

Génie Logiciel

MODULE 4 - TESTS DE LA SOLUTION

VENDREDI 27/11

*

Découpage de ce cours

Cours + TD1

Overview d'un projet informatique

Cours + TD2

Spécification et architecture technique de la solution

Cours + TD3

Construction de la solution

Cours + TD4

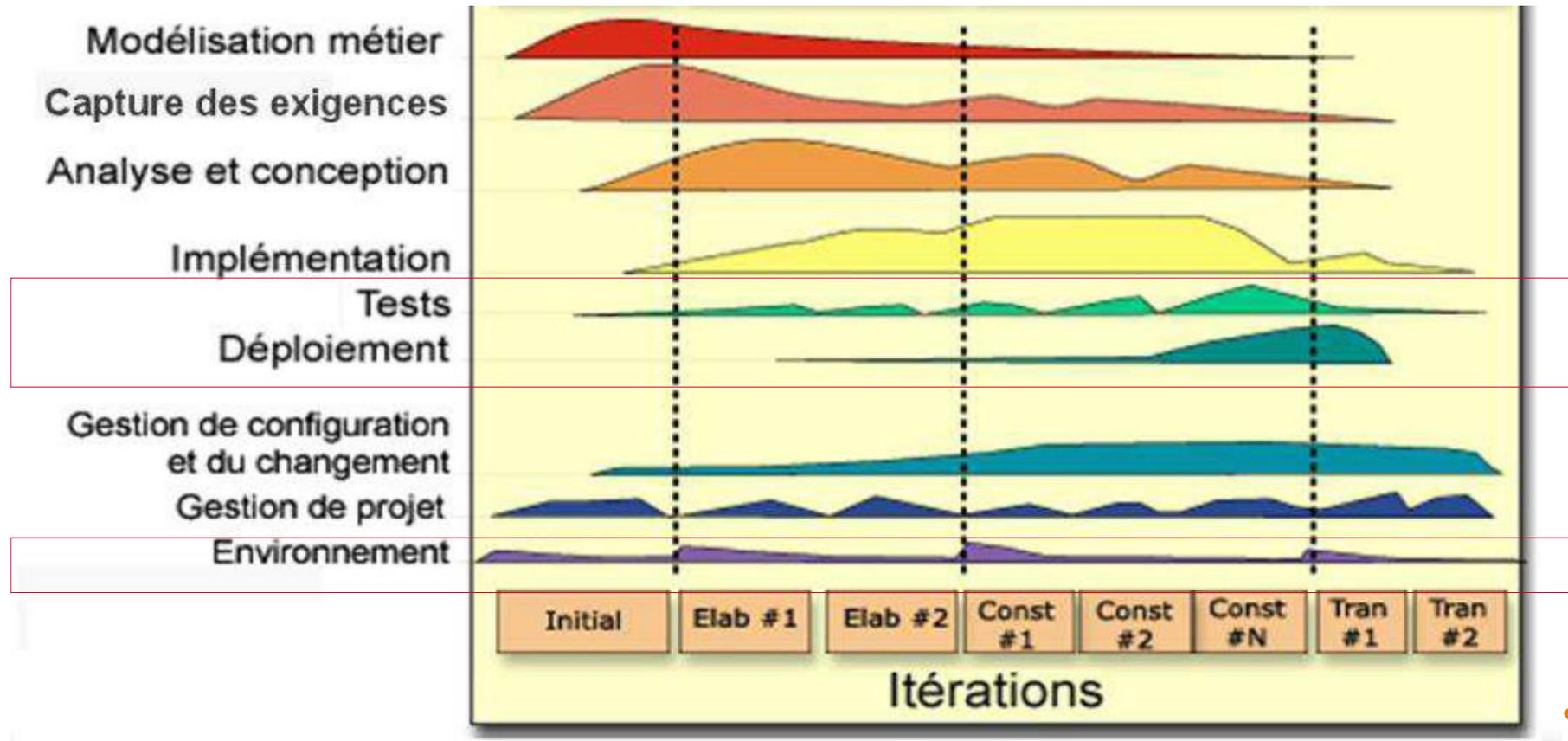
Tests de la solution

- Approfondissement du test unitaire
- Tests d'intégration
- Qualification
- Recette et MEP

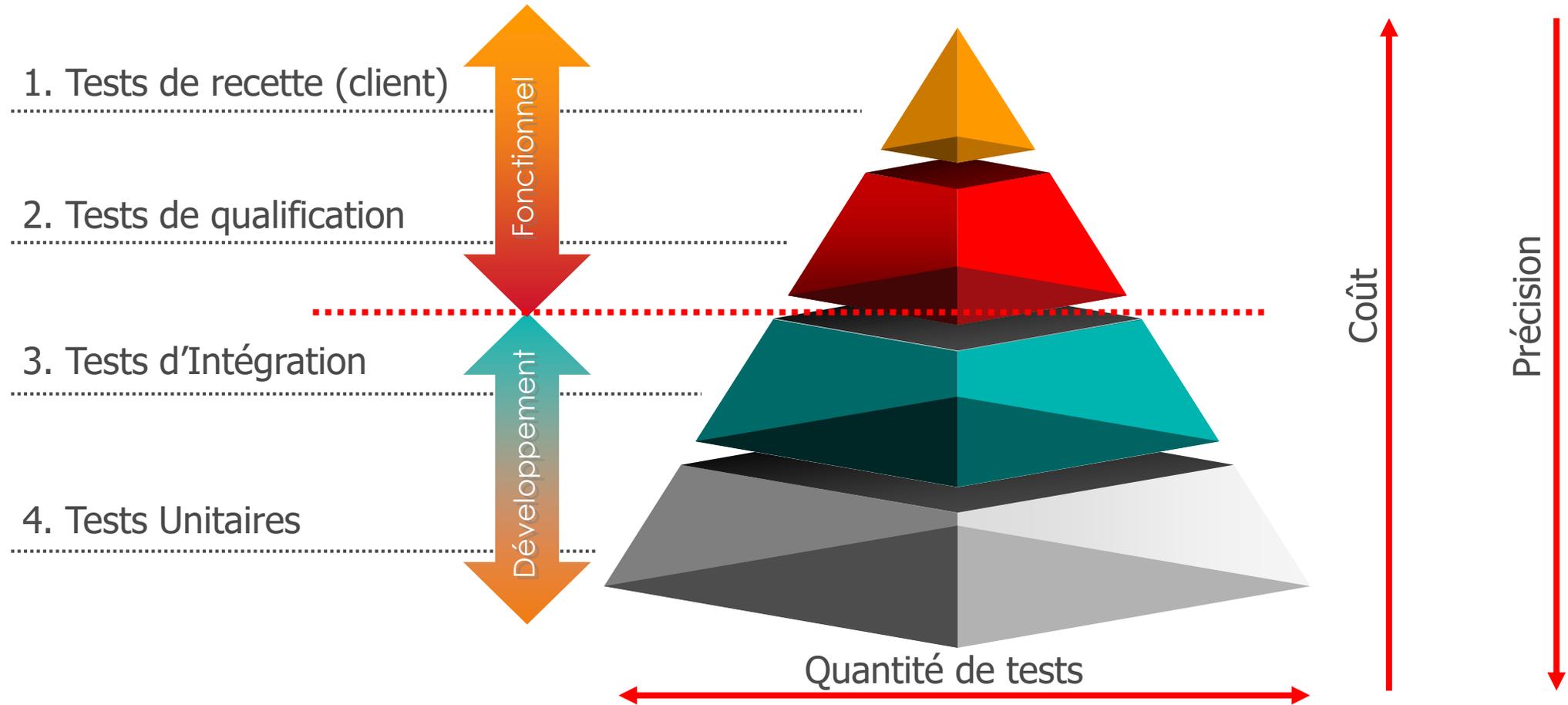
Connaissez-vous les différents niveaux de test (y compris les TU) ?

Dans chaque niveau :

- Qu'est-ce que l'on test ?
- Quelle est la quantité de test (sur une échelle S/M/L/XL) ?
- Quelle est le coût d'un test (sur une échelle S/M/L/XL) ?
- Quel est la précision du test pour identifier le bug (sur une échelle S/M/L/XL) ?

