Introduction aux tests du logiciel

F.X. Fornari
xavier.fornari@esterel-technologies.com
P. Manoury
pascal.manoury@pps.jussieu.fr

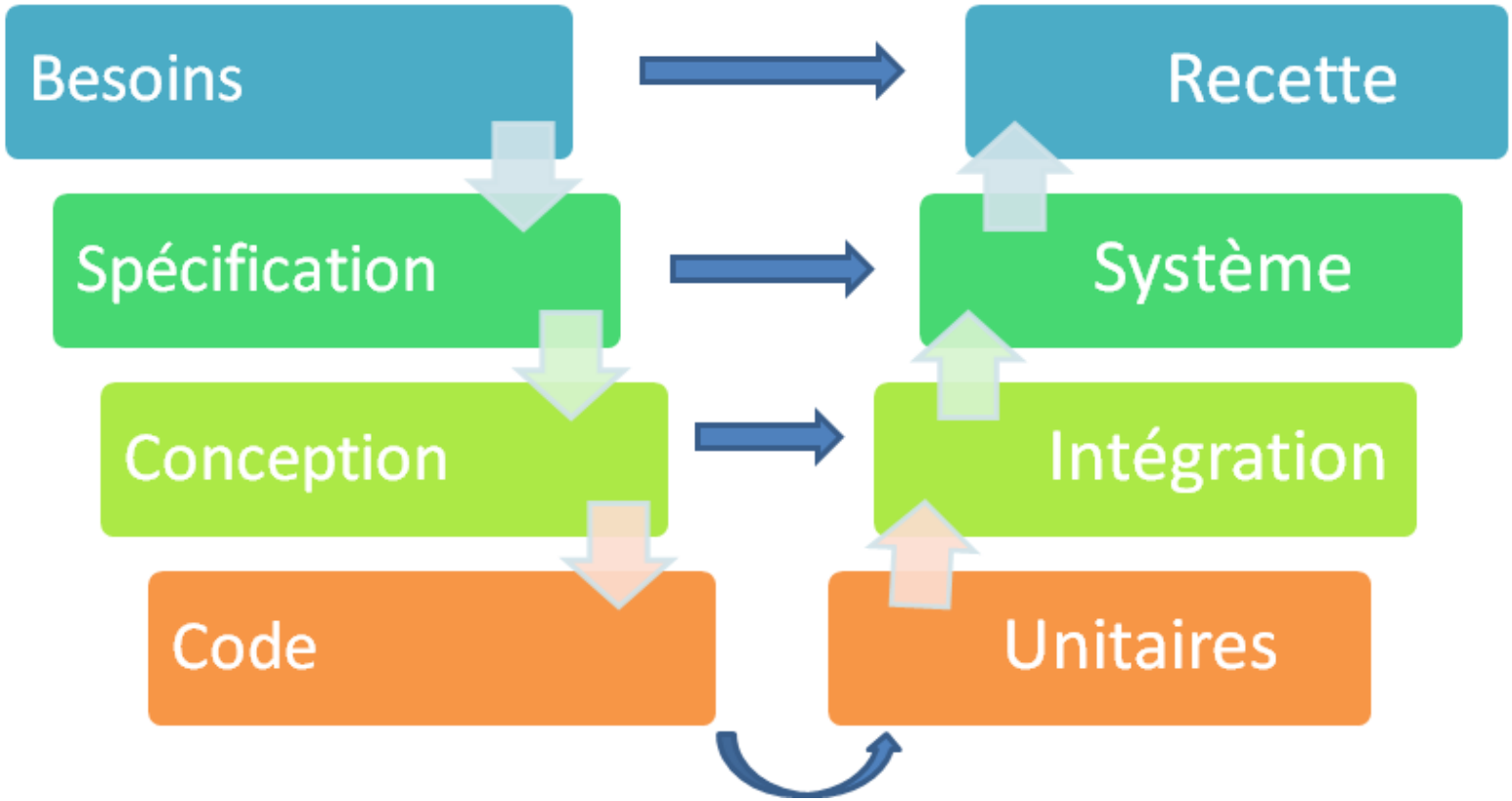
2013

Introduction

- Les cours précédants ont abordés les techniques de tests
- Ces techniques s'appliquent lors de différentes phases
- L'objectif du cours est de montrer comment *formaliser* les approches.

Quand commencer ?

Le test commence de suite ! Ici, le cycle en V. mais aussi approches incrémentale, agiles, ...



