Mettez en place les bonnes pratiques ITIL

Comprenez le référentiel de bonnes pratiques ITIL

Information Technology Infrstructure Library

ITIL, c’est un ensemble de **5 ouvrages** appelés aussi des **publications.**

Ces publications contiennent des **recommandations** qui sont des bonnes pratiques pour mieux **gérer votre service IT**.

Ces livres couvrent et détaillent tous les aspects de la gestion d’un service IT.

ITIL aborde :

* La mise en place de la **stratégie** de ce service (objectifs, définition du périmètre) ;
* La **conception** d’un service IT (disponibilité, capacité, sécurité, coûts…) ;
* Le **déploiement** d’un service (mise en oeuvre, test, déploiement) ;
* La **gestion opérationnelle** (incidents, problèmes, gestion des accès…) ;
* **L’amélioration en continu** du service (audit, analyse des forces et faiblesses).

Le terme Service IT ne signifie pas “département informatique d’une société”. Il désigne plutôt **un service fourni à l’aide de la technologie** : un accès à Internet, un service messagerie, de la téléphonie, un site web, une application web…

ITIL est aujourd’hui *LA référence* pour de nombreuses organisations et entreprises : secteur public ou privé, de l’entreprise du CAC 40 à la petite start-up en croissance.

**Comment ça marche ?**

La méthodologie éprouvée d'ITIL est liée à son **approche par les processus, fonctions et rôles** (ou responsabilités). Cela veut dire que :

1. Il y a au total **26 processus** qui vous expliquent comment faire. Il existe aussi bien un processus pour gérer les incidents qui arrivent au service desk (le support), qu’un processus pour gérer les coûts ou tester et valider un service avant de le déployer ;
2. Vous y trouverez le détail des **fonctions et rôles**, c'est-à-dire qui fait quoi ;
3. Il suffit enfin de mettre ensemble les pièces du puzzle en s’accordant et en définissant qui fait quoi et comment ;)

**Quels sont les bénéfices d’ITIL ?**

Adopter un référentiel tel qu'ITIL va vous permettre **d’améliorer la gestion de votre service IT.**

ITIL se concentre sur la valeur du service fourni : les **résultats** et non les outils.

Plus précisément, il peut permettre plusieurs choses :

* **Aligner les besoins des utilisateurs avec la technique.**
* **Améliorer la qualité du service fourni**. En introduisant un ensemble cohérent de processus, vous identifierez les **problèmes** et les **faiblesses** potentiels SI.
* **Réduire les coûts**. ITIL permet de s’assurer que vous pouvez délivrer un service dont vous pouvez **supporter le coût**.
* **Il est facilement transposable aux pratiques existantes d’une entreprise**. Une norme est un ensemble d'exigences à mettre en place afin d’atteindre un standard. Mais ITIL n’est pas une norme, **ces recommandations ne sont pas des exigences**.

**En résumé**

* ITIL® est une **bibliothèque** de 5 ouvrages recensant les bonnes pratiques dans la **gestion d’un service IT** ;
* Sa méthodologie s’articule autour de **processus, fonctions et rôles** en déterminant **qui fait quoi et comment** ;
* ITIL® permet d’**améliorer la qualité** d’un service IT, de **réduire les coûts** et d’**aligner les besoins des utilisateurs** avec la technique.

**Maitrisez la terminologie ITIL**

Processus

Il s’agit d’une suite d’activités qui s’exécutent de manière à **réaliser une tâche.**

Un processus transforme donc un ou **plusieurs éléments d’entrée en éléments de sortie** en suivant un objectif.

Un processus est mesurable :

* **Quantitativement**
* **Qualitativement**.

**Fonctions**

L’autre élément clé dans ce référentiel de bonnes pratiques est la notion de **fonctions**. Il s’agit des intervenants qui exécutent les activités d’un processus. C’est-à-dire :

* Les personnes ;
* Les équipes ;
* Les outils.

