JFTL 2019

Intégrez vos tests en mode SAAS

Retour expérience Espace Client Orange

orange™

Emmanuel LEON, Sébastien Alonzo, Romain Louveau 03/04/2019, v 1.8

Qui sommes-nous ?

Avez-vous imaginé le test « Aas » ?

ACTIVITÉS / PROJETS

Des offres aaS pour son besoin
tests univers Web, Mobiles, TV
Industrialisation tests
Support et Méthodologie
Capitalisation / formation

FUN FACT

- Cellule Expertise Transverse

- Proof of Concept

TECHNO / TOOLS

Labo Tests as A Service
Gitlab / Gitlab CI
Selenium web Driver / Robotframework
Jenkins
Automatisation
HP ALM, JIRA XRAY
virtualisation Android et IOS
TV Orange / Raspberry PI

Une cellule d'expertise Transverse de tests Logiciels reconnue

Présence chez Orange Depuis 10 ans. Notre entité est la DSI DFY

Un Labo tests as a Service pour 300 utilisateurs

Sommaire

Le care, un écosystème unifié orange.fr/sosh.fr et applications Orange & Moi et MySosh

Notifications unifiées

Kick-off projet Janv 2018

Insight Client : J'aimerais qu'Orange m'informe de mes actualités sur l'ensemble de mes contrats de façon proactive et centralisée **parce que** je souhaite avoir la bonne information au bon moment et je ne veux manquer aucune information concernant mes services **mais** je ne suis pas informé par Orange ou je ne sais pas où retrouver les informations.

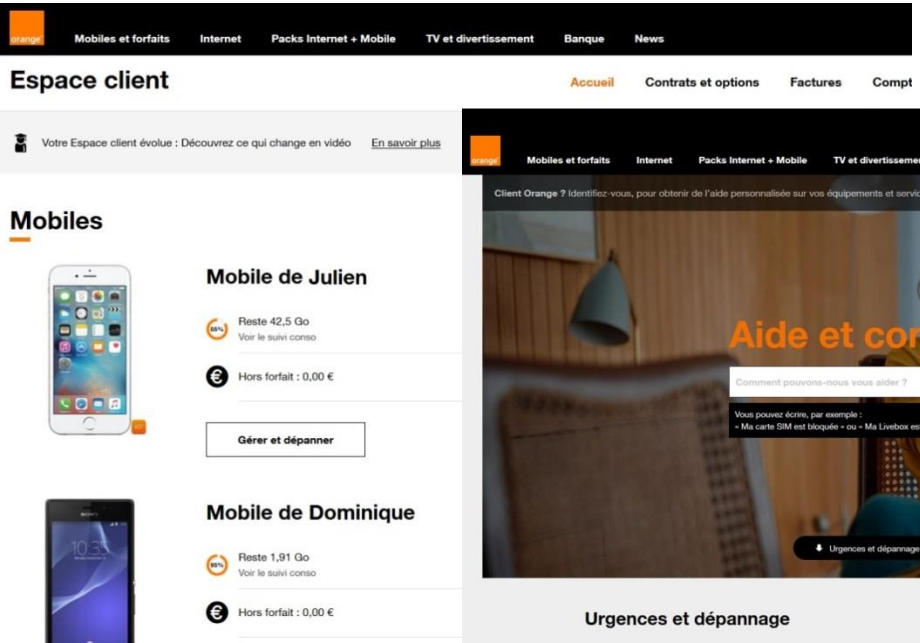
La promesse client : mettre le client au cœur du dispositif et le rendre autonome pour retrouver les informations qui le concernent dans un endroit unique et transverse à l'écosystème orange.fr/sosh.fr et aux applications Orange et moi et MySosh.

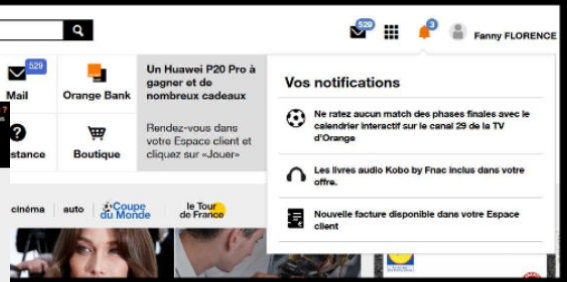
Objectifs at

Clients
4M VU

Résultats a

Clients
2,5M VU





Méthodologie SAFE Feature team

- Scaled Agile Framework, organisé par Train. Sprints synchronisés sur le contenu et non sur le rythme
- Equipe autonome orientée produit permettant la prise de risque et l'initiative

Technologie logicielle Single Page Application (SPA)

- Le test s'exécute sur le navigateur client
- Toutes les interactions avec le serveur sont des appels AJAX (JSON format)
- Appels à des micro-services REST
- Application plus fluide et Responsive

Outils d'intégration continue et de tests

- Git / GitlabCI, yarn, webpack, karma, protractor
- KARMA Javascript test runner : « cross-browser compatibility testing »
- Runner Gitlab : gestion de job et lancement automatique

GitLab

Outils de reporting de tests

- JIRA, Allure (framework) de rapports d'exécutions de tests
- Notion « Dashboard, catégories, suites »

JIRA

Outils de notification

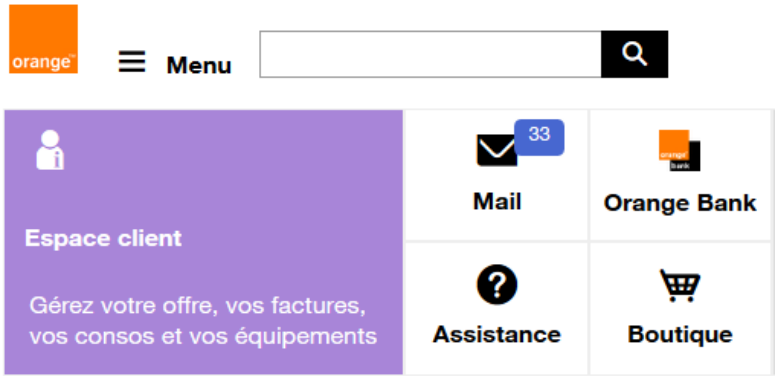
- Gestion de canaux, déclenchement sur étape Build)

Mattermost

1

Présentation du contexte

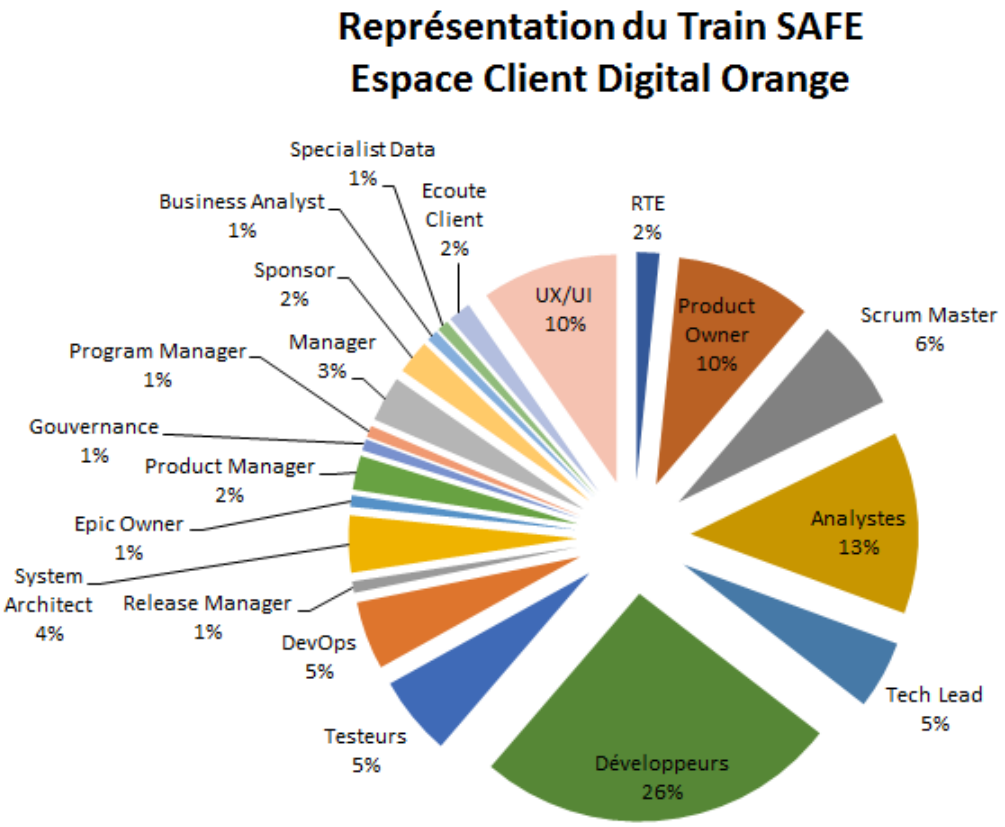
Equipe socle et architectes, processus et demande métiers avec le couple Product Owner / Release Manager et avantages, applications et environnements déployés.



