Beta Testing Checklist

- ☐ **1.** **Design test scenarios.** What's a "test scenario"? Each **test scenario** should be mirror image of a **"use scenario**" that's been guiding a team to design and develop the system. A *use scenario* describes one typical interaction a customer has with the system, such as withdrawing or depositing cash in an ATM machine.
- ☐ **2.** **Write a test procedure**. A test procedure specifies **how** testers will exercise test scenarios, including the **order** to follow. Specify tests in forward, reverse, and random sequences, and indicate the **results** to expect in each case.
- ☐ **3.** **Determine what data you need.** If your system stores values in a database, you'll need to load some typical data to test the scenarios. In the ATM example, values would include account balances — for testing withdrawal limits and giving balance information. Create the sample data sets and pre-load the systems to be tested. Don't forget to include extremely high and low values!
- ☐ **4.** **Plan specific roles for testers.** Schedule each tester to focus on specific test scenarios and related data sets. If there are enough testers, assign more than one to cover each test scenario. Each person will approach it differently.
- ☐ **5.** **Create a bug reporting system.** It could be designed as a form, a database, an e-mail message, or a combination. Have testers submit bug reports as they find errors in each round of testing.
- ☐ **6.** **Establish a test schedule.** The schedule should allow for several iterations of beta testing. Be sure to clear the schedules of testers for each round in which they will be participating.
- ☐ **7.** **Get all materials ready for testing.** The following items should be ready for the kickoff: A new or updated system, lists or descriptions of any bugs fixed, new or updated documentation, test scenarios and procedures, and so on.
- ☐ **8.** **Set a start date.** Then hold a kickoff meeting! Also schedule progress checks. If testers find numerous bugs — or especially critical ones — before reaching a given checkpoint, stop testing, fix the bugs and/or documentation, and return to **Step 1**. Ask before restarting: Are new test scenarios or data sets needed?
- ☐ **9.** **Perform a new round of testing for each new test baseline.** This means starting the complete test ***from scratch*** after each round of fixes. You can't sidestep this requirement, because each time something is fixed, it can "break" something else. Stop the cycles of testing only when no new bugs are evident.
- ☐ **10.** **Plan a reward for a job well done.** Testing is *very* tedious — so testers need a special incentive, like a post-testing party, to keep them focused on the goal.