

QualiTeam: A Support Tool When Learning Software Quality and Testing Concepts

Jorge Cervantes-Ojeda, María del Carmen Gómez-Fuentes 

Department of Applied Mathematics and Systems, Universidad Autónoma Metropolitana, Mexico City, Mexico

Email: jcervantes@cua.uam.mx, mgomez@cua.uam.mx

How to cite this paper: Cervantes-Ojeda, J. and del Carmen Gómez-Fuentes, M. (2022) QualiTeam: A Support Tool When Learning Software Quality and Testing Concepts. *Journal of Software Engineering and Applications*, 15, 1-17.

<https://doi.org/10.4236/jsea.2022.151001>

Received: October 10, 2021

Accepted: January 24, 2022

Published: January 27, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

QualiTeam is a web application to support the teaching-learning process on Software Quality Assurance, Quality Control and Testing introductory concepts. It has two main objectives: to facilitate the understanding of concepts learned in theory and to facilitate the monitoring of SW projects that students develop. The system gives the teacher control and the students a guide on the activities that must be carried out throughout a software project development. QualiTeam is a tool conceived to help in the challenge of providing students with concrete examples with which they can practice and clarify the topics taught in the classroom. With it, students can apply concepts that, in the initial training of a software engineer, are generally taught only at a theoretical level such as: review process, change requests, trouble reports, document version control and testing documentation management. QualiTeam is free and available online. It has been in operation for 5 years, through which improvements have been made until achieving a quite stable version.

Keywords

Software Engineering Education, Review Process, Testing Documentation, Teaching of SW Quality Assurance

1. Introduction

With the expansion of the use of computers throughout society and the consequent growth in the size of programs, Quality Assurance (QA), Quality Control (QC) and Testing have become fundamental aspects of Software (SW) development. There is no SW product, with a minimum degree of complexity, which is exempt from defects that, at least potentially, generate failures. Since SW systems are used more and more in devices in which the consequences of failures are extremely serious (death, failure in important missions, security breaches, financial

or social losses), it is no longer feasible to develop a product without using QA, QC and Testing that provide the necessary quality level. This is why Quality and Testing are part of the knowledge areas described in the Guide to the Software Engineering Body of Knowledge (SWEBOK) [1]. So, it is important that future SW developers have a good understanding of concepts related to SW QA, QC and Testing.

A “Quality and Testing” course addresses the fundamental principles of SW QA, QC and Testing documentation procedures. These aspects share the same academic challenge: the difficulty of practicing in a school laboratory the concepts that are learned in theory. QualiTeam is a web application that supports the control of documentation associated with different projects, including the administration of test documents in accordance with the IEEE-829 Standard [2]. With QualiTeam it is possible to practice several of the quality control procedures, which help students to better assimilate the concepts learned in a theoretical class. With QualiTeam, students can also practice the generation and control of Change Requests (CR) and Trouble Reports (TR). Additionally, students and teachers can use it to coordinate the teamwork of any scholar SW project.

Scarpino and Kovacs [3] reported the major problems that arose while applying a Software Quality Assurance (SQA) tool at a Company, and they emphasize the need of having qualified SW quality assurance personnel, confirming the importance of making an effort to improve training in this area. QualiTeam is meant to help in the teaching process of specific topics of this area.

Looking for the work that has been done regarding to teaching SW QA, QC and Testing, we found that Laporte *et al.* [4] reported the use of lectures, practical exercises and group projects for the teaching of quality assurance. Although they specify the standards on which they are based to teach each of the topics, they do not specify the way in which theoretical concepts are put into practice. On the other hand, they do not use a specialized SW tool for practical exercises. Regarding to tools to support activities related to these topics, Ibarra and Muñoz [5] made a review of tools and practices implemented for SQA and they found that more frameworks than tools have been developed. Muñoz reported in [6] that most of the tools for SQA are focused on the execution of dynamic tests to solve code problems, as well as on the continuous integration of software components. However, as it was already noticed by [7], quality assurance isn't just testing but also methods and approaches. The tool proposed in [5] is meant to be used by companies to support their SQA activities, while QualiTeam is meant to be used by undergraduate students in order to reinforce specific concepts included in syllabus.

As far as we know, there are no SW tools similar to QualiTeam, which is devoted to being used for students in order to put into practice what was learned in theory.

It is worth mentioning that in this work we explain some of the SQA and Testing subjects and their relationship with QualiTeam, however we will not give a detailed explanation of each of the subjects. The structure is as follows:

Section 2 contains a brief QualiTeam description. Section 3 has an explanation of the QualiTeam basic operation. Section 4 explains how to work with the QualiTeam subsystems. Sections 5 and 6 have results and conclusions respectively.

2. QualiTeam Description

2.1. QualiTeam Scope

In the SW industry, Quality Control (QC) means looking for defects and correcting them, at the lowest possible cost, and before they produce failures [8]. In general, this is achieved with tests specifically designed to verify that the SW does not fail, that is, that it meets the requirements. This verification is carried out throughout the development process and, therefore, on all generated intermediate products (milestones) are verified. The idea is to detect defects at the earliest possible stage to prevent them from spreading to the following stages, since the costs derived from their correction would be greatly increased [9]. QualiTeam is a tool to help during the process of software developers' training on SW QA, QC and Testing concepts. It supports control of documents associated with different projects. Documents are not tied to a specific software development model, and they can be intermediate products like: requirements specification, design document, code modules and also the documents that indirectly affect them: documentation rules, coding rules, procedure descriptions, test documents, etc.

2.2. QualiTeam Technical Features

QualiTeam is free and it can be accessed at <http://qualiteam.cua.uam.mx:8080/QualiTeam/>.

(To use QualiTeam, users must first register).

QualiTeam was developed in NetBeans and JavaServer Faces combined with Rich Faces libraries for the front-end. It uses MySQL for the database. QualiTeam has the following subsystems: Project Manager, Security, Document Manager, Change Requests, Error Reports and Tests.

In order to upload QualiTeam and making it available on the network, we configured a server with GlassFish, running on a Centos operating system, and MySQL server is used for the database.

Currently QualiTeam is in Spanish, however, if any non-Spanish-speaking University requests it, we will proceed to make the English version.

