

L'OPEN-SOURCE, UNE ALTERNATIVE VIABLE POUR LES ACTIVITÉS DU TEST



OVERVIEW

- 1 | L'open-source, une tendance de fond dans le monde du logiciel
- 2 | Quelles contraintes pèsent sur le choix de la solution ?
- 3 | Une solution open-source intégrée pour une utilisation facilitée
- 4 | D'autres variantes d'intégrations sont possibles
- 5 | Avantages et limites de la solution open-source proposée

CONTEXTE

- Il faut tester plus et mieux à budget constant
 - Tests sur mobiles et tablettes, sur de multiples versions de navigateurs
 - Tester en plusieurs points du cycle: intégration, tests système, pré-production ...
- Les outils de test open-source sont de plus en plus largement utilisés
- Les demandes de REX de clients à propos d'outils open-source se multiplient
 - Remettre en cause un coût de licence élevé
 - Remettre en question les fonctionnalités réellement utilisées
 - ➔ Il faut être capable de répondre à ces demandes **de manière construite**
- Fait suite à la présentation des JFTL 2015 sur la mise en place: « Choix et déploiement d'un outil de test : Retours d'expérience sur les spécificités de l'open source »



L'OPEN-SOURCE, UNE TENDANCE DE FOND DANS LE MONDE DU LOGICIEL

Un choix pertinent pour les activités de test

L'ESPRIT DE L'OPEN-SOURCE

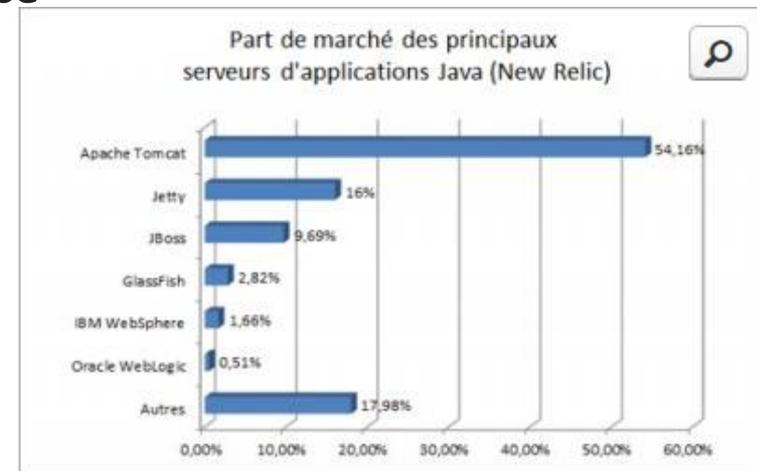
- Communauté et partage
 - Un code source disponible, modifiable et réutilisable
 - Une communauté de contributeurs participant au projet en fonction de leurs moyens
 - Développement
 - Documentation, tutoriels, forums
 - Remontée d'anomalies
- Pour l'utilisateur expérimenté, une maîtrise totale du logiciel
 - Des formats de fichiers ouverts et interopérables
 - Une sécurité importante (audit de code, réactivité face aux bugs critiques)
 - Possibilité d'améliorer le logiciel et de reverser l'évolution à la communauté
- L'open-source n'empêche pas une utilisation commerciale
 - Support
 - Fonctionnalités supplémentaires et services cloud

LA PÉRENNITÉ EST UN ENJEU MAJEUR DE L'UTILISATION DE LOGICIELS OPEN-SOURCE

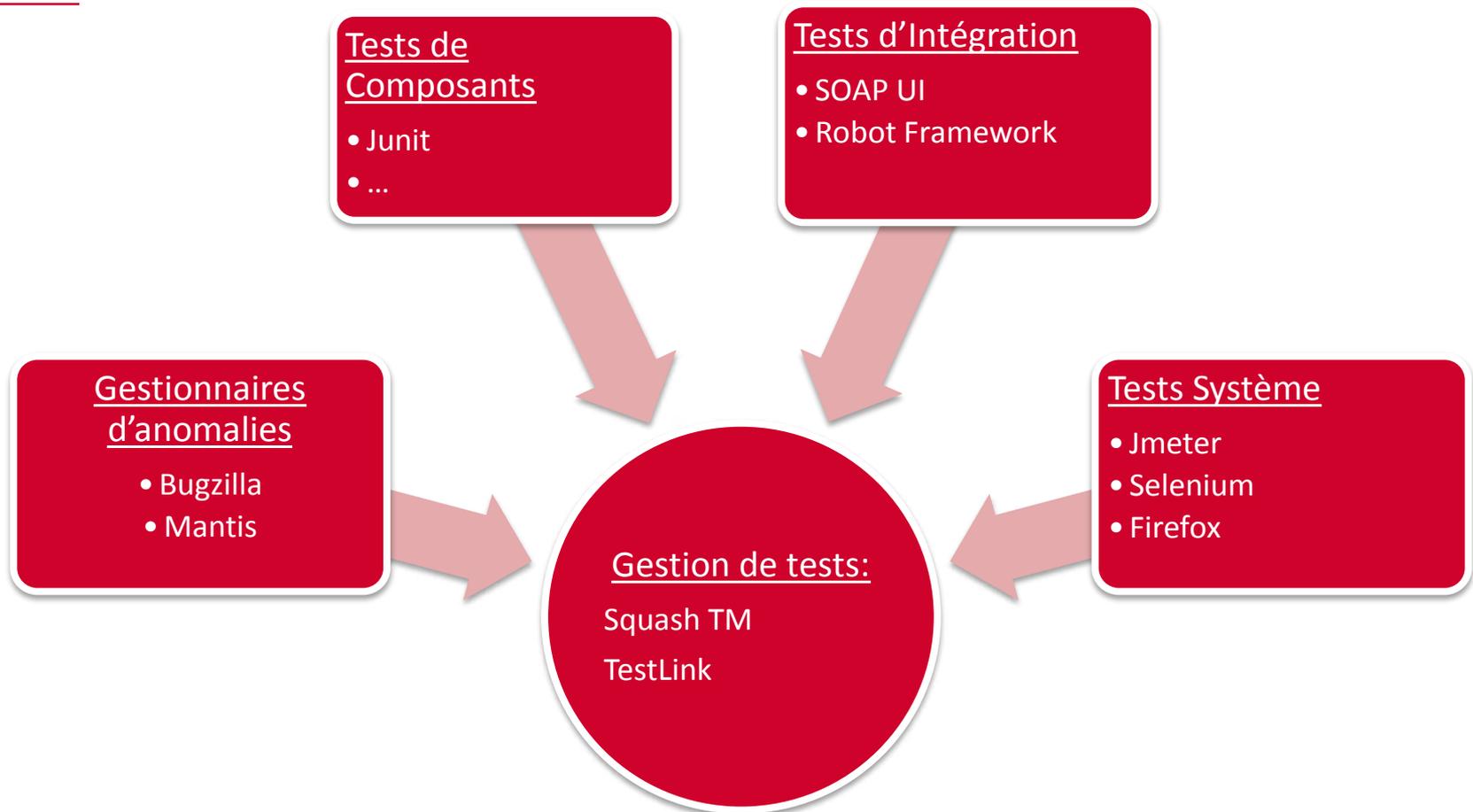
- Le risque majeur est un abandon du projet
 - Perte des bénéfices de l'open-source (évolutions, sécurité)
 - Mais il est toujours possible de maintenir / reprendre le projet
- L'utilisation de projets open-source nécessite une analyse sur:
 - La popularité du projet: nombreux utilisateurs, existence de formations
 - Le soutien éventuel d'une fondation ou entreprise
 - L'activité du projet: âge du projet et nombre de contributions récentes
- Même le soutien d'une entreprise ou d'une fondation ne protège pas toujours de ce risque:
 - La fondation Mozilla arrête le développement de Firefox OS (12/2015)

