Chapter 10: Delivering the system

1. If the prototype and the final working system differ in ways that affect how the user performs tasks, then the user may have trouble blocking out the features of the prototype, which they might be used to or they might prefer to the features in the final system. This type of task interference would apply in addition to the task interference from the user's previous procedures for performing tasks.

2. Many applications that are designed for home PC use are not meant to have separate users and operators, as the user will also be responsible for installing the software, configuring its settings and parameters, and backing up and restoring files. Therefore both user and operator functions must be included in the training.

3. In general, in any software designed for the general public (e.g. word processors, spreadsheet packages, home financial applications), the underlying system should be transparent to the user. This is generally considered good design because the visibility of system details, which are not directly related to user functions, makes the interface less user-friendly. For example, a textual document might be stored on a computer in pages where the page size is determined by the operating system, not by any attributes of the document itself. But when the user processes that document using a word processor, it should be broken up into pages that correspond to the printed appearance of the document, and the user should have no sense of where the boundaries are between memory pages.

5. Some questions to consider:

   - What is the background of the system's users? Are they likely to understand failure reports?
   - Does not reporting failures to the user make the system appear more reliable than it really is?
   - Are there cases in which the user or operator might want to take additional action in response to a failure, besides what the system automatically does?
   - Could there be patterns of failures over time in which each individual failure is not particularly alarming, but the trend may indicate a more serious problem?
   - Would it be appropriate to record failures in a location that a user or operator could check periodically, rather than intrusively notify the user each time a failure occurs?

6. Most of the failure messages are clear and straightforward and would be understandable to most users. In some cases, friendlier language could be used (e.g. "printer" or "disk," etc., instead of "device"). Also, additional information would be helpful in several cases (e.g. line numbers for program errors).