2.3. The QualiTeam Sub-Systems

The QualiTeam subsystems are illustrated in **Figure 1**.

Security module, users' access to the system, *Project Manager* manages the projects and allows access to the following subsystems: *Error Reports*, *Change Requests*, and *Tests*. The QualiTeam modules, except *Security*, use the *Document Manager* subsystem, which is a service subsystem. The main functions of

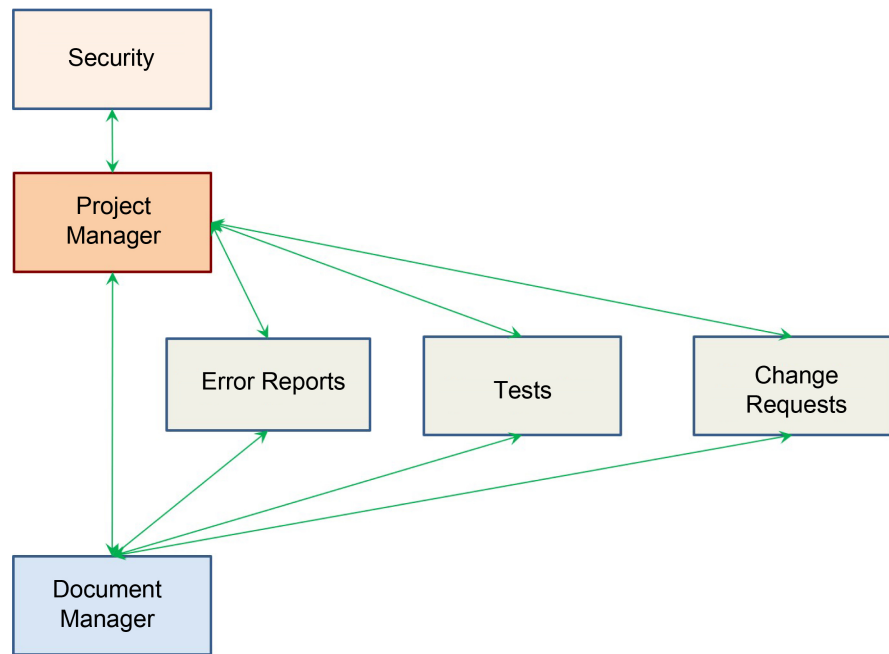


Figure 1. QualiTeam subsystems.

each of the subsystems are briefly mentioned below.

- *Security*: this subsystem is in charge of managing users and validating that the person who wants to enter the system correctly provides their user ID and password.
- *Project Manager*: manages the project members (users) and the main project documents. Once a project has been selected, it allows the access to: Error Reports, Change Requests and Tests subsystems.
- *Tests*: this subsystem is used to manage the different test documents, which are: Test Plan, Test Procedures, Test Logs, Test Designs and Test Cases.
- *Change Requests (CR)*: allows creating Change Requests on base documents (from previous projects) to build documents of a new project.
- *Error Reports*: it is used to manage the error reports generated on deliverable products (documents or the code of a project).
- *Document Manager*: provides services to the other subsystems. It is responsible for creating new documents and new versions of documents in the system, download and upload documents to and from the server, and displaying their properties. It also supports the document review process, that is, distribution lists, comments and change of status.

2.4. The QualiTeam Software Development Process

QualiTeam was developed following the waterfall process with subprojects. To connect the requirements phase with the design phase, the User Interface (UI) mockups were made and also we defined the UIs flow with the User Interface Transition Diagrams (UITD) [10]. Then, we defined the system architecture and continued with the detailed design, which was done through Detailed Sequence

Diagrams (DSD) [11]. During the design of the UIs, DTIUs, system architecture and DSDs, a peer review process was done in order to improve their quality. The Requirements Specification, Design, Tests and User Manual documents were generated.

The implementation began with the coding of the system skeleton. With this skeleton, it was possible to distribute the work to different people, each one had to code one or two subsystems. Module tests were carried out in each subsystem and later they were integrated into the whole system, where integration tests were performed. As a final stage, the server was configured and QualiTeam, with its database, were uploaded.

3. QualiTeam Basic Operation

3.1. Project Management

Once the user registers or enters with his/her correct account and password, a list is displayed with the registered projects in the system. The user can select a project from the existing ones or create a new one. When he/she creates a new project, he/she has the role of “leader” of that project. A leader can edit his/her project to modify its name and the members list. The following main documents are automatically generated: Project Plan, Requirements Specification, Design and Test Plan. Project members can also generate new project documents. The user who generates a new document has the role of its author. Users who are not project members can view the documents but they cannot generate new ones neither participate in any of the project’s activities.

3.2. Document Management

It is important to mention that whenever a document is generated, its related data are registered in the database. These document data are called the “Document Details”. However, the content of the document must be uploaded to the server in a PDF format. The “Document Details” interface contains the following data: identity, title, author, version, creation date, status and the effort (in hours) that the author invested in its elaboration along with some action buttons. The “Download Content” button transfers the document content from the server to the user’s computer. The author of a document and the project leader are authorized to edit the document details, so they have the “Edit Properties” button active. This button triggers an interface in which the title, the effort, and the distribution list can be modified. So, the editor can change the “author name” to transfer the authorship of the document to another project member. There is also the “Upload File” section which has an “Add” button in this User Interface to send the content of the document to the server. It is necessary for the document to be uploaded using the PDF name extension but the true format of the file can be just anyone. We suggest that other types of files should be named with their original extension but with an additional pdf extension. For instance, a filename “code.java” should be renamed to “code.java.pdf” before uploading it to QualiTeam.

The list of comments to the document can be accessed through the Document Detail interface with the “View Comments” button.

Figure 2 illustrates the user interface in which the documents of a project are displayed. To view the test documents, change requests or error reports in this project, it is necessary to access each of the subsystems (on the left side of the interface).

4. Working with QualiTeam Subsystems

In order to work with all the QualiTeam features, the user must know: the peer review process, generation of new document versions, testing documentation management, change requests and error reports. These topics are addressed below.