Dans une grande organisation, ces fonctions peuvent être éclatées en **départements**:

* Le **management** ;
* Les **développeurs** ;
* L'**exploitation** (admin système) ;
* Le **support niveau 1** ;
* Le **support niveau 2** ;
* Les **fournisseurs** externes ;
* L’outil de **ticketing** utilisé pour traiter les dossiers ;
* La **ligne téléphonique**.

**Rôles**

Un rôle est un ensemble de responsabilités, d’activités et d’autorités attribuées à une personne ou une équipe dans un contexte spécifique.

En IT, dans le processus de gestion des incidents, ce serait :

* Le **management** a pour rôle **coordonner le travail** des équipes ;
* Les **développeurs** ont pour rôle de **coder les modifications** qui corrigent l’incident ;
* L'**exploitation** a pour rôle de **superviser le SI et résoudre les incidents** liés à l’infrastructure ;
* Le **support niveau 1** a pour rôle d’effectuer une **première analyse** de l’incident ainsi que de **communiquer** avec les utilisateurs ;
* Le **support niveau 2** a pour rôle d’effectuer un **diagnostic technique** de l’incident et de trouver sa cause ;
* Les **fournisseurs externes** ont pour rôle de **rétablir des services externalisés** (accès Internet, data center) ;
* L'**outil de ticketing** a pour rôle de **recueillir l’ensemble des informations** relatives à l’incident ;
* La **ligne téléphonique** a pour rôle de **permettre les échanges** audio avec les utilisateurs ou entre les équipes.

Une **fonction** peut se voir attribuer plusieurs **rôles**. Le support de niveau 1 pourra effectuer un diagnostic de 1er niveau et résoudre l’incident s’il en a la capacité par exemple.

Services

Un service est un moyen de fournir de la valeur aux clients en facilitant les résultats que les clients veulent atteindre **sans en avoir la propriété des coûts et des risques**.

Dans l’IT, c’est pareil. En tant qu'administrateur système, **vous offrirez de nombreux services aux utilisateurs**, tels que la messagerie, le CRM, l'accès aux documents partagés. Les utilisateurs n’auront pas besoin de savoir que pour envoyer un mail, il faut mettre en place une infrastructure, des solutions de messagerie, des solutions réseau, du stockage…

**En tant que fournisseur, vous apportez de la valeur à un client via un service**. Le client utilise le service afin d’atteindre un résultat qui lui est propre. Cela, moyennant rémunération… ou pas.

Pour vous aider à bien comprendre, voici quelques exemples de ce qu’est un service IT dans ce cours :

* Le service IT fourni par un FAI est **l’accès à Internet** ;
* Votre hébergeur cloud vous fournit comme service IT **une infrastructure cloud** ;
* En tant qu’administrateur système cloud, vous fournissez comme services informatiques **un accès aux applications métier et aux mails**. Vous êtes le fournisseur, et vos clients sont les utilisateurs de ce service.

**Cycle de vie d’un service**

Dans ITIL, 5 phases/étapes constituent le cycle de vie d’un service, chacune étant traitée dans une publication. Vous vous en souvenez probablement, le référentiel ITIL est constitué de 5 livres !

Il s’agit de :

1. La **stratégie** des services (Service Strategy) ;
2. La **conception** des services (Service Design) ;
3. La **transition** des services (Service Transition) ;
4. **L’exploitation** des services (Service Operation) ;
5. **L’amélioration continue** des services (Continual Service Improvement).

**En résumé**

* Les concepts fondamentaux dans ITIL sont : **processus, fonction, rôles, service et cycle de vie d’un service** ;
* Un processus est une **suite d’activités coordonnées** qui décrit comment réaliser une tâche pour atteindre un objectif ;
* Une **fonction** est une entité organisationnelle, un groupe de personnes qui **exécutent les activités** d’un processus ;
* Un **rôle** est un ensemble de **responsabilités** allouées à **une fonction** ;
* Une **fonction** peut se voir attribuer **plusieurs rôles** ;
* Un **fournisseur** produit de la valeur pour un client via un service ;
* Un **service** est un **moyen de fournir de la valeur aux clients**, en facilitant les résultats que les clients veulent atteindre sans en avoir la propriété des coûts et des risques ;
* Le **cycle de vie d’un service** définit les différentes phases par lesquelles passe un service : de sa naissance à son exploitation quotidienne (voire sa suppression).