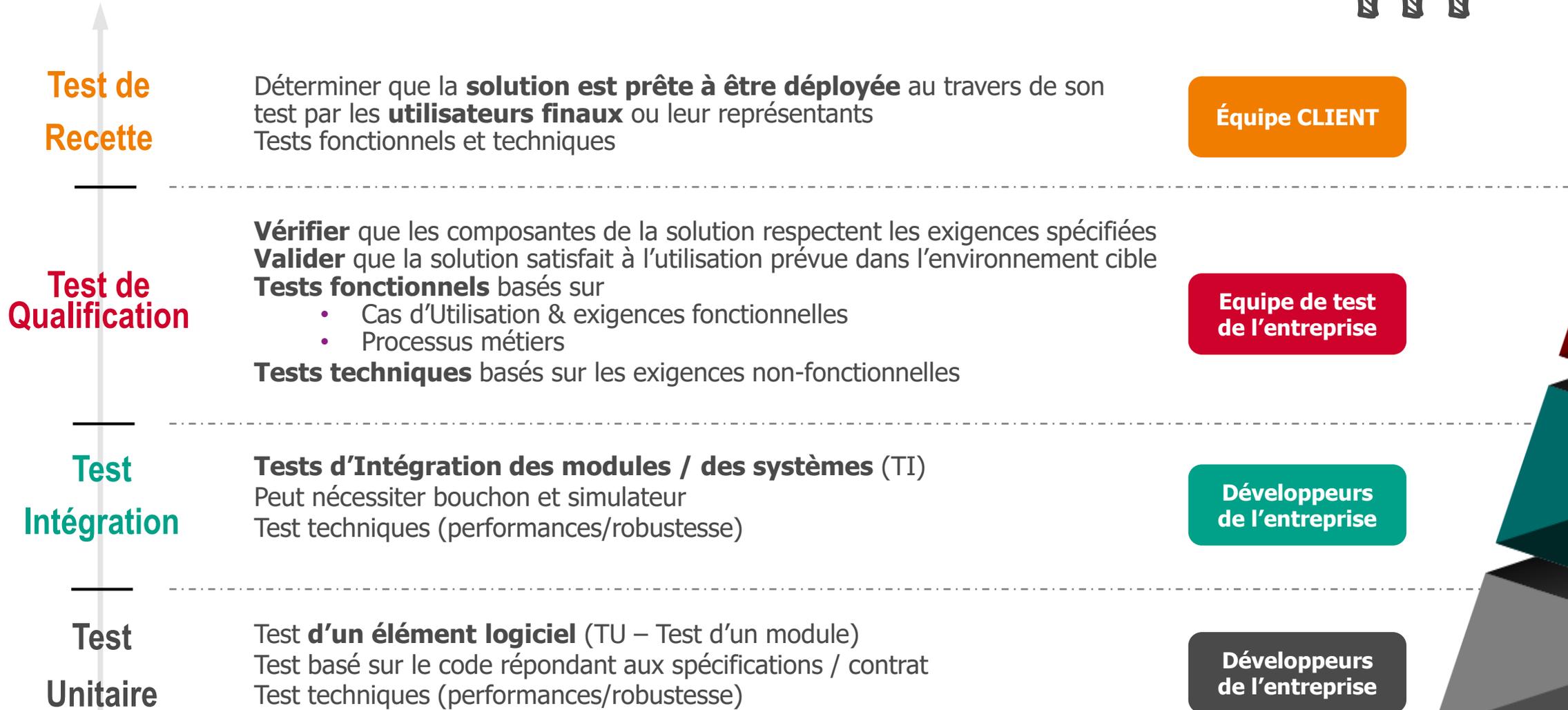


Niveaux de test



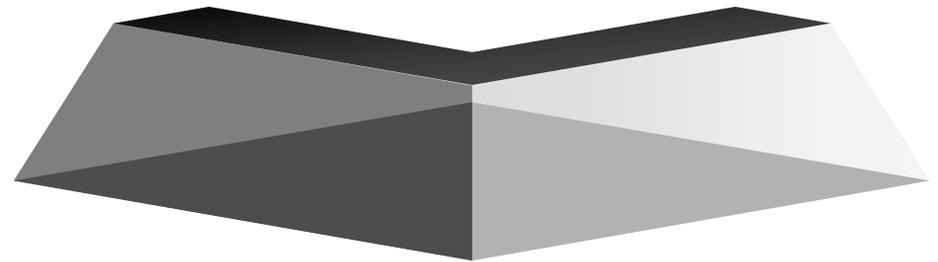
Pyramide vertueuse : on optimise l'effort de tests de sorte de ne pas retester à chaque niveau tout ce qui a été testé dans les niveaux inférieurs !

Niveaux de test



01

Approfondissement du test unitaire



Tests unitaire

EXERCICE

— Voici un exemple de spécification d'une opération

```
rechercherReservations(nom:String, prenom:String) : Reservation [0..*]
```

- └ Pré-conditions : le nom doit être obligatoirement renseigné
- └ Post-conditions : le système renvoie une liste de réservations

— Identifier les cas à tester

- └ Un conseil : essayez d'imaginer des situations pouvant mettre en défaut l'implémentation

— Lors de cette réflexion sur les différents tests à mettre en place, avez-vous constaté des imprécisions dans les spécifications ou les cas de test ?

- └ Si oui, rédigez une liste de questions au spécifieur pour préciser les spécifications

— Remplir le tableau suivant avec les input à tester :

