QUALITY MANAGEMENT AND IMPROVEMENT

To implement a quality plan

Jordi Rafael Lazo Florensa

Tiberiu Paiu

Roger Larriba Baudilio

Sergi Puigpinós Palau

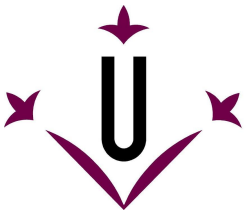
José Ramón Noguero de Llano

Xavier Loncà Sánchez

Senadin Puce

6 June 2021

Degree in computer engineering



Universitat de Lleida
Escola Politècnica Superior

0- Introduction

This document presents the design of a quality plan for our project called "Joint Project". The main focus of this project is to digitize the company Bookiernes, S.A.
Bokiernes,S.A. is a company whose activities focus on publishing and supplying paper books. This company has seen its sales and customers being reduced.
Taking into account the current situation described, the company has decided to implement a system that allows computerization of the workflow which will avoid economic losses, time and information between work processes. Likewise, it is intended that the system to be implemented centralizes the workflow and communication of the ones who are involved in publishing and supplying digital books.
Therefore, this project is based on digitizing the editorial so that it becomes an online bookstore.
That's where our quality plan comes into place, by defining such a plan we will be able to carry out this project with as few errors as possible and ensure that the software developed for bookiernes meets their standards and accomplishes their main goal which is to have better communication within the company. Then the main goal for this quality plan, is to define within this document, all the methods to apply during the development of the web application or project for the company, to ensure the quality of this is good enough to be released.

1- List of quality goals

1.1 Objectives

Objective 1:
Digitalize and centralize the workflow within the publisher, improving the communication of those involved in the editing and supply of books.

Objective 2:
Improving the distribution service of books produced by the publisher, implementing a web portal.

1.1 Qualitative

The data are generally nonnumerical.
Qualitative research relies on data obtained by the researcher from first-hand observation, interviews, questionnaires, focus groups, participant-observation, recordings made in natural settings, documents, and artifacts.
List of the qualitative goals for the company:

1. Improve worker's communication between workers
2. Improve workers/CEO happiness.
3. Computerize and centralize the workflow inside the editorial
4. Improve the distribution service of the books produced by the editorial.

Qualitative	
Specific	Make sure your quality objectives detail precisely what you want to accomplish, who is affected and which business processes are involved in the change.
Measurable	You should be able to measure your objectives. Word each goal in terms of percentages, dollar amounts or other quantifiable results.
Achievable	While your goals should push you toward greater success, they should also be realistic. Ask yourself how you can accomplish this goal.
Relevant	Don't just create random goals to satisfy ISO requirements. Set beneficial objectives that matter to you, your company and your bottom line.
Time-bound	For your goal to be useful, you must have a timeline to base it on. Plan to achieve your objective in the next quarter or year, and ask yourself what you can do each day to reach it within this period.

Table 1. *Qualitative goals.*

1.2 Quantitative

The data are generally numerical.
Quantitative research is a research strategy that focuses on quantifying the collection and analysis of data. It is formed from a deductive approach where the emphasis is placed on the testing of theory, shaped by empiricist and positivist philosophies

Quantitative	
Customer service	Maintain a customer satisfaction rate of X percent.
Performance	Increase product performance to x hours of use.
Efficiency	Improve operational efficiency by X percent.
Safety	Have zero safety incidents in the workplace or zero product recalls.
Delivery	Achieve X percent of on-time deliveries.

Table 2. *Quantitative goals.*

2- Review activities

This section outlines the “How” for the Plan, including the specific actions that should be taken to carry out the Plan.

2.1 Installation Plan Review

2.1.1 Scope

Thorough planning ensures that you have everything to meet all the prerequisites for the successful setup and installation of the system.
Planning minimizes errors during installation and enables a quicker upgrade or installation. This planning information helps you integrate the system into your data center, plan power and environmental needs, and prepare for any unique configurations you need to handle.

2.1.2 Type of review

The Software Installation Plan (SIP) is a plan for installing software at user sites, including preparations, user training, and conversion from existing systems.
The SIP is developed when the developer will be involved in the installation of software at user sites and when the installation process will be sufficiently complex to require a documented plan. For software embedded in a hardware-software system, a fielding or deployment plan for the hardware-software system may make a separate SIP unnecessary.

2.1.3 Schedule

To review an installation plan it is first necessary to review and verify the steps already established to see that they work correctly.

1. Install Docker
2. Containers and Images in Docker
3. Requirements File
4. Run the App

2.1.4 Specific procedures to be applied

1. Survey the target hosts

- Check what system software is to be installed on your hosts or target hosts.
- Verify that the host is ready for installation: system requirements, access privileges, memory and disk requirements, and so on.

2. Develop your installation sequence

- Review your deployment architecture and implementation specification.
- Identify particular situations that affect how you plan to install the required software for your deployment.
- Identify the interdependencies among the components you plan to install.
- Choose whether you will configure during installation or after the installation.
- Decide on the number and order of your installation sessions.