L'OPEN-SOURCE GAGNE EN CRÉDIBILITÉ

- Près de 90% des entreprises utilisent du logiciel open-source
 - Coûts réduits à fonctionnalités équivalentes
 - L'auto-formation est facilitée
- Les institutions mettent l'accent sur le logiciel libre
 - La gendarmerie a migré ses postes vers Linux (2011) et Open-Office (2004)
 - Publication du SILL (Socle Interministériel des Logiciels Libres)
 - Le Parlement Européen prend une résolution pour favoriser le logiciel libre
- Google ou Facebook misent sur l'open-source en rendant disponible nombre de projets dont ceux sur l'intelligence artificielle (décembre 2015)
- Les serveurs open-source Tomcat et Apache, sont très largement utilisés



LE DOMAINE DU TEST VOIT AUSSI ÉMERGER DES SOLUTIONS OPEN-SOURCE DE QUALITÉ ET VIABLES



- Certaines solutions open-source ont une version sous licence pour des fonctionnalités accrues (SOAP UI, Squash, ...)

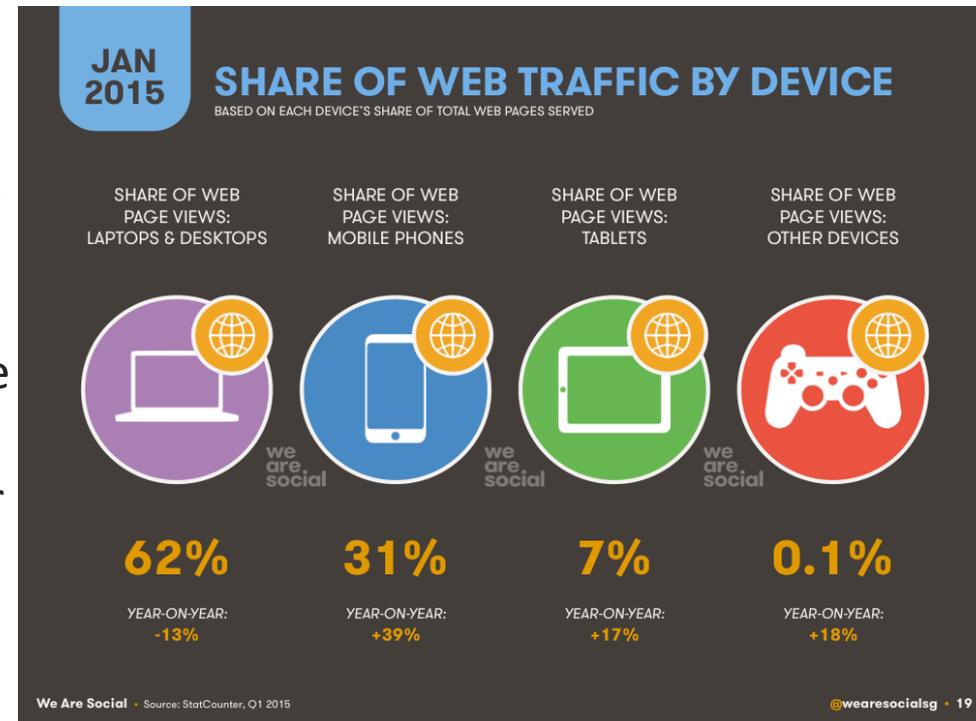


QUELLES CONTRAINTES PÈSENT SUR LE CHOIX DE LA SOLUTION ?

