

Mise en œuvre du test fonctionnel

Fabrice AMBERT – fabrice.ambert@femto-st.fr
Fabrice BOUQUET – fabrice.bouquet@femto-st.fr
Fabien PEUREUX – fabien.peureux@femto-st.fr
Ivan ENDERLIN, Jean-Marie GAUTHIER
Cédric JOFFROY, Alexandre VERNOTTE

Le test fonctionnel



- Le test fonctionnel vise à examiner le comportement fonctionnel du logiciel et sa conformité avec la spécification du logiciel.
- Méthode :
 - Identification des caractéristiques fonctionnelles (**exigences**)
 - Identification des points de contrôle et d'observation
 - Pour chaque exigence, définition des cas de test sous la forme de scénarios d'un point de vue utilisateur
- Guider le développement du logiciel (Acceptance Testing Driven Development)
- Comprendre, garantir et démontrer l'adéquation du besoin et de la solution proposée
- Documenter la solution avec des utilisations nominales et/ou particulières
- Maîtriser la non régression suite aux maintenances correctives et/ou évolutives

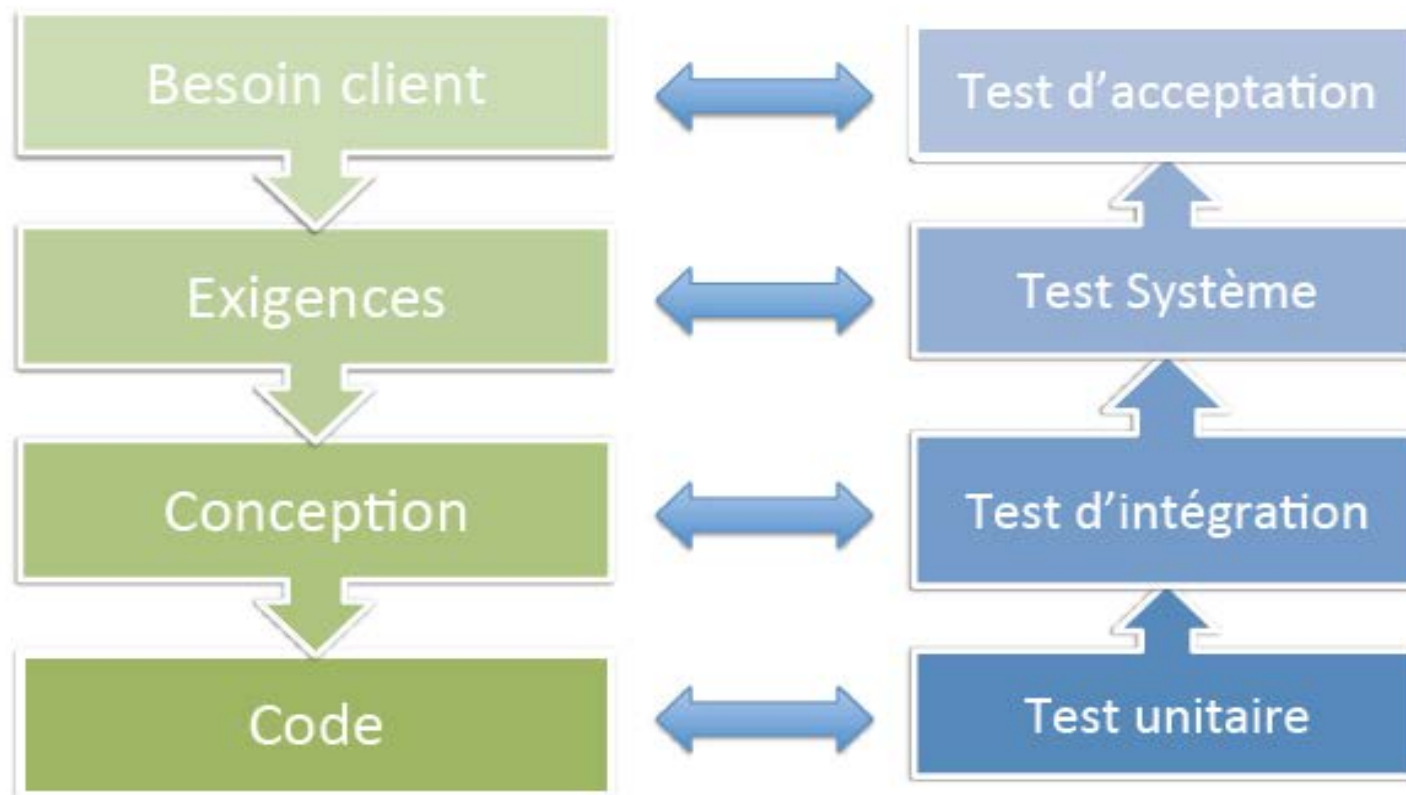
Exigences



- Une exigence est une contrainte exprimée par le client sur telle ou telle caractéristique du futur système
- Dans l'ingénierie système/logiciel, une exigence est généralement la description de ce que le système/logiciel livré doit être capable de faire (exigence fonctionnelle). On trouve aussi :
 - Exigences non fonctionnelles, portant sur la manière dont le système/logiciel doit exécuter ses fonctions
 - Exigences de sécurité,
 - Exigences de performance,
 - Exigences de qualité de service...
- A ces exigences « produit » s'ajoutent également les exigences « métier » (qui décrivent le « quoi ») et les exigences processus (qui décrivent le « comment », c'est à dire les contraintes).

Pourquoi des exigences

- La majorité des défauts (entre 40% et 60%) a comme origine le non respect des exigences.
- ⇒ Les exigences représentent un vecteur central d'amélioration.



Acteurs

- MOA :
 - Maître d'ouvrage : c'est l'initiateur du projet. Il est donc le représentant des utilisateurs finaux à qui l'ouvrage est destiné.
- MOE :
 - Maître d'œuvre : c'est l'entité retenue pour réaliser l'ouvrage. La MOA est client de la MOE à qui elle passe commande d'un produit nécessaire à son activité.