3. Gather configuration data

- Review the Administrator Server settings.
- Identify the configuration data that is required by the installer for the Configure Automatically During Installation option.

2.1.5 Who is responsible for carrying out the review activity?

The person or persons responsible for carrying out the review of the installation will be the technical engineer who has knowledge of the programs necessary to run the software. It will be under his responsibility to run the application on any device, find errors, if any, and fix them.

The person in charge of reviewing the *Installation Plan Review* will use the *Installation Plan Review template* to do the review.

Type of review activity	Date start-end	Pasat	Comment	Responsible
Install Docker	01/04/2021 - 05/04/2021	YES	There has been no problem in the installation and also in the docker configuration	Jordi Lazo Florensa
Containers and Images in Docker	01/04/2021 - 07/04/2021	YES	There hasn't been any problems with the containers or images from Docker	Roger Larriba
Requirements File	07/04/2021 - 08/04/2021	YES	All requirements needed are in the requirements.txt file located in the root folder of the project	José Ramón Noguero
Run the App	07/04/2021 - 08/04/2021	YES	The system runs correctly without any problems related to docker or other	Tiberiu Paiu

Table 3. *Installation Plan Review template*

2.2 Database Design Review

2.2.1 Scope

Design reviews are an important facet of the system development lifecycle for database applications. It is during the design review that all aspects of the database and application code are reviewed for efficiency, effectiveness, and accuracy. It is imperative that all database applications, regardless of their size, are reviewed to assure that the application was designed properly, efficient coding techniques were used, and the database is accessed and modified correctly and efficiently.

2.2.2 Type of review

- **Conceptual Design Review** - to validate the concept of the data and proposed application;
- **Logical Design Review** - a thorough review of all data elements, descriptions, and relationships, as well as comparison to and possibly remediation of the corporate data model;
- **Physical Design Review** - the database is reviewed in detail to ensure that all of the proper database parameter settings and other physical design choices were made, that a proper translation from logical model to physical database was made and that all denormalization decisions were formally documented;
- **Pre-Implementation Design Review** - an overall appraisal of the system components prior to implementation;
- **Post-Implementation Design Review** - formally review the application and database once it has run in production for a while to determine if the application is meeting its objectives.

2.2.3 Schedule

	Submission for review
T1	Distribution to other contractor reviewers Individual Review Collection of the other's contractors comments Delivery of the Comments Database to the author
T2	Review Meeting Communication of Review Meeting Decisions Implementation of Review Meeting Decisions Submission for Acceptance
T3	Verification of amendments Decision to accept the amendments

Table 4. *Submission for review*

2.2.4 Specific procedures to be applied

- A validation of the intent and purpose of the application
- An assessment of the logical data model
- An assessment of the physical data model
- A review and analysis of the physical DBMS parameters
- A prediction of the database performance
- An analysis of overall performance after production implementation
- Do not use spaces in the name of database objects or models
- Do not use SQL keywords as the name of database objects or models

2.2.5 Who is responsible for carrying out the review activity?

The person or persons responsible for carrying out the Database Design review will be the technical engineer who has knowledge on SQL and noSQL databases as well as database management. It will be under your responsibility to ensure the correct functioning of the database and the conservation of these.

Type of review activity	Date start-end	Passed	Type of error Description of the error	Responsible
Conceptual Design Review	dd/mm/yyyy - dd/mm/yyyy			
Logical Design Review				
Physical Design Review				
Pre-Implementation Design Review				
Post-Implementation Design Review				

Table 5. *Database Design Review template*

2.3 Test Plan Review

2.3.1 Scope

A Test Plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product. Test Plan helps us determine the effort needed to validate the quality of the application under test. The test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager.

2.3.2 Type of testing

These are the types of tests that will be implemented to verify that the software works correctly:

	Description
Unit Test	Test the smallest piece of verifiable software in the application.
API Testing	Test the API's created for the application.
Intregration Test	Individual software modules are combined and tested as a group.
System Test	Conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
Install/uninstall Testing	Focuses on what customers will need to do to install/uninstall and set up/remove the new software successfully.
Agile Testing	Testing the system using Agile methodology.
Custom / on Field Testing	Testing the product on their deployment itself.

Table 5. *Types of testing*

2.3.3 Procedures to be applied

- Analyze the product
  - Who will use the website?
  - What is it used for?
  - How will it work?
  - What are software/ hardware the product uses?
- Design the Test Strategy
  - The project's testing objectives and the means to achieve them
  - Determines testing effort and costs
- Define the Test Objectives
  - List all the software features (functionality, performance, GUI…) which may need to test.
  - Define the target or the goal of the test based on above features
- Define Test Criteria

    - Specify the critical suspension criteria for a test. If the suspension criteria are met during testing, the active test cycle will be suspended until the criteria are resolved.
    - Specify the criteria that denote a successful completion of a test phase. The exit criteria are the targeted results of the test and are necessary before proceeding to the next phase of development.
- Resource Planning
    - Detailed summary of all types of resources required to complete project task
- Plan Test Environment
    - Setup of software and hardware on which the testing team is going to execute test cases

2.3.4 Who is responsible for carrying out the test plan review?

The person or persons responsible for carrying out the Database Design review will be the technical engineer who has knowledge on SQL and noSQL databases as well as database management. It will be under your responsibility to ensure the correct functioning of the database and the conservation of these.

