In many engineering disciplines, it is known that a small-scale experiment needs to be tested in a medium scale first before implementing it in a large-scale system. The same is applicable for programming projects as well, and software development management must plan for piloting and redeveloping.

This is mandatory as change of user need is inevitable for software products or services. There needs to be a defined balance of the extent to which the developer needs to meet changing customer needs. The preparedness to meet this change needs to be incorporated in planning.

Practices like modularization, subroutine, defining intermodular interfaces, documentation, standard calling sequences and table-driven techniques, high-level language and self-documenting techniques, compile-time operations to incorporate standard declarations, quantization of change, numbering product versions with schedule and freeze date should be practiced.

The planning process of the entire organization should facilitate change. Designers need to be responsible for the changes they make and document the design decisions. Sociological attention should be applied to make workforce interchangeable between development and management roles.

Updating a software system often leads to new issues, which makes existing software system fragile with increasing updates. Thus, every software system must be developed from ground up after a threshold.