Représentation du Train SAFE pour l'ECD

Des applications déclinées par contenu

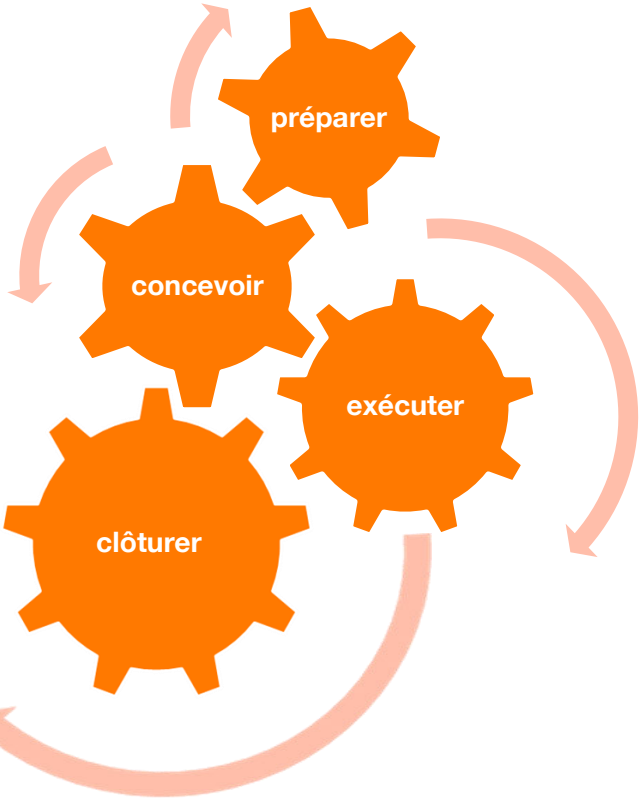
Des profils nécessaires, qui se regroupent en communautés pour assurer l'alignement du produit Espace Client



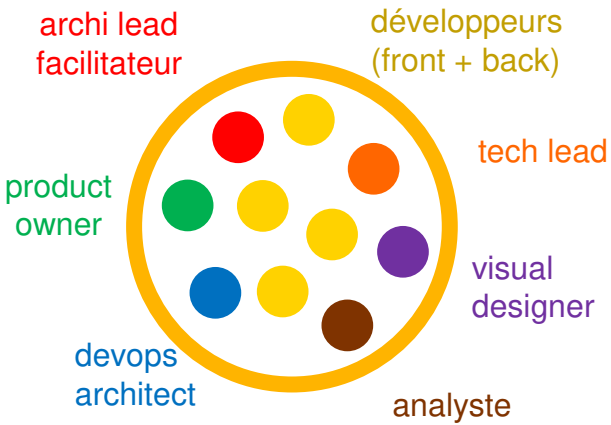
Attente des métiers : gagner en satisfaction client via des retours d'expérience

Une stratégie opérationnelle pour les DSI désirant « consommer » dans le Cloud des offres techniques de tests multi-OS multi-navigateurs multi-QoS

Un SI Labo Tests as a Service fonctionnant via une approche structurée de ses tests logiciels :



Organiser l'alignement du produit espace client par feature team



rôles additionnels en fonction du type et de la maturité de la Fteam:

- data analyst
- pilote processus, marketing

Un Backlog unique

accueil	suivi conso	facture	contrat	suivi commande	(...)

communauté des PO : aligner les backlogs et les MEP, traiter les dépendances

communauté des tech leads / développeurs : faire évoluer les règles d'architecture et de codage, diffuser les bonnes pratiques, organiser des revues de codes collectives

communauté des analystes : diffuser les bonnes pratiques, synchroniser les tests transverses, diffuser les principes fonctionnels,

communauté des devops-architects : faire évoluer les chaînes de déploiement continu et le monitoring, contribuer aux pratiques Cloud ready

équipe de pilotage de la valeur : piloter la roadmap globale, coordonner l'arrivée de fonctions transverses, garantir la cohérence des parcours utilisateur

Différentes applications

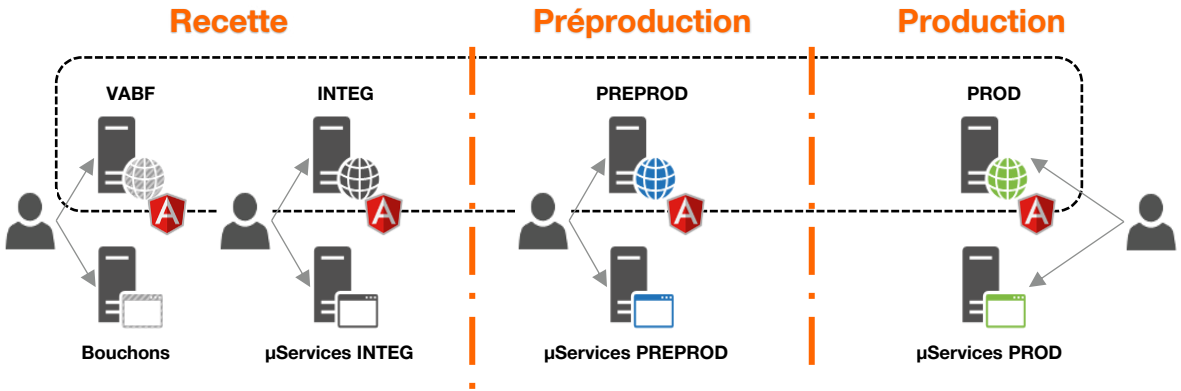
L'Espace Client se décline en plusieurs applications SPA

web
webview
conseiller
assistance

Constituées de

Features ecm-front & modules communs

Déployées sur plusieurs environnements



Les développeurs démarrent l'application en mode Standalone : une seule application Angular regroupe toutes les features..

L'écriture des tests automatiques concernant une évolution est inscrite dans son développement

Développement et tests en local

- *installation des dépendances (npm/yarn)*
- *compilation & construction (webpack)*
- *tests unitaires (karma)*
- *tests e2e (protractor)*

Build d'Intégration Continue

- *installation des dépendances (npm/yarn)*
- *compilation & construction (webpack)*
- *tests unitaires (karma)*

Build de tests e2e standalone

Exécution des scripts sur une instance locale de l'application (webpack-dev-server)

Publication des rapports (allure)