Nom	Prénom	Objectif du test



Test Driven Development

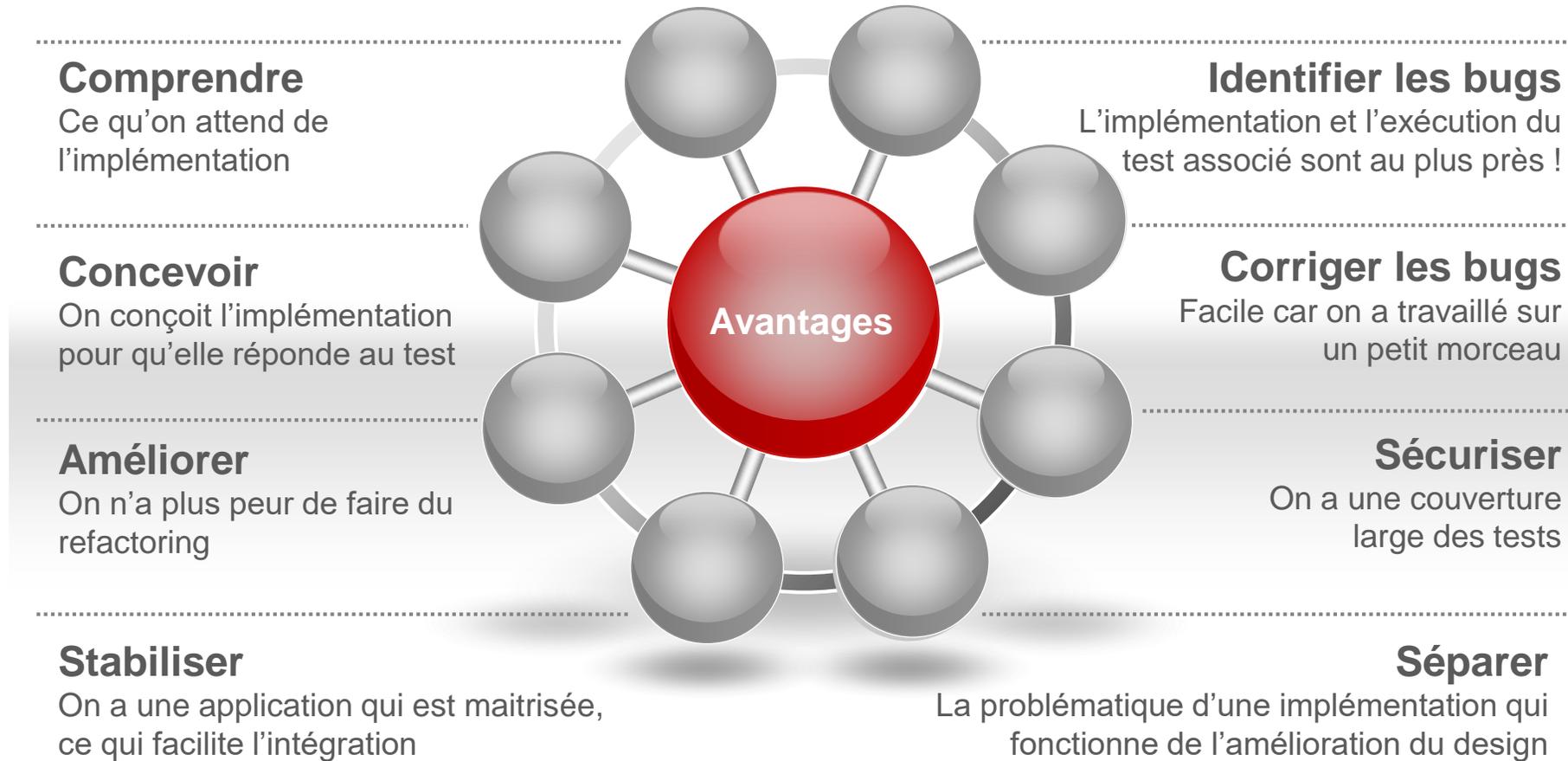
Définition

- Le TDD est un méthode qui consiste à implémenter une partie de l'application en se reposant sur les tests pour affiner le design et sécuriser l'implémentation
- **On traite une petite partie de la fonctionnalité à dev**
- **On développe le test correspondant**
- **On le joue : il échoue**
 - └ C'est normal, l'implémentation n'est pas réalisée
- **On implémente pour que le test n'échoue plus**
- **Quand l'implémentation est stable (succès des tests), on peut envisager du refactoring de code**
 - └ Pas de nouvelle fonctionnalité mais un meilleur design
 - └ Les tests sont un rempart contre la régression
- **On réfléchit à de nouveaux tests pertinents**
- **Pour corriger un defect, commencer par écrire le test qui le reproduit**

Même si cette pratique n'est pas « imposée » sur le projet. Chacun peut décider de l'adopter

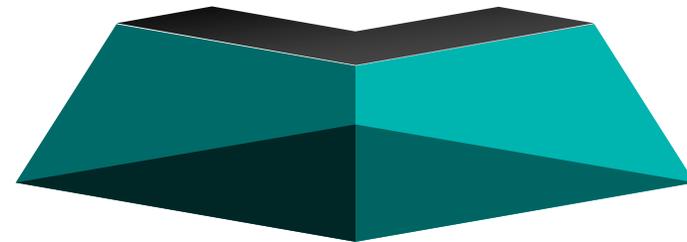
Test Driven Development

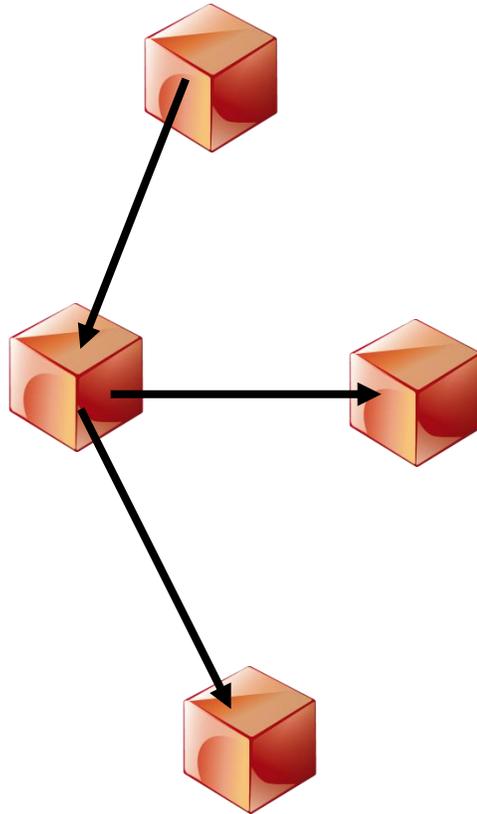
Les avantages



02

Les tests d'intégration



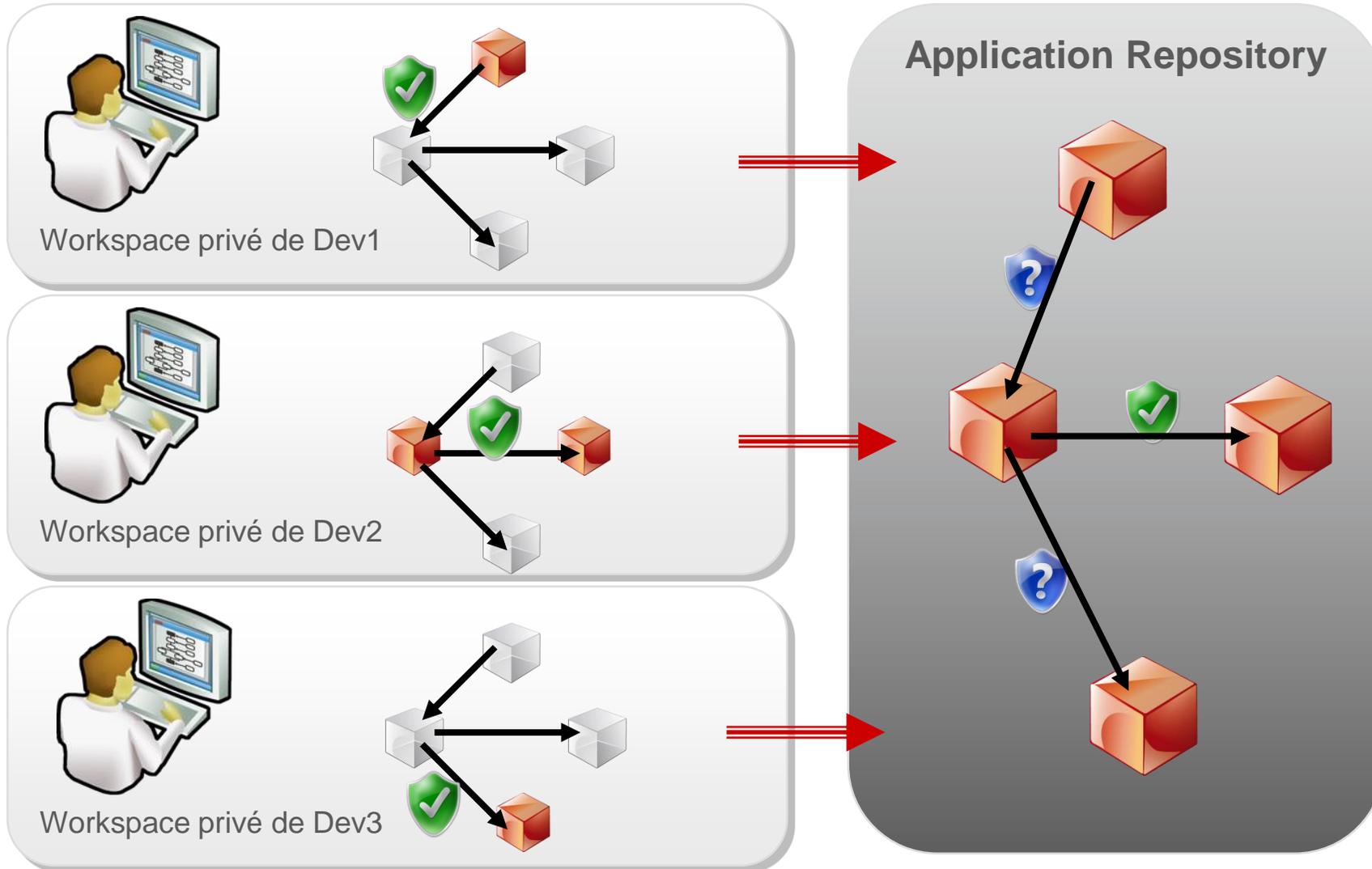


Qu'est-ce qu'un test d'intégration ?
Vous avez 4 modules (avec TU) qui interagissent,
qu'est-ce que vous testez en intégration ?

Qu'est-ce qu'un bouchon, quand l'utiliser ?

