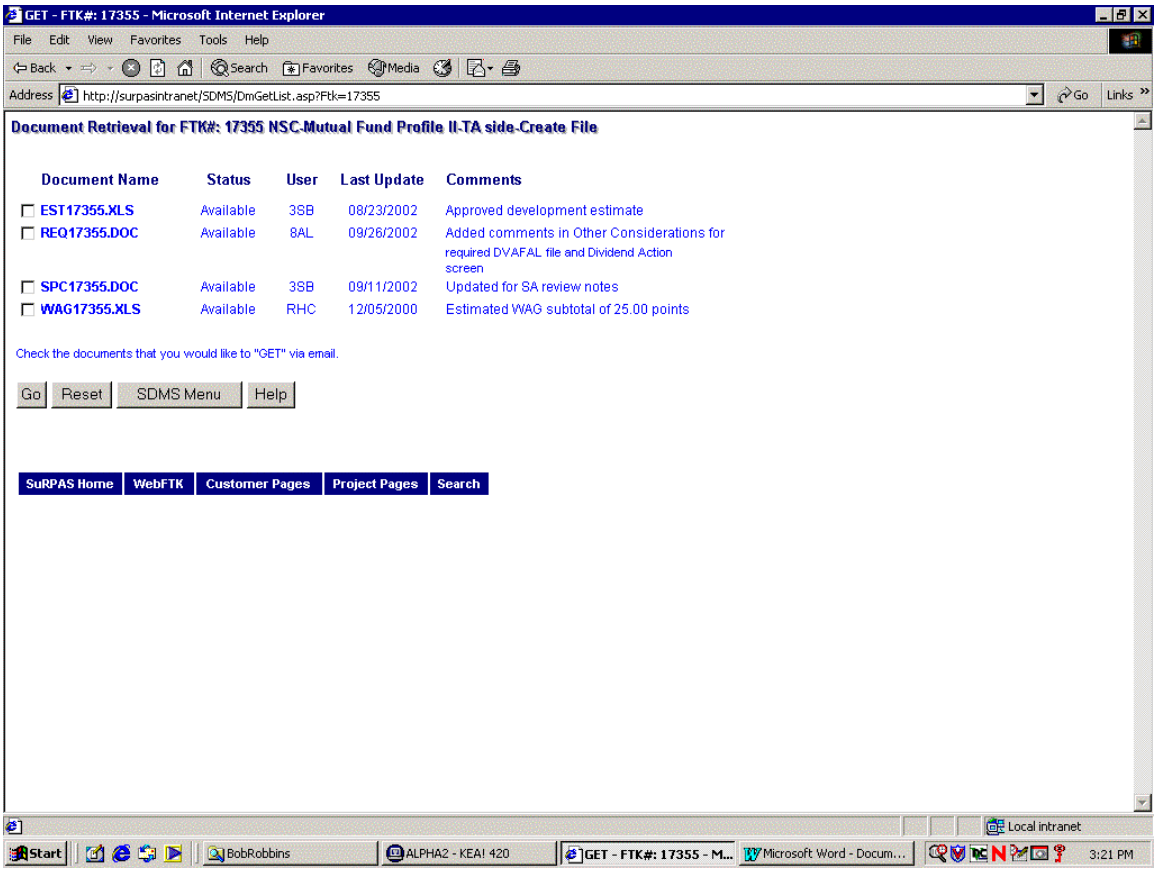

Figure 23-7 - SDMS Get Documents Screen

Check Out Documents

The Check Out Screen is used to lock a document for the purpose of updating the document. Select the documents that you would like to check out by clicking on the checkbox at the left of each document. If the checkbox is grayed out, it means that the document is already checked out.

Check out comments are required as a part of the Check Out process. You have a minimum of 5 lines and 255 characters. Longer comments will be truncated. Once you have completed and processed the form correctly, the document will be mailed to you. Save the attachment from the email to the hard drive on your PC. Once the document is in a folder on your PC, you can work with the document as you would any other document. Remember to check the document back in when you've completed your modifications. If you decide that you really didn't need to modify the document, use the Undo Screen to unlock the document.