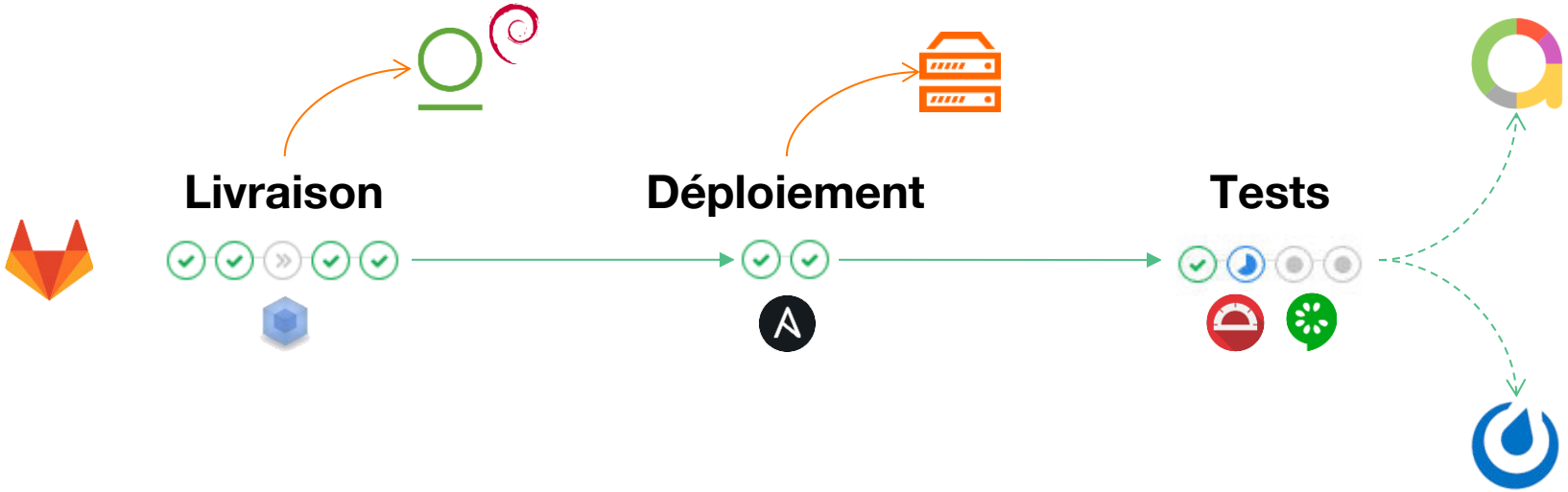
Tests e2e & Tests de sécurité sur l'application déployée

Agrégation des tests de toutes les features (protractor)

Scénarios dédiés à la sécurité (ZAProxy)

Cucumber (tests d'acceptance des US) ou Karma (tests unitaires).

Pipeline de Livraison



La livraison déclenche un déploiement en recette qui enchaine avec l'exécution des tests autos protractor/ cucumber

Les scénarios de test de toutes les feature sont agrégés pour tester chaque application déployée

Approche TDD et BDD avec Cucumber

En spécification, on sépare le travail entre l'analyste et le développeur :

- **l'analyste va écrire le test au format Gherkin, décrit dans les step.JS**
- **le code et le test du code sont versionnés en même temps sur la branche par le développeur.**

Exemple de scénario au format Gherkin basé sur un dictionnaire de phrases

```
Feature: ACT.Equipement.Changer-de-numero.UX
  En tant que client
  Je veux pouvoir changer le numéro de mon mobile

Scenario: 01 - Demande de dénumérotation en erreur avec code HTTP 500 sur le GET
  Given Je m'authentifie comme étant Ichigo
  When Je navigue vers la page changer votre numéro
  Then L'élément titre du message d'alerte doit contenir Désolés, une erreur technique s'est produite
  And L'élément message d'alerte doit être présent
  And L'élément message d'alerte doit contenir Le changement de numéro de mobile est momentanément indisponible.
  And Une requête Kenobi doit être envoyée avec les paramètres suivants:
    | codeRetour     | KDSI    |
    | fonctionnalite | ADENUM  |
    | rubrique       | M       |

Scenario: 02 - Demande de dénumérotation en erreur car non éligible sur le GET (code HTTP 429)
  Given Je m'authentifie comme étant Denver
  When Je navigue vers la page changer votre numéro
  Then La page changer votre numéro doit être présente
  And L'élément titre doit contenir Changer de numéro
  And L'élément titre du message d'avertissement doit contenir Désolés, le changement de numéro de mobile n'est
  And Le bouton Revenir à l'équipement doit être actif
  And Une requête Kenobi doit être envoyée avec les paramètres suivants:
    | codeRetour     | KDRG    |
    | fonctionnalite | ADENUM  |
    | rubrique       | M       |
  And Je clique sur le bouton Revenir à l'équipement
  Then L'URL doit contenir : /equipements/9400000001/31001
```

Les mots clés de la syntaxe GHERKIN

jbehave

cucumber

- **FEATURE** : fonctionnalité
- **BACKGROUND** : précondition pour une feature / x scenarios
- **SCENARIO**
- **GIVEN** : Etat du système (initialisation)
- **WHEN** : Action de l'utilisateur
- **THEN** : Comportement attendu
- **AND** : Compléter les états / actions
- **BUT** : En association avec le THEN (Sémantique)
- **SCENARIO OUTLINE** : Reproduire le scénario avec des données
- **EXAMPLES** : Charger des données sur 1 step précis

1

Présentation du contexte

Equipe socle et architectes, processus et demande métiers avec le couple Product Owner / Release Manager et avantages, applications et environnements déployés.

2

Pratique de développement de tests

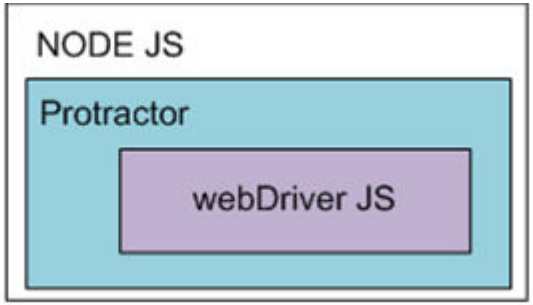
JIRA/confluence, création de la branche Tests e2e avec Protractor (framework de test angular) et Jasmine (tests techniques) ou Cucumber (tests d'acceptance des US) ou Karma (tests unitaires).

3

Diversité des solutions des Labo tests as a Service

Environnements de la Grid Selenium multi-navigateur avec émulation, container Docker, reporting au format Allure (outils en mode SAAS), enregistrement vidéo de la campagne de tests, gestion de ses (virtual Device) Android : récupération, création, suppression, démarrage, arrêt....

Deux modes d'intégration des tests AngularJS



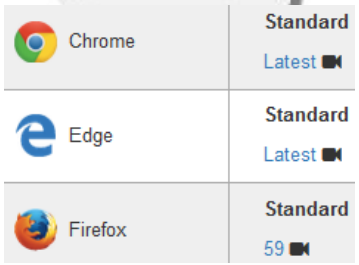
En local : autonomie
En distant : richesse de navigateurs

Serveur

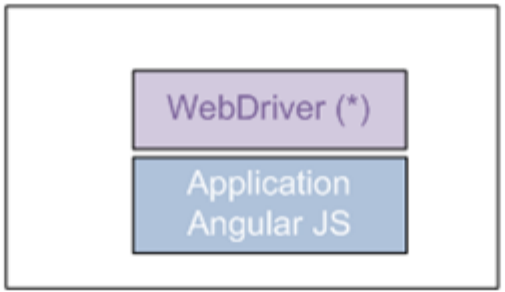
Server

Selenium Server (*)

(*) Exécution locale
(VM VDI)
ou à distance
(LTAAS)



Browser



(*) Selenium
webDriver
Chrome,
GeckoDriver,
SafariDriver

Présentation de la GRID Sélénium, runner Gitlab, image Docker Orange

labo test as a service

Accueil | Catalogue | Environnements de tests | Mes services | Support | Administration

