



# GREENSPECTOR

Exigences non-fonctionnelles par  
les budgets

JFIE 2017 | 21/11/2017

Olivier PHILIPPOT @SimplyGreenIT



- ▶ **7 years** of existence
- ▶ **15** employees
- ▶ Location **Nantes**



## ▶ **Software** Editor

Measure, scan,  
benchmark !

**Optimize** your  
mobile, web, IoT  
**applications** !

- ▶ **100%** software  
**eco-design**  
core competency



# Exigences non-fonctionnelles de qualité (NFR)

---



Performance



Utilisabilité



Fiabilité



Maintenabilité

Il s'agit d'une **exigence non-associée à une fonctionnalité demandée par le client mais cependant liée à un besoin client** sous laquelle un logiciel doit fonctionner ou être développé.

# Les limites des NFR : processus

---

## Exigences non-fonctionnelles vs fonctionnelles :

- **Souvent dépriorisées par les équipes**
- **Vérifiées en fin de cycle (en recette)** car souvent au niveau système
- Non adaptées aux nouvelles pratiques de développement (Agilité par exemple)
- Exigences incomplètes compte-tenu des systèmes complexes

### **Exemple: Affichage d'une page web en moins de 1 seconde**

- Dans quelles conditions (Wifi, 2G...) ?
- La vérification de l'affichage nécessite d'avoir le site web complet
- Quid de la fragmentation des plateformes (Android, Navigateur, desktop...)

**Elles sont généralement traitées en production.**



# Les limites des NFR : outils

---

## Outils de profiling

- + Mesures complètes des ressources mais non axées exigences
- Complexes, peu d'automatisation

## Outils de testing (Performance)

- + Offre existante sur les outils de performance
- Axés souvent uniquement backend