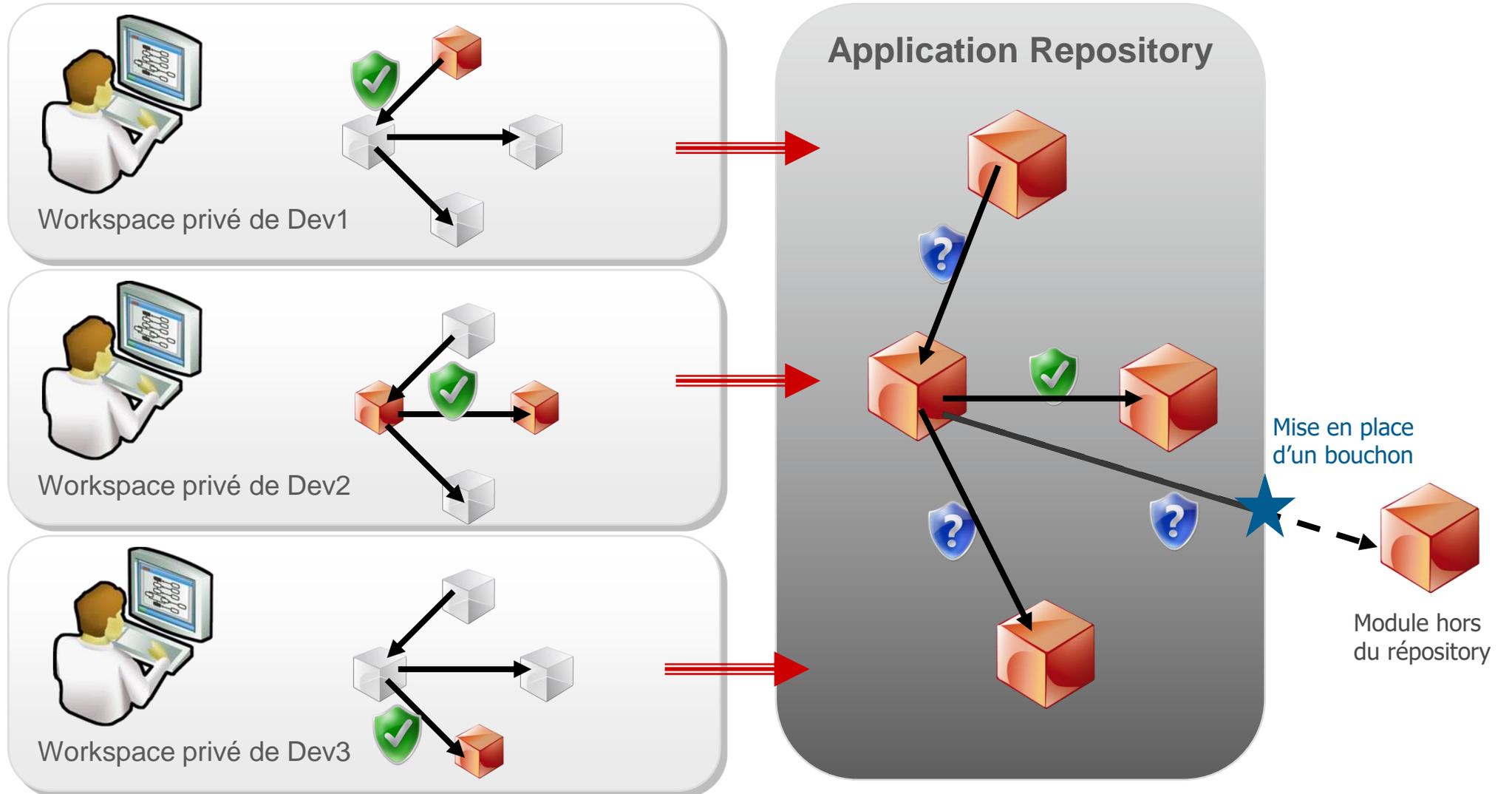
Tests d'intégration

Éventuels problèmes d'intégration



Tests d'intégration

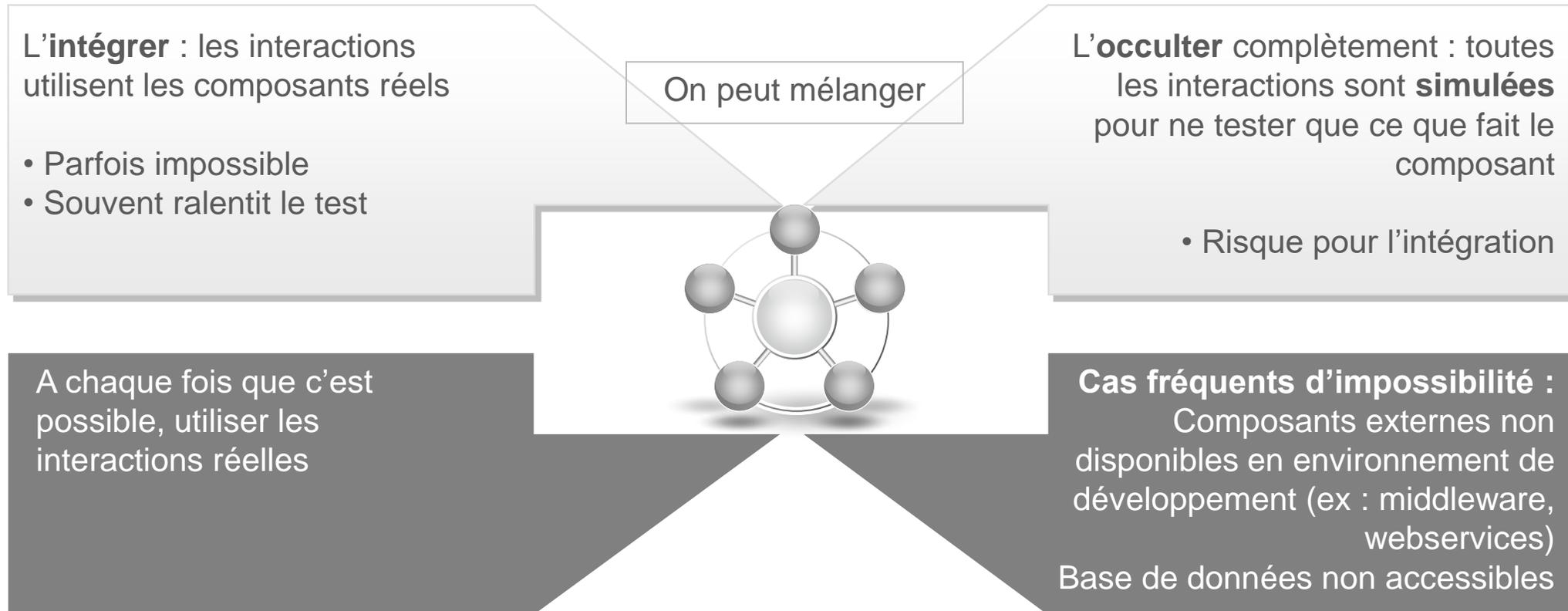
Introduction des bouchons



Tests d'intégration

La problématique du bouchonnage

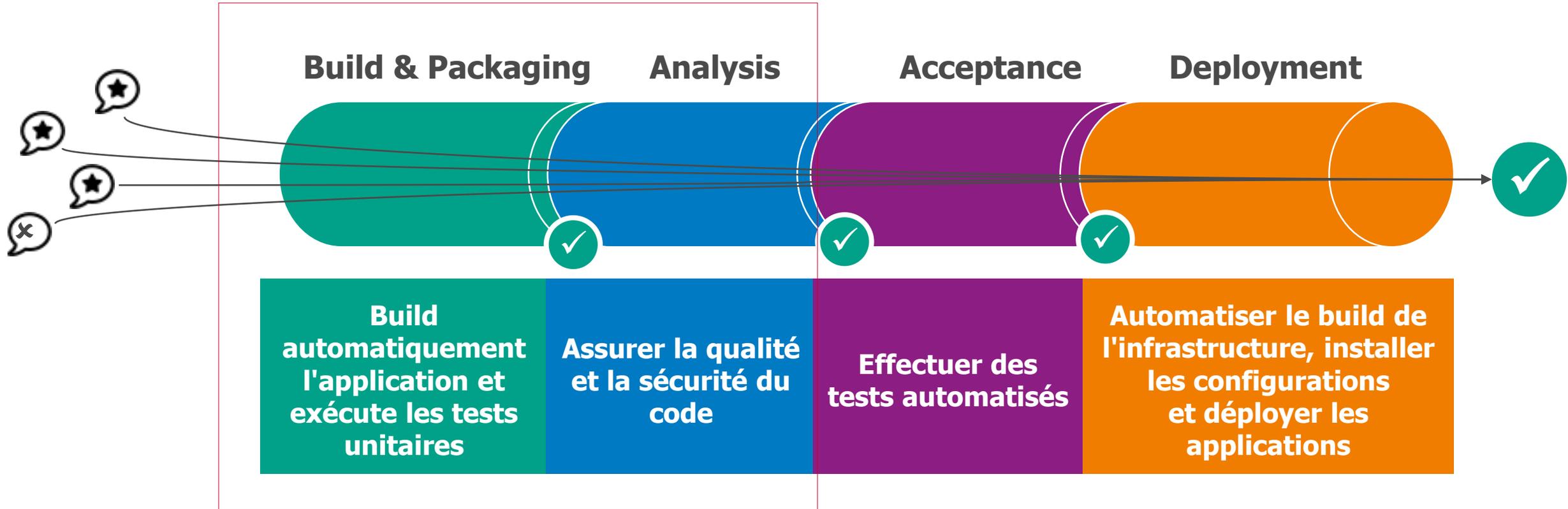
- 2 options pour tester l'intégration avec un module qui est en dehors de notre application



★ → Dans ce cas de figure, on peut utiliser un bouchon avec différents niveaux d'intelligence (stub, mock, ...)