Grid Selenium v3.141.59

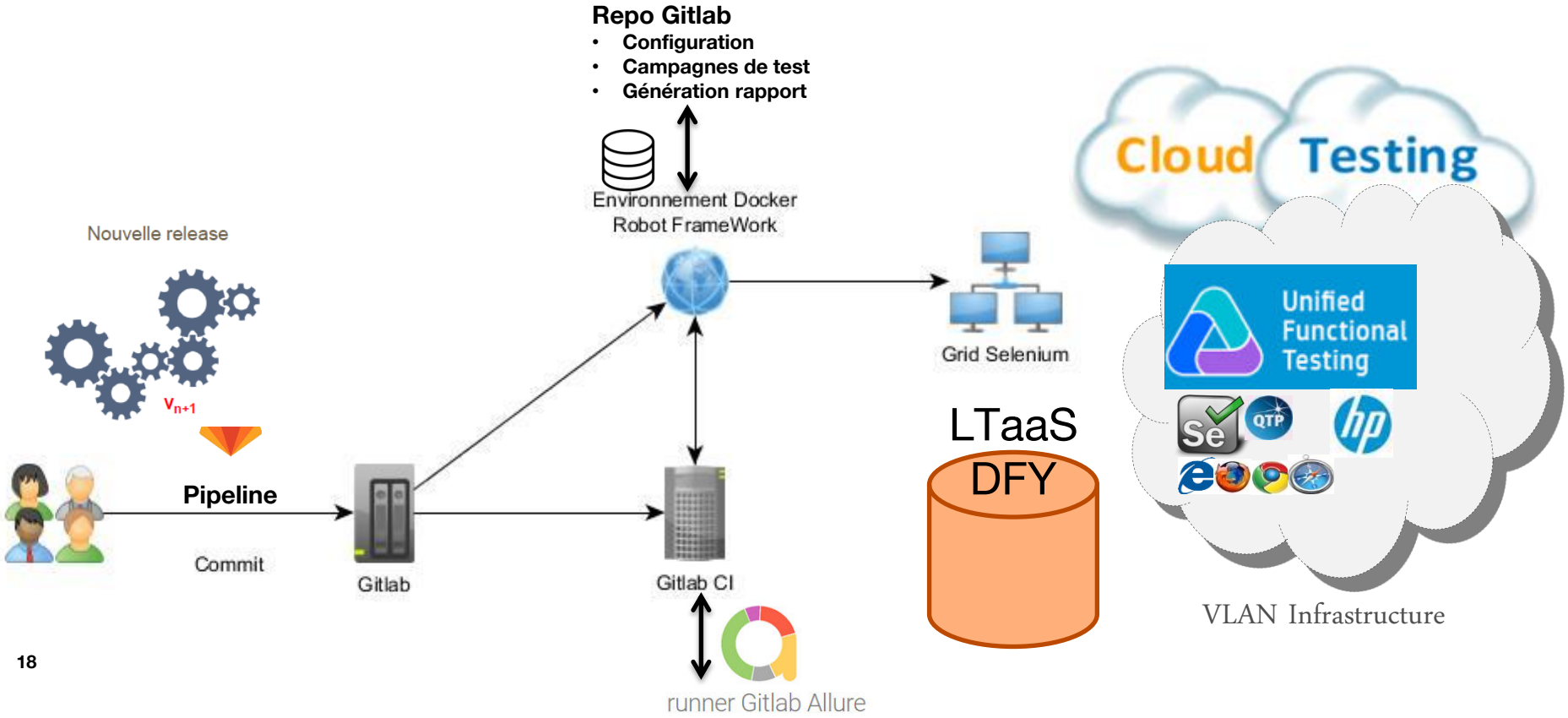
Android | iOS | Windows 7 | Windows 10 | macOS Mojave | Gitlab

Nom	Description
Allure	Runner Gitlab Allure Framework (+ d'infos)
Robot Framework	Image Docker Robot Framework (+ d'infos)

Industrialisation de la combinaison OS/ navigateur (hébergement de 3 à 4 nodes par machine sur 6 VM) pour un parc de 34 VM INFRA (services guacamole, RDP Manager, HP UFT, Allure, machines LINUX et Mac Mini..)

Intégration des tests sous Gitlab CI (workflow général)

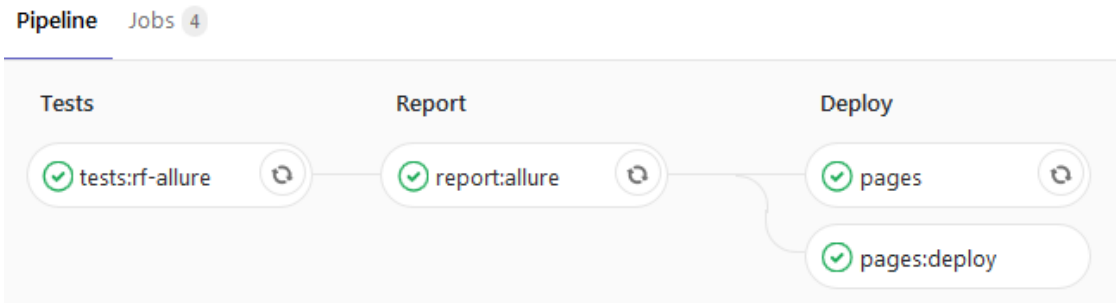
1. **Intégration des scripts de test en local + code dans un dépôt gitlab**
2. **Lors du Commit et du Push, exécution du test dans le Cloud du LtaaS**



L’intégration continue avec gitlab-CI sur le LTAAS

Les 5 étapes sont :

- création de l’environnement CI
- appel de l’image Docker
- l’état du pipeline passe « En cours »
- exécution des tests
- déploiement du pages



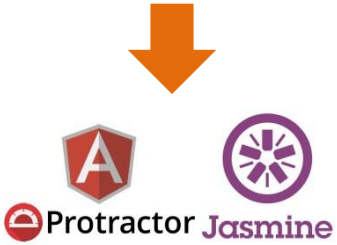
Intérêt de Gitlab CI :

- paralléliser l’exécution de tests sur différents nœuds
- prévoir, cadencer, exécuter des « *stages* »
- on peut voir si le pipeline est passé ou non (visuel)
- Le stage « Deploy » va permettre à gitlab de positionner la page internet sur le Pages du projet.

Grâce à Gitlab nous pouvons lier l’automatisation avec les « commits » des développeurs. Cela permet de détecter les régressions au plutôt

Passage de paramètres dans le code de l'automatisation

- **Un Framework par onglet**
- **Utilisation des paramètres du node Sélénium**



Chrome Latest

Nom du node Selenium : LTAAS-VM-Windows_7-Chrome_Latest
Version du node Selenium : 3.141.59
Etat du node Selenium : **Disponible**
Nombre d'instances : 4
Machine hôte : VM
OS hôte : Windows 7
Fonctionnalité vidéo (+ d'infos) : non

URL de la grid (`remote url`) : http://10.234.70.189:5551/wd/hub
Selenium (+ d'infos) :
- Version (`version`) : LTAAS-VM-Windows_7-Chrome_Latest
- Nom du navigateur (`browserName`) : chrome

Dans les exemples de configuration ci-dessous, le paramètre `${base_url}` est à remplacer et correspond à l'URL du site à tester.

Nightwatch | Robot Framework | Protractor

```
exports.config = {
    seleniumAddress: 'http://10.234.70.189:5551/wd/hub',
    multiCapabilities: [{
        browserName : 'chrome',
        version : 'LTAAS-VM-Windows_7-Chrome_Latest'
    }],
    specs: ['../scenarios/**/*.js'], // tests to execute
    framework: "jasmine2" // the framework used with custom options
};
```

