

Analyse : de l'Exigence à la Spécification

Ressource R3.03

P. Valarcher

IUT SF, UPEC

8 octobre 2025

Plan de la présentation

- 1 L'analyse des Exigences
- 2 La Formalisation des Exigences
- 3 Un exemple complet

Pourquoi l'analyse est-elle cruciale ?

- Une erreur détectée en phase d'analyse coûte **beaucoup moins cher** à corriger qu'une fois l'application déployée.

Pourquoi l'analyse est-elle cruciale ?

- Une erreur détectée en phase d'analyse coûte **beaucoup moins cher** à corriger qu'une fois l'application déployée.
- L'analyse est un **investissement** pour s'assurer de construire **le bon produit**.

Pourquoi l'analyse est-elle cruciale ?

- Une erreur détectée en phase d'analyse coûte **beaucoup moins cher** à corriger qu'une fois l'application déployée.
- L'analyse est un **investissement** pour s'assurer de construire **le bon produit**.
- Le but : Transformer un besoin flou en un cahier des charges clair et actionnable.

Partie 1 :

L'analyse des Exigences

(Requirements Analysis)

1.1 - Recueil des besoins : Distinguer la Demande du Besoin

La Demande : La solution envisagée par le client.

« *Je veux un bouton pour exporter ces données en PDF.* »

1.1 - Recueil des besoins : Distinguer la Demande du Besoin

La Demande : La solution envisagée par le client.

« *Je veux un bouton pour exporter ces données en PDF.* »

Le Besoin Réel : Le problème fondamental à résoudre.

« *Je dois pouvoir partager un rapport synthétique et non-modifiable avec ma direction chaque semaine.* »

1.1 - Recueil des besoins : Distinguer la Demande du Besoin

La Demande : La solution envisagée par le client.

« *Je veux un bouton pour exporter ces données en PDF.* »

Le Besoin Réel : Le problème fondamental à résoudre.

« *Je dois pouvoir partager un rapport synthétique et non-modifiable avec ma direction chaque semaine.* »

Votre Mission : Mener l'enquête pour trouver le **POURQUOI** derrière la demande.

L'art de poser les bonnes questions : Les bonnes pratiques

Posez des questions ouvertes qui encouragent la discussion :

- « *Pourriez-vous me décrire une journée type de travail avec le système actuel ?* »
- « *Qu'est-ce qui vous frustre le plus aujourd'hui ?* »
- « *Si vous aviez une baguette magique, que ferait l'application pour vous ?* »

L'art de poser les bonnes questions : Les bonnes pratiques

Posez des questions ouvertes qui encouragent la discussion :

- « *Pourriez-vous me décrire une journée type de travail avec le système actuel ?* »
- « *Qu'est-ce qui vous frustre le plus aujourd'hui ?* »
- « *Si vous aviez une baguette magique, que ferait l'application pour vous ?* »

Objectif

Révéler le contexte, la douleur de l'utilisateur et la vraie valeur attendue.

Les erreurs à éviter : Mauvaises pratiques

Évitez les questions fermées ou qui suggèrent déjà la solution :

- « *Voulez-vous un bouton bleu ici ?* »
- « *Donc il vous faut juste un tableau avec les noms et les dates ?* »
- « *Le système doit-il utiliser une base de données SQL ?* »

Les erreurs à éviter : Mauvaises pratiques

Évitez les questions fermées ou qui suggèrent déjà la solution :

- « *Voulez-vous un bouton bleu ici ?* »
- « *Donc il vous faut juste un tableau avec les noms et les dates ?* »
- « *Le système doit-il utiliser une base de données SQL ?* »

Risque

Passer à côté de besoins non anticipés et s'orienter trop vite vers des détails techniques.

1.2 - Exigences Fonctionnelles (EF) vs Non-Fonctionnelles (ENF)

Une exigence, pour être utile, doit être :

- Non-ambiguë
- Complète
- **Vérifiable (Testable)**

EF vs ENF : Deux questions distinctes

Exigences Fonctionnelles (EF)

Décrivent **CE QUE** le système doit faire.

- Les fonctionnalités, les actions, les comportements.

Ex : « L'utilisateur doit pouvoir se connecter avec un email et un mot de passe. »

Exigences Non-Fonctionnelles (ENF)

Décrivent **COMMENT** le système doit se comporter.

