# **FENCE**

***A fuzzy sociotechnical congruence measurer***

*Antonio Manjavacas Lucas*

# **Resumen**

# **Abstract**

# **INTRODUCTION**

## **Motivation of the project**

## **State of art**

# **OBJECTIVES**

## **Main objective**

## **Specific objectives**

# **METHODOLOGY**

This chapter will address the definition and justification of the working methodology used, as well as the planification of the project. The technological resources and tools employed will also be exposed in detail.

## **Agile methodologies**

The Agile Manifesto [1] was born in 2001 as a set of four values and twelve principles associated with the search for improvements in software development over conventional methodologies. In this manifesto, the assessments of expert software practitioners were declared, stating the importance of:

* *Individuals and interactions over processes and tools*.
* *Working software over comprehensive documentation*.
* *Customer collaboration over contract negotiation*.
* *Responding to change over following a plan*.

Agile methodologies promote the continuous delivery of software in short cycles, fostering involvement, training and adjustment to stakeholders needs [2]. These methodologies mean, in turn, greater flexibility, increased productivity and, consequently, cost reductions during changing projects, where the requirements can differ throughout their development.

Nowadays, the benefits associated with agile methodologies have made them widely used by practitioners, enhancing the perceived and internal quality of software development as well as promoting their profitability.

## **OpenUP**

Being aware of the advantages associated with the use of agile methodologies in software engineering and given the iterative and incremental nature of this project, the methodology chosen for its development was *OpenUP* [3].

*OpenUp* is a minimum and sufficient methodology, which means that it only considers the fundamental contents of software development, leaving aside aspects such as the management of large teams, technology-specific guidance or contractual situations. In spite of its simplicity, *OpenUP* covers in a complete and agile way the whole development process of a software project, being completely flexible to the nature of the project in which it is employed.

The main principles of *OpenUP* offer a direct mapping with the ones expressed in the agile manifesto and try to represent the working model to follow by using this methodology:

* **Collaborate to align interest and share understanding**, advocating for coordination and mutual understanding among stakeholders.
* **Balance competing priorities to maximize stakeholder value**, trying to maximize profits while conforming to project constraints.
* **Focus on the architecture early to minimize risks and organize development**.
* **Evolve to continuously obtain feedback and improve** in order to have continuous communication with stakeholders and demonstrate incremental value to them.

### **Phases and project planning**

The use of agile, iterative and incremental methodologies such as *OpenUP* facilitates the coordination and development of projects based on multiple modules that, once developed, add value to the desired final product.

As shown in *Figure* *1*, the organization of work followed by *OpenUP* distinguishes between three different perspectives based on personal, team and stakeholder levels:

* The personal effort of an *OpenUP* project is defined as a **micro-increment**, commonly measured in hours or days.
* From the team’s perspective, an **iteration lifecycle** reflects how micro-increments are applied to obtain stable and cohesive builds of the system being produced.
* Focusing on how an overview of the project is guaranteed for the stakeholders, *OpenUP* structures the **project lifecycle** into four phases: *Inception*, *Elaboration*, *Construction*, and *Transition*.

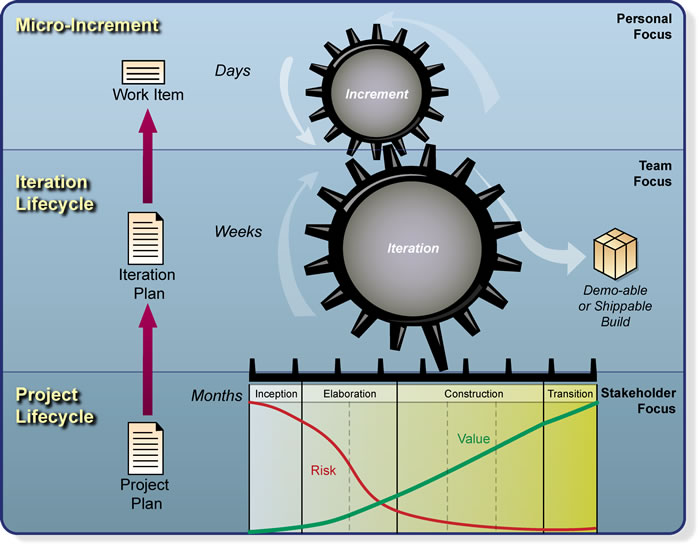


Figure 1. OpenUP layers.