The person in charge of reviewing the *Test Plan Review* will use the *Test Plan Review template* to do the review.

Type of review activity	Date start-end	Procedure	Comment	Responsible
Unit Test	dd/mm/yyyy - dd/mm/yyyy	Analyze the product		José Ramón Noguero de Llano Xavier Loncà Sánchez
		Design the Test Strategy		
		Define the Test Objectives		
		Define Test Criteria		
		Resource Planning		
		Plan Test Environment		
API Testing	dd/mm/yyyy - dd/mm/yyyy	Analyze the product		José Ramón Noguero de Llano Xavier Loncà Sánchez
		Design the Test Strategy		
		Define the Test Objectives		
		Define Test Criteria		
		Resource Planning		
		Plan Test Environment		
Intregration Test	dd/mm/yyyy -	Analyze the product		José Ramón Noguero de Llano Xavier Loncà
		Design the Test		

	dd/mm/yyyy	Strategy		Sánchez
		Define the Test Objectives		
		Define Test Criteria		
		Resource Planning		
		Plan Test Environment		
System Test	dd/mm/yyyy - dd/mm/yyyy	Analyze the product		José Ramón Noguero de Llano Xavier Loncà Sánchez
		Design the Test Strategy		
		Define the Test Objectives		
		Define Test Criteria		
		Resource Planning		
		Plan Test Environment		
Install/uninstall Testing	dd/mm/yyyy - dd/mm/yyyy	Analyze the product		José Ramón Noguero de Llano Xavier Loncà Sánchez
		Design the Test Strategy		
		Define the Test Objectives		
		Define Test Criteria		
		Resource Planning		
		Plan Test Environment		
Agile Testing	dd/mm/yyyy - dd/mm/yyyy	Analyze the product		José Ramón Noguero de Llano Xavier Loncà Sánchez
		Design the Test Strategy		
		Define the Test Objectives		
		Define Test Criteria		
		Resource Planning		
		Plan Test Environment		
Custom / on Field Testing	dd/mm/yyyy - dd/mm/yyyy			José Ramón Noguero de Llano Xavier Loncà Sánchez

Table 6. *Test Plan Review template*

2.4 Code Review

The person in charge of reviewing the *Code Review* will use the *Code Review template* to do the review.

Type of review activity	Files	Date start-end	Description of the error	Type of error	Responsible
Bloaters	url.py	dd/mm/yyyy - dd/mm/yyyy	The file contains many lines of code. (Sergi)	Large Class	Software unwrapper (Sergi and Tiberiu)
	model.py	dd/mm/yyyy - dd/mm/yyyy	A class contains many fields/methods/lines o f code. (Sergi)	Large Class	
	...				
	view.py	dd/mm/yyyy - dd/mm/yyyy	**Line 35 - 102** A method contains too many lines of code. Generally, any metho d longer than ten lines should make you star t asking questions. (Sergi)	Long Method	
			Line 185 - 350 A method contains too many lines of code. Generally, any metho d longer than ten lines should make you star t asking questions. (Tiberiu)	Long Method	
Object - Orientation Abusers	url.py				
	model.py				
	...				
	view.py				
Change Preventers	url.py				
	model.py				
	...				
	view.py				
Dispensables	url.py				
	model.py				

	...				
	view.py				
Couplers	url.py				
	model.py				
	...				
	view.py				

Table 7. *Code Review template*

2.4.1 Scope

Code review is a software quality assurance activity in which one or several humans check a program mainly by viewing and reading parts of its source code, and they do so after implementation or as an interruption of implementation.

2.4.2 Type of testing

Lots of bugs in software development are found in the code. To avoid them, the Code Smell techniques seen in the Software Engineering course have been applied.

- **Bloaters:** Bloaters are code, methods and classes that have increased to such gargantuan proportions that they're hard to work with. Usually these smells don't crop up right away, rather they accumulate over time as the program evolves (and especially when nobody makes an effort to eradicate them).
- **Object-Orientation Abusers:** All these smells are incomplete or incorrect application of object-oriented programming principles.
- **Change Preventers:** These smells mean that if you need to change something in one place in your code, you have to make many changes in other places too. Program development becomes much more complicated and expensive as a result
- **Dispensables:** A dispensable is something pointless and unneeded whose absence would make the code cleaner, more efficient and easier to understand.
- **Couplers**: All the smells in this group contribute to excessive coupling between classes or show what happens if coupling is replaced by excessive delegation.

2.4.3 Procedures to be applied

- **Better code quality**: improve internal code quality and maintainability (readability, uniformity, understandability, ...)
- **Finding defects**: improve quality regarding external aspects, especially correctness, but also find performance problems, security vulnerabilities, injected malware, ...
- **Learning/Knowledge transfer**: help in transferring knowledge about the codebase, solution approaches, expectations regarding quality, etc; both to the reviewers as well as to the author.
- **Increase sense of mutual responsibility**: increase a sense of collective code ownership and solidarity.

