Brooke Slampak

CS-320

**Module One Journal**

During the testing stage, the software gets checked to make sure it works like it’s supposed to. Testers run different tests to find bugs, glitches, or anything that doesn’t match the client’s requirements. This can include unit tests that check small pieces of code, integration tests that make sure parts of code work together, system tests which check the code as a whole, and user acceptance tests that see if it works for the people who’ll actually use it. If they find any problems, they report them, the developers fix them, and then it gets tested again until everything works as intended and required.  
 Testing is important because it helps catch issues with software before it is officially released. If bugs slip through, they can cause crashes, security issues, or other problems that are more expensive and time-consuming to fix post-launch. Good testing habits make sure the software is reliable, safe, and does what the client and users need it to do. It also helps keep the software’s userbase happy and more likely to keep using the program.  
 In some development approaches, testing happens earlier and more often. For example, in Agile or Extreme Programming, developers do something called test-driven development, where they write tests before they even write the code. This helps them build better code from the start and catch issues sooner. On the flip side, sometimes testing might get pushed later if there are last-minute changes or integrations with other systems that can’t be tested until the end. But putting off testing too long usually causes more problems than it solves, so most teams try to test early and often whenever it’s possible.