## Outils de monitoring

- + De nombreuses métriques
- Trop d'informations pour vérifier des exigences

# NFR de performance de + en + importantes



**53% de visiteurs** abandonnent le site si ce dernier ne se charge pas en **3 secondes**

*Source : Chrome Dev Summit 2017*

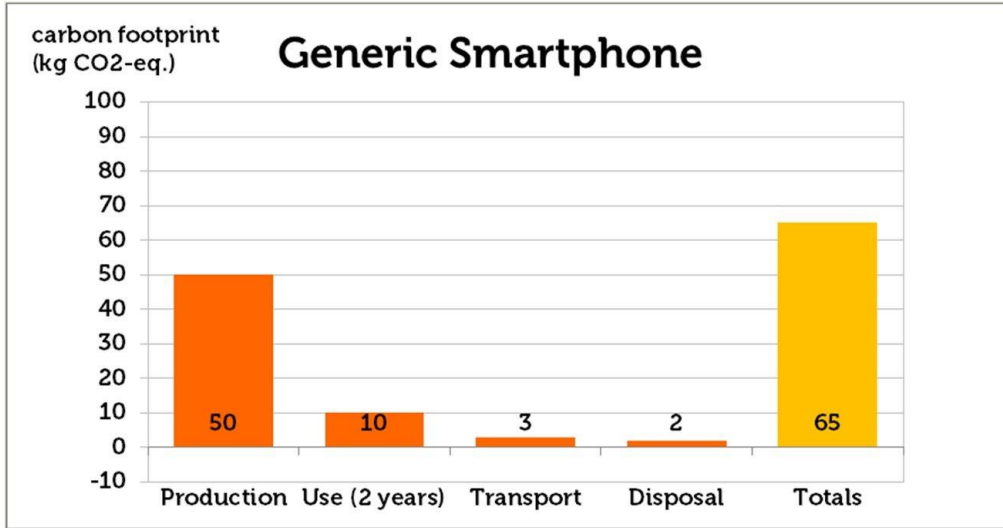
# Exigences renforcées avec la mobilité



**19 secondes** de chargement moyen en **3G**  
pour les sites mobile

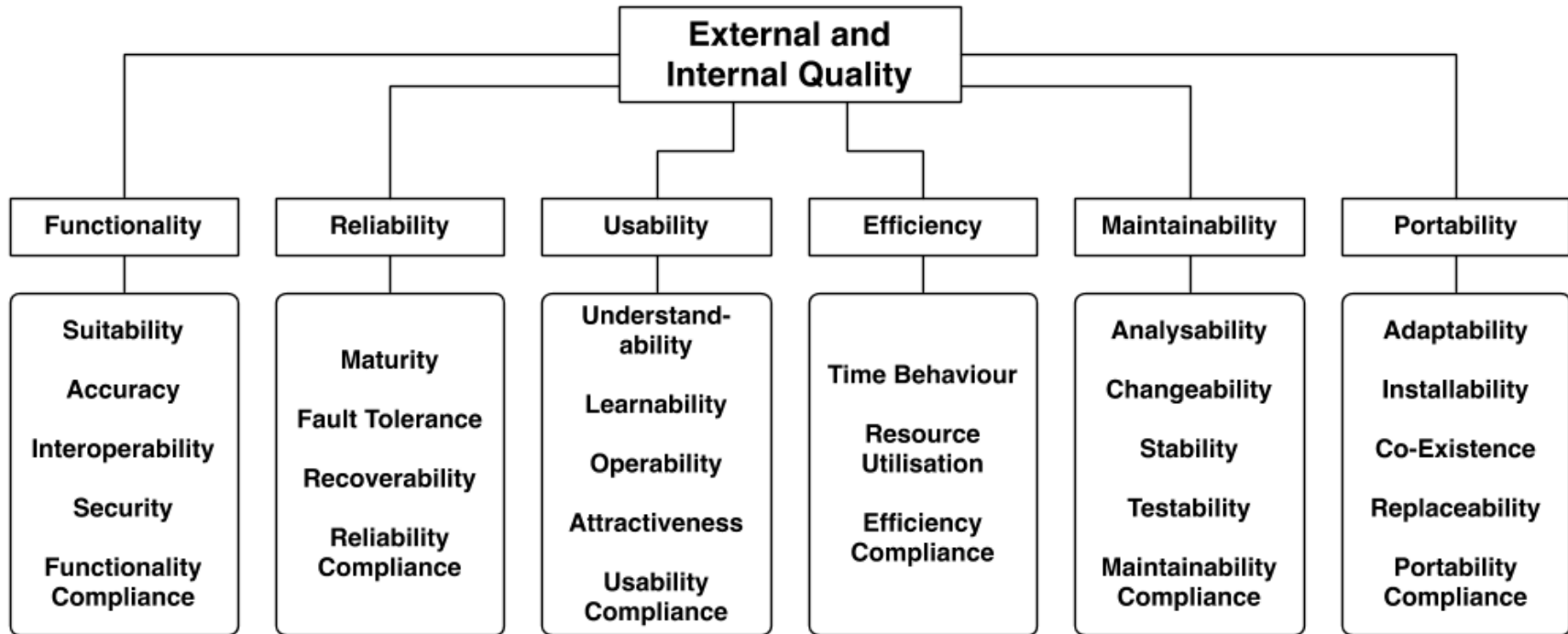
*Source : Chrome Dev Summit 2017*

# Passer de la performance à l'efficacité (voir plus)





# Les NFR selon ISO 9126



# Définition des NFR axée efficience

---

## ISO9126 sur l'efficience :

*"A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions."*

Il faut donc mettre en relation :

- Performance
- Usage ressource
- Conditions sur l'environnement

La définition d'une **NFR** doit donc lister au moins un élément de chaque domaine :

**Le système doit respecter [cette performance] en consommant [au maximum ce niveau de ressource] dans [cet environnement / condition d'exécution]**

# Notion de contrainte et de budget

---

La notion de consommation de ressources est une **contrainte à prendre en compte pour respecter une performance donnée.**

Les **conditions d'exécution** sont aussi des contraintes (La 2G implique par exemple une consommation limitée de bande passante)

La contrainte devient un **cadre pour les exigences non-fonctionnelles**

La notion de contrainte est adaptée dans le contexte de plateforme hétérogène (Smartphones toute génération et toute puissance par exemple)

Ces contraintes forment un **cadre qui peut être considéré comme un budget**

**La performance doit être respectée sans dépassement du budget ressource**

La notion de budget permet d'avoir un détail plus fin via une répartition budgétaire (Répartition fonctionnelle ou technique)

Limite : Nous n'avons pas de retour d'expérience sur l'application aux autres NFR. Cependant, il serait possible sûrement de l'appliquer à d'autres domaines.



Un certain besoin utilisateur non fonctionnel doit être respecté avec une (ou des) contrainte(s) de consommation matérielle.

Besoins ou  
exigences

Affichage en moins de 2  
secondes

8 heures d'autonomie

Durée de vie de la batterie  
de plus de 3 ans



Connexion 2G

Smartphone 1 coeur

Batterie 2000 mAh

Contraintes

Sans contraintes, pas d'efficiency mais uniquement de la performance !

L'adéquation entre le besoin et la contrainte se formalise avec un budget « Ressources » réparti sur les fonctionnalités.

**Besoin**  
8 heures d'autonomie

**Contrainte**  
Batterie 3000 mAh

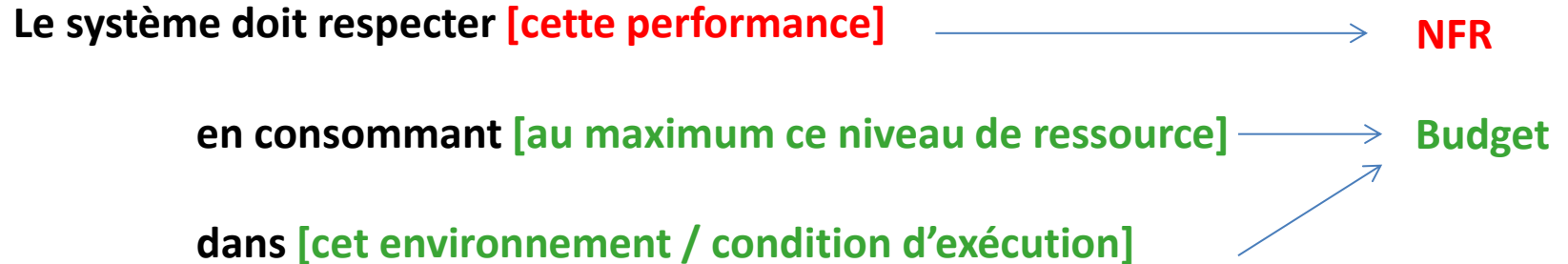
Un 1<sup>er</sup> budget nécessite  
des données estimées  
ou des 1<sup>ères</sup> mesures

