

M2 TIW – Module Projet

Testing

DANIELA TSAMALAX MOKELO

TODAY'S
LESSON

TESTING



SOMMAIRE

Généralités sur les tests

Les tests fonctionnels

Les tests techniques



SOMMAIRE

Généralités sur les tests

Les tests fonctionnels

Les tests techniques

Testing

Généralités



Pourquoi teste-t-on ?

- Pour s'assurer que le produit répond aux attentes du client

Comment connaître les attentes du client ?

- Les attentes sont formulées dans des documents de référence: user stories (SCRUM), SFD, SFG, STD, STG, DAT, ...

Comment savoir quels tests réaliser ?

- A chaque phase du projet va correspondre un type de tests particulier

Testing

Généralités



Qu'est-ce qu'un bug ?

- Un bug (ou non-conformité, ou défaut, ou anomalie) un comportement de l'application non conforme aux attentes client

Quelles sont les typologies de bug

- Un bug peut être technique (application non disponible) ou fonctionnel (l'appli ne réagit pas comme attendu)

Sévérité des bugs

- **Bloquant**: fonctionnalité critique impactée + pas possible d'accomplir l'action jusqu'au bout + pas de solution de contournement
- **Majeur** : fonctionnalité critique impactée + solution de contournement
- **Mineur**: tous les autres problèmes



- Plus un test est réalisé tôt, moins il est coûteux
- Une bonne campagne de test est une campagne qui remonte des bugs
- Si l'équipe ne dispose pas d'un testeur, les tests de qualification peuvent être effectués par les développeurs eux-mêmes lors de tests croisés
- Plus on avance dans le cycle de vie du projet, moins on doit trouver de bugs
- LA QUALITE EST L'AFFAIRE DE TOUS**



