

# Cours Génie Logiciel – M1 Informatique

## Partie II – Test logiciel

### Chapitre 6 – Outils de support aux tests

# Structure du chapitre 1

---

## 6.1 Types d'outils de test (K2)

6.1.1 Outils de support aux tests (K2)

6.1.2 Classification des outils de test (K2)

6.1.3 Outils d'aide à la gestion du test et des tests (K1)

6.1.4 Outils d'aide aux tests statiques (K1)

6.1.5 Outils d'aide à la spécification des tests (K1)

6.1.6 Outils d'aide à l'exécution et à l'enregistrement des tests (K1)

6.1.7 Outils de support de performance et de surveillance (K1)

6.1.8 Outils de support pour des besoins de tests spécifiques (K1)

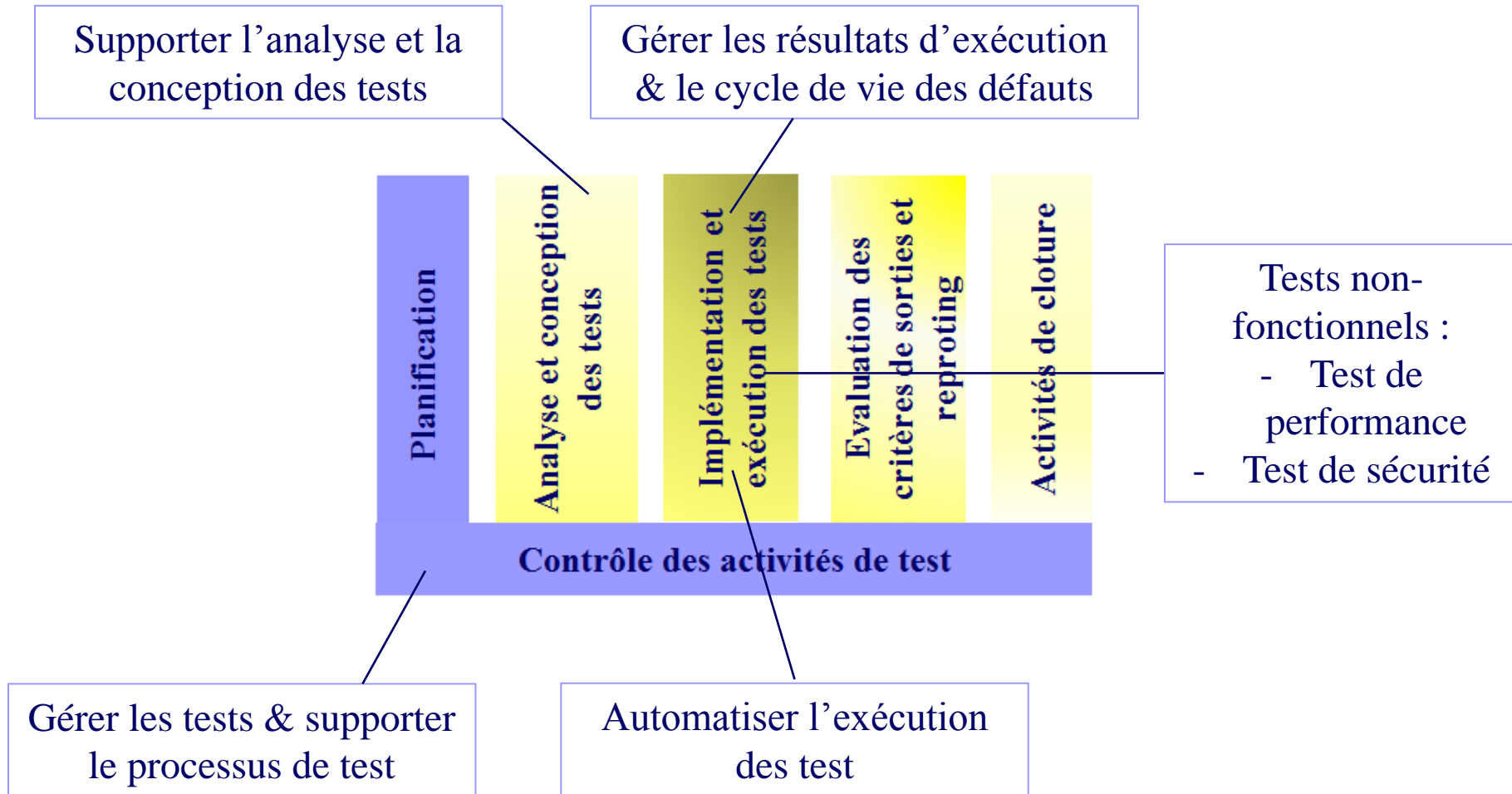
## 6.2 Utilisation efficace des outils : Bénéfices potentiels et Risques (K2)

