

Project Plan

CannonPrint Test Results Visualization Tool

Canon Production Printing

Date	:	06/03/2023
Version	:	0.1
State	:	Ongoing
Author	:	CodeCrusaders group

Version history

Version	Date	Author(s)	Changes	State
0.1	06/03/2023	CodeCrusaders	Basic document structure	Finished

Distribution

Version	Date	Receivers

Contents

1. Project assignment	4
1.1 Context	4
1.2 Goal of the project	4
1.3 Scope and preconditions	4
1.4 Strategy	4
1.5 Research questions	4
1.6 End products	4
2. Project Organisation	6
2.1 Stakeholders and team members	6
2.2 Communication	6
3. Activities and time plan	7
3.1 Phases of the project	7
3.2 Time plan and milestones	7
4. Testing strategy and configuration management	8
4.1 Testing strategy	8
4.2 Test environment and required resources	8
4.3 Configuration management	8
5. Finances and Risk	9
5.1 Project budget	9
5.2 Risk and mitigation	9

1. Project assignment

1.1 Context

Canon Production Printing (CPP), formerly known as **Océ** until the end of 2019, is a Netherlands-based subset of Canon that develops, manufactures and sells printing and copying hardware and related software.

Canon currently uses a lot of regression tests. These are automatic tests that run after every change to the codebase to ensure that the changes did not introduce unintended bugs. These include basic tests in which the printer is turned on virtually and more advanced tests which check if a printer can recover from specific error scenarios. These tests are done for multiple configurations of the printer; for example they test configurations with all hardware but also configurations in which there is no hardware available to print duplex sheets.

The current implementation of the regression dashboard that Canon uses its storage inefficiently and the interface is not user friendly and can be improved a lot.

1.2 Goal of the project

Our goal at CodeCrusaders is to improve the already existing test regression dashboard since the current implementation of it that Canon uses it's storage inefficient and the interface is not user friendly. We want to develop a stable and bug free solution that will improve the current working condition of Canon employees when dealing with unit tests and regression tests. They will be able to filter the tests and search for specific test results and even show similar tests with the same result. The data will be stored in a database and sensitive information will be protected.

1.3 Scope and preconditions

Inside scope:	Outside scope:
1 Documentation	1 Training manuals and instructions.
2 Software Solution	2 Hardware to run the application.
3 Test plan and test report	3 Infrastructure to set up the application.
4 User specific requirements	4 There will be no updates after the solution's final version has been agreed on and deployed.
5 UML diagram	

1.4 Strategy

After conducting research and analyzing the results, our team has decided to use the Agile SCRUM project management strategy. We took this decision because we want to stay flexible while making the most out of the time we have to make this project. We feel like the Agile way of development allows us to create a product that is as good of a representation of the clients needs as possible. This is achieved through the sprint setups. As at the end of each sprint, the client is presented with the product thus far, and as a result changes can be made dynamically and more efficiently.

Every sprint one of our members is the SCRUM master and the contact person. We will also have a sprint transcriber. The remaining 4 members are partakers.