D'une manière générale, la MOE fournit ce produit ; soit il le réalise lui-même, soit il passe commande à un ou plusieurs fournisseurs qui élaborent le produit sous sa direction. Il peut également parfois y avoir des intermédiaires entre MOE et MOA.

Evolution des pratiques



| | Capture des exigences | Conception préliminaire | Conception détaillée | Codage codage | Test unitaire | Test d'intégration | Test système |
|--------------|-----------------------|-------------------------|----------------------|---------------|---------------|--------------------|--------------|
| années 60-70 | 10% | | | 80% | | 10% | |
| années 80 | 20% | | 60% | | | 20% | |
| années 90 | 40% | 30% | | | 30% | | |
| années 2000 | 20% | 30% | | | 50% | | |

Traçabilité des exigences



- **Traçabilité :**
 - C'est la possibilité de lire facilement ce qu'il est advenu et ce qu'il est censé advenir de quelque chose.
- **Traçabilité des exigences :**
 - C'est le fait de pouvoir à tout instant connaître facilement l'origine et les liens entre les exigences, ainsi qu'entre les exigences et le reste du projet ou le contexte (notamment les besoins utilisateur, réalisation et tests).

Pourquoi la traçabilité des exigences

- Elle aide à répondre aux questions du type :
 - D'où vient une exigence ? (Quel besoin cette exigence couvre-t-elle ?)
 - Pourquoi a-t-on conçu la solution de cette manière et quelles étaient les autres possibilités ?)
 - Cette exigence est-elle nécessaire ?
 - Où met-on en œuvre cette exigence ?
 - Comment interpréter cette exigence ?
 - Quelle décision de conception affecte la mise en œuvre de l'exigence ?
 - Cet élément de conception est-il nécessaire ?
 - La solution réalisée est-elle conforme aux exigences ?
 - Comment testera-t-on cette exigence ?
 - Toutes les exigences sont-elles prises en compte ? (peut se lire : est-ce que le projet est terminé ?)
 - Et si non quel est le pourcentage d'exigences prises en compte ? Et quelles sont les exigences non (encore) prises en compte ?
 - Quel est l'impact du changement d'une exigence ?

CMMI

Le Capability Maturity Model Integration contient 22 domaines de processus avec différent niveau de maturité.

