# Slides 09\_25 re testing

* **System Tests** (aka integration tests, functional tests, or “black box” tests) test functionality of the system from the user’s perspective: does it do what it’s supposed to do?
  + Are the “user stories” implemented?
  + Do the inputs result in the expected outputs?
  + Browser automation and other “macro” tools can be used to automate.
* Testing needs a mix of automation and human testing.
* Automation serves to reduce manual labor. If a tester has to repeat simple steps many times, he either won’t do it, or he won’t do it well.
  + Imagine testing a 10-page online questionnaire, where you have to fill out all 10 pages just to test a change to the 10th page.
  + Be wary of metrics like “coverage” that give you false confidence in the results.
  + Human testing (also called exploratory testing) should be used where repetition is not needed, or automation is difficult. Humans can test systems in a lot more ways, in a lot less time.
  + You can still write “scripts” to document your tests and make them repeatable.
  + Focus on areas where there are real questions to answer – if you know the test will always pass or always fail, there’s no need to do it.
* **BDD = Behavior Driven Development**
  + Like TDD, but tests are written in plain language by the client or product owner, rather than by developers.
  + This is a great way to train your product owners to write better specifications! They will quickly learn if they’re asking for things they don’t really need, or failing to ask for things they ought to have asked for.
  + Cucumber ([http://cukes.info](http://cukes.info/)) is a leading product in this area. Anyone want to try it on their project?
* **Beginning with release v0.4 and continuing to the end of the class, your project will include a testing component.**
* From this point onward, the project repo should include a folder called `tests` and there should be some documentation in the README file about 'how to test' or **'how to run the tests'**.
  + Tests may be automated scripts or may be written documentation of manual test procedures.
  + These may be "validation" tests, i.e. user tests or A/B tests that test whether the product serves the customer's needs, or "verification" tests that make sure the software is of good quality, doesn't have bugs, etc.
  + Whatever type of testing you do, you must document how to carry out the tests.