4.1 Document Creation, Revision and Versions

4.1.1. The QualiTeam Peer Review Process

The SW peer review process in IEEE 1028 standard [12] is applicable not only for SW modules but also for the other documents of a SW Project. The procedure for conducting a document peer review is implemented in QualiTeam and it is illustrated in **Figure 3**. The procedure is as follows:

- 1) The author uploads a document and edits the list of persons who will review the document (distribution list).
- 2) The author or the project leader initiates the review process. This action informs all reviewers about their task on the document.
- 3) Reviewers are given a reasonable time to read the document carefully and,

Documentos del Proyecto: MoliGT

Id documento	Título	Versión	Modificado	Estatus	Autor	Detalle	Documento
100-71	Plan maestro de inspección	0.01	2019-06-20 12:57:38	Revisado	Hiram	Ver Detalle	Bajar Contenido
100-71	Plan maestro de inspección	0.02	2019-06-21 12:13:11	Revisado	Hiram	Ver Detalle	Bajar Contenido
100-71	Plan maestro de inspección	0.03	2019-06-21 12:30:33	Revisado	Hiram	Ver Detalle	Bajar Contenido
100-71	Plan maestro de inspección	1.0	2019-06-27 13:19:10	Aceptado	Hiram	Ver Detalle	Bajar Contenido
100-60	Reglas para la creación de manual de usuario	0.01	2019-06-19 16:20:08	En Revisión	2163071450	Ver Detalle	Bajar Contenido
100-52	Actualizar docs aceptados	0.01	2019-06-14 16:50:08	Revisado	2163071450	Ver Detalle	Bajar Contenido
100-52	Actualizar docs aceptados	0.02	2019-06-26 16:36:29	En Revisión	2163071450	Ver Detalle	Bajar Contenido
100-51	Nuevo Documento	0.01	2019-06-14 16:47:23	Rechazado	2163071450	Ver Detalle	Bajar Contenido
...	SOAP 1.6.0 Procedimiento	...	2019-05-30	En	...	Ver Detalle	Bajar Contenido

Figure 2. User interface with the project documents and access to the other subsystems.

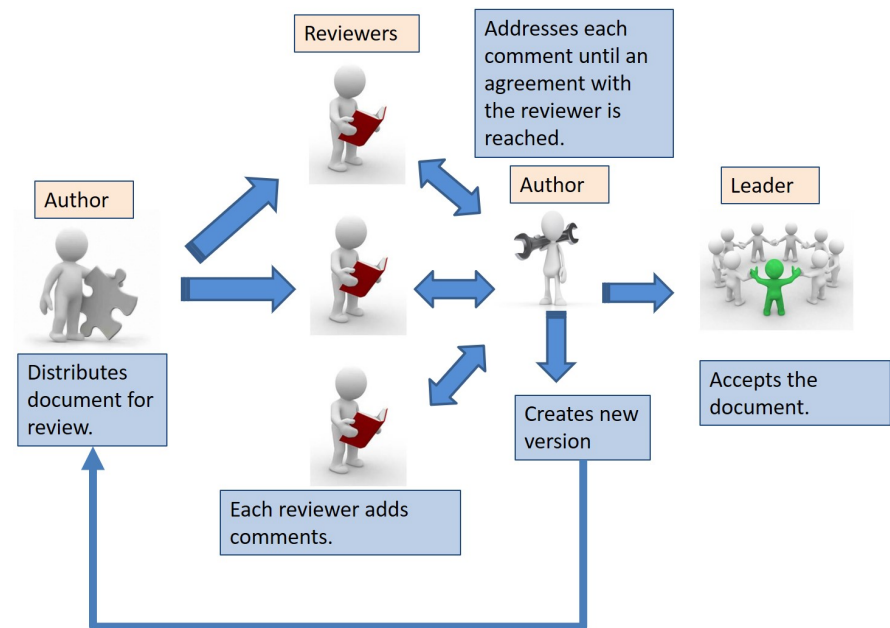


Figure 3. The QualiTeam peer review process.

based on their knowledge, experience and skills, find defects.

4) Based on the found defects, reviewers add comments to the document in a separate conversation thread for each defect. Each comment has the reviewer identity. All reviewers, the author, and the leader can add comments to each thread.

5) After all participants in a thread reach an agreement on what has to be done, the thread may be closed by a reviewer, the author or the project leader.

6) Only the author and project leader have access to the “Create Next Version” button which will be available only when all threads have been closed.

7) When a new version of a document is created, the status of the previous one is set to “Reviewed” and the process repeats from step 1. If there is no need to create a new version, then the document may be “accepted” by the project leader. In this case, a new version of the document is created with the status “Accepted” and with an integer version number, opposed to the fraction version numbers used during the review process.

8) If necessary, a document may also be rejected and thus it would not be part of the documentation of the project. Rejection should only be done when the reviewers’ comments justify this decision.

4.1.2. The Document Peer Review Cycle

When a document is created for the first time, it has the status “*In Creation*” and the version number is automatically 0.01. Once an author gives his/her document the status *In Review*, the team members designated as reviewers can read the document and add comments. Comments to improve the document are entered into the system as explained in the next subsection. The author creates a *new document version* containing modifications to the current version agreed in

the comments.

The current document must be changed to “*In Review*” status in order to be able to create a new version (the “new version” button is displayed). The subsequent version numbers will be automatically increased by 0.01. When a new version is generated, the current document status changes automatically from *In Review* to *Reviewed*, as illustrated in **Figure 4**. A *Reviewed* version cannot be edited.

The project leader gives the document the status Accepted/Rejected when he/she decides to finish the peer review process. Any document in Accepted/Rejected status cannot be edited.

4.1.3. Comments on a Document

The QualiTeam comments system is accessed through the “Comments” button in the “View Details” interface. Any user can view the comments of a document by selecting the “View Comments” button, however, only the members of the reviewers list, the author and the leader have permission to add comments. The user interface to view and enter comments is shown in **Figure 5**. On the left side column, the list of conversation threads is displayed. For each topic to be discussed, it is recommended that a new thread is added with the “New Thread” button. The list of comments associated with a selected conversation thread is displayed by clicking on one of the threads’ “See” buttons.

The author, project leader and reviewers can close a conversation thread with the “Close Thread” button. No one can add more comments to a thread that is closed. If someone wants to make more comments he/she will have to create a new thread.

