Rapid Application Development Testing

Rapid application development (RAD) is an effective software development paradigm that provides a systematic and automatable means of developing a software system under circumstances where initial requirements are not well known or where requirements change frequently during development.

This testing strategy assumes the RAD system

- is iterative.
- is evolutionary.
- contains RAD language with a defined grammar.
- provides reusable components capability (library and retrieval).
- uses implementation code from reusable components written in a high-level language.
- contains a sophisticated support environment.

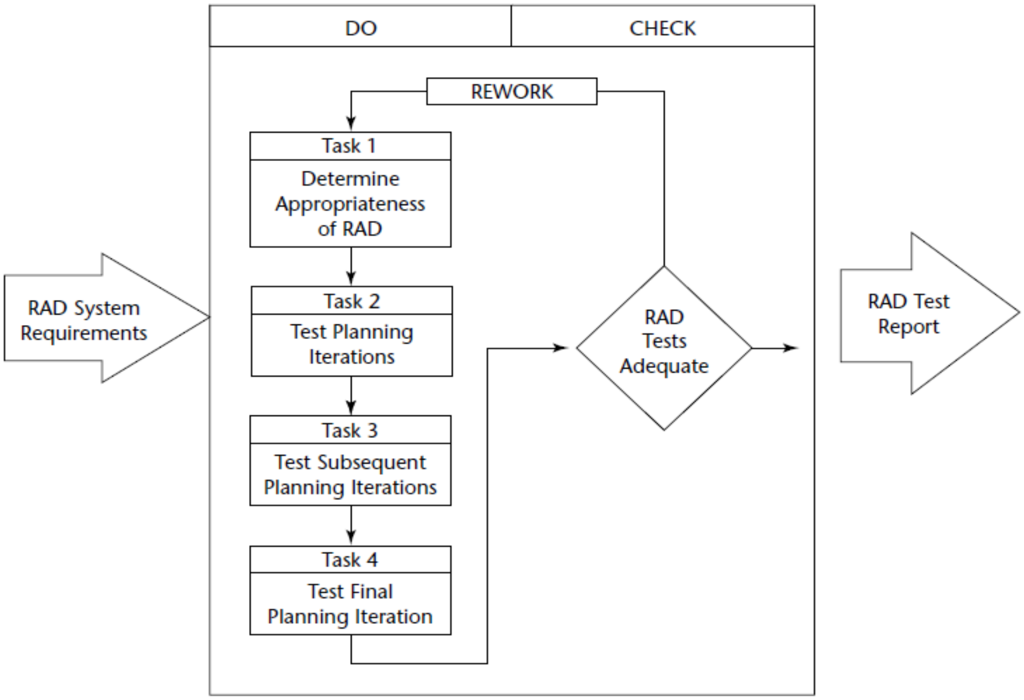

Figure 16-1 Workbench for testing RAD systems.

Spiral Testing

The proposed RAD testing strategy, termed *spiral testing*, remains iterative and parallels the RAD process. Spiral testing characterizes the varying types of RAD iterations by tailoring the testing process to account for these differences. Spiral testing distinguishes between the initial few RAD testing iterations, subsequent iterations, and the final few iterations.

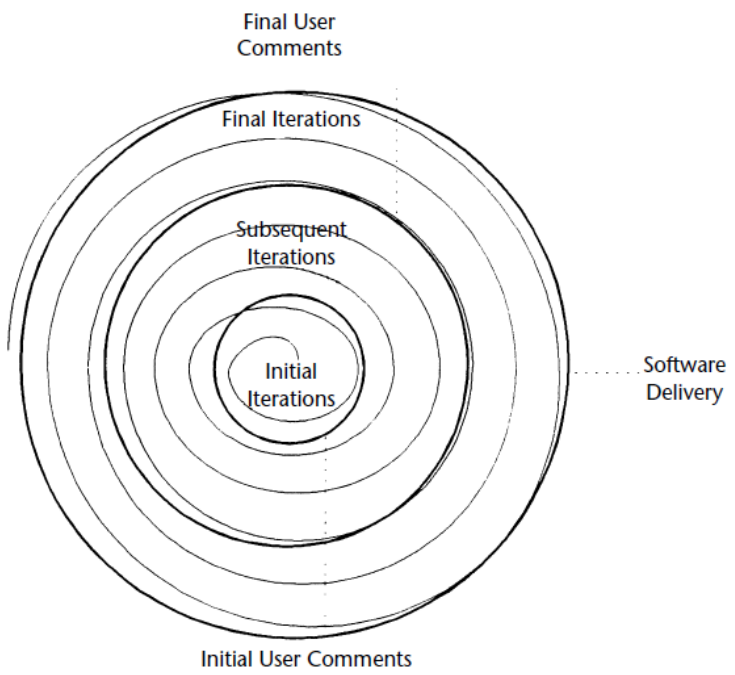

Figure 16-3 A "targeted" spiral.

Task 1: Determine Appropriateness of RAD

There are strengths and weaknesses to using the RAD concept to build software. If the advantages outweigh the disadvantages, RAD should be used.

RAD development offers the following strengths:

- System users get a quick view of the system's deliverables.
- The cost of risk associated with development is reduced because decisions can be made early regarding the usability of the first RAD prototype.
- Customer satisfaction can be improved through a series of developmental iterations rather than a focus on a single deliverable.
- If the project is developed by a project team familiar with the user's business, fewer developers are required because of their knowledge of the user business.
- Using powerful development tools can reduce the cycle time for developing the final system.

The problems associated with using the RAD model for development, on the other hand, include the following:

- Users and developers must be committed to rapid-fire activities in an abbreviated time frame; thus any lack of commitment negates most of the advantages of the RAD model.
- Diffusers are not continuously involved throughout the RAD cycles. Obtaining the necessary feedback at the appropriate times will facilitate development.
- Unless the system can be appropriately modularized and has access to reusable components, the reduced cost and schedule may not be achieved.

- Because the RAD concept does not require a fixed completion date, the risk is that the development team will continue through the RAD cycles past the point of economic return.

Task 2: Test Planning Iterations

- The initial iterations are where the test team will forecast the most important portions of the system to test.

- As the implementation hierarchy of the system takes shape, the testers establish test sections for path and integration testing.

- The long-term testing purpose is to build the framework for constructing the final acceptance-test oracle and to fit the intermediate testing activities into the overall development plan.

- The process will be manual for the most part, and this would be where initial testing tools and their databases/instrumentation would be initialized.

- The initial iteration phase would end at the RAD iteration in which the top-level requirements specification is established.

Task 3: Test Subsequent Planning Iterations

- Once the basic RAD framework is established, subsequent iterations commence in which developers enhance the RAD's functionality and demonstrate it for user/designer review.

- In the typical case, additional requirements are identified and the design matures in parallel over multiple iterations. Both are validated in the review process.

- At some point, sufficient requirements are identified to establish the overall system design.

Task 4: Test the Final Planning Iteration

- Once developers establish all requirements, the final few iterations of development are devoted to implementing the remaining functionality, followed by error correction.

- Therefore, the testers can devote their work to completing the test for acceptance testing, and to remaining unit testing and subsection integration testing.