- **Finding better solutions**: generate ideas for new and better solutions and ideas that transcend the specific code at hand.
- **Complying to QA guidelines**: Code reviews are mandatory in some contexts, e.g., air traffic software.

2.4.4 Who is responsible for carrying out the test plan review?

To review the code it will be necessary at least groups of two people who will check the code and between them two will look for errors or possible improvements. These two people will be programmers who have already worked on the software.

2.5 Document Review

Type of review activity	Files	Date start-end	Comment	Responsible
Syntactics	Document _01.docx	dd/mm/yyyy - dd/mm/yyyy	I have corrected some spelling errors (José Ramón)	José Ramón
	...			
	Document _x.docx		I have corrected some spelling errors (José Ramón)	
Format	Document _01.docx	dd/mm/yyyy - dd/mm/yyyy		
	...			
	Document _x.docx	dd/mm/yyyy - dd/mm/yyyy		

Table 8. *Document Review template*

2.5.1 Scope

A large number of errors in software development are found in the part of the documentation that is made or due to its non-existence. To avoid this, a document review has been created to check for syntactic errors or document format errors.

2.5.2 Type of testing

- **To gather background information**: reviewing existing documents helps you understand the history, philosophy, and operation of the program you are evaluating and the organization in which it operates.

- **To determine if implementation of the program reflects program plans:** the review of program documents may reveal a difference between formal statements of program purpose and the actual program implementation. It is important to determine if such a difference exists and to clarify the program intent before moving forward with the evaluation.
- **When you need information to help you develop other data collection tools for evaluation:** Reviewing existing documents to better understand the program and organization you are evaluating will help you formulate questions for interviews, questionnaires, or focus groups or develop an observation guide.
- **When you need data to answer what and how many evaluation questions:** Reviewing program documents is useful for answering basic evaluation questions related to the number and type of participants, number and type of program personnel, and program costs.

2.5.3 Procedures to be applied

1. **Is “native review” or “image review” being performed?**
2. **What is the overall goal or objective for this review?**
3. **How many reviewers will there be and what level of review needs to be performed?**
4. **What specific coding information are reviewers required to capture during review?**
5. **Do you need custom workflows or database administration tasks to be performed?**
6. **Will external or non-party reviewers need to access the review database?**
7. **Will advanced technologies be used to expedite document review?**
8. **How many documents need to be reviewed?**
9. **Have you budgeted time to accommodate for unforeseen variables?**
10. **Implement and fill the template provided**

2.5.4 Who is responsible for carrying out the test plan review?

To review the documentation, it will be necessary for the project manager of the group of software developers to review and read it again to improve and update it depending on how the software develops and the objectives of the project change.

2.6 Definition of Requirements Review

Scopet

Great quantity of errors in the development of software is troben in the part
The requirements, the more it increases the temos, the better it will be the cost of repairing error. For this reason, the create he document section using the quality Gateway criteria to try to minimize the errors.

Type of testing

Within Scope? The functionality of this text is to check if the requirement falls within the scope of the project.
Relevancy To test a requirement for its relevance, its intent must be compared to the purpose of the project.
Testing Completeness This test aims to check if there are any unspecified fields in the requirements.
Testing the fit criterion To overcome the ambiguity of a requirement, we add a Fit Criterion to measure the requirement and make it more accurate and testable.
Consistent terminology This section tests that all requirements are correctly worded and be understood by the people who read them.
Viable within constraints? Its functionality is to check that the requirements are compatible with
the restrictions.
Requirement or solution? Indicates the value that the requirement has for the user
Gold Plating Its purpose is to test that the requirements do not add functionality unnecessary or contribute unnecessarily to increase the cost.
Requirements creep It refers to the process of adding new requirements after considering them complete the current ones.

Procedures to be applied

For the Software Requirements Specification activity, what we will do is make the Quality Gateway for each of the requirements. If there is a requirement that does not comply with the activities of the Quality Gateway, indicate in the comments column and specify who does it.
The Responsable's name column is made up of Requirements Engineers, analysts and the Project Manager.
Specific procedures to be applied to the Quality Gateway:
Within Scope?
The functionality of this text is to check if the requirement falls within the scope of the project.
Relevancy
To test a requirement for its relevance, its intent must be compared to the purpose of the project.
Testing Completeness
This test aims to check if there are any unspecified fields in the requirements.
Testing the fit criterion
To overcome the ambiguity of a requirement, we add a Fit Criterion to measure the requirement and make it more accurate and testable.
Consistent terminology
This section tests that all requirements are correctly worded and be understood by the people who read them.
Viable within constraints?
Its functionality is to check that the requirements are compatible with
the restrictions.
Requirement or solution?

Indicates the value that the requirement has for the user

Gold Plating

Its purpose is to test that the requirements do not add functionality unnecessary or contribute unnecessarily to increase the cost.

Requirements creep

It refers to the process of adding new requirements after considering them complete the current ones.

The person in charge of reviewing the *Preliminary Design Review* will use the *Quality_Plan_Test template* and *Preliminary_Design_Review_checkilist* to do the review.