Check In Documents

The Check In Screen is used to add or replace documents in the secure SuRPAS document repository. Always use the browse button on the form to select the local copy of your document that you want to check in. The document selection control dialog does not allow the designation of a default origin folder. It will always default to your Desktop the first time through. If you have a favorite location the you use for documents (such as C:\My Documents), consider creating a shortcut on your Desktop that points to this folder. After you have selected the document, check New if this is the first time the document is being checked in. If you're not sure, use the Check Out Screen or the Get Screen to see what's currently available for the FTK in question.

Select the document class. The class must match the naming convention of your selected document. For instance, if your document is named TSTxxxxx.doc and the document class is Requirements, you will not be allowed to check in your document. If you know which project page that the FTK and it's related documents should be displayed on, select the entry or entries in the Projects list. To select multiple entries, hold down the Ctrl key as you click on your selections. Your Project selections are added to the list of Projects and the lack of a Project selection will not remove it from the existing Project list for the FTK. Use the Project screen to remove any unwanted choices. Last, enter appropriate comments. You have a maximum of 5 lines and 255 characters. Longer comments will be truncated. Click the Go button to Check In your document and unlock it for future modifications. An email will be sent to Interested Parties when Check In has been successfully completed.

Undo Document Check In

The Undo Check In Screen is used to remove the lock on a particular document. Use this screen if you have a document checked out that you no longer want to update (and have made no modifications to the existing document in the repository). Select the documents that you would like to undo by clicking on the checkbox to the left of each document. If the checkbox is already grayed out, there is nothing to undo for that document. When you have completed your selections, click the Go button to unlock the document (s).

Get Archived Documents

The Get Archived Screen allows you to retrieve a copy of previously archived versions of documents. The documents that you receive cannot be checked back in. These documents are for retrieval only. Along with the document name, you will see the date, time, and user who created the archived version displayed. Select the document that you would like to retrieve by clicking on the checkbox to the left of each document. Click the Go button to process the form and email the documents back to you. The document you receive is only a copy of the archived version and cannot be modified, updated, and/or checked back in to the repository.

Projects Screen

The Projects Screen is used to designate project pages on the SuRPAS Intranet that need to display the FTK. If you want to display the FTK on a project where it is not currently displayed, click on the checkbox to the left of the selection so that it becomes “checked”. To remove an FTK from a project page, “un-check” the checkbox to the left of the selection. When you are finished making your choices, click the Go button to perform the update. You will not see this change reflected immediately on the SuRPAS project pages. This information will be used when the project pages are automatically generated. Currently, this happens nightly and at about 12:30PM Monday through Friday afternoon.

To add projects, send an email to the WebMaster specifying the project or projects that you want added.

Daily News

Each night, a new Daily News page for SDMS documents is automatically generated. The page contains a list of all documents that have been checked in within the last two weeks. The SDMS Daily News page can be accessed by using the News, SDMS Daily News link on the home page.

HTML Conversion

As documents are checked in, an HTML version is created. This will occur as long as the document is less than 1 Megabyte (MB) in size. Documents that are too big will be converted at night by a batch process. This steps provides users with the fastest possible access to FTK related documents, especially for remote users. The HTML format is used to view the documents via a browser and the native file format will be provided when the document is retrieved.

Project Tracking

Overview

FTK (FAL Tracking) is a term that refers to both the process of tracking SuRPAS hardware and software projects, and the document that is created and used to manage these projects. The FTK document is only one of a number of programmer-related documents that have and will be discussed in this chapter (including the Requirements Specification and The Development Estimate).

In 1999 a web-based, browser enabled application was created to support the FTK process called WebFTK. Programmers must use WebFTK to manage projects assigned to them and their associated FTKs.

This section discusses:

- WebFTK
- The FTK Document
- The Devlopment Estimate

WebFTK

The WebFTK application is accessed from the SuRPAS intranet by selecting WebFTK from the side-bar list on the SuRPAS Home Page. When selected, the FTK main menu will appear, similar to Figure 23-8 seen below. The main menu allows the user to enter a legal FTK number. When the Go button is pressed the FTK profile menu will appear.

There are other choices available from the FTK line on the main menu. These include an FTK Search option, the ability to display the FTK-specific Notes page and FTK-specific Help. In addition, the blue bottoms below the FTK line provide the user access to the SuRPAS Home Page, the SuRPAS Document Managment System (SDMS), Customer and Project Pages, and FTK Search capability.