6 mAh par scan

2 heures de veille  
**Comportement**  
non-fonctionnel

200 scans par jour  
**Usage**

# Le budget dans la définition NFR ISO9126



# Budget dans la performance web

## Modèle RAIL (Google)

Response : 100ms

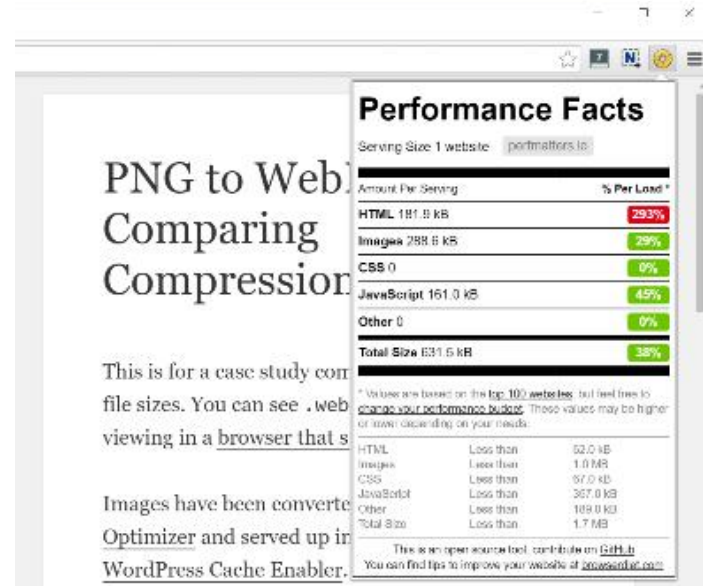
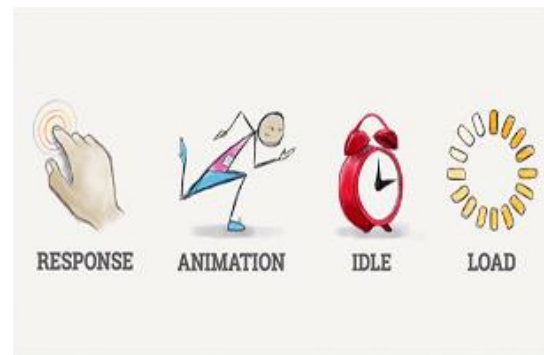
Animation : 16 ms

Idle : 100 ms

Load : 1s

+ : Répartition sur les éléments de la page

- : Absence de contrainte et de consommation ressource



# Avantages des budgets

---

Permet d'étendre la définition des NFR et de les répartir sur des **éléments plus granulaires**

**Par exemple :**

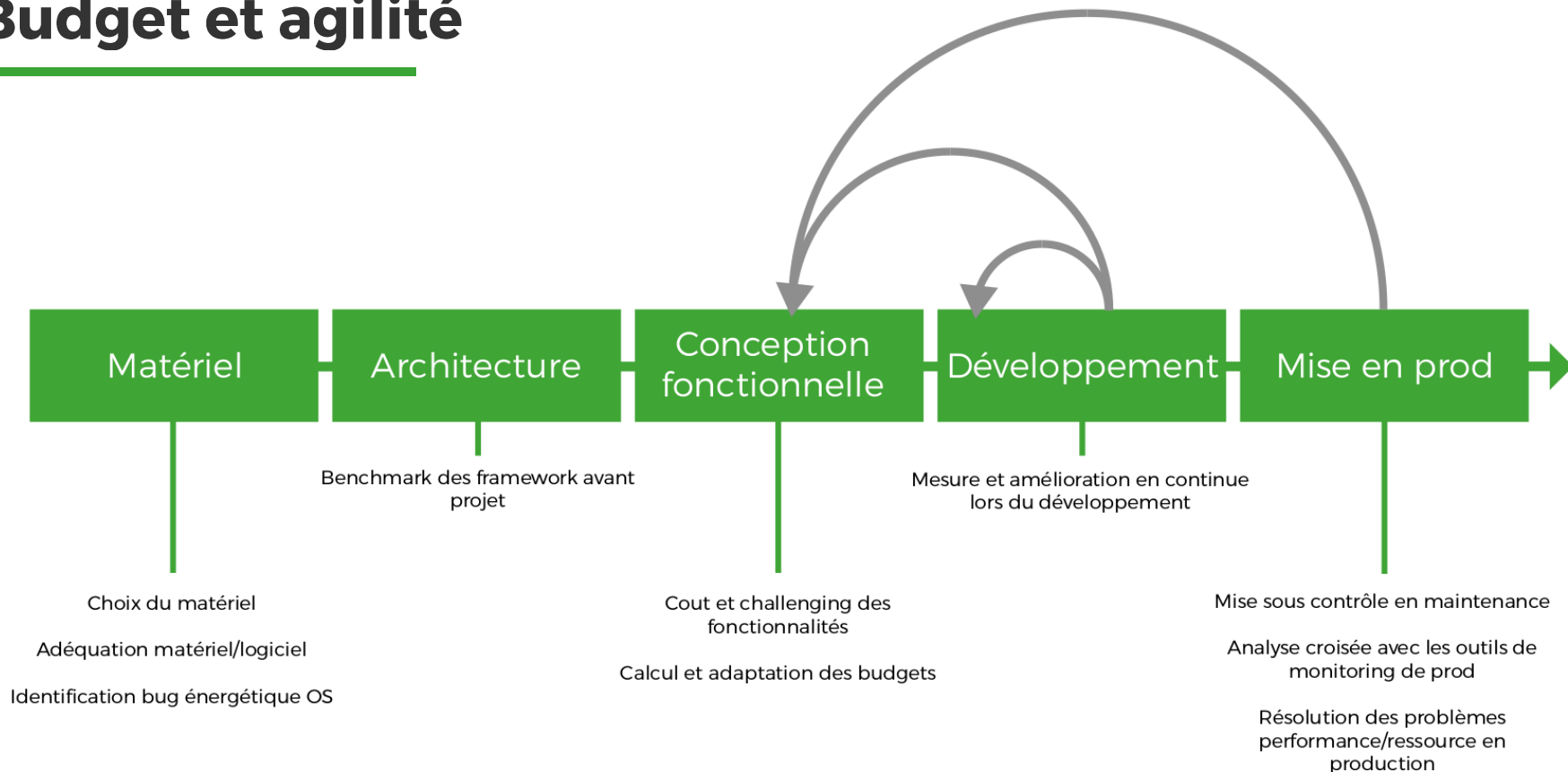
- Temps de chargement des éléments : JS, images, .... dans une page web
- Permet de surveiller le respect en **agilité**
- Se base sur le même principe que le nombre limite de tâche dans le **Kanban** (WIP)
- Engagement entre toutes les parties (pas uniquement les développeurs)