Les environnements du LTAAS et visualisation d'instance d'exécution

Firefox – 9 Nodes

Chrome – 4 Nodes

Android – 16 Nodes

IOS – 32 Nodes

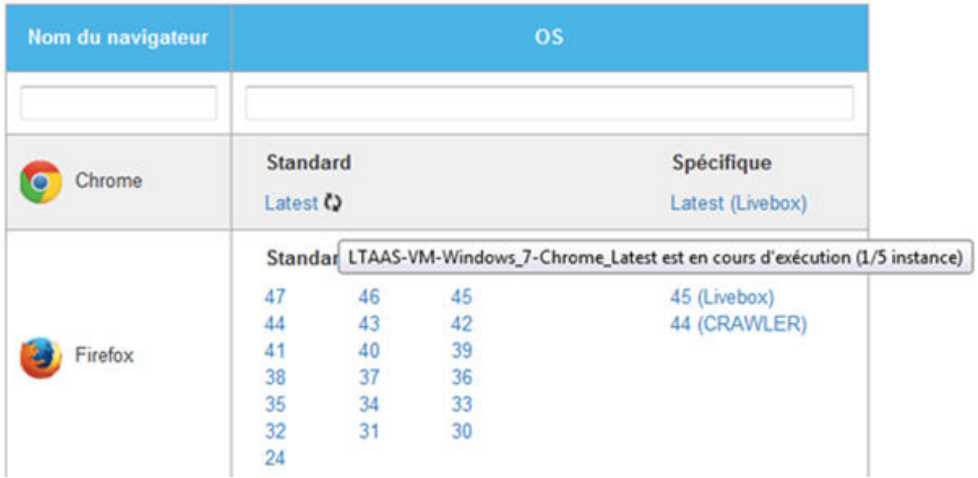
189 Machines virtuelles

8 Mobiles As A service

Internet Explorer – 4 Nodes

Safari – 1 Node

Karma – 1 Node Firefox



Karma : un serveur web exécutant le code source en fonction du code de test

Configuration et démarrage d'un serveur Karma sur l'environnement de développement de l'utilisateur

communiquant avec la grid Selenium du LTaaS

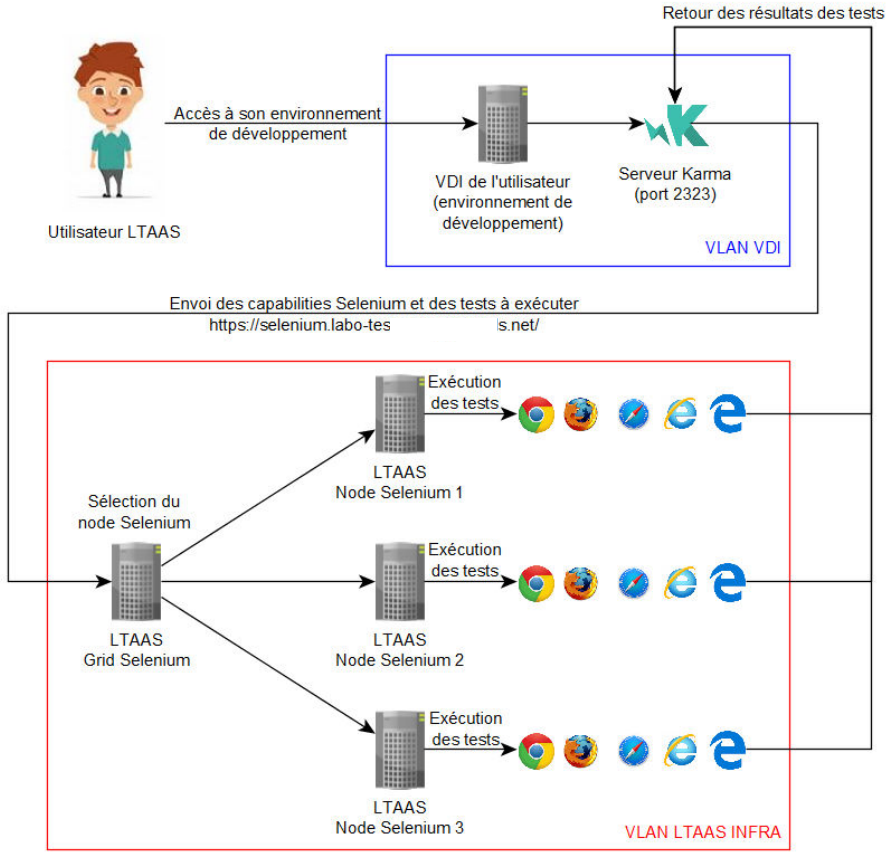
- exécution distante des tests
- spécification des *capabilities* pour sélectionner ses navigateurs dans la configuration Karma
- serveur Karma sur le port 2323

possible gestion en mode non *single run*

-Karma surveille les fichiers (code source + tests) et relance l'exécution des tests à chaque modification

l'utilisateur a un retour immédiat des résultats des tests (directement sur son environnement de développement)

- en mode non *single run* : retour à chaque modification de fichiers



Exemple d'utilisation de Karma

- Exécution distante sur un node Sélénium du LTaaS :
  - 1 instance occupée
  - 4 instances toujours libres

Firefox 62

Nom du node Selenium : LTAAS-VM-Windows_10-Firefox_62-Karma
Version du node Selenium : 3.141.59
Etat du node Selenium : **En cours d'exécution (1/5 instance)**
Nombre d'instances : 5
Machine hôte : VM
OS hôte : Windows 10
Fonctionnalité vidéo (+ d'infos) : non

- Le navigateur distant est connecté en permanence sur le serveur Karma de l'environnement de développement de l'utilisateur :

- L'utilisateur obtient ses résultats de tests : immédiatement :

```
Firefox 62.0.0 (Windows 10 0.0.0): Executed 55 of 55 SUCCESS (1.299 secs / 1.183 secs)

============================== Coverage summary ==============================
Statements   : 18.5% ( 128/692 )
Branches     : 5.7% ( 13/228 )
Functions    : 15.73% ( 28/178 )
Lines        : 18.5% ( 128/692 )
================================================================================
```

1

Présentation du contexte

Equipe socle et architectes, processus métier

2

Pratique de développement de tests

Intégration continue, Différents frameworks

3

Diversité des solutions des Labo tests as a Service

Environnements de la Grid Sélénium & container docker

4

Bilan et avantages

Chiffres-clefs

Avantages du Runner Gitlab Allure

- Gérer et stocker l'historique des campagnes de tests
- Obtenir facilement et rapidement un site Allure intégré directement sur le Pages du projet Gitlab

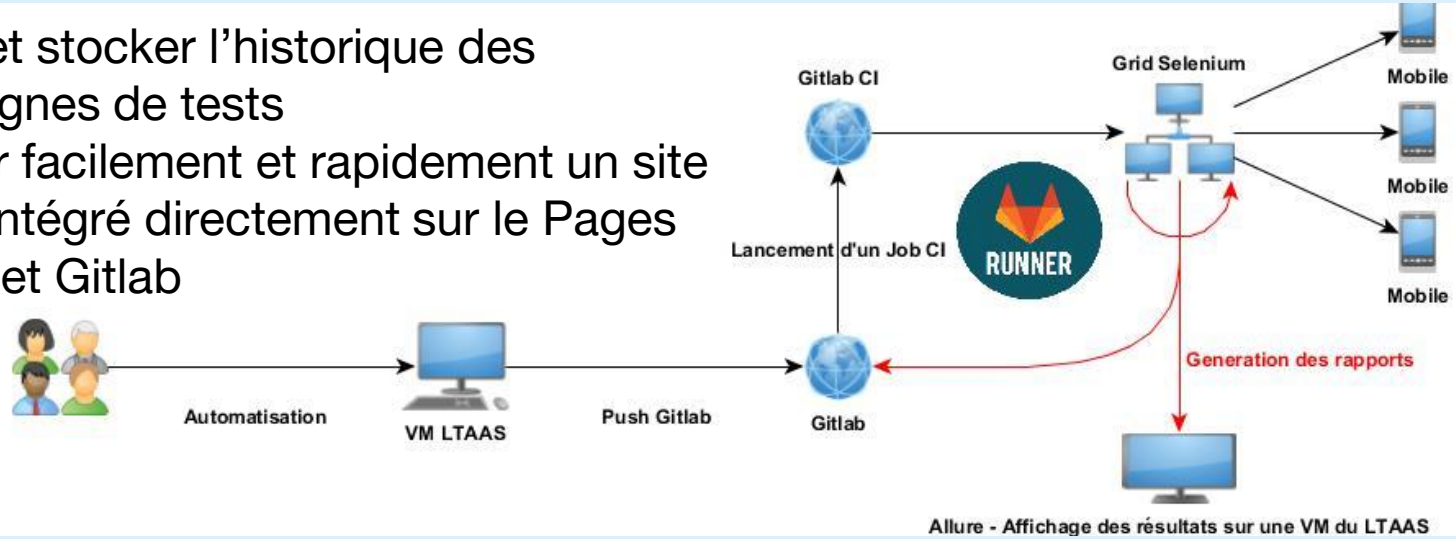
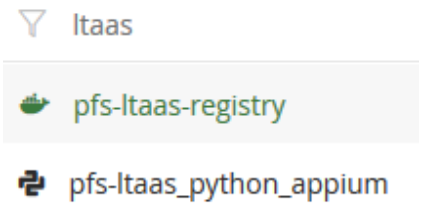


