

User Interface Guidelines

Document Number UICC-GUIDE-001

November 4, 1988

Jane Lewis (LEWISJ at RCKVM1)
Fran Phillips (PHILLIPF at RCKVM1)

International Business Machines Corporation
Systems Integration Division
System Technology and Products
User Interface Center of Competence
18100 Frederick Pike
Gaithersburg, MD 20879

IBM Internal Use Only

Preface

This document provides guidelines for design and development of the User Interface (UI), often referred to as the Human-Computer Interface. It is intended for both management and technical personnel who are responsible for a project's software User Interface.

Keying data into a system to schedule satellite contacts, touching a screen to select the WITHDRAW CASH option on an automatic teller machine, or using a joystick to play a video game are all examples of people interfacing with a computer. Different UI design elements must be employed to support each of their needs, which vary because of the nature of the tasks they are performing, and because of their own experience, expertise and personal characteristics.

The UI Guidelines define a user interface implementation methodology applicable to any system's development life cycle. The methodology identifies and describes the critical UI development activities which result in an interface that successfully meets both user needs and system requirements. This methodology does not require the development of any new documents; all UI documentation is integrated into established project documentation. The Guidelines are presented in the context of a phased process which can be implemented on diverse application development projects.

Contents

Part 1. Introduction	1
UI Guidelines Organization	1
User Interface (UI) Definition	2
UI Development Methodology	2
Management	2
Requirements Analysis and Definition	3
Design and Implementation	3
Test	4
UI Guidelines Relationship to Human-Computer Interface Requirements Specification (HCIRS) Bulletin	4
Phased UI Development Process	8
UI Activities Responsibilities	14
<hr/>	
Part 2. UI Development Process	19
Phase 1. UI Preproposal/Proposal	21
Goals	21
Tasks	21
Products	22
Related Project Activities/Products	22
Additional Information Sources	22
Issues To Be Addressed	22
Risks/Concerns	24
UI Resource Requirements Estimates	24
UI Configuration Control	24
User Documentation Development	24
User Reviews/Participation	24
UI Preproposal/Proposal Guidelines	27
Management	27
Requirements Analysis and Definition	29
Design and Implementation	31
UI Design Team Rationale, Composition & Responsibilities	33
UI Design Team Rationale and Responsibilities	33
UI Design Team Composition	33
Discipline Responsibilities	34
Systems Engineering (SE)	34
Software Engineering (SWE)	34
Hardware Engineering (HWE)	34
Human Factors Engineering (HFE)	34
Integration and Test (I&T)	34
Integrated Logistics Support (ILS)	34
QA	35
User	35
Usability	37
Usability Plan	37
Usability Objectives	37
Usability Evaluation/Measurement Methods	37

Schedule	39
Risks	39
Key Dependencies	39
Relationship Between Usability Objectives and UI Requirements	44
Usability Objectives	44
System UI Requirements	44
Software UI Requirements	44
Usability Checklist	46
Phase 1 Checklist	48
Phase 2. UI Concept Development	51
Goals	51
Tasks	51
Products	52
Related Project Activities/Products	52
Additional Information Sources	52
Issues To Be Addressed	53
Risks/Concerns	54
Concept Development Guidelines	55
Management	55
Requirements Analysis and Definition	55
Design and Implementation	57
User Profile Tool	59
User Profile Tool Analysis	64
HCIRS Bulletin	69
Mitre Guidelines	70
SAA/CUA	71
Dialog Style Trade-offs	72
User Documentation and Training Development	74
UI Life Cycle Phase 1 (Pre-proposal/Proposal)	74
UI Life Cycle Phase 2 (Concept Development) and Phase 3 (Requirements Analysis and Definition)	74
UI Life Cycle Phase 4 (Evaluation)	75
UI Life Cycle Phase 5 (Operations Support)	75
UI Life Cycle Phase 6 (Post-Installation)	76
Usability Checklist for User Documentation	78
Online HELP Checklist	79
Computer Messages	81
Computer Message Checklist	82
Phase 2 Checklist	83
Phase 3. UI Requirements Definition and Design	85
Goals	85
Tasks	85
Products	86
Related Project Activities/Products	86
Additional Information Sources	86
Issues To Be Addressed	87
Risks/Concerns	89
Maintaining User Involvement	89
Continued Usability Management	89
Incorporating Usability in UI Design	89
Maintaining UI Product Control	90
UI Requirements Definition & Design Guidelines	91
Management	91
Requirements Analysis and Definition	91

Design and Implementation	92
Test	93
Phase 3 Checklist	94
Phase 4. UI Evaluation	97
Goals	97
Tasks	97
Products	97
Related Project Activities/Products	98
Additional Information Sources	98
Issues To Be Addressed	98
Risks/Concerns	99
UI Evaluation Guidelines	101
Management	101
Design and Implementation	101
Test	102
Sample User Evaluation Questionnaire	103
Phase 4 Checklist	109
Phase 5. UI Operations Support	111
Goals	111
Tasks	111
Products	111
Related Project Activities/Products	111
Additional Information Sources	112
Issues To Be Addressed	112
Risks/Concerns	113
UI Operations Support Guidelines	115
Management	115
Requirements Analysis and Definition	115
Design and Implementation	115
Test	116
Phase 5 Checklist	117
Phase 6. UI Post-Installation	119
Goals	119
Tasks	119
Products	119
Related Project Activities/Products	119
Additional Information Sources	119
Issues To Be Addressed	120
Risks/Concerns	120
UI Post-Installation Guidelines	121
Management	121
Test	122
Phase 6 Checklist	123
Glossary	Glossry-1
Bibliography	Biblio-1

Figures

1.	User Interface Activities During Project Development	5
2.	Typical UI Activities Timeline	7
3.	User Interface Activities During Phase 1	20
4.	"Cost" of Change Over the UI Life Cycle	32
5.	UI Responsibilities and Interfaces by Discipline	36
6.	Sample Usability Plan Outline	40
7.	Sample Usability Plan Schedule (Interdependencies not shown)	43
8.	User Interface Activities During Phase 2	49
9.	User Documentation and Training Development	77
10.	User Interface Activities During Phase 3	84
11.	User Interface Activities During Phase 4	95
12.	User Interface Activities During Phase 5	110
13.	User Interface Activities During Phase 6	118

Tables

1. Overview of UI Goals, Tasks, and Products by Phase	8
2. Discipline Responsibilities for UI Activities	15
3. Usability Objectives and UI Requirements Examples	45
4. Dialog Style Trade-offs	72

Part 1. Introduction

These User Interface (UI) Guidelines identify, define and describe a process for designing and validating the software user interface during a system's development and deployment. The UI development process consists of six phases; each phase corresponds to a section in these Guidelines. Although the UI process has been divided into distinct phases, it is important to understand that the development process is iterative. The design may undergo modifications as it is refined during each phase of the process.

UI Guidelines Organization

This document contains the following sections:

- Part 1, "Introduction"
- Part 2, "UI Development Process"
- Phase 1, "UI Preproposal/Proposal"
- Phase 2, "UI Concept Development"
- Phase 3, "UI Requirements Definition and Design"
- Phase 4, "UI Evaluation"
- Phase 5, "UI Operations Support"
- Phase 6, "UI Post-Installation"
- Glossary
- Bibliography

Each phase consists of an Overview followed by a Workbook. The Overview orients the reader to UI development goals and issues. The Workbook provides the step-by-step activities and aids for implementing that phase of user interface development. Each phase or section contains:

- Overview
 - Goals
 - Tasks
 - Products
 - Related Project Activities/Products
 - Additional Information Sources
 - Issues To Be Addressed (presented as "thought provoking" questions)
 - Risks/Concerns
- Workbook
 - Guidelines for performing the tasks
 - Design aids to help perform the tasks
 - Checklist(s) (verification that development should proceed to the next phase)

Figure 1 on page 5 depicts the phased UI development process and lists major UI activities. The milestones used to orient the reader to phases in Figure 1 are intended to be generic descriptions of events, although they are commonly used Department of Defense (DOD) terms. Figure 2 on page 7 shows a timeline of these activities and relates them to established system development milestones. Table 1 on page 8 provides an overview of UI goals, tasks and products for each phase of UI development. Table 2 on page 15 identifies the discipline(s) responsible for each activity. The UI development process is linked to established Systems Engineering (SE) life cycle phases, as described in *SID Corporate Bulletin: System Life Cycle* (C-B 0-0100-001), and is tied to established project milestones and Software Engineering (SWE) life cycle phases as described in *SID Division Bulletin: The Software Life Cycle* (D-B 0-4010-001).

User Interface (UI) Definition

UI, as used in these Guidelines, is defined as the actions performed by humans using an interactive device, like a keyboard, mouse, trackball or touch screen, to provide information to or request processing from a computer. It also includes computer responses (including display panels) as a result of information/requests from the user.

Any time human intervention is required for the computer to complete a task or function, elements of UI are involved. UI includes the hardware devices that comprise a workstation (CRT(s), keyboards, printers, etc.) where human input takes place, the software that translates user actions and information into computer processing data, and the documentation, training and user aids (HELP panels, tutorials, keyboard templates, etc.) designed to support or assist the user in performing tasks on the system. Although the system development process includes both hardware and software, the scope of these Guidelines is limited to the software user interface and user support.

Have you ever deleted a file by mistake, realizing it the instant you pressed the ENTER key? Or have you ever received a SYSTEM ERROR message and had to guess how to correct it? A good UI would have required a confirmation of the delete request, provided an UNDO capability, and/or prevented the deletion of a critical file, and provided a more meaningful message and complete error recovery procedures. These are examples of UI issues.

UI Development Methodology

Early consideration of the human element is essential in establishing a system's design concept. Understanding the needs of the users and system goals impacts how we develop and design systems. ('User' is defined as the end user of a system, not IBM's customer or contracting agency. The user can be an operator, maintainer, controller or any person who performs a task by interacting with a computer.)

The major components of the methodology, which are applied throughout the six phases of the UI process are:

- **Management**
- **Requirements Analysis and Definition**
- **Design and Implementation**
- **Test**

This UI methodology does not require the development of any new documents; all UI documentation is integrated into standard project documentation.

Management

The management component includes:

- UI Design Team
- Usability Plan
- Configuration Control (CC)
- Compliance to UI Requirements

UI Design Team: The UI Design Team (UIDT) is a working group composed of representatives from all disciplines involved in the UI development process: Systems Engineering (SE), Software Engineering (SWE), Hardware Engineering (HWE), Human Factors Engineering (HFE), Integration and Test (I&T), Quality Assurance (QA), and Integrated Logistics Support (ILS). Representatives from

Reliability/Maintainability/Availability (R/M/A) and Information Development (ID) should also be included, as appropriate. A user representative, any subcontractor(s) associated with the user interface development and associated Subcontract Acquisition Manager (SAM) may be invited to participate. This team functions like the Interface Control Working Groups currently used by SE to design and control hardware and software interfaces (internal and external). The UIDT objective is to ensure the system provides a user interface which satisfies the users' needs and meets the system's requirements. Its purpose is to provide the technical and management guidance necessary to successfully implement the UI development process, thereby meeting the objective. The UIDT functions and responsibilities may be included in the Engineering Integration Workshop (EIW), whose purpose is to coordinate and manage the interdependencies of similar multiple discipline activities.