Tests d'intégration

L'intégration continue

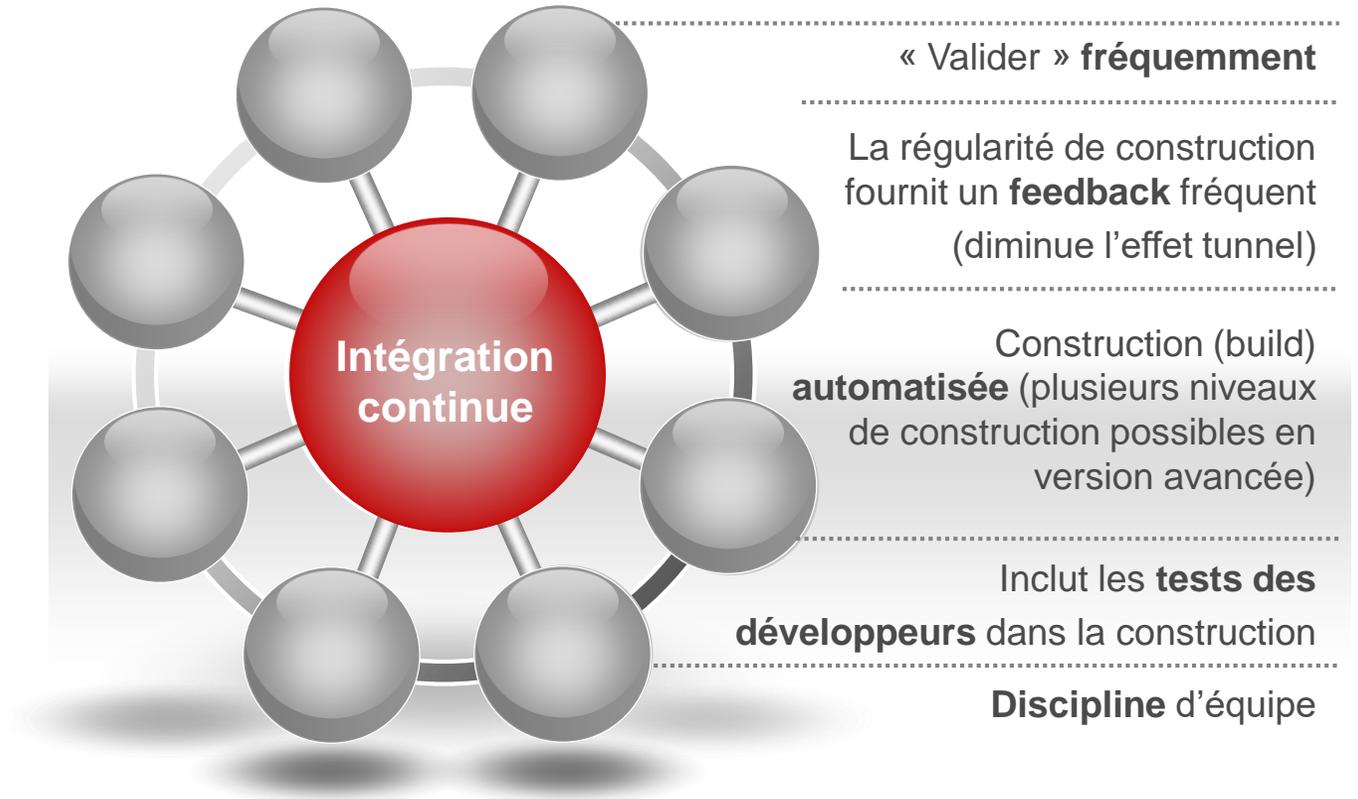


Sur un projet industrialisé, les tests d'intégration sont automatisés et joués dans les premières étapes de la pipeline

Tests d'intégration

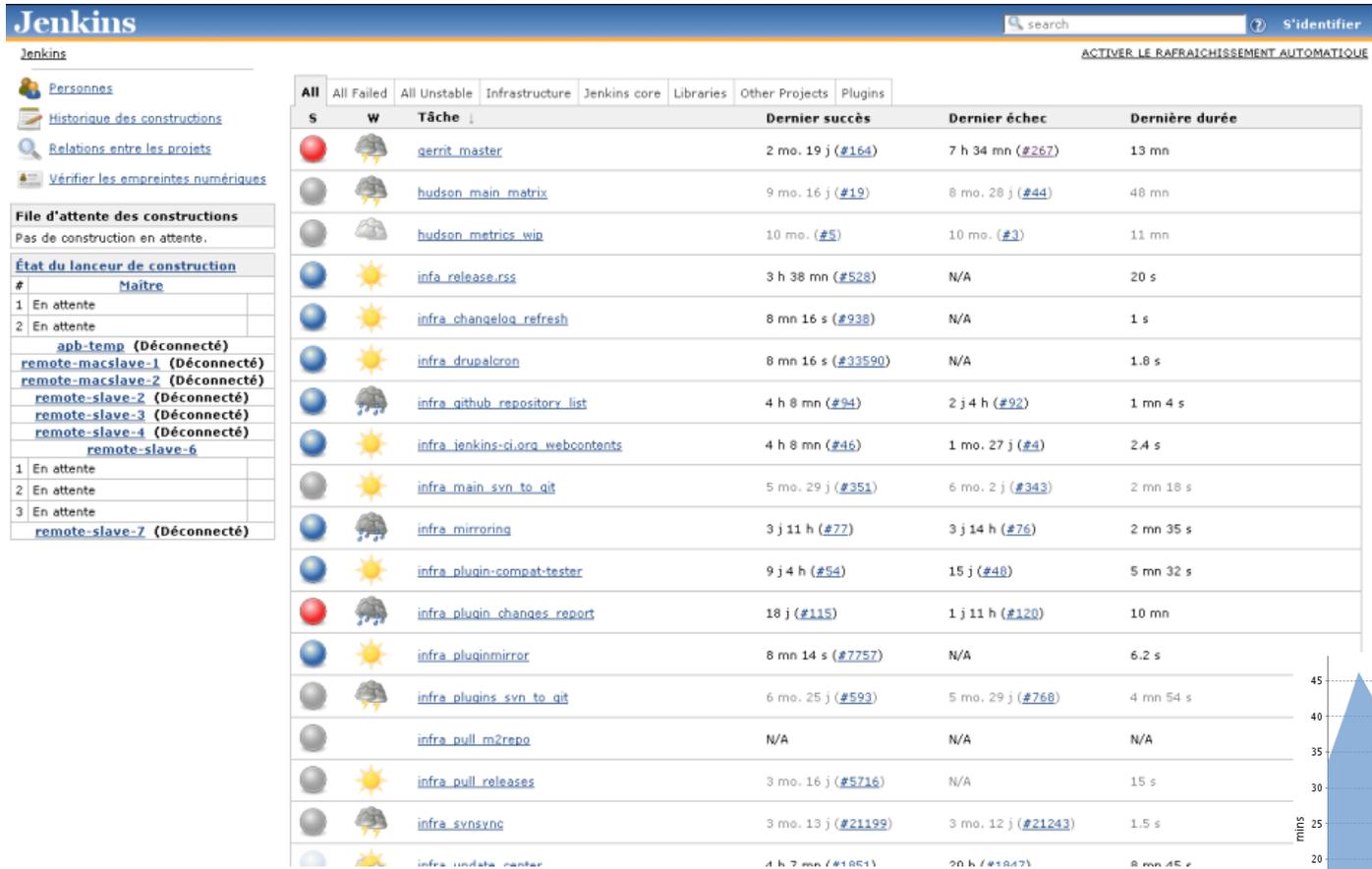
Les avantages de l'intégration continue

- C'est d'abord une volonté et une organisation, avant une utilisation d'outils
- Les développeurs participent activement à l'intégration : leurs tests sont rejoués et ne sont pas à la charge d'un intégrateur omniscient