Les différents niveaux

- Tests de recette: test de réception du logiciel chez le client final

Les différents niveaux

- Tests de recette: test de réception du logiciel chez le client final
- Tests intégration système: test de l'intégration du logiciel avec d'autres logiciels

Les différents niveaux

- Tests de recette: test de réception du logiciel chez le client final
- Tests intégration système: test de l'intégration du logiciel avec d'autres logiciels
- Tests systeme: test d'acception du logiciel avant livraison (nouvelle version par exemple)

Les différents niveaux

- Tests de recette: test de réception du logiciel chez le client final
- Tests intégration système: test de l'intégration du logiciel avec d'autres logiciels
- Tests systeme: test d'acception du logiciel avant livraison (nouvelle version par exemple)
- Tests Integration: test de l'intégration des différents composants (avec ou sans hardware)

Les différents niveaux

- Tests de recette: test de réception du logiciel chez le client final
- Tests intégration système: test de l'intégration du logiciel avec d'autres logiciels
- Tests systeme: test d'acception du logiciel avant livraison (nouvelle version par exemple)
- Tests Integration: test de l'intégration des différents composants (avec ou sans hardware)
- Tests Unitaires: tests élémentaires des composants logiciels (une fonction, un module, ...)

Les différents niveaux

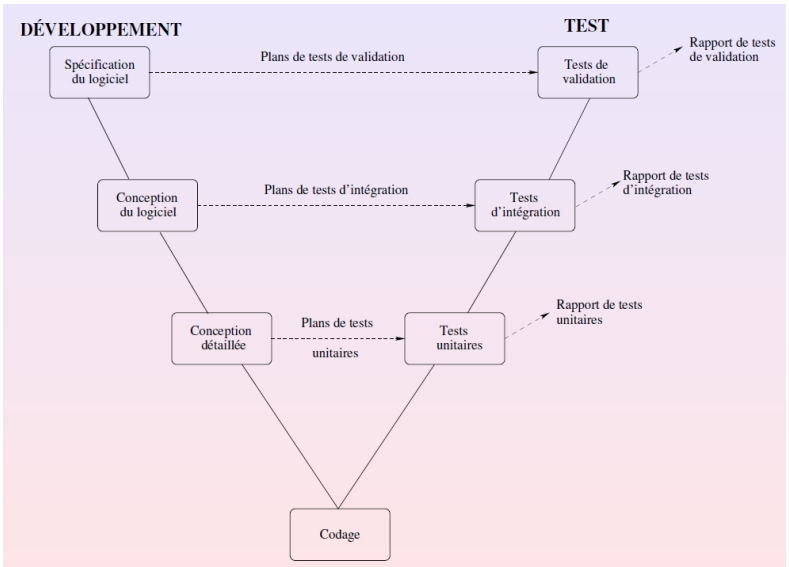
- Tests de recette: test de réception du logiciel chez le client final
- Tests intégration système: test de l'intégration du logiciel avec d'autres logiciels
- Tests systeme: test d'acception du logiciel avant livraison (nouvelle version par exemple)
- Tests Integration: test de l'intégration des différents composants (avec ou sans hardware)
- Tests Unitaires: tests élémentaires des composants logiciels (une fonction, un module, ...)
- Tests de non-régression

Les différents niveaux

- Tests de recette: test de réception du logiciel chez le client final
- Tests intégration système: test de l'intégration du logiciel avec d'autres logiciels
- Tests systeme: test d'acception du logiciel avant livraison (nouvelle version par exemple)
- Tests Integration: test de l'intégration des différents composants (avec ou sans hardware)
- Tests Unitaires: tests élémentaires des composants logiciels (une fonction, un module, ...)
- Tests de non-régression
- "Smoke-tests": jeu réduit de tests pour valider un *build* avant de passer à la validation

Les Phases de tests

- Le travail de testeur ne commence pas *après* la phase de codage, mais *en même temps*, et se poursuit en parallèle
- Afin d'éviter le "Big-Bang" à l'intégration, les phases de tests valident le logiciel depuis le module jusqu'au logiciel complet.



Les Tests Unitaires (TU]

But Validation de la conformité de chaque composant logiciel pris unitairement par rapport à sa spécification détaillée.

Les Tests Unitaires (TU]

But Validation de la conformité de chaque composant logiciel pris unitairement par rapport à sa spécification détaillée.

Quand ? Dès qu'une pièce de code a été codée et compilée correctement

Les Tests Unitaires (TU]

But Validation de la conformité de chaque composant logiciel pris unitairement par rapport à sa spécification détaillée.

