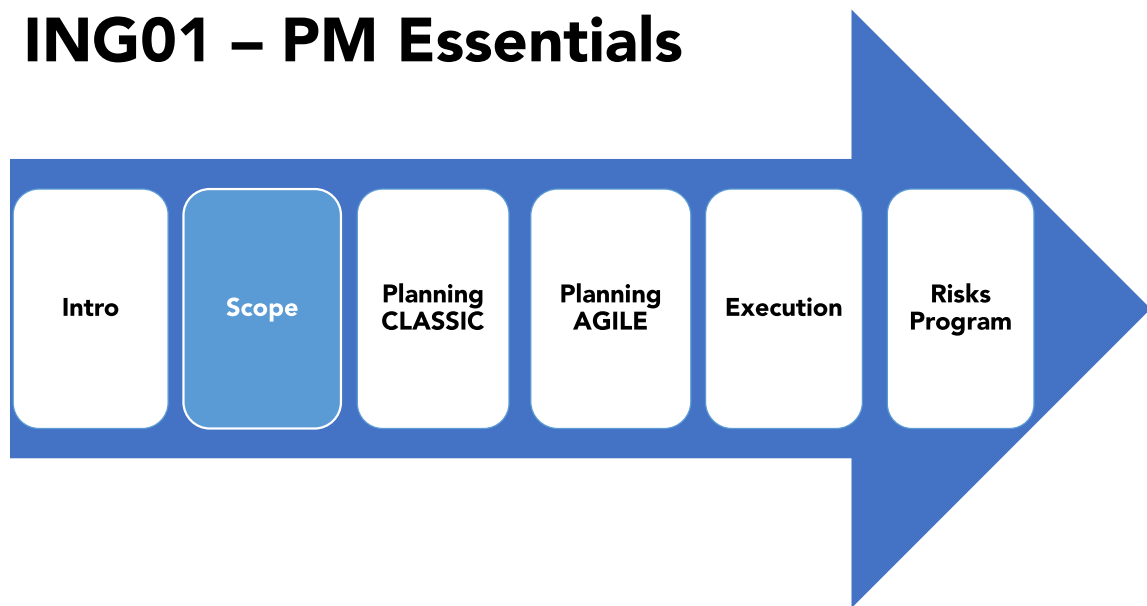


ING01 – PM Essentials



ING1-GPRO-Module 2

Product and Projet SCOPE Definition.

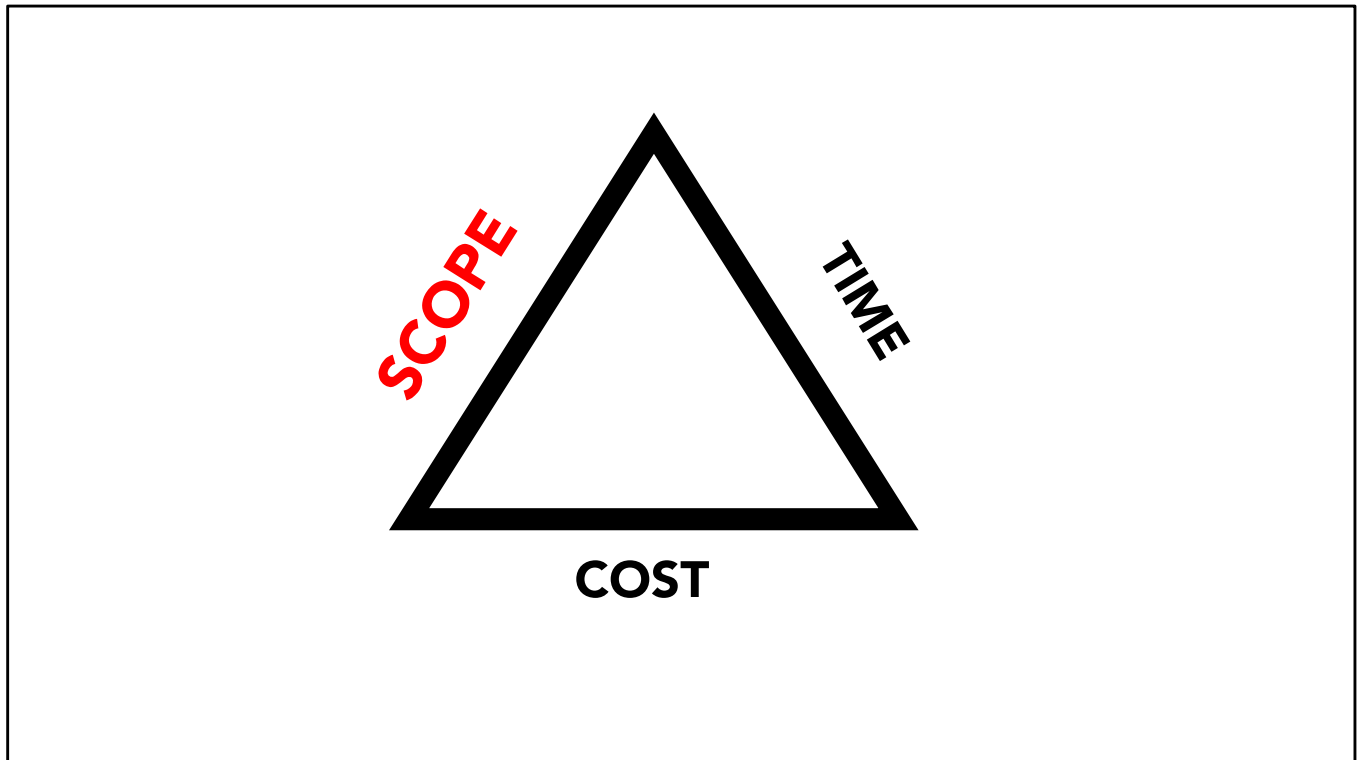
Définir le Produit et le périmètre du projet