**Organisez le SI avec ITIL**

**Pourquoi mettre en place le cycle de vie d'un service ?**

Car il ne sert a rien de chercher à maintenir un service qui à été **mal pensé et élaboré**. Il faut se poser des questions en amont.

**Le cycle de vie d’un service**

ITIL propose 5 publications, dont **chacune traite d’une phase** du cycle de vie d’un service. Ces 5 phases fournissent l’orientation pour atteindre un standard de qualité.

**Définissez la stratégie du service – Service Strategy – SS**

Les questions qu’on se pose à ce niveau sont : Quels **besoins** le service couvrira-t-il ? Comment fournir un service qui se **différencie de l’existant** ? Comment le service pourra-t-il **créer de la valeur** pour les utilisateurs ?

**Concevez le service –Service Design – SD**

Les questions qu’on se pose à ce niveau sont : **Combien de serveurs** me faudra-t-il ? **Quelles solutions** utiliser ? Devrais-je concevoir ou utiliser un **fournisseur externe** ? Quel niveau de **disponibilité, de capacité du serveur** dois-je prévoir ?

À cette étape, vous avez réalisé ce qu’on appelle le **SDP**—**Service Design Package**, une documentation qui servira à concevoir et mettre en oeuvre votre service dans l’environnement de production.

**Mettez en œuvre le service – Service Transition – ST**

Maintenant que toutes les questions pratiques ont été couvertes, vous allez **concevoir, tester et déployer le service** dans l’environnement de production.

Au terme de cette étape, **le service sera désormais accessible aux utilisateurs**.

**Exploitez le service – Service Operation – SO**

Dans cette quatrième étape, vous coordonnez les activités pour délivrer le service convenu aux utilisateurs.

Les 5 processus qui contiennent cette phase permettent de :

* Traiter les **demandes des utilisateurs** de ce service ;
* **Gérer l'accès** au service ;
* **Surveiller les évènements et les alertes** du service ;
* **Gérer les incidents** liés qui perturbent le service, en évitant des récurrences.

C’est une étape très importante, car c’est ici que les utilisateurs perçoivent la **qualité du service**.

**Améliorez continuellement votre service—Continual Service Improvement—CSI**

Cette étape contient **1 processus chargé de mesurer tous les autres processus** de toutes les autres phases, et de documenter ces résultats.

Les résultats de vos mesures vous serviront à évaluer la maturité des processus de tous les cycles. Souvenez-vous, l’une des caractéristiques d’un processus, c’est qu'il est mesurable.

**Définissez qui fait quoi — Fonctions et Rôles**

**ITIL propose 4 fonctions principales** qui pourront supporter les rôles nécessaires à l’exécution des processus.

Il s’agit entre autres des fonctions suivantes :

* Le ***Service Desk*** ou **centre de service**. C’est le point focal des utilisateurs du service. Il **recueille tous les besoins des utilisateurs** du service (incidents, demandes diverses) ;
* L’**exploitation** ou la **gestion des opérations**. C’est ici que le service est exploité (sauvegarde, supervision, surveillance, gestion des moyens généraux IT) ;
* La **gestion technique** ;
* La **gestion applicative**.

**En résumé**

ITIL recommande de **structurer la gestion d’un service IT** avec ce qu’il appelle **l’approche par le cycle de vie d’un service**.

* Le cycle de vie d’un service est le cadre organisationnel, il est constitué de 5 étapes ;
* Chaque publication ITIL traite d’une étape du cycle de vie d’un service ;
* Chacun de ses 5 livres décrit en détail les processus (26 processus au total) qui répondent à la problématique en objet de l’étape du cycle de vie ;
* Un **processus** est un **ensemble d’activités** à exécuter pour atteindre un objectif ;
* Certains processus couvrent **plusieurs phases** du cycle de vie des services ;
* Enfin, ITIL décrit les 4 fonctions principales qui supporteront les rôles nécessaires à l’exécution des activités contenues dans ces processus.