Thus, each of the phases of the *OpenUP* lifecycle are defined as follows:

* **Inception phase**: in this phase, it is a question of understanding what is intended to be produced and what are the objectives and limitations of the system to be developed, identifying the stakeholders and detailing their success criteria.
* **Elaboration phase**: it involves the procurement of a more detailed understanding of the system requirements; design, implementation, validation and establishment of the architecture baseline, providing a skeleton of the system structure; mitigation of essential risks and project planning in terms of time and costs.
* **Construction phase**: iterative development of the desired product, until a tested result is achieved and ready to be offered to users. The goal pursued during this phase is to minimize costs through resource optimization and parallelization of independent tasks.
* **Transition phase**: this last phase involves validating user expectancies, obtaining stakeholders approval and seeking to improve future projects based on well-documented lessons learned.

These phases, applied to the particular case of this project, are presented in the following way:

#### **Inception phase**

In this phase, the justification of the project was carried out, identifying its scope and the objectives to be pursued. Its feasibility in terms of risks, time and estimated costs was also assessed. Roles were assigned and, at the same time, an attempt was made at identifying the key functionalities of the system in accordance with the specifications defined with the identified stakeholders.

At this stage, the first meetings were held to establish the key functionalities of the system, to understand the competencies addressed and to provide an insight into how to proceed in the months ahead.

#### **Elaboration phase**

Once an overview of the project had been established, a more detailed understanding of the objectives pursued was sought in this phase. To perform this task, a series of interviews with stakeholders were planned and conducted, allowing to know the essential features that the system should meet. On this basis, the elicitation and formalised documentation of the system's requirements was carried out, allowing to proceed with the analysis and design of the functional modules that would compose the system.

Other tasks inherent to this stage were addressed, such as the choice of technological resources to be used, the development process employed, the definition of the system architecture baseline or the acquisition of domain-specific knowledge. Actors and use cases were also formally defined.

Finally, considering the set of formally stated requirements as well as the resulting skeleton of the system to be developed, a project planning consisting of time and cost estimation was elaborated. Based on the identified functionalities, the iterations that constitute the lifecycle were organized according to the priority expressed by the stakeholders.

#### **Construction phase**

In this stage, the implementation of the different functionalities of the system was carried out. This implementation was undertaken in an orderly manner based on the priorities agreed upon with the stakeholders in the previous phase.

Presented in priority order, the following system functionalities were developed:

1. *Main user interface.*
2. *Data management.*
3. *Fuzzy STC measurement.*
4. *Data visualization.*
5. *Recommendation system.*
6. *Settings and preferences.*

The implementation of each module was done in an iterative way, trying to agree with the stakeholders any kind of modification in the functionality or the interface considered during this development stage. As would be reflected in the planning, each functional module was associated with an iteration during the construction phase of the project.

#### **Transition phase**

Finally, the transition stage was dedicated to the documentation and testing of the system, confirming compliance with the requirements with the project tutors and elaborating the final project report.

The review of the project by the tutors as well as the preparation of the project presentation were also part of this stage.

### **Roles**

In this subsection, the set of basic OpenUP roles will be defined as well as the assignment of those roles for the concrete case of this project.

* **Analyst**: is the person in charge of identifying and understanding the problems and opportunities of the project, by knowing and interpreting the requirements expressed by the customer and end-users.
* **Architect**: designs and documents the system architecture, being responsible for technical decisions on the overall implementation of the system.
* **Developer**: each person in this role is responsible for the implementation of a set of parts of the system, adjusting it to the architecture and using the necessary technologies for the development.
* **Project manager**: is the person in charge of planning the project, fulfilling the objectives as well as communicating and coordinating with the stakeholders.
* **Tester**: person in charge of the identification, definition, implementation and execution of the tests over the system.
* **Stakeholder**: people that may be directly or indirectly affected by project realization. Normally a stakeholder is a person whose needs will be met after the project is completed, like end-users o customers.