As the project's director you must have a functional description of the product to be realized and the constraints and expectations that the project will have to meet.

This Module aims to answer this essential question.

En tant que réalisateur du projet vous devez disposer d'une description fonctionnelle du produit à réaliser et des contraintes et attentes auxquels le projet devra répondre.

Ce Module vise à répondre à cette question somme tout essentielle



Project Management aims at managing mainly SCOPE , TIME and COST

This module focuses on SCOPE management . Scope Management can be considered in different phases of the project Cycle.

This module concerns the Definition phase .

That means defining and agreeing with all the stakeholders about the scope of the product. Once this agreement is reached, then the project may be planned . During Execution phase , Scope management will be involved in controlling the performance and taking care of changes (eg. new features for the product , new documents to be provided ...)

Le Project Management vise principalement à gérer SCOPE , TIME et COST

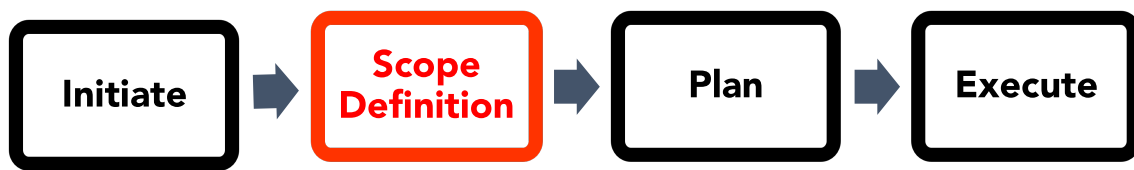
Ce module se concentre sur la gestion SCOPE (Périmètre) . La gestion du SCOPE peut être envisagée dans différentes phases du cycle du projet.

Ce module concerne la phase de définition .

Cela signifie définir et convenir avec tous les parties prenantes de la portée du produit et du projet. Une fois cet accord conclu, et le projet devra être planifié . Pendant la phase d'exécution du projet , la gestion du SCOPE sera impliquée dans le contrôle de la bonne exécution et la prise en charge des changements (par exemple de nouvelles fonctionnalités pour le produit , de nouveaux documents à fournir ...)

Project Life Cycle

Le cycle de vie projet:



In the project life cycle, the definition of the scope of the product takes place directly after the initialization of the project. This phase aims to establish with a good level of detail what is expected in terms of product. This step is obviously essential before starting project planning and implementation.

Dans le cycle de vie d'un projet, la définition du SCOPE du produit intervient directement après l'initialisation du projet, cette phase vise à établir avec un bon niveau de détail ce qui attendu en terme de produit.

Cette étape est indispensable avant de débiter la planification du projet et son exécution.

Product scope :



A Project is a temporary endeavor (effort) to create a unique product service or result.

- Description of What to do is « PRODUCT SCOPE » :

Product scope : the features and functions that characterize a product, service, or result.

- Description of How the product will be realized is called « PROJECT SCOPE » :

Project scope (Périmètre projet): the work performed to deliver a product, service, or result with the specified features and functions. This scope is determined as the result of the planning phase in Project life cycle.