- Les qualités et les contraintes.

Ex : « Le temps de connexion doit être inférieur à 2 secondes. »

Exemples d'Exigences Fonctionnelles (EF)

- **Gestion des utilisateurs :**

- Le système doit permettre à un administrateur de créer des comptes.
- L'utilisateur doit pouvoir réinitialiser son mot de passe via un lien email.

- **E-commerce :**

- Le système doit appliquer une réduction si le panier contient + de 3 articles identiques.
- Le système doit autoriser le paiement via Carte Bancaire et PayPal.

- **Reporting :**

- Le système doit permettre l'export des transactions au format CSV.

Exemples d'Exigences Non-Fonctionnelles (ENF)

- **Performance et Rapidité :**
 - Toutes les pages doivent se charger en **moins de 3 secondes**.
- **Sécurité :**
 - Les mots de passe doivent être stockés en utilisant l'algorithme Argon2id.
 - Le système doit déconnecter un utilisateur après **20 minutes d'inactivité**.
- **Fiabilité et Disponibilité :**
 - Le système doit être disponible **99.8%** du temps.

La frontière entre EF et ENF : Le Contexte est Roi

Exigence

« *Le système doit analyser un fichier pour détecter la présence de virus.* »

La frontière entre EF et ENF : Le Contexte est Roi

Exigence

« *Le système doit analyser un fichier pour détecter la présence de virus.* »

Contexte 1 : Logiciel Antivirus

C'est la raison d'être du produit.

Exigence Fonctionnelle

Contexte 2 : Partage de photos

C'est une contrainte de sécurité lors d'une autre action.

Exigence Non-Fonctionnelle

Résumé : EF vs ENF

EF = **CE QUE** le système FAIT

- **À faire** : Être précis, décrire une action observable.

ENF = **COMMENT** le système le FAIT

Résumé : EF vs ENF

EF = **CE QUE** le système FAIT

- **À faire** : Être précis, décrire une action observable.
- **À ne pas faire** : Rester vague (« *bien gérer les utilisateurs* »).

ENF = **COMMENT** le système le FAIT

Résumé : EF vs ENF

EF = **CE QUE** le système FAIT

- **À faire** : Être précis, décrire une action observable.
- **À ne pas faire** : Rester vague (« *bien gérer les utilisateurs* »).

ENF = **COMMENT** le système le FAIT

- **À faire** : Utiliser des métriques quantifiables.

Résumé : EF vs ENF

EF = CE QUE le système FAIT

- **À faire** : Être précis, décrire une action observable.
- **À ne pas faire** : Rester vague (« *bien gérer les utilisateurs* »).

ENF = COMMENT le système le FAIT

- **À faire** : Utiliser des métriques quantifiables.
- **À ne pas faire** : Utiliser des adjectifs subjectifs (« *le site doit être rapide* »).

Partie 2 :

La Formalisation des Exigences

De la discussion à l'action

Une fois les besoins recueillis, il faut les écrire pour qu'ils soient :

- Clairs pour le **client** (qui valide).
- Exploitable par l'**équipe de développement** (qui réalise).

De la discussion à l'action

Une fois les besoins recueillis, il faut les écrire pour qu'ils soient :

- Clairs pour le **client** (qui valide).
- Exploitable par l'**équipe de développement** (qui réalise).

Approche classique

Cahiers des charges de 100+ pages (longs, obsolètes, peu lus).

Approche Agile

Les **User Stories**, centrées sur l'humain et la valeur.

2.1 - Les User Stories : La Voix de l'Utilisateur

Definition

Une **User Story** est une description simple, courte et en langage naturel d'une fonctionnalité, racontée du point de vue de l'utilisateur.

2.1 - Les User Stories : La Voix de l'Utilisateur

Definition

Une **User Story** est une description simple, courte et en langage naturel d'une fonctionnalité, racontée du point de vue de l'utilisateur.

Son but n'est pas de tout documenter, mais d'**amorcer une conversation**.

C'est une **promesse de dialogue**.

La structure d'une User Story

Format populaire

En tant que <rôle>,
je veux <action>
afin de <bénéfice>.