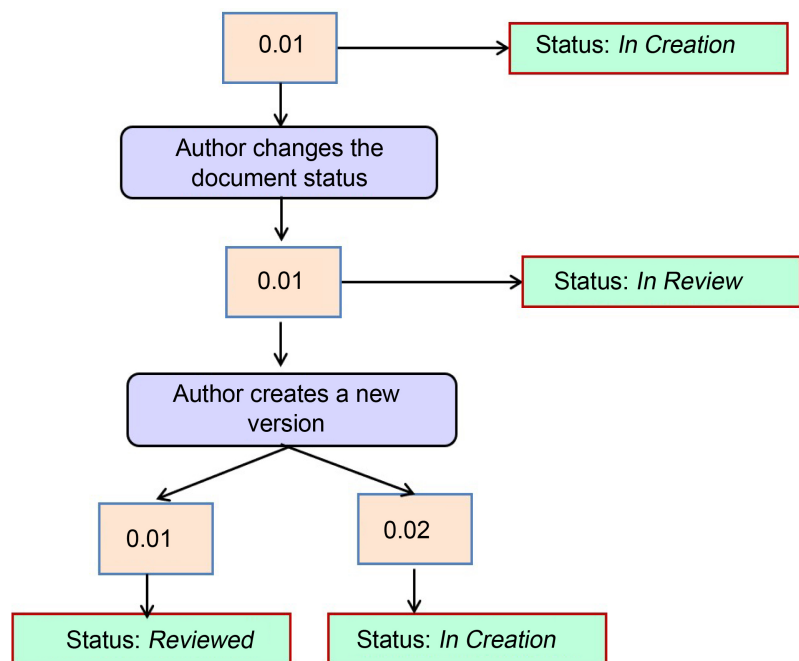


Figure 4. A new document version creation.

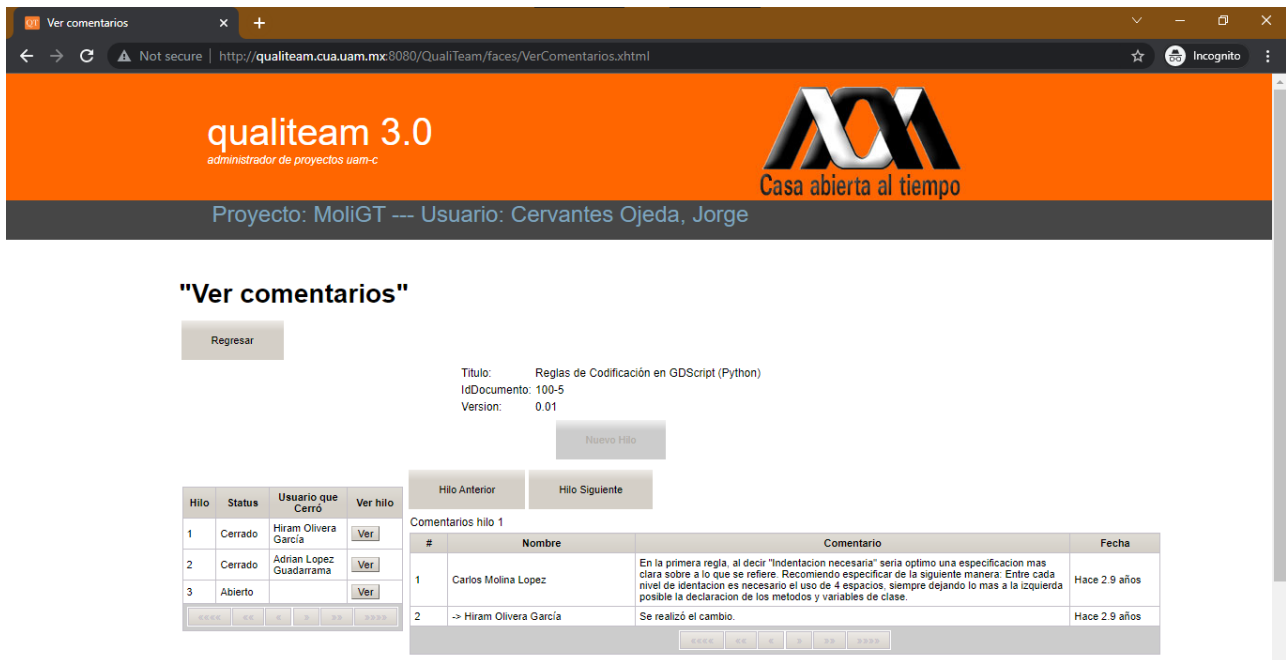


Figure 5. Interface to view and enter comments.

4.2. Testing

When a test is run, the observed behavior is contrasted against the expected behavior of the system or component. The IEEE 610.12-1990 standard [13] defines testing as “the process of operating a system or component under specific conditions, observing and recording the results, and performing an evaluation of some aspect of the system or component”. In order to execute the tests of a system, a structured process is carried out, this process is illustrated in Figure 6 and consists of test planning, test design, and test execution. Additionally, a control is carried out to supervise the design and execution of the tests. The planning of the tests is documented in the test plan. In it, the techniques and methods to be used are specified, as well as the environment, the necessary tools and the schedule. During the test design, the elements to be tested are identified. Test data and conditions are also specified depending on the elements to be tested. Also, the structure of the tests and the environment they require are designed. The tests are structured, organized by groups and categories, and prioritized based on a risk analysis.

Tests documents are managed in QualiTeam according to the IEEE-829 Standard [2]. To start the QualiTeam test subsystem, the desired project must be first selected and then click on the “tests” button. Figure 7 illustrates the documents that must be generated during a testing process. The *test plan* is one of the mandatory documents that are created automatically when a new project is generated. *Test plan* contains the schedule of the test activities, objectives and requirements of the tests, personnel who will do the tests, software and hardware resources, etc. The project leader uploads the content of the *test plan* document to the project.

