**1. Test strategy document:**  
This outlines the over all testing objectives, what testing goals exists and how is the over all testing will be performed linking all levels from unit test, component test, system test and integration test. This is not a standard or something - but it can be something of [this sort](http://www.luc.edu/its/pdfs/Test_Strategy_Template_.pdf).

**2. Test suit:**  
This is the collection of test cases and conditions on when and how each cases needs to be performed. Each set of inputs, procedures and expected output behavior against each elements. There are times when more than just success and failures are noted so that further analysis is done on this.

**3. Test environment/setup and procedures**  
If you are automating the testing process fully or partly, it is worthwhile to document how exactly the (various elements of) testing is going to get executed. It should be debated and validated if the testing performed here is correct method or not. The developer and QA associated should know how to operate the set of tools and what procedures to follow.

**4. Traceability Matrix:**  
This is a well defined matrix which identifies which set of test cases are relevant to promise each functionality point is assured to be steady. [Read more here](http://www.softwaretestingtimes.com/2010/04/traceability-matrix-from-software.html) or this [wiki](http://en.wikipedia.org/wiki/Traceability_matrix) link. Whenever a new bug is discovered or a new feature is requested, traceability matrix should be updated to capture these changes.

**5. Test results**  
Whether generated automated or performed manually, the results (detailed and summerised) should be captured in a test execution sheet. Most important thing to note down is that a. original observation (such as logs, actual output of the application) should be capture as relevant so that conclusions can be validate. b. the document needs to capture the build against which these test were carried out; different build may not produce the same behavior against the same test.

Procedure and formats can be developed as needed. Most important thing, from my personal experience is that instead of making water tight compliance to some format allow people do document this like a running diary making only few things mandatory and let people pour in more information freely. Testing is never static (at least for any reasonably complex project) so over time all these template must evolve continuously - quite often every next step could be a major departure from the last one. If either the templates are stale, or people do not follow that because it is too rigid, eventually much of the relevant knowledge through testing procedure won't be reflected right.