1.5 Research questions and methodology

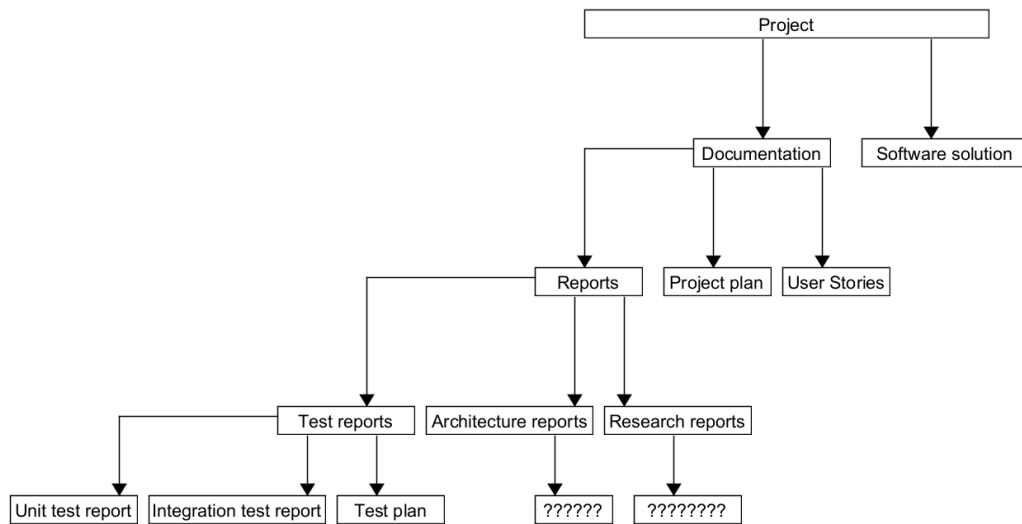
<<Describe the research questions that are most relevant to your project. For each research question, describe the approach and/or methodology. Use the Dot Framework to specify strategies and methods - see <http://www.ictresearchmethods.nl> for details.

Note that research is not only part of the initial phases (like analysis) of the project, but runs throughout the whole project. E.g., in the realization phases, you will probably do research in the Workshop and Lab context.

Realize that during the project your research questions may change, and that new ones will come up. That normal for any project, and is not a problem as long as you involve the right stakeholders, and keep your deliverables updated.

>>

1.6 End products



2. Project organization

2.1 Stakeholders and team members

Name	Abbreviation	Role and functions	Availability
Tom Buskens	T.B	Product Owner	Available through email Monday-Friday Every three weeks a meeting with the team.
Gupta Roopali	G.R	Project Coach	
Tim de Lange	T.L	Team member	423421@student.fontys.nl
Andrei ALEXandru	A.A	Team member	491992@student.fontys.nl
Radu Andrei Popescu	R.P	Team member	490388@student.fontys.nl
Adrian Vlad Pop	A.P.	Team member	493739@student.fontys.nl
Lê Anh Tuan	A.T	Team member	476005@student.fontys.nl
Munkh-Erdene Temuulen	M.T	Team member	

2.2 Communication

Our team has dedicated our online meetings to using Discord and Microsoft Teams. As for the in-person meetings, we use university rooms to come together. In each of the meetings we conduct and respectively each of our members gets to work on a problem, either individually or as a group.

Our group also takes notes of our meetings and separates them in their designated channels. We create meeting minute documents that are currently stored in a specific Discord server, later to be uploaded onto our GIT repository.

We do not take an attendance list, but we make sure that each of our team members are aware of the situation and work we have to do. We notify each other in case of absence. In case of an online meeting/campus, we make sure to get a specific time and location/platform for the meeting.

3. Activities and time plan

3.1 Phases of the project

In the first sprint we started analyzing the product and what is required of us. We started making plans for the project and drafting requirements and user stories. In addition, we have created an account on Jira because it is a good agile management tool to use and it helps us to show the backlog of the project and know the goals we need to achieve for each sprint. In addition to that, we chose the technologies for the project and started learning those technologies.

3.2 Time plan and milestones

This semester we have chosen to use Agile scrum methodology because it is focused on the client process. So, it makes sure that the client is continuously involved during every stage. In addition, the process is completely based on incremental progress. Therefore, the client and team know exactly what is complete and what is not. This reduces risk in the development process.

During this semester we divided the backlog and tasks of the project into sprints. The duration of each sprint is three weeks. During each sprint, the scrum process includes these three activities so that the team can communicate about the project.

1- Sprint planning: Before any work can begin, the Scrum team must meet to prioritize features for the product and create a product backlog of features. The product backlog is a list of tasks the team agrees to complete in an assigned sprint.

2- Sprint scrum meeting: A daily Scrum meeting is held to talk about the previous day's work, discuss hang-ups, and define what work will be completed that day. Each member of the team updates the group on what they've been working on and brings up any issues or questions.

