



Un guide de la modélisation Anaplan



v2021.01_FR

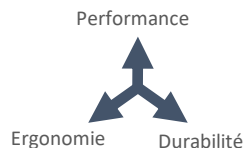
Août 2021

Introduction



Bienvenue dans le *Planual*, LE guide des meilleures pratiques pour la modélisation Anaplan. Il est le fruit de la consultation de nos salariés, clients, partenaires et Maîtres Anaplanners afin de recueillir, évaluer et établir les techniques et architectures permettant d'obtenir les meilleurs modèles possibles sur la plateforme Anaplan.

En élaborant ces normes, nous avons pris en compte trois objectifs principaux :



Les normes (règles) contenues dans le *Planual* recherchent le point d'équilibre entre ces trois aspects afin d'assurer le meilleur résultat à l'ensemble des utilisateurs d'Anaplan - les concepteurs de modèles comme les utilisateurs finaux. Nous avons essayé de rendre les règles aussi générales que possible. Le *Planual* ne concerne pas la syntaxe de formule en soi, bien qu'il contienne certaines directives pour structurer vos formules.

Il est important de noter que ces règles sont indicatives, et il y aura inévitablement des situations qui justifieront de les adapter. Pour vous y aider, nous avons inclus certaines exceptions connues et légitimes. Pour autant, nous vous recommandons de respecter les normes autant que possible (elles ont fait leurs preuves !) et de réfléchir posément aux conséquences lorsque vous vous en éloignez.

Tout comme la plateforme Anaplan, ce document est appelé à évoluer. Nous ajouterons et mettrons à jour les règles en fonction des nouvelles fonctionnalités. L'échange de vues est une valeur fondamentale chez Anaplan, et nous souhaiterions avoir vos commentaires pour nous permettre d'apporter toute amélioration et correction. Chacun de ces commentaires contribuera à améliorer et raffiner le *Planual*.

Note du traducteur

Je me suis efforcé de transposer en français ces règles dans un langage concis et clair, tout en restant précis.

J'ai choisi de ne pas systématiquement traduire les termes et commandes applicatifs en les conservant tels qu'ils apparaissent dans les menus Anaplan, par contre ils sont identifiés en *italique* (par exemple *Lists*, *Versions*, etc....). Les noms de fonctions sont gardés tels quels, en majuscules (par exemple CURRENTPERIOD). Veuillez noter également que Les liens externes renvoient à des ressources documentaires en anglais.

Merci aux relecteurs Anaplanners, clients comme salariés Anaplan, qui ont contribué à cet effort d'adaptation de ce formidable guide dans notre belle langue !

Yves Lévenard – Customer Success Business Partner – France

INTRODUCTION	2
LES REGLES DU ZEN ANAPLAN	6
10 COMMANDEMENTS	6
PLANS	7
CONVENTIONS DU PLANUAL	8
1 CENTRAL LIBRARY – LA BIBLIOTHEQUE CENTRALE	9
1.01 TIME	9
1.02 VERSIONS	12
1.03 UTILISATEURS ET ROLES (<i>USERS AND ROLES</i>)	13
1.04 PANNEAU DE CONTENU (<i>CONTENTS</i>)	14
1.05 LISTES	15
1.06 <i>SUBSETS</i>	19
1.07 <i>LINE ITEM SUBSETS</i>	20
1.08 EMOJIS	21
2 ENGINE – LE MOTEUR	21
2.01 MODULES	21
2.02 FORMULES	27
2.03 <i>LINE ITEMS</i>	32

2.04 SAVED VIEWS - VUES	33
3 UX - PRINCIPES DE L'INTERFACE UTILISATEUR	34
3.01 HIERARCHIE DES INFORMATIONS	34
3.02 SMART GROUPING – LE REGROUPEMENT INTELLIGENT	35
3.03 REDUISEZ LA CHARGE VISUELLE	36
3.04 DIVULGATION PROGRESSIVE	37
3.05 COHERENCE ET NORMES	38
3.06 AIDE ET INSTRUCTIONS	39
3.07 UTILISEZ LA BONNE REPRESENTATION DE DONNEES – GRILLES ET GRAPHIQUES	39
3.08 INFORMEZ LES UTILISATEURS DE LEUR AVANCEMENT	42
3.09 ADAPTEZ A LA REALITE DES UTILISATEURS	42
3.10 PARLEZ FREQUEMMENT AUX UTILISATEURS	43
4 UX BUILD – PARAMETRAGE DES TABLEAUX DE BORD	44
4.01 APPS	44
4.02 DASHBOARDS	46
4.03 FILTRES	49
5 DATA INTEGRATION – INTEGRATION DES DONNEES	52
5.01 ACTIONS	52
5.02 PROCESS	53

5.03	MODELES SOURCES	54
5.04	IMPORTS	54
5.05	EXPORTS	57
5.06	<i>IMPORT DATA SOURCES</i> – SOURCES D’IMPORT	58
5.07	DATA HUB	59
6	<i>APPLICATION LIFECYCLE MANAGEMENT</i>	62
6.01	<i>REVISION TAGS</i> - BALISES DE REVISION	62
6.02	<i>PRODUCTION LISTS</i> – LISTES DE PRODUCTION	63
6.03	ARCHITECTURE	65
6.04	<i>DEPLOYED MODE</i>	66
6.05	GESTION DES CHANGEMENTS EN COURS DE DEVELOPPEMENT	67
7	EXTENSIONS	68
7.01	EXCEL	68
7.02	POWERPOINT	69



Les règles du Zen Anaplan

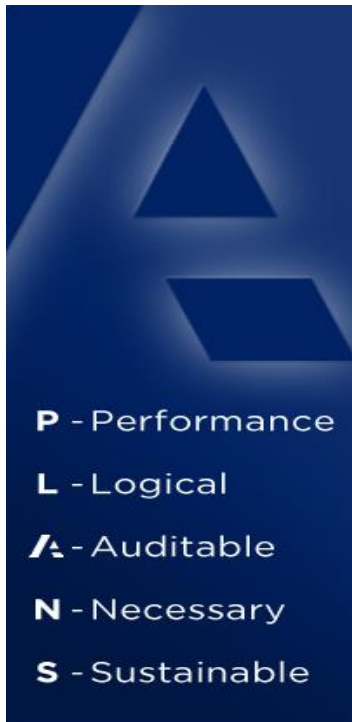
10 commandements

- Ce n'est pas parce que tu PEUX que tu DOIS
- Calcule une fois – référence plusieurs fois
- Préfère la simplicité à la complexité
- Ne calcule que lorsque de besoin
- La sparsité n'est pas toujours ton ennemie
- Fais grand mais rapide plutôt que petit et lent
- Favorise le booléen, et méfie-toi du texte
- Tu ne concaténeras point les textes
- Ne laisse pas les exceptions compliquer ton modèle
- SUMS et LOOKUPS sont tes amis, mais jamais ensemble

PLANS

PLANS est un ensemble de directives qui visent à aider à la conception et la structuration des modèles Anaplan. Adoptées dans leur globalité, ces bonnes pratiques permettent de fixer un socle commun référent pour tous les utilisateurs : Anaplanners, équipes et toute personne devant leur transmettre des modèles.

L'acronyme **PLANS** signifie :



Performant

Certaines formules et structures fonctionnent mieux que d'autres pour optimiser le moteur Anaplan Hyperblock™. Votre respect des normes **PLANS** aura pour corollaire naturel l'amélioration des **performances** du moteur. Bien sûr il y aura parfois des circonstances qui amèneront à « enfreindre les règles », mais le principal est d'être conscient des conséquences des choix faits et d'utiliser ces exceptions avec précaution.

Logique

Construisez vos modèles et formules à des fins **logiques**. Les calculs complexes sont plus efficaces lorsqu'ils sont décomposés de manière logique. Créez des modules à objectif unique suivant une structure logique de saisie (*Input*), de calcul (*Calculation*) et de restitution (*Output*), articulés avec des modules de données (*Data*) et d'administration (*System*).

Auditable

Utilisez des notes et des commentaires pour décrire les calculs. Décomposez les formules pour améliorer compréhension et performances ainsi que pour faciliter la maintenance future.

Nécessaire

Ne dupliquez pas les expressions ; stockez et calculez une fois, référencez plusieurs fois; utilisez la dimensionnalité appropriée, calculez uniquement lorsque **nécessaire**.

Stable

Construisez en tenant compte de l'avenir et pensez aux cycles des processus et aux mises à jour. Rendez le modèle flexible mais **stable** afin de faciliter les modifications ultérieures. Évitez le codage « en dur » (p. ex. *SELECT* avec des références directes à des items de listes).

Voyez ici la page d'accueil [PLANS sur Anaplan Community](#)

Conventions du *Planual*

Le *Planual* est organisé en articles, chapitres, règles et Exceptions.

Les règles sont numérotées dans le cadre des articles et du chapitre :

1.01-01 Ne jamais utiliser SELECT avec la dimension *Time*

Numéro d'article = 1 (*Central Library*)

Numéro de chapitre = 01 (Temps)

Numéro de règle = 01 (Ne jamais utiliser SELECT avec la dimension *Time*)

Les références à d'autres articles sont affichées sous l'énoncé de la règle :

Voir la règle : [Évitez la fonction SELECT](#)

Vous trouverez également des références à des articles *Anaplan Community* ou *Anapedia* qui détaillent plus avant le contenu de la règle – ces liens vous renverrons à des pages en anglais :

Best practice : [Time Range Application](#)

Les exceptions aux règles sont indiquées sous la règle en italique et avec un suffixe lettré :

1.01-01a Périodes génériques

1.01 Time

1.01-01

Ne jamais utiliser SELECT avec la dimension *Time*

Ceci va à l'encontre de l'aspect durable des principes PLANS, et le codage « en dur » posera problème lors de la mise à jour de la temporalité (*timescales*) du modèle. La meilleure option consiste à utiliser des modules avec des items formatés Temps (*Time*) à référencer dans des formules SUM ou LOOKUP.

Voir la règle: [2.02-14 Eviter d'utiliser SELECT](#)

1.01-01a Périodes génériques

Il est possible d'utiliser SELECT avec des périodes génériques telles que Actual Period, Current Period, YTD, YTG, ALL Periods

1 *Central Library* – La bibliothèque centrale

Cette section décrit les règles de la *Central Library* d'Anaplan. Cette "bibliothèque Générale" recense les **dimensions** du modèle.

1.01 Time

1.01-01

Ne jamais utiliser SELECT avec la dimension *Time*

Ceci va à l'encontre de l'aspect durable des principes PLANS, et le codage « en dur » posera problème lors de la mise à jour de la temporalité (*Timescales*) du modèle. La meilleure option consiste à utiliser des modules avec des items formatés Temps (*Time*) à référencer dans des fonctions SUM ou LOOKUP.

Voir la règle : [2.02-14 Evitez d'utiliser SELECT](#)

1.01-01a Périodes génériques

Il est possible d'utiliser SELECT avec des périodes génériques telles que Actual Period, Current Period, YTD, YTG, ALL Periods.