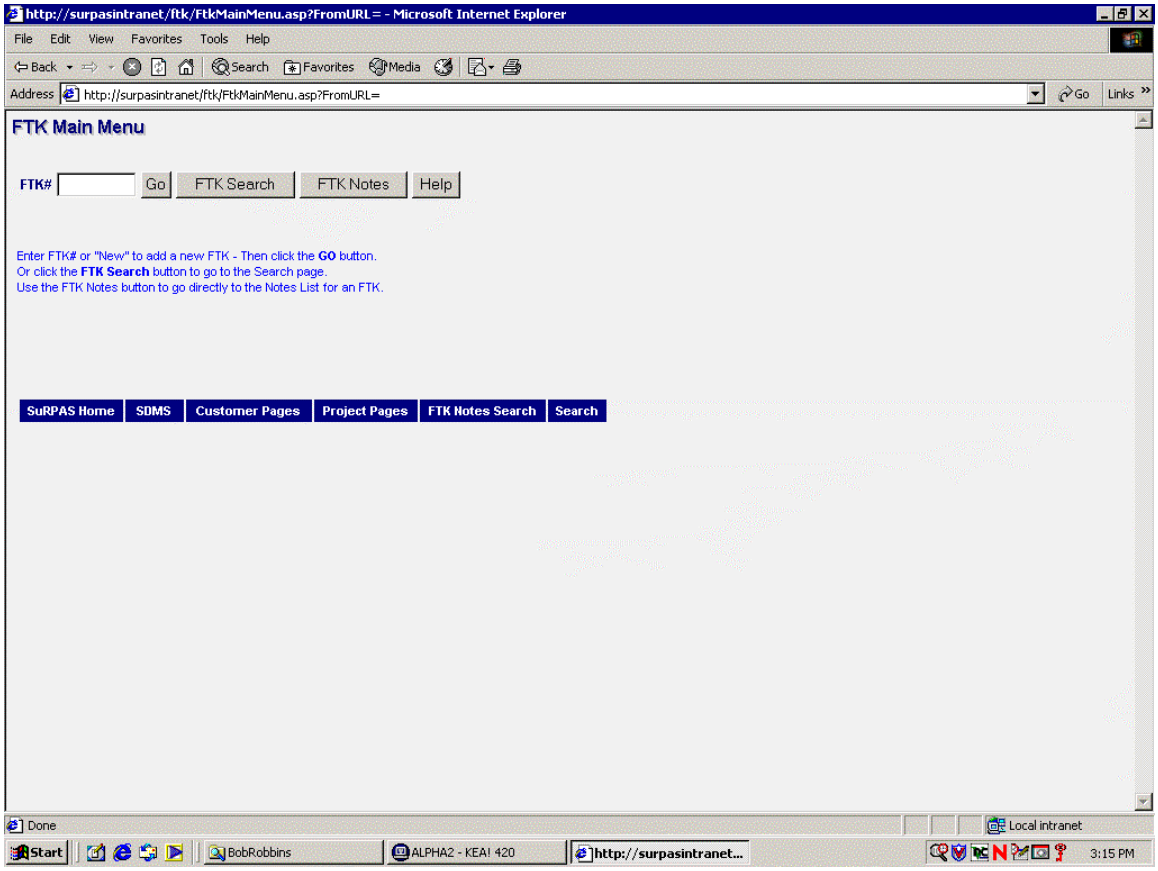

Figure 23-8 - WebFTK Main Menu

The FTK Profile

When a specific FTK is selected on the FTK Home Page, unless FTK Notes is chosen, the user will be moved to the FTK Profile page, as seen in Figure 23-9 below. The profile page displays basic information regarding the project, including billing information, schedules, interested parties, and task status.

The blue buttons on the left side-bar provide additional WebFTK functionality including the ability to update information on the profile page, access to FTK notes, access to FTK-related documents (through SDMS) and email capability.