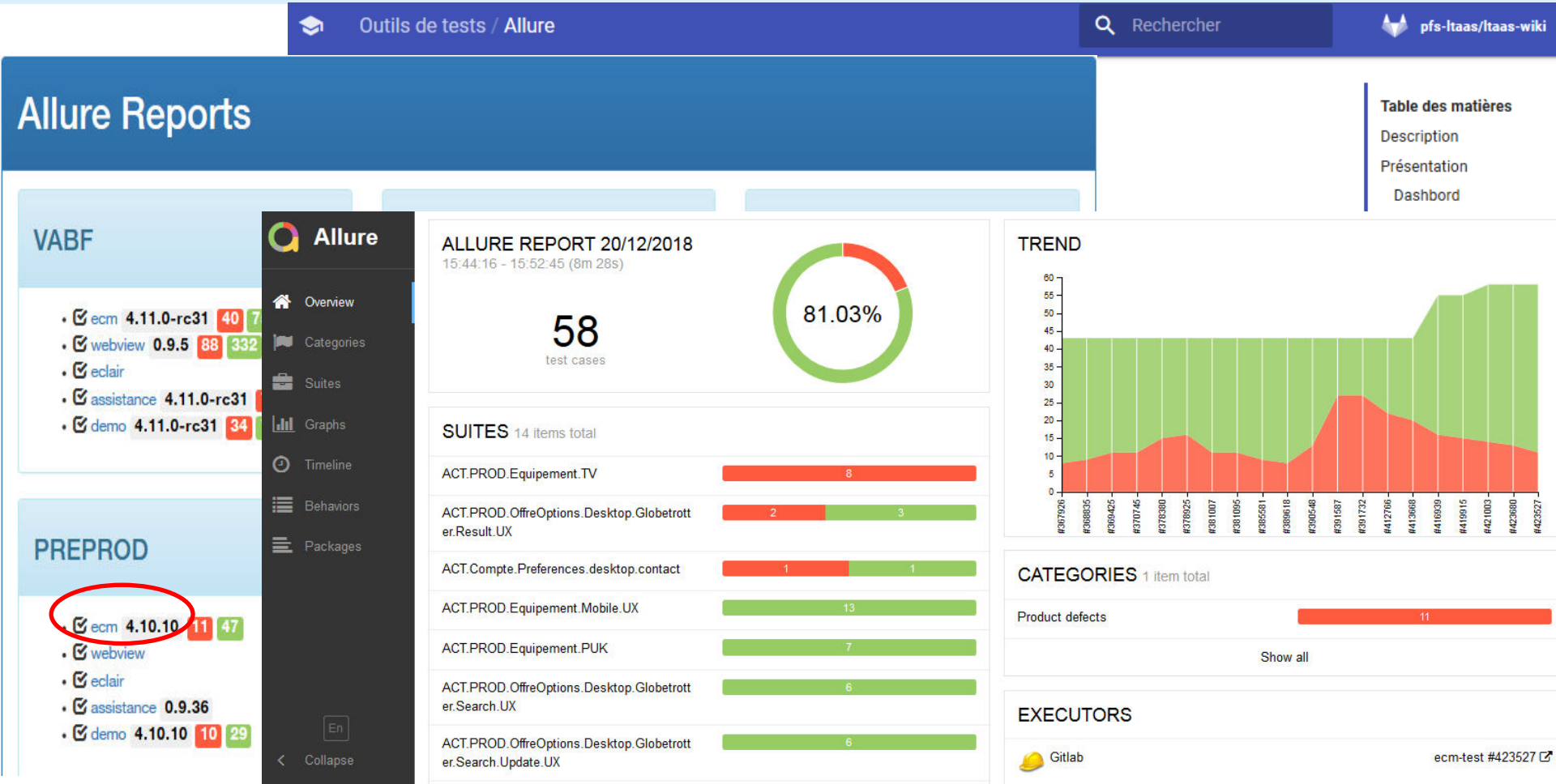
Image Docker sur l'Artifactory

JFrog Artifactory

- Outils préinstallés et préconfigurés pour tester avec Robot Framework
- Utilisable via Gitlab-CI
- Avec la fonctionnalité Vidéo
- Version 1.0.0 (avec `pip` en version 9.0.3)



Allure report : utilisation

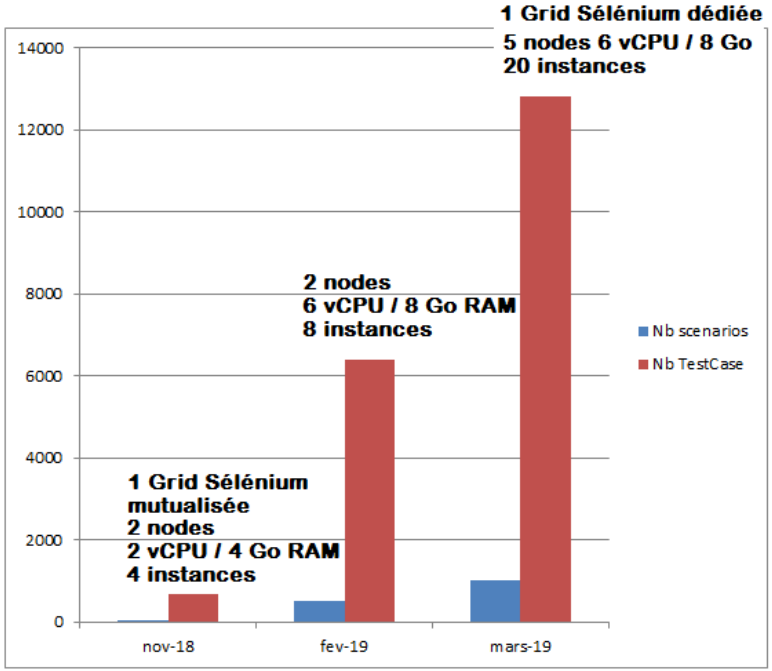


Résultats sur l'ECM (Espace Client Mutualisé)

1ère intégration

Décompte par les Runner

			Cucmber Desktop		Cucmber Mobile		Total		Time		
FT (Feature Team)	Protractor desktop	Protractor Mobile	Scenarios	Steps	Scenarios	Steps	Scenarios	Steps			Max instance
ECM-COMPTE	1	1	41	269	50	176	93	445			1
ECM-FAMILLE	0	0	17	153	1	3	18	156	197		1
ECM-FACTURE-PAIEMENT	9	9	43	398	35	269	96	667	481		1
ECM-EQUIPEMENT	9	0	149	1099	19	172	177	1271	871		2
ECM-OFFRE-OPTIONS	2	0	100	475	25	83	127	558	568		2
			0	0	0	0	146	0	331		1
			28	82	0	0	33	82	148		1
			64	287	4	16	91	303	706		1
			1	1	1	1	4	2	92		1
			0	0	0	0	137	0	196		1
						TOTAL FINAL	922	3484	3590	1 HEURE	