1.01-02

Utilisez par défaut *Model Calendar*

Comme le calendrier modèle est dynamique, il doit être utilisé pour la majorité des modules. Choisissez les paramètres de temps appropriés qui couvrent la plupart des besoins.

Best practice : [Time Range Application](#)

1.01-03

Utilisez *Current Period*

L'utilisation de la période actuelle (*Current Period*) dans les paramètres de temps permet l'utilisation des fonctions `CURRENTPERIODSTART()` et `CURRENTPERIODEND()` et ainsi être utilisé pour créer des modules de recherche référençant automatiquement la période en cours.

1.01-03a Unité de temps Jour

La granularité la plus faible pour les paramètres de temps est hebdomadaire. Par conséquent, si la période en cours doit refléter une échelle de temps quotidienne, le paramètre Current Period ne peut pas être utilisé.

1.01-03b Mises à jour par des non-administrateurs

Si vous avez besoin que d'autres utilisateurs que les administrateurs du Workspace mettent à jour le paramètre, un Line Item formaté peut être utilisé à la place.

1.01-05

Excluez par défaut les sous-totaux

Désactivez les paramètres « *include* » par défaut et n'incluez-les que si cela est absolument nécessaire. Ces options *Quarter totals*, *Year to date*, *Year to go*, etc... ajouteront autant de calculs d'agrégation dans chacun des modules dimensionnés avec la dimension *Time*, alourdissant d'autant les traitements. Vérifiez donc en avoir réellement besoin dans tous ces modules avant de les activer.

Notez que vous pouvez définir différents paramètres d'inclusion pour chacun des *Time Ranges*.

1.01-06

Convention de nommage des *Time Range*

Le nom doit être court, au format **FYxx-FYyy**, par exemple **FY20-FY21**. Cela permet à l'utilisateur de reconnaître immédiatement ce code dans la vue *Blueprint* des modules sans avoir à retourner consulter les paramètres de Temps (*Time Range*).

Un inconvénient pourrait être que vous devrez alors changer le nom des *Time ranges* au fil du temps ; dans ce cas vous pouvez préférer un terme générique, comme par exemple 'Historiques N-1, N-2'. Cela requiert moins de maintenance, mais n'est pas aussi clair.

Best practice : [Time Range Application](#)

1.01-07***Time Ranges***

Utilisez ces amplitudes de temps (*Time Ranges*) pour optimiser les modules pour lesquels le calendrier par défaut du modèle (*Model Calendar*) n'est pas approprié. Tenez compte de la dimensionnalité des données et configurez le *Time Range* en conséquence.

Best practice : [Time Range Application](#)

1.01-08**Unité de temps Journalière sur de grandes amplitudes de temps (*Timescales*)**

Posez-vous sérieusement la question de la nécessité d'une granularité quotidienne sur de grandes amplitudes de temps. Utilisez des *Time Ranges* pour borner de manière limitative les calendriers quotidiens et utiliser la fonction PREVIOUS plutôt que CUMULATE.

Voir la règle : [Quand utiliser PREVIOUS plutôt que CUMULATE ?](#)

Best practice: [Time Range Application](#)

Best practice: [Performance Issues with Cumulate Over Lists](#)

1.02 Versions

1.02-01

Utilisez le paramètre *Current Version*

L'utilisation de *Current* (ainsi que d'*Actual*) dans les paramètres de versions permet par la suite d'utiliser les fonctions CURRENTVERSION() et ACTUALVERSION() dans les calculs et les instructions SELECT ainsi que les fonctions ISCURRENTVERSION et ISACTUALVERSION() pour les valeurs booléennes. *Current version* agit également comme *Top Level* de la dimension *Versions* ; ainsi dans le cas d'un module source avec *Versions* lié à un module cible qui n'a pas la dimension, ce dernier récupèrera automatiquement la valeur *current version* – nul besoin d'insérer des fonctions ou un SELECT !

Voir la règle : [2.02-14 Avoid using SELECT](#)

1.02-02

N'utilisez pas de formules dans les paramètres de version

Bien qu'il s'agisse d'une solution simple et élégante pour des modèles simples, elle fait augmenter significativement la taille des modèles à chaque fois que des versions sont utilisées dans un module - même si vous n'utilisez pas les variances ainsi pré-calculées. En outre, dans les modules comprenant 3 dimensions ou plus et des lignes de calcul et des agrégations, les résultats peuvent ne pas être toujours précis car vous ne pourrez pas modifier la *summary method* pour une formule de version dans les paramètres versions.

1.02-03

Comment utiliser au mieux les paramètres *Edit From / Edit To*

Utilisez ces paramètres de préférence pour borner de manière large les accès en lecture/écriture aux versions – et pour un contrôle des accès plus granulaire, utilisez les techniques d'accès dynamique aux cellules (*Dynamic Cell Access*).

1.02-04

Ne reculez jamais la date de *Switchover*

L'utilisation du *Switchover* entraîne automatiquement l'effacement des périodes historiques antérieures. Déplacer cette date vers l'arrière/dans le passé reviendrait à remettre à Zéro les données précédemment stockées dans les périodes correspondantes. Donc pensez à modifier la date de *switchover* uniquement vers l'avenir.

1.02-05

Évitez d'avoir trop de *Versions*

La dimension *Versions* incorpore nativement des fonctionnalités qui n'existent pas dans les listes, telle que la *switchover date*. Cependant un nombre conséquent de version (plus de 10 notamment) peut avoir un réel impact sur les performances. Le cas échéant, reconsidérez le besoin des fonctionnalités natives de *Versions*, et envisagez de paramétrer une liste (dans *General Lists*) pour porter une partie de vos versions.

Best practice : [To Version or not to Version](#)

1.03 Utilisateurs et rôles (*Users and Roles*)

1.03-01

N'accordez l'accès en écriture aux listes que si nécessaire

À moins que les utilisateurs finaux n'aient besoin de modifier la liste (ajouter, supprimer des membres), l'accès n'a pas besoin d'être défini. L'ajout de rôles aux listes augmente l'utilisation de la mémoire, donc n'utilisez-la que si nécessaire. Un utilisateur peut modifier les données d'un module sans avoir besoin de le faire pour accéder à la liste

Best practice : [Pre-Allocation in Lists \(and Impacts to Model Performance\)](#)

1.03-02

Les performances et la taille peuvent dépendre de la liste des utilisateurs

Lorsque la liste des utilisateurs est fortement utilisée, le nombre d'utilisateurs ajoutés peut avoir un impact significatif en la matière. Soyez-en conscient et supprimez les accès au modèle des utilisateurs du *Workspace* lorsqu'ils ne sont pas requis.

1.03-03

Définir un tableau de bord d'accueil (*Landing Dashboard*) pour chaque rôle

Les différents rôles devraient logiquement avoir chacun un point d'entrée différent. Si possible, évitez d'utiliser une page d'accueil générique et créez un tableau de bord spécifiquement adapté à chaque rôle.

1.04 Panneau de contenu (*Contents*)

1.04-01

Masquez les modules aux utilisateurs finaux

Les utilisateurs devraient saisir des données uniquement par le biais de tableaux de bord. Supprimez donc les modules du panneau *Contents* pour eux.

1.04-02

Utiliser l'option *Show New Content* avec des *Functional Areas* différenciées

Pour réduire la maintenance, créez des *Functional Areas* distinctes pour les tableaux de bord et les modules (par exemple « Restitution » d'un part et « modules de restitution » d'autre part). Puis dans les paramètres pour *Contents*, activez l'option *Show New Content* pour toutes les *Functional Areas* utilisées pour les tableaux de bord et désactivez-la pour toutes les *Functional Areas* utilisées pour les modules. Ainsi les tableaux de bord nouvellement créés seront automatiquement ajoutés au panneau *Contents* par défaut lors de la création de nouveaux rôles, alors que les modules ne le seront pas.

1.05 Listes

1.05-01

Convention de nommage

Les hiérarchies devraient être préfixées d'une lettre indiquant le nom de la hiérarchie et d'un nombre indiquant le niveau. Par exemple, **P1** pour les catégories de produit, **P2** pour les familles de produit et enfin **P3** pour les produits. Il n'est pas nécessaire d'ajouter « L » comme préfixe aux listes.

Voir la règle : [1.08-01 N'utilisez pas les Emojis](#)

Best practice : [Good Practice Naming Conventions](#)

1.05-02

Utilisez toujours un code

L'utilisation de codes est plus efficace pour le chargement et l'utilisation de listes, donc efforcez-vous de toujours avoir un code pour les listes. Ceci est particulièrement important pour les listes de type *numbered lists*.

Voir la règle : [1.05-04 Les Numbered Lists doivent systématiquement avoir un code](#)

Best practice : [Hierarchy Management](#)

1.05-02a listes non hiérarchiques statiques

Pour des listes statiques simples (comme Oui/non), il est acceptable de ne pas avoir de code. Notez que même dans ce cas il serait utile d'utiliser les codes O/N.

1.05-03

Évitez d'utiliser les propriétés de listes

Les propriétés de listes pourraient être utilisées aux mêmes fins que les *Line Items*, mais elles ont de nombreuses limites. Restez simple et concentrez tous vos calculs en un seul endroit : les *Line Items* au sein de modules.

Best practice : [Data Hubs Purpose and Peak Performance](#)

1.05-03a Référez les Line items

Lorsque les exceptions suivantes sont utilisées, référez les Line items à l'aide d'une formule afin de garantir une trace d'audit.

1.05-03b Numbered lists (et actions connexes)

Les actions de type Assign, et les filtres associés nécessitent des propriétés de liste

1.05-03c Exports

Les propriétés de liste peuvent être utilisées comme étiquettes (labels) dans les exports

1.05-03d Conditions à la navigation dans les tableaux de bord

Les propriétés de liste sont nécessaires pour faciliter la navigation vers différents tableaux de bord en fonction de la sélection faite dans la liste

1.05-03e Liste déroulante dépendante

Les propriétés de liste sont nécessaires pour créer des listes dépendantes et leurs critères dépendante

1.05-04

Les Numbered Lists doivent systématiquement avoir un code

Si vous utilisez des actions *Create* ou *Assign* pour des *Numbered lists*, essayez d'incorporer une action pour « création de code » dans le cadre d'un autre processus (*process*).

Voir la règle : [1.05-02 Utilisez toujours un code](#)

Best practice : [Hierarchy Management](#)

Best practice : [Data Hubs Purpose and Peak Performance](#)

1.05-04a Combinaison de propriétés

S'il n'y a pas de code dans les données source, vous devrez utiliser une combinaison de propriétés. Évitez ces situations autant que possible, car cela ralentit les imports et il est plus difficile de travailler sans code.

1.05-05

Formatez la propriété *Display Name* en tant que liste si possible

Formatez cette propriété 'Nom d'affichage' pour les listes numérotées en format liste et non en texte. Ceci est plus efficace, épargne la création d'un élément, réduit les champs de texte et simplifie les mappings.

