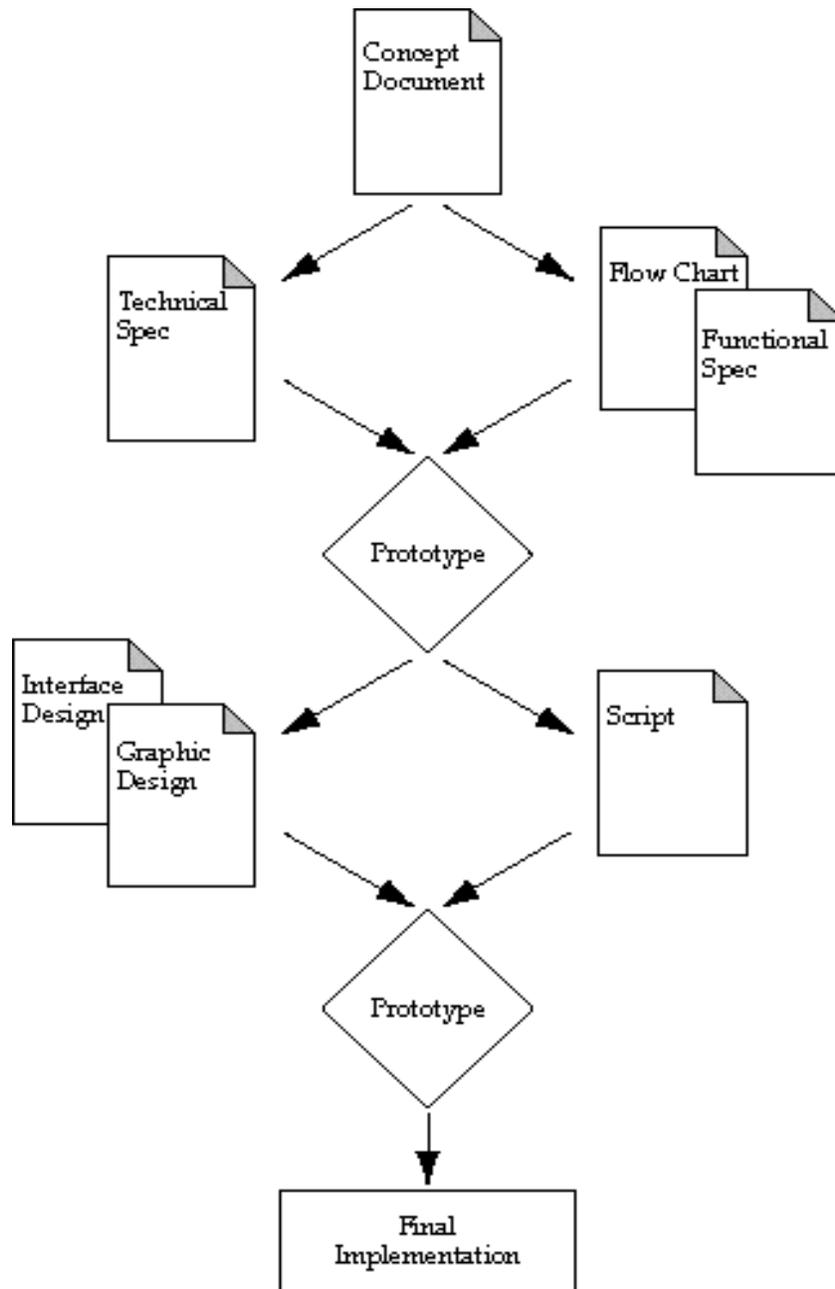


Application Design Process

Below is an outline of the design process for a multimedia project. These stages are completed before final production begins in order to assure consistency and to account for as many issues as possible before production resources are allocated.



CONCEPT DOCUMENT

The concept document provides the overall guidelines for a project. It should answer such questions as: What are the project's goals? Who is the target audience for this project? Why is multimedia the best method to achieve the stated goals? How will this project achieve those goals? What form will the final deliverable take? What material will make up the content of the project, and in what form? What are areas that present potential problems or concerns at the outset of the project?