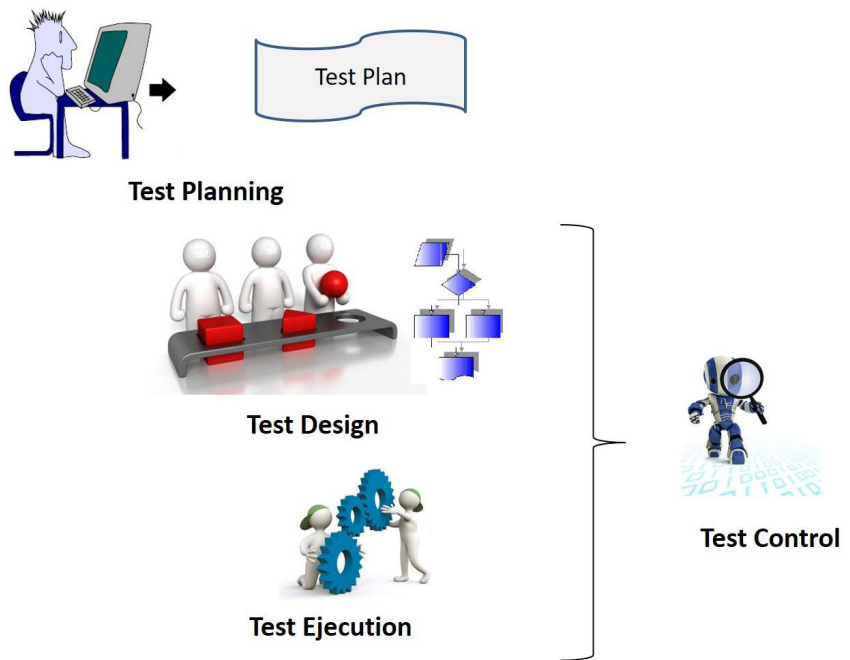


Figure 6. The test process.

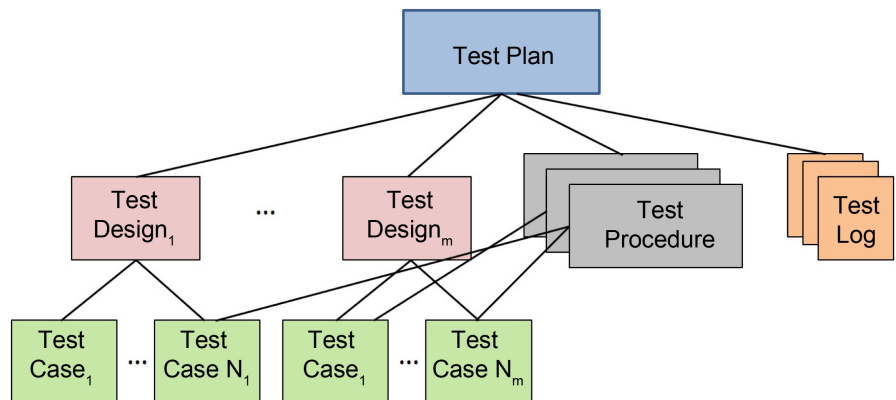


Figure 7. Test documentation in QualiTeam.

In QualiTeam, a *test design* groups together a set of *test cases*. Each *test design* includes the description of the common characteristics of this set's test cases. Each of the *test cases* describes only one test. On the other hand, a *test procedure* describes how the tests should be run. Then, a *test case* is associated with a *test design* and can be associated with a *test procedure* if necessary. There may be several *test procedures* associated with the *test plan*, as indicated in Figure 7.

Finally, a *test log* contains the information about the tests that were carried out and the obtained results such as whether the test was successful or not. Each *test log* is associated with the project's *test plan*. There can be several *test logs* associated with a *test plan*.

4.3. Trouble Reports and New Versions Generation

An event that occurred during the execution of a test or at any time after a

product (document) has been accepted that requires correction or clarification must be recorded in a Trouble Record (TR) and linked to some Accepted document. Every TR must go through the peer review process. At the end of this process, it has a final status: *Accepted* or *Rejected*. A TR is generated for each found defect, and when the project leader authorizes, a new version of the product (document or SW module), which includes attention to all generated TRs for it, is released. This is, the attention to the TR of a product leads to a new version containing modifications. The new version must include the list of attended TRs and their implementation. It is worth mentioning that this new version must go through a revision procedure in order to ensure its completeness and consistency.

QualiTeam's Trouble Records subsystem allows generating and viewing all the TRs of a project. To start the QualiTeam Trouble Records subsystem, the desired project must be first selected and then click on the "Trouble Reports" button.

The TRs subsystem shows the list of Trouble Reports generated for the accepted documents of a particular project. In the example in **Figure 8**, three TRs for the "Design" document are shown and one for the "Requirements" document. The TRs review process is the same as for any document, that is, it has the status "*In Creation*" first, and then it is set to "*In Review*" by the author and so on.

Each time the button "Create TR" is clicked, a list is displayed with all the accepted documents in the project, since it does not make sense to make a TR on a document that has not been accepted yet. The new TR will be associated with the selected "base document".



IdTRecord	Título TRecord	Version	Estatus	IdDoctoBase	Título DoctoBase	Versión DoctoBase	Detalle
100-17	Nuevo Reporte de Error	0.01	Revisado	100-1	Plan de Proyecto	1.0	Ver Detalle
100-17	Nuevo Reporte de Error	1.0	Aceptado	100-1	Plan de Proyecto	1.0	Ver Detalle
100-17	Nuevo Reporte de Error	1.01	Revisado	100-1	Plan de Proyecto	1.0	Ver Detalle
100-17	Nuevo Reporte de Error	2.0	Aceptado	100-1	Plan de Proyecto	1.0	Ver Detalle
100-25	Nuevo Reporte de Error	0.01	En Creacion	100-17	Nuevo Reporte de Error	2.0	Ver Detalle
100-46	Reporte para el plan maestro	0.01	Revisado	100-71	Plan maestro de inspección	1.0	Ver Detalle
100-46	Reporte para el plan maestro	1.0	Aceptado	100-71	Plan maestro de inspección	1.0	Ver Detalle
100-47	Reporte Reglas de doc	0.01	Revisado	100-10	Reglas de Documentación	1.0	Ver Detalle
100-47	Reporte Reglas de doc	1.0	Aceptado	100-10	Reglas de Documentación	1.0	Ver Detalle
100-50	Nuevo Reporte de Error	0.01	En Creacion	100-17	Nuevo Reporte de Error	2.0	Ver Detalle
100-8	Nuevo Reporte de Error	0.01	Rechazado	100-1	Plan de Proyecto	1.0	Ver Detalle
100-9	Nuevo Reporte de Error	0.01	Rechazado	100-1	Plan de Proyecto	1.0	Ver Detalle

Figure 8. List of trouble records for a project.