1.05-06**Utilisez le *Top Level* uniquement pour le niveau parent ultime**

Utilisez uniquement le *Top Level* pour les listes qui auront besoin de sommes – ceci exclut donc les codes de devise, les valeurs Vrai/Faux (indicateurs) ou les enfants au sein de listes composites.

Best practice : [Top Level Item and Parent Hierarchy](#)

1.05-07**Évitez le *Top Level* pour les *flat lists* volumineuses**

Si vous avez besoin de totaux, ajoutez plutôt des parents fictifs pour rendre les calculs plus efficaces.

Best practice : [Top Level Item and Parent Hierarchy](#)

1.05-08**Utilisez des listes composites plutôt que des listes disparates**

Les listes composites sont plus flexibles et permettent un calcul plus efficace ; essayez donc de former des hiérarchies adéquates dans la mesure du possible

1.05-08a Hiérarchies pour plans de comptes ou rapports financiers

Les hiérarchies de plans de comptes et de rapports financiers peuvent être gérées dans des listes non composites

1.05-09**Créer des titres intercalaires dans les *General Lists***

Insérez des titres intercalaires (listes fictives) afin d'organiser les listes générales, et présentez les *Subsets* et *Line Item Subsets* en bas

Voir la règle : [1.08-01 N'utilisez pas les Emojis](#)

1.05-10**Ne supprimez jamais une liste pour la recharger (sur une base quotidienne)**

Si vous devez rafraichir fréquemment une liste, utilisez une clé unique pour mettre à jour les valeurs plutôt que de supprimer et recharger la liste entière.

Best practice : [Data Hubs Purpose and Peak Performance](#)

Best practice : [Pre-Allocation in Lists \(and Impacts to Model Performance\)](#)

1.05-11**N'incorporez pas de valeurs numériques ou de dates dans le code d'une liste**

Les dates et les valeurs sont à considérer comme des données uniquement – ces notions ne doivent pas faire partie d'un code.

Best practice : [Data Hubs Purpose and Peak Performance](#)

1.05-12**Utilisez des formules pour déduire les propriétés d'une liste**

Il est très souvent possible de déduire les propriétés d'une occurrence de liste à partir de son code ; calculer des valeurs par formule est plus efficace que de stocker des champs texte.

Best practice : [Data Hubs Purpose and Peak Performance](#)

1.05-13**Utiliser des listes à plat pour stocker les métadonnées (dans un module système)**

Évitez d'avoir des hiérarchies dans votre modèle **Data Hub** ; elles ne doivent s'appliquer qu'aux modèles de planification.

Voir la règle : [5.07-01 Aucune hiérarchie](#)

1.06 Subsets

1.06-01

Convention de nommage

Préfixez les codes de vos *Subsets* avec le nom de la liste liée (par exemple 'P3 Produits : Active?' pour identifier les produits actifs au sein de la liste 'P3 Produits')

Voir la règle : [1.08-01 N'utilisez pas les Emojis](#)

Best practice : [Good Practice Naming Conventions](#)

1.06-02

N'utilisez pas de *Subsets* pour des listes volumineuses

Il est préférable de créer une liste à part si le *subset* représente plus de 75 % de la liste originelle.

1.06-03

Évitez les *Subsets* à occurrence unique

Si possible, essayez d'éviter les *subsets* comportant une unique occurrence. S'il y a un *Top level* défini pour la liste, le *subset* comporterait *de fait* toujours deux membres.

1.07 Line Item Subsets

1.07-01

Convention de nommage

Préfixez vos codes avec 'LIS:' puis un descriptif ou bien le module utilisé dans le *Line Item Subset*. Pour les *Line item subsets* utilisant plusieurs modules, utilisez un nom générique décrivant le mieux leur usage.

Voir la règle : [1.08-01 N'utilisez pas les Emojis](#)

Best practice : [Good Practice Naming Conventions](#)

1.07-02

Line Item subsets pour Version

Utilisez des *Line Item subsets* afin de créer différentes formules numériques et ainsi éviter des séries de conditions IF.

Best practice : [Line Item Subsets Demystified](#)

Best practice : [Decreasing the Length of Your Formulas](#)

Best practice : [Variance Analysis With Native Versions Made Easy](#)

1.07-03

Désactivez les options *Summary* sur les *Line Items Collect()*

Vous devriez désactiver les options *Summary* pour chaque *Line Item* utilisant la fonction COLLECT(). Si vous les laissez actives, ces options génèreraient des totalisations qui n'ont en général aucun intérêt.

Voir la règle : [2.01-10 Modules de calcul](#)

Voir la règle : [2.03-01 Désactivez par défaut les options Summary](#)

1.08 Emojis

1.08-01

N'utilisez pas les Emojis

N'utilisez pas d'Emoji ou de symboles dans les noms de listes, de modules ou d'actions. Ils peuvent occasionner des problèmes lors d'imports ou exports de données et sur certaines versions de navigateurs.

1.08-01a Intercalaires

Vous pouvez utiliser des emojis pour les intercalaires créés dans General Lists, Modules, Functional Areas, ...

2 Engine – Le moteur

Cette section décrit les règles relatives au moteur Anaplan. Au cœur d'Anaplan se trouve le moteur In-Memory **Hyperblock** qui combine la **flexibilité** des tableaux avec leurs feuilles de calcul, l'**évolutivité** et les **capacités d'interrogation** des bases de données relationnelles ainsi que les **capacités de calcul/agrégation** des cubes multidimensionnels.

2.01 Modules

2.01-01

Convention de nommage

Aussi courte que possible ! Utilisez des abréviations à caractères alphanumériques pour identifier les modules à partir d'un descriptif long. Il n'est pas nécessaire de préfixer les modules selon les lettres de l'acronyme DISCO ; il vaut mieux construire des préfixes basés sur les *Functional Areas* paramétrées.

Voir la règle : [4.02-10 Désactivez le nom du module](#)

Anapedia : [Naming convention](#)

2.01-01a gain de place pour les modules utilisateurs finaux

Les modules laissés apparents aux utilisateurs finaux peuvent être renommés avec un nom plus descriptif si nécessaire, afin d'économiser de l'espace sur le tableau de bord. Mais attention à ne pas trop « entasser » d'éléments sur un tableau de bord conformément aux bonnes pratiques de conception de l'expérience utilisateur.

2.01-02

Functional Areas

Utilisez les *Functional Areas* pour classer au mieux les modules de manière logique.

2.01-03

Utilisez des modules fictifs comme intercalaires

Créez ces modules vides afin de matérialiser visuellement des sections et aider à ordonnancer la liste – mais sans utiliser d’Emojis ou caractères spéciaux.

Voir la règle : [1.08-01 N'utilisez pas les Emojis](#)

2.01-04

Appliquez la méthodologie DISCO pour le paramétrage des modules

Regroupez les modules de même type au sein de chaque *Functional Area*. Accordez structure et paramétrage au type de module.

Best practice : [Best Practices for Module Design](#)

Best practice : [Reduce Calculations for Better Performance](#)

Best practice : [Formula-Structure-for-Performance](#)

2.01-05

N’utilisez pas DISCO pour les *Functional Areas*

La convention DISCO est conçue pour typer les modules, mais PAS pour les classer. Ainsi il pourra y avoir plusieurs types de modules regroupés dans la même *Functional Area*.

Voir la règle : [2.01-04 Appliquez la méthodologie DISCO pour le paramétrage des modules](#)

2.01-05a Modèles simples

*Si le modèle est simple, DISCO peut également être utilisé pour nommer les *Functional Areas*.*

2.01-06

Évitez les vues subsidiaires

Les vues subsidiaires (*subsidiary views*) sont difficiles à utiliser et à auditer. Évitez-les autant que possible, en particulier dans les modules de calcul ; créer plutôt des groupes de modules de calcul utilisant les mêmes dimensions.

Voir la règle : [2.01-12 Regroupez les formules de dimensionnalité identique](#)

2.01-06a Affichage dans un Dashboard pour analyse ou filtrage

Parfois, pour améliorer l'expérience de l'utilisateur final, les attributs doivent être affichés sur les tableaux de bord.

2.01-06b Formules de ratio

Les ratios doivent généralement apparaître dans un module, même s'ils ont rarement la même dimensionnalité.

2.01-06c Synchronisation de certains tableaux de bord

Si vous avez des tableaux de bord à synchroniser avec un module, un Line Item spécifique est parfois nécessaire.

2.01-06d Line items nécessaires à des tris

Parfois, les Line Items créés pour faciliter un tri n'ont pas besoin de la dimensionnalité complète du module.

2.01-06e Modules de reporting ou d'export

Pour filtrer dans des rapports ou pour fournir des attributs à l'exportations, vous pouvez avoir besoin de Lines Items pour les attributs sans dimension. Assurez-vous tout de même d'utiliser des vues subsidiaires afin de réduire les calculs.

2.01-07

Modules système *Time Settings*

Créez toutes les fonctions et filtres spécifiquement nécessaires à la dimension *Time* dans des modules distincts par semaine, mois, trimestre, année, etc. Ils se recalculeront uniquement lors de l'ouverture du modèle ou lorsque les paramètres de temps seront modifiés (dans la dimension *Time*).

2.01-08

Chaque hiérarchie importante devrait avoir un module système

Créez un module système pour gérer toutes les données et tous les attributs permanents d'une hiérarchie – si elle est utilisée. Créez au minimum des *Lines Items* pour le code et le parent. Si la hiérarchie n'est référencée dans aucune formule et dans aucun filtre ou sélecteur de tableau de bord, alors le module système ne sera pas nécessaire.

Best practice : [Best Practices for Module Design](#)

2.01-09

Utilisez des modules pour constantes et *Lookups*

Créez des modules sans dimensions pour stocker des constantes telles que des hypothèses fixes de temps (*Time*), ou des valeurs qui apparaîtraient de manière trop récurrente dans des fonctions "SELECT". Ces dernières seront référencées dans les modules cibles grâce à des fonctions LOOKUP.

Voir la règle : [Évitez la fonction SELECT](#)

2.01-10

Modules de calcul

Désactivez par défaut les options *Summary* en les positionnant sur la valeur *None*. Il est préférable d'utiliser des fonctions SUM pour gérer certaines agrégations en aval, selon les besoins. Tous les niveaux d'agrégation ne sont pas forcément nécessaires.

Best practice : [Reduce Calculations for Better Performance](#)

Best practice : [Formula-Structure-for-Performance](#)

2.01-10a Tous les niveaux d'agrégation sont nécessaires

Si tous les niveaux d'agrégation sont nécessaires, il est plus rapide d'utiliser l'agrégation native que la fonctions SUM

2.01-11

Organisez l'ordre des dimensions

Assurez-vous que l'ordre des dimensions soit cohérent entre les différents modules. En effet, les calculs sont plus rapides lorsque des modules à dimensions identiques sont paramétrés avec leurs dimensions communes dans l'exact même ordre (dans la liste '*Applies to*'). Pour les calculs, c'est d'ailleurs cet ordre qui a plus d'importance que le nombre de dimensions utilisées.