Quand ? Dès qu'une pièce de code a été codée et compilée correctement

Type de tests structurels

Les Tests Unitaires (TU]

But	Validation de la conformité de chaque composant logiciel pris unitairement par rapport à sa spécification détaillée.
Quand ?	Dès qu'une pièce de code a été codée et compilée correctement
Type de tests	structurels
Comment ?	sur machine hôte, généralement sans banc de tests

Les Tests Unitaires (TU]

But	Validation de la conformité de chaque composant logiciel pris unitairement par rapport à sa spécification détaillée.
Quand ?	Dès qu'une pièce de code a été codée et compilée correctement
Type de tests	structurels
Comment ?	sur machine hôte, généralement sans banc de tests
Qui ?	Pour les logiciels de faible criticité, elle peut être réalisée par l'équipe de développement (mais pas par le développeur ayant codé la pièce de code).

Les Tests d'Intégration (TI)

But Validation des sous-systèmes logiciels entre eux

- Tests d'Intégration Logiciel/Logiciel (interface entre composants logiciels)
- Tests d'Intégration Logiciel/Matériel (interface entre le logiciel et le matériel)

Les Tests d'Intégration (TI)

But Validation des sous-systèmes logiciels entre eux

- Tests d'Intégration Logiciel/Logiciel (interface entre composants logiciels)
- Tests d'Intégration Logiciel/Matériel (interface entre le logiciel et le matériel)

Quand ? Dès qu'un sous-système fonctionnel (module, objet) est entièrement testé unitairement

Les Tests d'Intégration (TI)

But Validation des sous-systèmes logiciels entre eux

- Tests d'Intégration Logiciel/Logiciel (interface entre composants logiciels)
- Tests d'Intégration Logiciel/Matériel (interface entre le logiciel et le matériel)

Quand ? Dès qu'un sous-système fonctionnel (module, objet) est entièrement testé unitairement

Type de tests des interfaces

Les Tests d'Intégration (TI)

But Validation des sous-systèmes logiciels entre eux

- Tests d'Intégration Logiciel/Logiciel (interface entre composants logiciels)
- Tests d'Intégration Logiciel/Matériel (interface entre le logiciel et le matériel)

Quand ? Dès qu'un sous-système fonctionnel (module, objet) est entièrement testé unitairement

Type de tests des interfaces

Comment ? logiciel/logiciel généralement sur machine hôte
logiciel/matériel sur machine cible avec un banc de test minimal (simulation des entrées, acquisition des sorties).

Les Tests d'Intégration (TI)

But Validation des sous-systèmes logiciels entre eux

- Tests d'Intégration Logiciel/Logiciel (interface entre composants logiciels)
- Tests d'Intégration Logiciel/Matériel (interface entre le logiciel et le matériel)

Quand ? Dès qu'un sous-système fonctionnel (module, objet) est entièrement testé unitairement

Type de tests des interfaces

Comment ? logiciel/logiciel généralement sur machine hôte logiciel/matériel sur machine cible avec un banc de test minimal (simulation des entrées, acquisition des sorties).

Qui ? toujours par une équipe de tests indépendante de l'équipe de développement.

Les Tests de Validation (TV)

But Vérifier la conformité du logiciel à la spécification du logiciel

Les Tests de Validation (TV)

But Vérifier la conformité du logiciel à la spécification du logiciel

Quand ? Dès que l'ensemble des sous-systèmes fonctionnels ont été testé et intégré

Les Tests de Validation (TV)

But Vérifier la conformité du logiciel à la spécification du logiciel

Quand ? Dès que l'ensemble des sous-systèmes fonctionnels ont été testé et intégré

Type de tests fonctionnels et de robustesse

Les Tests de Validation (TV)

But	Vérifier la conformité du logiciel à la spécification du logiciel
Quand ?	Dès que l'ensemble des sous-systèmes fonctionnels ont été testé et intégré
Type de tests	fonctionnels et de robustesse
Comment ?	toujours réalisés sur machine cible et nécessitent généralement la fabrication d'un banc de tests élaboré

Les Tests de Validation (TV)

But Vérifier la conformité du logiciel à la spécification du logiciel

Quand ? Dès que l'ensemble des sous-systèmes fonctionnels ont été testé et intégré

Type de tests fonctionnels et de robustesse

Comment ? toujours réalisés sur machine cible et nécessitent généralement la fabrication d'un banc de tests élaboré

Qui ? Ces tests sont toujours réalisés par une équipe de tests indépendante de l'équipe de développement.