**Plusieurs actions possibles suite aux dépassements**

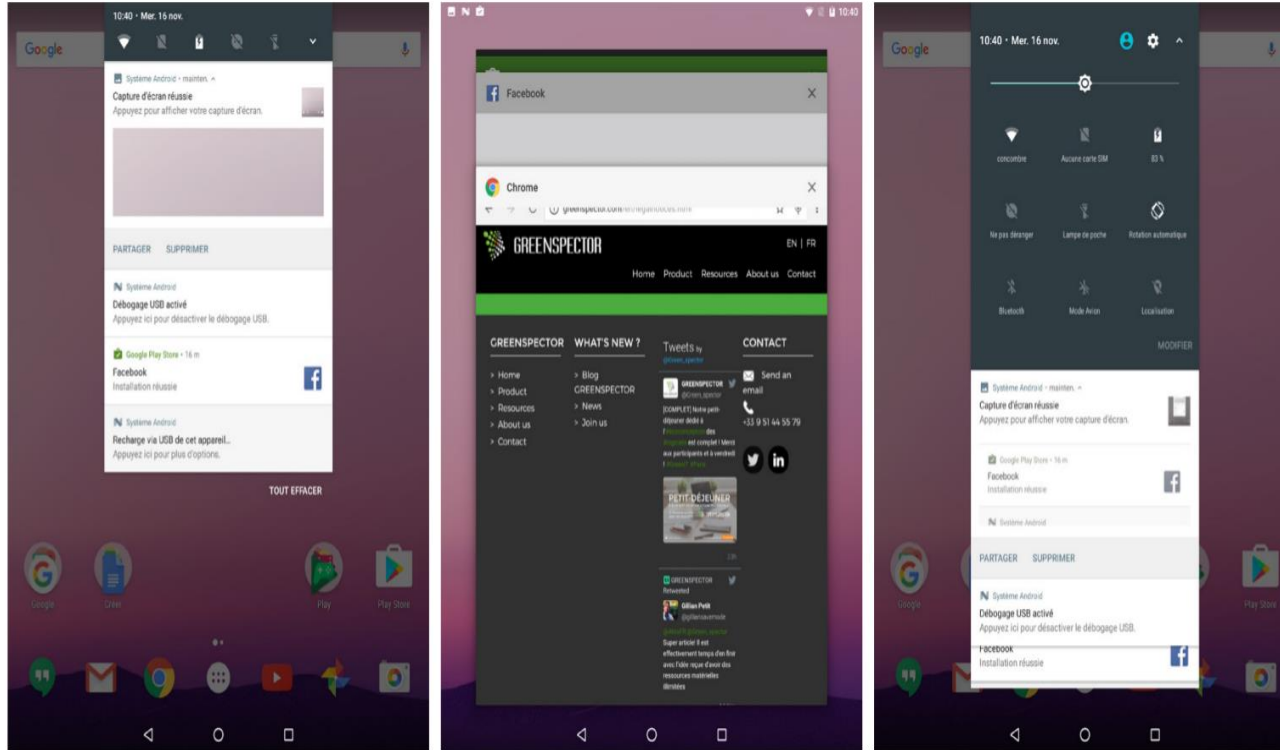
- Discussion en fin de sprint
- Optimisation
- Modification des besoins fonctionnels
- Transfert des budgets sur d'autres fonctions