Best practice : [Dimension Order](#)

2.01-12

Regroupez les formules de dimensionnalité identique

Créer des modules de calcul pour rassembler les *Lines Items* utilisant les mêmes dimensions. Évitez d'utiliser les vues subsidiaires.

Voir la règle : [2.01-06 Évitez les vues subsidiaires](#)

Best practice : [Line Item Subsets Demystified](#)

Best practice : [Decreasing the Length of Your Formulas](#)

2.01-13

Séparer les données de transaction des données de références

Conservez les données transactionnelles/d'exploitation (variant dans le temps) dans des modèles distincts des modèles de référence qui regroupent attributs et propriétés (réputés fixes).

Best practice : [Data Hubs Purpose and Peak Performance](#)

2.01-14

Évitez de combiner des filtres avec l'option *Select Levels to show* dans les vues

N'utilisez pas *Select Levels to show* en même temps que des filtres sur la même hiérarchie, c'est redondant et inefficace. Optez soit pour l'un, soit pour l'autre. Préférez l'utilisation d'un *Line Item* booléen dans un module Système (avec option *Summary à None*) pour trier vos vues.

2.01-15

Gérez vos filtres dans des modules systèmes

Efforcez-vous de centraliser vos critères de filtres dans des modules système. Les *Lines Items* créés à cet usage peuvent ensuite être réutilisés dans plusieurs modules cibles. N'oubliez pas de lister les modules cibles dans la section *Notes* du *Line Item* pour éviter toute suppression hâtive.

Voir la règle : [4.03-02 Gérez les filtres dans des modules dédiés](#)

2.01-16**Utilisez les *Data Tags***

Consignez dans cette section des modules leur type **DISCO** (*Data, Input, System, Calculation* ou *Output*) – sauf à décider d’exploiter ces Tags pour un autre usage.

2.01-17**Modules de gestion du DCA**

Créez des modules distincts pour gérer les croisements de dimensions nécessaires à la gestion d’accès selon la méthode *Dynamic Cell Access*. Créez autant de Lines Items que nécessaire.

Anapedia : [Dynamic Cell Access](#)

2.01-18**Évitez de dupliquer les modules volumineux**

La duplication d’un module contenant de nombreuses données, *a fortiori* lorsqu’il contient de multiples filtres et vues, peut être long et ‘bloquer’ l’accès au modèle le temps de la copie. Pour ces modules volumineux, il est bien probable que le re-paramétrage prendra finalement moins de temps que la duplication.

Voir la règle : [2.04-03 Gardez la vue par défaut simple](#)

2.01-19**Positionnez vos listes pour *Time* et *Versions* en premier**

Dans le cas où vous avez choisi de paramétrer des listes au lieu des dimensions natives pour gérer périodes et versions, assurez-vous de les placer en premier dans *General lists*. Ceci reflète la logique native de ces dimensions, et contribuera à garantir une bonne performance lors des interrogations entre vos listes et leur dimension native associée.

Best practice : [Dimension Order](#)

2.01-20

Paramétrez les dimensions appropriées à vos modules

A la création d'un module, ciblez bien l'étendue du calcul recherché et n'intégrez que les dimensions strictement nécessaires. Si votre formule ne retourne pas le résultat désiré, ne rajoutez pas les dimensions jusqu'à obtenir ce résultat : reconsidérer plutôt le calcul-même. Vérifiez également si vous n'auriez pas déjà un module *Calculation* avec les dimensions ciblées : il vous suffirait alors de simplement rajouter votre calcul au module existant.

Voir la règle : [2.01-04 Appliquez la méthodologie DISCO pour le paramétrage des modules](#)

Best practice : [Reduce Calculations for Better Performance](#)

2.02 Formules

2.02-01

Imbrications de IF

Évitez de combiner plusieurs *IF* dans une même formule. Il est préférable de décomposer la formule dans plusieurs *Line Items*, d'utiliser la fonction *LOOKUP* ou toute autre solution.

Best practice : [Line Item Subsets Demystified](#)

Best practice : [Decreasing the Length of Your Formulas](#)

2.02-02

Moins de 12 expressions dans une formule

Si il vous faut plus d'une phrase simple pour décrire ce que fait la formule, c'est que cette dernière est probablement trop longue. Veillez à ne pas empiler les fonctions.

2.02-03

Pas d'expressions récurrentes

Si l'expression est répétée dans la formule (ou dans d'autres modules), placez-la dans un *Line Item* distinct.

“Calcule une fois – référence plusieurs fois”.

2.02-04

Chaînes textuelles

Traitez les chaînes de texte avec précaution. Évitez les concaténations multiples et fractionnez les jointures communes dans des *Line Items*.

Voir la règle : [2.02-05 Concaténez dans la plus petite hiérarchie](#)

Voir la règle : [2.03-02 Évitez si possible le format TEXT](#)

Best practice : [Formula Optimization in Anaplan](#)

2.02-05

Concaténez dans la plus petite hiérarchie

Dans le cas où une concaténation est nécessaire, veillez à la gérer dans la hiérarchie la plus petite afin réduire la taille des chaînes de texte.

Voir la règle : [2.02-04 Chaînes textuelles](#)

Best practice : [Formula Optimization in Anaplan](#)

2.02-06

Comparaison de nombres

Il est plus rapide de vérifier que $A=B$ plutôt que $A-B=0$.

2.02-07

Utilisez des valeurs booléennes au lieu des 1 et 0

Les valeurs booléennes occupent le 1/8ème de l'espace-mémoire d'un nombre, donc utilisez *TRUE* ou *FALSE* - sauf si une valeur numérique est vraiment nécessaire.

2.02-08

N'utilisez jamais *SUM* et *LOOKUP* ensemble

L'utilisation des fonctions *SUM* et *LOOKUP* dans la même expression enclenche généralement des calculs extensifs et peuvent entraîner des calculs intermédiaires, surtout si la dimension *Time* est impliquée ou bien lorsque les structures source et cible sont très différentes. Le simple fait de les scinder dans différents modules et *Lines Items* réduit considérablement les calculs intermédiaires.

Best practice : [Formulas and their effect on model performance](#)

2.02-08a *LOOKUP* pour une constante

Si la fonction LOOKUP utilise un line item constant, la formule se comporte de fait comme un SELECT, ce qui lève l'objection de performance.

2.02-09

Règles d'agrégation (*ANY*, *ALL*, *FIRSTNONBLANK*)

Utilisez ces fonctions d'agrégation pour minimiser l'utilisation de *Line Items* et arguments *IF*.

2.02-10

Quand utiliser *PREVIOUS* plutôt que *CUMULATE* ?

Pour les grandes amplitudes de temps, l'utilisation de *PREVIOUS* donne des meilleures performances que *CUMULATE* en raison du nombre de «lectures» requises pour le calcul. Ainsi, l'expression devrait être : *'Calc line item' = 'data line item' + PREVIOUS('Calc line item')* plutôt que : *CUMULATE('data line item')*.

Best practice : [Performance Issues with Cumulate Over Lists](#)

2.02-10a *Courtes durées*

Lorsque le nombre de périodes est réduit, CUMULATE est plus rapide.

2.02-11

Évitez d'utiliser *TEXTLIST*

La fonction *TEXTLIST* mobilise beaucoup de mémoire pour les calculs et doit être évitée autant que possible ; une bonne alternative consiste à utiliser des indicateurs booléens avec l'option *ANY* au sein de modules systèmes (par dimension).

2.02-12

Ne jamais coder en dur les valeurs de listes

Évitez les références directes aux occurrences de liste, par exemple : *IF ITEM(list)=list.xxx*

2.02-13

Utilisez *POST* uniquement dans les cas spécifiques pour lesquels il a été créé

N'utilisez pas la fonction *POST* pour des situations de décalages temporels simples. *OFFSET*, *LAG* ou *MOVIGSUM* sont plus efficaces pour ces cas.

2.02-14

Évitez la fonction *SELECT*

Si possible, évitez le codage en dur avec la fonction *SELECT*. Gérez un module de constantes et utilisez *LOOKUP* pour vous y référer.

Voir la règle : [1.01-01 Ne jamais utiliser SELECT avec la dimension Time](#)

2.02-14a Versions

Vous pouvez utiliser SELECT pour les versions.

2.02-14b Pour les Top Levels dans des listes non-composites.

Dans les listes non composites, il n'est pas possible d'utiliser LOOKUP pour obtenir la valeur. Dans ce cas vous pourrez référencer le Top Level avec SELECT. Pour les autres cas, il est préférable d'utiliser des listes de parents fictives et des modules supplémentaires

2.02-15

FINDITEM pour les vides

La fonction *FINDITEM* est largement inefficace sur des listes contenant des vides, car elle doit de toute manière parcourir l'exhaustivité de la liste. Si la liste référencée contient peu de vides, alors recherchez d'abord les vides : *IF ISNOTBLANK(Line Item) THEN FINDITEM(List, Line Item) ELSE BLANK*. Si par contre la liste contient une majorité de vides, utilisez plutôt : *IF ISBLANK(Line Item) THEN FINDITEM(List, line item) ELSE BLANK*. Votre choix de formule dépendra donc de la proportion de vides dans vos listes : s'il y a une majorité de valeurs renseignées, vous utiliserez la première formule *ISNOTBLANK* / s'il y a une majorité de vides, vous utiliserez la seconde formule *ISBLANK*. Enfin, si vous avez la certitude que la liste ne contient aucun vide, vous pourrez alors simplement utiliser : *FINDITEM(List, line item)*

Voir la règle : [2.02-17 Placez la condition la plus courante en premier](#)

2.02-16

Utilisez des instructions conditionnelles pour échapper aux calculs inutiles

Lorsque vous utilisez des conditions multiples, efforcez-vous d'inclure une première condition qui, si elle est satisfaite, permettra de ne plus avoir à exécuter le reste du calcul.

2.02-17

Placez la condition la plus courante en premier

Pour l'efficacité de la formule, placez la condition avec l'occurrence la plus commune en premier dans la formule.

2.02-18

Décomposez les formules

Le moteur fonctionne plus efficacement lorsque les calculs sont divisés en *Line Items* distincts. Décomposez donc vos formules autant de fois que possible. C'est particulièrement vrai pour les calculs qui sont souvent référencés et/ou des calculs dont la formule ne change pas souvent.

Best practice : [Formula Structure for Performance](#)

Best practice : [Formula Optimization in Anaplan](#)

2.02-19

Évitez les longues chaînes de dépendances

Si possible, référez-vous dans vos calculs directement à la première source et évitez de créer de longues chaînes de dépendances. Ceci permet d'exécuter davantage de calculs en parallèle, ce qui augmente l'efficacité et réduit les temps de calcul.

Best practice : [Best Practices for Module Design](#)

2.02-20

N'utilisez pas RANK sur de grandes amplitudes de temps

La fonction *RANK* enclenche des calculs assez lourds et qui ne peuvent pas être exécutés de manière parallélisée. Utilisée sur de nombreuses périodes et/ou une liste significative de versions, elle sera peu performante. Il en va de même pour la fonction *RANKCUMULATE*.