SOMMAIRE

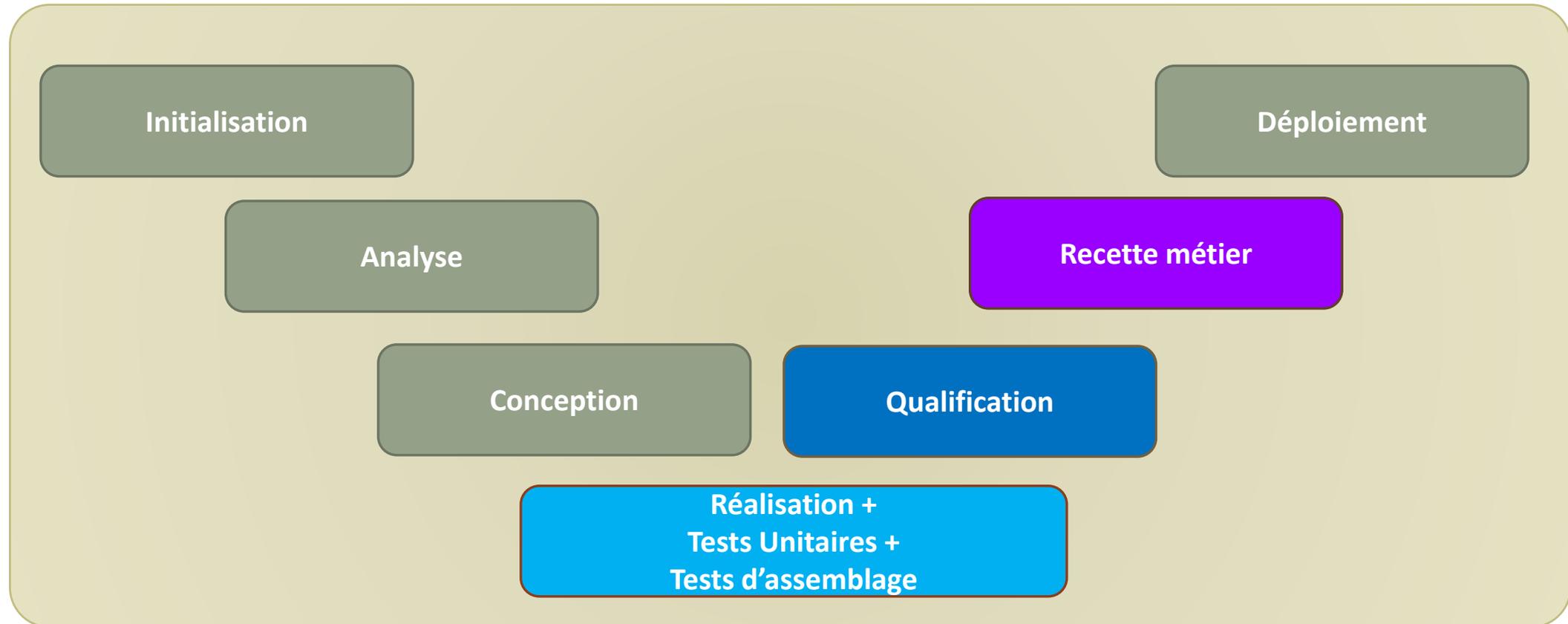
Généralités sur les tests

Les tests fonctionnels

Les tests techniques



Les différentes phases de test



Types de tests

Tests fonctionnels



Type de test	Définition	Acteur	Terme ISTQB
Fournisseur	1 ^{ère} phase de test : c'est le premier test dynamique effectué au niveau programme suite à une création ou modification du code. Il sert à tester le bon fonctionnement d'une fonction.	Développeur	Test Composant
	2 ^{ème} phase de test : elle vérifie que le code écrit par les différents développeurs est compatible une fois agrégé (test d'accrochages de leur code)	Développeur	Test d'Intégration Composant
	3 ^{ème} phase de test : elle vérifie le fonctionnement de l'application, incluant les interfaces internes et externes, et intégrant tous les composants hardware, software et infrastructure. La phase de qualification s'appuie sur les plans de tests issus des SFD. Les batchs sont testés avec des fichiers de tests (écrits à la main ou transmis par l'application fournisseur).	Testeur	Test Système + Test d'Intégration Système
Client	4 ^{ème} phase de test : elle vérifie que le logiciel est en mesure de traiter un processus métier dans son entier, selon les règles du métier et en faisant appel, le cas échéant, à d'autres briques du système d'information. La phase de recette s'appuie sur les exigences métier. L'objectif est de s'assurer que l'application répond au besoin métier	MOA/Métier	Test d'Acceptance