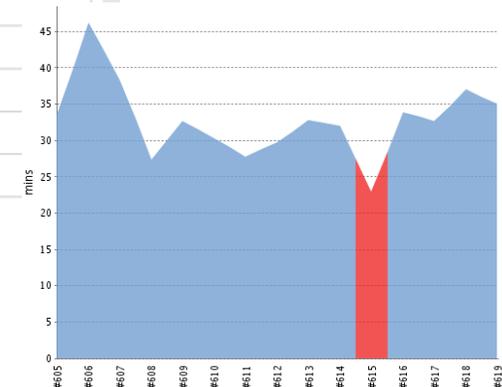
Tests d'intégration

Exemple de rapport d'intégration



The screenshot shows the Jenkins dashboard with a table of build jobs. The table has columns for status (S), worker (W), task name (Tâche), last success (Dernier succès), last failure (Dernier échec), and last duration (Dernière durée). The jobs listed include 'gerrit_master', 'hudson_main_matrix', 'hudson_metrics_win', 'infra_release_rss', 'infra_changelog_refresh', 'infra_drupaloron', 'infra_github_repository_list', 'infra_jenkins-ci.org_webcontents', 'infra_main_svn_to_git', 'infra_mirroring', 'infra_plugin_compat-tester', 'infra_plugin_changes_report', 'infra_pluginmirror', 'infra_plugins_svn_to_git', 'infra_pull_m2repo', 'infra_pull_releases', and 'infra_synsvnc'.

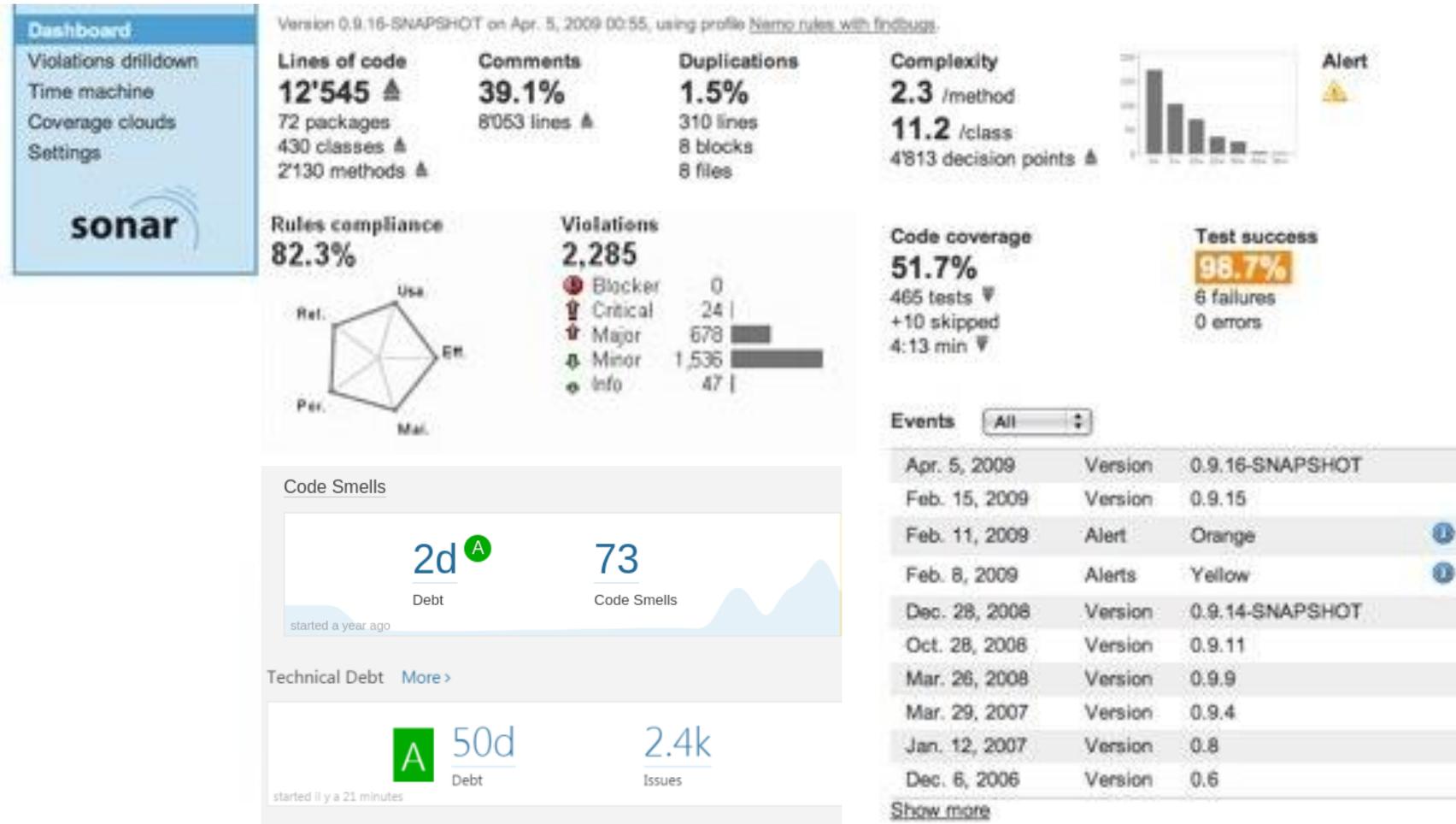
S	W	Tâche	Dernier succès	Dernier échec	Dernière durée
●	☁	gerrit_master	2 mo. 19 j (#164)	7 h 34 mn (#267)	13 mn
●	☁	hudson_main_matrix	9 mo. 16 j (#19)	8 mo. 28 j (#44)	48 mn
●	☁	hudson_metrics_win	10 mo. (#5)	10 mo. (#3)	11 mn
●	☀	infra_release_rss	3 h 38 mn (#528)	N/A	20 s
●	☀	infra_changelog_refresh	8 mn 16 s (#938)	N/A	1 s
●	☀	infra_drupaloron	8 mn 16 s (#33590)	N/A	1.8 s
●	☁	infra_github_repository_list	4 h 8 mn (#94)	2 j 4 h (#92)	1 mn 4 s
●	☀	infra_jenkins-ci.org_webcontents	4 h 8 mn (#46)	1 mo. 27 j (#4)	2.4 s
●	☀	infra_main_svn_to_git	5 mo. 29 j (#351)	6 mo. 2 j (#343)	2 mn 18 s
●	☁	infra_mirroring	3 j 11 h (#77)	3 j 14 h (#76)	2 mn 35 s
●	☀	infra_plugin_compat-tester	9 j 4 h (#54)	15 j (#48)	5 mn 32 s
●	☁	infra_plugin_changes_report	18 j (#115)	1 j 11 h (#120)	10 mn
●	☀	infra_pluginmirror	8 mn 14 s (#7757)	N/A	6.2 s
●	☁	infra_plugins_svn_to_git	6 mo. 25 j (#593)	5 mo. 29 j (#768)	4 mn 54 s
●	●	infra_pull_m2repo	N/A	N/A	N/A
●	☀	infra_pull_releases	3 mo. 16 j (#5716)	N/A	15 s
●	☁	infra_synsvnc	3 mo. 13 j (#21199)	3 mo. 12 j (#21243)	1.5 s
●	☀	infra_update_center	4 h 7 mn (#1851)	20 h (#1847)	8 mn 45 s



Résultats souvent accessibles via des pages web liées à l'outil d'intégration (Jenkins, Hudson)

