|  |
| --- |
| Fontys University of Applied Sciences - ICT |
| Project Plan |
| Amplexor Group Project – Group 3 |

|  |
| --- |
| Group 3 (Wait 4 IT) – Release Version  21/01/2021 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Date* | *Version* | *Author* | *Comments* |
| 04-09-2020 | 0.1 | **Group** | Initial document structure and content |
| 03-01-2021 | 1.0 | **Martin Georgiev** | Complete document rewrite; New document format and structure |
| 21-01-2021 | 1.1 | **Martin Georgiev** | Minor document changes; Added a section to the business process explaining the team’s solution to the product owner’s initial problem. |

Document Version History

Table of contents

[1 Project Assignment 3](#_Toc62096607)

[1.1 Context 3](#_Toc62096608)

[1.2 Project Goal 3](#_Toc62096609)

[1.3 Business process workflow 4](#_Toc62096610)

[1.3.1 The team’s solution (End of project update) 5](#_Toc62096611)

[1.4 Scope and Preconditions 6](#_Toc62096612)

[1.5 Project Strategy 7](#_Toc62096613)

[1.6 Research Questions 7](#_Toc62096614)

[1.7 End Products 9](#_Toc62096615)

[2 Project Organisation 11](#_Toc62096616)

[2.1 Stakeholders and team members 11](#_Toc62096617)

[2.2 Communication 12](#_Toc62096618)

[2.3 Organisational Structures 12](#_Toc62096619)

[3 Activities and time plan 13](#_Toc62096620)

[3.1 Phases of the project 13](#_Toc62096621)

[3.2 Time plan and milestones 14](#_Toc62096622)

[4 Testing strategy and configuration management 15](#_Toc62096623)

[4.1 Testing strategy 15](#_Toc62096624)

[4.2 Testing environment and required resources 15](#_Toc62096625)

[4.3 Configuration management 16](#_Toc62096626)

[5 Finances and risk 17](#_Toc62096627)

[5.1 Risk and mitigation 17](#_Toc62096628)

# Project Assignment

## Context

The company Amplexor, one of the content service leaders for the past 30 years, approached our team over at Fontys University of Applied Sciences with a pitch to research and develop a program extension (plug-in) for SDL Trados Studio that aids with automated document translation for the European Parliament. The team (Group 3) is tasked to work together with the product owner (Amplexor) and find a suitable solution that meets their demands.

## Project Goal

The Amplexor Global Content Suite (GCS) team receives an average of 63 pages that need to be translated using the EUR-Lex (EP) and TRIS (DG) public databases. Manually, it takes an estimated time of 5.25 hours each day to search for all references from the abovementioned documents. That is why the company is looking for a software solution to automate the entire process. The final product will automate reference lookup, document retrieval using the aforementioned databases and create a translation memory from it for the application SDL Trados Studio (CAT Tool). The company’s goal is to save both time and money (through manual labour).

## Business process workflow

Currently, Amplexor’s process is as follows:

1. The company receives a document from the European Parliament.
2. The document is loaded into SDL Trados Studio.
3. Reference numbers are manually extracted from that document.
4. Each reference number is looked up in the EUR-Lex/TRIS database.
5. The last consolidated version of the needed document is taken from the database.
6. All document content is put in a separate file in the form of raw text back into SDL Trados Studio. (a translation memory)

As the existing business process requires extensive manual labour, the goal of the software solution is to automate it to save as much time as possible. Ideally, the process will look something like this using the end product’s implementation:

1. The company receives a document from the European Parliament.
2. The document is loaded into SDL Trados Studio.
3. The user clicks a button that automatically extracts all reference numbers from the document.
4. The system looks up all found reference numbers and retrieves the last consolidated version of the needed documents.
5. The system strips the document of its’ HTML structure and places the raw text into a separate file, ready for use in SDL Trados Studio. (a translation memory)

As seen from the aforementioned steps of the workflow, most of the steps are automated and handled by the software solution (as seen from steps 3 to 5 from the second numbered list). The end-user will only need to load the file into SDL Trados Studio and start the reference number retrieval process with the click of a button.