SCOPE means defining what is IN and what is OUT

SCOPE also clarifies just what must be done without any unnecessary features or work

Un projet est un effort temporaire (effort) pour créer un service ou un résultat produit unique.

- *Description de ce qu'il faut faire est « PRODUCT SCOPE » : les caractéristiques et les fonctions qui caractérisent un produit, un service ou un résultat.*
- *Description de la façon dont le produit sera réalisé est appelé « PROJECT SCOPE » : Portée du projet (périmètre projet) : le travail effectué pour livrer un produit, un service ou un résultat avec les fonctionnalités et fonctions spécifiées. Le Périmètre projet est déterminé dans la phase de planning du cycle de vie du projet.*

- *SCOPE signifie définir ce qui est IN et ce qui est OUT du périmètre*
- *Il clarifie également ce qui doit être fait et uniquement , sans aucune caractéristique ou travail inutile*



**What can be found in Product Scope for
Application software development?**
**Que trouvera-t-on dans le Product Scope
d'un Projet de développement d'une
application ?**

Activité interactive demandant aux étudiants d'indiquer selon eux ce que l'on trouvera dans un project scope dans le cas d'un projet de développement d'une application.

Product Scope for Software application :

- **Functional Requirements**
- **Constraints (technical & Quality & Project)**

The Product Scope when it comes to Project to create application software consists mainly of Functional Requirements.

Functional Requirements describe the features expected by the customer that the application to be developed should offer to users.

There will also be technical constraints to the development of the application as well as quality constraints (e.g. Performances, availability, security..).

Product Scope (Périmètre Produit) pour un logiciel applicatif

- **Exigences fonctionnelles**
- **Contraintes (technique, de qualité et de projet)**

Le Périmètre produit lorsqu'il s'agit de Projet visant à créer un logiciel applicatif se compose principalement d'Exigences Fonctionnelles.

Elle décrivent les fonctionnalités attendues par le client que l'application à développer doit offrir aux utilisateurs .

On trouvera également des contraintes techniques qui s'imposent au développement de l'application ainsi que des contraintes de qualité (e.g. Performances , disponibilités , sécurité..).

Main Stakeholders in Scope Definition :

Customer team
Maitrise d'ouvrage



Realization /Development Team
Maitrise d'Œuvre



The Customer is responsible for the success of his project, it is represented by a customer project management team (in French Maitrise d'ouvrage) whose essential mission in the "Definition phase" of the project is the clear expression of the requirements.

The mission of the realization team (in French Maitrise d'oeuvre MO) is to build a product that fulfil the requirements of its client and thus understand them with fullness and detail.

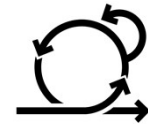
If the client is well organized, and if this project has value for him, he will have formalized his requirements more or less precisely. A relationship must be established between the Customer team and the realization team to clarify the requirements that will be in the product.

Two Approach in Project Management Software dev.:

Predictive/Classic Project Management



Agile Methodology



As described in this course introduction , we will consider 2 flavours in project Management :

Predictive /classic Project Management : it tends to be used when Product scope can be clearly and reasonably detailed, known and expressed.

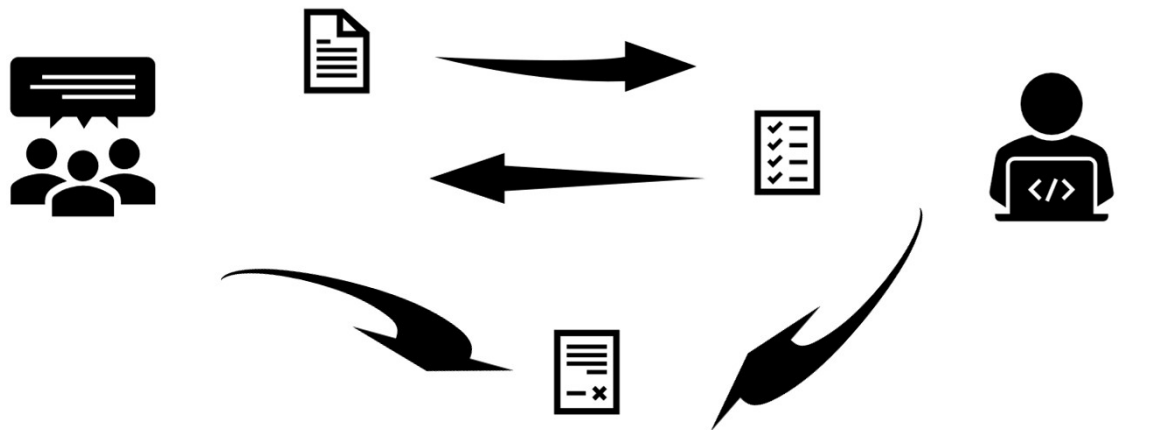
Agile methodology : it tends to be used mainly for software dev. Project where the Product scope is partially known and will be established in a collaborative and iterative manner.