La structure d'une User Story

Format populaire

En tant que <rôle>,
je veux <action>
afin de <bénéfice>.

- **QUI?** (<rôle>) : Un acteur précis du système (client, admin...).

La structure d'une User Story

Format populaire

En tant que <rôle>,
je veux <action>
afin de <bénéfice>.

- **QUI?** (<rôle>) : Un acteur précis du système (client, admin...).
- **QUOI?** (<action>) : L'exigence fonctionnelle, l'objectif.

La structure d'une User Story

Format populaire

En tant que <rôle>,
je veux <action>
afin de <bénéfice>.

- **QUI?** (<rôle>) : Un acteur précis du système (client, admin...).
- **QUOI?** (<action>) : L'exigence fonctionnelle, l'objectif.
- **POURQUOI?** (<bénéfice>) : **Le cœur de la story**. La valeur métier.

User Stories : Bonnes et Mauvaises Pratiques

Bonne Pratique : Le Bénéfice est clair

« En tant que client régulier, je veux sauvegarder mon adresse de livraison afin de ne pas avoir à la resaisir à chaque commande. »

User Stories : Bonnes et Mauvaises Pratiques

Bonne Pratique : Le Bénéfice est clair

« *En tant que client régulier, je veux sauvegarder mon adresse de livraison afin de ne pas avoir à la resaisir à chaque commande.* »

Mauvaise Pratique : Description de l'interface

« *En tant qu'utilisateur, je veux un champ de texte pour mon adresse.* »
(N'explique pas la finalité métier.)

Qu'est-ce qu'une "Epic" ?

Definition

Une **Epic** est une User Story de très grande taille, trop complexe pour être réalisée en une seule fois (un sprint).

- Sert de **conteneur** pour des exigences larges.
- Son but est d'être **découpée** en plusieurs User Stories plus petites et gérables.

Qu'est-ce qu'une "Epic" ?

Definition

Une **Epic** est une User Story de très grande taille, trop complexe pour être réalisée en une seule fois (un sprint).

- Sert de **conteneur** pour des exigences larges.
- Son but est d'être **découpée** en plusieurs User Stories plus petites et gérables.

Analogie

Si les User Stories sont les scènes d'un film, l'Epic est l'acte complet.

La qualité d'une User Story : Les critères INVEST

Un guide pour s'assurer qu'une User Story est prête pour le développement :

I - Independent (Indépendante)

N - Negotiable (Négociable)

V - Valuable (Ayant de la valeur)

E - Estimable (Estimable)

S - Small (Petite)

T - Testable (Testable)

2.2 - Les Cas d'Utilisation : Le Scénario de l'Interaction

- Une technique de spécification plus formelle, issue du standard **UML**.

2.2 - Les Cas d'Utilisation : Le Scénario de l'Interaction

- Une technique de spécification plus formelle, issue du standard **UML**.
- Si la User Story est la **voix** de l'utilisateur, le Cas d'Utilisation (Use Case) est le **script détaillé** de la conversation entre cet utilisateur et le système.

2.2 - Les Cas d'Utilisation : Le Scénario de l'Interaction

- Une technique de spécification plus formelle, issue du standard **UML**.
- Si la User Story est la **voix** de l'utilisateur, le Cas d'Utilisation (Use Case) est le **script détaillé** de la conversation entre cet utilisateur et le système.
- Il vise à décrire de manière **exhaustive** toutes les interactions possibles pour atteindre un objectif.

Structure d'un Cas d'Utilisation

Les éléments clés :

- Nom et Acteurs
- Préconditions (ce qui doit être vrai avant de commencer)
- **Scénario Nominal ("Happy Path")** : La séquence où tout se passe comme prévu.
- **Scénarios Alternatifs et d'Exception** : Que se passe-t-il si une erreur survient ?
- Postconditions (l'état du système en cas de succès ou d'échec)

Exemple : Cas d'Utilisation "Payer sa commande"

Scénario Nominal

- 1 Client clique sur « Payer ».
- 2 Système affiche formulaire.
- 3 Client saisit infos.
- 4 Système transmet demande.
- 5 Banque **autorise**.
- 6 Système affiche confirmation.

Scénario d'Exception

- 5a. Banque **refuse**.
- 6a. Système affiche un message d'erreur et invite à réessayer.

