# Functional Testing & Evaluation Checklist

- [ ] **1. Evaluate design alternatives.** Is your first design concept ideal? You will never know without creating several prototypes, comparing their properties, and then selecting the winning traits from the best performer(s).

- [ ] **2. Assess interface usability.** Is your first interface design concept certain to be the easiest one to use? To find out, invite representative users to try out the proposed interface in the form of paper and/or electronic mockups. Use the findings to make design improvements, and keep repeating the process.

- [ ] **3. Perform a difficulty analysis.** Ask these types of questions about each new version of your offering: Have you prevented all unnecessary bells and whistles from creeping into your system? Have you automated or kept to a bare minimum all of the tedious busywork, like installation and setup? Have you used a "hassle hunt" to remove any other customer annoyances?

- [ ] **4. Conduct regular alpha tests.** *Alpha tests* are the checks run on system components that are still under development. Effective test practices show that daily testing, using automated routines, is the surest way to discover and correct defects — *before* they become buried and hard to find. As you finish components, you can broaden the testing to higher levels.

- [ ] **5. Conduct beta testing.** Later in development, beta testers with varying degrees of system familiarity should validate it repeatedly. **Does it:**

    - [ ] **Operate in an error-free manner?** You will need to test all of the features against their requirements, with both normal and abnormal data, alone and in combination with other features, and in both typical and atypical sequences. Why? *Customers will do many "creative" things with your system!* You can't stop users from entering strange values or trying funny keystroke combinations. Therefore, a system should either prevent these actions, or respond to them elegantly.

    - [ ] **Support real-world goals and tasks?** For example, does it guide the users in completing each primary objective? Test the system using both onscreen and written instructions to verify accuracy and clarity.

    - [ ] **Function flawlessly under sub-optimal conditions?** Will the product or service operate in a "bullet-proof" mode during off-hours, in isolated conditions, bad weather, or remote locations? If not, it should gracefully stop the action without doing harm and let users know what to do next.