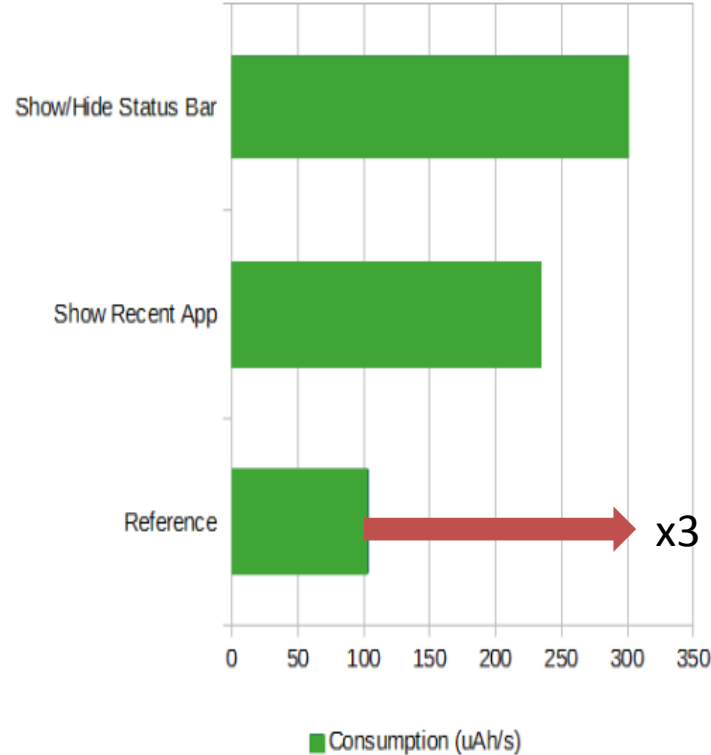
# Budget et agilité



# Retour d'expérience : Android System UI

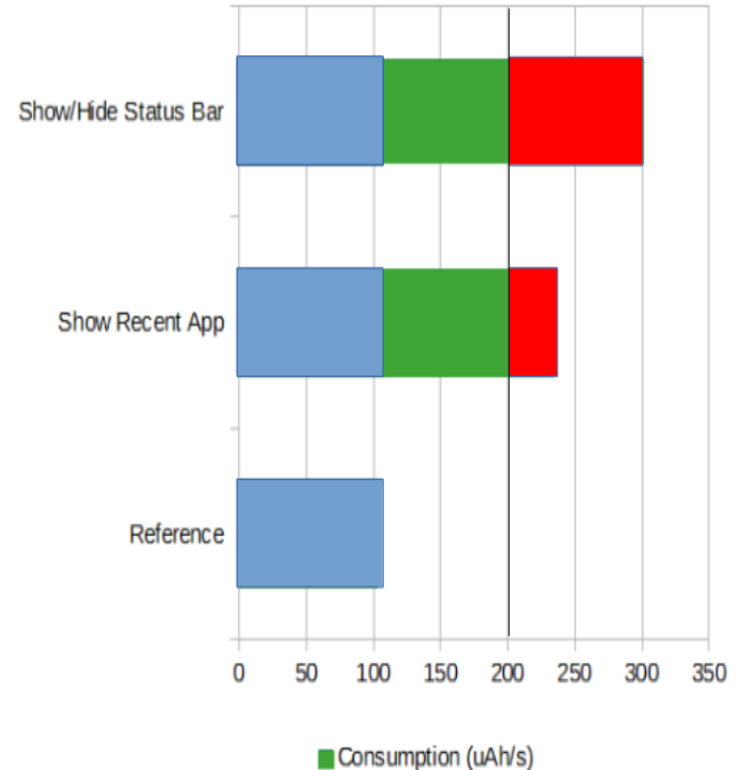
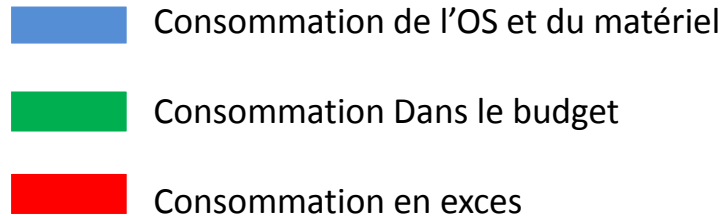


