1. GCM is composed of
   * Nucleus – the core engine of GCM
   * Util – the utility library
   * Plugins – 10 core plugins
2. Every concrete public method of GCM has at least one JUnit test in the test suite
3. The test suite excludes
   * Abstract methods
   * Interfaces – there are some exceptions where the concrete implementation of an interface is private
   * Enum.valueOf()
   * Enum.vlaues()
4. **Test only direct post condition contract stipulations**. A well-formed unit test should generally only test the specific capability. For example, suppose that there is a data manager that supports family membership for individuals. In testing a method for adding a person to a family, the test should show that 1) the person was not initially a member of the family, 2) the person’s family id is correct after the addition. It is not necessary, and often misleading, to further test for example that the person is included in the list of people in the family. The other tests for the data manager should cover such considerations as the focus of each test.
5. **Do not combine tests.** There is a false economy to combining tests even though it may seem that several related tests could be combined. For example, combining both tests for both the getter and the setter methods may seem obvious since the two tests will likely have nearly identical implementations. However, the two methods are also likely to have somewhat different precondition tests and the mixed focus of the test can be difficult to maintain over time.
6. **Test all preconditions.**
7. **Do not reuse objects that have thrown exceptions.** When violation of a precondition throws an exception (in GCM this nearly always the case), the object that was the source of the exception should be considered non-trusted. Although one can reason that it may remain in a valid state, in complex environments this can be difficult to ascertain. Thus reusing the object that threw the exception to test another precondition check may result in the test passing because the state of the throwing object is invalid. For example, consider the addition of person when the regions plugin is being used. The auxiliary data should include a region assignment. If it is omitted, the person will be added from the perspective of the person data manager, but the region data manager will throw an exception. The simulation is now in an unrecoverable state where at least one person does not have an associated region id.
8. **Prefer the ContractException class to other exception for precondition tests**. Any error that GCM explicitly detects is a violation of some contract. Doing this allows for an early diagnosis of the cause of an error since non-ContractExceptions can be reasoned to be a bug in either GCM or a plugin rather that with a client’s use of GCM.
9. **Do not test the classes used in the test suite**. Any classes that support testing and are important enough to warrant be tested should be located in a test support package included in the production code base.
10. **Prefer testing behaviors of actors and data managers via a full simulation run**.