Vous avez désormais une vision d'ensemble de l'utilité d'ITIL, comment il est structuré, et à quoi il sert. Dans la prochaine partie, je vous propose d’apprendre à garder le contrôle de votre SI en établissant une procédure opérationnelle de gestion des changements.

**Maîtrisez le vocabulaire de gestion des changements**

### Les dangers d'un changement non contrôlé

La conséquence naturelle d’un changement non contrôlé est un incident de production générant une **rupture partielle du service**, voire une **indisponibilité totale du SI**. Une procédure de gestion des changements va vous permettre de garder le contrôle sur votre SI en autorisant, priorisant, planifiant les actions nécessaires

### Procédure

### Changement

Non seulement un déploiement est considéré comme un changement, mais la modification et la mise hors service sont aussi des changements.

Voici quelques exemples de changements :

* Une **mise à jour** applicative ;
* Le **remplacement d’une carte réseau** ;
* L’**installation / désinstallation** d’une application ou d'un service ;
* L’**externalisation** d’un service vers un fournisseur externe ;
* Une **modification** de l’infrastructure ou d'une procédure.

En entreprise, les changements surviennent pour une variété de raisons, ce qui permet de définir **3 types de changement**.

#### Changement standard

#### Changement urgent

#### Changement normal

Tout changement doit être introduit par une une **RFC**.

### La demande de changement—RFC

**Le point de départ de tout changement, c’est la demande de changement**, en anglais *Request For Change* (**RFC**). C’est un élément très important. Il s’agit d’une demande formelle de changement d’un service, incluant les détails du changement proposé.

Quand le changement est majeur, on utilise une **proposition de changement**.

### La proposition de changement

C’est une **RFC détaillée** concernant un changement important avec des implications significatives. Elle est utilisée pour les **changements majeurs** qui impliquent **des coûts et des risques importants**.

La proposition de changement est complète. Contrairement à une RFC, elle inclut un business case complet (enjeux, alternatives, budget…) ainsi qu’un calendrier pour la réalisation et la mise à jour du changement.

Avant d’être exécutées, les RFCs et les propositions de changements sont soumises à l’approbation d’un comité consultatif. Il s’agit du CAB.

### Le comité consultatif—CAB

**CAB** est l’acronyme de *Change Advisory Board* ; en français, le **comité consultatif de changement**. C’est un groupe de personnes autorisant un changement après avoir passé en revue la RFC.

Le CAB donne le *GO* ou le *NO-GO* à tous les changements. Il valide aussi le **planning et le calendrier de changement**.

### Le comité consultatif d’urgence—eCAB

Lorsque votre besoin de changement est **urgent**, et que vous manquez de temps, il est alors nécessaire d’identifier un **groupe plus restreint de personnes**. Ces personnes auront l’autorité pour **prendre les décisions urgentes**.

### Le RACI

C’est une matrice, un tableau qui permet d’établir les rôles et responsabilités. Les rôles sont définis par chacune des lettres :

* **Réalisateur** : ce sont les personnes qui **exécutent l’activité** de la procédure ;
* **Approbateur** : c’est **l’autorité**, la personne qui **rend des comptes** sur l’activité. Il ne doit y avoir **qu’une seule personne** pour une activité donnée ;
* **Consulté** : les personnes dont les **conseils et opinions sont utiles** ;
* **Informé** : les personnes que vous devez **tenir informées** du déroulement des activités.

### Elément de configuration—CI

Un **élément de configuration**, ou *Configuration Item* (**CI**) est un **composant de votre SI**, susceptible de subir un changement. Cela inclut **tout ou partie d’un service IT**, du hardware, du software, ou encore des locaux (un data center), des personnes (le service desk) et même la documentation.

Il est important de **connaître l’ensemble des éléments de configuration de son SI** et d’en tenir un **inventaire** précis des attributs et relations entre CIs. Cela va vous permettre de savoir que le serveur X est mutualisé avec telle ou telle application, et qu’une action sur ce serveur affectera un nombre précis d’éléments.