The importance of each discipline assuming responsibility for activities (primary and support) and ensuring the interfaces and dependencies are well managed is critical to UI development.

Usability Plan: A Usability Plan is generated for UI development. It may be included in the proposal if desired by the proposal manager. It includes the usability objectives for the system, the evaluation criteria and measurement methods to ensure the objectives are met, a schedule, risks and key dependencies which may impact UI development or system acceptance. Following contract initiation, this plan is integrated into the SE Technical Management Plan (TMP) and the Software Management Plan. (Refer to *SID Corporate Bulletin: Technical Management Plan (C-B 0-2507-005)* for information on the TMP and *SID Corporate Practice: Software Project Management (D-P 3-7099-003)* for more information on the Software Management Plan.) Usability planning focuses on the development of and adherence to testable, measurable criteria used throughout system development to evaluate usability, demonstrates compliance with requirements for system sell-off, and demonstrates management commitment to usability.

Configuration Control: Configuration control of UI design, design changes, change history, panel development, error messages, and UI documentation is critical to maintaining a single UI baseline configuration throughout development. UI requirements traceability must also be managed to ensure all system requirements are contained in the final detail design.

Compliance to UI Requirements: Procedures for monitoring project-wide compliance with UI requirements are essential to achieve a consistent design and ensure the delivered system meets the customer's needs and contract obligations.

Requirements Analysis and Definition

The Requirements Analysis and Definition component involves using standard SE and HFE methods to identify and clarify users' needs (requirements) and to generate testable UI requirements. Application-specific requirements evolve from generic usability objectives, specified and implied system goals, design standards and users' needs. In defining system requirements and making UI design decisions, it is also important to obtain a profile of the users in terms of their experience, level of expertise and personal characteristics.

Design and Implementation

The Design and Implementation component includes selection of an approach or solution in response to users' needs and, developing UI products: panels, messages, online HELP, user documentation, training materials, and prototypes. Users should be involved early in design decisions via the Design Team. The design process is iterative (design → review → modify → review, etc.) with all disciplines and users having input to design decisions and reviews as development progresses.

Test

The Test component includes iterative user evaluations during development and system test. Objective and subjective user feedback can be obtained from prototype testing and system simulations to ensure the system will be usable. During system test, demonstrating the UI meets requirements is ensured by generating measurable requirements and establishing UI acceptance criteria with the customer early in system development.

UI Guidelines Relationship to Human-Computer Interface Requirements Specification (HCIRS) Bulletin

Underscoring the importance of developing and understanding UI requirements, SID has produced a corporate bulletin, *Human-Computer Interface Requirements Specification (HCIRS) Content and Development Process* (C-B 0-2507-011), which details the process of user interface design and the development of a user interface requirements specification. The development of a Human-Computer Interface (HCI) requirements specification may be accomplished by creating a separate Interface Requirements Specification (IRS) or by incorporating HCI requirements in the system and software specifications. The methodology presented in these Guidelines follows the second approach and details a process for developing the UI throughout a system's life cycle.

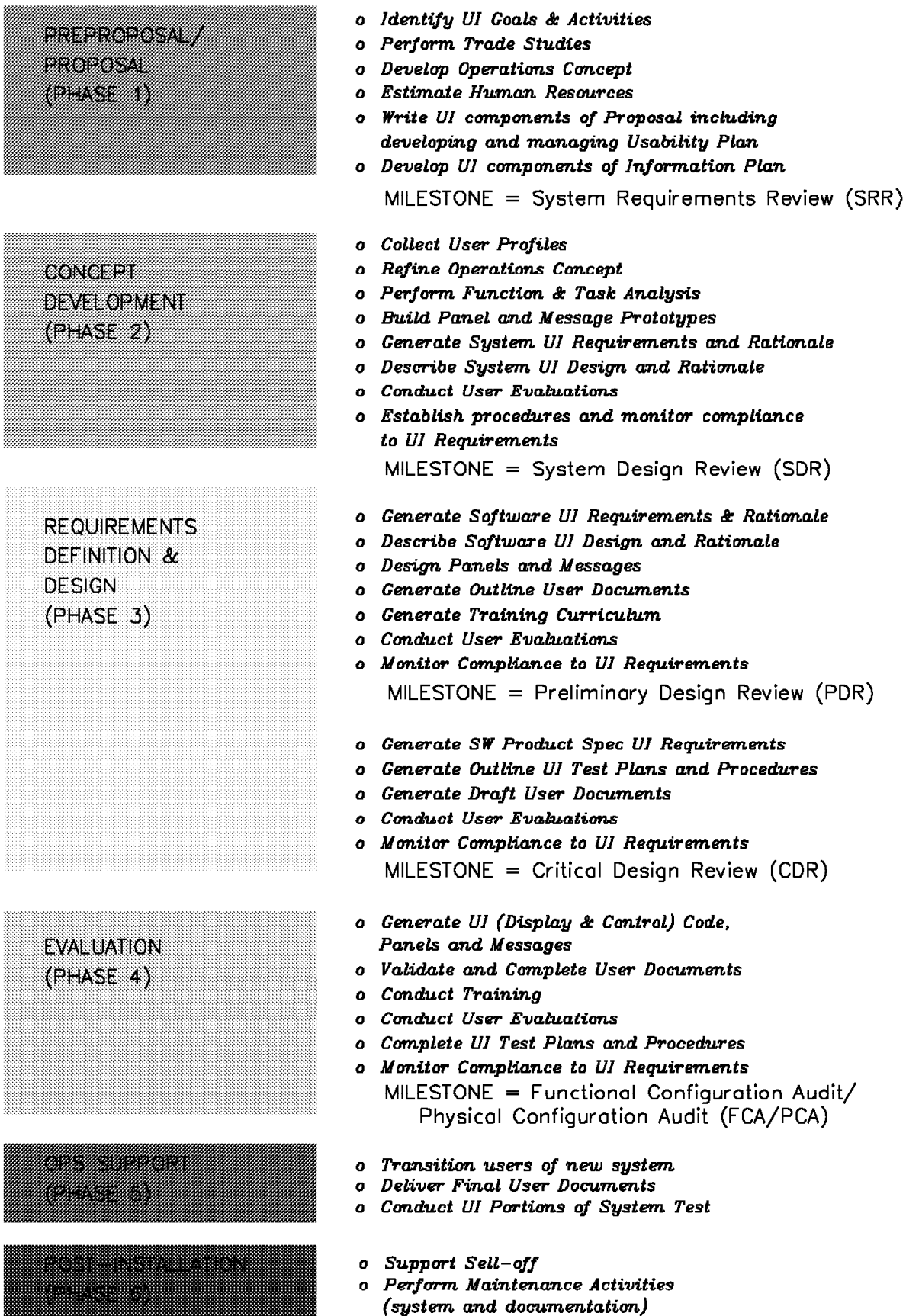


Figure 1. User Interface Activities During Project Development

(This page intentionally left blank.)