Comme décrit dans l' introduction de ce cours, nous considérerons 2 type de gestion de projet :

Gestion de projet Predictive et classique : elle a tendance à être utilisé lorsque le périmètre du produit peut être clairement et raisonnablement détaillé, qu'il est connu et exprimé.

Méthode agile : elle a tendance à être utilisée principalement pour les projets de développement logiciel dont la périmètre produit est partiellement connu et sera établi de manière collaborative et itérative.

Predictive/Classic Project Management:



In a traditional model for project Management , the relationship is “the contract” and the customer expects then a commitment in terms of cost and time to carry out. In this model, the client had to establish a “Customer Specification” by providing the most accurate information on its requirements and particularly its functional requirements. These requirements are formulated in textual form: “the product must”, or in the form of a “use case “.. In theory in this relationship model, the specifications must be as complete and precise as possible. Indeed, the changes, which will happen anyway, during the project will have an impact in cost and time that the client wishes to avoid. Thus, more often than not, a long work is done to describe in detail the entire perimeter of the project. Before the contract was signed. The production team receives this customer specification, and must analyze it to imagine the Product solution that can respond to it, and identify the work to be done and then organize it in a "project plan" that constitutes the answer to the specifications.

Dans un modèle traditionnel, la relation est le contrat et l'attente du client est un engagement en terme de cout et de délai pour la réalisation.

Dans ce modèle le client a du établir un Cahier des charges en renseignant de la façon la plus précise possible ses exigences et notamment ses exigences fonctionnelles. Ces exigences se formulent sous forme textuelle: le produit doit , ou sous forme de use case ..

En théorie dans ce modèle de relation, le cahier des charges doit être le plus complet et précis possible. En effet les changements, qui arriveront quand même, au cours du projet auront un impact en coût et en délai que le client souhaite éviter.

Ainsi, le plus souvent un long travail est réalisé pour décrire de façon détaillée tout le périmètre du projet. Avant la signature du contrat.

L'équipe de réalisation reçoit ce cahier des charges, et doit en faire l'analyse pour imaginer la solution Produit qui peut y répondre, et identifier le travail à réaliser puis l'organiser dans un « plan projet » qui constitue la réponse au cahier des charges.

Terms



- Customer Spécifications
- Functional requirements
 - Use cases
- Proposal (answer to Customer Specification)
- Approved Requirement Matrix
- Cahier des charges
- Exigences Fonctionnelles
 - Cas d'utilisation
- Proposition (réponse au cahier des charges)
- Tableau des exigences (approuvées)



Presence Project : Functional Requirements .



Recall Presence Project context and Ask to students to give somme example of Functionnal requirements

Extract from Requirement Matrix



ID	Section	Description de l'exigence	Commentaire
Exigences Fonctionnelles			
001	1	Application Mobile Evènements	
	1.1	Avant un évènement	
	1.1.1	S'authentifier en badgeant sa carte	
	1.1.2	Lister les évènements accessibles par l'utilisateur	
	1.2	Pendant un évènement	
	1.2.1	Débuter la prise de présence pour un évènement	
	1.2.2	Terminer la prise de présence pour un évènement	
	1.2.3	Scanner les cartes NFC pour noter la présence d'un étudiant	
	1.2.4	Enregistrer la présence d'un étudiant à la main	
	1.3	Navigation	
	1.3	Sécurité avec code pour naviguer dans l'application	

Here is a small part of the Requirement matrix for Presence Project



Agile Methodology: The Product Backlog

In an Agile approach, the product to be built is described by in a document Called "Product Backlog" that will live throughout the construction of the product.

The features we want to develop are described in the form of "user stories".

The detail of these user stories is refined according to their priority in terms of achievement and value for the project.

The most important and prioritarian are refined in precise and specific user stories , the needs that are less important can me formulated as a generic idea called EPIC.

The main difference with the classical approach is the fact that all the requirements are not fully specified and will be refined , restated iteratively during project execution .

Dans une approche Agile, le produit a construire est décrit par un document de « Product Backlog » qui va vivre tout au long de la construction du produit .