Types de tests

Tests fonctionnels

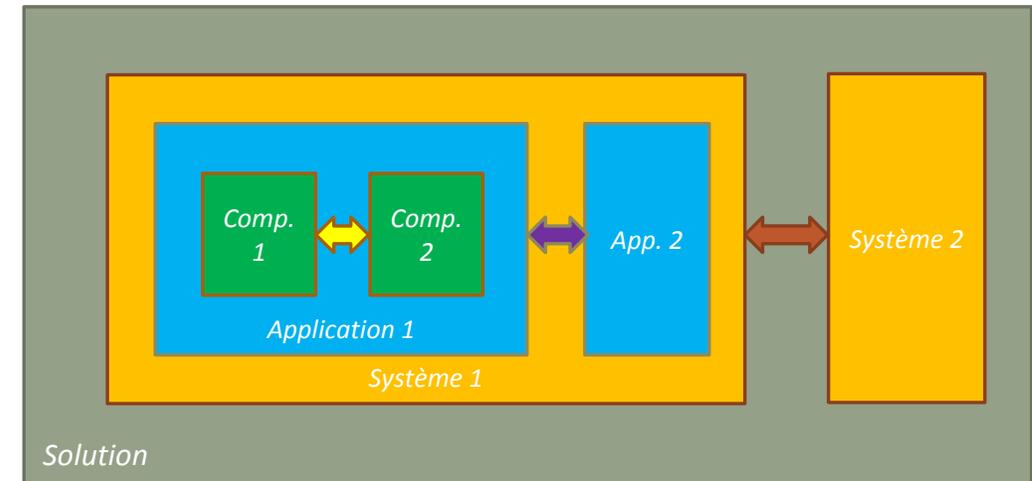


Type de test	Quand?	Qui ?	Quoi?	Outils
Test unitaire	Avant de commencer à développer	Celui qui va développer le code	<p>Cas passant / Cas KO / Cas limite / La gestion des erreurs</p> <p>Les tests unitaires doivent être exhaustifs (balayer tous les cas)</p>	Manuel JUnit / NUnit...
Test d'assemblage	Quand quelques fonctionnalités sont développées	Celui qui a développé le code	<p>Cas passant / Cas KO / Cas limite / La gestion des erreurs</p> <p>Les tests d'assemblage doivent être exhaustifs (balayer tous les cas)</p>	Manuel JUnit / NUnit
Tests de qualification	Dès que quelques modules sont disponibles	<p>Testeur (à défaut un développeur)</p> <p><u>Pas la personne qui a développé le module</u></p>	<p>On teste les cas d'utilisation: scénario nominal, scénario alternatif, quelques cas d'erreur (test du singe)</p> <p>Les tests de qualification doivent être exhaustifs</p>	<p>Excel</p> <p>HP ALM</p> <p>ClearQuest</p> <p>Mantis</p> <p>Trello</p> <p>...</p>

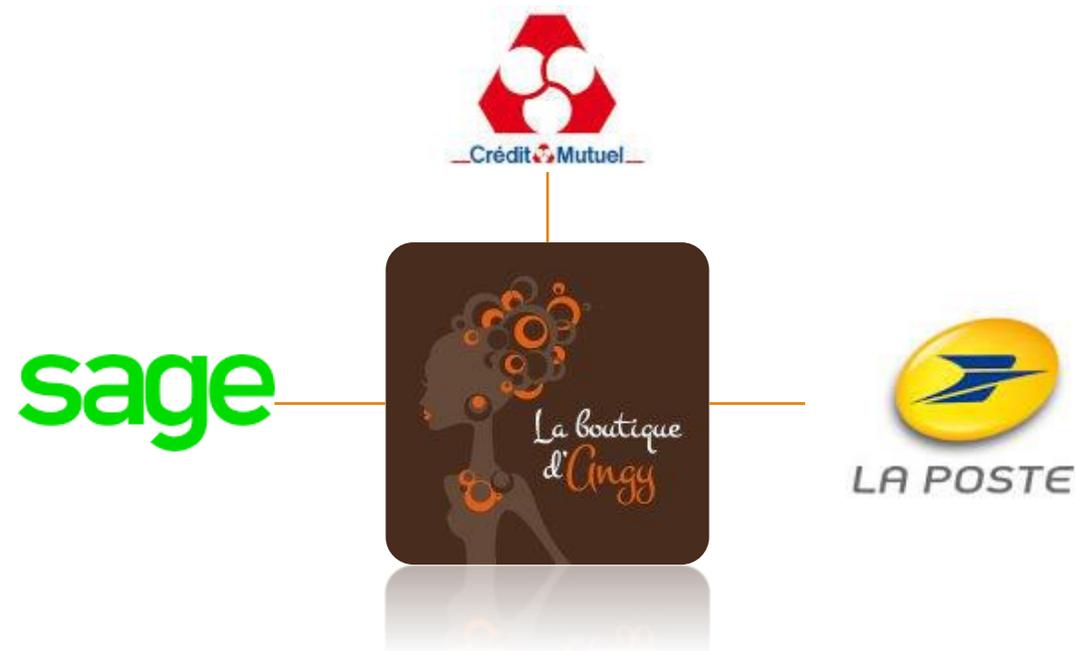


Niveaux de tests

	Niveau de test	Définition
Test unitaire	Composant (fonction/méthode)	C'est le premier test dynamique effectué au niveau programme suite à une création ou modification du code
Test d'assemblage	Intégration entre composant	Teste le fonctionnement des composants de l'application, par sous-ensembles, et leur l'interfaçage entre eux
Test de qualification	Applicatif	Teste le fonctionnement de l'application globale
	Système	Teste les interfaces avec d'autres applications
Tests de recette	Intégration entre les systèmes	Teste le fonctionnement de toutes les applications, incluant les interfaces internes et externes à l'organisation, et intégrant tous les composants hardware, software et infrastructure
	Solution	Vérifie que le produit livré correspond bien aux exigences de l'utilisateur



Exercice 1 : Indiquer s'il s'agit d'un composant, d'une application ou d'un système



Élément	Catégorie
Site Internet « La boutique d'Angy »	
Paiement en ligne sécurisé avec le Crédit mutuel	
Comptabilité avec Sage	
Logistique avec La Poste	
Gestion du panier	
Authentification	
Compte utilisateur	

Exercice 2 : tests unitaires



Quels tests unitaires peut-on faire ?