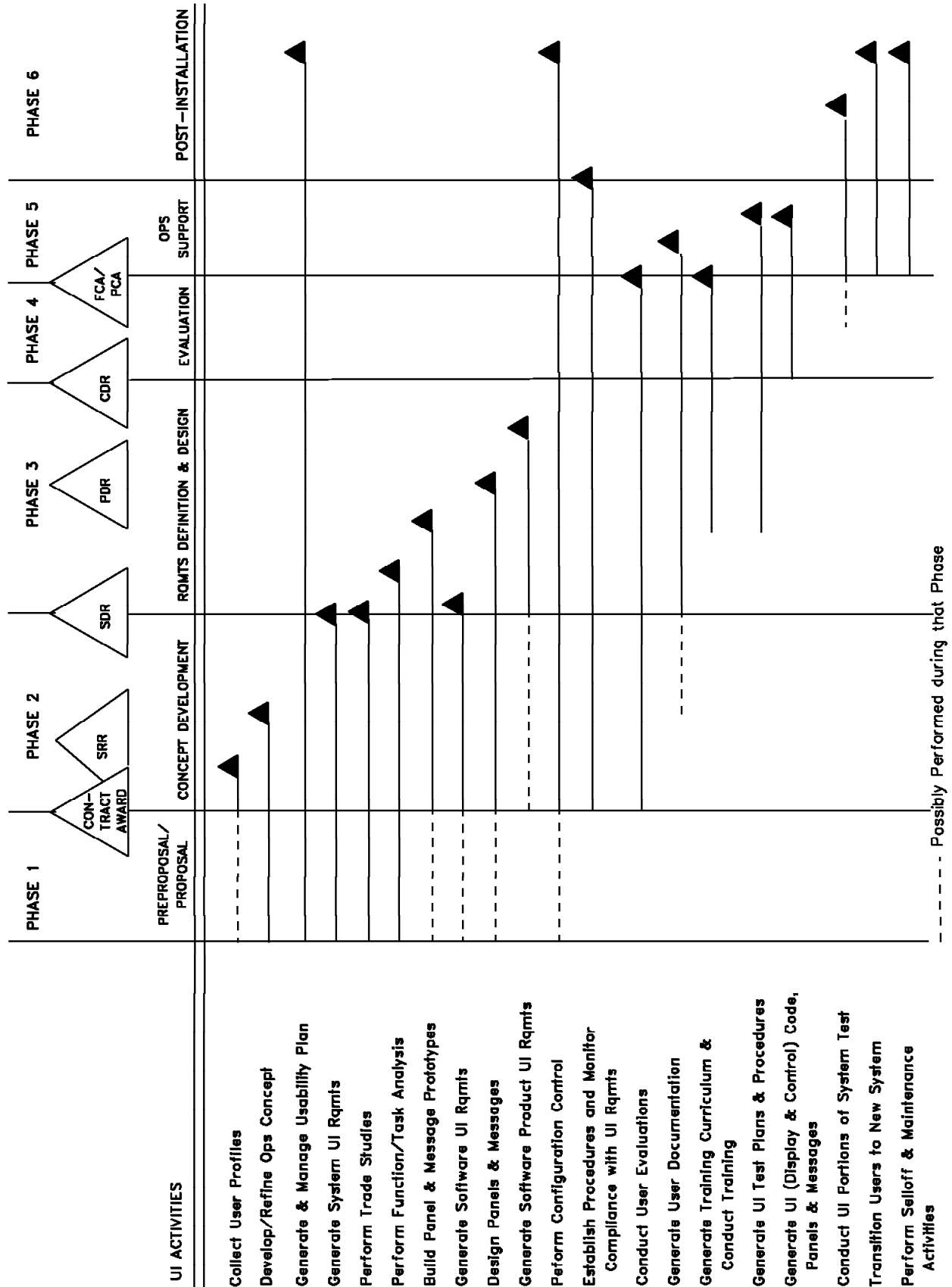


Figure 2. Typical UI Activities Timeline

Phased UI Development Process

Table 1 (Page 1 of 6). Overview of UI Goals, Tasks, and Products by Phase

GOALS	TASKS	PRODUCTS
<i>Phase 1 - Preproposal/Proposal</i>		
<ul style="list-style-type: none"> • Form UI Design Team • Develop project usability objectives and plans • Identify and define UI activities 	<p><u>Management</u></p> <ul style="list-style-type: none"> • Form UI Design Team • Develop Usability Plan • Estimate resource requirements and schedules • Identify risks and describe activities to minimize them • Identify and define UI portions of deliverable documents • Develop UI portions of Information Plan • Establish UI configuration control procedures <p><u>Requirements Analysis</u></p> <ul style="list-style-type: none"> • Identify/describe activities to define UI requirements • Begin requirements definition • Begin developing UI portions of Ops Concept document <p><u>Design and Implementation</u></p> <ul style="list-style-type: none"> • Support system hardware and software design decisions • Generate sample panels, online HELP, error messages, user documentation, and training materials 	<ul style="list-style-type: none"> • Usability Plan • UI project/proposal resource estimates • UI requirements for draft System Requirements Specification and supporting rationale report • UI portions of draft Operations Concept document • UI portions of Information Plan • Sample panels, online HELP, messages, user documentation, and training materials

Table 1 (Page 2 of 6). Overview of UI Goals, Tasks, and Products by Phase		
GOALS	TASKS	PRODUCTS
<i>Phase 2 - Concept Development</i>		
<ul style="list-style-type: none"> • Refine the operations concept • Complete system UI requirements and design development • Begin software UI requirements and design development • Obtain early user feedback on system definition and UI design 	<p><u>Management</u></p> <ul style="list-style-type: none"> • Review and update Usability Plan • Review and update Information Plan • Apply configuration control procedures for UI products • Establish compliance monitoring procedures for conformance to UI requirements <p><u>Requirements Analysis</u></p> <ul style="list-style-type: none"> • Collect and analyze user profile(s) • Describe users' tasks and perform Task Analysis • Generate operational scenarios • Complete UI system requirements development • Develop information and training requirements • Support function analysis and functional allocation • Allocate UI software requirements <p><u>Design and Implementation</u></p> <ul style="list-style-type: none"> • Support trade study activities for hardware and software selection • Generate UI portions of Ops Concept document • Refine approach for panels, error messages, online HELP, user documentation and training • Select input/output devices and dialog style(s) • Design panels and messages • Build panel and message prototypes • Write user documentation outline(s) • Develop Training Outline • Conduct user evaluations of prototype panels, messages and online HELP 	<ul style="list-style-type: none"> • Updated Usability Plan • Updated Information Plan • UI portions of Ops Concept document • UI requirements for System Requirements Specification and supporting rationale report • UI portions of System • Design Description Document and supporting rationale report • UI requirements for draft Software Requirements Specification and supporting rationale report

Table 1 (Page 3 of 6). Overview of UI Goals, Tasks, and Products by Phase

GOALS	TASKS	PRODUCTS
<i>Phase 3 - Requirements Definition and Design</i>		
<ul style="list-style-type: none"> • Ensure project-wide compliance with UI requirements • Complete software UI requirements and design development (including panel, message and online HELP design) • Complete software UI detail design • Complete training development • Complete user documentation draft (approx. 85% complete) • Obtain objective and subjective feedback from users on prototypes 	<p><u>Management</u></p> <ul style="list-style-type: none"> • Update Usability Plan, as needed; monitor progress • Update Information Plan, as needed • Continue monitoring compliance with UI requirements • Enforce configuration control procedures • Conduct UI product technical reviews <p><u>Requirements Analysis</u></p> <ul style="list-style-type: none"> • Complete UI software requirements development • Complete software UI detail design development <p><u>Design and Implementation</u></p> <ul style="list-style-type: none"> • Complete UI Software Design Guidance Document • Complete panel, message and online HELP design • Complete user documentation outline(s) • Develop user documentation draft(s) • Develop training curriculum <p><u>Test</u></p> <ul style="list-style-type: none"> • Develop UI test plans and procedures outline(s) • Conduct user evaluations of panels, messages, online HELP, and user documentation prototypes 	<ul style="list-style-type: none"> • UI requirements for Software Requirements Specification and supporting rationale report • Panel, message and online HELP designs for Software Requirements Specification • UI Design for Software Design Guidance Document • User Documentation Outline(s) • UI Detail Design for Software Product Specification • User Documentation Draft(s) • UI Test Plan and Procedures Outline(s) for System Test Plans and Procedures • Training Outline(s)/Curricula

Table 1 (Page 4 of 6). Overview of UI Goals, Tasks, and Products by Phase		
GOALS	TASKS	PRODUCTS
<i>Phase 4 - Evaluation</i>		
<ul style="list-style-type: none"> • Obtain user evaluations of baseline system • Complete training • Complete user documentation development • Generate UI (display & control) code, panels and messages 	<p><u>Management</u></p> <ul style="list-style-type: none"> • Update Usability Plan (TMP), as needed; monitor progress • Continue monitoring compliance with UI requirements • Continue configuration control procedures • Perform UI product technical reviews <p><u>Design and Implementation</u></p> <ul style="list-style-type: none"> • Complete user documentation development and perform validation • Develop functional descriptions for user documentation • Conduct training • Develop UI (display & control) code • Generate panels and messages <p><u>Test</u></p> <ul style="list-style-type: none"> • Begin unit testing • Conduct user evaluations of baseline system • Analyze user evaluation results to determine impact to UI requirements • Complete UI test plans and procedures 	<ul style="list-style-type: none"> • UI (display & control) code, panels and messages • Functional Descriptions • User Documentation • Training • UI portions of System Test Plans and Procedures

Table 1 (Page 5 of 6). Overview of UI Goals, Tasks, and Products by Phase

GOALS	TASKS	PRODUCTS
<i>Phase 5 - Operations Support</i>		
<ul style="list-style-type: none"> • Transition users to new system • Deliver user documentation final(s) to customer 	<p><u>Management</u></p> <ul style="list-style-type: none"> • Update Usability Plan (TMP), as needed; monitor progress • Continue monitoring compliance with UI requirements • Continue configuration control procedures • Perform UI product technical reviews • Obtain customer approval of user documentation <p><u>Requirements Analysis</u></p> <ul style="list-style-type: none"> • Support PTR/AAI activities resulting from user evaluation, system test and FCAs/PCAs <p><u>Design and Implementation</u></p> <ul style="list-style-type: none"> • Support customer verification of user documentation, if required • Complete update of user documentation final(s) and deliver to customer • Support users' transition to new system • Complete UI code, panel and message development • Conduct UI code, panels and and messages unit test(s) • Begin developing UI product maintenance procedures, if required <p><u>Test</u></p> <ul style="list-style-type: none"> • Complete unit testing • Support UI portions of system test 	<ul style="list-style-type: none"> • User Documentation Final(s)

Table 1 (Page 6 of 6). Overview of UI Goals, Tasks, and Products by Phase		
GOALS	TASKS	PRODUCTS
<i>Phase 6 - Post-Installation</i>		
<ul style="list-style-type: none"> • Complete transition of users to new system • Archive all UI materials • Implement required maintenance procedures for all UI products • Obtain customer approval and acceptance of system 	<p><u>Management</u></p> <ul style="list-style-type: none"> • Monitor final transition and sell-off activities • Complete configuration control procedures required to transition UI product responsibilities to customer • Collect material to be archived • Send archive materials to SID records retention • Write "lessons learned" document • Implement required maintenance procedures <p><u>Test</u></p> <ul style="list-style-type: none"> • Complete UI portions of system test • Complete UI sell-off activities 	<ul style="list-style-type: none"> • N/A

UI Activities Responsibilities

UI activity and product ownership is key to achieving a successful user interface. Disciplines which may have support responsibilities for an activity play as important a role in the completion of that activity as does the discipline with primary responsibility. In some cases primary responsibility may be product management, while support responsibility may include technical activity performance. Teamwork is essential to the UI methodology described in these Guidelines.

Table 2 (Page 1 of 2). Discipline Responsibilities for UI Activities		
DISCIPLINE	PRIMARY	SUPPORT
Human Factors Engineering (HFE)	<ul style="list-style-type: none"> • Perform Task Analysis • Perform user profile analysis • Generate UI portions of Requirements Rationale Report • Generate UI portions of Design Rationale Report • Design panels & messages • Build prototypes • Conduct user evaluations • Conduct usability tests 	<ul style="list-style-type: none"> • Support Usability Plan development • Support Ops Concept development • Support UI requirements definition • Support user documentation development • Support training
Hardware Engineering (HWE)	<ul style="list-style-type: none"> • Describe UI detail design in HW Product Specifications 	<ul style="list-style-type: none"> • Support Usability Plan development • Support Ops Concept development • Support Function Analysis • Support Trade Studies • Support UI requirements development • Support user documentation development • Support training
Software Engineering (SWE)	<ul style="list-style-type: none"> • Describe UI detail design in SW Product Specifications • Determine requirements implementation feasibility • Develop UI code, panels, & messages 	<ul style="list-style-type: none"> • Support Usability Plan development • Support Ops Concept development • Support UI requirements & design rationale development • Support Function Analysis • Support Trade Studies • Support UI requirements development • Support panel & message design • Support panel & message prototype development • Support user documentation development • Support training • Support usability evaluation & system test

Table 2 (Page 2 of 2). Discipline Responsibilities for UI Activities

DISCIPLINE	PRIMARY	SUPPORT
Systems Engineering (SE)	<ul style="list-style-type: none"> • Chair UI Design Team • Generate & manage Usability Plan • Develop UI portions of Ops Concept document • Perform Function Analysis • Perform Task Analysis • Generate system & software UI requirements • Monitor compliance to UI requirements • Perform configuration control • Design panels & messages • Perform usability evaluation • Build prototypes • Conduct user evaluations 	<ul style="list-style-type: none"> • Support UI portions of Information Plan development • Support UI requirements & design rationale development • Support UI HW & SW Product Specifications development • Support user documentation development • Support training • Support UI code development • Support UI test documentation development • Support system test
Integration & Test (I&T)	<ul style="list-style-type: none"> • Generate UI portions of test plans & procedures • Perform system test 	<ul style="list-style-type: none"> • Support Usability Plan development • Support prototype testing • Support design and code development for testability • Support usability evaluation • Support user documentation validation
Integrated Logistics Support (ILS)	<ul style="list-style-type: none"> • Develop UI portions of Information Plan • Develop user documentation • Develop training materials • Conduct training • Perform documentation maintenance activities • Validate user documentation 	<ul style="list-style-type: none"> • Support Usability Plan development • Support Ops Concept development • Support system test
Quality Assurance (QA)	<ul style="list-style-type: none"> • Monitor conformance to quality standards 	<ul style="list-style-type: none"> • Support UI design inspections • Support UI code inspections • Review user documentation • Review training materials
User	<ul style="list-style-type: none"> • Provide input to design decisions • Participate in prototype testing 	

