Auditing a Salesforce Org

Functional Audit

One workshop mid-audit to discuss their process
One workshop after the audit to discuss findings, step forwards

General stuff to check

Administrator presence - Number of admins, experience, etc. (https://help.salesforce.com/HTViewSolution?id=000007548)
Organization Security - Check the automated health check in Setup > Health Check
User-Friendliness - custom app ? limited number of tabs ? how many fields per page layout ? is LEX-enabled ? Page layouts make somewhat sense ?
Maintainability - using best practices, nomenclatures, not using old notes or old attachments, low-ish number of automations per object, good structure in automations if many, avoid multiple sources of automation on one object.
Usage - have users logged in in the past 30 days ?

Limits

Data limit - Setup > Data Storage
Storage Limit - Setup > File Storage
Object limits - Run Optimizer if possible or check each object limit.
APEX limits - API calls per 24h, errors when tracking a user, etc.

Security

OWD & role reviews
Review of Profiles & Permission sets, flag any VAD or MAD access
Review Permission Set Groups & Muting Permission Sets
Review of any external access

Review Sharing Rules

Data Model

Review of limits and object usage
Review of field usage
Review of general architecture
Solution design of Refactoring if needed

Data

Duplicates ?
Reportable ?
Owners make sense ?
Old records ?
Quality ?

Automation

Number of Processes/Workflows per object
Process Builder best practice audit
Flow best practice audit

- How many flows are there ?
> few: not used, flag it as admin probably doesn't know about them, check how the flow looks internally to check
> medium: can be anything, but OK
> A lot: are they well named ? If yes yay, that's generally an admin that knows their shit. If not I'm in for a world of hurt straight out the bat.

- how big are the flows ?
> small : don't care
> medium: don't care
> big: see if there's squares in the same order a few times, that means they should be using subflows. you'll notice it
> huge: should be using subflows anyway

- variables
> no input variables : they're using it as a script, train admin, but at least it's easy shit

> only input variables : they have no idea wtf input vars do, train them and check the flow for other bullcrap
> normal usage: yay

- elements
> are the elements named consistently ?
>> Yes: yay
>> No: fuck

> are all elements present used ?
>> yes: YAY:
>> no: fuck

> are all elements that could be reused reused, or are there a ton of throwaway elements ?
>> tons of shit that gets used once because they don't know how to use assignments and vars: fuck
>> reusable stuff: yay

- structure
> are the DMLs outside of the flow main structure ?
>> yes: YAY
>> no: fuck

> Are there elements in loops other than assignments or decisions ?
>> yes: fuck
>> no: YAY

- maintainability
> can the flows be read ?
>> yes: phew
>> no: fuck

> do you know wtf they're here for in less than 30 minuets ?
>>yes: somebody did their job right
>>no: fuck

> are the structures of the flow consistent and make it easy t maintain (subjective)?
>> yay
>> fuck

> are there flows that should be APEX or other shit ?
>> yes: list them, explain why they shouldn't be that way
>> no: cool beans

Validation Rules best practice review
Workflows review

APEX

High-level APEX review: naming, basic structure
Code audit: standard best practices, optimization
Architecture audit
Solution design of refactoring if needed

Generation of the Audit report

Revision #7
Created 2019-08-19 09:21:56 UTC by Windyo
Updated 2023-03-07 00:48:17 UTC by thejamesjames