— Niveau de maturité 2 – Discipliné —

Gestion de configuration - CM - Configuration Management

Mesure et analyse MA - Measurement and Analysis

Surveillance et contrôle de projet - PMC - Project Monitoring and Control

Planification de projet - PP - Project Planning

Assurance-qualité processus et produits - PPQA - Process and Product Quality Assurance

Gestion des exigences - REQM - Requirements Management

Gestion des accords avec les fournisseurs - SAM - Supplier Agreement Management

— Niveau de maturité 3 – Ajusté —

Analyse et prise de décision - DAR - Decision Analysis and Resolution

Gestion de projet intégrée - IPM - Integrated Project Management +IPPD

Définition du processus organisationnel - OPD - Organizational Process Definition +IPPD

Focalisation sur le processus organisationnel - OPF - Organizational Process Focus

Formation organisationnelle - OT - Organizational Training

Intégration de produits - PI - Product Integration

Développement des exigences - RD - Requirements Development

Gestion des risques - RSKM - Risk Management

Solution technique - TS - Technical Solution

Validation - VAL - Validation

Vérification - VER - Verification

— Niveau de maturité 4 – Géré quantitativement —

Gestion de projet quantitative - QPM - Quantitative Project Management

Performance du processus organisationnel - OPP - Organizational Process Performance

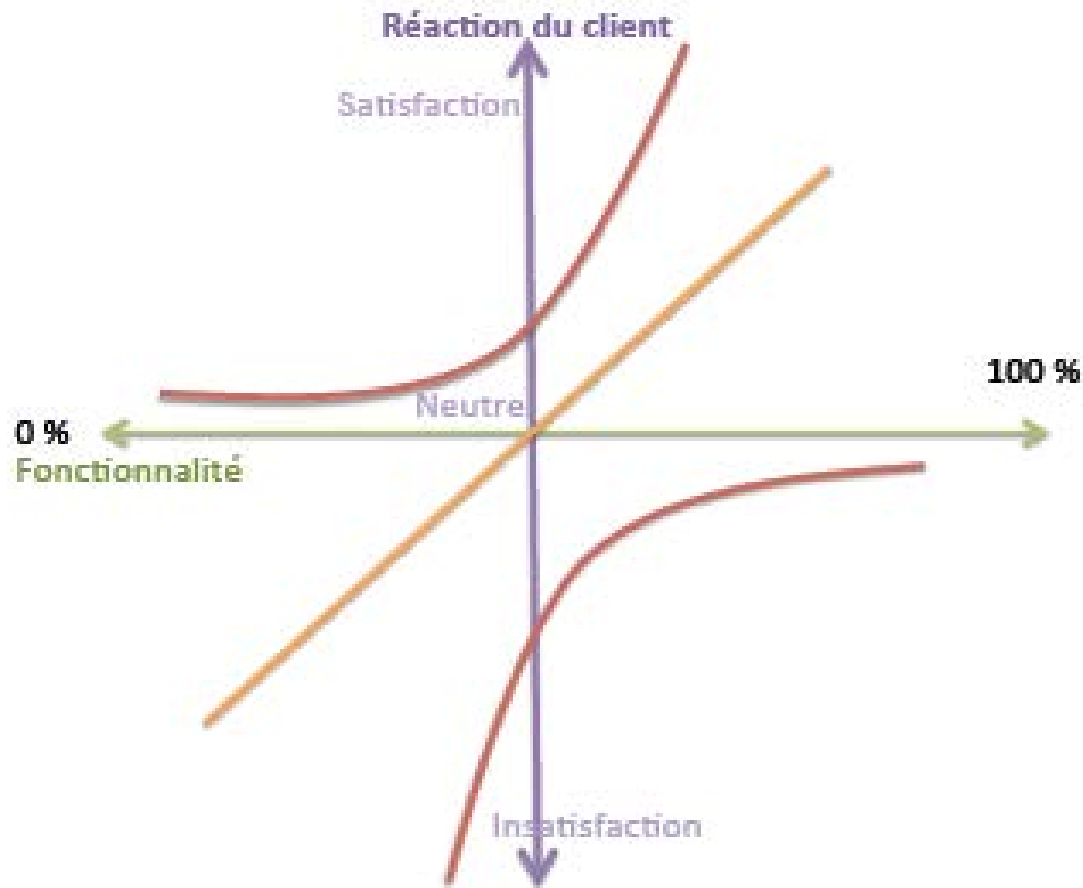
— Niveau de maturité 5 – En optimisation —