### Base de données des éléments de configuration—CMDB

La **CMDB**, pour *Configuration Management Database*, ou **base de données des éléments de configuration**, contient les informations sur les CI. Vous y placerez tous les détails et attributs des CI de votre SI : inventaire du matériel, des logiciels, applications. Elle devra contenir aussi le détail des relations entre tous les CI de votre SI.

**En résumé**

**Dans ce chapitre**, je vous ai présenté le **vocabulaire normalisé** de la gestion des changements. Vous devrez retenir que :

* Un **changement** se définit par l’ajout, la modification ou la suppression d’un service ou d’un composant de ce service ;
* Il existe **3 types de changement** : changement **normal**, changement **standard** et changement **urgent**. À la différence des changements normaux, les changements standard et urgents suivent une **procédure spécifique** ;
* Une **RFC** est une demande de changement, initiant tout changement ;
* Quand le changement est **majeur**, on utilise une **proposition de changement**, qui est une RFC incluant un business case complet ;
* Le **CAB** un **comité consultatif** autorisant un changement après avoir passé en revue la RFC ;
* Quand le besoin de changement est **urgent**, et que l’on manque de temps, on fait appel à l’**eCAB** ;
* Le **RACI** définit qui fait quoi. Il permet d’établir les rôles et responsabilités pour chacune des activités d’un processus ;
* Un **CI**, Configuration Item, est un **élément de configuration** de votre SI (software, hardware, documentation) ;
* La **CMDB**, quant à elle, est le registre qui inventorie tous les CI de votre Si, ainsi que leurs relations.

## Organisez les changements dans votre SI

### Les demandes de changements sont diverses

Le point de départ de tout changement est la **RFC**. Pour des raisons diverses, **tout le monde peut demander un changement**. Il peut s’agir d’une demande concernant :

* L’**installation d’un serveur** d’impression pour gérer plus rapidement les impressions ;
* Une **mise à jour de sécurité** pour pallier une faille dans l’OS ;
* Une **migration vers une infrastructure cloud** qui pourrait permettre de faire des économies à long terme ;

Toutes les demandes ne se valent pas.Certaines sont urgentes, d’autres un peu moins, mais présentent un risque réel si elles ne sont pas effectuées dans un certain délai.

La règle des **7R** va vous aider à y voir un peu plus clair.

### Découvrez la règle des 7R

La règle des 7R vous aidera à vous poser les questions essentielles lors de l’analyse d’une RFC. Vous pourrez ainsi établir le périmètre de ce changement :

* **Requis** : qui a requis ce changement ?
* **Raison** : quelle est la raison de ce changement ?
* **Retour** : quel est le retour attendu de ce changement ? Quels sont les bénéfices attendus ?
* **Risque** : quels sont les risques encourus à la réalisation ou la non-réalisation de ce changement ?
* **Ressources** : quelles sont les ressources nécessaires pour réaliser ce changement ? De quoi aurons-nous besoin ?
* **Responsable** : qui est responsable de la réalisation de ce changement ?
* **Relation** : quelle relation avec d’autres changements ? Existe-t-il des dépendances ?

Une fois le **périmètre du changement** établi, il est temps de catégoriser la demande de changement et ainsi faire apparaître les demandes devant être traitées en priorité.