# Retour d'expérience : Android System UI



# Retour d'expérience : Android System UI

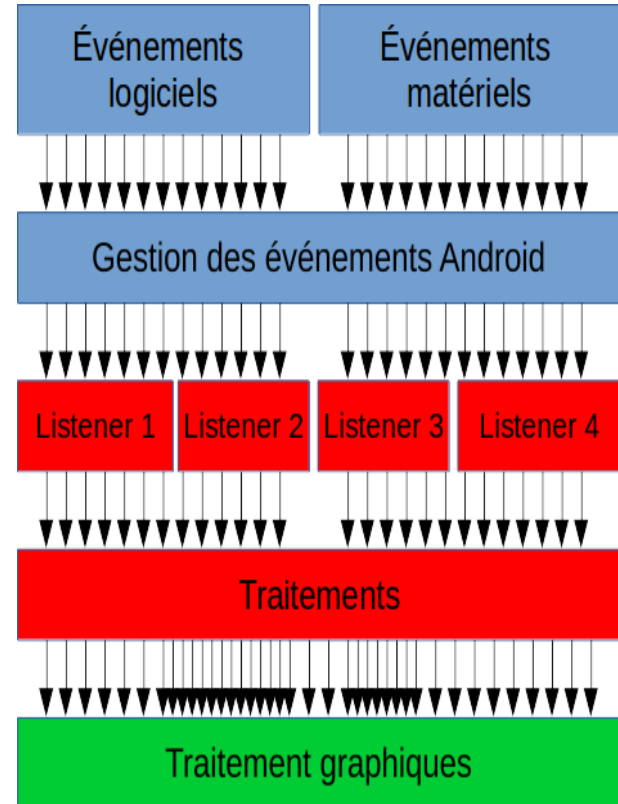
**Budget de 200  $\mu\text{Ah/s}$**   
**Dépassement important**  
**Performance OK sur matériel performant**  
**Optimisation nécessaire**



# Retour d'expérience : Android System UI

Le dépassement de budget permet de :

- Détecter un **problème de consommation ressource malgré une performance correcte**
- Lever des **problèmes de performance** sur certains devices
- Limiter le temps de debug et de profiling
- Remettre en cause la gestion des interruptions de Android



# Retour d'expérience : Android System UI

2 jours de correction pour une amélioration notable

Exemple de correction limitant l'impact des events

```
public void setIsAirplaneMode(IconState icon) {  
    // Greenspector-UpdateReduce: We have to check if the airplane mode has changed  
    if (mInfo.airplaneModeEnabled == icon.visible) return;  
    mInfo.airplaneModeEnabled = icon.visible;  
    refreshState(mInfo);  
}
```

Pour des gains globaux non négligeables

ShowHideStatusBar_Energy				
Platform CPU	16.71 %	15.51 %	-1.20 %	-7.2 %
Platform Discharge	59.36 mAh	55.55 mAh	-3.81 mAh	-6.41 %
Platform Discharge per second	301.96 µAh/s	273.88 µAh/s	-28.08 µAh/s	-9.3 %