Partie 3 :

Un exemple complet

Cas Pratique : "InnovateBox"

Le Besoin

L'entreprise Alpha Corp veut une plateforme web où les employés peuvent soumettre des idées et les managers les consulter.

Cas Pratique : "InnovateBox"

Le Besoin

L'entreprise Alpha Corp veut une plateforme web où les employés peuvent soumettre des idées et les managers les consulter.

Exigences de Haut Niveau :

- REQ-01: Les employés doivent pouvoir soumettre de nouvelles idées.
- REQ-02: Les managers doivent pouvoir consulter la liste des idées.
- REQ-03: Le système doit envoyer une notification au manager.
- REQ-04 (ENF) : Toutes les communications doivent être sécurisées (HTTPS).

Pour l'Employé (REQ-01)

*En tant que **employé**, je veux **soumettre une idée (...)** afin de **partager une suggestion d'amélioration** pour l'entreprise.*

Approche Agile : Les User Stories

Pour l'Employé (REQ-01)

*En tant que **employé**, je veux **soumettre une idée (...)** afin de partager une suggestion d'amélioration pour l'entreprise.*

Pour le Manager (REQ-02)

*En tant que **manager**, je veux **consulter la liste de toutes les idées (...)** afin d'avoir une vue d'ensemble des innovations proposées.*

Détailler une User Story : Critères d'Acceptation

- **Critère 1 : Scénario "Soumission réussie"**
 - ÉTANT DONNÉ QUE je suis connecté, LORSQUE je remplis le titre et la description et que je clique sur "Soumettre", ALORS mon idée apparaît en haut de la liste.

Détailler une User Story : Critères d'Acceptation

- **Critère 1 : Scénario "Soumission réussie"**

- ÉTANT DONNÉ QUE je suis connecté, LORSQUE je remplis le titre et la description et que je clique sur "Soumettre", ALORS mon idée apparaît en haut de la liste.

- **Critère 2 : Scénario "Champs obligatoires"**

- ÉTANT DONNÉ QUE je suis sur le formulaire, LORSQUE je clique sur "Soumettre" sans remplir le titre, ALORS un message d'erreur s'affiche et l'idée n'est pas envoyée.

Formalisation de "Soumettre une idée"

- **Cas d'Utilisation** : CU-01 - Soumettre une idée
- **Acteur Primaire** : Employé
- **Précondition** : L'Employé est authentifié.
- **Scénario Nominal (Happy Path)** :
 - 1 L'Employé clique sur « Proposer une idée ».
 - 2 Le Système affiche un formulaire.
 - 3 L'Employé saisit titre et description.
 - 4 ...

Synthèse : User Story vs. Cas d'Utilisation

| Caractéristique | User Story | Cas d'Utilisation |
|--------------------|--|---|
| Objectif Principal | Expliquer le POURQUOI (valeur métier) | Décrire le COMMENT (interaction) |
| Format | Court et informel (En tant que...) | Structuré et formel (document texte) |
| Niveau de détail | Léger, une "promesse de conversation" | Élevé et exhaustif |
| Force principale | Focus sur la valeur utilisateur | Réduit l'ambiguïté, anticipe les problèmes |
| Quand l'utiliser ? | Approche Agile (backlog, sprint) | Processus critiques, régléments |

Conclusion

- L'analyse est une démarche structurée pour **réduire les risques**.

Conclusion

- L'analyse est une démarche structurée pour **réduire les risques**.
- Elle commence par une **enquête** pour trouver le vrai besoin.

Conclusion

- L'analyse est une démarche structurée pour **réduire les risques**.
- Elle commence par une **enquête** pour trouver le vrai besoin.
- Elle se formalise via des outils centrés sur la **valeur** (User Stories) ou l'**exhaustivité** (Cas d'Utilisation).

Conclusion

- L'analyse est une démarche structurée pour **réduire les risques**.
- Elle commence par une **enquête** pour trouver le vrai besoin.
- Elle se formalise via des outils centrés sur la **valeur** (User Stories) ou l'**exhaustivité** (Cas d'Utilisation).
- Un temps bien investi en analyse garantit que l'effort de développement est dirigé vers **le bon objectif**.

Questions ?