### The team’s solution (End of project update)

The team is happy to report that our projection for the business process workflow that our application plugin will create is more or less correct. In other words, almost all of the tasks are handled automatically by the software solution (the SDL Trados plugin) and very minimal manual labour is required by the end-user. As a result, the plugin saves hours of work on a daily basis, providing both financial and time-saving benefits to the product owners (Amplexor).

The software solution in question provides the following business process workflow:

1. The company receives a document from the European Parliament.
2. The document is loaded into SDL Trados Studio.
3. The user starts the translation memory creation process with the click of a button.
4. The system a reference numbers extraction process from the target file.
5. The system creates a translation memory based on the retrieved document data (their last consolidated version) from the EUR-Lex public database. That document information is processed and exported into a raw translation memory file, ready for use in SDL Trados Studio.

## Scope and Preconditions

It is important to determine what kind of preconditions exist before starting work on the project. The list below mentions all endpoints that belong to the project in the form of the internal and external project scope.

|  |  |
| --- | --- |
| **Internal scope:** | **External scope:** |
| 1. SDL Trados Integration | 1. Final Solution (Plug-in) |
| 1. Following an Agile approach | 1. Software source code |
| 1. Development in C# using the .NET Framework | 1. Technical documentation |
| 1. Azure development environment | 1. Help Integration in SDL Trados |

## Project Strategy

The project team plans to fully utilize Scrum (An agile methodology) and the Azure development environment in order to establish and standardize a stable communication and all project stakeholders. The project will be divided into a total of six sprints, starting from sprint 0, which will be used for the proper introduction to the project’s goal and strategy development. This approached is preferred by all of the team members as they have previous experience with scrum and are comfortable with it. Additionally, it leaves some room for alterations after the end of the initial planning phase.

## Research Questions

As research is an important aspect of the initial period of the project, the following section will list several questions that will help determine why the final software solution will help the current business flow of Amplexor.

**Main Question:** How will Amplexor improve the productivity and consistency of translations for employees by developing an SDL-Trados Plugin?

**Sub-questions:**

1. How can we ensure that the user stories are applied and done efficiently and securely? **Lab:** System test, Security Test, Unit Test  
   **Field:** Task Analysis - Library PO
   1. How can a dropdown menu in SDL Trados be created in a short amount of time?
   2. How can we solve the problems before the system runs?
   3. What type of security can we incorporate to ensure the safety of the application?
2. Can we evaluate and update the plugin based on the stakeholder’s demands & feedback?  
   **Library:** SWOT analysis  
   **Field:** Stakeholder analysis, Explore user requirements, Problem analysis  
   **Showroom:** Product review, Peer review  
   **Workshop:** Brainstorm, IT architecture sketching, Prototyping, Requirements prioritization
   1. How can we ensure that the stakeholder’s requirements will be met?
   2. How can we determine that we have set out the goals and not forgot any?
   3. What can we do to be certain that the product owner’s feedback is up to date with the progress in the project?
3. Are there any tutorials or support pages that provide an example of how to make a plugin similar to the one that has to be made for Amplexor?  
   **Library:** Community Research, Availability Product Analysis, Best good and bad practices  
   **Field:** Document analysis, Problem analysis  
   **Workshop:** Brainstorm
   1. What would be some useful resources that we can use to help in using Trados Studio?
   2. What working examples can we use as inspiration to build our plugin?

## End Products

The end products of the project can be divided into three major categories (as seen from the graph below):

* **Documentation** – All technical files that were created throughout the project.
  + **Project Plan** – This document. Used to share the team’s plans for the project to all relevant stakeholders.
  + **Test Plan & Report** – Document that describes all plans for software testing.
  + **TICT Report** – Also known as the Technology Impact Cycle Tool (TICT) Report, this document evaluates the impact the final solution will have on society.
  + **Research Report** – Document that goes in-depth about questions that were thoroughly researched in order to advance with the project.
  + **Architecture Document** – Showcases and describes all software structures using the C4 Model.
* **Software** – All code files and/or snippets that will be provided to Amplexor at the end of the project duration.
  + **Source code** – C# .NET Files
  + **Plug-in files** – The final executable that can be used to install into SDL Trados Studio
  + **Demo** – All demo content that was created during the development period.
  + **Unit tests** – Test-related files and structures used to verify the validity of the software
* **Communication** – While not all of it is stored digitally, communication is an important aspect that contributed to the state of all other end products.
  + **Sprint reviews** – Reviews that happen once every three or so weeks. A product of these reviews is written/vocal feedback.
  + **Weekly meetings** – Similarly, this results in written/vocal feedback.
  + **E-mail updates** – Information that was transferred in the form of e-mails between project stakeholders.
  + **Sprint retrospectives** – A methodology used to grade the project’s sprint intervals.
  + **Scrum standup** – A methodology used to get internal feedback and progress update between team members (internal process)

