

Discipline by Discipline: Testing (article)

Discipline by Discipline: Testing

Over the last few weeks I've been posting ideas on how to effectively manage each discipline within a project using a combination of Agile and Unified Process techniques in order to optimize the 'throughput' of business ideas into high quality software. Today I'll spend some time on the Testing discipline which ensures that a high quality solution is being provisioned.

Iterative Testing: A Paradigm Shift

One of the first steps in developing an effective iterative testing approach is recognizing some of the areas that it differs substantially from more traditional testing approaches. A paper I wrote several years ago called *Managing Iterative Testing in an Agile Development Project* provides some practical guidance in considerably more detail to what is below.

- **Testing at the Source:** One of the most important distinctions is the emphasis on Agile and Lean techniques of '*quality at the source*'. The intent is to minimize the hand-offs that occur between different groups managing testing and development. Each team member is responsible for ensuring tests or acceptance criteria exist at each level of the development lifecycle. **Success criteria** at the Business Modeling level identify the underlying business model of the system being developed, Requirements written using an **Executable Requirements** approach have clear pass/fail conditions and can be automated, **Unit Tests** and a **Test Driven Development** approach ensure high quality software and the opportunity to refactor the architecture of the system with knowledge of where changes are impacting application functionality.
- **Test Coverage Emphasis:** With the techniques described above the emphasis of individuals in a test capacity becomes focused on testing across user

stories or use cases and ensuring business rules and non-functional requirements are adequately tested. The Test Plan is an important tool in identifying the test strategies that will support the more granular Executable Requirements and Unit Tests mentioned above.

- **Incremental and Adaptive Testing:** Iterative development testers must be accustomed to testing only what has been delivered within each iteration, and therefore to planning and executing incremental tests of the same – but evolving – functions over time. Testers should have visibility and engagement in capturing requirements early in the lifecycle. They must understand that the absence of a feature in an early iteration not contracted to support that feature is not a defect. They should be able to distinguish clearly between regression testing and planned incremental testing.
- Some more guidance can be found at a short article I wrote in 2002 called [Blending A Test Strategy Into an Iterative Development Project](#)

If you're trying to assess how effective your project's test plan is you might be interested in an article I wrote called [Ten Tough Questions to Ask When Developing a Project Test Plan](#).