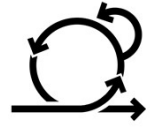
Les fonctionnalités que l'on souhaite développer sont décrites sous la forme de "user stories".

Le detail de ces user stories est raffiné en fonction de leur priorité en terme de réalisation et de valeur pour le projet.

Les plus importants et les plus prioritaires sont affinés dans des user stories précises et spécifiques, les besoins qui sont moins importants peuvent me formuler comme une idée générique appelée EPIC.

La principale différence avec l'approche classique est le fait que toutes les exigences ne

*sont pas entièrement spécifiées et seront affinées , réformulées, précisées lors de l'exécution du projet .
C'est une approche itérative.*



User Story:

**En tant que <qui>, je veux
<quoi> afin de <pourquoi>.**

L'intégralité du travail à réaliser est découpée en incréments fonctionnels et les activités de développement s'organisent autour de ces incréments appelés "User Stories".

Techniquement il va s'agir de "morceler" l'ensemble des objectifs fonctionnels d'un projet en User Stories.

Adopter la pratique des User Stories implique de tenir pour généralement vraies un certain nombre d'hypothèses sur ces incréments: on peut les réaliser dans un ordre arbitraire, et chacun indépendamment des autres peut avoir une valeur pour l'utilisateur.

Un récit utilisateur est une phrase simple dans le langage de tous les jours permettant de définir avec suffisamment de précision le contenu d'une fonctionnalité à développer. Elle contient généralement trois éléments descriptifs de la fonctionnalité, sous la forme :

En tant que <qui>, je veux <quoi> afin de <pourquoi>

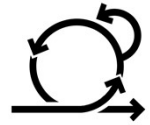
Le *qui* indique l'utilisateur du point de vue duquel on se place. Il s'agit souvent

d'un rôle dans l'usage du produit . Pour mieux illustrer la diversité des besoins, on peut également utiliser le concept de [persona](#), c'est-à-dire une personne fictive, représentative de catégories d'usagers du produit, et à laquelle on peut s'identifier pour mieux comprendre ses attentes. L'identification et la description des personas se fait alors avant de commencer l'écriture des récits utilisateurs.

Le *quoi* décrit succinctement la fonctionnalité ou le comportement attendu. Le but du récit n'est pas d'en fournir une explication exhaustive. Cette dernière est obtenue de façon interactive au cours d'une conversation avec les utilisateurs concernés ou leurs représentants².

Le *pourquoi* permet d'identifier l'intérêt de la fonctionnalité et d'en justifier le développement. Il permet également de mieux évaluer la priorité du récit. Lorsque des points importants sont identifiés au moment d'écrire un récit, on peut faire une annotation. Ceci permet de garder la phrase simple, tout en gardant à l'esprit ces points⁷.

Chaque récit utilisateur doit être complété avec des **critères d'acceptation**. C'est une liste d'éléments qui permettront à l'utilisateur de confirmer que la fonctionnalité, une fois livrée, correspond effectivement aux attentes². Le fait de définir ces critères à l'avance permet aussi d'assurer que la description de la fonctionnalité est suffisamment précise pour être réalisable.



User Story:

**As a <type of user >, I want
<some goal > so that <some
reason >.**

The entire work to be done is cut into functional increments and development activities are organized around these increments called "User Stories".

Technically it will be a case of "breaking" all the functional objectives of a project into User Stories. Adopting the practice of User Stories implies taking for generally true a number of assumptions about these increments:

they can be realized in an arbitrary order, and each independently of the others can have value for the user.

A user story is a simple phrase in everyday language that allows you to define with sufficient precision the content of a feature to be developed.

It usually contains three descriptive elements of the feature, in the form: As a <type of user >, I want <some goal> so that <some reason>.

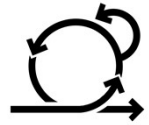
The <type of user > tells the user from the point of view of which we place ourselves. This is often a role in the use of the product. To better illustrate the diversity of needs, we can also use the concept of persona, that is, a fictitious person, representative of categories of users of the product, and to whom one can identify to better understand his expectations.

The identification and description of the personas is then done before starting the writing of user stories.

The result is a short description of the expected functionality or behavior. The

purpose of the story is not to provide an exhaustive explanation. This is obtained interactively during a conversation with affected users or their representatives. The <some goal> identifies the value of the feature and justify its development. It also helps to better assess the priority of the story.

