|  |  |  |
| --- | --- | --- |
| NOMBRE USUARIO | VERSIONAMIENTO | FECHA - HORA |
| Angela valdez | 1.0 |  |
| Lourdes villca | 1.1 |  |
| Ericka Viraca | 1.2 | 07/05/2016 16:01 |
| Julieta Escalera | 1.3 | 07/05/2016 16:33 |
| Yuri Saravia | 1.4 | 08/05/2016 09:18 |
| Olivia Reyes Parra | 1.  1.6 | 08/05/2016 17:00 |

Well, my change is a correction, we forget the versioning table manual, we must complet.

Versioning first change

## SOFTWARE QUALITY ASSURANCE

Software quality assurance (SQA) consists of a means of monitoring the [software engineering](https://en.wikipedia.org/wiki/Software_engineering" \o "Software engineering) processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as [ISO 9000](https://en.wikipedia.org/wiki/ISO_9000" \o "ISO 9000) or a model such as [CMMI](https://en.wikipedia.org/wiki/CMMI" \o "CMMI).

SQA encompasses the entire [software development](https://en.wikipedia.org/wiki/Software_development" \o "Software development) process, which includes processes such as requirements definition,[software design](https://en.wikipedia.org/wiki/Software_design" \o "Software design), [coding](https://en.wikipedia.org/wiki/Computer_programming" \o "Computer programming), [source code control](https://en.wikipedia.org/wiki/Revision_control" \o "Revision control), [code reviews](https://en.wikipedia.org/wiki/Code_review" \o "Code review), [software configuration management](https://en.wikipedia.org/wiki/Software_configuration_management" \o "Software configuration management), [testing](https://en.wikipedia.org/wiki/Software_testing" \o "Software testing), [release management](https://en.wikipedia.org/wiki/Release_management" \o "Release management), and product integration. SQA is organized into goals, commitments, abilities, activities, measurements, and verifications.

Software quality assurance, according to [ISO/IEC 15504](https://en.wikipedia.org/wiki/ISO/IEC_15504" \o "ISO/IEC 15504) v.2.5 (SPICE), is a supporting process that has to provide the independent assurance in which all the work products, activities and processes comply with the predefined plans and ISO 15504

## TEST CASES

If someone wants know more about TEST CASES, I leave some links below:

http://www.kaner.com/pdfs/GoodTest.pdf

<http://www.elementool.com/ebook/SoftwareTestCases.pdf>

<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf>

And for everybody a joke:

[](https://www.google.com.bo/imgres?imgurl=http://www.guy-sports.com/fun_pictures/computer_bag.jpg&imgrefurl=http://www.guy-sports.com/humor/computers/computer_jokes.htm&docid=pfKLJz36SeVbyM&tbnid=UwaY4jJRrHpUzM:&w=308&h=250&bih=537&biw=784&ved=0ahUKEwiktKiv3cjMAhXGpB4KHehxDAQQMwgqKA8wDw&iact=mrc&uact=8)

A **test case**, in [software engineering](https://en.wikipedia.org/wiki/Software_engineering" \o "Software engineering), is a set of conditions under which a tester will determine whether an [application](https://en.wikipedia.org/wiki/Software_application" \o "Software application), [software system](https://en.wikipedia.org/wiki/Software_system" \o "Software system) or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a *[test oracle](https://en.wikipedia.org/wiki/Oracle_(software_testing)" \o "Oracle (software testing))*. In some settings, an oracle could be a [requirement](https://en.wikipedia.org/wiki/Requirement" \o "Requirement) or [use case](https://en.wikipedia.org/wiki/Use_case" \o "Use case), while in others it could be a [heuristic](https://en.wikipedia.org/wiki/Heuristic" \o "Heuristic). It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as *[test scripts](https://en.wikipedia.org/wiki/Test_script" \o "Test script)*, particularly when written - when they are usually collected into [test suites](https://en.wikipedia.org/wiki/Test_suite" \o "Test suite).

**HOW TO WRITE A BUG REPORT**The report needs to be clear; having all the required information, if the report is well written, other testers, managers and developers will easily understand it.

Consider the following tips to write a good defect report.

1. **CONDENSE:** Say it clearly but briefly, this is applicable for all the defect’s fields especially the title.
2. **ACCURATE:** Is it a defect or could it be user error, misunderstanding?
3. **NEUTRALIZE:** Just the facts. No zingers. No zingers. No humor. No emotion.
4. **PRECISE:** Explicitly, what is the problem?
5. **ISOLATE:** What has been done to isolate the problem?
6. **GENERALIZE:** What has been done to understand how general the problem is?
7. **RECREATE:** What are the essentials in triggering/re-creating this problem? (environment, steps, conditions)
8. **IMPACT:** What is the impact to the customer? What is the impact to be tested? Sell the defect.
9. **DEBUG:** What does development need to do, to make it easier to debug? (traces, dumps, logs, immediate access, etc)
10. **EVIDENCE:** What documentation will prove the existence of the error?

Evidence may take the form of documentation from user guides, specifications, requirements, and designs. It may be past comments from customer de-facto standards from competing products, or results from previous versions of the product.