Attention to Trouble Reports: Creating a New Version of the Product

Section 5.1.2 explains how the decimal numbering of the versions of a document works: 0.01, 0.02, etc. When a document is accepted, an integer version number (1.0, 2.0, etc.) is generated. In order to incorporate TRs into a new version of their base document, the author of that base document must create a new version of it. The version number of that new version is its next decimal version. For example, 1.01, and the document must pass through the peer review process. After that version, the subsequent decimal versions are: 1.02, 1.03, etc. When the document is accepted, QualiTeam automatically assigns the next integer number version, which in this example would be 2.0. **Figure 9** illustrates the way in which new version numbers, decimal and integer, are generated.

4.4. Change Requests

When Requirements of a new project are very similar to those of previously released projects, then the Change Requests (CR) procedure may be used. The idea of a CR is to reuse documents and/or code from previous projects in order to optimize the documentation and configuration of a new project. CRs are documents describing the modifications to be made to an already accepted product, which is called the base product. A product can be a document (requirements specification, design, tests, etc.) or a SW module from a different project. These modifications are not originated from a defect in the base document but, instead, they arise from the need to make changes to the base project to build a new similar project that does not necessarily replace the base project. Through practice, we observed that there is confusion when teaching the CR concept. With QualiTeam CRs subsystem, students can practice with CRs and clarify its use.

As it is illustrated in **Figure 10**, the documentation of a new project may comprise CRs to base products along with the corresponding base product,

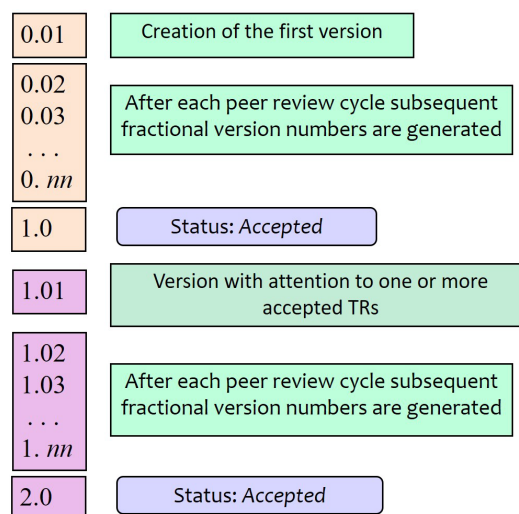


Figure 9. Sequence of decimal and integer version numbers of a document.

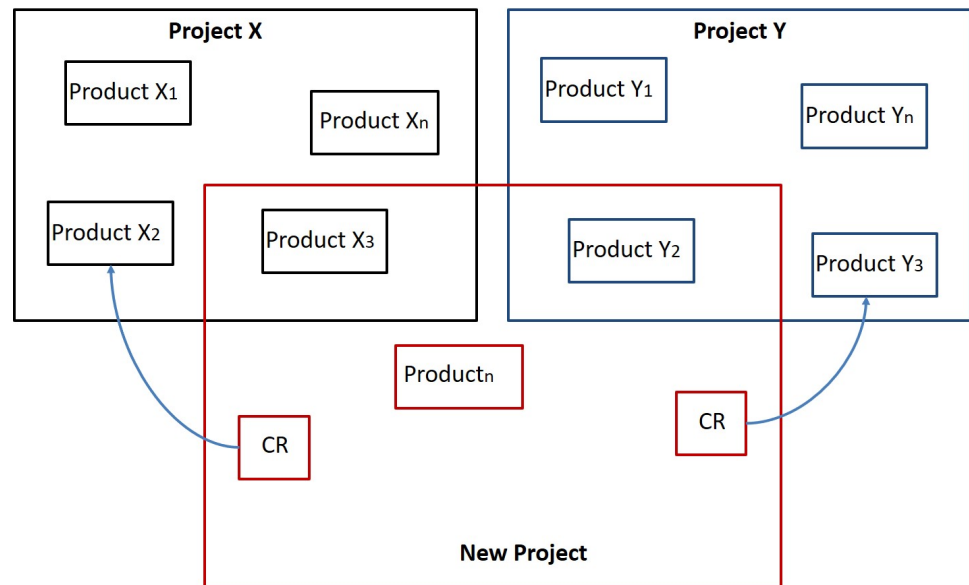


Figure 10. A new project from products of previous projects.

products generated from the merging of base products and their CR, new products and even unchanged products from previous projects.

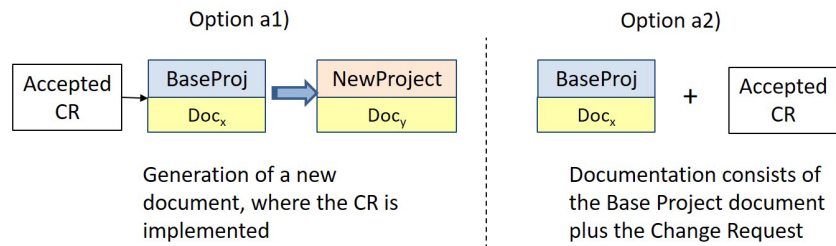
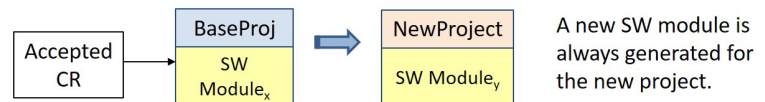
Implementation of Change Requests

Figure 11 shows that, when the product is a document (not code), there are two ways to use a CR. In the first option (**Figure 11 a1**), the base product and the CR are merged in a new product that would replace the both. In the second option (**Figure 11 a2**), the documentation includes both: the base document and the accepted CR on that base document. The latter case is useful when the CR describes only a few changes. All CR documents must be reviewed and eventually accepted to be part of the project's documentation.