Part 2. UI Development Process

PREPROPOSAL/
PROPOSAL
(PHASE 1)

- o *Identify UI Goals & Activities*
- o *Perform Trade Studies*
- o *Develop Operations Concept*
- o *Estimate Human Resources*
- o *Write UI components of Proposal including developing and managing Usability Plan*
- o *Develop UI components of Information Plan*

MILESTONE = System Requirements Review (SRR)

CONCEPT
DEVELOPMENT
(PHASE 2)

REQUIREMENTS
DEFINITION &
DESIGN
(PHASE 3)

EVALUATION
(PHASE 4)

OPS SUPPORT
(PHASE 5)

POST-INSTALLATION
(PHASE 6)

Figure 3. User Interface Activities During Phase 1

Phase 1. UI Preproposal/Proposal

Goals

- Form the UI Design Team, which is responsible for all UI tasks.
- Develop project usability objectives and plans.
- Identify and define the UI activities to be performed throughout the project life cycle.

Tasks

Management:

- Form the UI Design Team.
- Develop the Usability Plan (refer to “Usability” on page 37 for more information) to be integrated in the proposal (as required), the SE Technical Management Plan (TMP) and the Software Management Plan. (Refer to the *SID Systems Engineering Standards*, Manual 10-09 for more information about the TMP and *SID Software Standards*, Manual 33-09 for more information on the Software Management Plan.)
- Estimate human resource requirements and schedules.
- Identify risks and describe activities planned to minimize them as required for the Technical Risk and Performance Plan (TRPP). (Refer to the *SID Systems Engineering Standards*, Manual 10-09 for more information about the TRPP.)
- Identify and describe UI portions of all deliverable documents.
- Develop UI portions of Information Plan (refer to “User Documentation and Training Development” on page 74 for more information.)
- Establish UI configuration control procedures.

Requirements Analysis and Definition:

- Identify and describe activities required to define UI requirements.
- Begin requirements definition.
- Begin developing UI portions of the Operations Concept Document.

Design and Implementation:

- Support system hardware and software design decisions.
- Generate examples of candidate panels, online HELP, error messages, user documentation and training materials.

Products

Usability Plan

UI project/proposal activities resource estimates for proposal

UI portions of Proposal

UI Requirements for draft Systems Requirements Specification (required for SRR) and supporting rationale report

UI portion of draft Operations Concept Document (required for SRR)

UI portion of Information Plan

Sample panels, online HELP, messages, user documentation and training materials

Related Project Activities/Products

Trade studies for selecting hardware and software components

Proposal development

Requirements analysis for development of the draft Systems Requirements Specification

Additional Information Sources

- Phases 2-6 of these guidelines, which provide specific guidelines for performing the activities described in the Preproposal/Proposal.
- *SID Systems Engineering Standards*, Manual 10-09
- *SID Software Standards*, Manual 33-09
- Bibliography

Issues To Be Addressed

Management:

What are the usability objectives? How will these objectives be met? What are the dependencies and risks?

What are the UI products and supporting activities?

What end-user involvement in the requirements definition and implementation process is feasible and/or desired?

What actions can be taken to ensure early customer approval of the UI, such as Technical Interchange Meetings (TIMs), demos, working groups, prototyping, acceptance criteria, and acceptance testing of customer approved specifications?

Does IBM have the expertise in-house to perform the work or should subcontracting be considered? What are the resource estimates and schedule trade-offs?

What installation and operations support will be needed; e.g.; on-site, on-call or telephone support; time period; level of effort?

What standards (MIL-STD, IBM, etc.) in addition to those imposed by the customer, if any, should be implemented?

Requirements Analysis and Definition:

What are the goals of the new system, e.g., automating manual tasks, replacing an old automated system, or increasing system response time?

What are the inadequacies in the old system that have generated the need for a new system? What is satisfactory in the old system that should be preserved? What must be improved in the new system?

What changes will the new system cause to existing automated or manual processes?

How can differences in manual vs. automated performance of tasks, and old vs. new user interface with the computer be minimized to assure customer acceptance of the system?

What is the relationship between end-users and the contracting customer(s)? Will this impact design and/or customer/user acceptance?

What information about the users' skill levels, experience and other areas of diversity is needed?

What UI requirements can be derived from explicit and implied HW and SW requirements, e.g., types of workstations, types of dialog?

Which UI requirements are vague? How can these requirements be further refined into measurable, testable criteria?

What types of analyses (task, workload, decision/action, etc.) will be required to define users' requirements?

What types of tasks will be performed by the users? How often? For what purpose? Can they be modified? Is there a benefit to be realized by modifying or automating users' tasks?

To what extent is error recovery time-critical?

How will standards such as MIL-STDs, DOD Standards and others impact design decisions?

Design and Implementation:

What level of automation is acceptable/desirable/feasible to the user within the bounds of current technology?

Are the UI requirements and user desires achievable with current technology?

Are requirements being imposed that do not make sense or are inconsistent with good design?

Do the UI requirements overly constrain the choice of hardware?

What types of input/output devices are needed or preferred?

What types of dialogs would be appropriate?

What are the trade-offs between developing an effective interface, one that is self-explanatory and does not distract from the task at hand, and relying on the information contained in the user documentation, on-line HELP, and training to provide this information?

Risks/Concerns

The following key UI risks/concerns must be addressed during proposal preparation.

- UI Resource Requirements Estimates
- UI Configuration Control
- User Documentation Development
- User Reviews/Participation

UI Resource Requirements Estimates

Multi-disciplinary ownership of UI activities, while essential to successful UI development, increases the potential for omission and/or duplication of estimated resources. Incomplete understanding of the UI development process may also contribute to incorrect resource estimates. Systems Engineering (SE) should develop a UI product-oriented Work Breakdown Structure (WBS), which incorporates the UI Design Team approach and includes human resource estimates for all disciplines. This should reduce the potential for incorrect estimates. Include estimates for SEVERAL ITERATIONS of UI design and corresponding technical reviews before reaching customer/IBM agreement on requirements and design. Include someone with project UI experience in early proposal reviews.

UI Configuration Control

Configuration control of the UI technical baseline including design changes, change history, panel development, error messages and UI documentation is critical to maintaining a common and consistent user interface design throughout system development. This prevents inconsistencies in the baseline design as it evolves and ensures efficient use of resources. Effort for development and implementation of these control procedures must be included in all SE resource estimates. (See Phase 1, “UI Preproposal/Proposal” for more information on configuration control.)

User Documentation Development

User documentation should be developed in parallel with the UI design, NOT after the design is complete. User documentation can help identify problems in the UI design before it is “frozen” by providing a vehicle for user comments. Also, difficulty in describing a function may be an accurate forecast of difficulty in performing it. Writers of the user documentation have a broad view of the system that may be unique, especially on large projects, and they can identify consistency and other issues of global concern. (See “User Documentation and Training Development” on page 74 for more information on user documentation.) Several delivery cycles for customer reviews should be estimated, as well as internal reviews and validation. If interim software release deliveries of the system are required, identify the unique requirements for those deliveries in addition to the requirements for the final system delivery.

User Reviews/Participation

Early user involvement in design discussions, prototype evaluations, and document and training material reviews is critical to the development of UI requirements which satisfy the users needs, meets the system’s requirements, and prevents costly revisions later in the development process. This involvement may require additional time in early phases since UI development is an iterative process. Failing to estimate the magnitude of this effort correctly in the proposal may create overruns later. User participation should be planned, scheduled and the required resources estimated during the proposal phase. This information should be contained in the proposal, in addition to identifying and defining the customer’s responsibility for providing users for these activities.

Users often resist changes to a familiar system, especially when the rationale and benefit to them is not apparent. The UI Design Team will be challenged with questions, comments and requests for change. Neglecting to address these concerns may result in rejection of the interface and potentially the entire system. Proposal estimates must allow for resolution of these issues and time for working group meetings, technical reviews, site visits and demos.

(This page intentionally left blank.)

UI Preproposal/Proposal Guidelines

Consider the activities listed below when estimating the UI development efforts for a proposal. The proposal should respond to the RFP, and instill confidence in the customer/users that the IBM development process will provide a successful system with efficient use of resources. Phases 2 through 6 of this document provide detailed discussions of these activities and should be reviewed during proposal development to assess the resources required for proposal estimates.

Management

1. Form a UI Design Team (UIDT), consisting of members from the disciplines listed below. (Refer to “UI Design Team Composition” on page 33 for detailed information on the responsibilities of the UIDT.)
 - Systems Engineering (SE)
 - Software Engineering (SWE)
 - Hardware Engineering (HWE)
 - Human Factors Engineering (HFE)
 - Integration and Test (I&T)
 - Quality Assurance (QA)
 - Integrated Logistics Support (ILS)

Representatives from Reliability/Maintainability/Availability (R/M/A) and Information Development (ID) should be included, as appropriate. A user representative, subcontractor(s) associated with the user interface development and the Subcontract Acquisition Manager (SAM) should also be requested to attend meetings, as appropriate. The user representative will not attend all UIDT meetings since s/he has no direct responsibility in the development process. The user will be requested to attend those meetings where his/her presence may assist in clarifying or resolving design issues.

The UIDT will provide technical and management guidance necessary to successfully implement the UI development process. For the proposal, the UIDT identifies and defines the UI products and activities necessary to support other system products (i.e., TMP, System Requirements Specification, Ops Concept document, Test Plans and Procedures, etc.). It also generates the Usability Plan (refer to step 3).

UI development is *not only* a front-end activity, but continues to the end of a project (refer to the activities identified in Phases 2 - 6). Be sure to estimate resources for UI effort through system acceptance. Projects with multiple software releases, Design Action Items (DAIs) and Program Trouble Reports (PTRs) require UI effort to interface with users and to monitor compliance with UI requirements throughout the entire project life cycle.

(ALL DISCIPLINES RESPONSIBLE)

2. Assure all UI products and their required efforts are identified in the end product Work Breakdown Structure (WBS). (For more information on the WBS see *SID Corporate Bulletin: Systems Engineering Work Breakdown Structure* (C-B 0-4010-002).)