pérennité, évolutivité, modularité

LA SOLUTION D'AUTOMATISATION DOIT ÊTRE RÉACTIVE POUR COUVRIR LE PÉRIMÈTRE WEB

- Le domaine du Web est en forte évolution depuis quelques années
 - Sites en responsive design
 - Refontes régulières de l'ergonomie
 - Applications mobiles
- La concurrence entre navigateurs se traduit par de nouvelles versions régulièrement (tous les 2 mois pour Chrome et Firefox).
- Les plates-formes mobiles évoluent vite (~1 version majeure par an)



UNE SOLUTION DE TEST ADAPTATIVE PERMETTRA DE COLLER AU MIEUX À L'ENVIRONNEMENT CLIENT

- Les environnements de test des clients sont multiples. Exemple:
 - Logiciels propriétaires pour la gestion des tests (HP QC)
 - Logiciels maisons + QTP pour l'automatisation des tests
 - Jira pour la gestion de projet et des incidents
- Le caractère open-source permettra d'adapter plus facilement le logiciel à l'environnement (ex: connecteurs vers des outils tiers)
 - Via une contribution directe
 - A travers le financement du projet (ou commande à l'éditeur / un tier)
- Les logiciels open-source se basent sur des formats de données standards
 - Simplification de l'exploitation de ces données dans d'autres outils (ex: reporting)
 - Migrations facilitées

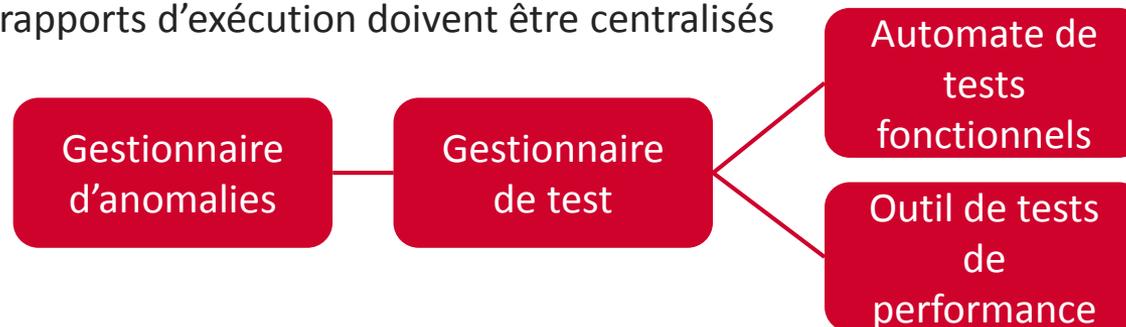


UNE SOLUTION OPEN-SOURCE INTÉGRÉE POUR UNE UTILISATION FACILITÉE

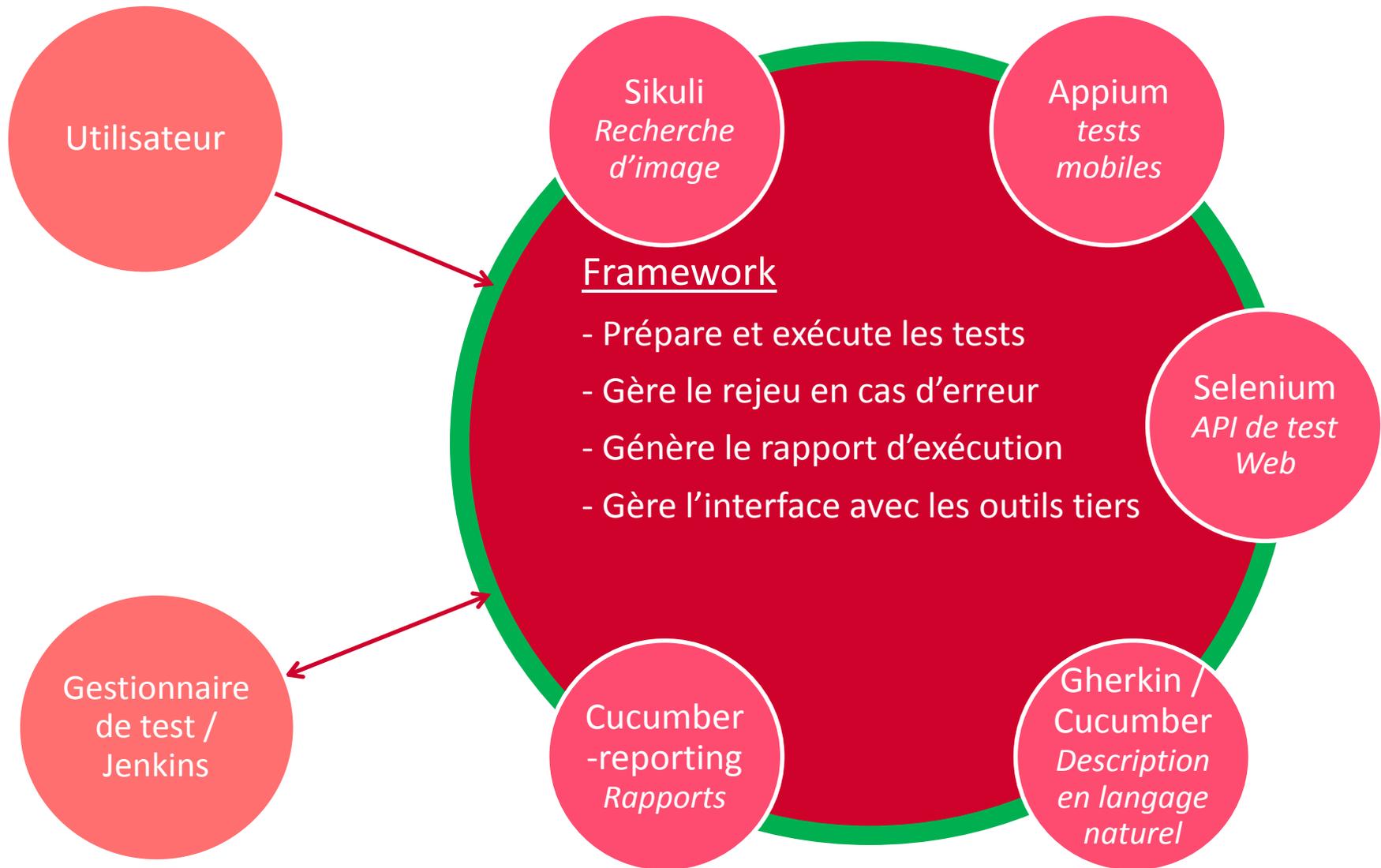
Un exemple de solution mise en œuvre dans le domaine du Web

QUELS OUTILS PEUVENT INTÉGRER LA SOLUTION DE TEST ?

