

- Level 0** There is no difference between testing and debugging.
- Level 1** The purpose of testing is to show correctness.
- Level 2** The purpose of testing is to show that the software does not work.
- Level 3** The purpose of testing is not to prove anything specific, but to reduce the risk of using the software.
- Level 4** Testing is a mental discipline that helps all IT professionals develop higher- quality software.

Level 0 is the view that testing is the same as debugging. This is the view that is naturally adopted by many undergraduate Computer Science majors. In most CS programming classes, the students get their programs to compile, then debug the programs with a few inputs chosen either arbitrarily or provided by the professor. This model does not distinguish between a program's incorrect behavior and a mistake within the program, and does very little to help develop software that is reliable or safe.

In **Level 1** testing, the purpose is to show correctness. While a significant step up from the naive level 0, this has the unfortunate problem that in any but the most trivial of programs, correctness is virtually impossible to either achieve or demonstrate. Suppose we run a collection of tests and find no failures. What do we know? Should we assume that we have good software or just bad tests? Since the goal of correctness is impossible, test engineers usually have no strict goal, real stopping rule, or formal test technique. If a development manager asks how much testing remains to be done, the test manager has no way to answer the question. In fact, test managers are in a weak position because they have no way to quantitatively express or evaluate their work.

In **Level 2** testing, the purpose is to show failures. Although looking for failures is certainly a valid goal, it is also inherently negative. Testers may enjoy finding the problem, but the developers never want to find

problems—they want the software to work (yes, level 1 thinking can be natural for the developers). Thus, level 2 testing puts testers and developers into an adversarial relationship, which can be bad for team morale. Beyond that, when our primary goal is to look for failures, we are still left wondering what to do if no failures are found. Is our work done? Is our software very good, or is the testing weak? Having confidence in when testing is complete is an important goal for all testers. It is our view that this level currently dominates the software industry.

The thinking that leads to **Level 3** testing starts with the realization that testing can show the presence, but not the absence, of failures. This lets us accept the fact that whenever we use software, we incur some risk. The risk may be small and the consequences unimportant, or the risk may be great and the consequences catastrophic, but risk is always there. This allows us to realize that the entire development team wants the same thing—to reduce the risk of using the software. In level 3 testing, both testers and developers work together to reduce risk. We see more and more companies move to this testing maturity level every year.

Once the testers and developers are on the same “team,” an organization can progress to real **Level 4** testing. Level 4 thinking defines testing as *a mental discipline that increases quality*. Various ways exist to increase quality, of which creating tests that cause the software to fail is only one. Adopting this mindset, test engineers can become the technical leaders of the project (as is common in many other engineering disciplines). They have the primary responsibility of measuring and improving software quality, and their expertise should help the developers. Beizer used the analogy of a spell checker. We often think that the purpose of a spell checker is to find misspelled words, but in fact, the best purpose of a spell checker is to improve our ability to spell. Every time the spell checker finds an incorrectly spelled word, we have the opportunity to learn how to spell the word correctly. The spell checker is the “expert” on spelling quality. In the same way, level 4 testing means that the purpose of testing is to improve the ability of the developers to produce high-quality software. The testers should be the experts who train your developers!

As a reader of this book, you probably start at level 0, 1, or 2. Most software developers go through these levels at some stage in their careers. If you work in software development, you might pause to reflect on which testing level describes your company or team. The remaining chapters in Part I should help you move to level 2 thinking, and to understand the