

User Interface Development Process

Phase 1: Scoping

1. Business Definition Document – A high-level product description
2. Project Plan
 - Staffing and Organization
 1. There should be a single decision-maker with final authority in order to avoid deadlocks and delays
 2. If possible, separate the UI designer and UI implementor roles; the two roles can conflict and separating them forces the designer and implementor to work on necessary compromises
 - Cost/Benefit Analysis
 1. A detailed analysis of the costs of time, personnel, and equipment, with an explanation of why each type of item is necessary
 - Planning
 1. List every User Interface task and activity
 2. The planning should be completed before other tasks are begun, but also be open to revision
3. User Profile
 - Determination of computer literacy, motivation to use, computer or relevant software experience, frequency of use, user turnover rate, and any other relevant factors.
 - Determination of what is more important: Ease of Use or Ease of Learning
 - Development of a user profile should precede the task analysis stage
4. Hardware/Software Definition
 - Determine the capabilities and constraints of the hardware and software relevant to this project (for example, operating system, computer hardware, expected uses of the software product in question)
 - Development of a hardware and software definition can be done in parallel with the user profile, but must come after the project plan, and before task analysis

Phase 2: Functional Specification

1. Task Analysis
 - Interview users about their activities relevant to the UI project
 - “Why” can reveal major goals
 - “How” can reveal the actions used to accomplish such goals
 - Determine work-flow patterns, conceptual frameworks, and other patterns that drive process that users go through
 - Tools to determine these issues can be interviews, questionnaires, usage statistics and others
 - Development of a task analysis must precede goal setting
2. User Interface Goal Setting
 - Absolute goals are quantifiable in some concrete fashion. Relative goals are generally perceptual acknowledgments of success
 - The goals will be used as a target to focus development efforts
 - Development of the UI goal setting must follow task analysis, but can be done in parallel with the training and documentation definition

3. Training and Documentation Definition

- A set of training and documentation tools developed based on user profiles, task analyses, and the decision to focus on ease of learning vs ease of use
- Can be classes, videos, on-line tutorials, etc
- Development of training and documentation tools must follow the UI goal setting and task analysis stages, but can be in parallel with the user interface mockup

Phase 3: Design

1. User Interface Mockup

- Doesn't need to be a real software mockup. It can be pen and paper or something else
- The purpose is to provide a high-level overview of the interface and it's navigation
- The mockup must demonstrate different classes of objects, their attributes and relationships, and the actions those objects take
- Cannot be done until previous stages are complete, and must come before other stages

2. Style Guide

- A high-level document describing generic user interface standards, rules, guidelines, and conventions. Covers the following:
 1. Screen layout
 2. Keyboard key functions
 3. Location and style of messages
 4. Standard use of terminology
 5. Color guidelines
 6. Standard mechanics for generic operations
- A sample table of contents might be:
 1. Introduction
 2. Overview of functionality
 3. User profile(s)
 4. Hardware constraints and capabilities
 5. Statement of usability goals
 6. Training and documentation
 7. Conceptual design
 8. Dialog styles
 9. Input and output
 10. Organization of functionality
 11. Screen layout and design
 12. Feedback
 13. User aid
- Development of a style guide must come after the mockup but before the detailed user interface design

3. Detailed User Interface Design

- The style guide should be fully communicated and understood. If necessary, a group could be formed to vet designs according to the style guide standards
- Style guides could also be enforced (as possible) by building the style guide standards into any toolkits or libraries to be used by the software
- Development of a detailed user interface design can be in parallel to general system architecture design, but must come after the development of the style guide

4. User Interface Prototype
 - A full-blown functional prototype. It doesn't need to be functionally complete, but capable enough according to the following:
 1. Core functions, or the most important functions
 2. Most problematic functions
 3. Functions most representative of the final product
 4. Functions most likely to be sequentially operated
 - Must follow the ideas detailed in the detailed UI design stage. Usability testing can be in initial development at this point, and application code specs can be developed in parallel
5. Prototype User Interface Test Plan
 - Identify core, high-frequency user tasks. These are specific user tasks with supporting test materials
 - Determine the types of data to be collected from these tasks
 - Determine the number and type of users to be tested
 - Schedule users for the first iteration of testing
 - Be prepared to amend the project plan depending on the results of the testing
 - Development of the plan can be done in parallel with the detailed user interface design or the user interface prototype
6. Prototype User Interface Testing
 - Each iteration should follow these steps:
 1. Define usability issues to be tested
 2. Design a test to collect data relevant to these issues
 3. Select and schedule test users
 4. Test the users and collect the data
 5. Analyze data and formulate recommendations for amending or updating the project plan or style guide as necessary
 6. Create a new prototype based on the recommendations
 - Techniques
 1. Structured observation – watching for repeated patterns as shown by multiple users during the tests
 2. Benchmarking – Testing the user interface against the absolute goals determined earlier to see how well performance is measured
 3. Comparison tests – Testing multiple designs to compare against each other. Also test objective results of performance and subjective results of user satisfaction
 - Update the style guide to reflect test results
 - Application code specs can be developed at this time, but user interface code specs should not begin until after testing is complete

Phase 4: Development

1. Training and Documentation
 - Complete this after the design phase is complete.
2. User Interface Test Plan
 - Compare the performance of users according to ease of use vs. ease of learning
 - Follow the prototype UI test plan when developing this UI test plan
 - Development can be in parallel to any development stage task
 - Update the project plan to reflect test results

Phase 5: Testing and Implementation

1. Use the benchmark testing techniques outlined earlier
2. Give minimal instruction to the users
3. Videotape the test to collect data
4. Compare the data against goals
5. Testing iteration process:
 - Test the users and collect data
 - Analyze the data
 - Redesign the UI design based on the results
 - Start testing again as necessary
6. Update the style guide after testing, and reformulate the project plan as necessary
7. Testing and implementation can be done in parallel with hardware installation testing, but after general software product testing
8. Extended evaluation – Wait a period of time after the software has been deployed, and test or interview the “expert” users that have developed. This will inform the development of new software products or software product updates