6.2.1 Bénéfices potentiels et risques liés aux outils de test (pour tous les outils) (K2)

6.2.2 Considérations spéciales pour quelques types d'outil (K1)

## 6.3 Introduire un outil dans une organisation (K1)

# Les outils logiciels couvrent l'ensemble du processus de test - Exemples



# 6.1 Types d'outils de test (K2)

---

6.1.1 Outils de support aux tests (K2)

6.1.2 Classification des outils de test (K2)

6.1.3 Outils d'aide à la gestion du test et des tests (K1)

6.1.4 Outils d'aide aux tests statiques (K1)

6.1.5 Outils d'aide à la spécification des tests (K1)

6.1.6 Outils d'aide à l'exécution et à l'enregistrement des tests (K1)

6.1.7 Outils de support de performance et de surveillance (K1)

6.1.8 Outils de support pour des besoins de tests spécifiques (K1)

## 6.1.1 Outils de support aux tests (K2)

---

### Types d'usage

- ◆ Réalisation des tests (e.g. outils d'exécution, ou de génération de données de tests)
- ◆ Assistance au processus de test (e.g. gestion des tests, gestion des exigences, gestion des incidents...)
- ◆ Assistance au processus de développement logiciel en lien avec les tests (gestion de projet, gestion de configuration...)
- ◆ Autres outils utiles (e.g. feuille de calcul)

# Objectifs de l'utilisation des outils

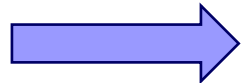
---

- ◆ Améliorer l'efficacité des activités de test par :
  - l'automatisation des tâches répétitives
  - l'aide aux activités de test manuelles (planification, conception, reporting, ...)
- ◆ Automatiser des activités qui nécessitent des ressources importantes
  - Exemple : Tests statiques outillés,
- ◆ Automatiser des activités qui ne peuvent pas être exécutées manuellement
  - Exemple : Tests de performance en charge
- ◆ Augmenter la fiabilité des tests
  - Limiter les erreurs humaines dans l'exécution des tests ou l'interprétation des résultats

## 6.1.2 Classification des outils de test (K2)

---

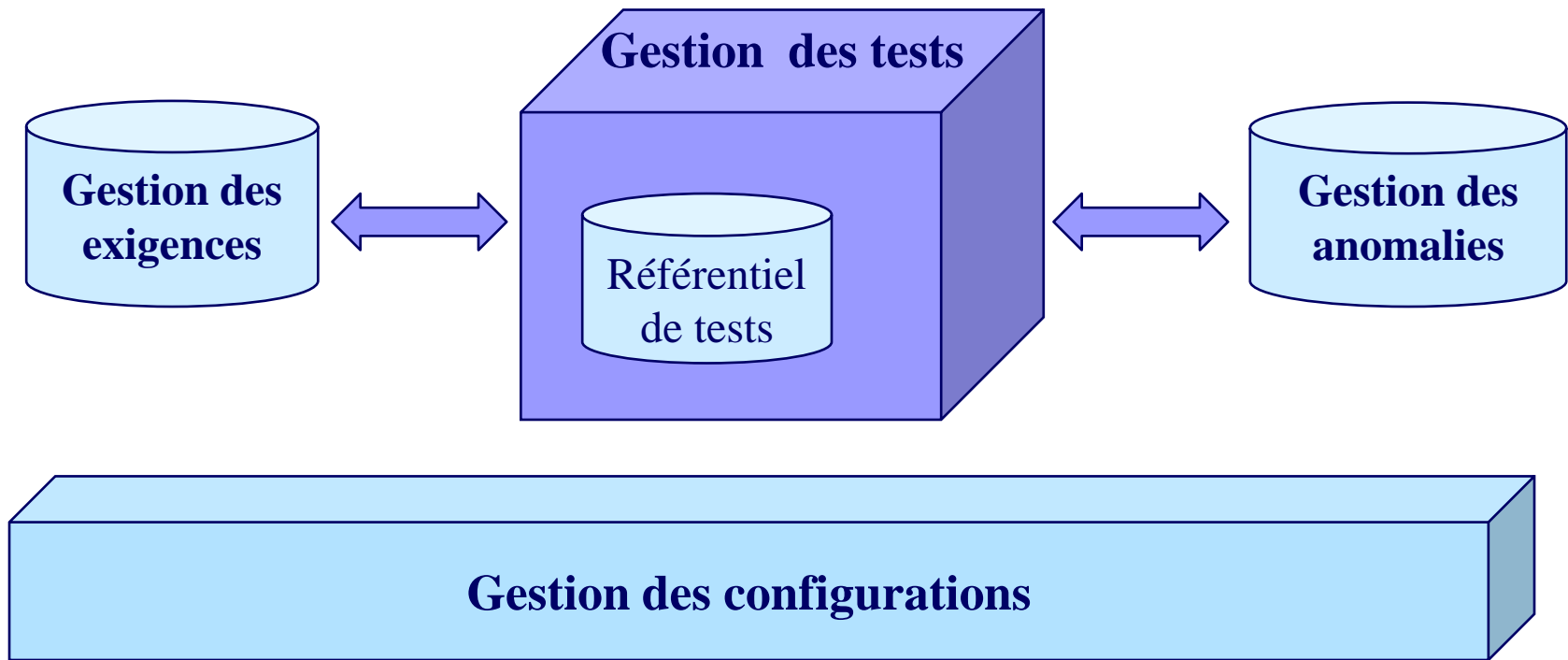
◆ Plusieurs classification possibles :