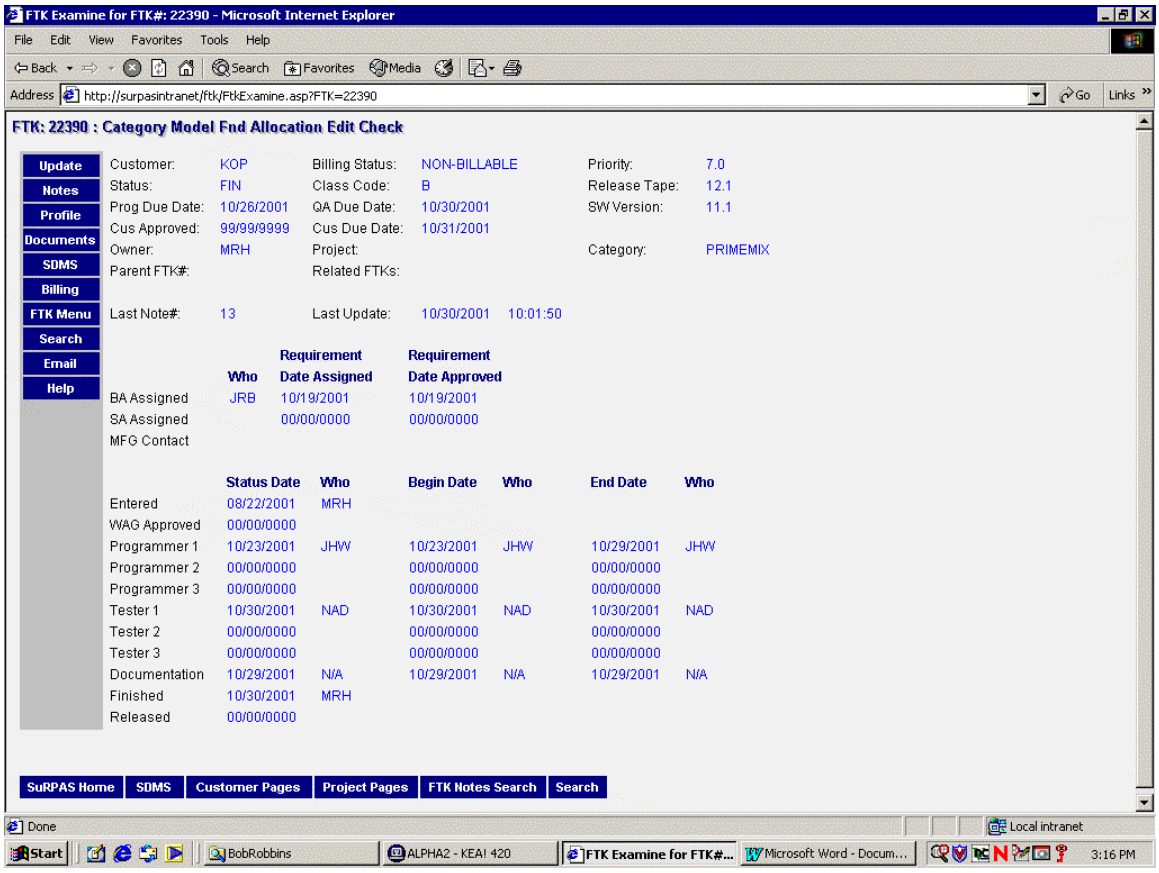

Figure 23-9 - FTK Document Profile Screen (in WebFTK)

FTK Notes

The selection of FTK Notes from a number of WebFTK screens will result in the appearance of the Notes screen, as seen in Figure 23-10 below. FTK notes are entries made to the FTK document through WebFTK, that reflect important information about the project. A note can be entered by any Interested Party or by the Code Management System (CMS) through an automated entry. Each note contains the note entry date, the author of the entry, a note title, and a description.

The FTK notes screen shows all existing notes for the project and also allows the addition of new notes through the selection of the Add Note blue button on the left side-bar.