(SE RESPONSIBLE)

3. Develop the Usability Plan. This plan includes measurable, high level usability objectives for the system, the evaluation criteria and measurement methods to ensure the objectives are met, a schedule for the UI development that conforms to the project schedule and milestones, the risks and key dependencies which may impact UI development, and a transaction matrix which identifies the information each discipline must provide the person responsible for the product or activity. The objectives in this plan

are incorporated in the proposal. Once the contract is awarded, the plan is folded into a section of the SE Technical Management Plan (TMP). (Refer to “Usability” on page 37 and “Usability Checklist” on page 46 for more information.)

(SE RESPONSIBLE)

4. Develop Information Plan(s) which include objectives, requirements, dependencies, assumptions, schedules, status, and evaluation criteria for all project technical and maintenance documentation (including all UI documentation) and training. This information may be required in specified formats on some DOD contracts, or as an internal planning and tracking document on others. Plan to validate user documentation accuracy as part of system test. (Refer to “User Documentation and Training Development” on page 74 for more details on Information Plan(s).)

(ILS RESPONSIBLE)

5. Specify prototyping activities and user evaluations. User reviews and comments can add considerable time to the UI effort, but time saved by detecting problems early is small compared to the time required to fix problems later, as shown in Figure 4 on page 32. Iterative design (as described in step 2 under “Design and Implementation” on page 92) including user evaluations, is a key element of the UI methodology described in these guidelines. Therefore, plan accordingly for these activities in resource estimates and project schedules.

(SE AND HFE RESPONSIBLE)

6. Establish Configuration Control (CC) procedures to maintain a single UI baseline configuration and track UI product changes throughout the development process. The procedures will apply to all UI products during design and development, including:

- UI system requirements and rationale
- UI system design description and rationale
- UI software requirements and rationale
- UI software design guidance
- Panels (design and code)
- Messages (design and code)
- Online HELP (design and code)
- User documentation
- Training materials

The responsibility for ensuring these procedures are followed is shared by different members of the UIDT. For example, SE and ILS are responsible in Phases 1 and 2. SWE also shares the responsibility in Phases 3 and 4 when code development is in progress.

These procedures should include:

- Establishing and implementing baseline control procedures prior to initiation of formal configuration management.
- Establishing a person accountable for maintaining control of each document and database.
- Coordinating every baseline release and change with appropriate technical discipline personnel.
- Establishing location and name of central repository for all data.
- Establishing a data bank scheme that can feed data into user documentation without rekeying the data.
- Selecting development products/tools which can support requirements documentation, user documentation, and training materials.
- Establishing levels of files for “approved” and “working” versions of UI products and instituting controls for converting “working” levels to “approved” levels.
- Establishing inspection procedures for design and development of UI products.

- Identifying measures to prevent unapproved configuration changes from entering the system.
- Establishing conventions for tracking change history, such as adding comments/remarks to files explaining modifications/updates to existing data (including date changed, initials of responsible person, and reason for change).
- Establishing naming conventions for all data to ensure multiple releases can be maintained concurrently.
- Providing documentation and training for configuration control procedures.
- Maintaining a history or rationale for requirements and design considerations, evaluations, and recommendations. This information is included or supplements the Requirements Rationale Report (RRR), the Design Rationale Report (DRR) and the Design Guidance Document (DGD) maintained by SE for system and software requirements. (Refer to *SID Systems Engineering Standards*, Manual 10-09 for information on the RRR.)
- Applying formal Configuration Management (CM) procedures to all customer approved (authenticated) baseline documents and databases or to all completed documents and databases by major milestones (i.e., SRR, SDR, PDR, CDR).

(SE AND ILS RESPONSIBLE)

Requirements Analysis and Definition

1. If a clear profile of the system user(s) is not provided in the RFP, propose methods to obtain this information prior to definition of user requirements. An aid for collecting user profile information is provided in Phase 2, "UI Concept Development." This aid is a sample questionnaire that can be tailored for specific projects.

(SE AND HFE RESPONSIBLE)

2. Search through all documents, such as the Operational Need Document (OND), Request for Proposal (RFP) and Statement of Work (SOW) to identify all explicit and implied UI requirements. Make a list of all UI requirements, using paragraph number for identification and traceability, and include an interpretation for each requirement. A well written proposal maps exactly to the RFP, paragraph by paragraph.
 - a. In the RFP, there may be a separate UI section with explicit requirements; explicit requirements are expressed as "shalls". Every "shall" must be addressed in the proposal, tracked through development, and compliance demonstrated at system sell-off.
 - b. Other relevant UI requirements may be found throughout the RFP, especially in the functional SW and/or HW requirements sections (e.g., hardware requirements for panning, zooming, or windowing imply certain UI requirements).
 - c. UI requirements may be implied or only vaguely defined. UI requirements may be found by searching for USAGE statements. Any interaction between the user and the computer is a UI requirement. Look for:
 - What the panels will look like.
 - How the user will know what to do.
 - When the user will have to know or do something.
 - Performance requirements involving tasks that include user action as well as system performance.

- Types and skill levels of users.
- What reports and other hardcopy printouts will look like (if required)

(SE AND HFE RESPONSIBLE)

3. Avoid vague, ill-defined, cliché UI terms, such as “user-friendly”, “easy-to-read”, “expedient”, “state-of-the-art”, even if these are the terms used in the RFP.

(ALL DISCIPLINES RESPONSIBLE)

4. The UI sections of the proposal contain the usability objectives and the UI requirements derived from the OND, RFP and SOW. (Refer to “Relationship Between Usability Objectives and UI Requirements” on page 44.) These requirements should also be included in the Human Performance/Human Engineering section of the draft System Requirements Specification (section 3.2.9 in a DOD-STD-2167 formatted specification). Supporting rationale for the UI system requirements will be documented in the Requirements Rationale Report (RRR). The UI requirements will be further decomposed into the software UI requirements and will be documented in the Software Requirements Specification (SRS). The proposal should describe the system and software specification documents and should include resource estimates to develop these documents and monitor compliance with the UI requirements.

(SE RESPONSIBLE)

5. Assess the testability of all proposed usability objectives and UI requirements. Avoid risks such as proposing to ensure compliance with some vague standard of user-friendliness without definition. In the Test section of the proposal, describe UI test procedures and how acceptance criteria will be determined. (Refer to “Usability” on page 37 for more information on UI testing.)

(I&T, SE AND HFE RESPONSIBLE)

6. Assure all requirements for referenced MIL-STDs or other standards are identified and addressed in the proposal. Many UI requirements in MIL-STDs are vague; compliance is not easily demonstrated unless the requirements are defined and tailored to the proposed application.

(SE AND HFE RESPONSIBLE)

7. Evaluate the OND for information on the operators’ concept of operations (also known as Ops Research). This information will be included in the Operations (Ops) Concept document. The Ops Concept document describes the operators, tasks and goals of the system, from the users’ perspective. (See Phase 2, “UI Concept Development” for more information on developing this document.) Include resource estimates for visits to customer sites, if required.

(SE AND HFE RESPONSIBLE)

*NOTE: Performing operations research in support of the Ops Concept document development is critical to the UI design process, and is required per SE Technical Management Practices. The ops research process provides specific information, such as how the system will be used, by whom, importance of tasks performed, and frequency of tasks performed. **The importance of going through this process of defining the system from the users’ perspective and confirming our understanding of it with the users cannot be overstated.** Proposed resource estimates should include this process.*

8. Identify and define the types of studies and analyses (function, task, timeline, decision/action, workload, critical incident, activity, etc.) needed to specify user requirements.

(HFE AND SE RESPONSIBLE)

9. Determine how users will participate in the review process, e.g., hardcopy, online, forum, or conference disks.

(SE AND ILS RESPONSIBLE)

Design and Implementation

1. Describe the activities of SE, HWE, SWE and HFE in selection of candidate hardware and software (e.g., trade studies) which impact UI design decisions. These trade studies should be formally documented as they may be factors in the sell-off process and can be a source of historical information in phases.

(SE, HFE, SWE, AND HWE RESPONSIBLE)

2. Provide candidate examples of draft panel hierarchies, panels, online HELP, and error messages. Identify and describe the types of documents that will constitute user documentation such as *Users' Manuals, Position Handbooks, Programmers' Manuals, Maintainers' Manuals, etc.*, and their presentation vehicle (hardcopy, online, both). Discuss how the effectiveness of the UI, in conjunction with online HELP and user documentation, can minimize the training required.

(SE, HFE, SWE, AND ILS RESPONSIBLE)

UI Design Team Rationale, Composition & Responsibilities

UI Design Team Rationale and Responsibilities

The UI Design Team (UIDT) is a working group composed of representatives from all disciplines involved in the UI development process (refer to “UI Design Team Composition” below). This team functions like the Interface Control Working Groups used by SE to design and control hardware and software interfaces (internal and external) and the Engineering Integration Workshop (EIW). The UIDT objective is to ensure the system provides a user interface which satisfies the users’ needs and meets the system’s requirements. Its purpose is to achieve the objective by providing technical and management guidance necessary to successfully implement the UI development process. The UIDT responsibilities include:

- Identifying UI activities/products and assigning UI responsibilities
- Planning and scheduling UI activities and interdependencies
- Identifying and specifying UI requirements
- Establishing the UI design
- Ensuring the delivered system complies with UI requirements

The UIDT delegates responsibility to the representative of the discipline having the necessary expertise to accomplish the task. This results in a single person being accountable for the UI activity or task and allows for efficient management. All activities or tasks identified in these guidelines are managed by the UIDT.

The formation of a UIDT, with the common goal and responsibility for UI development reinforces the necessity for cooperation between the interdependent disciplines. Additionally, involvement of a user representative (as an invited participant for specific meetings) may identify concerns early in the design process. Many disciplines will be involved to some degree throughout the life of the project.

The UIDT has the authority to approve or disapprove major UI products and decisions, and develop procedures by which approval is obtained. A UIDT member, probably the SE representative who assumes technical lead, will serve on the Configuration Control Board (CCB) and will signoff on all relevant Document Coordination and Approvals (DCAs) to assure UI representation to project coordination and approval activities. The SE technical lead should also maintain control to keep the group focused on its charter responsibilities.

UI Design Team Composition

The UIDT is made up of members from the following disciplines:

- Systems Engineering (SE)
- Software Engineering (SWE)
- Hardware Engineering (HWE)
- Human Factors Engineering (HFE)
- Integration and Test (I&T)
- Quality Assurance (QA)
- Integrated Logistics Support (ILS)

Representatives from Reliability/Maintainability/Availability (R/M/A) and Information Development (ID) should be included, as appropriate. A user representative, subcontractor(s) associated with the user interface development and the Subcontract Acquisition Manager (SAM) should also attend meetings, as appropriate.