- A partir des activités de test supportées
- Open-source / Commercial
- A partir du types de technologies supportées
  - » Pour les outils d'automatisation de l'exécution

## 6.1.3 Outils d'aide à la gestion du processus de test et des tests (K1)

---



Le gestionnaire de test est le pivot du support au processus de test

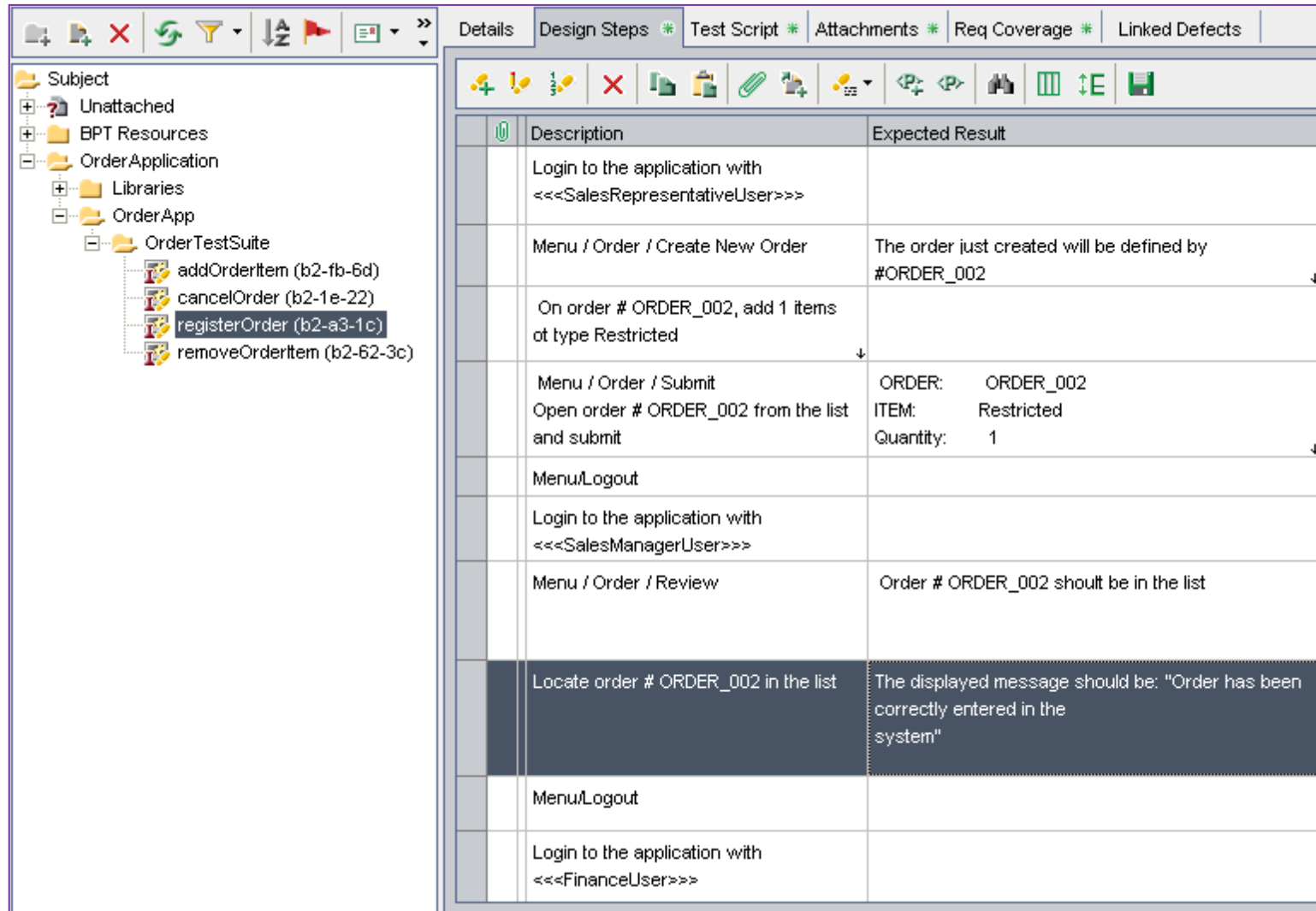


# Gestionnaire de test

---

- ◆ Gestion des cas de test (création, mise à jour, traçabilité)
- ◆ Interface avec des outils d'exécution des tests
- ◆ Interface avec des outils de traçage des défauts et de gestion des exigences
- ◆ Gestion des résultats des tests et synthèse des rapports d'avancement
- ◆ Facilitent la collecte des mesures et les améliorations des processus (tableaux de bord)
- ◆ Intégration avec la gestion de configuration