2.02-21

N'utilisez pas la fonction *TIMESUM* pour la dimension *Time*

TIMESUM ne devrait pas être utilisée pour des *Line Items* dimensionné avec *Time*, car il s'appliquerait à chacune des périodes existantes - utilisez plutôt dans ce cas la fonction *MOVINGSUM*.

Best Practice - à voir et revoir sur le sujet :



2.03 Line Items

2.03-01

Désactivez par défaut les options *Summary*

La plupart des *Lines Items* de votre modèle n'auront pas besoin des calculs synthétiques déclenchés par les options *Summary*. Il est donc recommandé de les désactiver par défaut – quitte à les ajouter par la suite, si besoin. Ceci est particulièrement vrai pour les modules de calcul et les *Lines Items* de formatage conditionnel.

Voir la règle : [2.01-10 Modules de calcul](#)

2.03-01a Modules pour les utilisateurs

Les options Summary peuvent être nécessaires dans les modules de saisie ou de restitution exposés aux utilisateurs au travers des Dashboards et pages.

2.03-02

Évitez si possible le format *TEXT*

Anaplan est optimisé pour traiter les nombres et les booléens - comme la plupart des systèmes informatiques. Les formats de texte utilisent plus de mémoire que tout autre format en raison des techniques informatiques mises en œuvre pour stocker ces données. Réduisez donc leur utilisation autant que possible.

Voir la règle : [2.02-04 Chaînes textuelles](#)

2.03-03

Intercalaires et en-têtes

Assurez-vous que les *Lines Items* créés en tant qu'intercalaires ou en-têtes de rubriques dans les modules soient bien paramétrés avec le format *No Data*. Ceci évitera des calculs inutiles.

2.04 Saved Views - Vues

2.04-01

Convention de nommage

Donnez un nom significatif à la vue (par exemple « Tableau de bord », « vue filtrée », « Export », etc.). Par contre, il n'est pas nécessaire d'ajouter le nom du module dans le nom de la vue.

Anapedia : [Naming convention](#)

2.04-02

Filtres dans les vues

Filtrez sur une valeur booléenne et n'utilisez qu'un filtre par vue.

Voir la règle : [4.02-01 Utilisez des filtres efficaces](#)

Best practice : [Filter Best Practice](#)

2.04-03

Gardez la vue par défaut simple

Conservez la vue par défaut du module telle qu'elle est, sans masquage, ni filtrage ni mise en forme conditionnelle.

2.04-04

Vues imbriquées

Si possible, imbriquez vos dimensions par ordre de taille (du plus général au plus détaillé).

3 UX - Principes de l'interface utilisateur

Cette section couvre les principes de conception de l'expérience utilisateur, c'est-à-dire les éléments visibles et actionnés par les utilisateurs finaux de l'application – la *User eXperience* ou **UX**. Ils sont tirés de principes fondamentaux qui s'appliquent à toutes les applications, mais que nous avons adaptés ici à Anaplan.

3.01 Hiérarchie des informations

3.01-01

Principe

Une hiérarchie des informations bien faite facilite la lecture du contenu. Cela flèche un parcours que l'œil de l'utilisateur suivra de manière naturelle dans la page/sur l'écran.

3.01-02

Créez un fil narratif

Utilisez une page d'accueil pour fournir des informations clefs. Elle renseignera les nouveaux utilisateurs sur le contenu du modèle, tout en fournissant un résumé ainsi que des liens de navigation vers des pages plus détaillées.

3.01-03**Commencez par la synthèse**

Placez les informations synthétiques dans une première catégorie, puis les pages de données détaillées ensuite. Ajoutez des instructions pour guider et expliquer aux utilisateurs ce qu'il est attendu d'eux.

3.01-04**Ordre d'importance**

Réfléchissez à ce qui est le plus important et placez-le en haut de la page, par exemple les indicateurs-clefs (*KPIs*).

3.02 *Smart Grouping* – le regroupement intelligent

3.02-01**Principe de Gestalt**

Notre cerveau a tendance à regrouper les éléments pour simplifier la compréhension. Le regroupement visuel de vos pages aide l'utilisateur à comprendre ce qu'il voit et à trouver ce qu'il recherche plus rapidement.

Best practice : [Top 5 Design Best Practices for UX—Desktop and Mobile](#)

3.02-02**Table des matières**

Organisez votre contenu en utilisant des catégories qui soient logiques pour vos utilisateurs, par exemple par domaine fonctionnel, périmètre de responsabilités, processus... Pensez à votre fil narratif et regroupez et ordonnez vos pages en conséquence.

3.02-03**Regroupez les données connexes**

Regroupez les données similaires et/ou liées.

3.03 Réduisez la charge visuelle

3.03-01

Le principe de Hick

Mettez moins d'éléments sur une page pour une prise de décision plus rapide.

Best practice : [Top 5 Design Best Practices for UX—Desktop and Mobile](#)

3.03-02

Synthétisez les données

Lorsque vous travaillez sur vos pages d'accueil, essayez de réduire ou d'agréger les données de ces pages et proposez des liens sur le contenu pour aller vers des pages plus détaillées.

Voir la règle : [3.01-03 Commencez par la synthèse](#)

3.03-03

Restez simple

Limitez l'utilisation de la couleur dans les graphiques, ne surchargez pas la page et gardez-là aussi simple que possible.

3.03-04

Séparez-les

Pensez à utiliser plusieurs pages au lieu de concentrer tous les éléments sur une unique page.

3.04 Divulgateion progressive

3.04-01

Principe

Présentez vos données par étapes ; démarrez par les éléments simples et exposez la complexité lorsque vous en avez besoin et pas avant.

Best practice : [Top 5 Design Best Practices for UX—Desktop and Mobile](#)

3.04-02

Synthèse

Plutôt que de tout mettre sur une page, fournissez des informations clefs avec un lien vers une autre page qui fournira plus de détails aux utilisateurs.

Voir la règle : [3.03-02 Synthétisez les données](#)

3.04-03

Analyse détaillée

Sur les pages détaillées, les utilisateurs peuvent effectuer des tâches plus précises, telles que l'analyse détaillée, la manipulation et la modification des données. Idéalement, la principale grille de données de cette page devrait représenter une vue plus détaillée des informations de la page précédente.

3.04-04

Utilisez le panneau *Contents*

Utilisez le panneau de contenu pour placer des liens utiles afin de naviguer *vers* et *depuis* la page en cours, sans pour autant les insérer dans la page-même lorsque cela ne semble pas utile.

3.05 Cohérence et normes

3.05-01

La cohérence est essentielle

Les utilisateurs ne devraient pas avoir à se demander si des mots, des situations ou des actions différents signifient la même chose. Utilisez des formes (par exemple, des images, noms, mots...) et des couleurs homogènes pour communiquer.

3.05-02

Conventions de nommage

Nommez de manière identique les mêmes éléments dans les différentes pages ; notamment conservez la cohérence entre l'intitulé du lien à destination d'une page et le nom de la page ciblée, de sorte que les utilisateurs ne soient pas perdus lorsqu'ils se déplacent d'une page à l'autre.

3.05-03

Positionnement

Utilisez un positionnement cohérent des éléments tels que les liens, en plaçant la même information au même endroit sur chacune des pages.

3.05-04

Couleurs

Utilisez les mêmes couleurs pour les mêmes indicateurs affichés dans les différentes pages.

3.06 Aide et instructions

3.06-01

Aide

Fournissez un accès facile à l'aide à l'utilisation de vos applications.

3.06-02

Instructions

Utilisez du texte et des instructions dans la mesure possible, tout en veillant à rester clair et succinct afin de ne pas submerger l'utilisateur.

3.06-03

Info-bulles

Utilisez les info-bulles personnalisables pour décrire les visualisations et informer l'utilisateur sur ce qu'il voit.

3.07 Utilisez la bonne représentation de données – Grilles et graphiques

3.07-01

Grilles de données

Ces grilles sont le meilleur choix pour visualiser de grandes séries de données détaillées et les analyser, par exemple des chiffres de vente en fonction des produits et du temps. Par contre les grilles ne devraient pas être utilisées pour afficher des grandes tendances ou des comparaisons de données agrégées à un haut niveau.

3.07-02

Graphiques – Courbes

Les graphiques linéaires sont les meilleurs pour voir évoluer les tendances sur une période continue, par exemple des ventes mensuelles. Ils sont particulièrement utiles pour comparer plusieurs catégories de données, par exemple les revenus de différents produits par mois. Utilisez des couleurs contrastées pour que les utilisateurs puissent facilement différencier les séries. Affichez une légende pour indiquer clairement chaque série.

Best practice : [Line Charts](#)

3.07-03

Graphiques – Colonnes

Les graphiques à colonnes sont parfaits pour comparer les valeurs entre différentes lignes ou items, par exemple le réalisé par rapport aux prévisions. Les graphiques à colonnes sont également utiles pour comparer les items dans le temps, par exemple les ventes mensuelles.

Best practice : [Column Charts](#)

3.07-04

Graphiques – Barres

Les graphiques à barres sont parfaits pour comparer les valeurs entre différentes lignes ou items, par exemple le réalisé par rapport aux prévisions. Évitez de les utiliser pour comparer les valeurs dans le temps. Les graphiques à barres sont parfois préférables aux graphiques à colonnes dans le cas où les données ont des étiquettes trop longues pour être affichées sur l'axe des abscisses.

Best practice : [Bar Charts](#)

3.07-05

Graphiques – Colonnes/Barres empilées

Les graphiques à colonnes/barres empilées sont idéaux pour représenter les relations d'une partie à son ensemble sur plusieurs séries, par exemple un résultat mensuel divisé en marge et en charges. Utilisez des couleurs contrastées pour que les utilisateurs puissent facilement différencier les piles.

Voir la règle : [3.07-03 Graphiques – Colonnes](#)

Best practice : [Column Charts](#)

3.07-06

Graphiques – en secteurs et en anneaux

Les « camemberts » ou anneaux sont la meilleure option pour comparer les parties d'un même ensemble. Par contre, ils ne montrent pas les variations dans le temps.

Best practice : [Pie Charts](#)

3.07-07

Graphiques – en entonnoir

Les diagrammes en entonnoir sont souvent utilisés pour représenter les étapes d'un processus de vente et indiquer le montant des revenus potentiels pour chaque étape. Ce type de graphique peut également être utile pour identifier les problèmes éventuels dans les processus de vente d'une organisation. Un diagramme en entonnoir est similaire à un graphique en barres de pourcentage empilées.

Best practice : [Funnel Charts](#)

3.07-08

Graphes chronologiques

Un graphique chronologique est un moyen efficace de visualiser un processus par ordre chronologique. Comme les détails sont affichés graphiquement, les points importants dans le temps peuvent être facilement vus et compris. Ils sont souvent utilisés pour représenter l'avancement d'un projet, ces chronologies fonctionnent comme une sorte de calendrier des événements dans une période de temps spécifique.