Discipline Responsibilities

The responsibilities of each discipline are listed below and shown graphically in Figure 5 on page 36. UI activity and product ownership is critical to achieving a successful user interface. While one discipline may have primary responsibility for a product, several disciplines may share technical responsibility for the UI activities that support that product.

Systems Engineering (SE)

The SE representative is responsible for the Usability Plan; UI requirements; configuration control procedures; designing and prototyping panels, messages and online HELP; and usability evaluations.

The SE representative assumes the UI technical lead position and chairs the UIDT. This position will require expertise in SE, HFE, ID and customer interface. This ideal skill composite is rare, therefore complimentary resources must be ensured within the UIDT. Since the SE manager is responsible for the overall management of the UI process, the UI technical lead reports to the SE manager. The UI technical lead will be referred to as the SE(UI) in these Guidelines.

The SE(UI) also assigns action items; organizes UIDT meetings; represents the UIDT at CCB and DCA; establishes schedules and monitors progress of all UI activities; and identifies and tracks UI concerns, issues and solutions. SE(UI) is the primary technical interface to the customer for UI issues.

Software Engineering (SWE)

The SWE representative is responsible for reviewing all UI products and schedules for implementation feasibility; coordinating the integration of UI requirements into SW Product Specifications, UI (display & control) code, panels, messages, and online HELP; and configuration control procedures for all SWE generated UI products. This representative must be responsive to both user concerns and other SWE functions so that user needs are satisfied across the entire system. On large projects, SWE representatives from all functions may be needed to support some UIDT working meetings since each function may have unique UI needs.

Hardware Engineering (HWE)

The HWE representative is responsible for reviewing all UI products for HW requirements components, identifying and tracking issues relative to HW configurations, and for coordinating the integration of UI requirements into HW Product Specifications and equipment development.

Human Factors Engineering (HFE)

The HFE representative is responsible for performing user profile, task and other human factors analyses; designing and prototyping panels, messages, and online HELP; performing usability testing; conducting user evaluations; and identifying human factors concerns and issues.

Integration and Test (I&T)

The I&T representative is responsible for reviewing schedules for feasibility, for developing and reviewing UI products such as UI Test Plans and Procedures for testability, and identifying I&T concerns and issues.

Integrated Logistics Support (ILS)

The ILS representative is responsible for generating user documentation and training plans, schedules, and products, and supports the development of the Ops Concept document.

QA

The QA representative identifies quality concerns and monitors/reviews all UI products for conformance to quality standards.

User

The user representative identifies usability concerns in the design and assists in resolution of other concerns. The user also reviews UI products for responsiveness to requirements. If feasible, this representative will serve as the focal point for all user(s) comments and concerns. This representative will not attend all UIDT meetings since the user has no direct responsibility in the development process; the user will be requested to attend those meetings where their presence may assist in the clarification or resolution of design issues.

NOTE: While user involvement is necessary, there are associated risks. One user's "wish list" may be in conflict with customer requirements and may not represent the needs of the entire user community.

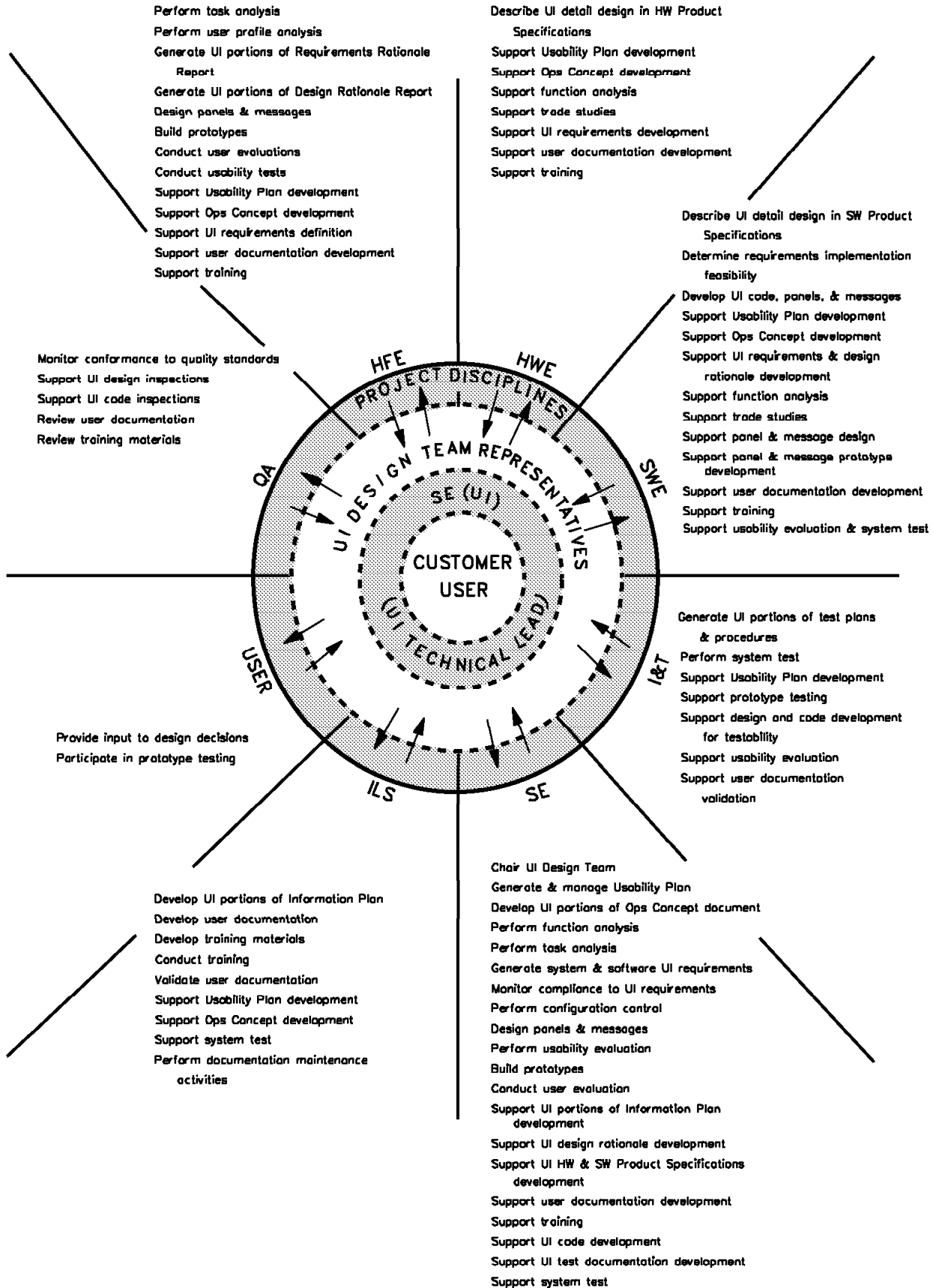


Figure 5. UI Responsibilities and Interfaces by Discipline

Usability

If a system meets all requirements, but the users cannot learn the system easily and perform their task(s) with a minimum number of errors, the system is not “usable” and will not be accepted by the users. “Usability” is defined as ease of learning and use. A plan which ensures usability is engineered into the UI design must be generated at proposal time and refined during system development as the users’ requirements are further defined. Requirements for panels, interaction techniques, error recovery, and user documentation evolve from this plan. Usability planning focuses on development of and adherence to testable, measurable criteria that will be used throughout the system development process to evaluate usability, demonstrates conformance to requirements for system sell-off, and demonstrates management commitment to usability.

Usability Plan

The Usability Plan contains the system usability objectives, usability evaluation/measurement methods, schedule, risks, and key dependencies. There is no established format for the plan, and brevity is preferable. A suggested outline for a Usability Plan is provided in Figure 6 on page 40. The Usability Plan demonstrates a commitment to manage for usability; the usability objectives expressed in the plan become the basis for specific system usability requirements. (Refer to “Relationship Between Usability Objectives and UI Requirements” on page 44 for more information about how usability objectives become the foundation for system and software UI requirements.) The plan must be developed during the Preproposal/Proposal Phase and may be included in the Proposal, along with other provided or assumed UI requirements.

Once the contract is awarded, the plan is integrated in the SE Technical Management Plan (TMP) and the Software Management Plan. The TMP becomes the vehicle for managing usability throughout the project life cycle. Since the TMP provides a format for discussion at Management Status Reviews (MSRs), the Usability Plan will be examined periodically to ensure usability objectives are met. Software Usability Plan milestones are integrated in the Software Management Plan.

At a minimum, the Usability Plan should contain the following information.

Usability Objectives

Usability objectives are derived from explicit and implied UI requirements in the RFP, SOW and/or OND, imposed standards (e.g., MIL-STDs and DOD Standards) and other standards (e.g., Mitre Guidelines and SAA/CUA). (For more information on Mitre Guidelines, refer to “Mitre Guidelines” on page 70. For more information on SAA/CUA, refer to “SAA/CUA” on page 71.) Usability objectives are drafted by the UIDT in Phase 1, and refined with support from the users in later phases. Two examples of usability objectives are:

Users will not need documentation to perform tasks on the system.

Context sensitive HELP will always be available on-line.

Usability Evaluation/Measurement Methods

Usability Evaluation Methods: The following usability evaluation methods may be employed to prove that a system is meeting and has met the usability objectives:

- Usability Testing
- Iterative User Evaluations
- Sell-off Testing

Usability testing provides a method to evaluate a system’s ease of use based upon specific test criteria. Five basic methodologies for formal usability testing are formalistic, iterative, naturalistic, field study, and survey. These methods are employed by Human Factors Engineers, using controlled procedures.

Iterative user evaluations, like formal iterative usability testing, are performed with users on representative tasks or prototypes at every level of development to identify and resolve problems in the UI design. These evaluations are usually conducted by SE and HFE with support from I&T. Iterative user evaluations are a key component in the UI development process and are presented in these guidelines as an effective means of ensuring usability.

Sell-off testing is performed by I&T to prove compliance with requirements based on successful execution of the system test plans and procedures.

Usability Measurement Methods: Usability evaluation and measurement criteria are used to plan and analyze the results and measurements of tests. Agreement with the customer as to explicit acceptance criteria should be established and documented in the Usability Plan as early as possible. Usability, being intangible, is not directly measurable, so observable measures or behaviors must be selected which constitute a valid measure of usability. These measures can be objective or subjective. Some objective measures are:

- Percentage of users who successfully complete task
- Speed of task performance
- Number of user errors per task
- Amount of time to recover from errors
- Number of panels and/or steps in task
- Frequency and amount of time spent using HELP or documentation
- Percentage of tasks completed without errors
- Time to learn specific functions
- Retention of commands over time

Some subjective, unmeasurable but recordable criteria include: likes, dislikes, usage, problems, and productivity rating. Subjective evaluations by users or user-like groups can provide valuable information that should be included in the evaluation. Customer/IBM agreement on a range of acceptable values can be applied to subjective criteria for sell-off (e.g., 10 out of 12 users will be satisfied with online HELP).

