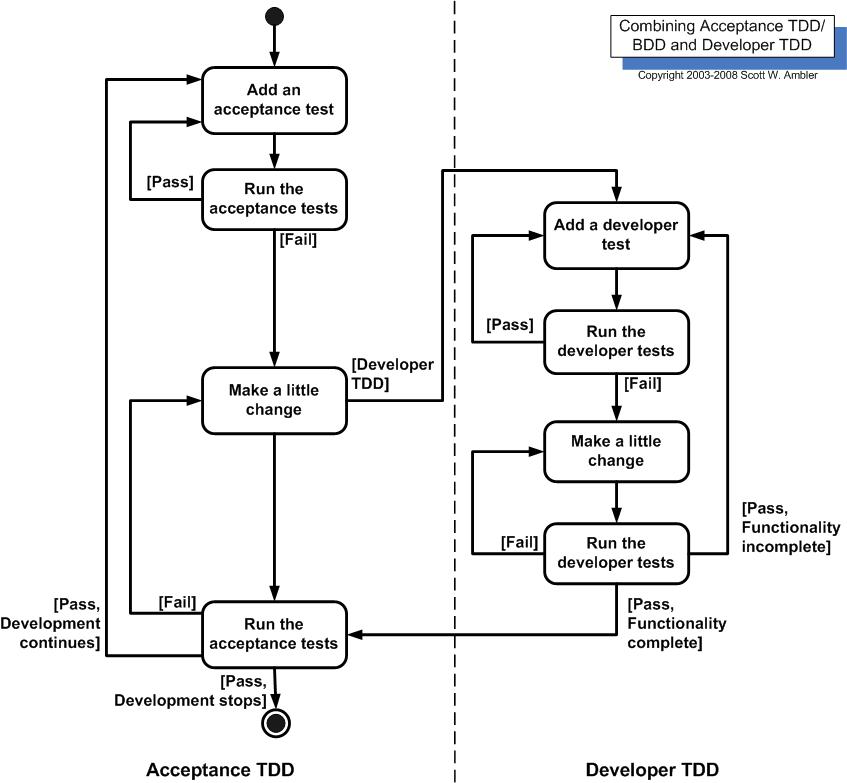
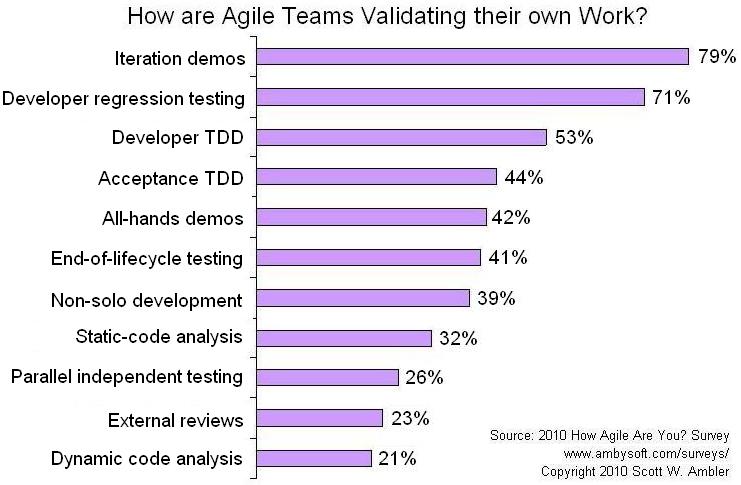
bb

There are two levels of TDD:

1. **Acceptance TDD (ATDD)**.  With ATDD you write a single [acceptance test](http://www.agilemodeling.com/artifacts/acceptanceTests.htm), or behavioral specification depending on your preferred terminology, and then just enough production functionality/code to fulfill that test. The goal of ATDD is to specify detailed, executable requirements for your solution on a just in time (JIT) basis. ATDD is also called Behavior Driven Development (BDD).
2. **Developer TDD**. With developer TDD you write a single developer test, sometimes inaccurately referred to as a unit test, and then just enough production code to fulfill that test. The goal of developer TDD is to specify a detailed, executable design for your solution on a JIT basis. Developer TDD is often simply called TDD.

[Figure 2](http://agiledata.org/essays/tdd.html#Figure2) depicts a UML activity diagram showing how ATDD and developer TDD fit together.  Ideally, you'll write a single acceptance test, then to implement the production code required to fulfill that test you'll take a developer TDD approach. This in turn requires you to iterate several times through the write a test, write production code, get it working cycle at the developer TDD level



**Figure 3. Testing via the xUnit Framework.**