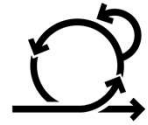
When important points are identified when writing a story, an annotation can be made. This keeps the sentence simple, while keeping these points in mind. Each user story must be completed with acceptance criteria. This is a list of items that will allow the user to confirm that the feature, once delivered, actually meets expectations. Defining these criteria in advance also ensures that the description of the feature is accurate enough to be achievable.



Presence Project : Functional Requirements as user stories?

Recall Presence Project context and Ask to students to express one functional requirement as a user story .

Extract from Product Backlog



Product B Présence Événement

	En tant que	Je veux	Afin de	Critère d'acceptation
0001	Resp. Événement	m'authentifier en badgeant ma carte et voir les événements qui me sont associés sur le device d'enregistrement	pouvoir retrouver l'événement en cours	badgeage - reconnaissance, affichage des événements
0002	Resp. Événement	lancer la session d'enregistrement de la présence sur un événement	commencer le badgeage de présence des participants	device prêt à l'opération de recueil de présence
0003	Participant	pouvoir badger ma carte RFC	confirmer ma présence à l'événement	retour OK sur badge
0004	Resp. Événement	Pouvoir proposer un enregistrement manuel	être capable d'enregistrer des participants sans carte	prêt à saisir
0005	Participant	Saisir mes informations d'identification pour un enregistrement manuel	enregistrer ma présence	présence enregistrée
0006	Resp. Événement	Clore la session d'enregistrement	fermer la prise de présence sur l'événement et produire la liste de présence	Evenement clos, plus de prise en compte de nouvelle présence. Liste de présence disponible.

Here is a small part of the Product backlog for Presence Project

Get the Product Scope

This slide introduce the section dedicated to building the Product Scope.
A product scope is required to start the project under the form of a detailed customer Specification or a Product Backlog depending on the methodology

**Situation 1 : Customer team provides
fully Documented Product/Project
Scope**
**Dev team reviews the Scope with
appropriate Product Owner**

In Situation 1 , Realization team needs to review the product scope provided by customer team to understand the content and possibly start negotiating with Customer to agree on the actual content expected to be developed.

In Agile methodology a Project Owner manages the Product back log content. If Customer has defined a Product backlog that means the Project Owner is clearly one person identified in Customer team .

Situation 2 : Customer team does not provide Product/Project Scope

Dev team needs to collaborate with Customer Team to build this Product scope

Depending on the nature of the projects, it is common for the Product scope to be not clearly documented. This is the case for the vast majority of end-of-study projects or projects carried out during internships. Indeed, the client does not invest to the maximum, unlike strategic or expensive projects, and often does not take the time to clarify his needs.

However, it is important for you to have a functional perimeter established with the customer. Thanks to this perimeter that we will organize the project, its stages, its deliverables, its releases ... It is therefore up to you to formalize this functional perimeter by replacing the customer's shortcomings.

Nevertheless, it is necessary to co-build it and define with the client a way of working to keep this functional perimeter alive. So you have to convince the client that this is a thing to do, plan the formalization mode, plan meetings and with whom. Etc.. You can't, and shouldn't, do a Product Backlog alone based on Project Charter. We have to co-build it.

In the case of a perimeter partially known, an Agile approach that allows iterations is to be preferred because it will be difficult to accurately and fully capture the entire perimeter of the client.

Suivant la nature des projets, il est fréquent que le périmètre fonctionnel ne soit pas clairement documenté. C'est le cas de l'immense majorité des Projets de fin d'étude ou

projets exécutés lors de stages . En effet, le client ne s'investit pas au maximum, contrairement à des projets stratégiques ou coûteux, et souvent ne prend pas le temps de mettre au clair ses besoins.

Pour autant, il est important pour vous, d'avoir un périmètre fonctionnel établi avec le client. Des spécifications fonctionnelles de réalisation sont une étapes ultérieures qui préparent une réalisation d'une partie du périmètre, elles ne se substituent pas au périmètre fonctionnel du projet. C'est grâce a ce périmètre que l'on va organiser le projet, ses étapes, ses livrables ,ses releases ...

Il vous appartient donc de formaliser ce périmètre fonctionnel en se substituant aux manquements du client . Il n'en reste pas moins qu'il faut le co-construire et définir avec le client un mode de travail pour faire vivre ce périmètre fonctionnel.

Donc, il faut convaincre le client que c'est une chose à faire, prévoir le mode de formalisation, prévoir les réunions et avec qui . Etc..

Vous ne pouvez pas, et ne devez pas, faire un Product Backlog dans votre coin sur la base du Project Charter. Il faut le co construire.