Best practice : [Timeline Charts](#)

3.07-09

Graphique en cascade

Un graphe en cascade est généralement utilisé à des fins analytiques, en particulier pour comprendre ou expliquer la transition graduelle de la valeur d'une entité qui est soumise à une augmentation ou diminution de sa valeur. Typiquement, on l'utilisera pour montrer les différentes sources d'une variation de ventes ou de résultats entre deux périodes.

Best practice : [Waterfall Charts](#)

3.07-10**Combinaison de graphiques**

Les diagrammes combinés sont utiles lorsque vous souhaitez valider la relation entre deux variables ayant des amplitudes et des échelles différentes de mesure mais qui sont liées de manière significative.

Best practice : [Combination Charts](#)

3.08 Informez les utilisateurs de leur avancement

3.08-01**Donnez de la visibilité**

Idéalement les pages devraient toujours tenir les utilisateurs informés de ce qui se passe, par le biais de commentaires appropriés et dans un délai raisonnable. Le fait d'être informés de leur état d'avancement actuel dans le processus en cours aide les utilisateurs à décider de ce qu'ils doivent faire ensuite pour atteindre leurs objectifs, à déterminer si leurs actions ont été efficaces, ou bien simplement corriger une erreur.

3.09 Adaptez à la réalité des utilisateurs

3.09-01**Utilisez le langage des utilisateurs**

Utilisez leur langage avec des mots, des expressions et des concepts qui leurs sont familiers.

3.09-02**Prenez en compte le processus métier**

Il est conseillé de définir précisément le processus, et de le refléter dans Anaplan en suivant le cheminement et les conventions du processus réel lorsque vous positionnez les informations, de sorte qu'elles apparaissent dans un ordre naturel et logique.

3.10 Parlez fréquemment aux utilisateurs

3.10-04

Retours d'expérience

Interviewez les utilisateurs lors de votre première rencontre, exposez-leur vos idées et démontrez vos pages pendant la phase de *build*. Par la suite, vous pouvez de nouveau les interviewer ou les observer pendant qu'ils utilisent l'application. Identifiez où ils bloquent et à quels moments ils rencontrent des difficultés – et réalisez les ajustements nécessaires.

3.10-01

Pendant la conception

Mettez les utilisateurs au cœur de la conception. Consultez-les pour connaître leurs besoins et afin qu'ils vous aident à définir ce que vous allez construire et comment vous allez le construire.

3.10-02

Pendant les paramétrages

Échangez avec eux tout au long de la phase de *build* afin de vous assurer que vous êtes sur la bonne voie.

3.10-03

Après la mise en production

Vérifiez avec eux après le *Go Live* que tout fonctionne pour le mieux.

4 UX Build – paramétrage des tableaux de bord

Cette section traite de la construction des tableaux de bord Anaplan – les *Dashboards*, *Pages* et *Apps*.

4.01 Apps

4.01-01

Convention de nommage

Faites aussi court que possible. Utilisez des préfixes numériques pour les *Pages* pour obtenir un ordonnancement conforme dans la table des matières de l'App.

Best practice : [Understanding Anaplan's UX with Analogies, FAQs and Best Practices](#)

4.01-02

Utilisez les *Custom Views*

Utiliser ces *Custom Views* dans le paramétrage des pages permet d'actionner des options avancées disponibles uniquement dans ce contexte, par exemple pour les formatages conditionnels.

Best practice : [Understanding Anaplan's UX with Analogies, FAQs and Best Practices](#)

4.01-02a Vues récurrentes

Il peut être préférable d'utiliser des Module Views si les mêmes données sont présentées dans plusieurs types de cards (par exemple dans une grille/grid et un graphique/chart).

4.01-02b Largeur de colonnes

Utilisez des Module Views lorsque vous devez paramétrer des colonnes d'une certaine largeur.

4.01-03

Créez un module *Output* pour les *Custom Views*

Lorsque les *Lines Items* intermédiaires n'ont pas besoin d'être vus par les *Page Builders*, paramétrez un module *Output* sur lequel vous baserez votre *Custom View*.

Voir la règle : [2.01-04 Appliquez la méthodologie DISCO pour le paramétrage des modules](#)

4.01-04

Utilisez les *Boards* pour les données agrégées

Privilégiez le type de page *Boards* pour présenter des indicateurs-clefs, des graphiques, des analyses de variance, des vues synthétiques en provenance de plusieurs modules, des pages d'accueil.

Best practice : [Understanding Anaplan's UX with Analogies, FAQs and Best Practices](#)

4.01-05

Utilisez les *Worksheets* pour les données volumineuses

Privilégiez le type de page *Worksheets* pour des tableaux qui permettront de saisir et réviser de grandes quantités de données, de les pivoter, etc...

Best practice : [Understanding Anaplan's UX with Analogies, FAQs and Best Practices](#)

4.01-06

Créez des pages pour l'app. mobile

Quand vous créez des pages pour app. mobile, prêtez une attention particulière à la disposition spatiale des *cards* et anticipez bien la navigation des éléments (défilement à l'écran des *cards* de gauche à droite pour chaque ligne dans les pages *Board*). Pour un résultat optimal, envisagez de créer une app. spécifiquement dédiée à l'utilisation sur mobile.

Best practice : [Top 5 Design Best Practices for UX—Desktop and Mobile](#)

4.01-07

Utilisez les *card templates*

Lorsque plusieurs cartes du même type utilisent un même module source, pensez à l'enregistrer comme modèle/*card template* – vous pourrez ainsi paramétrer plus rapidement vos *Boards* et *Worksheets* en réappliquant ce *card template*.

4.02 Dashboards

4.02-01

Démarrez la conception par l'interface utilisateur

Dès le début du processus de conception, constituez vos récits utilisateurs (les *User Stories*) et intégrez les principes *UX*.

De fait : démarrez par la finalité !

Best practice : [Designing Dashboards](#)

4.02-02

Évitez le défilement horizontal

Évitez les situations où l'écran défile horizontalement. Il est préférable de figer la grille et d'opérer le défilement au sein de la grille.

Voir la règle : [4.02-03 Tenez compte de la taille de l'écran des utilisateurs finaux](#)

4.02-03

Tenez compte de la taille de l'écran des utilisateurs finaux

Ne construisez pas en fonction de la taille de votre seul écran. Ciblez la taille la plus petite d'écran que vos utilisateurs sont susceptibles d'utiliser – et adaptez à cette taille. Vous devriez modifier la résolution de votre écran pour tester l'apparence qu'auront vos tableaux de bord pour certains utilisateurs.

Voir la règle : [4.02-02 Évitez le défilement horizontal](#)

4.02-04

Ajoutez des sélecteurs de page plutôt que des sélecteurs de module

Utilisez de préférence les sélecteurs de page plutôt que les sélecteurs de modules/grilles, en particulier si les dimensions à sélectionner sont répétées en plusieurs endroits du tableau de bord.

Voir la règle : [4.02-05 Désactivez les sélecteurs de page de module](#)

4.02-05

Désactivez les sélecteurs de page de module

Vous pouvez afficher les sélections faites dans chaque élément (grille, module, ...) mais efforcez-vous de faire opérer le choix des valeurs de dimensions depuis les sélecteurs de la page.

Voir la règle : [4.02-04 Ajoutez des sélecteurs de page plutôt que des sélecteurs de module](#)

4.02-06

Hiérarchies à nombreux niveaux

Créez un tableau de bord distinct avec des modules de sélection afin de garder les principaux tableaux de bord le plus épuré possible. Cela permet également une navigation plus efficace au sein des différents niveaux de l'arborescence hiérarchique.

4.02-07

Évitez les boutons de navigation

Utilisez le panneau *Contents*, le panneau de navigation *UX*, votre page d'accueil ou des liens sur les cartes - mais n'encombrez pas les tableaux de bord avec de multiples boutons de navigation.

4.01-07a Processus itératifs

Lorsque l'utilisateur a besoin de basculer périodiquement entre les tableaux de bord, il est acceptable d'avoir des boutons de navigation.

4.02-08

Évitez les boutons 'Actualiser'

Expliquez aux utilisateurs comment utiliser l'option *Refresh* de la barre d'outils plutôt que de créer des boutons d'action «actualiser» dans vos *Dashboards*.

4.02-09

Publiez les graphiques une première fois avec des filtres complets

Si le graphique a un axe filtré, définissez le filtre de sorte que tous les éléments soient affichés, puis publiez le graphique. Ensuite, réglez les critères de filtre à nouveau comme vous le souhaitez ; le graphique appliquera alors le filtre.

4.02-10

Désactivez le nom du module

Utilisez des zones de texte dans le tableau de bord pour décrire la grille, plutôt que d'afficher le nom technique du module sous-jacent.

Voir la règle : [2.01-01 Convention de nommage](#)

4.01-10a Gain de place

Si l'espace à l'écran est rare, le texte d'en-tête peut prendre plus d'espace que souhaité, mais tenez compte de l'expérience globale de l'utilisateur et n'encombrez alors pas le tableau de bord.

4.02-11

Publiez des actions *Process* plutôt que des actions *Import/Export*

Les actions *Import* et *Export* doivent être republiées en cas de modification ou de remplacement, tandis que les actions *Process* sont toujours cohérentes avec les actions paramétrées en leur sein.

Voir la règle : [5.01-02 Actions destinées aux utilisateurs dans Process](#)

4.02-12

Évitez les boutons d'action *Delete* pour les utilisateurs finaux

Les utilisateurs finaux peuvent potentiellement supprimer des données historiques. Veillez donc à ce que la maintenance des éléments de liste soit gérée de manière séparée, dans le cadre du processus d'administration. Par exemple, un *dashboard* dédié à ce type de maintenance pourra être publié à l'attention uniquement des administrateurs.

Voir la règle : [4.02-11 Publiez des actions Process plutôt que des actions Import/Export](#)

4.02-13

Stocker les URL pour images dans un module distinct

Conservez toutes les URL d'images utilisées pour une liste dans un module système distinct. Prêtez attention à la dimensionnalité requise.

Voir la règle : [2.01-04 Appliquez la méthodologie DISCO pour le paramétrage des modules](#)

4.02-14

Activez les tableaux de bord et pages personnels

Laissez les utilisateurs personnaliser leurs vues sur les données avec *Personal Dashboards* et *My Pages*.

Voir la règle : [6.04-03 Master Dashboards](#)

Best practice : [Personal Dashboards - Tips and Tricks](#)

4.02-15

Limitez les modules avec de nombreux *Line Items* formatés en *List*

Un trop grand nombre de *Line Items* au format *List* dans un même tableau de bord peut nuire aux performances. Essayez de limiter le nombre de *Line Items* affichés dans un seul module ; le cas échéant répartissez-les sur plusieurs modules.

4.03 Filtres

4.03-01

Utilisez des filtres efficaces

Pour des performances optimales, gérez vos filtres sur la base de conditions sur valeurs booléennes pour chaque axe.

Voir la règle : [4.03-06 Évitez de filtrer sur des dimensions imbriquées](#)

Best practice : [Filter Best Practice](#)

4.03-02

Gérez les filtres dans des modules dédiés

Efforcez-vous de conserver les filtres dans des modules système distincts. Ils peuvent ensuite être réutilisés dans différents modules.