Diagram

Description automatically generated

All end products lead to the creation of a plug-in that is built-in for SDL Trados Studio, which will be used to automatize Amplexor’s translation process of documents for the European Parliament.

# Project Organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| Nick Sprock (Business Consultant)  [Nick.sprock@amplexor.com](mailto:Nick.sprock@amplexor.com) | N.S. – Nick Sprock | Product Owner  (Amplexor) | Available via Email  Meetings can be organized |
| Dennis Van Aelst  (Solution Manager)  [Dennis.vanaelst@amplexor.com](mailto:Dennis.vanaelst@amplexor.com) | D.V.A. – Dennis Van Aelst |
| Wouter Boot  (Business Consultant)  [Wouter.boot@amplexor.com](mailto:maiWouter.boot@amplexor.com) | W.B. – Wouter Boot |
| Nicole Zuurbier-Munneke  [n.munneke@fontys.nl](mailto:n.munneke@fontys.nl) | N.Z.M. – Nicole Zuurbier-Munneke | Project Tutor  Semester coach | Available via Email  Meetings can be organized  Planned Meetings Friday 9:00 – 16:00 |
| Galin Savov  [432557@student.fontys.nl](mailto:432557@student.fontys.nl) | G.S. – Galin Savov | Scrum Master | Available via Email  Available in lecture slots |
| Martin Georgiev  [426060@student.fontys.nl](mailto:426060@student.fontys.nl) | M.G. – Martin Georgiev | Team Members | Available via Email  Available in lecture slots |
| Aleksandar Ivanov  [425981@student.fontys.nl](mailto:425981@student.fontys.nl) | A.I. – Aleksandar Ivanov |
| Angel R. Angelov  [431676@student.fontys.nl](mailto:431676@student.fontys.nl) | A.A. – Angel R. Angelov |
| Aleksandra Gomoyurova  [441198@student.fontys.nl](mailto:441198@student.fontys.nl) | A.G. – Aleksandra Gomoyurova |
| Mirela Cristiana Gherlan  [435027@student.fontys.nl](mailto:435027@student.fontys.nl) | M.C.G. – Mirela Cristiana Gherlan |
| Yash Jagwani  [432295@student.fontys.nl](mailto:432295@student.fontys.nl) | Y.J. – Yash Jagwani |

## Communication

In terms of communication, the project’s team members communicate with each other using a variety of online platforms to perform meetings and chat – Discord, WhatsApp, Microsoft Teams.

All communication between the group and the semester coach is done either via e-mail or using Microsoft Teams – planned meetings every Friday. If any questions occur between stakeholders, the former is preferred but if the question in hand is too complex for a short answer, a meeting is organized.

## Organisational Structures

# Activities and time plan

## Phases of the project

The project’s phases will be defined mainly by the Scrum agile methodology. Therefore, documentation and software will be done in parallel with each other:

* **Initial Phase (Sprint 0)**
  + Problem analysis (Documentation)
  + List of requirements (Documentation)
  + Initial communication with product owner (PO)
  + Dividing team member roles
* **Development Phase (Sprint 1 – 4)**
  + Software development (Creating the plugin and extending its feature set)
  + Writing additional documentation (test plan, architecture document, research reports, etc.)
  + Unit testing
  + Frequent meetings between stakeholders
* **End Phase (Sprint 5)**
  + Quality assurance
  + Reflection
  + Evaluation
  + Project handover

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Estimated Effort** | **Start date** | **Finish date** |
| Sprint 0 | Low - Medium | 31 Aug. | 18 Sept. |
| Sprint 1 | Medium | 18 Sept. | 9 Oct. |
| Sprint 2 | High | 9 Oct. | 6 Nov. |
| Sprint 3 | High | 6 Nov. | 27 Nov. |
| Sprint 4 | High | 27 Nov. | 16 Dec. |
| Sprint 5 | Medium | 16 Dec. | 22 Jan. |

Estimated effort is based on a general prediction on the amount of work that will be required to complete each sprint. Additionally, the time taken is also taken into consideration.