Analyse causale et résolution - CAR - Causal Analysis and Resolution

Innovation et déploiement organisationnel - OID - Organizational Innovation and Deployment

Priorité des exigences

Le modèle de Kano : Satisfaction / Insatisfaction (ne sont pas symétriques)



Priorité des exigences

- **Formulaire de Kano** : questionnaire en deux parties
 1. Perception client face à des caractéristiques
 2. Perception client face à l'absence

| | | | | | | | |
|---|--|--|--|--|--|--|--|
| 5 | | | | | | | |
| 4 | | | | | | | |
| 3 | | | | | | | |
| 2 | | | | | | | |
| 1 | | | | | | | |

| | | | | | | | |
|---|--|--|--|--|--|--|--|
| 5 | | | | | | | |
| 4 | | | | | | | |
| 3 | | | | | | | |
| 2 | | | | | | | |
| 1 | | | | | | | |

Formulaire Positif

Formulaire Négatif

Technologies



- SQUASH TM : gestion de la traçabilité des exigences / cas de test
<http://www.squashtest.org/>
- ZEST : gestion des cas de test
<http://zest.smartesting.com>
- Concordion (Java) / Cucumber (C++) : automatisation de l'exécution des spécifications

Robot : exigences



- **Ex0** : La cartographie ne peut être réalisée qu'à l'arrêt.
- **Ex1** : Une première cartographie est systématiquement effectuée lorsque le robot atterrit.
- **Ex2** : A l'état initial, aucune case n'est connue.
- **Ex3** : La cartographie renseigne chaque case par la nature de son terrain.
- **Ex4** : La cartographie permet de couvrir un carré de 9x9.
- **Ex5** : La cartographie permet d'agréger les éléments des cases couvertes aux éléments de cartographie déjà connus.

Robot : tests



TEST 1 : Aucune cartographie en vol (Ex0)

1. **Action** : Faire une cartographie
2. **Observation** : La cartographie n'est pas réalisée (Ex0)

TEST 2 : Une première cartographie est réalisée à l'atterrissage (Ex1)

1. **Action** : Faire atterrir le robot
2. **Observation** : Une cartographie est réalisée (Ex1), Chaque case du carré de 9x9 est renseignée (Ex3,Ex4), Aucune case en dehors du carré de 9x9 n'est renseignée (Ex2,Ex4)

TEST 3 : Deux cartographies successives donnent le même résultat (Ex5)

1. **Action** : Faire atterrir le robot, Faire une cartographie
2. **Observation** : La cartographie doit être identique à celle de l'atterrissage (Ex5), Chaque case du carré de 9x9 est renseignée (Ex3,Ex4), Aucune case en dehors du carré de 9x9 n'est renseignée (Ex2,Ex4,Ex5)

Robot : tests



TEST 4 : Cartographies identiques avec un mouvement d'aller-retour (Ex5)

1. **Action** : Faire atterrir le robot, Avancer le robot, Reculer le robot, Faire une cartographie
2. **Observation** : La cartographie doit être identique à celle de l'atterrissage (Ex5), Chaque case du carré de 9x9 est renseignée (Ex3,Ex4), Aucune case en dehors du carré de 9x9 n'est renseignée (Ex2,Ex4,Ex5)

TEST 5 : Agrégation des cartographies avec un mouvement (Ex5)

1. **Action** : Faire atterrir le robot, Avancer le robot, Faire une cartographie
2. **Observation** : La cartographie obtenue est un rectangle de 10x9 (Ex5), La cartographie contient celle de l'atterrissage plus une ligne (Ex5), Chaque case du rectangle de 10x9 est renseignée (Ex3,Ex4), Aucune case en dehors du rectangle de 10x9 n'est renseignée (Ex2,Ex4,Ex5)