- Besoin initial: automatiser les tests d'un site web / d'une application mobile
 - De nombreux outils existent (Selenium, Watir, Sahi, ...) mais n'ont pas les caractéristiques souhaitées
 - Usage industriel: un automate unique pour tous les tests fonctionnels
 - Description des tests en pseudo code
 - Enregistrement automatique de Vidéos
 - Tests de plate-forme mobiles aisé
- Une volonté: faire bénéficier nos clients de ces travaux
 - Il faut s'intégrer dans l'écosystème supporté par notre Direction Industrielle: Jmeter, Selenium, SpiraTest, HP QC
 - On présente au client une solution complète
 - Il faut associer l'automate à un gestionnaire de tests et d'anomalies
 - Les rapports d'exécution doivent être centralisés

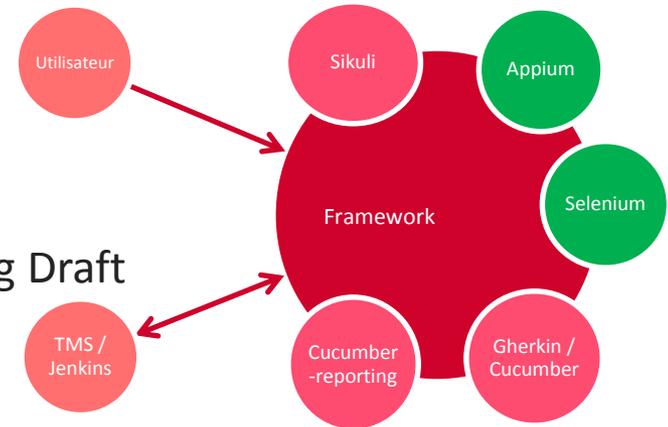


L'AUTOMATE COMBINE LA PUISSANCE DE NOMBREUX OUTILS OPEN-SOURCE

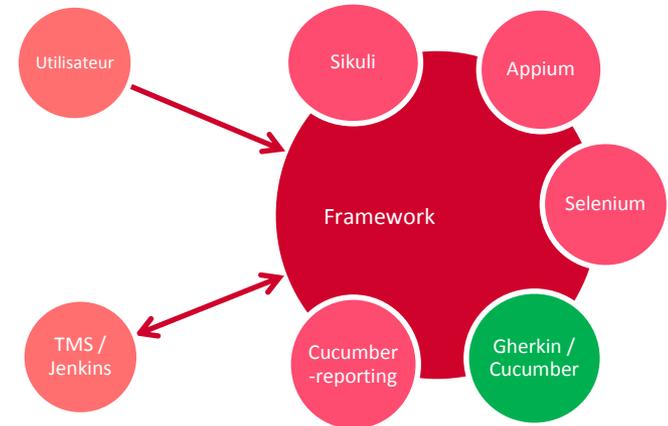


L'AUTOMATE EST BASÉ SUR SELENIUM, L'API PHARE POUR TESTER LE WEB

- Selenium Webdriver (<http://seleniumhq.org>)
 - Multi-navigateur (évolue à leur rythme)
 - Webdriver est un standard de fait: W3C Working Draft
 - Selenium IDE permet d'enregistrer (et rejouer) des scénarios depuis Firefox => écriture simplifiée des test
 - Selenium est le plus connu et le plus complet des outils d'automatisation de tests Web
 - La disponibilité du code source permet de comprendre / adapter le comportement de Selenium lorsqu'il ne fonctionne pas avec un site web
- Appium (<http://appium.io>)
 - Apporte le support des plates-formes mobiles à Selenium **sans recompilation**
 - Unifie les API d'automatisation intégrées à iOS et Android
- Gestion possible du parc d'automates via Selenium-Grid ou Jenkins



GHERKIN / CUCUMBER APPORTENT LA DESCRIPTION DES TEST EN LANGAGE NATUREL



@Vente1

Feature: Vente en ligne de poissons de la sélection à la co

Scenario: VEL01 Vente en ligne

Given Sur la page 'Accueil' URL '\$url\$'

When ACCU01 Cliquer sur le bouton 'Enter the store'

Then La page 'Menu Principale' s'affiche

When MENU01 Cliquer sur la catégorie 'Fish'

Then La page 'Détail Fish Disponible' s'affiche

Fichier .feature

Description possible par un non technicien

```
@Given("^Sur la page 'Accueil' URL '(.*?)'")
public void goAccueil(String url) {
    driver.manage().timeouts().implicitlyWait(2, TimeUnit.SECONDS);
    connect(url);
}
```

```
@When ("^ACCU01 Cliquer sur le bouton 'Enter the store'$")
public void cliquerBtContinuer() {
    webDriver.findElement(By.LinkText("Enter the Store")).click();
}
```

```
@Then("La page 'Détail Fish Disponible' s'affiche")
public void surLaPageFish() {
    WebElement elt = webDriver.findElement(By.xpath("//*[@id='Catalog
org.junit.Assert.assertTrue(elt.getText().equals("Fish"));
}
```

```
public void AccueilAvecMessage(String message) {
```