# Le gestionnaire de test – Exemple HP Quality Center



The screenshot displays the HP Quality Center interface for editing a test suite. On the left, a tree view shows the project structure: Subject > BPT Resources > OrderApplication > Libraries > OrderApp > OrderTestSuite. The 'OrderTestSuite' folder is expanded, showing five test steps: 'addOrderItem (b2-fb-6d)', 'cancelOrder (b2-1e-22)', 'registerOrder (b2-a3-1c)' (which is selected), and 'removeOrderItem (b2-62-3c)'. The main area shows the details of the selected test step, 'registerOrder (b2-a3-1c)'. The 'Design Steps' tab is active, displaying a table with two columns: 'Description' and 'Expected Result'. The table contains the following steps:

Description	Expected Result
Login to the application with <<<SalesRepresentativeUser>>>	
Menu / Order / Create New Order	The order just created will be defined by #ORDER_002
On order # ORDER_002, add 1 items of type Restricted	
Menu / Order / Submit Open order # ORDER_002 from the list and submit	ORDER: ORDER_002 ITEM: Restricted Quantity: 1
MenuLogout	
Login to the application with <<<SalesManagerUser>>>	
Menu / Order / Review	Order # ORDER_002 should be in the list
Locate order # ORDER_002 in the list	The displayed message should be: "Order has been correctly entered in the system"
MenuLogout	
Login to the application with <<<FinanceUser>>>	

# Gestion des tests – de nombreux outils disponibles

---



## 6.1.4 Outils d'aide aux tests statiques (K1)

---

- ◆ Outil d'aide aux revues
  - Check-list, suivi des revues, ...
- ◆ Analyse statique outillée du code
  - Lié aux langages / Multilangages
- ◆ Outils de vérification de modèle
  - Ex. sur UML

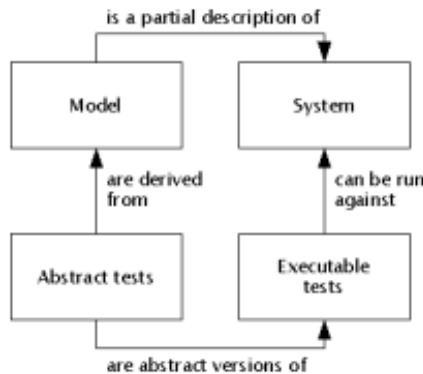


## 6.1.5 Outils d'aide à la spécification des tests (K1)

---

### ◆ Outils de conception de tests

- Exemple : génération de tests à partir de modèle – Model-Based Testing



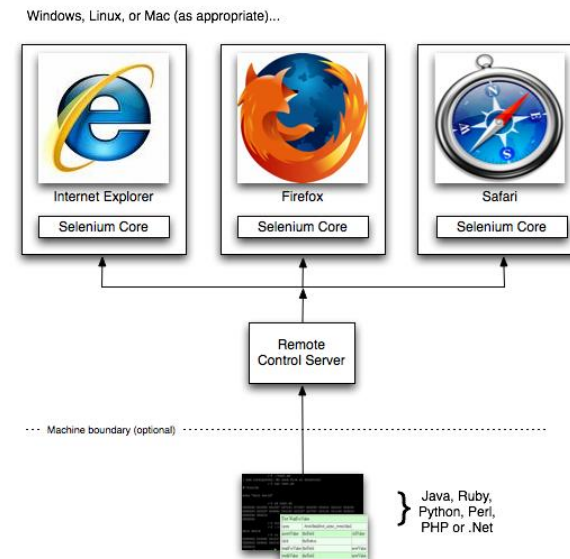
### ◆ Outils de préparations de données de tests

- Exemple : Pair-wise (cf [www.pairwise.org](http://www.pairwise.org) )

## 6.1.6 Outils d'aide à l'exécution / Automatisation des tests (K1)

- ◆ De très nombreux outils et framework d'automatisation de tests
  - En fonction des technologies & langages : ex. C#, C++, Java, ...
  - En fonction du type de système : Mobile / ERP / Web / Embarqué / ...
  - En fonction du type de tests : fonctionnel / sécurité / ...