TECHNICAL SPECIFICATION

Define the technological framework of your project. This includes details such as:

- delivery platform (Mac, PC, or both)
- operating system and version
- delivery method (CD-ROM, Zip, Web)
- projected RAM and storage requirements
- CD-ROM speed (if applicable)
- screen size and color depth
- audio & video requirements
- networking or telecommunication requirements
- intermediate software, such as browsers, plug-ins, etc.
- fonts
- description of palettes to be used
- external code resources
- additional system software
- additional hardware configurations

This component of the application design should be as specific as possible. The technical specs define the limitations of the project and provide the production team with a technological framework from which to develop the application. Any subsequent design features must adhere to the technical specs. It is important to realize who the final users for the project will be and create the technical specs appropriately (remember, your target user may not have a 500 MHz CPU with 256 MB of RAM, a 16 Gigabyte hard drive, and an 8x DVD-ROM drive - or they may!).

FLOW CHART

A flow chart or story board is a visual map of the application. It should detail each screen in the application, including a list of major components, such as text, audio and video assets, interface controls, and input and output methods (text entry and data fields, etc.). Much of the detail listed in the flow chart is related to the information in functional specification and may be integrated with the functional spec.

The flow chart is key to determining the navigational flow of the application. Be sure to include all navigational controls at this stage, including things such as help and/or quit buttons. Use this map to evaluate the path taken by the user to get to the information you are presenting.

FUNCTIONAL SPECIFICATION

The functional specification outlines what the application actually does. In particular, the functional spec should detail the function of each control on every screen, as well as any “behind-the-scenes” functionality that is required, such as a database structure, or the ability to read or write external text files, or dynamic updating of information from a Website. Many functions are common from screen to screen, such as calling a help screen or quitting the application, and should be incorporated into custom handlers which can be written once but called many times.

SCRIPT

Depending upon the nature of an application, a script might be necessary. Much like a traditional movie or television script, the script for an interactive title is a description of the user experience, both in terms of content and environment. In a project such as an interactive game or animated presentation, the script would describe characters, their personalities and appearance, and the dialog of their interaction with users and each other. For a project such as a tourist information kiosk, the script is used to describe the environment in which the user experience takes place, what the mood of the experience should be, what effect it should have on the user.

GRAPHIC DESIGN

The graphic design of an application defines the “look” of an application. It is the most immediate way of conveying visual messages to the user. Treatment of the following items should be considered in the creation of a graphic design:

- backgrounds & control panels
- colors & textures
- type – titles & text, bitmap & ASCII
- containers for type, video, and controls
- “white” space
- alignment & positioning

INTERFACE DESIGN

The interface is the main means of interaction between the user and the application. Most interface functions can be broken down into one of three categories: getting information from the user, returning information to the user, and initiating action.

For user input, a selection of multiple choice buttons or a text field where the user can enter information are most frequently used, but variable interface elements such as sliders are also useful for modifying settings like sound volume level. Returning information to the user occurs in a variety of forms, text output fields displaying messages, digital video windows for appropriate content, audio feedback to confirm user actions, or altering graphics to indicate changing states in the application. Any number of interface components can be used to initiate actions, but the most common form is the button. In this case, a button is anything that is clickable, regardless of its graphical shape.

The most important aspect of an interface design is the ability of the user to get the application to do what he or she wants. Interface designers must be careful of the assumptions they make regarding the user’s level of understanding. Users have a certain amount of information in their heads that interface designers can rely on, such as the fact that buttons are pushed, knobs are turned, switches are switched, and sliders are slid. Any additional information the user needs should be supplied by the interface. If the system is too complex, a help system must be integrated into the design, but, a good interface should be able operable without additional instruction, even if a help system is needed to explain the functionality of an application.

PROTOTYPE

Prototypes are preliminary versions of an application that are created to test various aspects of a design before final production. Prototypes are found in one of two categories, interface prototypes and functional prototypes.

An interface prototype allows designers to test the way users respond to the interface in terms of clarity, consistency, navigability, and ease of use. Interface prototypes also enable designers to test various methods of performing the same activity to determine which method best suits the intended users.

A functional prototype, on the other hand, is used to build and test a certain feature without the overhead of the complete program. These prototypes simplify the process of creating and debugging new features or controls. The scripts from the prototype containing event handlers can then be integrated into the complete application with minor modifications.

Test as much of any design as possible, use the results of the tests to revise the design, then test it again.

FINAL IMPLEMENTATION

Use feedback from prototype testing, make any necessary modifications to the application design, and build it. Once built, the program should be rigorously tested for functionality and content correctness, as well as performance. Any errors or problems found will then be addressed, a new version built and tested again. This cycle is repeated until an acceptable level of quality is achieved and the final version delivered or deployed.