

R3.03 - Flow Charts

TD 5

Diagrammes pour cahiers des charges

DUT info Vannes

G. Kerbellec - 2026

1- Se mettre d'accord avec un client

On pense souvent à tort comprendre les besoins du client. Sauf que nous ne partageons pas forcément les mêmes expériences et les mêmes contextes, ce qui produit de nombreuses possibilités d'incompréhensions, malentendus, frustrations.

<https://www.youtube.com/watch?v=FN2RM-CHkuI>

Sur un vrai contrat, les conséquences peuvent être dramatiques.

Pour éviter les drames, on essaie de se mettre d'accord en produisant des cahiers des charges. Il est essentiel de produire des cahiers des charges suffisamment clairs et attractifs, afin de créer une dynamique avec l'ensemble des acteurs du projet.

Il est essentiel de produire de nombreux schémas et des tableaux récapitulatifs.

De plus, il faut mettre à disposition de nombreux diagrammes permettant de situer :

- le QUOI (besoins fonctionnels)
- le COMMENT (solutions techniques)
- le QUAND (organisation)

2 - Diagrammes de besoins fonctionnels

Flow Charts

Diagrammes traduisant le déroulement/chemin de l'application

<https://drawio-app.com/flowcharts>

<https://drawio-app.com/flowcharts/#:~:text=Flowcharts%20in%20draw.io,to%20Flowcharts%2C%20then%20click%20Apply>

<https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>

norme BPMN simplifiée =>

- **Ovale (Terminator)** : Uniquement pour le **Début** et la **Fin**.
- **Rectangle (Process)** : Une **action** (ex: "L'utilisateur clique", "Le serveur calcule").
- **Losange (Décision)** : Une **question** qui ne peut avoir que deux sorties ou plus (ex: "Est-ce valide ?" -> OUI part à droite, NON part en bas).
- **Parallélogramme (Input/Output)** : Entrée ou sortie de **données** (ex: "Saisir email", "Afficher erreur").

Use Cases

répond à la question "qui a le droit de faire quoi ?"

<https://fr.venngage.com/blog/diagrammes-de-utilisation>

<https://drawio-app.com/uml-use-case-diagrams-with-draw-io>

Le Bonhomme (Acteur) : Ce n'est pas forcément un humain ! Ça peut être un logiciel externe (ex: "Système Bancaire" ou "API Google Maps").

L'Ovale (Cas d'utilisation) : C'est une **fonctionnalité** majeure. Le texte à l'intérieur doit toujours commencer par un **verbe à l'infinitif** (ex: "Créer un compte", "Payer le panier").

Le Rectangle (Frontière du système) : Tout ce qui est dans le rectangle est votre application. Les bonhommes (acteurs) doivent être **à l'extérieur** du rectangle.

Il y a deux types de flèches pointillées entre les ovales :

- **<<include>> (Obligatoire)** : Le cas A **a besoin** du cas B pour fonctionner.
 - *Exemple* : Pour "Faire un virement" (A), il faut obligatoirement "Vérifier le solde" (B).
 - *Moyen mnémotechnique* : C'est comme une sous-routine ou une fonction partagée.
- **<<extend>> (Optionnel)** : Le cas B **peut** se déclencher pendant le cas A, mais pas tout le temps. C'est une option ou un cas d'erreur.
 - *Exemple* : Pendant "Valider le panier" (A), on peut parfois "Saisir un code promo" (B). Le code promo n'est pas obligatoire pour valider le panier.

Pour faire un bon Use Case, suivez cette checklist :

1. **Identifiez les Acteurs** : Qui touche au système ? (Client, Admin, API Stripe...).
2. **Identifiez les "Gros Buts"** : Qu'est-ce qu'ils veulent accomplir ? (Acheter, Rembourser, Publier).
3. **Ne mettez PAS les détails** :
 - *Mauvais* : Ovale "Cliquer sur OK", Ovale "Saisir mot de passe". (C'est trop détaillé).
 - *Bon* : Ovale "S'authentifier".
4. **Tracez les lignes** : Reliez les Acteurs aux Ovales.

Diagrammes de Séquences

<https://drawio-app.com/examples>

3 - Diagrammes de solutions techniques

Site map (liste des pages et transitions)

Prototypes (figma, ...)

Diagrammes de classes

Schémas de BD

Diagrammes de déploiement, schémas de réseaux

4 - Organisation

Organigramme

Diagrammes de Gantt