Exemple : Open-source  
Selenium.  
Aspect multi-navigateur



## 6.1.7 Outils de support de performance et de surveillance (K1)

---

### ◆ Outils de test de performance - test de charge & test de stress

- Les outils de test de performance surveillent et rapportent sur la façon dont se comporte un système selon une grande variété de conditions d'usage simulées en termes de nombre d'utilisateurs simultanés, de leur modèle de montée en charge, de fréquence et de pourcentage relatif de transactions
- La simulation de la charge est réalisée au moyen de la création d'utilisateurs virtuels effectuant un ensemble choisi de transactions diffusées à travers divers machines de test identifiés comme injecteurs de charge

### ◆ Outils de surveillance (monitoring)

- Les outils de surveillance analysent continuellement, vérifient et rendent compte de l'utilisation de ressources systèmes spécifiques, et donnent des alertes sur de possibles problèmes de service

## 6.1 Les types d'outils de tests

---

- ◆ LO-6.1.1 Classer les différents types d'outils de test selon leur sujet et les activités du processus de tests et le cycle de vie logiciel. (K2)
- ◆ LO-6.1.3 Reconnaître les outils qui peuvent aider les développeurs dans leurs tests. (K2)



## 6.2 Utilisation efficace des outils : Bénéfices potentiels et Risques (K2)

---

6.2.1 Bénéfices potentiels et risques liés aux outils de test (pour tous les outils) (K2)

6.2.2 Considérations spéciales pour quelques types d'outil (K1)

## 6.2.1 Bénéfices potentiels et risques liés aux outils de test (pour tous les outils) (K2)

---

### ◆ Intérêt des outils :

- Réduction du travail répétitif
- Accroissement de la cohérence et de la répétitivité
- Objectivité
- Accès partagé à l'information



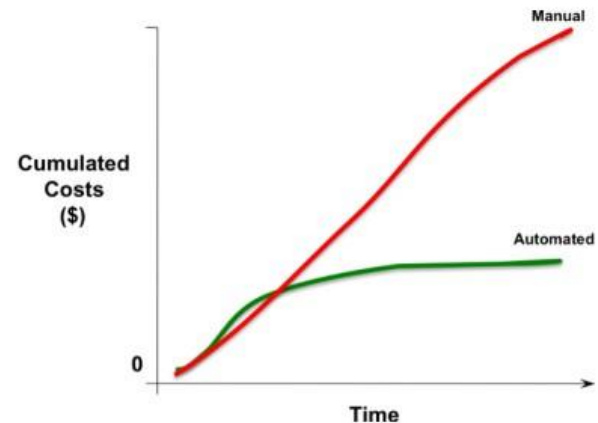
### ◆ Risques liés aux outils :

- Attentes irréalistes
- Sous-estimation du temps, du coût et de l'effort
- Confiance excessive dans l'outil de test (ex. test structurel automatique)
- Peu de support sur l'outil et / ou pas d'évolution
- Incapacité à supporter une nouvelle plateforme

## 6.2.2 Considérations spéciales pour quelques types d'outil (K1)

- ◆ Les outils de test, quelqu'importe leur catégorie, demandent un temps et un effort d'apprentissage pour en avoir le bénéfice
- ◆ Le ROI – Retour sur Investissement – d'un outil dépend de multiples facteurs, et en particulier l'adéquation de l'outil avec le système sous test, les compétences de l'équipe, le processus mis en œuvre, ect ...

Exemple : test automatisé



## 6.2 Utilisation efficace des outils : Bénéfices potentiels et Risques (K2)

---

- ◆ LO-6.2.1 Résumer les bénéfices et risques potentiels d'automatisation et d'utilisation d'outils pour les tests. (K2)
- ◆ LO-6.2.2 Reconnaître les considérations spécifiques pour les outils d'exécution des tests, l'analyse statique, et les outils de gestion de tests. (K1)

# Quiz – section 6.2

---

Q1 - Quels sont les bénéfices potentiels de l'utilisation d'outils pour mener les activités de test ?

- a) Plus grande qualité de code, diminution du nombre de testeurs nécessaires, meilleurs objectifs de test.
- b) Plus grande réutilisabilité des tests, diminution du travail répétitif, évaluation objective.
- c) Meilleure réactivité des utilisateurs, diminution du nombre d'exécutions de test, objectifs inutiles.
- d) Plus grande qualité de code, diminution du travail papier, moins d'objections aux tests.

## Quiz – section 6.2

---

Q2 - Qu'est-ce qui constitue un risque potentiel dans l'utilisation d'outils de support aux tests ?

- a) Des attentes irréalistes, on surestime les tâches pouvant être réalisées par l'outil.
- b) Prise en compte insuffisante de l'outil, on continue à faire des tests manuels alors qu'un outil d'exécution de test a été acheté.
- c) L'outil peut trouver des défauts qui n'existent pas.
- d) L'outil va répéter exactement les mêmes choses que celles faites la fois précédente.