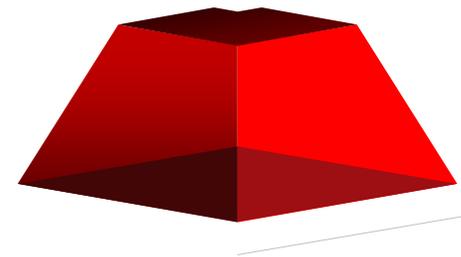
Tests d'intégration

Exemple de dashboard SONAR



03

Qualification

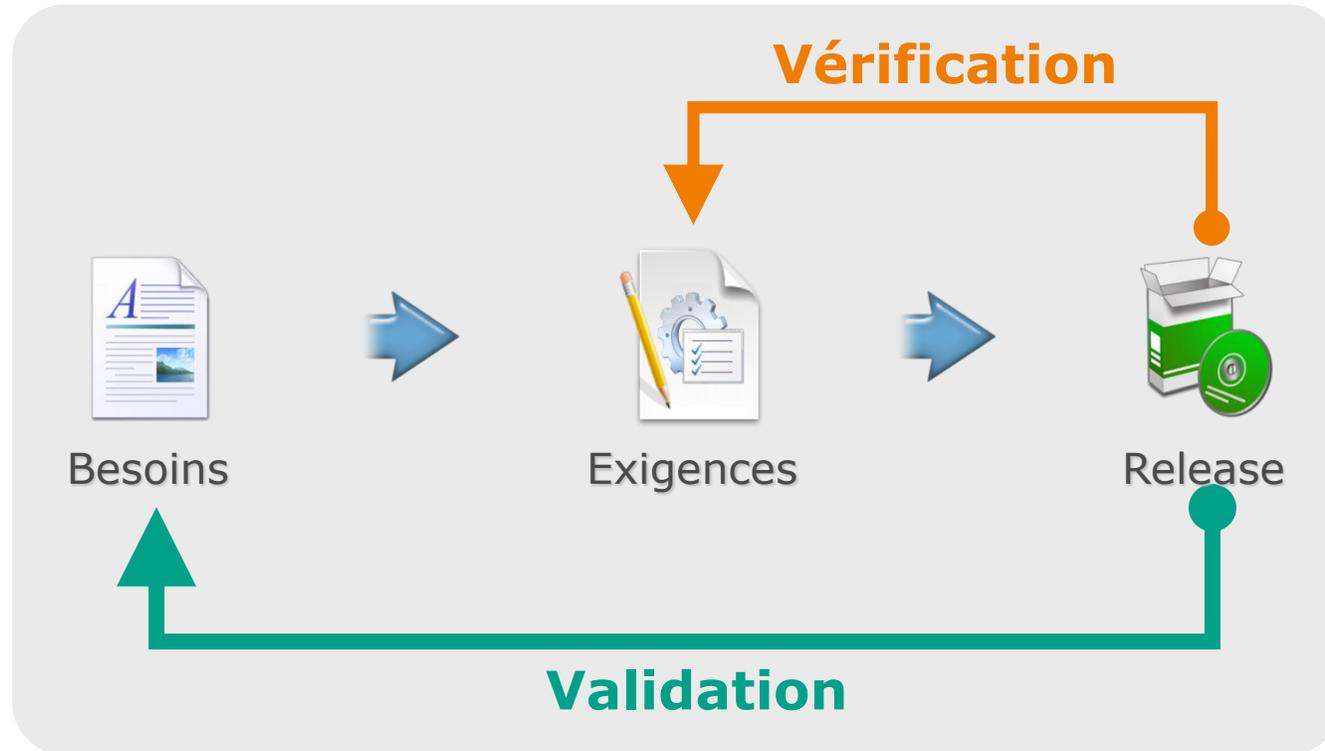




Une fois que l'on a déployé une version de la solution, comment on organise les tests de qualification ?

- Qui va tester ?
- Comment est-ce qu'il va tracer ses tests ?
- Quel niveau de granularité ?
- Qu'est-ce qui va être testé ?
- Doivent-ils être exhaustifs ?

Définition du test de qualification



Vérification : confirmation par l'examen et la fourniture de preuves objectives que des **exigences** spécifiées ont été remplies [ISO 9000]

Validation : confirmation par l'examen et la fourniture de **preuves** objectives que les exigences, pour un usage ou une application voulue, ont été remplies [ISO 9000]



Objectif des tests : réduire les risques
Ces Tests ne sont pas du « débogage »

Ces tests sont écrits en langage naturel, avec des étapes et sont réalisés concrètement sur l'application déployée, on s'abstrait du code. L'équipe de test prépare aussi les jeux de données associés

Les principes généraux des tests de qualification

— Les tests indiquent la présence de défauts

- └ Mais ne peuvent pas en prouver l'absence.
- └ Les tests diminuent la probabilité que des défauts restent dissimulés, mais n'est pas une preuve d'exactitude si aucun défaut n'est découvert.

— Les tests exhaustifs sont impossibles

- └ Tester toutes les combinaisons d'un logiciel est une tâche sans fin...
- └ Le compromis est de réaliser l'analyse des risques et des priorités pour pouvoir pondérer les efforts de tests.

Exemple : test de l'IHM suivante

- 20 écrans
- 4 menus / 3 options
- 10 champs
- 2 types de données
- 100 valeurs possibles

$$\begin{array}{r} 20 \\ \times 4 \times 3 \\ \times 10 \\ \times 2 \\ \times 100 \\ \hline = 480.000 \\ \text{cas de tests} \end{array}$$

Valeurs courante d'exécution

- Expert = 10 s
- Moyen = 1 mn
- Junior = 10 mn



55 jours
pour un
expert !



**Plutôt que des tests exhaustifs,
utilisons les risques et les priorités
pour focaliser les efforts de tests !**



Selon vous :

Qu'est ce que la couverture de test ?

Qu'est ce que la traçabilité des exigences ?

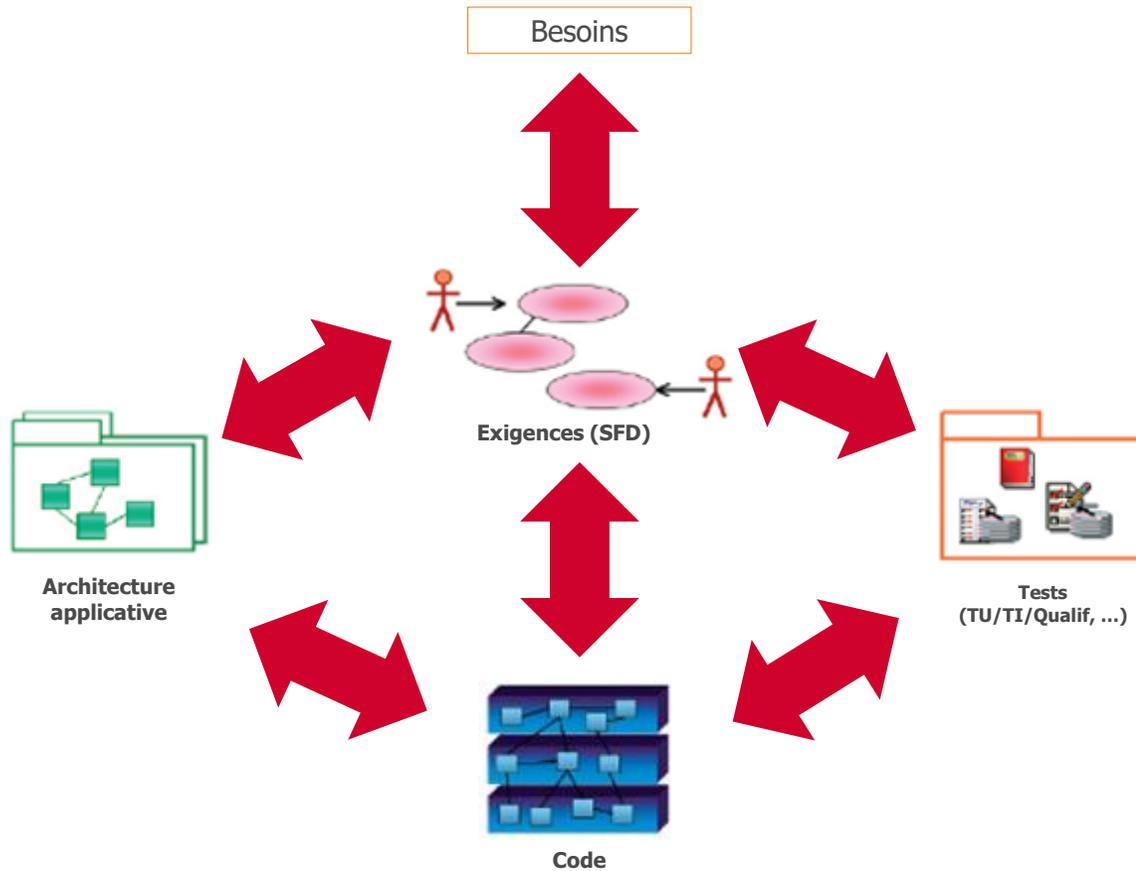
Pourquoi l'application de ces notions sont importantes ?