The general assignment of the *OpenUP* roles for this project is detailed in *Table 1*.

|  |  |  |
| --- | --- | --- |
| **Role** | **Person** | **Charge** |
| Analyst | Antonio Manjavacas Lucas | Grade student |
| Architect |
| Developer |
| Project manager |
| Tester |
| Stakeholder | Aurora Vizcaíno Barceló | Project tutor and ESI professor |
| Stakeholder | José Ángel Olivas | Project tutor and ESI professor |

Table 1. Assigned roles.

On the one hand, Aurora Vizcaíno was the main stakeholder in the monitoring and feedback of the project: she provided the key requirements of the system, served as a guide throughout its planning and development and acted as a contact person for any queries during the project.

On the other hand, José Ángel Olivas was responsible for recommending and guiding the implementation of the modules related to the competencies of the *Computing* intensification, such as the underlying fuzzy logic mechanisms used to describe the communications and coordination between users.

Finally, the tasks of analysis, design, development, testing and documentation were carried out by Antonio Manjavacas, the grade student responsible for the project.

## **Resources**

### **Hardware resources**

### **Software resources**

# **RESULTS**

## **Inception phase**

## **Elaboration phase**

### **Requirements**

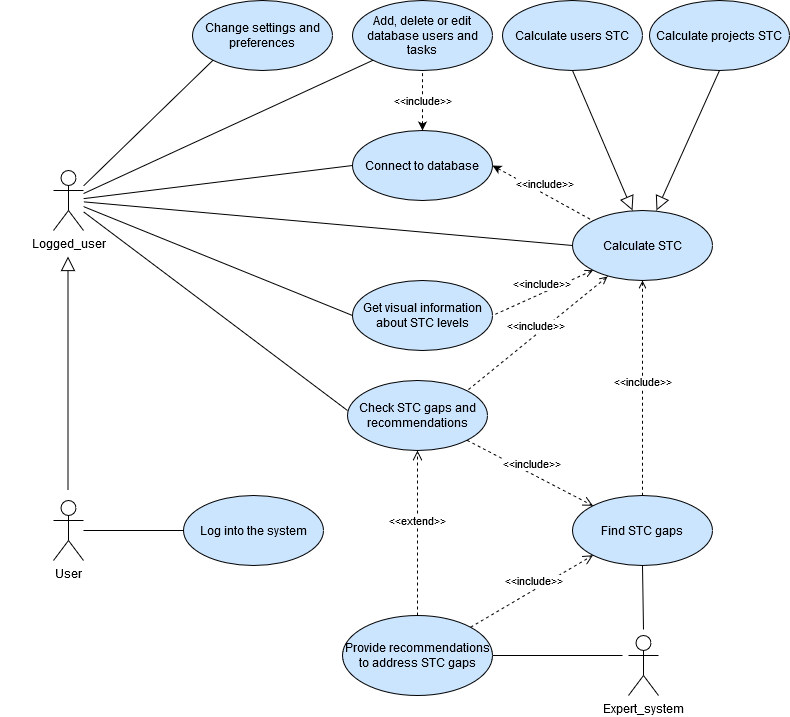
#### **Functional requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Objectives** | **Priority** |
| *FR1* | User STC measurement | The tool must calculate the STC level of the users in an organisation. | High |
| *FR2* | Project STC  measurement | The tool should calculate the STC level of an organisation's projects. | High |
| *FR3* | Factory STC  measurement | The tool must calculate the STC level between factories of an organisation. | Optional |
| *FR4* | Obtain social dependencies | The tool must obtain the dependencies between users in an organization. | High |
| *FR5* | Obtain task dependencies | The tool must obtain the dependencies between tasks in an organization. | High |
| *FR6* | Show social dependencies | The tool must show the dependencies between users in an organization. | High |
| *FR7* | Show task dependencies | The tool must show the dependencies between tasks in an organization. | High |
| *FR8* | Gaps detection | The tool should detect coordination gaps. | Medium |
| *FR9* | Gap resolution | The tool should provide information on how to address existing coordination gaps. | Medium |
| *FR10* | User STC visualization | The tool should display visual information on the users' STC levels. | Low |
| *FR11* | Project STC visualization | The tool should display visual information on the projects' STC levels. | Low |
| *FR12* | Factory STC visualization | The tool should display visual information on the factories’ STC levels. | Optional |
| *FR13* | Login system | The application will have a login system. | Medium |
| *FR14* | Communication awareness | The person will be able to evaluate his or her perceptions of communication with those offered by the tool. | Optional |
| *FR15* | Settings and preferences | The tool will allow to set preferences regarding STC calculations. | Low |