Dans les cas de périmètre flou au départ, une approche de type Agile qui permet des itérations, est à privilégier car il sera difficile de capter avec précision et entièreté tout le périmètre du client.

Collecting requirements

Recueillir les exigences



Documentation Analysis

The documentation of the existing systems

Press Articles , comparative studies, market research,

Internet forums

Existing specifications often contain requirements that can be repeated general or detailed functional

specifications of existing commercial software are also a wealth of information that can be reused, after analysis, to develop requirements: anomaly sheets, test reports, help-desk reports Standards, Existing in competitors or other company in a comparable field

System interface analysis(search for information on interfaces with external systems interacting) **Analysis of existing user interfaces**

Brainstorming.

Brainstorming is a technique used to generate and collect multiple ideas related to project and product requirements.

Interviews.

An interview is a formal or informal approach to elicit information from stakeholders by talking to them directly. It is typically performed by asking prepared and spontaneous questions and recording the responses. Interviews are often conducted on an individual basis between an interviewer and an interviewee. Interviews are also useful for obtaining confidential information.

Focus groups - WORKSHOP.

Focus groups bring together prequalified stakeholders and subject matter experts to learn about their expectations and attitudes about a proposed product, service, or result. A trained moderator guides the group through an interactive discussion designed to be more conversational than a one-on-one interview.

Questionnaires and surveys.

Questionnaires and surveys are written sets of questions designed to quickly accumulate information from a large number of respondents. Questionnaires and/or surveys are most appropriate with varied audiences, when a quick turnaround is needed, when respondents are geographically dispersed, and where statistical analysis could be appropriate.

Benchmarking.

Benchmarking involves comparing actual or planned products, processes, and practices to those of comparable organizations to identify best practices, generate ideas for improvement, and provide a basis for measuring performance. The organizations compared during benchmarking can be internal or external.

Formalizing the requirments

Formaliser les exigences



Formalizing means :

- Giving a Structure
- Modelizing
- Prioritizing .

In a Predictive / Classical approach the structuring is free: Sentences , use case description , table of requirements

In a Agile /Scrum approach, a product backlog strutation is required in the form of user Stories

Il s'agit d'activités de type

- *Structurer*
- *Modéliser*
- *Prioritiser .*

Dans une démarche traditionnelle la structuration est libre : Cahier des charges , use case , tableau des exigences

Dans une démarche Scrum, une struturation en Product Backlog s'impose sous la forme de user Stories

4 types of Requirements (software Projects)

- **Functional Requirements**
- **Technical Constraints**
- **Quality requirements**
- **Project Requirements**

4 types d'exigences (projets dev de logiciel)

- **Exigences Fonctionnelles**
- Contraintes techniques
- Exigences Qualité
- Exigences du Projet

Functional Requirements

Exigences Fonctionnelles

Not Dev. Functional Specifications



Functional requirements describe the functionality of the software as expected by the customer.

Functional Requirements are the essential part of product backlog or customer specifications.

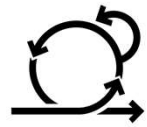
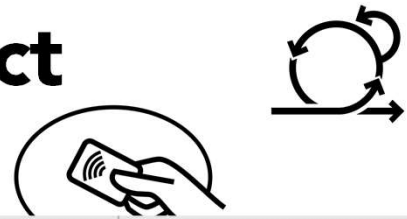
Warning, these are not functional specifications of development that they will come in the course of the project to specify how the features will be realized (screen mockup, description of the course of the application, data manipulated).

Les exigences fonctionnelles décrivent les fonctionnalités que doivent réaliser le logiciel tel qu'attendu par le client .

Elles représentent la part essentielle du product scope , Product Backlog , ou cahier des charges

Attention , ce ne sont pas des Spécification fonctionnelles de développement qui elles viendront en cours de réalisation du projet pour préciser comment les fonctionnalités seront réalisées (mockup d'écran , description du déroulement de l'application , données manipulées).