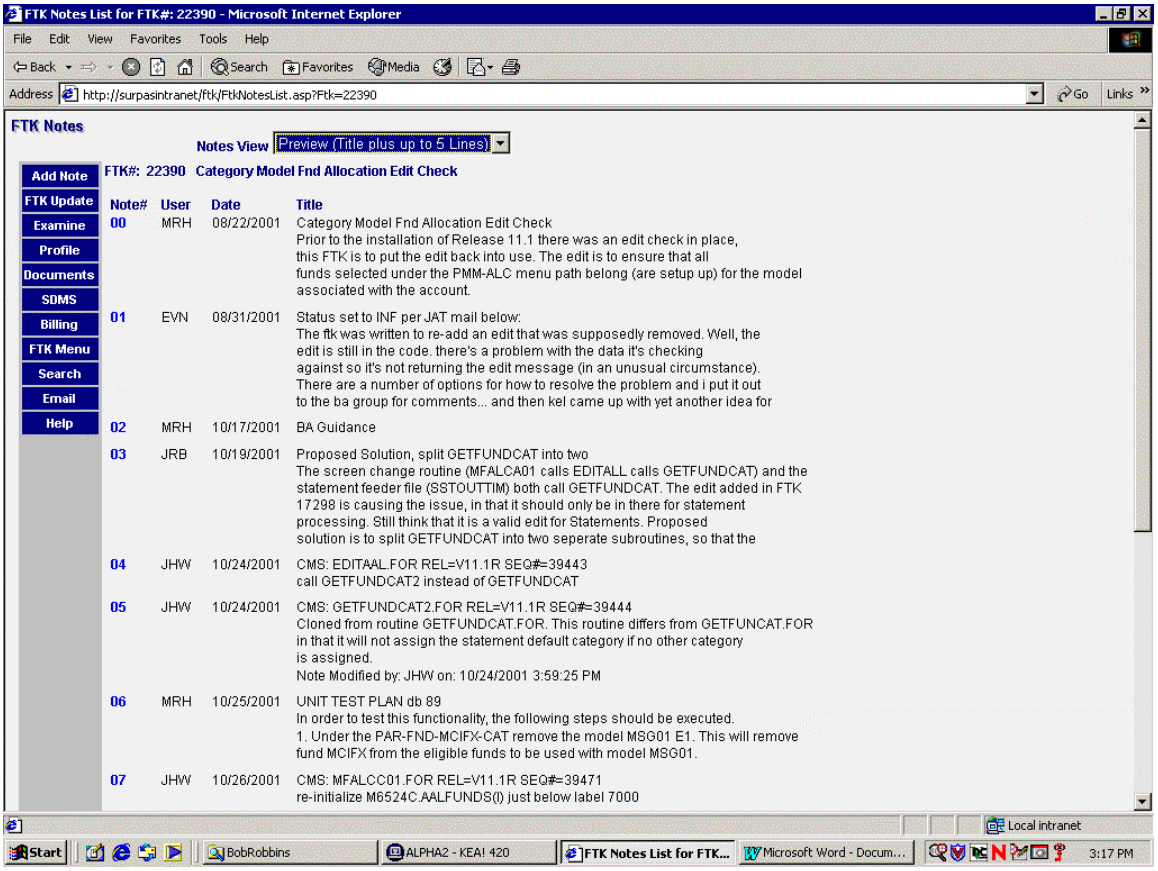

Figure 23-10 - FTK Notes Screen (in WebFTK)

Code Development

Overview

Once the Requirements Specification document has been reviewed and approved and the Programmer has completed the high level design documentation sections of the FTK (in WebFTK), coding can begin. It is the Programmer's responsibility to adhere to the coding standards, the design document and the unit test plan as he/she is coding the project solution.

This section discusses:

- Unit Test
- Code Review

Unit Test

When designing a unit test plan for a project, it is important to cover a number of areas that must be addressed over and above the normal unit test areas. These topics are covered in the next sections

New SuRPAS Screens

- Verify the screen is displayed when following the menu path specified by the requirements. Make a screen print.
- If data is being saved on the screen, verify this data is stored on the database. Make a screen print of parmaint (if a parameter is changing) or create a WB/UDMS report of the data saved from the screen. Verify that any modifications to data are updated accordingly. Verify that the correct data is re-displayed on the screen after the data is saved.
- Verify that you can't enter the same key data more than once.
- Verify that each field follows the rules specified by the requirements.
- Verify that each edit message is displayed properly on the screen and the cursor is returned to the field where the error occurred. Make a screen print.
- Verify that all PFx keys work as specified by the requirements.

Updated SuRPAS Screens

- Follow the same rules as above for each new field added to the screen.
- Test a few of the old fields to verify that their functionality was unchanged.
- Verify that any modifications made by a PF1 update only changes the byte positions for those fields which were modified. Any other byte position should remain unchanged.

Additional Parameters in an Event

Additional parameter screen

- Follow the same rules for unit testing screens (above).
- When testing dates, make sure that the from date is less than or equal to the to date.
- When testing dates, makes sure only valid dates are accepted.