When a CR is created on a software module, a new software module is always generated for the new project, and it is the result of merging the base project module with the CR, as it is shown in **Figure 11(b)**.

The QualiTeam CR subsystem is accessed by selecting a Project and then clicking on the "Change Requests" button. The CR subsystem displays the CR list of the current project. In the example of **Figure 12**, the CR subsystem shows two CRs, CR 5-5 was made on the "Requirements Specification" document in Project "QualiTeam 1.0" and CR 5-6 was made on the "Design" document also in "QualiTeam 1.0".

Every time the "Create Change Request" button is clicked, an interface is shown as in **Figure 13**, that shows the list of all projects, except the current project (in our example "QualiTeam 2.0"). When a base project is selected, its documents in accepted status are displayed on the right. To create a new CR, it is necessary to first select a *base project* and then the document to which the CR will be associated, that is, the *base document* to which the changes described in the CR must be applied.

a) Documents**b) Software Modules****Figure 11.** Change request implementation.

Regresar		Crear Solicitud de Cambio					
Id	Título Sol. de Cambio	Versión	Estatus	Id Docto. Base	Título DoctoBase	Versión Docto Base	Detalle
65-14	schematopracing02.sql	0.01	En Revision	64-19	schematopracing02.sql	1.0	Ver Detalle
65-15	Model.java	0.01	En Revision	64-47	Model.java	1.0	Ver Detalle
65-16	UI.java	0.01	En Revision	64-50	UI.java	1.0	Ver Detalle
65-17	ModelForView.java	0.01	En Revision	64-60	ModelForView.java	1.0	Ver Detalle
65-18	EditEventBean.java	0.01	En Revision	64-73	EditEventBean.java	1.0	Ver Detalle
65-19	editevent.xhtml	0.01	En Revision	64-9	editevent.xhtml	1.0	Ver Detalle
65-20	MainBean.java	0.01	En Revision	64-91	MainBean.java	1.0	Ver Detalle
65-5	Requerimientos Quinietas	0.01	En Revision	63-194	Especificación de requerimientos	1.0	Ver Detalle
65-6	Diseño Quinietas	0.01	En Revision	63-194	Especificación de requerimientos	1.0	Ver Detalle
65-7	Controller.java	0.01	En Revision	64-10	Controller.java	1.0	Ver Detalle

Figure 12. CRs list of the current project.**5. Results**

QualiTeam was launched at the beginning of 2017. Through these 5 operation years the following maintenance activities have been done:

- *Correction of the non-detected defects during the testing phase*—Most defects were corrected during the first months of the system operation. The current version is already quite stable.
- *Improved features*—Users' feedback allowed us to make improvements on the review process operation, specifically in the conversation threads for the reviewers' comments. Improvements were also made to the information presentation in some of the user interfaces.

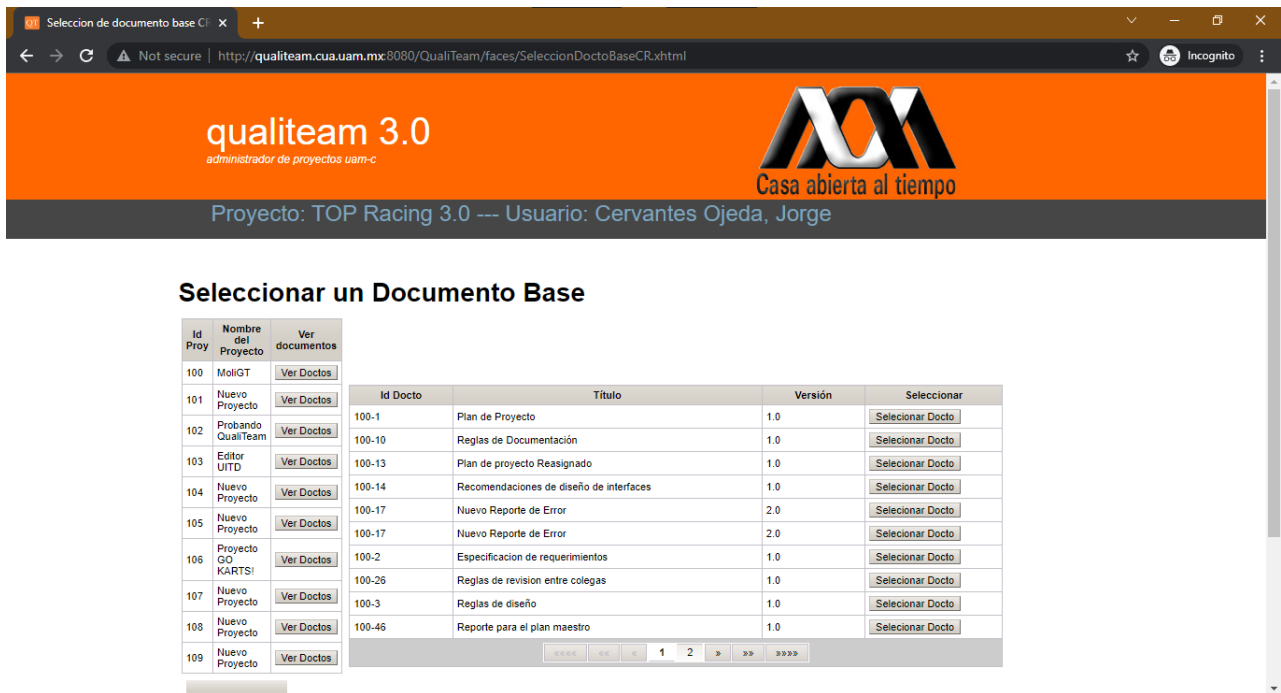


Figure 13. Selecting a base document within the selected base project.

- *Code refactoring*—The code has been refactored in order to facilitate its maintenance, improve performance and solve some bugs. Current version is 3.0.

Our students have reinforced what they learned in the SQA and Testing lectures by doing the “Practices with QualiTeam” included in [14]. This book is available on internet and it is free.