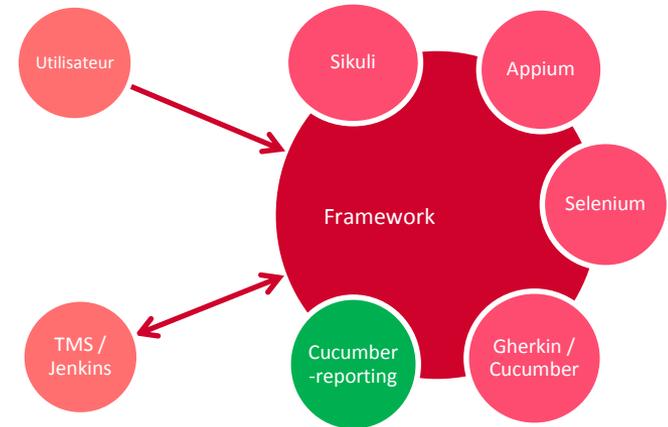
```
@When("MENU01 Cliquer sur la catégorie 'Fish'")
public void cliquerCategorieFish() {
    webDriver.findElement(By.xpath("//*[@id='SidebarContent']/a[1]"));
}
```

Fichier Java

Implémentation par un automaticien

CUCUMBER-REPORTING A ÉTÉ AMÉLIORÉ POUR LES BESOINS DE L'AUTOMATE

Feature	Scenarios			Steps					
	Total	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Unde
Affichage accueil	2	2	0	8	8	0	0	0	



@accueil1

Feature: Affichage accueil

Scenario: ACCUEIL01 Affichage accueil

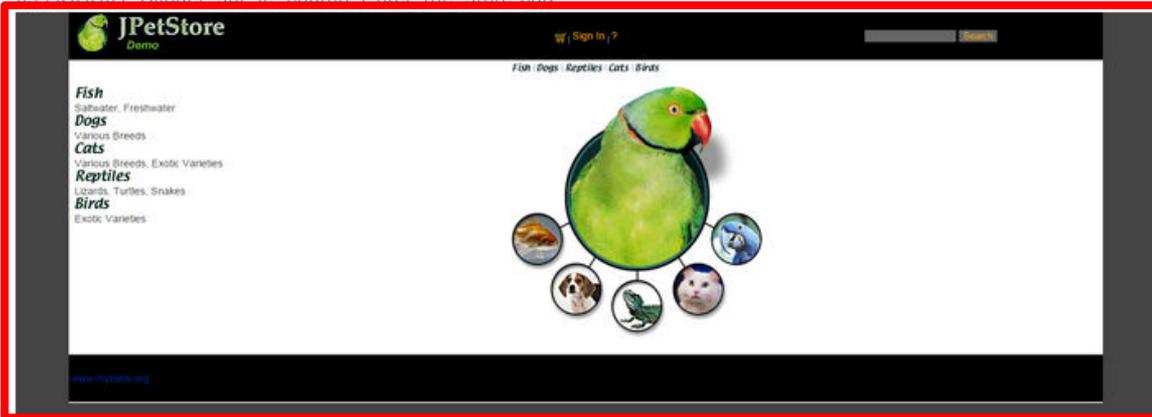
Given Sur la page 'Accueil' URL 'http://192.168.56.101:8090/jpetstore'

02-Sur_la_page_Accueil_URL_http-192.168.56.101-8090jpetstore.png

When ACCU01 Cliquer sur le bouton 'Enter the store'

01s 320ms

03-ACCU01 Cliquer sur le bouton 'Enter the store.png'



Changements apportés au projet initial

Then La page 'Menu Principale' s'affiche

01s 449ms

Then Finalisation du scenario courant

000ms

04-capture_video.avi

05-dernier_etat.png

06-traces.zip

07-networkTraces.zip

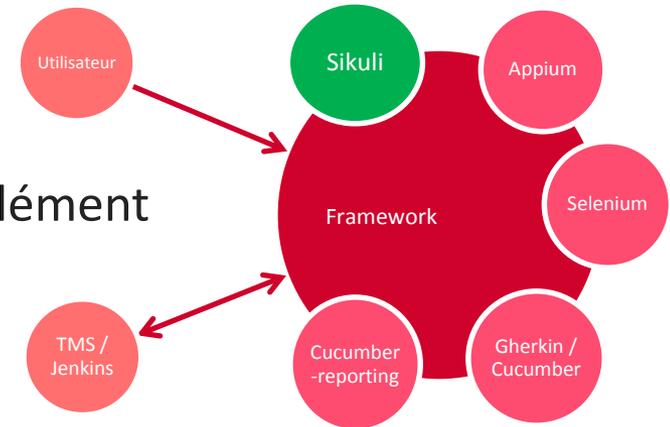
System.out

```

2015-12-23 14:14:06,373 INFO [automate] -----
2015-12-23 14:14:06,374 INFO [automate] Test ACCUEIL01_Affichage_accueil
2015-12-23 14:14:06,374 INFO [automate] -----
  
```

SIKULI APPORTE LA RECHERCHE D'IMAGES DANS LES PAGES WEB

- Utile lorsque Selenium ne peut pas trouver un élément dans une page
 - Recherche de l'image de l'élément
 - Actions clavier / souris
- Utile pour les technologies non supportées par Selenium
 - Flash
 - Silverlight
 - Applet Java
- Permet des contrôles « visuels »
 - Ne pas en abuser car la maintenance est plus coûteuse



QU'APPORTE L'OPEN-SOURCE À L'AUTOMATE DE TEST ?