Site de e-commerce: ajouter au panier

Quand le client ajoute un article au panier, le système vérifie que l'article est présent en stock. Si la quantité est suffisante en stock, l'article est ajouté au panier. Sinon, un message d'erreur informe le client. 2 cas de figure sont possibles: soit il reste quelques articles en stock mais pas la quantité désirée. Dans ce cas, le client peut choisir de les ajouter ou non. Soit le stock est égal à 0, dans ce cas, le client peut mettre une alerte pour être informé lors du prochain approvisionnement.

Identification

Quand l'utilisateur saisit son identifiant et son mot de passe, le système recherche son compte. Si le compte existe et que le mot de passe est correct alors, les données utilisateur sont chargées en session et un message de bienvenue accueille l'utilisateur. Si le compte n'existe pas, un message d'erreur est affiché. Si le compte existe mais le mot de passe est incorrect, un message d'erreur est affiché. Au bout de 3 saisies incorrectes, le compte utilisateur est bloqué.

Exercice 3 : tests de qualification



Rédiger un scénario de test pour valider la commande

Quand l'utilisateur a fini sa sélection, il valide son panier.

Si l'utilisateur est déjà identifié, alors le processus de commande peut se poursuivre.

Si l'utilisateur n'est pas identifié, il est redirigé vers la page de login. Après l'authentification, il retournera sur la page de commande et le processus de commande pourra se poursuivre.

La commande consiste en:

1. validation du panier
2. saisie de l'adresse de livraison
3. paiement en ligne

En cas d'échec du paiement, la commande ne peut pas se poursuivre.

Le client a toutefois la possibilité de sauvegarder son panier pendant 24h. Au bout de 24h, sans paiement, le panier sera vidé et les articles réservés seront remis dans le stock.

#	Description	Résultat attendu	OK / KO	Résultat constaté



SOMMAIRE

Généralités sur les tests

Les tests fonctionnels

Les tests techniques

Types de tests

Tests techniques



Type de test	Définition	Acteur
Tests de charge ou robustesse	Ils permettent de vérifier s'assurer qu'en cas de pic d'activité, l'application réagit conformément aux spécifications.	Développeur / Expert technique
Tests de performance	Ils permettent de vérifier que l'application respecte bien les temps de réponse spécifiés dans les exigences techniques. Ils ne font pas appel à des techniques ou outils spécifiques (si ce n'est les outils nécessaires à la mesure des temps de traitement).	Développeur / Expert technique
Tests de fonctionnement dégradé	Ils permettent de vérifier le comportement du système en cas de défaillance d'un composant, et concernent donc la sûreté de fonctionnement du logiciel. Le composant défaillant peut être : poste local, serveur, base de données etc.	Développeur / Expert technique

Tests de robustesse et test de performance



- Nécessite des outils spécifiques (ex: SQL Load Runner)
- Nécessite un environnement spécifique
- Nécessite un budget spécifique
- Peuvent être coûteux (temps, profil experts)
- Peuvent remettre en cause l'architecture de la solution

Tests de robustesse et test de performance



Éléments pouvant nuire à la robustesse et à la performance

- Les requêtes SQL
- La gestion des transactions
- Le faible espace disque
- Le manque de RAM

Optimiser les performances et la robustesse

- Optimiser les requêtes sql
- Déporter des traitements côté client
- Optimiser la gestion des transactions
- ↗ RAM + HDD

Exercice 4 : Tests de performance et de robustesse

Proposez des optimisations



Durant la période des fêtes de fin d'année, le site, la boutique d'Angy multiplie par 100 sa fréquentation journalière.

Les dirigeants de la boutique voudraient s'assurer que cette année, le site pourra répondre aux nombreuses sollicitations.

Le site internet est vieux et n'est pas du tout optimisé pour la navigation mobile. Il n'utilise pas de gestion de cookies ni de gestion du cache.

Le site est hébergé sur un serveur mutualisé. Un audit récent a montré qu'un achat génère en moyenne 30 call à la base de données. 1 call de base de données = 1 transaction.