3- Sprint review and retrospective:

During a sprint review, the Scrum master, product owner, development team, and stakeholders review what they have accomplished during the sprint compared to what they intended to accomplish. This meeting may include demonstrating the product for the customer or stakeholders. Any necessary changes are implemented. During a sprint retrospective meeting, the Scrum team takes a closer look at the sprint itself. What went well and what could be improved in the process, so the team can become more efficient and agile over time.

4. Testing strategy and configuration management

4.1 Testing strategy

Testing envision	Percentage code coverage	Available automation
Unit testing	30%	N/A
Acceptance criteria	N/A	Automation available
Component testing	N/A	N/A
Integration testing	N/a	N/A

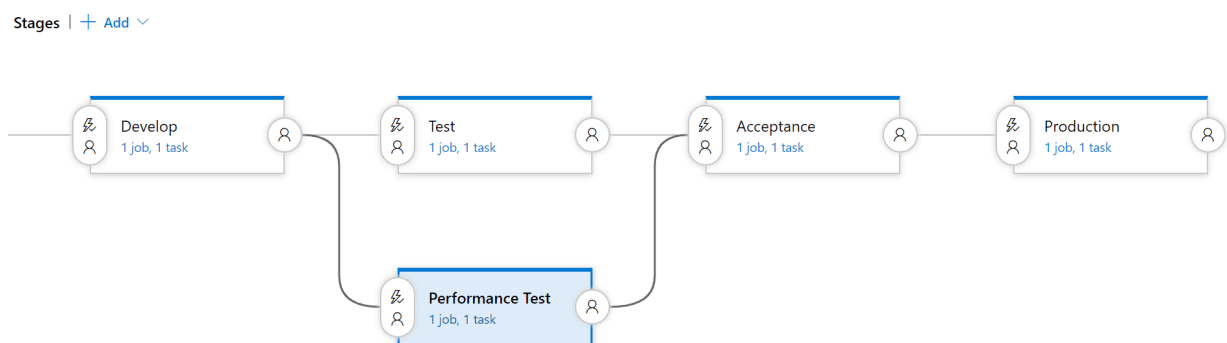
Our group has decided that the most relevant types of testing that must be looked upon are unit tests and acceptance criteria. We plan to expand onto component testing where we want to get an overview of the UI (user interface) workflow and functionality.

The goals for this project are to test the most relevant functionality and avoid redundancy. Our defined goals are to have proper component testing and show only the most relevant results.

4.2 Test environment and required resources

Our group has decided to use CI/CD environment as a test environment on Gitlab because this allows us to receive and respond to customer feedback much faster than before. Updates could be deployed and made live for the user in as little as just minutes after the developer has written the code.

We have used a DTAP(Development, Testing, Acceptance, Production) environment for the project. This picture describes the CI/CD environment of the project.