● Gains

- Coût réduit (120 j/h de développement) par rapport à des licences d'outils propriétaires
- Adaptation du comportement de Selenium là où il ne parvenait pas à agir sur un élément (outil propriétaire également inopérant)
- Intégration de multiples fonctionnalités autour de Selenium
- Compatible avec les dernières versions de navigateurs là où QTP demande une mise à jour de licence

● Extensible

- OpenCV: comparaison d'images
- BrowserMob Proxy: tests sous contrainte réseau, analyse des temps de chargement des pages

Implantations
+20 pays

Implantations
+23 pays

⊕ GET assertselenium.cor	200 OK	assertselenium.cc	10 KB	1.28s
⊕ GET remote-login.php?	200 OK	r-login.wordpres	20 B	642ms
⊕ GET style.css?m=1337&	200 OK	s2.wp.com	4.3 KB	592ms
⊕ GET _static??/wp-inclu	200 OK	s1.wp.com	7.4 KB	632ms
⊕ GET global-print.css?m	200 OK	s0.wp.com	1.9 KB	495ms
⊕ GET static??/wp-inclu	200 OK	s0.wo.com	37.1 KB	1.02s

- Contrôles métiers (PDF, Emails, BDD)

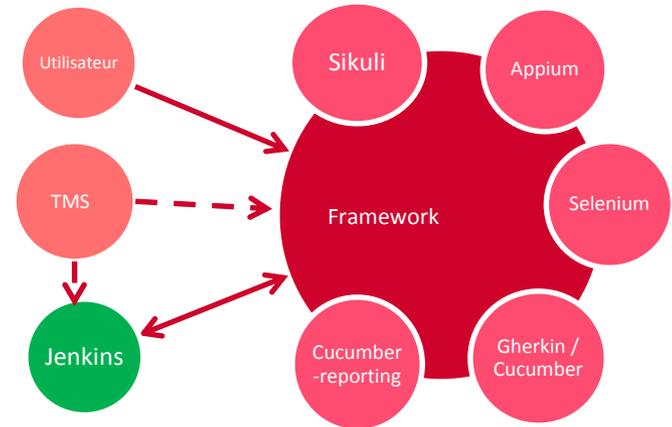
● Coûts

- Maintenance afin de rester à jour vis-à-vis des évolutions de navigateurs et OS mobiles

JENKINS, LE CHEF D'ORCHESTRE DES OUTILS DE TEST

- Logiciel initialement destiné à être utilisé en intégration continue

- De très nombreux plugins
- Permet aussi de faire le lien entre plusieurs outils dédiés au test: HP QC, Jmeter, Selenium, Squash TM, TestLink



- Facilite l'utilisation de l'automate en dehors de l'équipe test
 - Vérifier la disponibilité d'une plate-forme à intervalles réguliers
 - Réutilisation des briques de test en intégration, pré-production, production
 - Ex: Effectuer un suivi de production
 - Centralise les IHM des outils

JMETER, UN OUTIL DE TEST DE PERFORMANCE COMPLET ET PERSONNALISABLE

- Des fonctionnalités complètes pour les tests de performance
 - Architecture distribuée pour des injections massives
 - Mesure des indicateurs de charge serveur (perfmon agent & JMX)
 - Des services existent pour utiliser Jmeter en Cloud
- Un plugin Jenkins permet d'automatiser le lancement
 - Le test de performance peut être ajouté au processus d'intégration continue
- De nombreux plugins disponibles
 - Permettent l'extension des fonctionnalités de base (reporting, protocoles: SSH, SFTP, ...)
 - jmeter-plugins (open-source)
 - Plugins commerciaux pour des protocoles spécifiques (Vidéo, Adobe, ...)
- S'interface avec Selenium
 - Utilisation de scripts BeanShell / Javascript pour lancer des actions navigateur

SQUASH TM / TESTLINK: DES GESTIONNAIRES DE TESTS COMPLETS

	Squash TM 	TestLink 
Projet	Open-source fonctionnalités complémentaires sous licence (LDAP, lien Redmine, ...)	Projet Open-source complet disposant d'une forte notoriété
Support	Via l'éditeur de Squash, Forum	Forum uniquement
Personnalisations	Architecture modulaire: plugins Possibilité de réalisation par l'éditeur	Contributions complexes et code peu clair
Automatisation	Via Squash TA, lien bidirectionnel (Selenium, Sahi, ...)	Avec Jenkins, lien montant uniquement (via API REST).

- Squash TM préféré à TestLink dans la solution pour:
 - L'intégration native de l'automatisation
 - Une ergonomie plus soignée
 - Le support de l'éditeur, notamment pour les évolutions
 - Les multiples demandes clients à son propos
 - Une roadmap claire