Type of review activity	Requirements	Date start-end	Preference	Comment	Responsible
Within Scope?	User story: 1.0	dd/mm/yyy - dd/mm/yyyy	First	I have detected that the requirement that belongs to the User Story 2.0 is not in the scope of the project (José Ramón)	Requirements engineer (José Ramón) Software analyst (Jordi Lazo)
	User story: 2.0		Second		
	...	...	...		
	User story: x.x	dd/mm/yyyy - dd/mm/yyyy	Latest		
Relevancy	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				
Testing completenss	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				
Testing the fit criterion	User story: 1.0				

	User story: 2.0				
	...	...	...		
	User story: x.x				
Consistent terminology	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				
Viable within constraints?	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				
Requirement or solution?	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				
Requirement Value	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				
Gold Plating	User story: 1.0				
	User story: 2.0				
	...	...	...		

	User story: x.x				
Requirements creep	User story: 1.0				
	User story: 2.0				
	...	...	...		
	User story: x.x				

Table 9. *Code Review template*

2.7 Preliminary Design Review

2.7.1 Scope

This review has the objective of determining that the baseline of the product(hardware, software,etc) has a reasonable expectation of satisfying the requirements within the current schedule and budget.

2.7.2 Type of review

The PDR reviews the hardware,software, systems and architecture you will be working with.

2.7.3 Schedule

This has to be tested as soon as the baseline has been created.

2.7.4 Checklist of questions to be asked in the review

1. Does the status of the technical effort and design indicate operational test and evaluation success (operationally effective and suitable)?
2. Can the preliminary design, as disclosed, satisfy the draft Capability Development Document?
3. Has the system allocated baseline been established and documented to enable detailed design to proceed with proper configuration management?
4. Are adequate processes and metrics in place for the program to succeed?
5. Have sustainment and human systems integration design factors be
6. Are the risks known and manageable for integrated testing and developmental and operational evaluation?
7. Is the program schedule executable (technical/cost risks)?
8. Is the program properly staffed? Has the program's cost estimate been updated?

9. Is the program executable within the existing budget and with the approved system allocated baseline?
10. Is the preliminary system level design producible within the production budget?
11. Is the updated CARD consistent with the approved allocated baseline?
12. Has the Computer system and software architecture design been established, and have all Computer Software Configuration Items (CSCIs), Computer Software Components (CSCs), and Computer Software Units (CSUs) been defined?
13. Are Software Requirements Specifications (SRSs) and Interface Requirement Specifications (IRSs), including verification plans, complete and baselines for all CSCs and do they satisfy the system/subsystem functional requirements?
14. Do the Interface Control Documents (ICDs) trace all software interface requirements to the CSCIs and CSUs?
15. Has the computer system and software design/development approach been confirmed through analyses, demonstrations, and prototyping in a relevant environment?
16. Has the preliminary software design been defined and documented? Have software increments been defined and have capabilities been allocated to specific increments?
17. Have software trade studies addressing COTS, reuse, and other software-related issues been completed?
18. Has the software development process been defined in a baselined Software Development Plan (SDP) and is if reflected in the Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)?
19. Do the software development schedules reflect contractor software processes and IMP/IMS software events for current and future development phases?
20. Have the software development environment and test/integration labs been established with sufficient fidelity and capacity?
21. Have unique software risks have been identified and assessed and have mitigation plans been developed and implemented?
22. Have software metrics been defined and reporting process implemented, and are they being actively tracked and assessed?
23. Does the Test and Evaluation Master Plan (TEMP) address all CSCI plans, test facilities, and test plans, including testing required to support incremental approaches (e.g. regression tests)?
24. Is there a life-cycle sustainment plan and does it include software support requirements?
25. Have the software development estimates (i.e. size, effort (cost), and schedule) been updated?
26. Have all required software-related documents been baselined and delivered?

The person in charge of reviewing the *Preliminary Design Review* will use the *Quality_Plan_Test template* and *Preliminary_Design_Review_checkilist* to do the review.

Type of review activity	Yes/No	Responsible
Does the status of the technical effort and design indicate operational test and evaluation success (operationally effective and suitable)?	-	Jordi Rafael Lazo Florensa Tiberiu Paiu
Can the preliminary design, as disclosed, satisfy the draft Capability Development Document?		José Ramón Noguero de Llano Xavier Loncà Sánchez
Has the system allocated baseline been established and documented to enable detailed design to proceed with proper configuration management?		Roger Larriba Baudilio Sergi Puigpinós Palau
Are adequate processes and metrics in place for the program to succeed?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Have sustainment and human systems integration design factors be		Roger Larriba Baudilio Sergi Puigpinós Palau
Are the risks known and manageable for integrated testing and developmental and operational evaluation?		José Ramón Noguero de Llano Xavier Loncà Sánchez
Is the program schedule executable (technical/cost risks)?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Is the program properly staffed? Has the program's cost estimate been updated?		Roger Larriba Baudilio Sergi Puigpinós Palau
Is the program executable within the existing budget and with the approved system allocated baseline?		Jordi Rafael Lazo Florensa Tiberiu Paiu