Extract from Product Backlog



Product B Présence Événement

	En tant que	Je veux	Afin de	Critère d'acceptation
0001	Resp. Événement	m'authentifier en badgeant ma carte et voir les événements qui me sont associés sur le device d'enregistrement	pouvoir retrouver l'événement en cours	badgeage - reconnaissance, affichage des événements
0002	Resp. Événement	lancer la session d'enregistrement de la présence sur un événement	commencer le badgeage de présence des participants	device prêt à l'opération de recueil de présence
0003	Participant	pouvoir badger ma carte RFC	confirmer ma présence à l'événement	retour OK sur badge
0004	Resp. Événement	Pouvoir proposer un enregistrement manuel	être capable d'enregistrer des participants sans carte	prêt à saisir
0005	Participant	Saisir mes informations d'identification pour un enregistrement manuel	enregistrer ma présence	présence enregistrée
0006	Resp. Événement	Clore la session d'enregistrement	fermer la prise de présence sur l'événement et produire la liste de présence	Evenement clos, plus de prise en compte de nouvelle présence. Liste de présence disponible.

Here is a small part of the Product backlog for Presence Project

Technical Constraints & Interfaces

Contraintes Techniques & Interfaces



Il convient de recenser avec le client les contraintes techniques à prendre en compte généralement se décompose en :

- Interfaces du produit avec l'environnement client. (Existence de Spécification techniques d'interface ?)
- Environnement physique d'utilisation : Sur quel environnement cible le produit doit fonctionner .
- Environnement matériel et logiciel : Normes et standards du client , choix de produits logiciels du marché à utiliser , contraintes devops ...

Presence Project : Technical Constraints?



Imagine the technical constraints applicable to presence project
Eg:

- mobile android 9.0 version 28
- Interface CRI for authentification
- developments of backend & backoffice delivered in Dockers containers

Quality Requirements

Exigences Qualité



ISO25000 Software quality norm can be a good guideline to review quality domain and check with customer the requirements that can be applied .

Pour rechercher les exigences qualité avec le client on peut s'appuyer sur les domaines qui qualifient

la qualité d'un produit logiciel selon la norme ISO25000

ISO 25000

- | | |
|--|---|
| • Functional Suitability
Adéquation fonctionnelle | • Reliability
Fiabilité |
| • Performances | • Security
Sécurité |
| • Compatibility
Compatibilité | • Maintainability
Maintenabilité |
| • Usability
Facilité d'utilisation | • Portability
Portabilité |

Here are the domains next slide gives subdomains .

Voici les domaines principaux qu'identifie la norme pour un produit logiciel , on peut se reporter également à la slide suivante qui donne une décomposition de ces domaines de qualité en sous domaines

ISO 2500 Quality Domains



Presence Project : Quality Requirements?



Imagine some quality Constraints applicable to presence project

Eg:

- Easy reading NFC on the portable (95% of success and response time <2s)
- Security : Not possible to have fake presence recording

Project Requirement

Contraintes de Projet et Services d'accompagnement



Understanding the customer's needs requires to take into account:

- Project constraints for example:
 - Cost
 - Planning
 - Installation; move to production , acceptance
 - Migration
 - Development environments
- Needs in terms of support services
 - Integration
 - User support
 - technical support
 - Corrective Support
 - Evolution Support
 - Documentation
 - Training

La compréhension des besoins du client s'accompagne également de la prise en compte de :

ses contraintes Projet par exemple :

- *Charges coûts*
- *Planning*
- *Installation; mise en exploit; recette*
- *Migration*
- *Env de dev*

Ses besoins en terme de services d'accompagnement

- *Exigence d'intégration*
- *Support utilisateur support technique*
- *Maintenance corrective*
- *Livraison de version maintenance évolutive*
- *Documentation*
- *Formation usera*
- *Généralisation*
- *Évolution probables*
- *Impact prévus du nouveau système*

La compréhension des besoins du client s'accompagne également de la prise en compte de :

ses contraintes Projet par exemple :

- *Charges coûts*
- *Planning*
- *Installation; mise en exploit; recette*
- *Migration*
- *Env de dev*

Ses besoins en terme de services d'accompagnement

- *Exigence d'intégration*
- *Support utilisateur support technique*
- *Maintenance corrective*
- *Livraison de version maintenance évolutive*

- *Documentation*
- *Formation users*
- *Généralisation*
- *Évolution probables*
- *Impact prévus du nouveau système*

Presence Project : Project Requirements?



Imagine some Project Constraints applicable to presence project

Eg:

- A first release to test for a given event at EPITA planned at a specific date
- Provide all required documentation and transfert in order other team can maintain and make evolutions.

Conclusion : Definition of Product SCOPE is **mandatory before starting any planning or worse any software development**

The product Scope is then essential . Normally the product scope is under Customer Team Responsibility. in Agile methodology Product Scope is the Product Backlog managed by the Product Owner.

The project Scope will be defined in the planning phase of the life cycle and will express more the project deliverables we intend to provide