Autres Tests dans le cycle

On peut avoir les tests suivants:

- Test d'Intégration Système:
  - Ces tests permettent d'assurer la compatibilité du logiciel avec d'autres
  - Par exemple: un logiciel de gestion interopérant avec d'autres logiciels chez un client

Autres Tests dans le cycle

On peut avoir les tests suivants:

- Test d'Intégration Système:
  - Ces tests permettent d'assurer la compatibilité du logiciel avec d'autres
  - Par exemple: un logiciel de gestion interopérant avec d'autres logiciels chez un client
- Test de recette:
  - Test servant à assurer la conformité et la confiance avant livraison finale.
  - Se décompose parfois entre *test de recette utilisateur*, impliquant les utilisateurs finaux, et *test d'exploitation*, impliquant le service informatique client.

Autres Tests dans le cycle

On peut avoir les tests suivants:

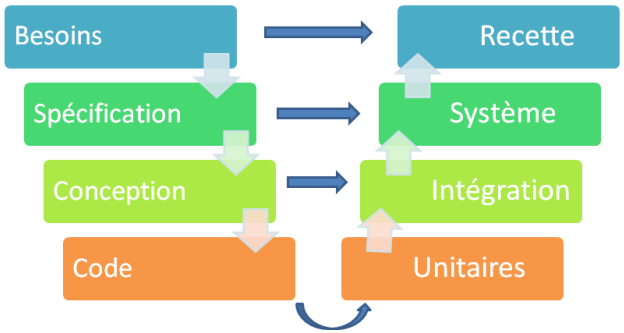
- Test d'Intégration Système:
  - Ces tests permettent d'assurer la compatibilité du logiciel avec d'autres
  - Par exemple: un logiciel de gestion interopérant avec d'autres logiciels chez un client
- Test de recette:
  - Test servant à assurer la conformité et la confiance avant livraison finale.
  - Se décompose parfois entre *test de recette utilisateur*, impliquant les utilisateurs finaux, et *test d'exploitation*, impliquant le service informatique client.
- Test de non-régression
  - Assure la non-régression entre deux versions du logiciel, pour les mêmes fonctionnalités
  - Est utilisé au cours du cycle de développement, si celui-ci est fait par incréments

La planification

- Ces différents niveaux de tests doivent être planifiés (c'est l'objet du plan projet).

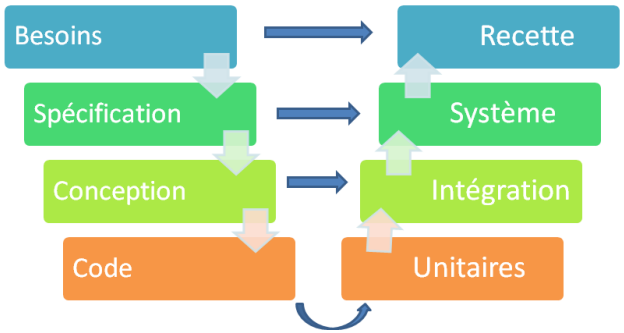
La planification

- Ces différents niveaux de tests doivent être planifiés (c'est l'objet du plan projet).
- Théoriquement, on prépare les tests en même temps que le développement correspondant au niveau.



La planification

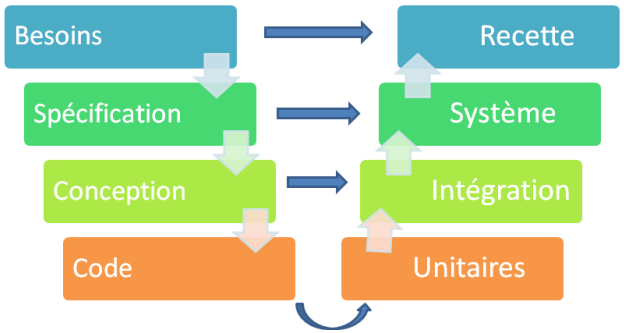
- Ces différents niveaux de tests doivent être planifiés (c'est l'objet du plan projet).
- Théoriquement, on prépare les tests en même temps que le développement correspondant au niveau.



- On peut aussi le faire incrémentalement, ou en *pipeline*. Demande une gestion de projet très fine.