Is the preliminary system level design producible within the production budget?		Roger Larriba Baudilio Sergi Puigpinós Palau
Is the updated CARD consistent with the approved allocated baseline?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Has the Computer system and software architecture design been established, and have all Computer Software Configuration Items (CSCIs), Computer Software Components (CSCs), and Computer Software Units (CSUs) been defined?		Roger Larriba Baudilio Sergi Puigpinós Palau
Are Software Requirements Specifications (SRSs) and Interface Requirement Specifications (IRSs), including verification plans, complete and baselines for all CSCs and do they satisfy the system/subsystem functional requirements?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Do the Interface Control Documents (ICDs) trace all software interface requirements to the CSCIs and CSUs?		Roger Larriba Baudilio Sergi Puigpinós Palau
Has the computer system and software design/development approach been confirmed through analyses, demonstrations, and prototyping in a relevant environment?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Has the preliminary software design been defined and documented? Have software increments been defined and have capabilities been allocated to specific increments?		Roger Larriba Baudilio Sergi Puigpinós Palau

Have software trade studies addressing COTS, reuse, and other software-related issues been completed?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Has the software development process been defined in a baselined Software Development Plan (SDP) and is if reflected in the Integrated Master Plan (IMP) and Integrated Master Schedule (IMS)?		Roger Larriba Baudilio Sergi Puigpinós Palau
Has the preliminary software design been defined and documented? Have software increments been defined and have capabilities been allocated to specific increments?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Have unique software risks have been identified and assessed and have mitigation plans been developed and implemented?		Roger Larriba Baudilio Sergi Puigpinós Palau
Have software metrics been defined and reporting process implemented, and are they being actively tracked and assessed?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Does the Test and Evaluation Master Plan (TEMP) address all CSCI plans, test facilities, and test plans, including testing required to support incremental approaches (e.g. regression tests)?		Roger Larriba Baudilio Sergi Puigpinós Palau
Is there a life-cycle sustainment plan and does it include software support requirements?		Jordi Rafael Lazo Florensa Tiberiu Paiu
Have the software development estimates (i.e. size, effort (cost), and schedule) been updated?		Roger Larriba Baudilio Sergi Puigpinós Palau

Have all required software-related documents been baselined and delivered?		Roger Larriba Baudilio Sergi Puigpinós Palau

Table 10. *Preliminary Design Review*

2.8 Test Readiness Review

2.8.1 Scope

Test Readiness Review is generally created and the review is performed by the QA team representative. The results are shared with the PMs and the other team members to signify whether the test team is ready or not to move into the test execution phase.

2.8.2 Specific procedures to be applied

Software must pass the following tests:

- All the Requirements finalized and analyzed
- Test Plan created and reviewed
- Test Cases preparation done
- Test Case review and sign off
- Test Data availability
- Smoke Testing
- Is Sanity Testing done?
- Team aware of the roles and responsibilities
- Team aware of the deliverables expected of them
- Team aware of the Communication protocol
- Team's access to the application, version control tools, Test Management
- Team's trained
- Technical Aspects- Server1 refreshed or not?
- Defect reporting standards are defined

Type of review activity	Yes/No	Responsible
All the Requirements finalized and analyzed	YES	Roger Larriba Baudilio

Test Plan created and reviewed	NO	Xavier Loncà Sánchez
Test Cases preparation done	NO	Roger Larriba Baudilio
Test Case review and sign off	NO	Xavier Loncà Sánchez
Test Data availability	NO	Roger Larriba Baudilio
Smoke Testing	NO	Xavier Loncà Sánchez
Is Sanity Testing done?	NO	Roger Larriba Baudilio
Team aware of the roles and responsibilities	YES	Xavier Loncà Sánchez
Team aware of the deliverables expected of them	YES	Roger Larriba Baudilio
Team aware of the Communication protocol	YES	Xavier Loncà Sánchez
Team's access to the application, version control tools, Test Management	YES	Roger Larriba Baudilio
Team's trained	NO	Xavier Loncà Sánchez

Technical Aspects- Server1 refreshed or not?	-	Roger Larriba Baudilio
Defect reporting standards are defined	YES	Xavier Loncà Sánchez

Table 11. *Test Readiness Review*

3- Software tests

1. Information			
#Test	**Test Description**	**Test date**	**Responsible**
1	As a user, I would like to be able to access the system	dd/mm/yyyy	
2	As a user, I would like the system to recognize me and take me to the appropriate page	dd/mm/yyyy	
3	As a writer, I would like to be able to present the book in an automated and computerized way.	dd/mm/yyyy	
4	As a main editor, I would like to have a list of all the books submitted by the writers to the publisher	dd/mm/yyyy	
5	As the main editor, I would like the system to display a page with all the information related to the book and all the actions I can take with it.	dd/mm/yyyy	
6	As an editor I would like to have a list where I can find all the books I have assigned.	dd/mm/yyyy	
7	As an editor I would like to be able to make requests for images in a computerized way and that they are registered	dd/mm/yyyy	
8	As the main layout designer I would like the system to show me	dd/mm/yyyy	

	all the images requests that have been requested by the editors.		