### Catégorisez vos changements

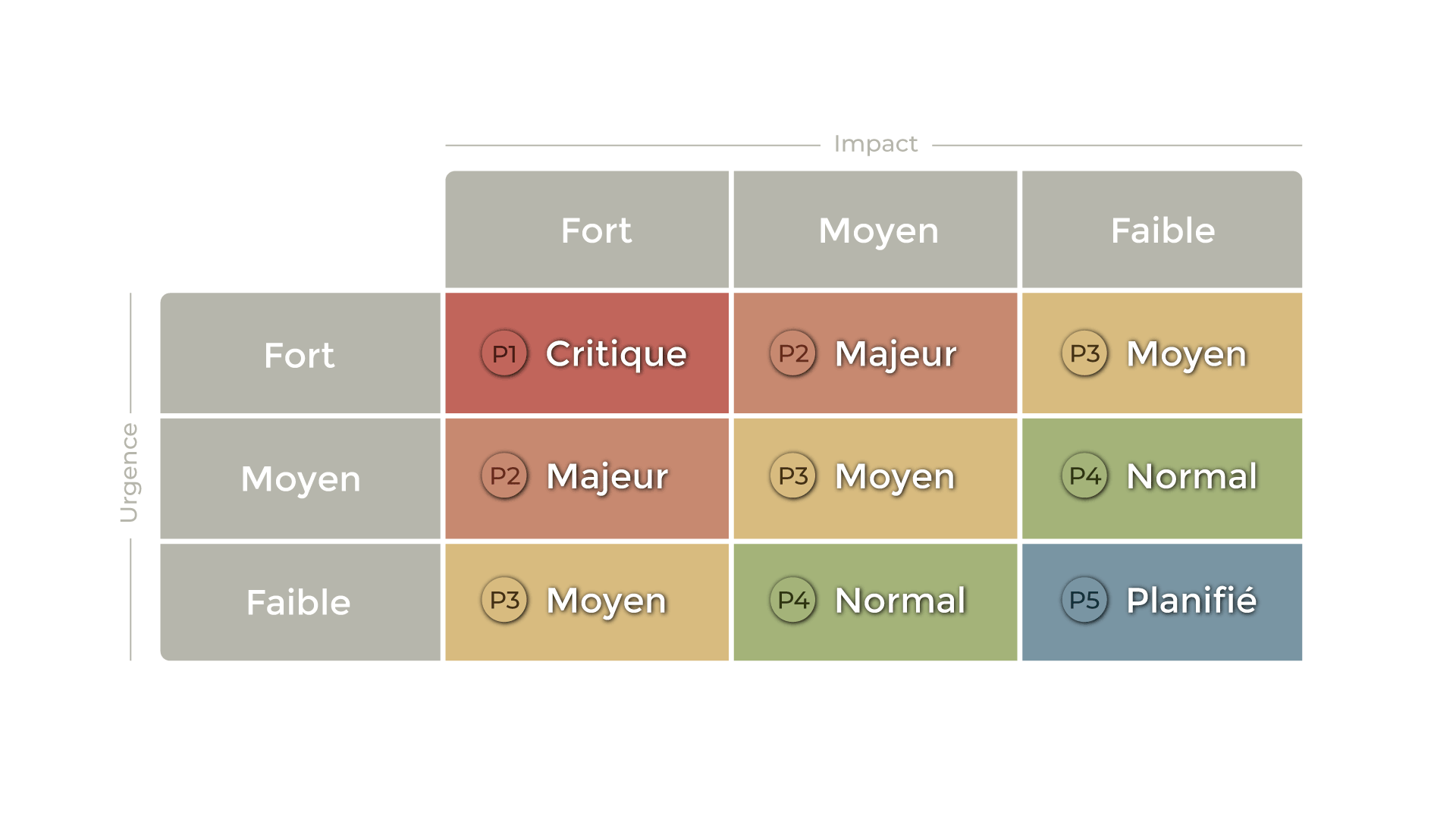
**Catégoriser un changement consiste à lui affecter une priorité.** Dit autrement, il s’agit de lui affecter une importance relative.

Une autre méthode consiste à déterminer la **priorité** d’un changement grâce à une matrice prenant en compte **l’impact et l’urgence** de ce changement :

* **L’impact** représente l’ampleur du périmètre et/ou le nombre d’utilisateurs concernés par ce changement. Il doit être inscrit dans la RFC ;
* **L’urgence** va mesurer le niveau de criticité sur l’activité, l’urgence à mettre en place le changement.

En évaluant l’impact et l’urgence sur **3 niveaux** (fort, moyen, faible), vous pourrez ensuite déterminer la priorité du changement, c'est-à-dire un **couple urgence–impact**.

Vous obtenez **5 niveaux de priorité** que vous pourrez affecter aux changements.