Couverture des tests

- **Mesurer la couverture des tests permet de fournir des indicateurs d'avancement de la qualification et de qualité de la solution**
 - └ Avancement de la spécification des tests : taux d'exigences couvertes par les cas de test
 - └ Avancement de l'exécution des tests : taux d'exigences testées
 - └ Qualité de la solution : taux d'exigences vérifiées OK

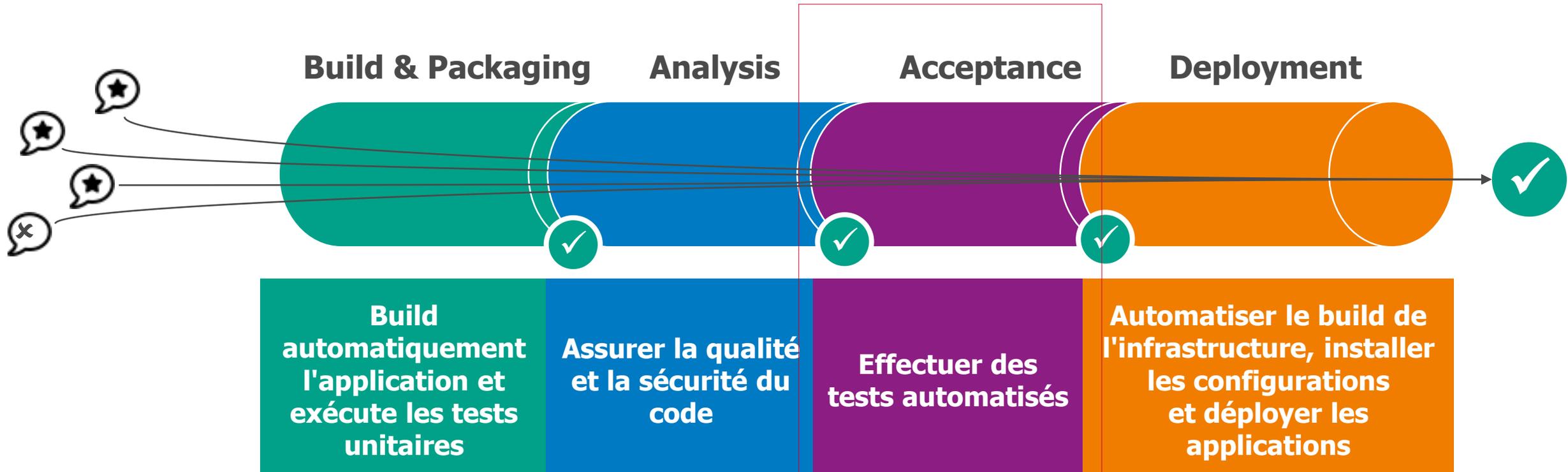
- **Optimiser l'effort de test**
 - └ Les exigences de test qui correspondent à des **fonctions critiques** (plus haut risque) doivent bénéficier des **meilleures couverture de test**
 - └ On veillera à ce que les éléments prioritaires obtiennent une couverture aussi complète que possible, avant de s'attaquer au reste des fonctionnalités

Traçabilité des exigences – Rôle pivot des Exigences



La traçabilité permet de mieux identifier les impacts sur l'ensemble de l'application

Rappels de la pipeline



Sur un projet industrialisé, une partie de ces tests peuvent être automatisés mais cela a un coût (notamment sur des tests de non-regression)

04

Tests de recette et mise en exploitation





Comment décider de transférer la solution au client ?
Qu'allons nous transférer au client ?
Quelles sont les actions du client ? Que teste-il ?
Quand donner le GO pour la mise en production ?

Tests de recette

Livrables à produire par l'entreprise :

- Le package de déploiement + toute la documentation (SFD, STD, Guides utilisateurs)
- Les tests qui ont été déroulés par l'entreprise et leurs résultats. Preuve de la couverture des exigences
- Les métriques qualité
- Bon de livraison avec le périmètre livré et les éventuelles réserves

Tests réalisés par le client :

- Déploiement sur un environnement client
- Avec des données complètes, finales (dump de production si existant), sans bouchons
- Nombre de tests très fluctuant en fonction des clients... A minima il réalise les tests passants de bout en bout et quelques cas d'erreur les plus fréquents. Scénarios métier d'un utilisateur final.
- L'entreprise fait le support au client : analyse et requalification des anomalies/« defects » rencontrés, correctifs, redéploiements réguliers
- Cette phase peut amener des demandes d'évolution → à développer par l'entreprise
- Réalisation de tests de perf avec données réelles

Une fois les tests réalisés, le client annonce le GO/NOGO pour la mise en production (MEP)

Mise en production

Pré-production :

- Souvent toutes les gestes de la MEP sont d'abord reproduites sur l'environnement de « Pré-production ». Cette étape est souvent réalisée par les équipes d'exploitation du client.
- Ces gestes doivent être automatisés au maximum pour éviter les erreurs humaines

MEP :

- Mise en production sur l'environnement final
- Si application existante : peut nécessiter un temps d'arrêt, limité
- L'entreprise reste disponible pour le client

Après la MEP :

- Conduite du changement et formations auprès :
 - └ Des utilisateurs
 - └ De l'équipe d'exploitation
 - └ De l'équipe de maintenance
- Délai de garantie (sur lequel l'entreprise s'engage à corriger les anomalies)
- Démarrage de la TMA : contrat de maintenance applicative (pour réaliser des correctifs et des évolutions)

Exemples de bugs marquants

Impact financier

Echec du premier lancement d'Ariane V

- 1996 : La fusée Ariane V explose en vol à son premier lancement
- Défaut : Logiciel de plate forme inertielle repris tel quel d'Ariane IV sans nouvelle validation. Ariane V ayant des moteurs plus puissants s'incline plus rapidement que Ariane IV. Les capteurs ont bien détecté cette inclinaison d'Ariane V, mais le logiciel l'a jugée non conforme au plan de tir (d'Ariane IV), et a provoqué l'ordre d'auto destruction
- Défaillance : Coût du programme d'étude d'Ariane V : 6 milliards d'€ et 10 années de travail, pour 39 secondes de vol....

Image de marque

La SNCF victime d'une vaste panne informatique

- Mai 2010 : Le système de réservation de billets SNCF est paralysé un vendredi début de week-end de la Pentecôte.
- Défaut : A l'origine de cette panne, selon le PDG, «le chargement de nouvelles données qui ont abîmé, entre guillemets, le système de réservation».
- Défaillance : Impossibilité d'acheter en gare (guichet, bornes automatiques) et sur le site Internet de la compagnie ferroviaire un voyage. Rétablissement de la situation le lendemain

Vies humaines

Un logiciel à l'origine d'accidents de la route

- Entre 2009 et 2011, la Lexus ES350 a été impliquée dans de nombreuses sorties de route à grande vitesse
- Défaut : Graves lacunes dans le fonctionnement de l'ordinateur de bord. Celui-ci pouvait enclencher l'accélération jusqu'à 150 km/h et désactiver la pédale de frein
- Défaillance : Entre 200 et 400 morts aux Etats-Unis.

Les mots clés du cours

Pyramide de test :
quantité, coût
et précision

Différents
niveaux de
tests et
acteurs
jusqu'au GO
MEP client

On ne peut pas
tout tester,
réduire les
risques

Intégration
continue
permet
d'automatiser
certains tests

Tracabilité et
couverture
autour des
exigences



05

Et le chef de projet ?

Le rôle du chef de projet

2

DIMENSIONS
COMPLÉMENTAIRES



Assurer la cohérence entre pilotage et ingénierie

Le rôle du chef de projet (liste non exhaustive)

Global :

- « Chef d'orchestre » de l'équipe
- Gère staffing, budget, planning
- Relation client
- Prise de décisions

Fonctionnel / Technique :

- Il participe à la vente du projet (si possible)
- Il doit connaître le fonctionnel général et l'architecture technique du projet
- Défendre des « sujet complexes » auprès du client

RH :

- Relation entreprise (commercial, directeur d'agence, direction industrielle), suivi de la marge, de la rentabilité du projet
- Gestion RH : staffing, suivi de la carrière des équipes
- Affecte les tâches ou les délègue aux référents technique et référents fonctionnels (en fonction de la taille de l'équipe)

05

Contrôle de connaissance : rappels

Contrôle de connaissance

- Sous la forme d'un examen final
- Questions ouvertes sur la compréhension du cours et un exercice de modélisation UML (diagramme de classe). Elles seront principalement basées sur les mots clés du cours (qu'il faut savoir expliquer)
- Pas d'accès au cours pendant l'examen
- Début Janvier (date à définir par l'école)

En attendant :

- Mise en place d'un Klaxoon avec QCM qui restera à dispo jusqu'à l'examen (sera positionné dans le Moodle, un mail sera envoyé a une personne de la classe)
- Evaluation du cours et axes d'amélioration via Klaxoon pour avoir vos feedback (sera positionné dans le Moodle également)