

## Session Audit

Session Audit Session proposée par Joel Rotelli - dev freelance depuis 2006 qui intervient sur de la maintenance et reprise de sites existants

Audit et techniques de gestion de TMA.  
Quelles sont les différentes façons ?

1/ Regarder la liste des modules installés

2/ code source : utilisation module hacked pour vérifier si le core et / ou modules ont été modifiés - Voir module site audit qui intègre Hacked + des tas de métriques - paramétrages de cache, variables (pbm de ce module : pas possibilité de customiser les critères de bien/pas bien sans modifier le code du module.

<https://drupal.org/project/hacked>

[https://drupal.org/project/site\\_audit](https://drupal.org/project/site_audit)

2bis : vérifier architecture (emplacement themes, emplacement drupal custom au meme endroit que module )

Saisi a un script perso pour ça : nb de themes activés, nb templates/themes, nb de taxo, etc..

-> dispo sur le github de Makina Corpus (drupal site audite)

[https://github.com/makinacorpus/drupal\\_audit](https://github.com/makinacorpus/drupal_audit)

3/Module architecture : (ressemble bcp au script de simon). Type de contenus, taxo, permissions, etc...

<https://drupal.org/project/architecture>

4/ Module entity graphe : plugin pour drushs, gérer les dépendances entre entités (peut etre pas les node ref ? )

5/ Module github osinet : QA : controles : graphe des dépendances, vérifie les vues en overrides, code php dans le vues, (Note : drupal 8 a supprimé php filter), jeu de traduction pas complet, termes inutilisé dans les free tagging

6/ note : si une image n'est plus utilisée, drupal supprime automatiquement. Si c'est image importée manuellement, fais ton script - bonne chance

media : permet d'importer en masse des fichiers pour qu'ils soient gérer en drupal - media 2 +views : tout supprimer en batch

7/ Grep tout ce qui contient du php dans le theme : accès aux variables globales dans les themes, accès DB dans les themes

8/ note : moins c'est commenté, plus l'audit sera cher :p

9/ construction de requete par assemblage de chaine - requete DB hors api - module coder (attention aux faux positifs) - php sniffer (prérequis pour module coder)

10 / Bibli PhpCPD : repère les copy paste pour le duplicate de code (exemple : 4 modes de connexion à facebook )

11/ Module Schema : il examine le schéma de la BDD, module invoke all hook schema et

compare avec la base : ce qui a changé, ce qui est manquant et ce qu'il y a en trop -> SUPER CELUI LÀ !

12/ Checker les doubles fichiers .info : Grep info\_files pour trouver les modules présents plusieurs fois dans le site (dans module/custom ET module/contrib)  
- registry rebuild

13/ Version du core et des modules et MAJ nécessaires - attention à la dépendance des versions de modules

En général, il vaut mieux mettre à jour le core d'abord ? autre approche, les petits d'abord sinon pour être vraiment secure, mise à jours successives en passant par toutes les versions intermédiaires

Pour le testing : faire tourner la suite de tests à chaque maj - selenium, BEHAT (voir session BEHAT)

14/ Comportement du cache avec heisen-cache : analyse le comportement du cache (à faire tourner sur la preprod idéalement sauf problème en prod), choisir ce qu'on veut observer comme comportement. Collecte événement, requête, activation, désactivation - marche avec tous les caches courants.

Coupables de perf de cache usuels : la taille du cache de traduction augmente. Si on atteint la max taille de memcache, bam, la chaîne qui dépasse le seuil fait memcache ne sert plus à rien.

-> Ca, ça se passe qd tu mets un nom de variable (de contenu : valeur de champs ou valeur de titre) dans la fonction t.

-> cache update NE DOIT PAS être dans memcache : LE LAISSER EN BASE (attention, après installation de memcache, il faut configurer pour que le cache update reste dans la base).

ATTENTION :

Note : pas d'UI.

15/ ne jamais avoir update et l10n update enabled en prod -> introduit un single point of failure - site ne doit pas dépendre d'un autre serveur pdt le cycle de page (or, l'update consulte drupal.org pour vérifier les maj de modules)

16/ Toujours activer cache et agréger css / js en prod.

hook\_(css|js)\_alter : sortir un js qui fait planter l'aggrégation, pour que le reste soit agrégé mais pas le js qui plante)

grep pour détecter les extensions de fichiers qui n'ont pas leur place dans l'install drupal (.zip, tar, tgz etc ...)

Checker les .info qui contiennent des ref vers des fichiers qui n'ont rien à faire -  
Audit d'un module : ouvrir à la main le .info