



# TESTS FONCTIONNELS ÉLÉMENTAIRES

Analyse 3

2020-2021

## PLAN

### MCD

**Modèle conceptuel des données**

Diagramme de classes  
(rappels) Documentation

### MCT

**Modèle conceptuel des traitements**

Diagramme de Use Cases (UC)  
Documentation

Conceptuel

### UC Specification

Documentation de UC  
Interface utilisateur  
Diagramme d'activité (rappel)

### PTFE

**Plan de tests fonctionnels élémentaires**  
Documentation

Fonctionnel

### MTD-MTT

### UC Realization

Diagramme de séquence  
Diagramme de classes techniques

### Design Pattern

Technique

## Méthodes

- QUALITÉ DU SI (Rappels)

- «Ensemble des caractéristiques d'une entité qui lui confèrent l'aptitude à satisfaire des besoins exprimés ou implicites », ISO 8402



## ● Quelques bugs célèbres

- - la Lexus ES350
  - Le vol inaugural d'Ariane 5
  - Le bug de l'an 2000
  - ...

[wikipédia](#)

## ● QUALITÉ DU SI (Rappels)

### ● Critères d'évaluation (par l'utilisateur)

- Pertinence (demandes de l'utilisateur)
- Apport de bénéfices
- Fiabilité
- Performances
- Convivialité

Critères techniques permettant d'atteindre cette qualité:

- Évolutivité
- Réutilisabilité
- Portabilité



## ● Moyen d'assurer la qualité

### ● Qualité fonctionnelle

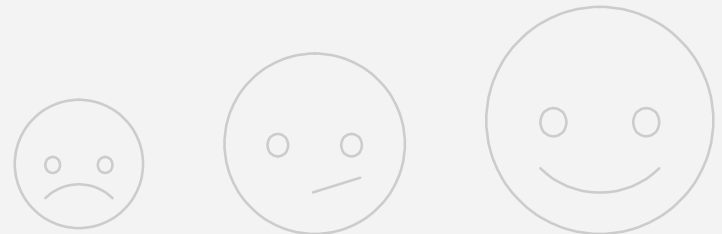
- Validation des spécifications
  - Description des UCs, “storyboards”, maquettes, ...
  - Par les utilisateurs
  - Avant le développement
  - Définit un cahier des charges
  - Définit le niveau de qualité
    - Performance
    - Fiabilité
    - Pertinence
    - Convivialité



## ● Moyen d'assurer la qualité

### ● Qualité fonctionnelle

- Vérifications des qualités après développement
  - Tout le système ou une partie
  - A l'aide de tests
    - Faire tourner le SIA
    - En simulant des conditions réelles
    - Confronter les résultats à ceux attendus



## ● Moyen d'assurer la qualité

### ● 2 types de tests

- White box (boîte blanche, ouverte)
- Black box (boîte noire, fermée)





## ● Moyen d'assurer la qualité

### ● 2 types de tests

- White box (boîte blanche, ouverte)
  - On a accès au code source
  - On vérifie le code
    - lecture
    - vérifications automatiques
      - erreurs courantes
      - toutes les instructions accessibles
      - ...
  - Géré par les informaticiens



## ● Moyen d'assurer la qualité

### ● 2 types de tests

- Black box (boîte noire, fermée)
  - On n'a pas accès au code source
  - On définit
    - des scénarios
    - des jeux de données en entrée
    - les résultats attendus pour chaque scénario / jeu de données
  - On teste chaque scénario sur le logiciel développé
  - Peut être un métier à part entière

## ● Moyen d'assurer la qualité

### ● 2 types de tests

- Black box (boîte noire, fermée)
  - On compare
    - les résultats obtenus
    - les résultats attendus
  - Si le test est OK : c'est validé !
  - Si le test est KO : il faut corriger !

## ● Que tester ?

- - Tests fonctionnels
    - Teste la pertinence (par rapport aux besoins)
    - Teste la fiabilité (résultats exacts, pas de pertes de données, ...)
  - Tests de convivialité (ergonomie)
    - look (statique)
    - feel (dynamique)

## ● Que tester ?

- - Tests de performance
    - Volumes de données, charges
    - En fonction de l'architecture, de la configuration
  - Tests de robustesse
    - Charges extrêmes
    - Pannes équipement: test de reprise (récupération données, reprise traitements)
    - Résistance à l'intrusion (sécurité)