Voir la règle : [2.01-15 Gérez vos filtres dans des modules systèmes](#)

4.03-03

Utilisez des filtres de temps sur tous les tableaux de bord

N'utilisez pas la fonction *show/hide* pour la dimension *Time* ; ces fonctions sont statiques. L'utilisation d'un module *Time Settings* permet de s'assurer que le filtrage sera dynamiquement mis à jour lors de changements dans le *Model Calendar*.

Voir la règle : [2.01-07 Modules système Time Settings](#)

4.03-04

Filtrer les sélections par ordre de taille

Si vous utilisez plusieurs conditions dans un filtre, incluez les conditions par ordre croissant de la taille des listes - avec les conditions numériques en dernier.

4.03-05

Utilisez des filtres *User* pour fournir des rapports dynamiques

Créer des modules de filtre dimensionnés par la liste des utilisateurs afin de gérer un filtrage dynamique d'accès aux données.

Voir la règle : [1.03-02 Les performances et la taille peuvent dépendre de la liste des utilisateurs](#)

Best practice : [Increase end user adoption with smart filters](#)

4.03-06

Évitez de filtrer sur des dimensions imbriquées

Évitez autant que possible de filtrer sur des dimensions imbriquées (*Nested Dimensions*). Essayez d'autres pivots, et si aucun autre pivot n'est possible et que vous devez filtrer en l'état, assurez-vous au moins d'optimiser les filtres.

Voir la règle : [4.02-01 Utilisez des filtres efficaces](#)

Best practice : [Filter Best Practice](#)

4.03-07

Utilisez les notes afin de documenter les *Line Items* utilisés dans des filtres

Utilisez les zones Notes de la vue *Blueprint* pour signaler systématiquement les *Line Items* utilisés comme critère de filtres. Ceci vous aidera notablement lors d'audits et recherches sur les *saved views* et autres filtres dans les pages et *dashboards*.

5 Data Integration – Intégration des données

Cette section détaille les bonnes pratiques pour les **Imports**, **Exports** et **Data Hubs**.

5.01 Actions

5.01-01

Convention de nommage

Utilisez des préfixes numériques pour indiquer l'ordre d'exécution dans le processus (par exemple : **1.1** Import Produits, **1.2** Import Détails sur Produits). Il n'est pas nécessaire d'inclure le nom du module technique dans le nom.

Voir la règle : [5.02-01 Convention de nommage](#)

Anapedia : [Naming convention](#)

5.01-02

Actions destinées aux utilisateurs dans *Process*

Les actions *Import* et *Export* doivent être republiées en cas de modification ou de remplacement, tandis que les actions *Process* sont toujours cohérentes avec les actions paramétrées en leur sein, ce qui facilite leur gestion.

Voir la règle : [4.02-11 Publiez des actions Process plutôt que des actions Import/Export](#)

*5.01-02a Actions sur les
Numbered list*

Certaines actions sur listes numérotées (Numbered Lists) telles que Create, Assign, Copy Branch et Delete Branch ne peuvent pas être ajoutées à un Process.

5.01-03

Réduisez les actions utilisateurs (en général)

Passez en revue de manière critique la nécessité de chaque action pilotée par un utilisateur, en tenant compte notamment de l'impact d'accès simultanés (*user concurrency*). Essayez plutôt d'utiliser des formules. Cela peut nécessiter des modules supplémentaires, mais l'expérience utilisateur devrait en bénéficier.

5.01-04

Supprimez les actions à usage unique

Supprimez les imports ponctuels qui n'ont servi qu'une seule fois ; supprimez l'action ainsi que la source de données.

5.02 *Process*

5.02-01

Convention de nommages

Utilisez des noms « conviviaux » pour les *Process* destinés aux utilisateurs finaux. Pour les *data hubs* et les *process* d'administration, utilisez des préfixes numériques pour indiquer l'ordre des *Process*.

Voir la règle : [5.01-01 Convention de nommage](#)

Anapedia : [Naming convention](#)

5.02-02

Réduisez le nombre d'actions dans un *Process*

Chaque action d'un *Process* déclenche un recalcul du modèle, essayez donc de réduire autant que possible le nombre d'actions.

5.03 Modèles sources

5.03-01

Supprimez les sources inutiles

Dans le cas d'imports à usage unique ou lorsque les modèles source ne sont plus nécessaires, supprimez la référence au modèle dans la liste 'Import data sources'.

5.04 Imports

5.04-01

A partir du système source, créez un référencement unique pour les attributs

Dans l'idéal, il s'agit d'un fichier distinct et unique ; c'est préférable à l'importation d'attributs depuis le même fichier que celui utilisé pour les données transactionnelles.

Voir la règle : [5.04-02 Créez des fichiers séparés pour les attributs et les données](#)

Best practice : [Data Hubs Purpose and Peak Performance](#)

5.04-01a Code > 60 caractères

Si le code est > 60 caractères, vous devrez utiliser une combinaison de propriétés.

5.04-02

Créez des fichiers séparés pour les attributs et les données

Le fichier de données devrait avoir identifiants et valeurs des dimensions correspondantes. Les données non dimensionnelles telles que les attributs devraient se trouver dans un fichier distinct.

Voir la règle : [5.04-01 A partir du système source, créez un référencement unique pour les attributs](#)

Best practice : [Data Hubs Purpose and Peak Performance](#)

5.04-03

Données et périodes ne doivent pas faire partie de votre clef

Réfléchissez à ce qui rend réellement l'enregistrement unique ; incluez uniquement des attributs dans la clef, et non des valeurs de données ou des dates.

Voir la règle : [1.05-11 N'incorporez pas de valeurs numériques ou de dates dans le code d'une liste](#)

Best practice : [Data Hubs Purpose and Peak Performance](#)

5.04-04

N'incluez que les champs nécessaires

Dans vos paramètres d'import, utilisez le bouton *Ignore* pour exclure de l'import les champs indésirables.

5.04-05

Pré-agrégez dans la source

Les agrégations doivent être effectuées dans le système source, partout où c'est possible. Ceci est susceptible de réduire la taille du fichier d'import, et donc contribuer à une importation plus rapide.

5.04-06

Importez au bon niveau de granularité

Importez les données au seul niveau de granularité requis. Par exemple, il est inutile d'importer des données détaillées à un niveau hebdomadaire si la planification est faite par mois.

5.04-07

Importez avec les bons formats

Le format des *Lines Items* cibles de l'import doit refléter les données, notamment les formats *List*, nombres et dates. Utilisez le format texte uniquement lorsqu'il s'agit d'un véritable champ textuel.

5.04-08**N'utilisez jamais une liste comme source d'importation**

La définition d'un import autre que depuis un fichier devrait toujours être effectuée à partir d'une vue de module. Cela permet de filtrer et d'inclure uniquement les éléments requis pour chaque importation.

[5.04-09 Utilisez toujours des vues enregistrées comme sources d'importation](#)

5.04-09**Utilisez toujours des vues enregistrées comme sources d'importation**

Les modules et les vues enregistrées doivent être utilisés comme source pour les imports autres que depuis un fichier.

Best practice : [Filter Best Practice](#)

5.04-10**Standardisez le format date**

Dans la mesure du possible, utilisez un format unique de date générique (par exemple YYYYMMDD) pour simplifier les importations et ne plus avoir à gérer les discordances et les modifications de dates.

Best practice : [Untangling Anaplan Time Mapping in Imports](#)

5.04-11**Pensez à utiliser les fichiers zip pour accélérer vos imports**

Anaplan accepte les fichiers zip en import ; privilégier ce format pour les imports volumineux améliorera les temps d'importation.

5.05 Exports

5.05-01

Convention de nommage

Soyez simple ; utilisez des termes tels que « Exporter », « Construire », etc. Il n'est pas nécessaire d'inclure le nom du module dans le nom de la vue.

Anapedia : [Naming convention](#)

5.05-02

Incluez uniquement ce qui est nécessaire

N'incluez que les *Line Items* requis pour l'exportation. Créez plusieurs vues si le module est utilisé pour différents fichiers que vous importerez par la suite dans vos systèmes aval. Le nombre de colonnes dans la vue impacte la vitesse d'exportation : moins il y a de colonnes et plus l'exportation est rapide.

5.05-03

Dissociez les vues destinées aux imports de *Lists* et de modules

Créer deux vues dans le même module source. Une pour l'importation de la liste (avec *name*, *code* et *parent*) et une autre pour les attributs du module système associé.

Voir la règle : [5.05-02 Incluez uniquement ce qui est nécessaire](#)

5.05-04

Optimisez les filtres dans les vues

N'utilisez qu'un seul critère de filtre... Si vous avez besoin de critères multiples, combinez-les en un seul *Line Item*.

Voir la règle : [4.02-01 Utilisez des filtres efficaces](#)

5.06 *Import Data Sources* – Sources d'import

5.06-01

Convention de nommage

Renommez dès que possible les sources d'importation automatiquement intégrées lors du premier import. Dans le cas d'une vue enregistrée, indiquez le nom du module (par exemple, 'module.vue enregistrée'). Pour les importations faites depuis un autre modèle, inclure une mention abrégée au modèle (par exemple 'Model_module.vue enregistrée').

Anapedia : [Naming convention](#)

5.06-02

Supprimez les sources indésirables

Dans le cas des imports à usage unique ou lorsque les sources ne sont plus nécessaires, supprimez les sources dans la liste '*Import data sources*'.

5.07 Data Hub

5.07-01

Aucune hiérarchie

Il ne devrait y avoir aucune hiérarchie de listes composites dans le Hub ; elles ne sont pas nécessaires dans ce contexte. Vous pouvez les créer le temps de tester les actions, mais supprimez-les dès que les tests sont finis.

Best practice : [Data Hubs Purpose and Peak Performance](#)

5.07-01a Validation nécessaire

Si les données doivent être consolidées pour permettre la vérification par rapport aux systèmes source, vous pourriez garder les hiérarchies. Mais même dans ce cas, les modules 'à plat' avec attributs permettent de visualiser certaines totalisations, ce qui pourrait suffire aux validations.

5.07-02

Aucun module d'analyse

Efforcez-vous de garder les modules analytiques hors du *Hub*, au sein des modèles d'exploitation.

5.07-03

Listes 'plates' pour l'exportation des données

Utilisez des structures de liste à plat pour créer des modules et vues pour les exportations aval.

5.07-04

Exportez aux bons formats

Prenez soin de formater les données aux formats souhaités et à la bonne granularité pour les systèmes en aval.

Voir la règle : [5.04-06 Importez au bon niveau de granularité](#)

5.07-05

Utilisez les modules système

Utilisez les modules système pour filtrer les données (période en cours, année fiscale en cours, etc.).

Voir la règle : [2.01-15 Gérez vos filtres dans des modules systèmes](#)

Best practice : [Data Hubs Purpose and Peak Performance](#)

Best practice : [Filter Best Practice](#)

5.07-06**Les données de référence devraient être créées par le système source**

Évitez autant que possible de créer les données de référence (*master data*) dans le *Hub*. Elles devraient provenir du ou des systèmes source(s).