Table 12. *Software Test - Information*

2. Apply				
#Test	**Expected results**	**Actual results**	**Comment**	**Pass/fail**
1	An error message is displayed if the user does not enter name or password. An error message is displayed in case you enter incorrectly enter name or password. An error message is displayed in case who keywords like DELETE in the fields of the name or password.			
2	That the system takes the user to the main page corresponding to their role. That access to publishers in parts of the system restricted to the main publisher is not allowed. Writers should not be allowed access to parts of the system that are restricted to the main publisher or publisher			
3	When the book has been submitted to the publisher, a new object is created in the list in the "books presented in the publisher" section of the main publisher. When the book has been submitted to the publisher, a new object is created in the list in the "presented books" section of the writer. When the book has been submitted to the publisher, it has the status of "Presented"			
4	All the books with their respective data appear in the list. That the books listed be sorted by arrival.			
5	That all the information of the book in question is displayed correctly. That all the buttons lead to the correct functionality.			

	That the "download the book" button downloads the corresponding book correctly in PDF format and the selected version from the list of all versions of the book.			
6	All the books assigned to the editor appear in a list. Books have to be sorted by arrival order.			
7	When a request is made, it is stored correctly in the database. The request cannot contain a body and an empty title. When submitting a request, the user must be redirected to the "requested images" page. A confirmation message is displayed if the request could be sent An error message is displayed if the request could not be sent. When a request is made, it must be addressed to the main layout designer			
8	When a request is made, it is stored correctly in the database. The request cannot contain a body and an empty title When submitting a request, the user must be redirected to the "requested images" page. A confirmation message			

Table 13. *Software Test - Apply*

3. Document settings			
Role	**Name**	**Signature**	**Date**
Service Owner	Bookiernes, S.A.		dd/mm/yyyy
Developers	Jordi Rafael Lazo Florensa Tiberiu Paiu Roger Larriba Baudilio Sergi Puigpinós Palau		dd/mm/yyyy

	José Ramón Noguero de Llano Xavier Loncà Sánchez Senadin Puce		

Table 14. *Software Test - Document settings*

4- Acceptance tests for sw externally developed?

Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

- Error messages are displayed correctly and accordingly
- When an user logins inside, this is redirected accordingly
- Book's status changes correctly
- All lists display the correct information
- Every detail page display the correct information and functionalities
- Every button does the correct functionality
- Book's can be assigned correctly
- Chat messages are send correctly
- Notifications display the message correctly
- Every petition done inside the system it's correctly handled and sent to the corresponding user
- Pages display the petition's information correctly
- System handles user files correctly
- Books are published correctly to the webpage
- Books can be visualised correctly on the webpage
- Books are translated correctly to the languages desired by the editors
- System can create and delete users correctly

1. Scope (In Scope – Out of Scope)	
In scope	**Out of scope**
In Scope (*List features that are tested*).	Out of Scope (*List features that are not tested*).

Table 15. *Acceptance Test - Scope*

2. Assumptions and Constraints	
Assumptions	**Constraints**
Assumption (*List the UAT assumptions*).	Constraint (*List the UAT constraints*).

Table 16. *Acceptance Test - Assumptions and Constraints*

3. Risks					
ID	**Risk**				
1	"The system loses connection to the database"				
	Risk likelihood				**Exploring countermeasures**
	Consequences	**Likely**	**Possible**	**Unlikely**	You will have one or more extra databases on other servers. Which will be updated periodically. Use a database on the same site as the web server or improve the connection between the bbdd and the server.
	The data stored in the database cannot be inserted, modified or deleted	High	Moderate	Low	
	Useful error messages are shown to the attackers.	Moderate	Moderate	Low	
	Only part of the data can be saved.	High	Moderate	Low	
	Workers can't do their jobs and book publishing stops	Severe	Moderate	Moderate	

Table 17. *Acceptance Test - Risks*

4. Team Roles & Responsibilities		
Name	**Roles**	**Responsabilities**
Name (*List names of people involved in testing*).		
Name (*Add more rows if needed*).		

Table 18. *Acceptance Test - Team Roles & Responsabilities*

5. Entry criteria	
ID	**Criteria**
5.1	Entry Criteria (*Factors that must be present to enable the start of the UAT. Example: Testing environment/ data is available*).

Table 19. *Acceptance Test - Entry criteria*

6. Requirements-Based Test Cases	
ID	**Test Cases**
6.1	Test Case (*Identify the test cases along with the expected results*). Example: Test Procedure: Login with a corporate user account. Username: abc Password: abc Expected Results: An error will be displayed for the wrong credentials.

Table 20. *Acceptance Test - Requeriments-Based Test Cases*

–

7. Test Results				
ID	**Test Cases**	**Pass/Fail**	**Tested By**	**Date Tested**
7.1	Test Case *(Name the test case)*. Example: Test Procedure: Login with a corporate user account. Username: abc Password: abc Expected Results: An error will be displayed for the wrong credentials.			dd/mm/yyyy
	Test Case *(Add more rows if needed)*.			

