**How to Review SRS Document and Create Test Scenarios – Software Testing Training on a Live Project – Day 2**

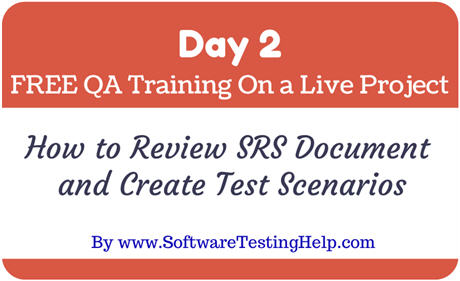
Posted In | [Software Testing Templates](http://www.softwaretestinghelp.com/category/software-testing-templates-test-cases/), [Software Testing Training](http://www.softwaretestinghelp.com/category/software-testing-training/), [Testing Life cycle](http://www.softwaretestinghelp.com/category/testing-life-cycle/)

This is a second tutorial in our **‘free online software testing training on a live project’** series. If you are new here please check the first introduction tutorial: **[End to End Software Testing Training on a Live Project.](http://www.softwaretestinghelp.com/free-online-software-testing-qa-training-course/" \o "Software Testing Training Course on a Live Project)**

*Let us now get into a detailed analysis of how an SRS walkthrough happens, what is it that we need to identify from this step, what pre-steps we need to take before we begin, what are the challenges we could face etc. in a detailed manner.*

**SDLC’s Design Phase:**

The next phase in the SDLC is “Design”- this is where the functional requirements are translated into the technical details. The dev, design, environment and data teams are involved in this step. The outcome of this step is typically a Technical Design Document- TDD. The input is the SRS document both for the creation of the TDD and for the QA team to start working on the QA aspect of the project- which is to review the SRS and identify the test objective.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/qa-training-day-2.jpg)

**What is an SRS review?**

SRS is a document that is created by the development team in collaboration with business analysts and environment/data teams. Typically, this document once finalized will be shared with the QA team via a meeting where a detailed walkthrough is arranged. Sometimes, for an already existing application, we might not need a formal meeting and someone guiding us through this document. We might have the necessary information to do this by ourselves.

***SRS review is nothing but going through the functional requirements specification document and trying to understand what the target application is going to be like.***

The formal format and a sample were shared with you all in the previous article. It does not necessarily mean that all SRSs are going to be documented that way exactly. Always, **form is secondary to the content**. Some teams will just choose to write a bulleted list, some teams will include use cases, some teams will include sample screenshots (like the document we had) and some just describe the details in paragraphs.

**Pre-steps to software requirements specification review:**

**Step #1:** Documents go through multiple revisions, so make sure we have the right version of the reference document, the SRS.

**Step #2:** Establish guidelines on what is expected at the end of the review from each team member. In other words, decide on what deliverables are expected from this step – typically, the output of this step is to identify the test scenarios. Test scenarios are nothing but one line pointers of ‘what to test’ for a certain functionality.

**Step #3:** Also establish guidelines on how this deliverable is to be presented- I mean, the template.

**Step #4:** Decide on whether each member of the team is going to work on the entire document or divide it among themselves. It is highly recommended that everyone reads everything because that will prevent knowledge concentration with certain team members. But in case of a huge project, with the SRS documents running close to 1000 pages, the approach of breaking up the document module wise and assigning to individual team members is most practical.

**Step #5:** SRS review also helps in better understanding if there are any specific prerequisites required for the testing of the software.

**Step #6:** As a byproduct, a list of queries where some functionality is difficult to understand or if more information needs to be incorporated into functional requirements or if mistakes are made in SRS they are identified.

**What do we need to get started?**

* The correct version of the SRS document
* Clear instructions on who is going to work on what and how much time have they got.
* A template to create test scenarios
* Other information on- who to contact in case of a question or who to report in case of a documentation inconsistency

**Who would provide this information?**

Team leads are generally responsible for providing all the items listed in the section above. However, team members’ inputs are always important for the success of this entire endeavor.

Team leads often ask- What kind of inputs? Wouldn’t it be better to assign a certain module to someone interested in it than to a team member who is not? Wouldn’t it be nice to decide on the target date based on the team’s opinion than thrust a decision on them? Also, for the success of a project, templates are important. As a general rule, templates have a higher rate of efficiency when they are tailored to the specific team’s convenience and comfort.