## ● Que tester en analyse ?

### ● ◦ Tests fonctionnels

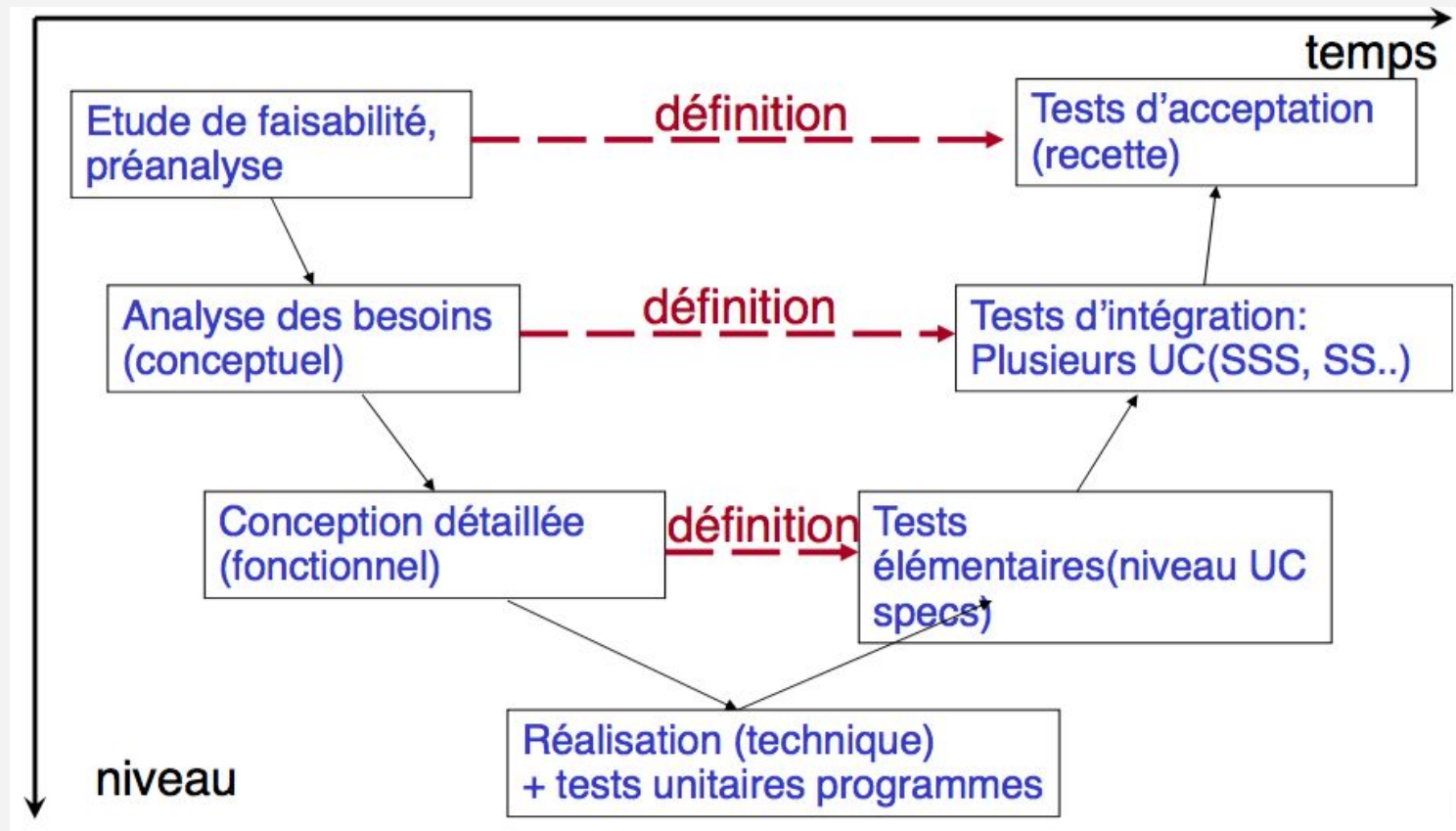
- Décrits à partir du MCT et UC Specification
- Permettent de vérifier que le logiciel
  - fait ce qu'on en attend
  - empêche de faire ce qu'on ne peut pas faire (messages d'erreur)

## ● Que tester en analyse ?

- - Différents niveaux
    - Tests unitaires
    - Tests élémentaires
    - Tests d'intégration
    - Tests de système

## Que tester en analyse ?

- Différents niveaux : diagramme en V





## ● Que tester en analyse ?

- - Différents niveaux : diagramme en V
    - Montre le parallèle entre la décomposition de l'analyse et l'assemblage des éléments testés pour réaliser le logiciel complet
    - Les plans de tests définiront le contenu des tests et leur déroulement dans le temps

## ● Scénarios et données de tests

- - Impossibilité d'être exhaustif dans les tests
    - Trop de combinaisons
    - Importance de la définition des scénarios
    - Vérification partielle à partir d'un échantillon de données (jeux de tests) correspondant à ces scénarios

## ● Tests fonctionnels élémentaires

- - On teste un UC
  - Un scénario
    - = un flux du UC
    - = un chemin dans le diagramme d'activité
      - **depuis le début**
      - **jusqu'à une fin**