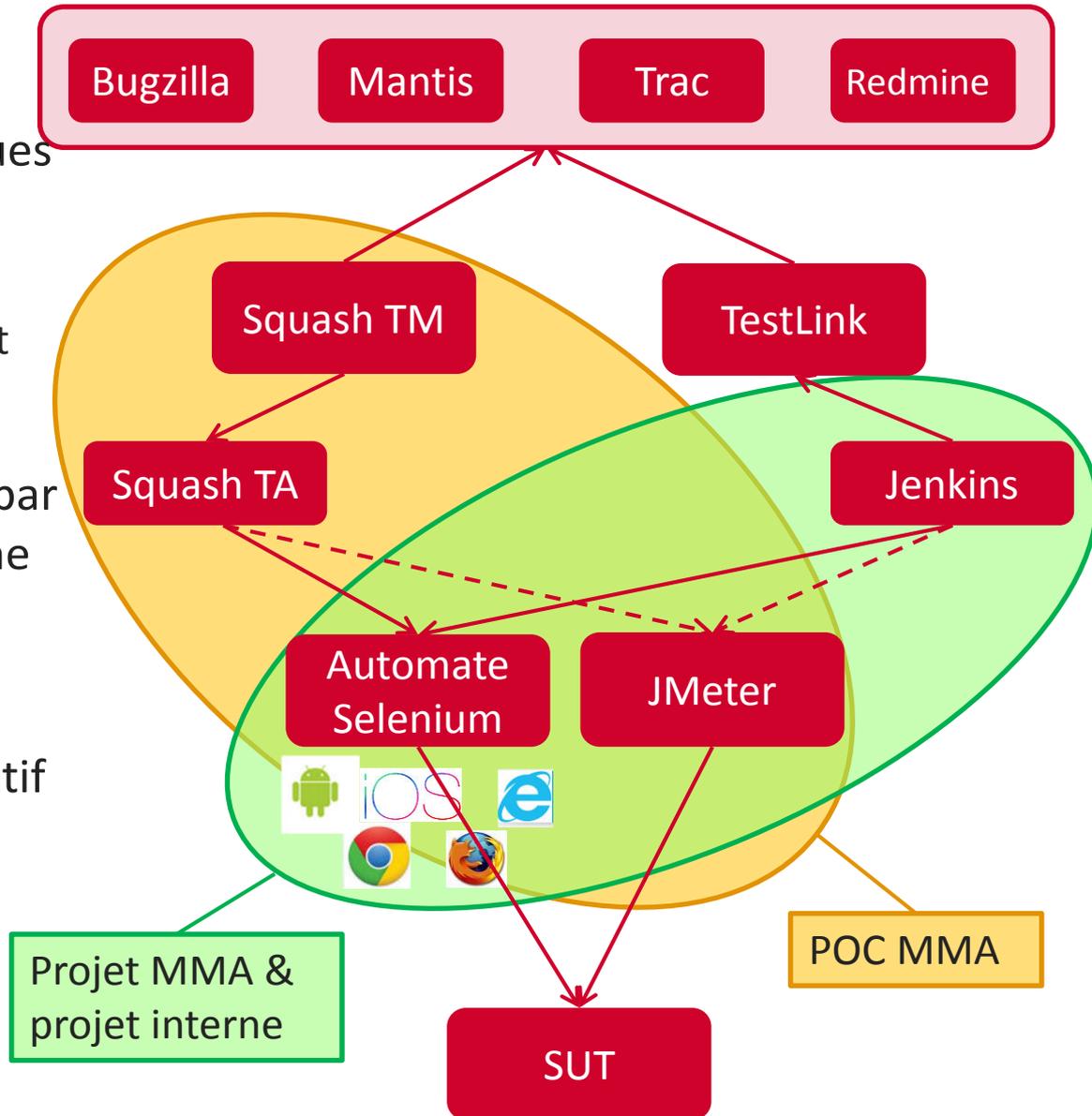
LE CHOIX DU GESTIONNAIRE D'ANOMALIES EST LARGE

- Squash TM et TestLink s'interfaçent notamment avec Bugzilla, Mantis, Redmine, Trac
- Fonctionnalités de gestion d'anomalies équivalentes. Le choix se fera sur:
 - Les besoins spécifiques clients
 - Les fonctionnalités annexes (ex: formatage des textes, plugins, wiki, ...)

Gestionnaire d'anomalies pur		Logiciel de gestion de projet	
<p>Bugzilla</p> 	<p>Pour: multi-projets recherche des possibles doublons nombreux plugins et outils tiers</p> <p>Contre: intégration avec les gestionnaires de source complexe</p>	<p>Trac</p> 	<p>Pour: possibilité de formater les descriptions de tickets</p> <p>Contre: ergonomie basique</p>
<p>Mantis</p> 	<p>Pour: intégration avec les gestionnaires de source intégration d'un Wiki</p> <p>Contre: ergonomie basique</p>	<p>Redmine</p> 	<p>Pour: possibilité de formater les descriptions de tickets</p> <p>Le plus populaire dans sa catégorie</p>

LA SOLUTION OPEN-SOURCE COUVRE LES BESOINS COURANTS DU TEST

- Toutes les solutions retenues sont largement utilisées
 - Des formations existent
 - Les documentations sont nombreuses
- Certaines sont soutenues par des entreprises du domaine du test (édition, cloud, ...)
- Chaque élément pris individuellement est évolutif (plugins, ...)