Enfin, la base de données est ancienne et n'a ni vues ni index (hormis les clés primaires)



SOMMAIRE

Généralités sur les tests

Les tests fonctionnels

Les tests techniques

Environnements et données de tests



Outils, livrables, données de tests,...

Type de test	Outils	Livrables d'entrée	Données de test	Cas de tests	Livrables de sortie / Résultats de tests
Unitaire	Junit, PHPUnit, Nunit, ...	Classes de tests	Les jeux de tests sont intégrés dans la classe de test et les fonctions correspondantes	A chaque combinaison (« if », « case », ...) doit correspondre un cas de test	Retour de la fonction
Assemblage	Manuel (Excel, ...)		Préparés manuellement (insertions en base, fichiers créés pour les tests, ...) Il est parfois nécessaire d'utiliser des bouchons	Test des scénarios nominaux et des principaux scénarios alternatifs A défaut d'effectuer les tests sur les terminaux cibles, au moins utiliser les résolutions cibles.	Logs, écriture en BDD, fichiers de sortie, IHM modifiée, ...
Qualification	Clearquest, HP ALM, Excel, ...	Plans de tests (scénarios)	Données réelles simplifiées Il est parfois nécessaire d'utiliser des bouchons	Tous les scénarios métiers, même les cas les plus rares. Les tests doivent être effectués sur les terminaux cibles	Résultat des plans de tests Fichiers générés par l'application, résultats consultables dans l'IHM
Recette métier	Clearquest, HP ALM, Excel, ...	Plans de tests (scénarios)	Données de prod anonymisées Tous les flux sont ouverts, aucune utilisation de bouchons	Processus métier. Généralement les cas plus fréquents	Résultats des campagnes d'exécution Fichiers générés par la solution, résultats consultables dans l'IHM



Environnements

Environnements	Définition	Tests Unitaires	Tests d'Assemblage	Tests de Qualification	Recette Métier
Développement	Environnement utilisé par les développeurs pour effectuer leurs développements. Il correspond en général à l'IDE et à la base de données installés sur le poste de développement. Il est donc unique pour chaque développeur	X			
Pré-Intégration	Environnement utilisé pour tester le code après que celui-ci ait été mis en commun avec les autres développeurs. L'application est déployée dessus, la base de données est commune. Les jeux de données présents sont minimalistes		X		
Qualification	Environnement utilisé pour tester les scénarios métiers. Certaines interfaces peuvent être activées.			X	
Démonstration	Utilisé dans les projets agile. Il permet la mise à disposition au Product owner des incréments de fin de sprint.				X
Intégration	Environnement proche de la production. Les interfaces avec les systèmes externes sont activées. Il permet de jouer les processus métier de bout en bout.				X
Pré-production	Environnement iso-prod (même dimensionnement, mêmes OS, ...) utilisé pour tester les correctifs avant déploiement en production				X
Production	Environnement utilisé par les utilisateurs finaux au quotidien pour réaliser leurs besoins.	-	-	-	-



Synthèse

Type de test	Phase projet	Niveau de test	Document de référence	Qui porte l'action ?	Qui conçoit le test ?	Qui exécute le test ?	Quels sont les outils utilisés ?	Sur quel environnement les tests sont-ils effectués ?	Preuves d'exécution
Unitaire	Dev.	Composant	STD	Fournisseur	Equipe de développement	Equipe de développement	JUnit, ...	Développement	PIC
Assemblage	Dev.	Intégration entre composants	SFD	Fournisseur	Equipe de développement	Equipe de test	Manuel	Intégration	PIC
Qualification	Qualification	Système	SFD	Fournisseur	Equipe de test	Equipe de test	Excel, HP ALM, ...	Qualification	Fichiers de tests + captures d'écran + éventuels fichiers générés par l'application
Qualification	Qualification	Intégration entre les systèmes	SFD	Fournisseur	Equipe de test	Equipe de test	Excel, HP ALM	Qualification	Fichiers de tests + captures d'écran + éventuels fichiers générés par l'application
Recette métier	Recette	Solution	User stories / Exigences métier	Client	MOA/Métier	MOA/Métier	Excel, HP ALM	Démonstration Intégration Pré-production	