It should therefore be noted that, team leads more than anything are team members. Getting your team onboard on the day-to-day decisions is crucial for the smooth running of the project.

**Why a template for test scenarios – isn’t it enough if we just make a list?**

It sure is. However, software projects are not ‘one-man’ shows. They involve [team work](http://www.softwaretestinghelp.com/team-building-in-software-testing/" \o "Software testing team building). Imagine in a team of 4- if each one of them decide to review one module of the software requirements specification each. Team member A has made a list on a sheet of paper. Team member 2 used an excel sheet. Team member 3 used a notepad. Team member 4 used a word doc. How do we consolidate all the work done for the project at the end of the day?

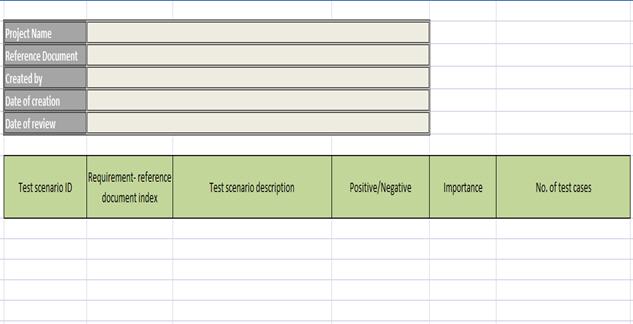
Also, how can we decide which one is the standard and how can we say what is right and what’s not if we did not create the rules to begin with?

That is what template is- A set of guidelines and an agreed format for uniformity and concurrence for the entire team.

**How to create a template for QA Test scenarios?**

[Templates](http://www.softwaretestinghelp.com/category/software-testing-templates-test-cases/" \o "Software testing templates) don’t have to be complicated or inflexible.

All they need to be are an efficient mechanism to create a useful testing artifact. Something simple like the one we see below:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/test-scenarios-template-2.jpg)

The header of this template contains the space required to capture basic information about the project, the current document and the reference document.

**The table below will let us create the test scenarios. The columns included are:**

**Column #1: Test scenario ID**  
Every entity in our testing process has to be uniquely identifiable. So, every test scenario has to be assigned an ID. The rules to follow while assigning this ID have to be defined. For the sake of this article we are going to follow the naming convention as: TS(prefix that stands for Test Scenario) followed by ‘\_’ , module name MI(my Info module of the Orange HRM project) followed by ‘\_’ and then the sub section (eg: MIM for My info module, P for photograph and so on)followed by a serial number. An example would be: “TS\_MI\_MIM\_01”.

**Column #2: Requirement**  
It helps that when we create a test scenario we should be able to map it back to the section of the SRS document where we picked it from. If the requirements have IDs we could use that. If not section numbers or even page numbers of the SRS document from where we identified a testable requirement will do.

**Column #3: Test scenario description**  
A one liner specifying ‘what to test’. We would also refer to it as test objective.

**Column #4: Importance**  
This is to give an idea about how important certain functionality is for the AUT. Values like high, medium and low can be assigned to this field. You could also choose a point system, like 1-5, 5 being most important, 1 being less important. Whatever the value this field can take, it has to be pre-decided.

------------

**Column #5: No. of Test cases**  
A rough estimate on how many individual test cases we might end up writing that one test scenario. **For example**: To test the login- we include these situations: Correct username and password. Correct username and wrong password. Correct password and wrong username. So, validating the login functionality will result in 3 test cases.

**Note:** You can expand this template or remove the fields as you see fit.

**For example**, you can add “Reviewed by” in the header or remove the date of creation etc. Also in the table, you can include a field “Created by” to designate the tester responsible for a certain test scenarios or remove the “No. of Test cases” column. The choice is yours. Go with what works best for the entire team.

**Let us now review our Orange HRM SRS Document and create the test scenarios**

**Tip:** check out the table of contents in the SRS sample we provided in 1st tutorials to get a good idea on how any document is going to flow and how much of work it might involve.