5.07-07**Conservez l'analyse transactionnelle hors des modèles de planification**

Utilisez le *Data Hub* (ou un autre modèle de reporting) pour héberger les données transactionnelles détaillées, hors des principaux modèles de planification. De grandes quantités de données historiques peuvent augmenter la taille du modèle de planification et entraîner une réduction des performances.

5.07-08**Pré-agrégez pour vos exports**

Il est plus efficace d'agréger les données au sein du *Data Hub* préalablement à leur exportation, plutôt que chercher à les agréger au moment de l'import dans le modèle cible.

5.07-09**Évitez d'utiliser un *Top Level* pour de grandes listes de transactions**

Ne créez pas de liste de données transactionnelles avec un *Top Level*. Ceci déclencherait des calculs de sommation complets - même si un unique élément de liste est ajouté.

Voir la règle : [5.07-10 Ajouter des sous-totaux intermédiaires aux listes de transactions](#)

Best practice : [Top Level Item and Parent Hierarchy](#)

5.07-09a Totalisation à valider

Si une vérification du total de toutes les transactions est nécessaire, vous pourriez activer un Top Level. Mais une fois de plus, il est préférable d'utiliser les simples totalisations des modules d'attributs.

5.07-10

Ajouter des sous-totaux intermédiaires aux listes de transactions

Si vous avez besoin de totaux à des fins de validation, créez des sous-totaux intermédiaires dans la liste de transactions. Cela réduit considérablement la charge de calcul.

Voir la règle : [5.07-09 Évitez d'utiliser un Top Level pour de grandes listes de transactions](#)

5.07-11

Maintenez votre *Data Hub* dans un *Workspace* distinct

Réservez un workspace dédié au *Data Hub*. Ceci permettra au modèle de « grossir » en limitant la possibilité que la taille interfère avec les interfaces amont ou aval. Cela permet également de garantir la séparation des tâches (les utilisateurs responsables de l'alimentation des données dans le *Hub* n'ayant pas accès aux modèles de production).

6 Application Lifecycle Management

Cette section détaille les bonnes pratiques de l'Application Lifecycle Management (ALM)

6.01 Revision Tags - Balises de révision

6.01-01

Créez une norme de dénomination

Soyez cohérent dans la façon dont vous nommez vos révisions. Il est conseillé d'utiliser une nomenclature majeure/mineure (par exemple R1.01, R1.02, R2, etc.).

Faites attention : n'oubliez pas que le libellé est permanent et ne peut pas être modifié après la création de la balise !

6.01-02

Créez une balise de révision au moins une fois par jour

Pendant les périodes de développement intense, assurez-vous que des balises de révision sont créées régulièrement et surtout testées périodiquement en synchronisation avec un modèle « shell » (« coquille » de test).

Voir la règle : [6.01-05 Modèle de test « shell »](#)

Best practice : [Revision Tag Best Practices](#)

6.01-03

Documentation des révisions

Si vous avez besoin d'une description plus complète des révisions, créez une liste simple puis renseignez les détails de révision dans un module associé contenant des champs tels que *Date d'approbation User Story*, *Développeur*, *Date de test*, *Date de signature*, etc...

Best practice : [Revision Tag Best Practices](#)

6.01-04**Synchronisez régulièrement la Production**

Réduisez le risque d'erreurs de synchronisation en effectuant une synchronisation régulière de manière à maintenir à niveau la Production et le Développement.

Voir la règle : [6.01-02 Créez une balise de révision au moins une fois par jour](#)

Best practice : [ALM Explained—Part 1: Compatibility](#)

6.01-05**Modèle de test « shell »**

Configurer un modèle « coquille » (à l'aide de l'option '*create from revision*') qui ne contient aucun membre de la liste de production – et sur lequel vous testerez périodiquement révisions et synchronisations.

Voir la règle : [6.01-02 Créez une balise de révision au moins une fois par jour](#)

Best practice : [ALM Explained—Part 2: Testing](#)

6.02 Production Lists – Listes de production

6.02-01**Déterminez quelles listes sont '*production*' avant la première synchronisation**

Ne définissez pas toutes les listes comme '*production*' d'emblée : revenir ultérieurement sur ce paramètre pour rétablir '*structural*' puis synchroniser aura pour effet de supprimer les données de la liste de production existante, même si les membres sont exactement identiques.

Best practice : [Prepare Applications for ALM](#)

Best practice : [ALM Explained—Part 3: Avoiding Data Loss](#)

6.02-02

Protection de formule - codage en dur

Ne codez surtout pas 'en dur' une référence à un élément d'une liste de production car ce membre peut être supprimé par un utilisateur final.

Best practice : [Formula Protection](#)

6.02-03

Analysez de manière critique la nécessité pour chaque liste d'être 'production'

Seules les listes qui doivent être modifiées dans le cadre du processus métier ou alimentées par imports doivent être paramétrées comme listes de production.

Best practice : [ALM Explained—Part 3: Avoiding Data Loss](#)

6.02-04

Faites mieux que tenter votre chance

Passez en revue les informations de la colonne '*referenced by*' et contrôlez l'existence de références en dur dans les formules avant de lancer la synchronisation – n'attendez pas de vérifier en Production après synchronisation pour, le cas échéant, restaurer le modèle en cas de problème.

Voir la règle : [6.02-02 Protection de formule - codage en dur](#)

Best practice : [ALM Explained—Part 3: Avoiding Data Loss](#)

6.03 Architecture

6.03-01

Ne revenez jamais sur le statut *deployed* du modèle

Une fois que l'ALM a été initiée et que le mode *Deployed* est activé, le modèle de Production ne doit **JAMAIS** être sorti du mode déployé.

Best practice : [ALM Explained—Part 1: Compatibility](#)

6.04-01a Création du modèle de développement

Lors de la copie d'un modèle de Production pour créer l'environnement ALM initial, ou lors de la re-cr ation du mod le DEV dans le cadre d'une « r initialisation ».

6.04-01b Il n'y a pas d'autres exceptions !

 a n'en vaut pas la peine. ALM assure un contr le, mais il existe des r gles, et le mode Deployed en est la clef de vo te.

6.03-02

Utilisez *Deployed* en mode DEV pour plus de s curit 

Il est tout   fait possible de param trer le mode d ploy  pour les mod les de d veloppement, et cela peut m me  viter que des changements structurels involontaires ne soient effectu s en dehors des proc dures de d veloppement pr vues.

6.03-03

Garder les mod les de test en mode *Deployed*

Les mod les de test doivent  tre consid r s comme des mod les de production. Cela donne une v ritable repr sentation des tests et emp che  galement les synchronisations malencontreuses depuis TEST vers PROD.

Best practice : [ALM Explained—Part 2: Testing](#)

6.04 Deployed Mode

6.04-01

Retour vers le futur

Il s'agit d'une technique qui rétablit l'état de révision du modèle, vous permet d'effectuer une correction, de synchroniser vers la Production, puis de rétablir le Développement en son état d'origine.

Best practice : [Saving incomplete changes during development](#)

6.04-02

Switchover

Après une révision, créez plusieurs balises de révision avec des dates de *switchover* différentes avant de commencer le nouveau développement.

Best practice : [Managing Frequent Structural Changes During Development](#)

6.04-03

Master Dashboards

Les *Master Dashboards* suppriment tous les *Personal Dashboards*. Les techniques permettant de minimiser l'interruption de service pour les utilisateurs incluent la création d'une copie et l'utilisation d'un processus de migration pour déplacer les utilisateurs vers les nouveaux *dashboards*.

Voir la règle : [4.02-14 Activez les tableaux de bord et pages personnels](#)

Best practice : [Personal Dashboards - Tips and Tricks](#)

6.05 Gestion des changements en cours de développement

6.05-01

Faites de DEV la source de tout

La configuration DEV> TEST et DEV>PROD est plus flexible ; elle permet de créer et de supprimer plusieurs modèles de test sans compromettre la PROD. Enfin, cette configuration permet d'organiser la séparation des tâches (les testeurs ne développent pas, etc..).

Best practice : [ALM Explained—Part 1: Compatibility](#)

Best practice : [ALM Explained—Part 2: Testing](#)

6.05-02

Modèles archivés

Les modèles compatibles (DEV, TEST ou PROD) peuvent être archivés sans rompre le lien entre eux.

Voir la règle : [6.05-03 Modes du modèle](#)

Best practice : [ALM Explained—Part 2: Testing](#)

6.05-03

Modes du modèle

Lors de la restauration de modèles archivés, restaurez en mode *Deployed* s'il s'agit d'un modèle de Production ou de Test.

Voir la règle : [6.05-02 Modèles archivés](#)

6.05-04

Small is beautiful : Gardez le modèle de Développement petit

Réduisez autant que possible la taille du modèle de Développement ; pour ce faire, essayez d'utiliser un simple échantillon d'éléments des listes de production. Utilisez la fonctionnalité "*Create from revision*" pour constituer le modèle.

Best practice : [Prepare Applications for ALM](#)

6.05-05

Utilisez un *Data hub* pour alimenter le modèle de développement

Utilisez les vues enregistrées du *Data Hub* pour alimenter le modèle de développement. Cela permet d'importer des données et des structures et de contribuer au développement initial puis aux tests suivants de composants.

7 Extensions

Cette section détaille les bonnes pratiques pour les Add-ins MS Excel ou Powerpoint.

7.01 Excel

7.01-01

Convention de nommage

Ne changez pas le nom du module Anaplan car cela peut rendre la connexion inutilisable dans la feuille Excel.

Best practice : [Best Practices About Excel Add-In](#)

7.01-02

Utilisez des zones nommées

Il est conseillé d'utiliser les noms Excel dans vos formules - plutôt que des références de cellules.

Best practice : [Best Practices About Excel Add-In](#)

7.01-03

Contrôle des versions

Il est conseillé d'être à jour de la dernière version disponible – dès qu'elle est validée par les tests utilisateurs.

Best practice : [Governance for the Office Add-Ins](#)

7.01-04

Informations sur la version

Vous trouverez les informations relatives aux versions ici : [Version Information for Excel Add-in](#).

7.02 PowerPoint

7.02-01

Convention de nommage

Ne changez pas le nom du module Anaplan car cela peut rendre la connexion inutilisable dans la diapositive PowerPoint.

Best practice : [Best Practices About Microsoft Office for the Add-Ins](#)

7.02-02

Définissez des vues (*saved views*) dans les modules Anaplan pour optimiser la performance

Efforcez-vous de limiter les volumes de données interrogées par l'Add-In afin de garantir une bonne performance. Dans le cas où vous cherchez à restituer quelques données d'un module en contenant beaucoup, il peut être utile de créer un nouveau module dédié à cette interrogation pour l'add-in.

Best practice : [Best Practices and Tips & Tricks About PowerPoint Add-In](#)

7.02-03

Information sur la version

Vous trouverez les informations relatives aux versions ici : [Version Information for PowerPoint Add-in](#)



<https://community.anaplan.com/Planual>