# Quiz - Réponses

---

- ◆ Section 6.2
  - Q1  $\rightarrow$  b
  - Q2  $\rightarrow$  a

## 6.3 Introduire un outil dans une organisation (K1)

---

- ◆ Evaluation de la maturité de l'organisation, de ses forces et de ses faiblesses et identification des possibilités d'amélioration du processus de test par le support d'outils.
- ◆ Evaluation au regard d'exigences claires et de critères objectifs.
- ◆ Une preuve de concept, à l'aide d'un outil de test pendant la phase d'évaluation pour vérifier qu'il fonctionne efficacement avec le logiciel en cours de test et dans l'infrastructure courante ou pour identifier des modifications requises de cette infrastructure pour utiliser l'outil efficacement
- ◆ Evaluation du vendeur (aspects de formation, de support et de commerce y compris) ou des fournisseurs de service de support en cas d'outils non commerciaux
- ◆ Identification des exigences internes pour le soutien et la tutelle dans l'utilisation de l'outil
- ◆ Evaluation de besoin de formation par rapport aux compétences en automatisation de tests de l'équipe de test courante
- ◆ Evaluation du rapport coût/bénéfice basé sur un cas métier concret



# Exemples de critères d'évaluation pour un outil de gestion des tests

---

## I. Adéquation fonctionnelle :

- 10 critères : 1.Exigences 2.Cas de test 3.Pas de Test 4.Scénarios 5.Campagnes  
6.Exécutions 7.Anomalies 8.Transverse 9.Administration 10.Automatisat°

## II. Ergonomie et prise en main :

- L'outil est-il **facile à utiliser** ? A appréhender ?
- Le **workflow** de travail est-il efficace ? Favorise-t-il la **productivité** ?

## III. Qualité technique :

- L'outil est-il facile à **déployer** ? A exploiter ?
- Est-il performant ? Stable ?
- Est-il facilement **interfaçable** avec des systèmes tiers ?

## IV. Pérennité :

- La **communauté** des développeurs ou l'éditeur est-il actif ? Réactif ?
- L'outil est-il **populaire** ? La communauté des **utilisateurs** est-elle importante ? Active ?
- Quel est le mode gouvernance ?

## V. Coûts :

- Quel coût de **licence** ? De **support/maintenance** ?
- Quelles stratégies d'accès aux **versions majeures** ?

Source :

<http://www.clubqualitelogicielle.fr/index.php/component/jdownloads/finish/69/1667?Itemid=0>

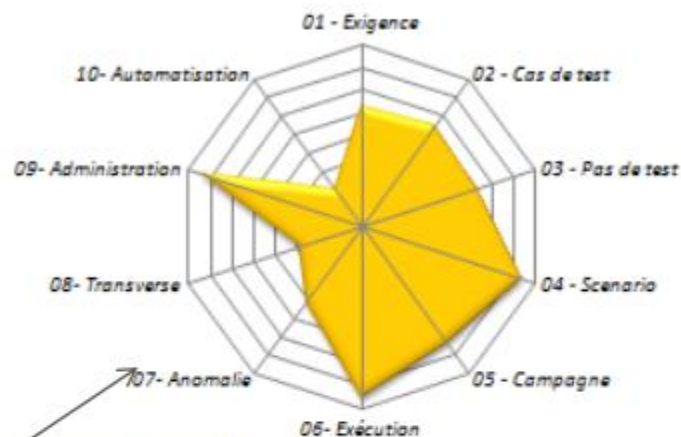
# FICHE D'OUTIL: TESTLINK 1.9.6

Informations	
ANNÉE DE CRÉATION	2005
MODÈLE	Open Source
VERSION	1.9.6 (version 1.9.9 sortie en novembre 2013)
EDITEUR	Communauté
COMMERCIALISATION	On-Premise
COÛTS D'ACHAT	Gratuit
SUPPORT / MAINTENANCE	Pas de support / maintenance professionnel
DÉPLOIEMENT	Client léger
ROADMAP	Pas de visibilité publique
RÉFÉRENCES	Grands Comptes (Ministères, Banque, Assurance)
PLATEFORMES EN PRODUCTION	Quelques dizaines d'utilisateurs concurrents

Périmètre	
GESTION DU PATRIMOINE	<input checked="" type="checkbox"/> Exigences <input checked="" type="checkbox"/> Cas de test <input checked="" type="checkbox"/> Campagnes
GESTION DES ANOMALIES	<input checked="" type="checkbox"/> Intégré <input checked="" type="checkbox"/> Interfacé JIRA, Mantis, Bugzilla, Trac
GESTION DE L'AUTOMATISATION	<input checked="" type="checkbox"/> Intégré <input checked="" type="checkbox"/> Interfacé Jenkins (réalisé pour Testlink)
PLUGINS / EXTENSIONS	