**Section 1** is the purpose of the document. No testable requirements there.

**Section 2.1** – Project Overview- Audience- no testable requirements there either.

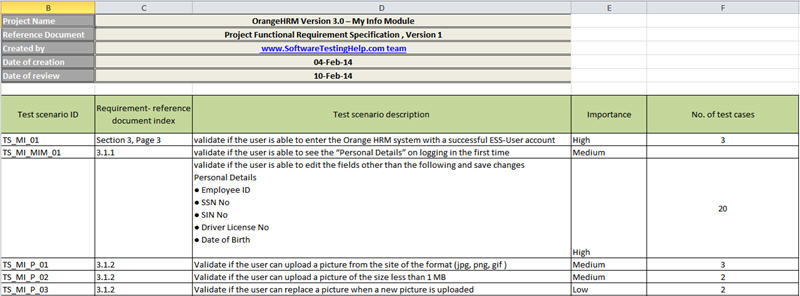
**Section 2.2**- Hardware and hosting- This section is talking about how the Orange HRM site is going to be hosted. Now, is this information important to us testers? The answer is Yes and No. Yes, because when we test we need to have an environment that is similar to the real time environment. This gives us an idea on how it needs to be. No because, it is not a testable requirement- a kind of prerequisite for the testing to happen.

**Section 3:** There is a login screen here and the details of the type of account we need to have to enter the site. This is a testable requirement. So it needs to be a part of our Test scenarios.

Please see the test scenarios document where test scenarios for a few sections of the SRS have been added. For practice, please add the rest of the scenarios in a similar manner. However, I am going right to section 4 of the document.

**Section 4:**Aesthetic/HTML Requirements and Guidelines- This section best explains how some requirements might not make sense to the test team at the time of SRS review, but the team should make a note of them as testable requirements all the same. How to test them and if we need specific set up/any team’s help to validate it are details we might not know at this point of time. But making them a part of our testing scope is the first step to ensure that we do not miss them.

**Sample Test Scenarios for OrangeHRM Application: *(click to enlarge image)***

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/test-scenarios-template.jpg)

**=> *Please refer and [download the test scenarios document](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/Test-scenarios_Orange-HRM-3.0-SoftwareTestingHelp.xlsx" \o "Test scenarios document" \t "_blank) for more information.***

**Some important observations regarding SRS review:**

**#1.** No information is to be left uncovered.  
**#2.** Perform a feasibility analysis on whether a certain requirement is correct or not and also if it can be tested or not.  
**#3.** Unless a separate performance/security or any other form of test teams exists- it is our job to make sure that all nonfunctional requirements have to be taken into consideration.  
**#4.** Not all information is targeted at the testers, so it is important to understand what to note and what not.  
**#5.** The importance and no. of test cases for a test scenario need not be accurate and can be filled in with an approximate value or can be left empty.

**To sum up, SRS review results in:**

* Test scenarios list
* Review results – documentation/requirement errors found by statically going through/verifying the SRS document
* A list of Questions for better understanding- in case of any
* Preliminary idea of how the test environment is supposed to be like
* Test scope identification and a rough idea on how many test cases we might end up having- so how much time we need for documentation and eventually execution.

**Important points to note:**

**#1.** Test scenarios are not external deliverables (not shared with Business Analysts or Dev teams) but are important for internal QA consumption. They are our first step towards 100% test coverage goal. Test scenarios once complete undergo a peer review and once that is done, they are all consolidated. For more details on how QA documents are reviewed, check out the article: [How to Perform Test Documentation Reviews in 6 Simple Steps.](http://www.softwaretestinghelp.com/test-documentation-reviews/" \o "Test Documentation Reviews)

**#2.** We could use a test management tool like [HP ALM](http://www.softwaretestinghelp.com/learn-hp-quality-center-qc-in-4-days/" \o "HP ALM Training) or [qTest](http://www.softwaretestinghelp.com/test-management-tool-qtest-review/" \o "qtest review) to create the test scenarios. However, the Test scenarios creation in real time is a manual activity. In my opinion, it is more convenient manually. Since it is step 1 we do not need to bring out the big guns yet. Simple excel sheets are the best way to go about it.

**The next step to this series is that**- we will work on creating test cases and get further into the test designing phase. Before that we will also get into – **What test planning is? Where it fits into the entire QA project? As always, work with us for best results.**

**QA Training Day 3:** [How to write SRS document from scratch.](http://www.softwaretestinghelp.com/how-to-write-test-plan-document-software-testing-training-day3/" \o "How to Write a Test Plan Document from Scratch (Download a Real Plan))