D'AUTRES VARIANTES D'INTÉGRATIONS SONT POSSIBLES

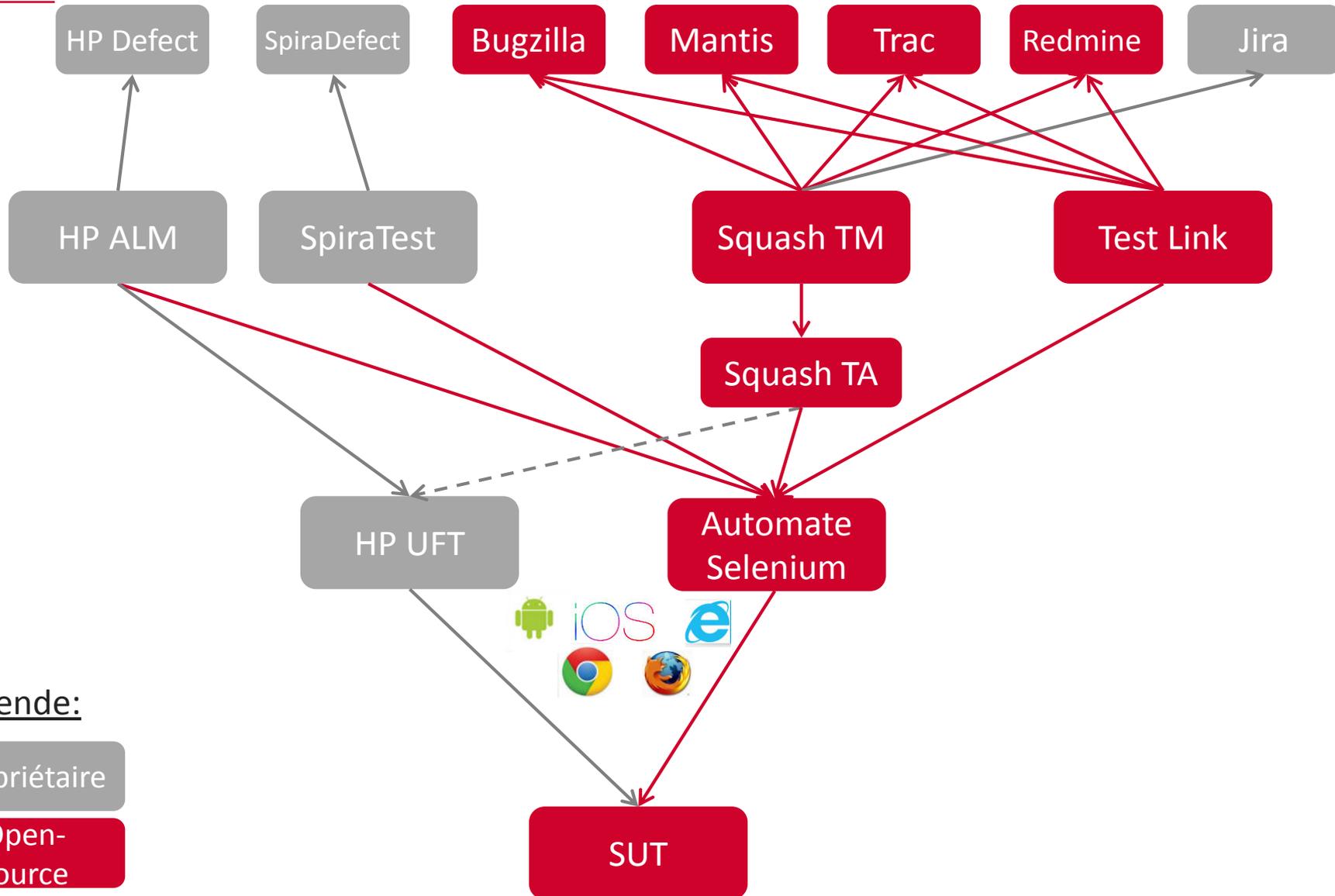
S'adapter aux besoins des projets

LES LOGICIELS OPEN-SOURCE ET PROPRIÉTAIRES SONT COMPLÉMENTAIRES

- C'est le choix du client qui guidera l'intégration
 - Disponibilité d'autres outils devant être conservés, dans la chaîne de test
 - Compétences nécessaires:
 - Disponibilité de compétences localement sur une solution particulière
 - Accessibilité de formations / tutoriels
 - Besoins non couverts par l'une des solutions
 - Le budget alloué guidera la recherche de la solution optimale

- Il faut prendre le meilleur des 2 mondes. Exemple:
 - Automate Selenium pour son coût et sa rapidité d'évolution
 - HP ALM car il possède des fonctionnalités avancées, indisponibles dans des logiciels open-source

LES SOLUTIONS POSSIBLES SONT MULTIPLES





AVANTAGES ET LIMITES DE LA SOLUTION OPEN-SOURCE PROPOSÉE

LA SOLUTION DE TEST OPEN-SOURCE DISPOSE DE BEAUCOUP D'ATOUTS

Forces

- **Adaptation fine** aux besoins des clients (outils choisis et développements spécifiques possibles)
- **Prise en main facilitée**: nombreux tutoriels, forums, ...
- **Coût réduits (attention aux coûts cachés!)**:
 - Le coût de licence n'augmente pas avec le nombre d'utilisateur
 - Capitaliser les efforts d'automatisation entre plusieurs phases de test (réutilisation des briques)
- **Évolution rapide**: corrections de bugs, fonctionnalités.
- **Services professionnels basés sur les solutions les plus populaires**: formations, cloud, fonctionnalités spécifiques
- **Pérennité** des solutions choisies

Faiblesses

- **Moins de fonctionnalités** qu'une solution propriétaire haut de gamme
- **Intégration des outils moins poussée**
 - Ex: Pas de gestion de parc de terminaux mobiles en natif
 - Différences ergonomiques entre les outils
- **Il faut assurer la maintenance de la solution**, au même titre que d'autres outils internes et donc s'organiser en conséquence (personnel, support externe)

Opportunités

- **Unifier les pratiques** entre la solution test mobile HP et la solution open-source
- Vendre un service plutôt qu'une licence tierce
 - Des **formations**
 - Des **personnalisations** de la solution globale (reporting spécifique, analyses BI, ...)
- Adresser des clients souhaitant migrer leur solution propriétaire

Menaces

- **Instabilité possible** liée à l'intégration de plusieurs outils (Selenium / Appium / Squash TM)
 - Mise en place de tests de non-régression
- **Dépendance** envers des projets externes pour la prise en charge de nouveaux OS (ex: Windows Phone) ou navigateurs (ex: Edge)
 - Il est possible d'investir pour animer ces projets
 - Remplace la dépendance à une mise à jour logicielle d'un éditeur

CONCLUSION

- Tous les aspects du test peuvent être couverts par de l'open-source
 - Fonctionnalités
 - Fiabilité
 - Pérennité
- L'automatisation des tests est ce qui bénéficie le plus du monde de l'open-source
 - Individuellement, de nombreux outils de qualités existent
 - Intégrés au sein d'un automate, la solution devient très puissante
 - L'évolution face aux changements technologiques est l'aspect le plus important
- La formation des étudiants au test logiciel passe par l'open-source



QUESTIONS