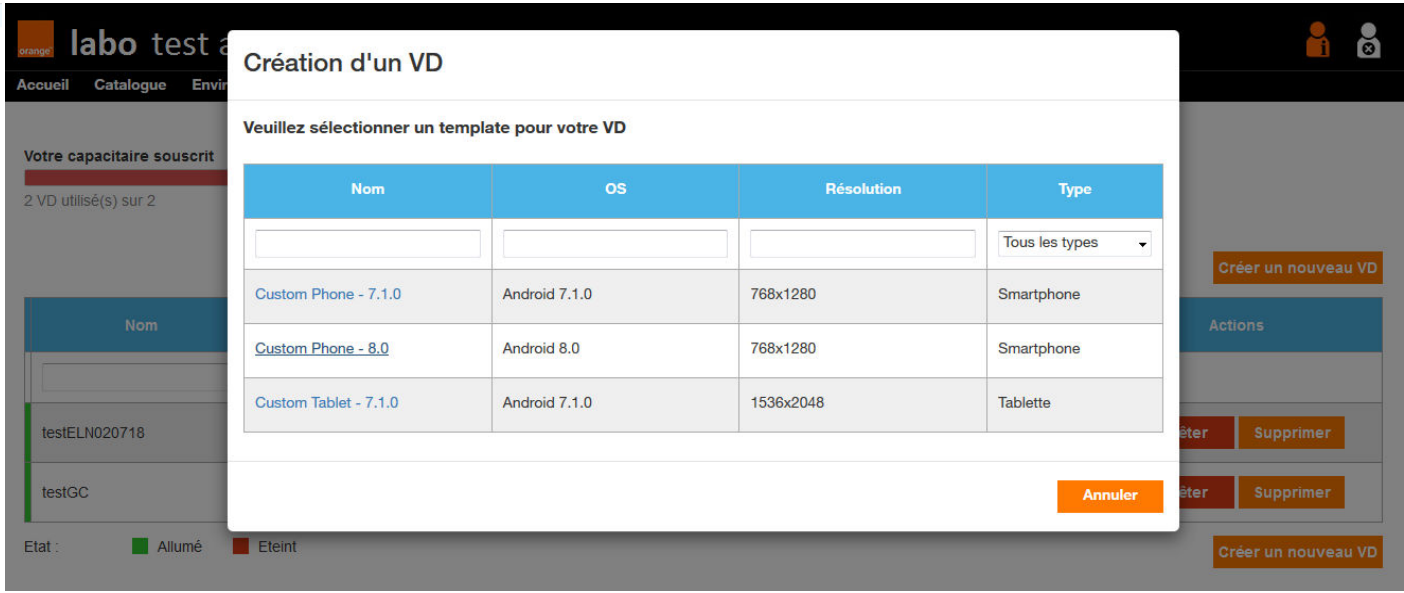


Il y a un job qui exécute à 12h et 21h le redéploiement complet de toutes les features Teams + tous les tests

La Solution Mobile du LTAAS pour Android

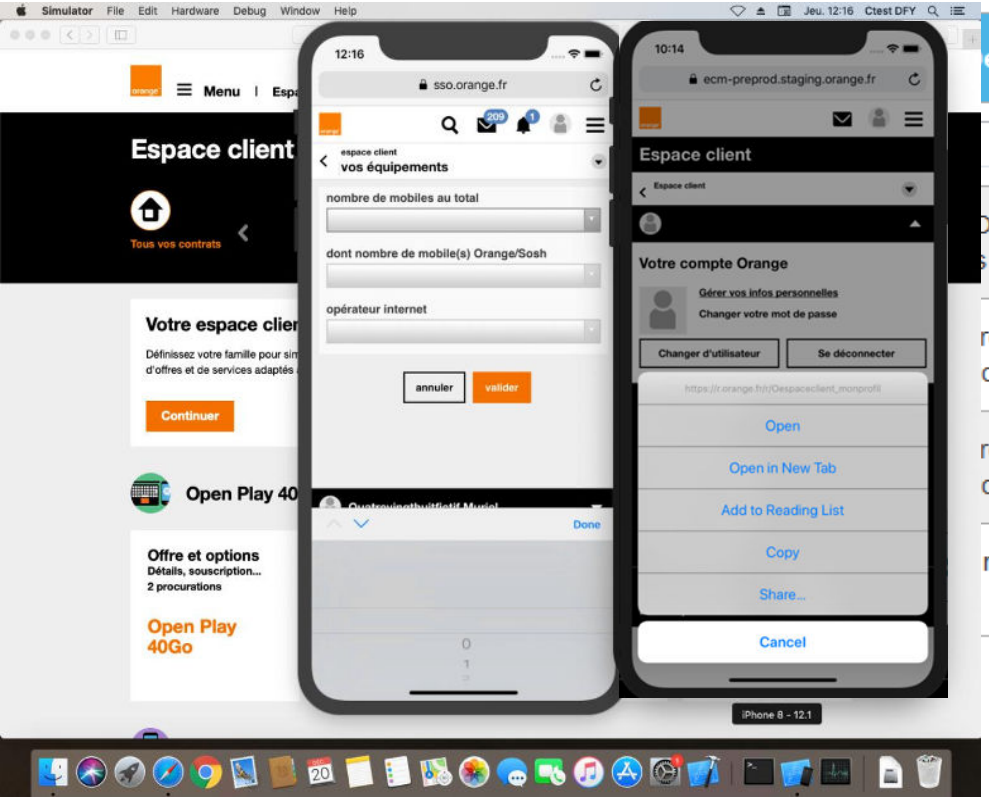
Le LTAAS propose la gestion de VD (Virtual Device)
Le middle-office gère les VD : récupération, création, suppression, démarrage, arrêt...
Une fois que le VD est créé, l'administrateur peut le démarrer, le stopper, puis le supprimer…


```
status:          "success"
vds:
  0:
    uuid:        "6a51d822-3688-4435-bb9f-c56473c82e52"
    name:        "testGC"
    ip:          "192.168.56.101"
    state:       "On"
    type:        "phone"
    screenWidth: "768"
    screenHeight: "1280"
    templateUuid: "c1cf6d47-cc64-45d3-abd9-61156d7de08b"
    os:          "Android"
    osVersion:   "7.1.1"
    accountId:   "11x4"
    machineId:   "40x1796"
  1:
    uuid:        "f13823fd-5f17-4ab6-ba56-906b0521ef20"
    name:        "testELN020718"
    ip:          "192.168.56.103"
    state:       "On"
    type:        "tablet"
    screenWidth: "1536"
    screenHeight: "2048"
    templateUuid: "bdf63cc3-ab08-418c-adfc-1d885aaab6f3"
    os:          "Android"
    osVersion:   "7.1.1"
    accountId:   "11x4"
    machineId:   "40x1796"
```

Accès aux machines de la salle serveur via le front-office

- Le LTAAS permet l'accès en lecture seule pour les utilisateurs
- Et en contrôle pour les administrateurs (admin base)

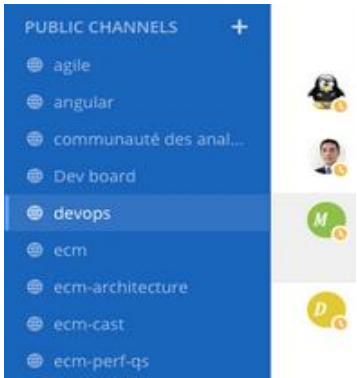


escription	Outils	Actions
OpenSTF) : mobiles physiques s à la grid Selenium	STF	Accéder
roid dédiés rattachés à la grid on)		Accéder
roid dédiés rattachés à la grid on)		Accéder
mutualisés rattachés à la grid	iOS	Accéder

Bilan, apports et difficultés

1. Investissement à long terme INFRA (coûts rentables sur x>8 à 10 projets)
2. Réglage des timeouts Cucumber nécessaires. Durée globale moyenne d'une campagne ~ 9min pour 12 816 TC ~1000 scenarios (5 nodes, 16 instances, 1 Grid Sélénium dédiée ECM..)
3. Aujourd'hui, on répond avec un système qui fonctionne mieux que ce qui est actuellement en place, on avance à petit pas et on adaptera en permanence pour que le système global soit le plus efficace possible

« En avançant, on apprend ». Réutilisation de scenarios avec Cucumber alors qu'avec Karma/Jasmine où il faut prévoir de la factorisation de code



- Test : Vision commune / Langage commun
- Documentation à jour versionnée avec le code

4. Les notifications Mattermost sont à privilégier aux échanges par mail pour permettre le partage avec les membres de l'équipe

L’apport d’une méthodologie est un plus..