La planification

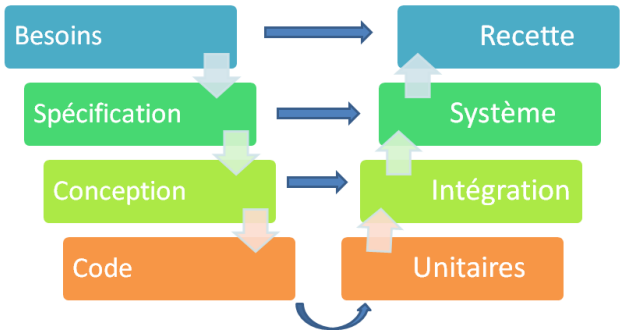
- Ces différents niveaux de tests doivent être planifiés (c'est l'objet du plan projet).
- Théoriquement, on prépare les tests en même temps que le développement correspondant au niveau.



- On peut aussi le faire incrémentalement, ou en *pipeline*. Demande une gestion de projet très fine.
- En Extreme-Programmming: paire de développeurs/paires de testeurs.

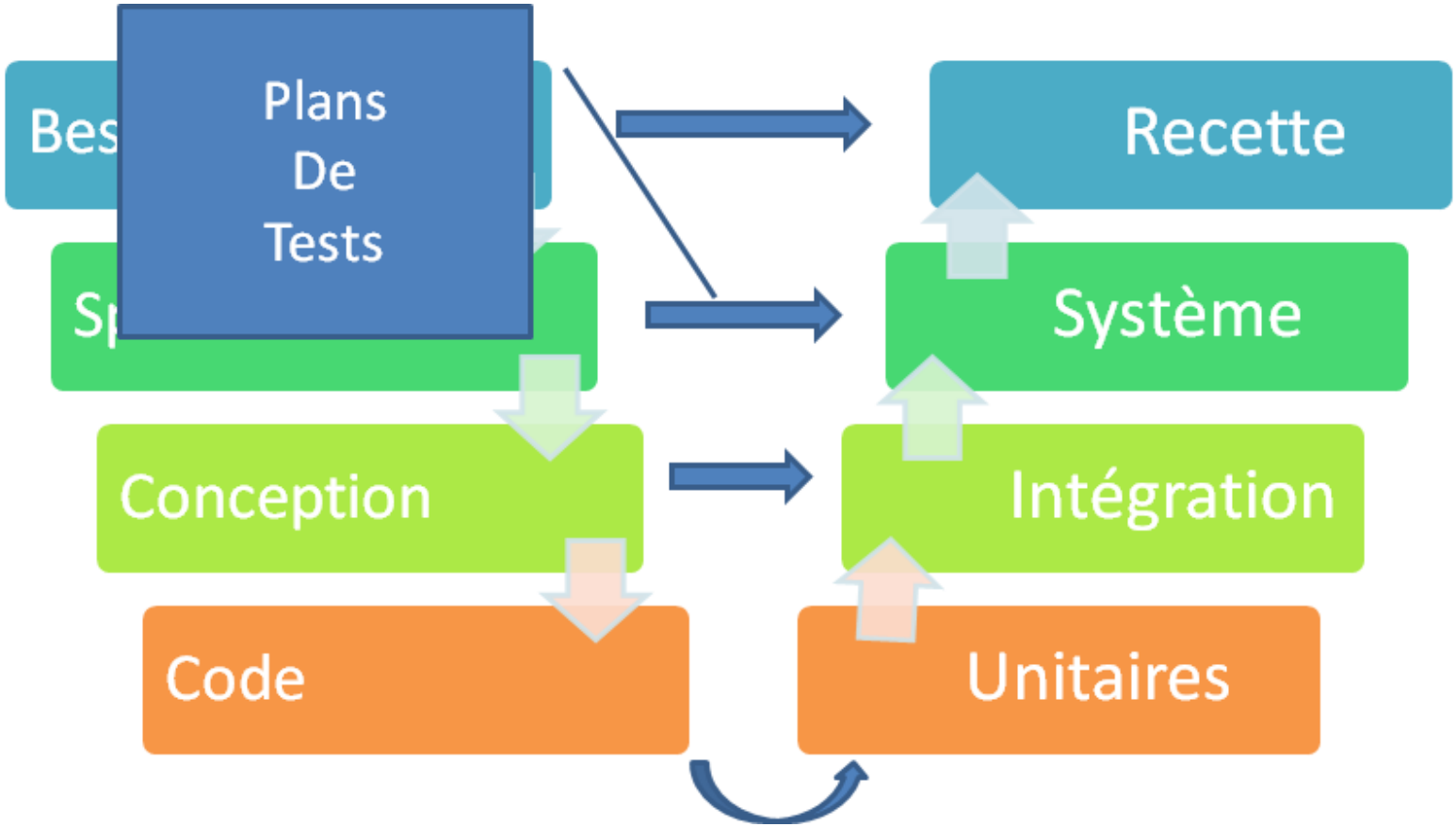
La planification

- Ces différents niveaux de tests doivent être planifiés (c'est l'objet du plan projet).
- Théoriquement, on prépare les tests en même temps que le développement correspondant au niveau.