The PF4 Examine screen

- Verify that the examine screen reflects what was entered on the additional parameter screen. Make a screen print.

Events and Reports

- Step through the event and verify that it accurately reflects the requirements. Sometimes it may be necessary to place the code in debug to verify that conditional logic is working the way it is expected.
- Verify that each step of the event is functioning properly. For example, if a scratch file is created first and then reports are generated off of that data, you will need to make sure that the scratch file is being generated properly before verifying the reports.
- Look at the data in the standard files to ensure proper formating and information. Do not just verify that the data on the screen is correct. The file may actually be incorrect and the screen formatting is compensating for the error.
- Verify that the reports reflect the requirements. Verify totals and subtotals.
- Verify that the report page and total breaks reflect the requirement.
- Print out and save any reports produced from the event.
- Verify each byte of an interface file and make sure it matches the requirements.
- Verify data that hasn't changed as well as the changes to an event.
- Verify all error conditions and make sure that they are properly handled.

Installation Procedures

- Not all projects will require installation procedures. If an ftk does require one, follow the same guidelines as testing an event. Both the Install procedure and the Verify procedure need to be unit tested. Any results that show that these have functioned properly should be printed and saved.
- Typically, it is easier to test the Verify procedure first. Once the Verify procedure proves that the Install procedure did not run (it should abend in some way), the Install procedure can be tested. The Verify procedure should then be re-run after a successful completion of the Install to show that it runs successfully.
- Verify that the Install is re-runable. An Install must be re-runable and not cause any corruption of data if it is re-run.

Compile All Results

- Throughout your unit testing you should be taking snapshots of screens or of data before and after changes have been made either through WB or UDMS. These should all be saved and given to the QA tester during the turnover. It is beneficial to save any output that corresponds to functionality stated in the requirements. REMEMBER – The more proof you have that your code is working as required, the better prepared you will be.

Schedule Turnover Meeting With the QA Tester

- As the last step in the process, a meeting should be scheduled with the QA tester to turnover the unit test results. Once this turnover occurs and the QA tester approves the results, the code can be moved into the Test SW tree for QA testing.

The Code Review

The Purpose of a Code Review is to ensure that newly written code adheres to programming standards, screen standards, and business requirements. The quality of a product is better when the code has been reviewed because the code review examines and discusses program design and maintainability. Code reviews can be a training tool for a new programmer as well as a method for getting a seasoned programmer familiar with new functionality. There are instances where code reviews have uncovered issues that would not have been discovered during testing. It is cheaper for an organization to uncover and fix bugs before the project goes into the testing phase. The farther the project is in its life cycle, the more expensive a bug is to correct.

Code Review Roles

Author: The author is the programmer of the code. The author must respond to questions and provide clarification during the code review. This is a **required** role.

Moderator: The moderator is the SA, and they keep the meeting focused. The moderator should also be used as an inspector. This is a **required** role in the code review.

Recorder: The recorder documents issues and solutions raised during the meeting, and should also be used as an inspector. The recorder updates the FTK with the outcome of the meeting and emails that information to the review participants. The recorder is **not** a required role, and thus the code review should not be cancelled if the recorder cannot attend. The moderator or one of the inspectors can perform the recorder role, and the role may be assigned during the meeting.

Reader (user): Typically a tester fills this role. They begin the meeting by giving a general overview of the project, and they are also responsible for ensuring the code satisfies the project requirements. This is a **required** role in the code review.

Reader (tech): A programmer (the author of the code) typically fills this role. This role walks everyone through the code describing the implementation.

Inspector: Usually there are 3 inspectors per review. The inspector is responsible for inspecting the code for accuracy. This role can be combined with the other roles, allowing for a total of 3 inspectors plus the author. Two inspectors are required for a code review.

PSG: A representative from PSG must be invited to attend the code review (send email to TMI). The PSG attendee will be used as an inspector. This is **not** a required role and the code review should **not** be called off if the PSG rep. cannot attend.

Code Review Rules

The programmer initiates the code review. The Code Review Roles form should be initially filled out by the programmer with the FTK information, and forwarded to the SA to signify when the project will be ready to be reviewed.

The participants and roles in the code review will be determined and assigned by the system analyst with the team manager's (TM) approval. The SA will update the Roles form with the participants and their roles, and send the form to the TM for approval. The TM may want additional people involved whom the system analyst would not normally invite (for training purposes, for example). Participants should not be limited to members of the author's team.