# **Pourquoi utiliser l'énergie comme NFR de performance ?**

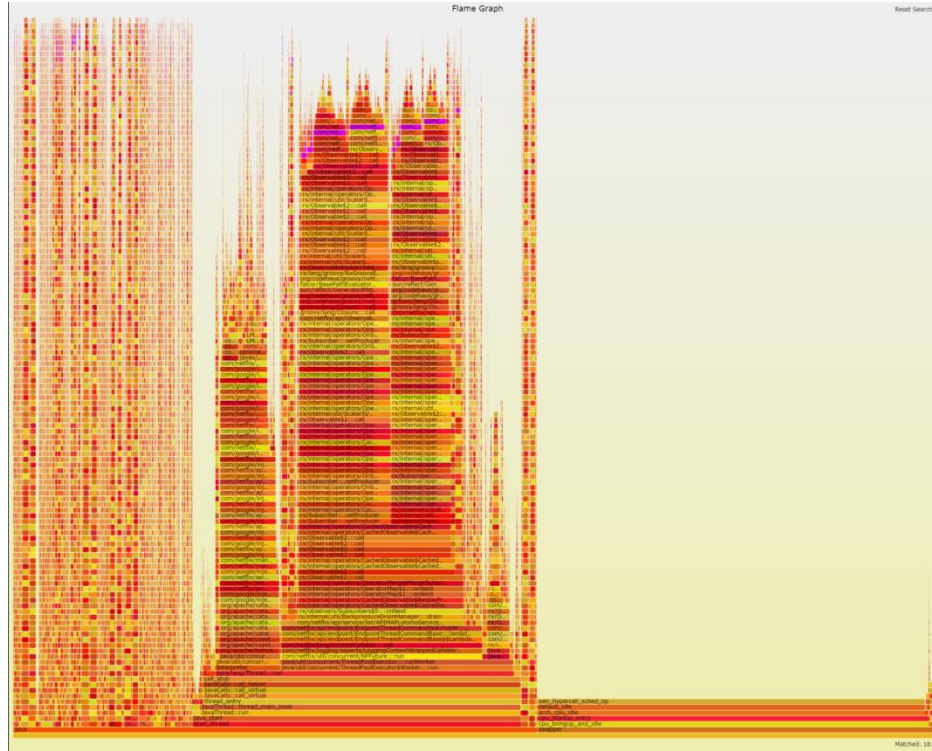


# Pourquoi mesurer ?

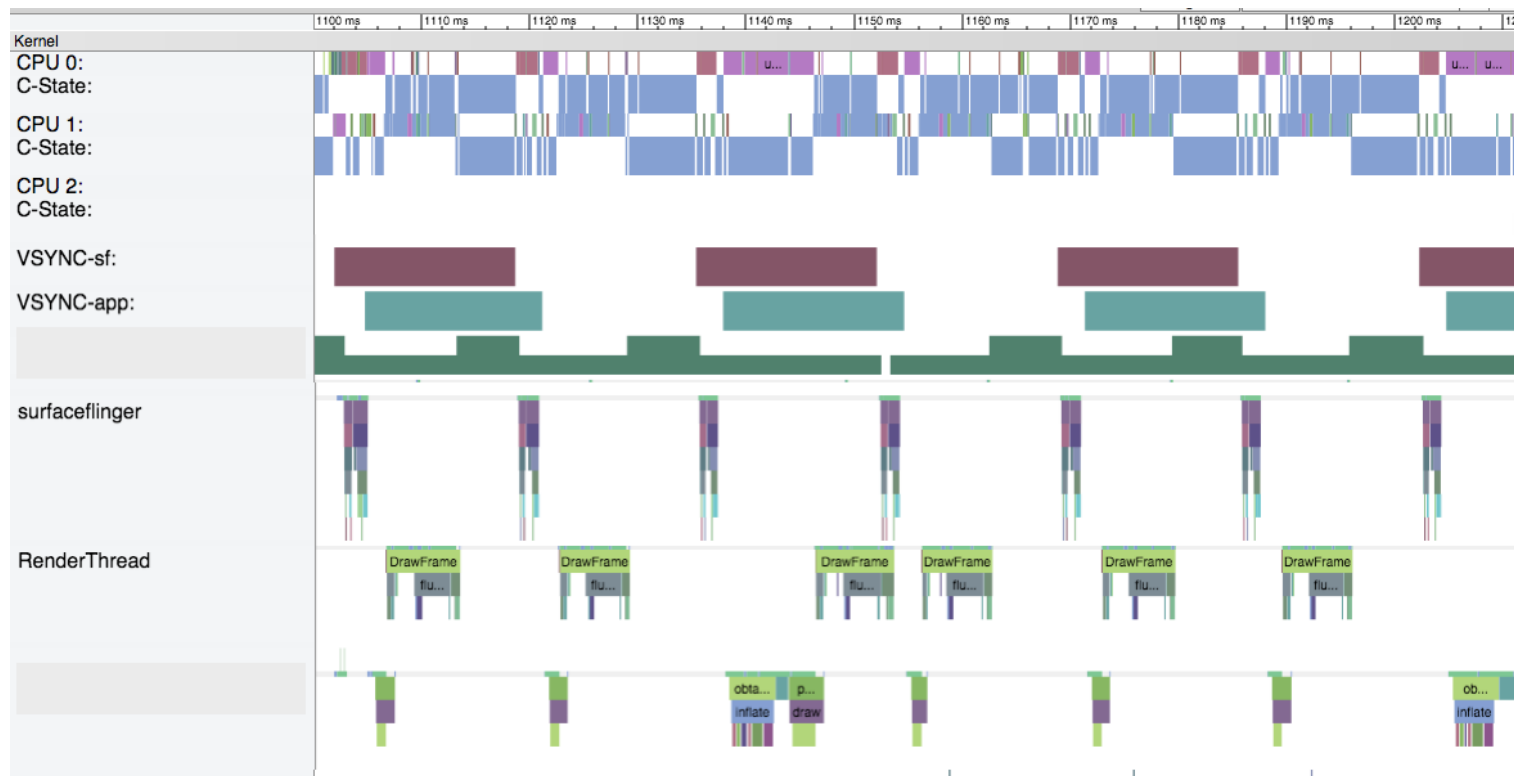




# Difficulté d'analyse du niveau de performance



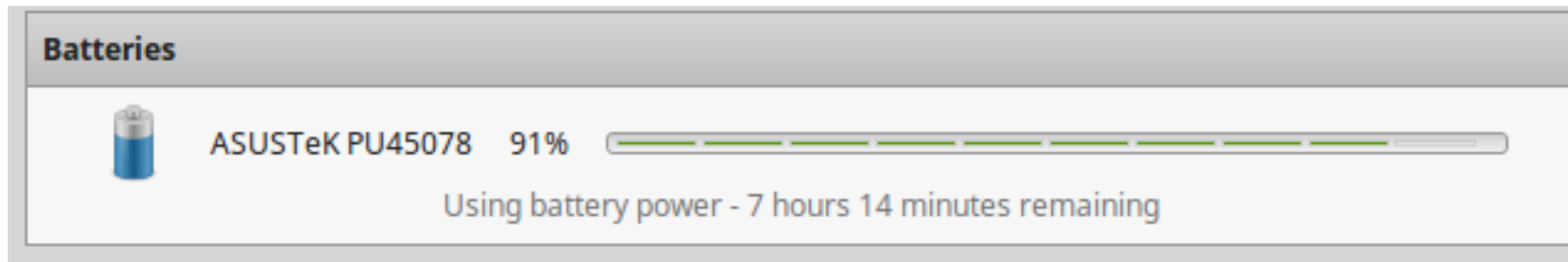
# Difficulté d'analyse du niveau de performance