#### **Non-functional requirements**

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Objectives** |
| *NFR1* | Language labels | Language labels will be used to reflect the degree of coordination, communication and dependencies, as well as the level of language skills. |
| *NFR2* | Database | A relational database will be used to store information about users, tasks, projects and factories. |
| *NFR3* | Web application | The application will be deployed as a web service. |

### **Use cases and modules**



|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Objectives** | **Priority** |
| *M1* | Main user interface | Development of the initial user interface, including the login and the essential application windows. | High |
| *M2* | Data management | Connection to the database, management of database elements and login functionality. | High |
| *M3* | Fuzzy STC measurement | Extraction of assignments, dependencies and current coordination. Calculation of coordination requirements and level of STC at user, project and factory levels. | High |
| *M4* | Recommendation system | System for the detection of coordination gaps and recommendation of solutions to improve the levels of STC based on expert knowledge. | Medium |
| *M5* | Data visualization | Graphic representation of the STC levels, allowing to see comparisons and the variation of these levels along time. | Low |
| *M6* | Settings and preferences | Customization and tool settings configuration. | Low |

### **Planification**

|  |  |  |
| --- | --- | --- |
| **Iteration** | **Summary** | **Estimated time** |
| *IT0* | Definition of general objectives, first draft and preliminary meetings. | 1 week |
| *IT1* | Planning, requirements elicitation, role assignment, use case modelling and design. | 2 weeks |
| *IT2* | *Module 1* implementation | 1 week |
| *IT3* | *Module 2* implementation | 2 weeks |
| *IT4* | *Module 3* implementation | 2 weeks |
| *IT5* | *Module 4* implementation | 2 weeks |
| *IT6* | *Module 5* implementation | 2 weeks |
| *IT7* | *Module 6* implementation | 1 week |
| *IT8* | Documentation and tests | 2 weeks |
| *IT9* | Review and submission preparation | 1 week |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Inception phase** | **Elaboration phase** | **Construction phase** | | | | | | **Transition phase** | |
| *IT0* | *IT1* | *IT2* | *IT3* | *IT4* | *IT5* | *IT6* | *IT7* | *IT8* | *IT9* |
| **From**: 7 of January  **To**: 31 of January | **From**: 1 of February  **To**: 15 of February | **From**: 16 of February  **To**: 23 of February | **From**: 24 of February  **To**: 8 of March | **From**: 9 of March  **To**: 22 of March | **From**: 23 of March  **To**: 5 of April | **From**: 6 of April  **To**: 19 of April | **From**: 20 of April  **To**: 26 of April | **From**: 27 of April  **To**: 10 of May | **From**: 11 of May  **To**: 17 of May |
| **1 week** | **2 weeks** | **1 week** | **2 weeks** | **2 weeks** | **2 weeks** | **2 weeks** | **1 week** | **2 weeks** | **1 week** |
| **1 week** | **2 weeks** | **10 weeks** | | | | | | **3 weeks** | |
| **Total: 16 weeks** | | | | | | | | | |

### **Costs estimation**

### **Design**

### **Data model**

## **Construction phase**

### **Module 1. Main user interface**

### **Module 2. Data management**

### **Module 3. Fuzzy STC measurement**

### **Module 4. Recommendation system**

### **Module 5. Data visualization**

### **Module 6. Settings and preferences**

## **Transition phase**

# **CONCLUSIONS**

## **Achievement of the objectives**

## **Competencies justification**

## **Lessons learned**

## **Future work**

## **Personal appraisal**

# **ANNEXES**

# **References**

[1] ‘Manifesto for Agile Software Development’. [Online]. Available: https://agilemanifesto.org/. [Accessed: 31-Jan-2020].

[2] R. C. Mora, *Conversaciones con CEOs y CIOs sobre Transformación Digital y Metodologías Ágiles*, 1st ed. Madrid: Agibilibooks, 2017.

[3] R. B. MacIsaac Onno van der Straaten,Bruce, ‘Eclipse Process Framework Project (EPF) | The Eclipse Foundation’. [Online]. Available: https://www.eclipse.org/epf/. [Accessed: 28-Jan-2020].