Online HELP and hardcopy or online user documentation should be tested for usability concurrent with prototype and system testing. Parameters to be analyzed should include format, functionality, readability, retrievability, and style.

Sample User Evaluation Procedures: Procedures similar to the ones presented below could be used for iterative user evaluations or incorporated into system test plans and procedures for sell-off testing. During sell-off testing, these procedures could be used to verify measurable UI requirements when performance objectives and acceptance criteria have been agreed to by the customer and documented. (For example, 4 out of 5 users shall be able to successfully complete task X in 3 minutes or less with no errors.) User evaluations should take into account increased proficiency as users gain experience on the system. The evaluations should strive to reflect typical operations and hardware environment(s).

NOTE: HFE responsibility for designing the iterative user evaluations and analyzing the results is important to ensure validity and reliability of the findings.

1. Define test cases and acceptance criteria, with customer concurrence, if customer approval of results is required or appropriate.
2. Select and isolate a few functions/tasks representative of the area or type of interaction of interest. Prototypes can be used early in the project, or actual system functions later at sell-off.
3. Provide short but adequate training for the selected users. (If automated error tracking or time-tagging mechanisms are not being used, the group should be small enough for testers to observe every user action -- maybe only one at a time.)
4. Ask the user(s) to perform the task(s).

5. Track number of errors per task, time to complete each task successfully, time to complete subsequent similar tasks, number of references to HELP or documentation, or other variables, depending on the criteria being measured.
6. Ask users for their overall reaction to the system (subjective measure).

Repeating the exercise at later times will provide comparative data about the user's ability to learn and retain data as well as information about how easy the system is to learn, reduced memory load, etc. Data collected in this very simple user evaluation provides valuable information about the UI criteria being examined. Additional subjective but valuable information can easily be obtained during these tests, by asking the users to talk to themselves while working their way through the tasks. Sources of frustration and confusion can be identified by tracking these comments against the panel displayed at the time, or against the user action required or performed at the time of the comment.

In addition to the measurable criteria above, the questions presented in "Usability Checklist" on page 46 can be tailored to the application and presented to the users as a brief questionnaire, as another simple way of collecting feedback from users.

Schedule

Schedules should be provided in the Usability Plan which include all key UI activities, their interdependencies, dates, and milestones. Any schedule changes should include explanatory information or history in case justification is required later.

The information and format used in Figure 2 on page 7 may be used as the basis for the schedule, as shown in Figure 7 on page 43.

Risks

Any risks to successful implementation of the UI should be identified in this section of the Usability Plan. Whenever possible, contingency plans should be prepared. Notify the Program Manager of these issues as they emerge.

Key Dependencies

Dependencies can include support from other IBM disciplines, customer activities or approvals, availability of products, success of prerequisite functions, etc. This section should also include a transaction matrix which identifies the information each discipline must provide the person responsible for the UI product or activity.

NOTE: The Usability Plan/TMP is the vehicle by which usability is managed throughout the project life. The objectives, measurement methods, schedule, risks, and dependency issues must be addressed by project management and the UIDT.

Usability Plan

This outline is provided as a guide for generating a Usability Plan. It also contains examples of the components.

I. Usability Objectives

The following usability objectives have been identified for the TBS system:

A. Panels

1. Consistency will be maintained for terminology, display format and syntax.
2. Standard English will be used for all online and hardcopy text.
3. Two levels of dialog interaction will be provided (e.g., menu selection and command language).

B. Messages

1. Error messages will provide information and recovery procedures.

C. User Documentation

1. Online HELP and tutorials will be provided.
2. User documentation will be provided for every function.
3. Consistency will be maintained for terminology, display format and syntax.

D. Performance

1. Feedback will be provided for all user actions.
2. The interface will be tailorable by the users.
3. Checking logic will be provided to prevent critical user errors.
4. A fastpath for experienced users will be provided.
5. Report generation capability will be provided.

E. Input/Output Devices

1. Mouse and keyboard interaction devices will be supported.

Note that these are very high level measurable objectives.

The manner and extent to which they will be implemented is not provided as it probably will not be determined at the time of the proposal.

Figure 6 (Part 1 of 3). Sample Usability Plan Outline

II. Usability Evaluation/Measurement Methods

The usability evaluation and measurement methods that will be employed to test the usability of the TBS system are:

A. Methods

1. Rapid prototyping sessions with users
2. Iterative prototype testing by users and selected "user-like" representatives
3. User and internal IBM paper reviews of panels, messages, HELP and user documentation
4. Online validation of user documentation against baseline system
5. Formal inspection testing of completed baseline system
6. Formal demonstration testing of completed baseline system

B. Criteria

1. Representative task(s)
2. Numbers of errors
3. Number of participants to complete task with no errors
4. Time to complete task(s)
5. Number of calls to online HELP
6. Number of references to user documentation
7. Participant satisfaction

III. Schedule

Following is the schedule of major UI activities, milestones and interdependencies. (Figure 7 on page 43 is an example of a Usability Plan schedule.)

IV. Risks

The following risk areas have been identified for the TBS system development:

A. Risks

1. Need ample opportunity to observe and talk with users
2. Hardware constraints to use the XYZ terminal may conflict with UI requirements for input devices (e.g, joystick)
3. Performance and UI trade-offs will have to be assessed
4. Scope of user documentation and training may be impacted by unplanned growth of user new hires

B. Contingency Plans

Figure 6 (Part 2 of 3). Sample Usability Plan Outline

V. Key Dependencies

The UI development process is dependent upon the following people, events, or activities:

A. Dependencies

1. Customer Dependencies

- a. Prototyping with user personnel
- b. All approvals on deliverable documents must be received within 30 days of delivery.

2. Product Dependencies

- a. Customer's Ops Concept document must be received by MM/DD/YY.
- b. Availability of users for collection of user profile information.
- c. Availability of new products.

3. Support Activity Dependencies

- a. Installation of hardware for training.
- b. Availability of block time for system test.

B. Contingency Plans

Figure 6 (Part 3 of 3). Sample Usability Plan Outline

MONTHS AFTER CONTRACT AWARD

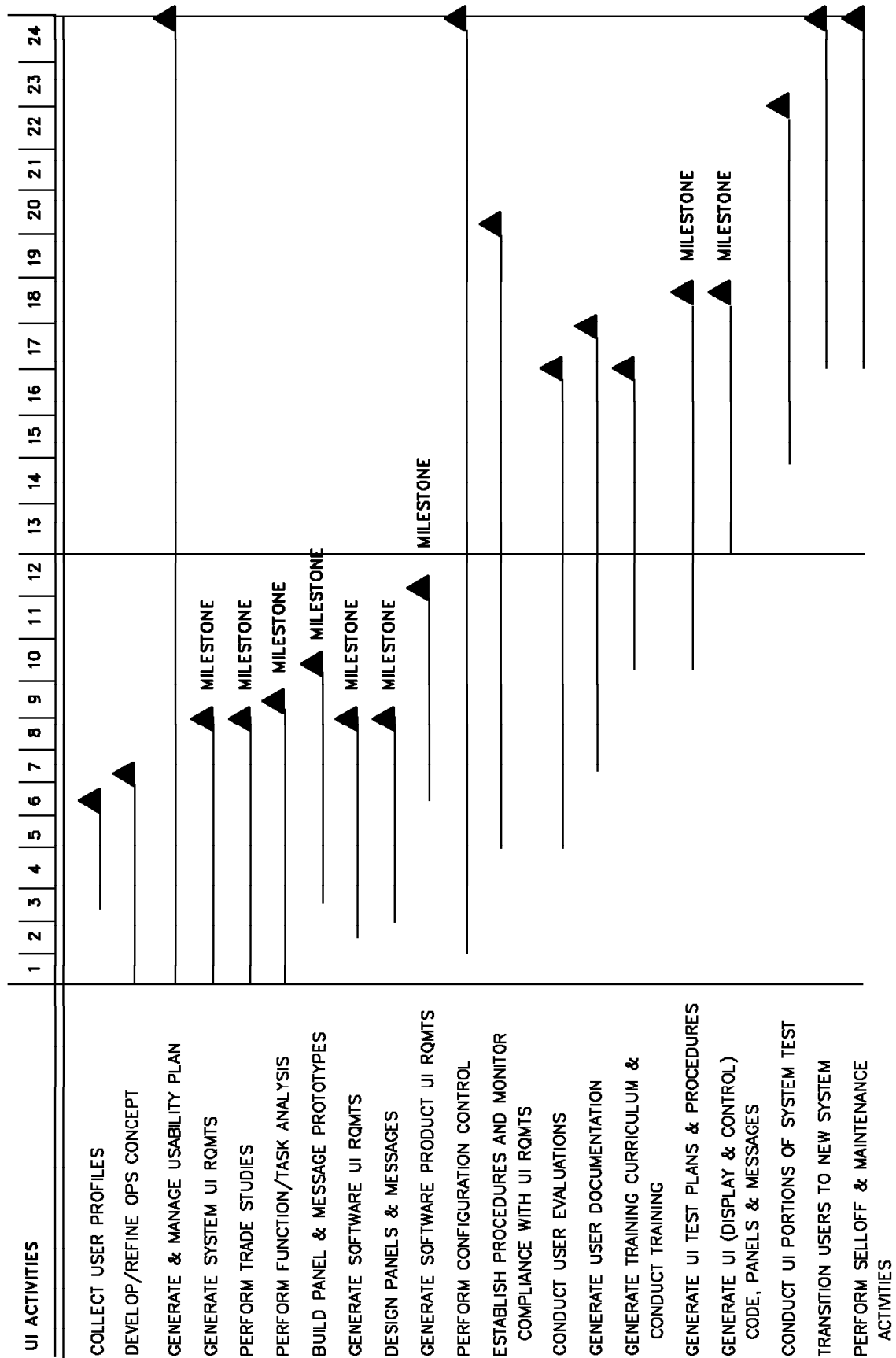


Figure 7. Sample Usability Plan Schedule (Interdependencies not shown)

Relationship Between Usability Objectives and UI Requirements

Each usability objective is rephrased and specified as one or more system requirements (shall statements). Each system requirement is then allocated to more detailed requirements as the design progresses. This process, allocating UI objectives to requirements and re-allocating requirements at various design levels is discussed in the following paragraphs and demonstrated in the examples presented in Table 3 on page 45.