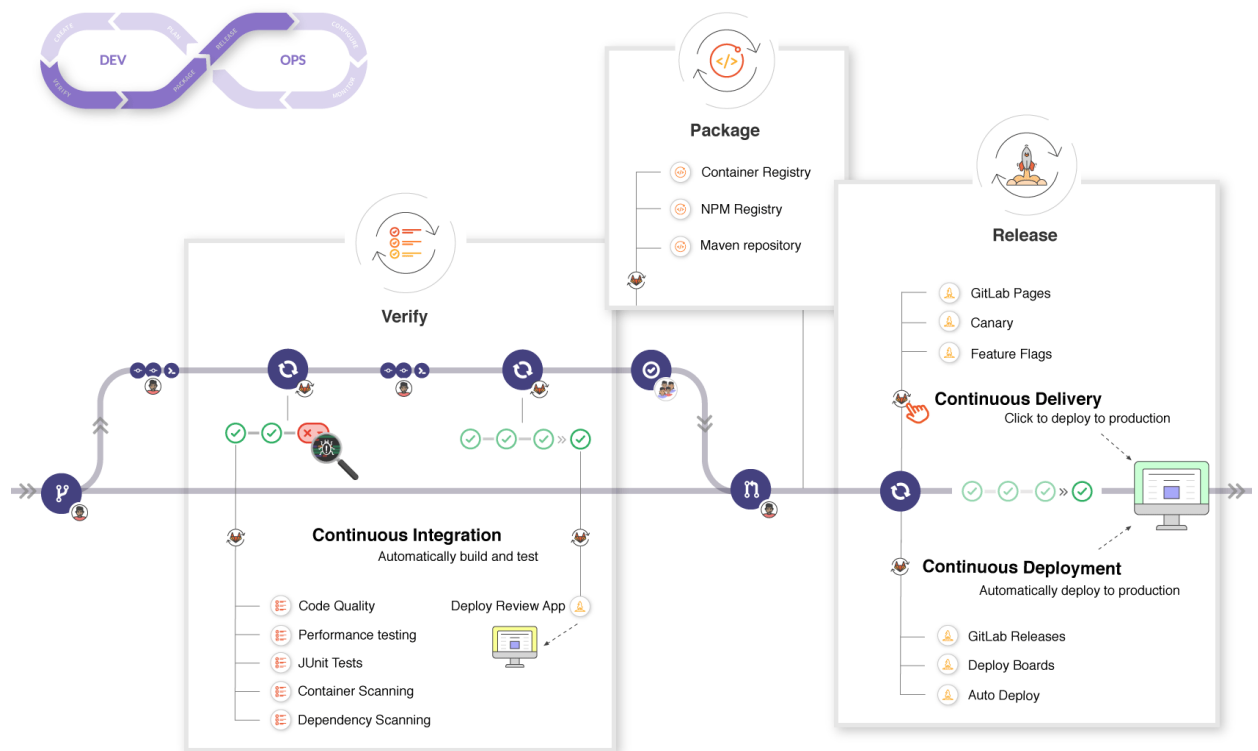


After researching the best way to test, it has been found that Unit-testing, integration test and acceptance test are the most relevant tests to use for the project to identify errors, gaps, or missing requirements of the project.

4.3 Configuration management

What is continuous integration?

GitLab CI (Continuous Integration) service is a part of GitLab that builds and tests the software whenever the developer pushes code to the application. GitLab CD (Continuous Deployment) is a software service that places the changes of every code in the production which results in every day deployment of production.



Why has our team decided to use GitLab for our project?

GitLab is a web-based DevOps lifecycle tool that provides a Git repository manager providing wiki, issue-tracking and continuous integration and deployment pipeline features, using an open-source license, developed by GitLab Inc.

We think GitLab is a professional environment and has many advantages that help with configuration and file management. We plan to keep everything up to date and provide the necessary commit labels for better storing.

5. Finances and risk

5.1 Risk and mitigation

Risk	Prevention activities	Mitigation activities	Probability
1 Scope Creep	Create a very clear project plan and scope	Focus on the hughes priority tasks and discuss with the client the important features	Medium
2 Team Burnout	Work in a controlled and scheduled manner	Take breaks when feeling too tired	High
3 Misaligned expectations	Get a clear view of what the client wants every time before starting a new sprint	Rediscuss requirements with the client and start the sprint again.	Low
4 Inadequate testing	Test the product after every feature has been finished	Fix the problem as soon as possible	Low
5 Lack of documentation	Make sure that documentation is up to date with the latest software version	Update the documentation as soon as possible	Low
6 Dependency management	Make sure that all technologies are compatible when starting to work on a project	Find the quickest way to get to a stack of compatible technologies for the project	Medium
7 Technical Debt	Make sure that thorough software development principles are applied when implementing your application	Analyze the code and fix the found flows in it. After this step make sure that all tests are passing and the solution is functioning as planned.	Low
8 Project track tutor/ client unavailable	Communicate often with the tutor so that there is time to prepare if this occurs	Find a replacement teacher/person with similar expertise to answer questions	Medium