- On peut aussi le faire incrémentalement, ou en *pipeline*. Demande une gestion de projet très fine.
- En Extreme-Programmming: paire de développeurs/paires de testeurs.
- De toute façon, il y a rebouclage permanent (bugs, changements de specs): il faut s'adapter !

Plans de tests



Travail en phase de descente

Durant les phases de descente du cycle, le testeur élabore les *Plans de Tests*, et fabrique les bancs de tests.

Les plans de tests décrivent essentiellement:

- La stratégie de tests mise en place
- Les moyens mis en œuvre (matériel, logiciel, et humain)
- L'ensemble des fiches de tests

Plan de Tests

C'est la description globale de la mise en œuvre. (fiche)

1. Introduction
   - Contexte, Organisation, Réferences

Plan de Tests

C'est la description globale de la mise en œuvre. (fiche)

1. Introduction
   - Contexte, Organisation, Réferences
2. Objectifs et Obligations
   - Quels sont les objectifs du test
   - Quels sont les contraintes (délai, régression, ...)

Plan de Tests

C'est la description globale de la mise en œuvre. (fiche)

1. Introduction
   - Contexte, Organisation, Réferences
2. Objectifs et Obligations
   - Quels sont les objectifs du test
   - Quels sont les contraintes (délai, régression, ...)
3. Risques et dépendances
   - relevés des risques (code non livré, ...), liens avec d'autres outils

Plan de Tests

C'est la description globale de la mise en œuvre. (fiche)

1. Introduction
   - Contexte, Organisation, Réferences
2. Objectifs et Obligations
   - Quels sont les objectifs du test
   - Quels sont les contraintes (délai, régression, ...)
3. Risques et dépendances
   - relevés des risques (code non livré, ...), liens avec d'autres outils
4. Hypothèses et exclusions

Plan de Tests

C'est la description globale de la mise en œuvre. (fiche)

1. Introduction
   - Contexte, Organisation, Réferences
2. Objectifs et Obligations
   - Quels sont les objectifs du test
   - Quels sont les contraintes (délai, régression, ...)
3. Risques et dépendances
   - relevés des risques (code non livré, ...), liens avec d'autres outils
4. Hypothèses et exclusions
5. Critères de départ et d'arrêt
   - départ: définit les éléments nécessaires pour tester
   - arrêt : définition de critères (taux d'erreurs, revues, ...)

Plan de Tests

C'est la description globale de la mise en œuvre. (fiche)

1. Introduction
   - Contexte, Organisation, Réferences
2. Objectifs et Obligations
   - Quels sont les objectifs du test
   - Quels sont les contraintes (délai, régression, ...)
3. Risques et dépendances
   - relevés des risques (code non livré, ...), liens avec d'autres outils
4. Hypothèses et exclusions
5. Critères de départ et d'arrêt
   - départ: définit les éléments nécessaires pour tester
   - arrêt : définition de critères (taux d'erreurs, revues, ...)
6. Contrôle du projet
   - Qui fait quoi
   - Besoins (formations, etc...)
   - Communication des problèmes, Rapports d'avancement

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

1. Introduction (fiche)

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

1. Introduction (fiche)
2. Exigences

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

1. Introduction (fiche)
2. Exigences
   - 2.1 Politique de test: vue d'ensemble, domaines, catégorisation des tests, ..

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

1. Introduction (fiche)
2. Exigences
   - 2.1 Politique de test: vue d'ensemble, domaines, catégorisation des tests, ..
   - 2.2 Environnement du test

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

1. Introduction (fiche)
2. Exigences
   - 2.1 Politique de test: vue d'ensemble, domaines, catégorisation des tests, ..
   - 2.2 Environnement du test
   - 2.3 Attributions des participants