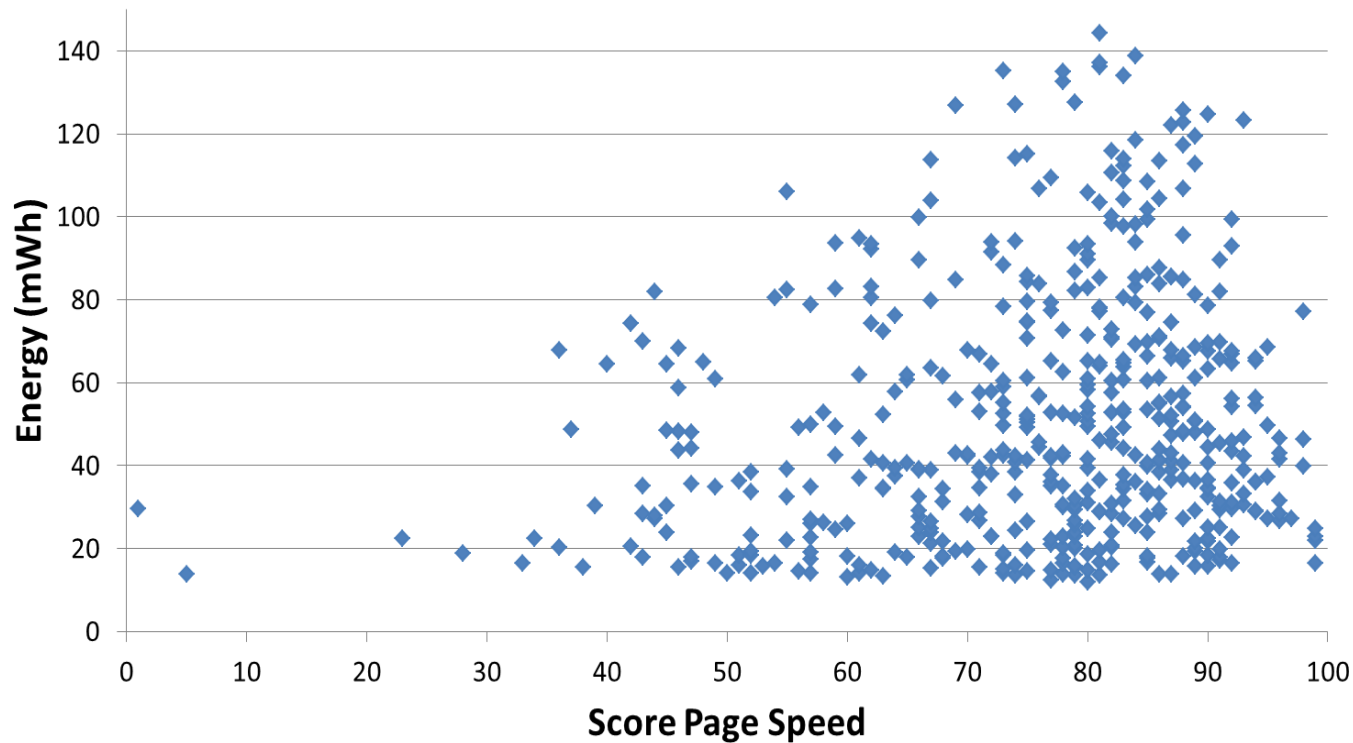
# Energie comme métrique NFR

---

Au-delà du besoin de réduire la consommation d'énergie, la métrique **énergie est un indicateur majeur pour piloter** la consommation de toutes les parties du système (CPU, mémoire, réseau...)



# Limite de la performance pure



# La mesure en continu

---

**L'intégration du concept dans les plateforme d'intégration continue (PIC) permet de contrôler en continu les budgets.**

Tout outil qui permettra de mesurer les **métriques classiques** et d'être **automatisable** est compatible :

- Web Page Tests
- Outils de chrome (Lighthouse)
- Instrumentation de performance classique (CPU, mémoire...)

**Pour ajouter l'énergie :**

- Power API (modélisation de l'énergie sur PC)
- API Android
- Greenspector
- Instrumentation dans les data centers



# Lien avec les tests fonctionnels

---

**Les outils de mesure doivent être utilisés en parallèle de tests automatisés.**

Nécessité d'avoir un périmètre maîtrisé (Unité fonctionnelle, unité de temps, contraintes connues)

Nécessité d'être répétable

Les tests fonctionnels sont idéals (un budget associé à une fonctionnalité)

Attention, les outils de **Real User Monitoring** (Dynatrace, Appdynamics...) ou **de profiling ne permettent pas forcément cette approche**

Notion de budget et de contrainte complexe (manque unité de temps entre autre)

# Conclusion

---

**Les NFR atteignent leurs limites** telles qu'appliquées avec les méthodes actuelles.

La notion de **budget** et de contraintes associées permet d'**intégrer tous les acteurs** et de mieux formaliser et tester les NFR

De plus, la notion de budget va permettre l'**intégration dans les process agiles**

La **performance** pure (temps de réponse) est une métrique importante mais doit être **complétée par l'énergie**.



# GREENSPECTOR

Empower your applications

GREENSPECTOR® est une marque déposée.  
SAS au capital de 85 440 € - RCS Nantes 528 071 939  
6 rue Rose Dieng-Kuntz – 44300 Nantes - France

**Nous contacter :**

**GREENSPECTOR**

[contact@greenspector.com](mailto:contact@greenspector.com)

Tél. : 09 51 44 55 79

6 Rue Rose Dieng-Kuntz

CS 90729 – 44307 Nantes

Cedex 3

[www.greenspector.com](http://www.greenspector.com)



@Green\_spector