Once it is approved, the Roles form will be forwarded by the TM to the programmer, and the programmer will schedule a meeting at least 3 days in advance. The programmer is responsible for providing a printed copy of the code to be reviewed. In the meeting invitation message the programmer should also include a copy of the Roles form. All team leaders should be copied on the invitation. Any attendees who are remote should either be emailed the code review package or sent a list of programs to print.

- Printed material should be new code or a major routine change. If changes to a large program are minimal, a parallel diff report may be distributed. If diffs are distributed, the programmer will bring 1 copy of the entire program to the code review for reference.
- The CFDs should be printed and reviewed to ensure that the comments are appropriate for the fields.

Participants need to be familiar with the requirements and are responsible for reviewing the code prior to the meeting. The moderator will suspend the meeting if the participants are not prepared. The SA will review the code for adherence to the requirements. No other participants are required to do this type of review.

The reviewers will meet at the scheduled time and review the code for accuracy, maintainability, use of standards, and adherence to the design. Style issues should not be addressed in the review unless it is agreed that the style is not easily supported or does not conform to standards.

- The programmer will bring a copy of the unit test results to the meeting for review. This should **not** be handed out to all participants before the meeting.

Throughout the course of the review, the SA and code review participants will point out areas that need QA testing, based on the routines being reviewed. If a tester is not present, the recorder should email the information to the tester.

Based on issues identified, meeting members will determine whether a second review is needed. The SA should review issues uncovered during the code review process before the project is considered complete.

Code Review Notes

The code review should be held as long as there are a minimum of 2 participants other than the author. The SA must attend.

All code will be reviewed. Not reviewing code is an exception requiring SA and TM approval. If the project is small and has minimal risk, the system analyst may review the code and not call for a formal code review.

Guidelines for types of projects that do not need a review:

- minor screen changes
- non-critical events that have minor modifications
- upgrades

If there are a few routines that require the same type of coding change, one routine should be completed and informally reviewed before the others are started. The programmer initiates the informal review, and the 3-day notice is not to be enforced by the SA.

Code Review Checklist

√ As you unit test your FTK, you should begin to think about approximately when you will finish testing. When you have almost completed your unit testing, send out the Code Review Roles sheet to your team manager and the SA on the project. Your team manager and SA will determine the attendees for the code review. Once this document is returned, you can schedule the code review.

√ Schedule the code review at least three days ahead of time. This will allow the attendees at least 2 full days to prepare. You should use the Lotus Notes Scheduler to invite your attendees. This ensures that their calendar is free.

√ When scheduling the event you should include the following:

- Code Review Roles document – defining the roles of those who are to attend.
- If you are a remote employee, you should include where the code resides online, and which attendee can print the material for the rest of the group.
- At this time if you wish, you can also include a copy of the design specifications and/or the requirements documentation.

√ The following should be distributed to each attendee at least 3 days prior to the review:

1. When distributing the code review source listings please use the Code Review Roles doc (that's what gets mailed back to you from your TM) as the cover page with the date, time, and location of the code review meeting written on it. If it's an informal code review (i.e., no Code Review Roles document), put the FTK# with a brief description and the date, time, and location of the code review at the top of the 1st page.

2. Provide to all participants **either**:
   a print out of your modules (for changes to existing modules please signify areas that have changed by marking vertical lines next to the areas in the right margin of the printout),
   or:
   if the modification is a small change to an existing module and the context is not necessary, print out a DIFF report.

   If you are not certain which of the above is appropriate, discuss this with the SA. If a DIFF was distributed to the group instead of the module, please bring **one** full copy of the module to the code review for reference. It is beneficial to distribute the code in the same order that it will be reviewed (consider printing them out in their hierarchical calling tree order) . It may be helpful to number the printouts so that everyone can refer to a specific page during the review. For copying & distribution purposes please utilize the "front desk" person to get this done for you.

3. Distribute with the code, printed CFD's, CRD's, PAR's (if new or modified) to each participant.

4. One copy of your unit test results should be brought to the code review. These can be reviewed during the meeting, if necessary. Afterwards, they should be given to the QA tester.

5. One copy of the design document.

6. One copy of the requirements.

7. Anything that may be pertinent to this FTK – in case questions arise.