We have taught the course “Quality and Testing” to undergraduate students of Computer Engineering since 2008. In the first 9 years (2008-2016) QualiTeam was not available to support the classes, while the following 5 years the students had QualiTeam to apply the procedures learned in theory. One of the course assignments is to document the procedures seen in class and apply them during the development of a project. We noticed a general improvement between the procedures’ descriptions prepared by the students from 2008 to 2016 and those prepared by the students from 2017 to 2021. In some cases, the procedures were improved by students by introducing adaptations to their particular projects when using QualiTeam. Even during the years 2020 and 2021, in which the courses were taught online due to the Covid-19 pandemic, the procedures’ descriptions prepared by the students indicated that they were well understood.

On the other hand, it is important to encourage teamwork among students, since most of the projects in the software industry are done by teams. QualiTeam records the work of each author and each document reviewer. This makes it easier for teachers to detect to what extent students actually collaborate with the project. This helped in the students’ evaluation and encouraged them to remain active in producing meaningful work for the team, since they experienced

in practice getting a bad grade when their participation in the teamwork was not as expected.

QualiTeam is maintained by the authors, at Universidad Autónoma Metropolitana.

6. Conclusions

The training of students in the area of SW Quality Assurance leads to a challenge which is: providing students with concrete examples with which they can practice and clarify the topics taught in the classroom. Jaccheri [15] proposed an interesting approach which consists of having interactions with the SW local industry. While these interactions are undoubtedly beneficial for future SW developers' training, it is worth mentioning that finding companies willing to collaborate with students can be quite difficult. Moreover, linking companies with students is out of the reach of many teachers/researchers. QualiTeam was conceived to help overcome this challenge.

QualiTeam is a free tool that helps in the teaching-learning process of the following software quality assurance procedures: peer review process, trouble reports, change requests, document versions control and management of tests documentation according to the IEEE-829 Standard [2]. With QualiTeam, the teacher can see the record of the individual contributions to a project of each participant, which helps in guiding students when someone does not fully understand a process. Additionally, the record of each student's work promotes a more equitable participation during the development of team projects and helps the educator in the students' assessments.

According to [16] various studies have demonstrated that blended learning systems, a mixture of face-to-face and online education, have important advantages over fully face-to-face or fully online courses. QualiTeam made remote work much easier during the online classes in the present Covid-19 pandemic suggesting that it could be adopted by online and blended education.

QualiTeam has successfully helped the teaching of "Quality and Testing" at the Universidad Autónoma Metropolitana in Mexico City for five years now and we hope that other universities can also benefit from this teaching tool. Users' comments are always welcome; thanks to them we can keep on improving the tool.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Abran, A., Moore, J.W., Bourque, P., Dupuis, R. and Tripp, L. (2004) Software Engineering Body of Knowledge. *IEEE Computer Society, Angela Burgess*.
- [2] Software Engineering Technical Committee (1998) IEEE Standard for Software Test

- 829-1998. *IEEE Computer Society*, **345**.
- [3] Scarpino, J. and Kovacs, P. (2008) An Analysis of a Software Quality Assurance Tool's Implementation: A Case Study. *Journal of the International Association for Computer Information Systems*, **9**, 9.
 - [4] Laporte, C.Y., April, A. and Bencherif, K. (2007) Teaching Software Quality Assurance in an Undergraduate Software Engineering Program. *Software Quality Professional*, **9**, 4.
 - [5] Ibarra, S. and Muñoz, M. (2018). Support Tool for Software Quality Assurance in Software Development. 2018 *7th International Conference on Software Process Improvement (CIMPS)*, Guadalajara, 17-19 October 2018, 13-19.
<https://doi.org/10.1109/CIMPS.2018.8625617>
 - [6] Muñoz, M. (2019) Disminuyendo la Brecha del Aseguramiento de la Calidad entre Gestor y Miembros del Equipo Mediante el Uso de Herramientas de Software. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 18-32.
<https://doi.org/10.17013/risti.31.18-32>
 - [7] Feldman, S. (2005) Quality Assurance: Much More than Testing: Good QA Is Not Only about Technology, but also Methods and Approaches. *Queue*, **3**, 26-29.
<https://doi.org/10.1145/1046931.1046943>
 - [8] Godbole, N.S. (2005) Software Quality Assurance, Principles and Practice. Alpha Science International Ltd., Oxford.
 - [9] Horch, J.W. (2003) Practical Guide to Software Quality Management. Artech House Publishers, Norwood.
 - [10] Gómez, M.C. and Cervantes, J. (2013) User Interface Transition Diagrams for Customer-Developer Communication Improvement in Software Development Projects. *Journal of Systems and Software*, **86**, 2394-2410.
<https://doi.org/10.1016/j.jss.2013.04.022>
 - [11] Gómez-Fuentes, M.C. and Cervantes-Ojeda, J. (2019) Sequence Diagrams Tailored for Software Design Used to Build a Carpooling Management System. 2019 *7th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Mexico City, 23-25 October 2019, 116-122.
<https://doi.org/10.1109/CONISOFT.2019.00025>
 - [12] IEEE Standard (2018) IEEE 1028-2008—IEEE Standard for Software Reviews and Audits. <https://standards.ieee.org/standard/1028-2008.html>
 - [13] IEEE Standard (1990) IEEE 610.12-1990—IEEE Standard Glossary of Software Engineering Terminology, 1990. https://standards.ieee.org/standard/610_12-1990.html
 - [14] Cervantes, J. and Gómez, M.C. (2017) Calidad y Pruebas en el Desarrollo de Software. Universidad Autónoma Metropolitana, México.
<http://www.cua.uam.mx/biblioteca-dr-miguel-leon-portilla/acceso-a-los-recursos-de-informacion/libros-electronicos/division-de-ciencias-naturales-e-ingenieria/8>
 - [15] Jaccheri, M.L. (2001) Software Quality and Software Process Improvement Course Based on Interaction with the Local Software industry. *Computer Applications in Engineering Education*, **9**, 265-272. <https://doi.org/10.1002/cae.10000>
 - [16] Nácher, M.J., Badenes-Ribera, L., Torrijos, C., Ballesteros, M.A. and Cebadera, E. (2021) The Effectiveness of the GoKoan E-Learning Platform in Improving University Students' Academic Performance. *Studies in Educational Evaluation*, **70**, 101026.
<https://doi.org/10.1016/j.stueduc.2021.101026>