Préparation

- Choix méthodologie (MBT, BDD, ATDD)
- Organiser ses pools (VM, terminaux..)
- Choix d’un automate ET un Scheduler
- Formation des ressources
- Ouverture de flux
- Couverture de tests
- Type de test : fonctionnel, stress, robustesse, KPI

Spécification

- Suite de tests
- Fonctions – création actions réutilisables
- Keywords
- Passer les paramètres et penser « généricité »
- Utiliser les JDD dans les listes ou les fichiers

Exécution

- Exécution en local
- Vérifier accès au dépôt
- Identifier version et plate-forme
- Vérifier exécution depuis la Grid Mutualisée
- Fermer le navigateur après chaque test

..appliquée lors de chaque Sprint

Rapport test Scheduler le développement

Clôture

Merci

https://github.com/allure-framework

http://www.protractortest.org/#/ (Jasmine, Mocha, Cucumber)

https://karma-runner.github.io

Source interne Orange : « GHERKIN & BDD approach », Cédric Hervé, membre de la Communauté DevOps SI France d'Orange

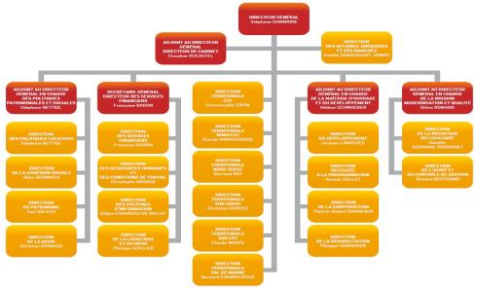
Contact :

Emmanuel LEON

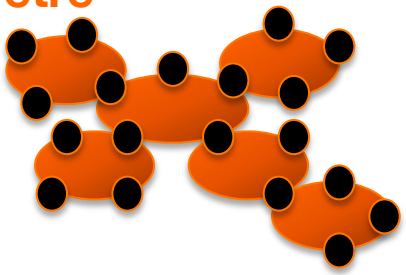
Emmanuel.leon@orange.com

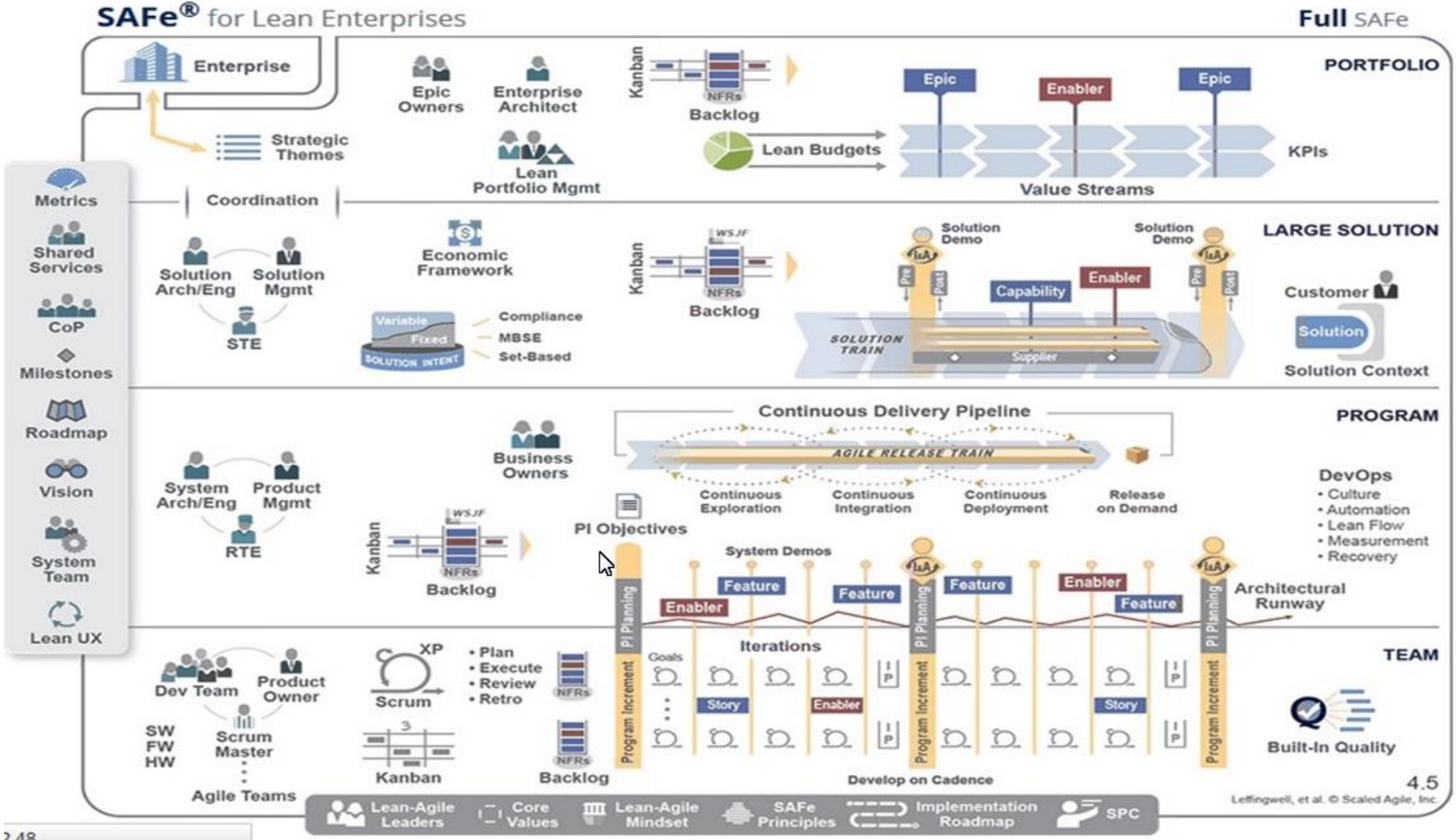
SAFE impose une gouvernance adaptative : être plus efficace, plus dynamique et prendre des décisions rapidement

1. Une organisation basée sur le consensus autour d'une **raison d'être**
2. **Organiser ses livraisons** avec un system demo par itération
3. Une **distinction claire entre les rôles et les personnes**
4. Les compétences **DEVOPS** sont une marge de manœuvre essentielle pour faire progresser la System Team
5. **Les tensions soulevées** lors des réunions ne sont pas celles des personnes, mais celles du système
6. Une approche résolument **pragmatique,** avec une logique **"Test and learn" pour favoriser l'agilité opérationnelle.**
7. «**Ressentir et ajuster**», et «ne pas prévoir et planifier»



Raison d'être




SAFe® for Lean Enterprises
Full SAFe
Enterprise
Strategic Themes
Epic Owners
Enterprise Architect
Lean Portfolio Mgmt
Kanban
NFRs
Backlog
Lean Budgets
Epic
Enabler
Epic
KPIs
Value Streams
PORTFOLIO
Metrics
Shared Services
CoP
Milestones
Roadmap
Vision
System Team
Lean UX
Coordination
Solution Arch/Eng
Solution Mgmt
STE
Economic Framework
Variable
Fixed
SOLUTION INTENT
Compliance
MBSE
Set-Based
WSJF
Kanban
NFRs
Backlog
Solution Demo
Pre
Post
Capability
Enabler
SOLUTION TRAIN
Supplier
LARGE SOLUTION
Customer
Solution
Solution Context
System Arch/Eng
Product Mgmt
RTE
Business Owners
Continuous Delivery Pipeline
AGILE RELEASE TRAIN
Continuous Exploration
Continuous Integration
Continuous Deployment
Release on Demand
PI Objectives
System Demos
Feature
Enabler
Architectural Runway
PI Planning
Iterations
Goals
Story
Program Increment
IP
PROGRAM
DevOps
• Culture
• Automation
• Lean Flow
• Measurement
• Recovery
Dev Team
Product Owner
SW
FW
HW
Scrum Master
Agile Teams
XP
Scrum
Kanban
• Plan
• Execute
• Review
• Retro
Develop on Cadence
TEAM
Built-In Quality
4.5
Leffingwell, et al. © Scaled Agile, Inc.
Lean-Agile Leaders
Core Values
Lean-Agile Mindset
SAFe Principles
Implementation Roadmap
SPC

Annexe (exemple de copie écran Instance Chrome du LTAAS Orange)