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

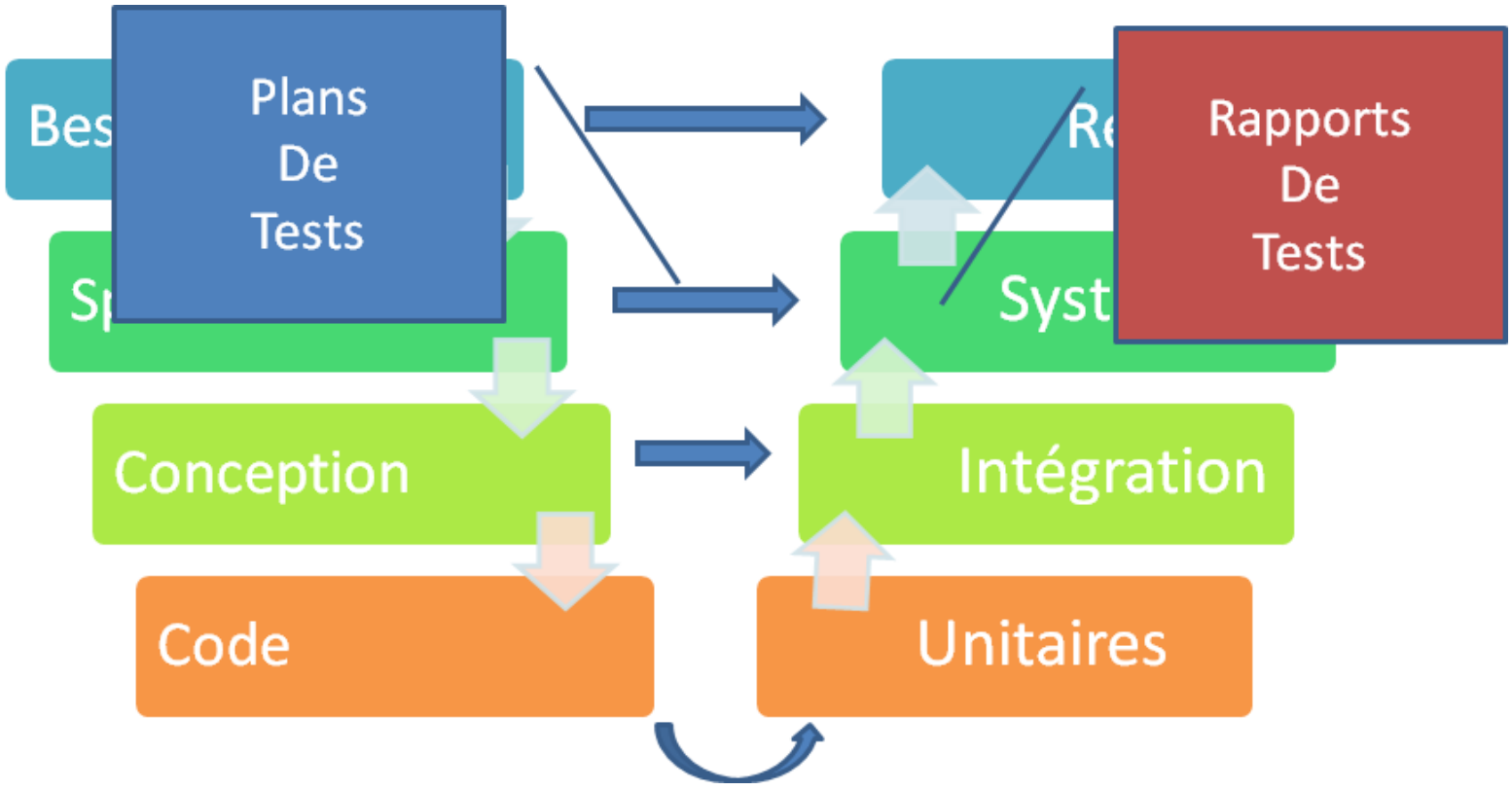
1. Introduction (fiche)
2. Exigences
   - 2.1 Politique de test: vue d'ensemble, domaines, catégorisation des tests, ..
   - 2.2 Environnement du test
   - 2.3 Attributions des participants
   - 2.4 Caractéristique du test
     - les scénarios, enregistrement des résultats, acceptance,
     - documentation des tests

Spécification de test

La spécification de test détaille la réalisation de chaque test/catégories de tests

1. Introduction (fiche)
2. Exigences
   - 2.1 Politique de test: vue d'ensemble, domaines, catégorisation des tests, ..
   - 2.2 Environnement du test
   - 2.3 Attributions des participants
   - 2.4 Caractéristique du test
     - les scénarios, enregistrement des résultats, acceptance,
     - documentation des tests
3. Mode de test
   - 3.1 Activités préliminaires
   - 3.2 Réalisation du test
   - 3.3 Scénario de test (fiche)