Usability Objectives

The usability objectives are expressed as design objectives. These objectives may be generic and apply to many different types of interactive systems, but they are measurable, testable, and easily tailored into specific requirements for the application being developed. Usability objectives are derived from explicit and implied UI requirements in the RFP, SOW and/or OND, imposed standards (e.g., MIL-STDs and DOD Standards) and other standards (e.g., Mitre Guidelines and SAA/CUA). (For more information on Mitre Guidelines, refer to “Mitre Guidelines” on page 70. For more information on SAA/CUA, refer to “SAA/CUA” on page 71.) See the examples in Table 3 on page 45.

System UI Requirements

The usability objectives are rephrased (as required) as “shalls” in the System Requirements Specification. There is a one-to-one correspondence between the “shalls” in the System Requirements Specification and the usability objectives, although additional project specific UI requirements may be needed as the operational concept is developed. These system UI requirements are developed during Phase 1, “UI Preproposal/Proposal” and Phase 2, “UI Concept Development” of the UI development process. See the examples in Table 3 on page 45.

Software UI Requirements

Software UI requirements are derived from and traceable to the system UI requirements in the System Requirements Specification. These software requirements are developed during Phase 2, “UI Concept Development” and Phase 3, “UI Requirements Definition and Design” of the UI development process. UI design standards such as the Mitre Guidelines and SAA/CUA, user profiles, user characteristics, and system UI requirements are used to develop software UI requirements. These requirements are documented in the Software Requirements Specification and their allocated detail design requirements in the Software Product Specification. The UI requirements are enforced via the compliance procedures established in Phase 2. As seen in the Table 3 on page 45, these specific requirements describe panel and message design sufficiently for final design and implementation.

Table 3. Usability Objectives and UI Requirements Examples		
USABILITY OBJECTIVE	SYSTEM UI RQMT	SOFTWARE UI RQMT
Error checking will be provided to protect the user from irreversible, non-recoverable errors.	User confirmation shall be required to protect the user from irreversible, non-recoverable errors.	All DELETE FILE requests shall display an interim panel with the following message: FILE *** IS ABOUT TO BE DELETED. PRESS PF11 TO CONFIRM DELETE. PRESS PF2 TO CANCEL DELETE.
Online assistance for the user will be provided.	Context sensitive online HELP shall be provided for every field on every panel.	Every panel shall provide context sensitive HELP on a pop-up window upon user request. The user request shall be a mouse click, first on the HELP icon and then on the selected field. All fields shall be selectable for HELP.
System will provide context information to the user at all times.	The system shall identify the user's location in the panel hierarchy at all times.	Current level and previous level function icons shall be displayed on the top right of every panel positioned at characters 63 through 72 on lines 2 and 3. The icons shall be separated by a slash, "/".

Usability Checklist

(Updated since original publication)

This checklist highlights usability concepts that should be addressed throughout the design, development and evaluation of the user interface. These concepts may provide the foundation for measurable usability objectives, project standards and testable requirements when tailored for the application and quantified, if necessary.

1. Few concepts are required to perform basic tasks and few calls to HELP or references to user documentation are required to learn the system at the basic level.
2. Informative feedback is provided for all user actions (e.g., PRINT REQUEST BEING PROCESSED).
3. HELP is always available.
4. HELP is context specific. The user is provided with help for a specific problem; searching for information is not required.
5. User documentation is clear, concise, complete, readable and task-oriented. Subject matter is easily located and information is consistently presented, echoing screen terminology and formats.
6. Consistency is maintained for terminology, abbreviations, display formats and syntax. (Screen elements are located in the same place, and presented in the same manner on every screen. PF key assignments are consistent throughout the entire system including HELP.)
7. The interface is flexible and can be tailored by the users to their skill level; or options are provided for different skill levels, such as menu selection and command language dialogs or fastpaths.
8. A minimum of memorization by the user is required to perform tasks.
9. The system provides all possible data, minimizing the need for data entry.
10. Checking logic is included to prevent critical user errors (e.g., PRESS PF11 TO CONFIRM DELETE REQUEST).
11. Error messages provide complete information and identify or describe recovery procedures.
12. Operational impact (idle or wasted time and operations) to recover from errors is kept to a minimum. The user does not need to seek additional information off-line.
13. System is forgiving and reduces the users' ability to make mistakes. (Most common user errors and actions are anticipated; system either assumes user intention, requesting confirmation of an assumption, or speeds the user through recovery actions, rather than issuing an error message.)
14. Screens are simple, uncluttered and self-explanatory.
15. Instructions and prompts are clear, complete and written in the imperative voice (e.g., PRESS ENTER, rather than ENTER SHOULD BE PRESSED).
16. The use of negatives is avoided in prompts and messages.
17. The number of actions required to perform tasks is kept to a minimum.
18. The users always know where they are in the screen hierarchy (e.g., previous and next screens are obvious; function name is provided; screens have meaningful unique names/identifiers, etc.).
19. The screen sequence and dialog promotes formation of a system conceptual model (e.g., "messy desk" model).
20. Screen layouts are logically organized. (Similar tasks or actions are grouped together and presented in the order performed.)

21. Available vs unavailable functions are identified on every screen (e.g., inactive functions are "greyed out").
22. Computer jargon on screens and in messages is avoided (e.g., SYSTEM HALT - RE-IPL or SYNTAX ERROR).
23. Intended action(s) on screens is obvious. (Prompts, instructions, and visual clues are provided to guide users.)
24. Icon meanings are apparent and consistent.
25. Options/lists are presented in logical order (e.g., sequential, alphabetic, numeric) or task-oriented groupings.
26. Defaults are provided whenever possible to reduce user actions (e.g., most common or frequent requests/entries are filled in and available for one stroke acceptance or overtyping to change).
27. Screen types are distinguishable visibly by function (e.g., input screens look different than output screens).
28. Color is used consistently and redundantly with other visual clues (e.g., orange AND blinking signify an alarm state).
29. Color selection addresses the color sensitivity of all elements of the user population.

Phase 1 Checklist

This checklist highlights major issues addressed in Phase 1. Items 5-10 should be considered during proposal development and then verified in proposal review.

- 1. The UIDT was formed and representatives participated in resource estimates for all UI activities to be performed.
- 2. The Usability Plan including usability objectives, evaluation and measurement methods, schedules, risks and dependencies was developed.
- 3. UI portions of the Information Plan were developed.
- 4. Configuration control procedures for all UI products were established.
- 5. Each customer requirement had a proposed response.
- 6. The proposal included rationale (justification) for all deviations from the customer's UI needs/requirements and for any enhancements, options, or changes.
- 7. The proposal demonstrated an understanding of requirements.
- 8. The proposal was reviewed for clarity and completeness of the following UI elements:
 - Deliverable products and services
 - Test/Verification of each requirement
 - Schedule of UI activities/products
 - Rationale for selected configuration/approach (comparison/evaluation of other approaches, if required)
- 9. All referenced documentation (MIL-STDs, Data Item Descriptions (DIDs), DOD standards, etc.) was reviewed to assure compliance as specified in the proposal and required by the customer's RFP.
- 10. Sample panels, messages, online HELP, user documentation and training materials were generated in support of highlevel UI design decisions for the proposal.

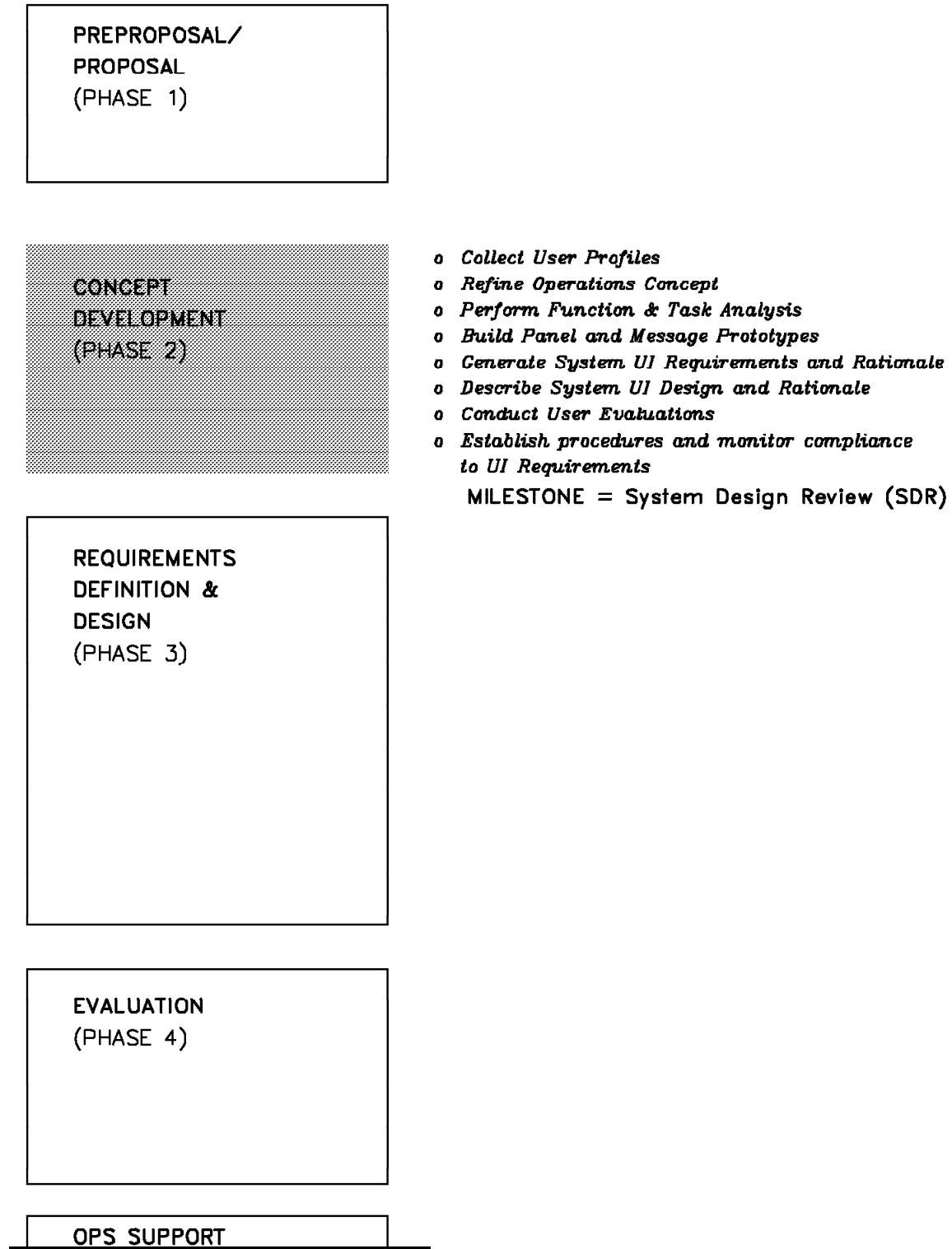


Figure 8. User Interface Activities During Phase 2