Table 21. *Acceptance Test - Test Results*

8. Document Signatures			
Role	**Name**	**Signature**	**Date**
Service Owner	Bookiernes, S.A.		dd/mm/yyyy
Developers	Jordi Rafael Lazo Florensa Tiberiu Paiu Roger Larriba Baudilio Sergi Puigpinós Palau José Ramón Noguero de Llano		

	Xavier Loncà Sánchez Senadin Puce		
Others (if needed)	*Add more rows if needed.*		

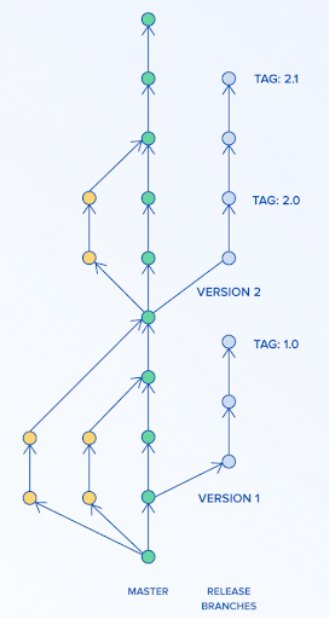
Table 22. *Acceptance Test - Document Signatures*

5- Configuration management

5.1 Software change control procedures

Branches (Github)

To control the different versions of the project's app, the team has to use a version control such as Github. The 'Master' branch is used as the primary branch where all merges happen and all the secondary or feature branches begin. For each member of the team or for each functionality that needs to be implemented, a new branch will need to be created. When there's a version of the application stable enough to publish or present to someone, this is going to be put on the release branch.



Scrum Process Canvas(Zenhub)

In order to keep track of what the team's members're doing and what they've to do, the team uses Zenhub since it facilitates the project management tasks and fits in their iterative/agile development model. And with it, they track the changes that the software is having at each moment.

5.2 Dates for release of versions

In each sprint or delivery, the team has been developing and implementing the various planned functionalities. When all the functionalities are achieved or the finish date in each sprint, the team has to prepare a release version for various reasons. Some of these reasons are for being able to make a review of the sprint and actualize the project/product backlog.

Files	Date	Github branch link	Comment
Version 1	*dd/mm/yyyy*	https://github.com/sergiPuigpinosPalau/BookiernesSystem/tree/BookLists_branch	
Version 2	*dd/mm/yyyy*	https://github.com/sergiPuigpinosPalau/BookiernesSystem/tree/BookLists_branch	

Table 23. *Dates of release of versions*

5.3 Documentation control

To ensure the correct maintainability, creation, revision and disposition of the documents related to the project, the team has to implement a series of good practices in order to achieve them. The standard ISO 9001:2015 requires to control the following aspects:

Identification	Storage
Protection	Retrieval
Retention	Review
Approval	Disposition
Legibility	Change tracking

Even if the team doesn't provide all of these, it would be a great point to have as a reference. The developers will have to specify in which way they are gonna obtain them, mainly through which software they are gonna use.

File name	Last revision date	Link file (drive)	Comment
Document_01.docx	dd/mm/yyyy	Document_01.docx	

Table 24. *Documentation control*

In case a new integrate or member joins the group, to know the paths of all the different files in the project, it needs to have a Google Drive account and request access from one of the team members. When one of them gives it permission to access and read the files in the shared folder, it can find the file it's searching for by searching on the corresponding subject folder.

5.4 Communication with the client

The team has to define which type of communication they're gonna maintain with the client. Show in which form / elicitation techniques that are gonna be used and how periodically the client is gonna be informed with the changes or having meetings with the development team.

Type	Date	Contents	Stakeholders	Responsible	Comment
Interviews	dd/mm/yyyy	Link interviews https://www.youtube.com/watch?v=eZvKaP18zXI			
Mail	dd/mm/yyyy	Dear Mrs. ... See you soon,			
From	dd/mm/yyyy	Link from forms.gle/MVmFHfh6sbZuuAFm8			

Table 25. *Communication with the client*

6- Costs

Here the team or the responsible for doing this task should put a detailed list of which expenses is gonna have to assume the company in order to develop the project. Between these expenses are: the estimated cost of developing each functionality (delimited in user stories), the employees' wage, the hardware/software that gonna require the new system, teaching new recruits or forming the actual members and the documentation that gonna use the team if acquired from third parties (templates for example).

User stories	Price	Priority	Responsible	Start date	Ending date	Duration	Percent complete
1.1	2300€	NORMAL	Tiberiu	dd/mm/yyyy	dd/mm/yyyy	8	100%
1.2	300.00€	EASY	Sergi	dd/mm/yyyy	dd/mm/yyyy	2	100%
2.1	5000.00€	HIGH	Jordi	dd/mm/yyyy	dd/mm/yyyy	7	100%
...	...	...	...	...	...	...	...
x.x	2500.00€	NORMAL	Roger	dd/mm/yyyy	dd/mm/yyyy	4	80%
Project Budget	Estimated:	Baseline:	Task Costs:	Estimated:		Baseline:	Actual:
	325.000,00€	300.000,00€		318.000,00€		300.000,00€	178.150,00€

Table 26. *Costs*