Plans de tests



Travail en phase de remontée

Le testeur exécute les fiches de tests décrites dans les plans et produit les rapports de tests associés.
Ces rapports contiennent essentiellement:

- la synthèse des résultats de tests
- les résultats de tests détaillés
- la trace d'exécution des tests
- Exemple:
  - Modèle de résultats (fiche)
  - Journal de tests (fiche)

Rapport de test

C'est l'élément final après une "campagne" de tests

1. Introduction (fiche)

Rapport de test

C'est l'élément final après une "campagne" de tests

1. Introduction (fiche)
2. Vue d'ensemble
   - descriptions des activités

Rapport de test

C'est l'élément final après une "campagne" de tests

1. Introduction (fiche)
2. Vue d'ensemble
   - descriptions des activités
3. Ecarts
   - s'il y a des points particuliers, les mettre en exergue

Rapport de test

C'est l'élément final après une "campagne" de tests

1. Introduction (fiche)
2. Vue d'ensemble
   - descriptions des activités
3. Ecarts
   - s'il y a des points particuliers, les mettre en exergue
4. Analyse
   - détail de la réalisation des tests, des problèmes rencontrés éventuellement
   - détail de la couverture de code si besoin

Rapport de test

C'est l'élément final après une "campagne" de tests

1. Introduction (fiche)
2. Vue d'ensemble
   - descriptions des activités
3. Ecarts
   - s'il y a des points particuliers, les mettre en exergue
4. Analyse
   - détail de la réalisation des tests, des problèmes rencontrés éventuellement
   - détail de la couverture de code si besoin
5. Résultats
   - résumé des résultats

Rapport de test

C'est l'élément final après une "campagne" de tests

1. Introduction (fiche)
2. Vue d'ensemble
   - descriptions des activités
3. Ecarts
   - s'il y a des points particuliers, les mettre en exergue
4. Analyse
   - détail de la réalisation des tests, des problèmes rencontrés éventuellement
   - détail de la couverture de code si besoin
5. Résultats
   - résumé des résultats
6. Résumé des activités
   - résumé synthétique des résultats mais aussi des activités et de leurs déroulements

La gestion du processus de tests

- Tester un logiciel est un activité à part entière
- Cette activité doit être en constante amélioration
- Il faut donc pouvoir mesurer l'activité
- Il faut avoir des éléments de mesure $\Rightarrow$ il faut des données **objectives**
- ces données doivent être collectées pour être utilisées lors du *Post-Mortem*

Exemple d'analyse

- Processus de développement et de tests
  - Est-ce que le logiciel et les tests ont été bien faits ? (fiche)
- Mise en œuvre du processus de test (fiche)

Environnement

- De même l'environnement de test doit être analysé

Environnement

- De même l'environnement de test doit être analysé
- En amont, pour sa mise en place

Environnement

- De même l'environnement de test doit être analysé
- En amont, pour sa mise en place
- En aval, pour déterminer s'il a répondu aux demandes, et s'il faut l'améliorer

Environnement

- De même l'environnement de test doit être analysé
- En amont, pour sa mise en place
- En aval, pour déterminer s'il a répondu aux demandes, et s'il faut l'améliorer
- C'est le cas par exemple pour un outil (pour automatiser les tâches, ou faire de l'analyse). Exemple:
  - L'adéquation aux besoins (fiche)
  - Ses caractéristiques (fiche)
  - Son usage (fiche)
  - Son accessibilité (fiche)

Conclusion

- La mise en œuvre de tests demande de la rigueur

Conclusion

- La mise en œuvre de tests demande de la rigueur
- Il faut définir un cadre de réalisation
  - méthodes, objectifs, technologies, resources, ...
  - Ce cadre doit être connu de tous

Conclusion

- La mise en œuvre de tests demande de la rigueur
- Il faut définir un cadre de réalisation
  - méthodes, objectifs, technologies, resources, ...
  - Ce cadre doit être connu de tous
- Les tests font partie du produit, l'accompagne durant sa vie
  - ils évoluent avec lui
  - ils sont aussi un produit de la spécification

Conclusion

- La mise en œuvre de tests demande de la rigueur
- Il faut définir un cadre de réalisation
  - méthodes, objectifs, technologies, resources, ...
  - Ce cadre doit être connu de tous
- Les tests font partie du produit, l'accompagne durant sa vie
  - ils évoluent avec lui
  - ils sont aussi un produit de la spécification
- Le *reporting* est important
  - il donne la mesure de la confiance
  - il doit être lui-même vérifiable !