## ● Tests fonctionnels élémentaires

- - Types de tests
    - Tests sur les saisies (input)
      - Vérification des messages d'erreur
    - Tests sur les calculs
      - Données en entrées valides
        - Cas normaux et limites
      - Vérifier les résultats des calculs

## ● Tests fonctionnels élémentaires

- - Types de tests
    - Tests sur les mises à jour du système
      - Vérifier que les contraintes d'intégrité du système soient respectées
        - ex: supprimer le responsable d'un lecteur mineur
    - Tests sur les outputs
      - Mise en page
        - pagination, mise en page, ...
        - possibilité de navigation

## ● Limites des tests

- Un bon test
  - détecte les erreurs
  - permet de les corriger

! Un logiciel qui a réussi tous les tests n'est pas nécessairement sans erreur !

## ● Mise en oeuvre des tests

- Le test est un processus “destructif”
  - On cherche des erreurs
    - alors qu’on construit lorsqu’on programme
  - Les tests peuvent coûter de 30 à 60% du coût d’un logiciel
    - le coût de certaines erreurs est encore bien plus important...

- Définition des tests fonctionnels élémentaires

- Pour définir les tests, nous allons utiliser le gabarit suivant :

POUR CHAQUE DIAGRAMME D'ACTIVITÉ

Les tests fonctionnels élémentaires seront déduits du diagramme d'activité. Nous les décrirons, pour plus de simplicité, dans la documentation du diagramme d'activité.



## Définition des tests fonctionnels élémentaires

La description devra suivre la logique suivante :

- Numéroté séquentiellement chaque action et alternative dans le diagramme d'activité
- pour chaque scénario :
  - donner lui un numéro de séquence, une description courte et le chemin complet (ou explicite) numéroté
  - pour chaque cas possible de ce scénario, donner:
    - une description courte
    - les résultats attendus
    - si la description d'un cas et de ces résultats attendus peut en recouvrir plusieurs (avec des valeurs différentes), l'indiquer
      - en suffixant son numéro de ".X" (ex: Cas 2.x)
      - en suffixant sa description d'une phrase telle que "chaque cas reprendra une combinaison différentes des valeurs de la liste "XXX"" où XXX est une liste de valeurs qui sera définie plus loin.
  - définir avec précision les valeurs correctes et incorrectes des cas décrits

*Pour les cycles d'un chemin du diagramme, faites un scénario séparé en indiquant le chemin répété (par ex. [...-5-8-9-5-...]). Ne pas inclure alors ce cycle dans la description des autres scénarios.*

La syntaxe à respecter est la suivante:

## **# Tests élémentaires fonctionnels**

### **# Scénarios**

#### **## Scénario 1**

[Description du scénario] [[chemin défini par une séquence de numéros séparé par “-”]]

1. Cas 1[x] : [Description du premier cas du scénario]

- Résultats attendus : [Description des résultats attendus]

2. ...

#### **## Scénario 2**

...

#### **## Valeurs correctes et limites:**

- [attribut] :

- [description d'un ensemble de valeurs correctes/limites pour cet attribut]

- ...

#### **## Valeurs erronées**

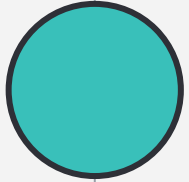
- [attribut] :

- [description d'un ensemble de valeurs erronées pour cet attribut]

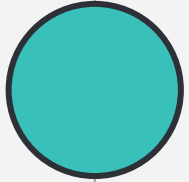
- ...

## ● Tests fonctionnels élémentaires

- On teste un UC
- Un scénario
  - = un flux du UC
  - = un chemin dans le diagramme d'activité
    - depuis le début
    - jusqu'à une fin
- **Pour chaque scénario => au moins un cas**



## Démo MédiCab: PTFE



# Exercices



## Exercice 1

Voici un scénario d'un UC quelconque.

### Scénario 1

Cas 1.X: Combinaison des valeurs de la liste "Liste"

Résultat attendu : *un résultat R1*

Cas 2 : Saisie des valeurs *A* pour l'entrée 1 et *B* pour l'entrée 2

Résultat attendu : *un résultat R2 (autre que R1)*

### Liste

- Entrée 1
  - *Valeur 1*
- Entrée 2
  - *Valeur 2.1*
  - *Valeur 2.2*

Combien y a-t-il d'exécution à faire pour exécuter les tests du scénario 1?  
Justifiez votre réponse en donnant les entrées de chaque exécution.



## Exercice 2



Voici le diagramme d'activités d'un distributeur de billets.

1. Combien y a-t-il de scénarios?
2. Lister les données en entrée et en sortie, les calculs, les tests et les mises à jour.
3. Lister pour chaque élément de la liste précédente les valeurs erronées, limites et correctes.

## Tests fonctionnels élémentaires