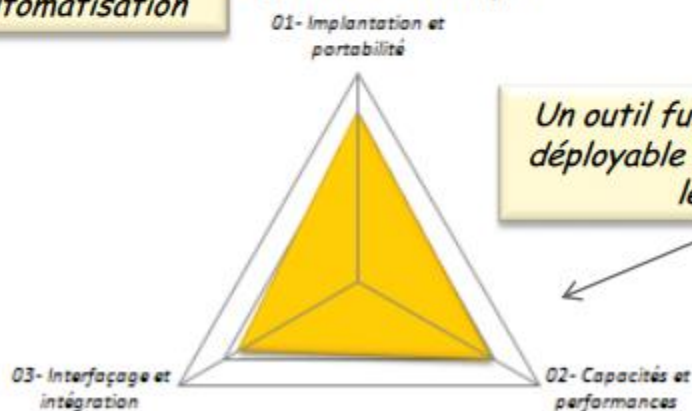
# LES RÉSULTATS : TESTLINK 1.9.6

## 01 - Adéquation fonctionnelle



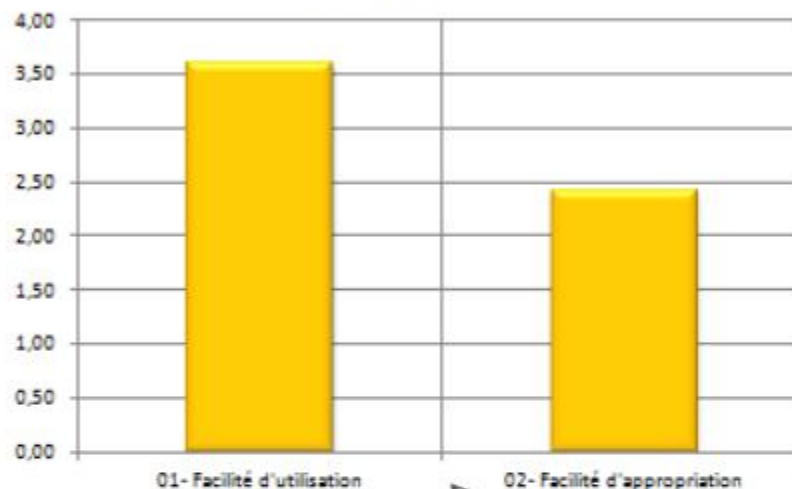
*Des manques sur la gestion transverse, des exigences, des anomalies et de l'automatisation*

## 02 - Qualité technique



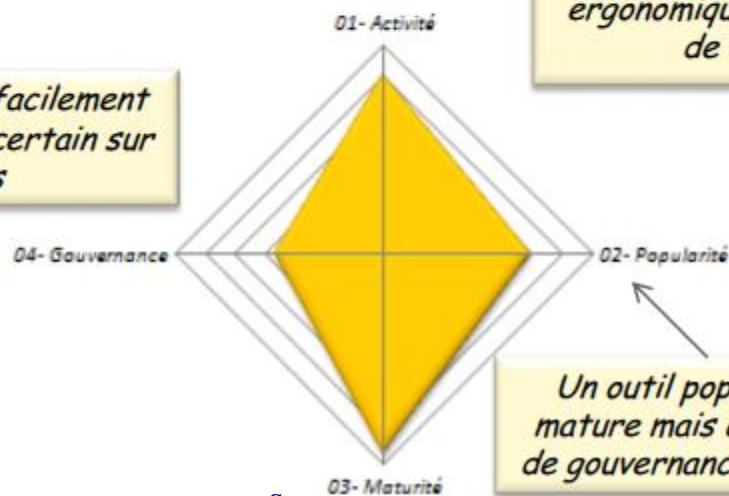
*Un outil full web, facilement déployable mais incertain sur les perfs*

## 02 - Ergonomie



*Un outil peu intuitif, avec quelques principes ergonomiques mais peu de doc.*

## 04 - Pérennité



*Un outil populaire, mature mais au mode de gouvernance opaque*

Exemple d'analyse Outil

Source : <http://www.clubqualitelogicielle.fr/index.php/component/jdownloads/finish/69/1667?Itemid=0>

# LES RÉSULTATS : TESTLINK 1.9.6



## POINTS FORTS :

- ✓ Communauté d'utilisateur importante (la plus active des outils open source)
- ✓ Technologie full web



## LIMITES :

- ❑ Workflow de travail déroutant, peu intuitif
- ❑ Projet peu actif (une release majeure tous les ~18 mois)
- ❑ Pas de support professionnel



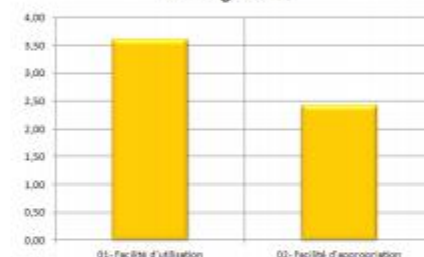
## POINTS DE VIGILANCE :

- ❑ Performance ? (modèle de données concentré sur une table principale, mélange de code objet et non objet)
- ❑ Committers sont peu nombreux et peu ouverts à l'intégration de développements issus de sociétés tierces.