# Testing strategy and configuration management

## Testing strategy

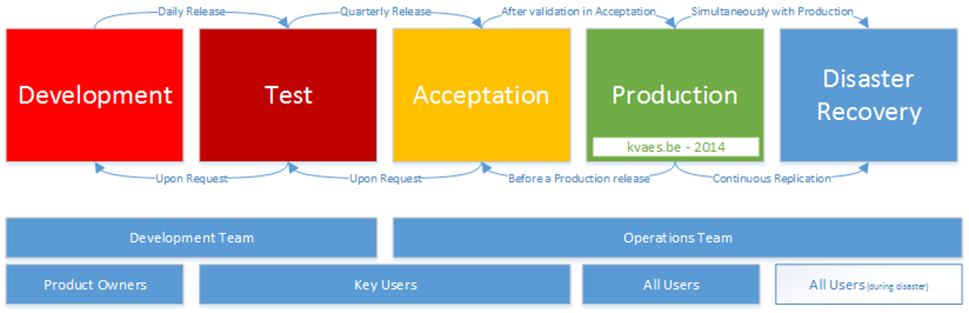
The team plans to cover a few levels of testing – unit, system and acceptance testing. Moreover, it is expected that the project’s unit tests on the software code will reach at least 80% code coverage.

* **Unit testing** – The lowest level of testing strategy, unit tests will be written for the final software code and will be executed automatically to verify code validity.
* **System testing** – Second to the highest level of testing strategy, this will be done manually to check if the application produces all desired outputs and confirm that components interact with one another and with the system as a whole without issues.
* **Acceptance testing** – The highest level of testing strategy, this will also be done manually to see whether the system satisfies the acceptance criteria – user needs and requirements, business processes, etc.

## Testing environment and required resources

We plan to utilize the DTAP Testing environment as part of our testing strategy. The acronym DTAP finds its origin in the words Development, Testing, Acceptance and Production. The DTAP-street is a commonly accepted method to have a phased approach to software development.

A typical flow works as follows :

* **Development** – This environment is where the software is developed. It is the first environment that is used. Changes are very frequent, as this is the first area where creativity is forged into a product.
* **Test** – A developer is (hopefully) not alone. In the test environment, the complete code base is merged and forged into one single product. The first attempts at standardization and alignment towards the future production environment are made here.
* **Acceptation** – Once the development team feels that the product is ready, it will be deployed to acceptance. This is a look-alike of the production and used by operations as a staging environment for production releases.
* **Production** – At this level, the product is ready for production and deployment to end-users starts.

The easiest way to manage our test environment is through automation. We can use Continuous Integration (CI) tools like Jenkins and SonarQube to assure code quality as they are a great fit for this purpose. The tool not only helps to automate this deployment process, but it also assists in running test suites.

## Configuration management

The team plans to use Gitflow as our branching strategy – a branching model that is based around project releases. Furthermore, tools such git bash will be utilized to make version management a possibility.

If requests occur during the development period, team members will be assigned to handle them accordingly. Additionally, code will go through a review before it gets merged back to the master branch. This can happen either verbally (through chatting/social systems) or using GitHub/GitLab’s review system.

# Finances and risk

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Time | Strict planning and time management | Assigning additional people to given tasks and/or providing extra hours to the project. |
| 1. Technical issues | Initial testing | Dedicating time to revolve occurring issues before development continues work on new features |
| 1. Knowledge | Initial research | Halting development to do additional research so that no other issues occur |
| 1. License Issue | Communication with product owner/stakeholders | Use of virtual machines to cover for software application trials. |

The abovementioned risks have been considered before development on the end product starts. Moreover, some countermeasures have been added so that team members do not fall for these pitfalls.