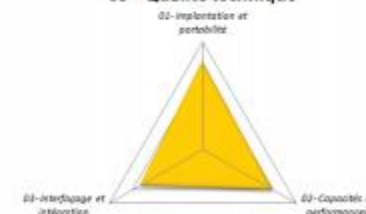
### 01 - Adéquation fonctionnelle



### 02 - Ergonomie



### 03 - Qualité technique



### 04 - Pérennité



# Phases d'évaluation de l'outillage de test

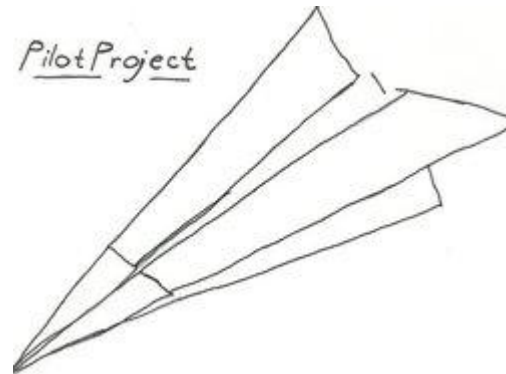
---

## ◆ POC – Proof of Concept

- Essai limité, sur un cas « exemple », qui permet de se faire une première idée de l'outil

## ◆ Projet Pilote

- Mise en œuvre sur un cas réduit, mais avec mesure de métriques permettant d'avoir des données représentatives des bénéfices de l'outil





# Phases d'évaluation outillage de test - Pilote

---

- ◆ Apprendre l'outil plus en profondeur.
- ◆ Voir comment l'outil s'adapte à des processus et pratiques existants, et comment ces derniers devraient évoluer.
- ◆ Décider d'une manière standard d'utiliser, de gérer, de stocker et de maintenir l'outil et le testware (p.ex. décider d'une convention de nommage pour les fichiers et les tests, créer des bibliothèques et définir la modularité des suites de tests).
- ◆ Evaluer si les bénéfices escomptés seront atteints pour un coût raisonnable

# Démarche de déploiement - Facteurs de succès de l'intégration d'un outil

---

- ◆ Etendre l'outil au reste de l'organisation de façon incrémentale
- ◆ Fournir de la formation et une assistance aux nouveaux utilisateurs
- ◆ Établir des guides d'utilisation.
- ◆ Surveiller l'utilisation de l'outil et les bénéfices recueillis
- ◆ Fournir le support pour l'équipe de test pour un outil donné
- ◆ Recueillir l'expérience acquise de toutes les équipes

## 6.3 Introduire un outil dans une organisation (K1)

---

- ◆ LO-6.3.1 Exposer les principes majeurs liés à l'introduction d'un outil dans une organisation. (K1)
- ◆ LO-6.3.2 Exposer les objectifs d'une preuve de concept pour l'évaluation d'un outil et une phase de pilotage pour l'implémentation d'un outil. (K1)
- ◆ LO-6.3.3 Reconnaître que des facteurs autres que l'acquisition d'un outil sont nécessaires pour un bon support des tests par les outils. (K1)



# Certification ISTQB / CFTL

---

L'ISTQB a été fondé à Edinbourg en novembre 2002.

Il s'agit d'une organisation de certification à la qualification des systèmes d'information.

L'ISTQB comprend 48 représentations nationales (à juillet 2014).

Le Comité Français des Tests Logiciels (CFTL) a été fondé en 2004 pour couvrir la France et les pays francophones qui ne possèderaient pas de comité national ISTQB.

# Certification ISTQB / CFTL

---

Les éléments traités dans le syllabus niveau Fondation :

1. Fondamentaux des tests
2. Tester pendant le cycle de vie logiciel
3. Techniques statiques
4. Techniques de conception de tests
5. Gestion des tests
6. Outils de support aux tests

# Certification ISTQB / CFTL

---

Le format de l'examen est un questionnaire à choix multiples (QCM)

Il comprend :

- 40 questions
- 4 réponses possibles pour chaque question
- 1 seule réponse correcte par question

Il faut répondre correctement à au moins 26 questions pour réussir l'examen de certification, soit 65% de réponses correctes)

Durée de l'examen : 1 heure

# Répartition du temps de formation et des questions par chapitre

---

	<i>Time (min)</i>	<i>Time</i>	<i>Number of Questions</i>	<i>Range</i>	<i>K1 (+/- 1)</i>	<i>K2 (+/- 1)</i>	<i>K3/K4 (+/- 1)</i>
Chapter 1	155	18%	7.17	7 to 8	4	3	
Chapter 2	115	13%	5.32	5 to 6	4	2	
Chapter 3	60	7%	2.77	2 to 3	2	1	
Chapter 4	285	33%	13.18	12 to 15	4	2	6
Chapter 5	170	20%	7.86	7 to 8	3	3	2
Chapter 6	80	9%	3.70	3 to 4	3	1	
Total	865	100%	40	40	20	12	8