### Planifiez vos changements

De manière générale, vous devrez éviter que vos changements s’interfacent avec d’autres processus de changement (prestataires et fournisseurs externes). Autant que possible, établissez votre **planning de changement en fonction des besoins du business**, plutôt qu’en fonction des besoins informatiques.

Évitez d'effectuer vos opérations aux heures de forte utilisation du service.

**Evitez d’effectuer un changement le vendredi**, car en cas de pépin, l’incident pourrait durer tout le weekend ! D’ailleurs cette règle est suivie dans de nombreuses entreprises.

### Établissez un plan de remédiation

Aucun changement ne devrait être mis en œuvre sans au préalable fournir un **plan de remédiation**. On parle aussi de *Backout Plan*.

Il s’agit de prévoir un plan de retour dans le cas où le déploiement se passe mal (*oui, ça peut arriver !*). Idéalement, **établissez une solution qui vous permettra de redonner au service sa situation initiale**. Restaurez vos sauvegardes du système pour le software, prévoyez du matériel de secours dans le cas du hardware.

Un **snapshot** est une sauvegarde complète du système qui peut être restaurée dans le cadre d’un retour arrière.

### En résumé

Organiser les changements consiste à :

* Définir les activités à mettre en œuvre dans votre processus de changement ;
* Prévoir un processus pour chaque type de changement : **Normal**, **Standard**, **Urgent** ;
* **Commencer par le changement normal**, et simplifier le processus pour obtenir un changement standard, ou changement urgent, sans pour autant supprimer les contrôles et l’analyse des risques liés à ce changement ;
* Utiliser la règle des **7R** afin d’établir le périmètre du changement : **Requis, Raison, Retour, Risques, Ressources, Relation, Responsable** ;
* En fonction de **l’impact et de l’urgence**, catégoriser vos changements en leurs affectant des **priorités** ;
* **Planifier vos changements**, ce qui vous permettra de limiter l’impact sur le service d’un arrêt pour maintenance ;
* Enfin, toujours prévoir un **plan de retour arrière** dans l’éventualité d’un mauvais déroulement.

## Définissez les activités d'un processus de gestion des changements

### Découvrez ce qu'est une procédure de gestion des changements

Votre **procédure opérationnelle** de gestion des changements devra comporter, pour chaque type de changement (urgent, normal, standard) :

* Un **processus de gestion des changements** ;
* Un **RACI pour chaque type de changement**.

### Les activités d’un changement Normal

À la manière d’une recette de cuisine, votre processus doit **préciser les différentes étapes** à réaliser lors d’un changement dans votre SI.

De façon générale, pour un changement normal, vous devrez effectuer les actions suivantes :

#### 1. Créer et enregistrer la RFC

#### 2. Passer en revue la RFC

#### 3. Analyser et évaluer le changement

#### 4. Autoriser le changement

#### 5. Implémenter le changement

#### 6. Revoir et clôturer le changement

### Les activités d’un changement Standard

Les changements standard sont **préautorisés**, car ils présentent **peu de risques**. Il vous revient d’adapter votre processus de changement normal pour le simplifier

Prenons par exemple une demande d’installation d’un nouveau matériel, une imprimante reliée à un serveur d’impression, dans un service. La mise en oeuvre de ce changement standard va se dérouler comme suit :

1. **Création et enregistrement de la demande** via un formulaire ;
2. **Validation de la demande par le manager de service** et le service financier (qui constituent le CAB) ;
3. **Achat, test de la configuration et déploiement de l’imprimante** ;
4. **Revue du formulaire** (avec ajout des détails de l’opération, production de la documentation), puis clôture du dossier.

### Les activités d’un changement Urgent

Un changement urgent requiert des **actions immédiates**. Il est très souvent mis en œuvre pour répondre à une **rupture de service ou des changements correctifs**.

De façon générale, vous devez exécuter **les mêmes activités** que celles d’un changement normal, sauf que cette fois, vous mettez tout en œuvre pour qu'il soit effectué **le plus rapidement possible.**

Cela implique de mettre en place un **eCAB** (un comité consultatif d’urgence), constitué de **membres disponibles et mobilisables sans délai**, et ayant une connaissance du périmètre abordé.

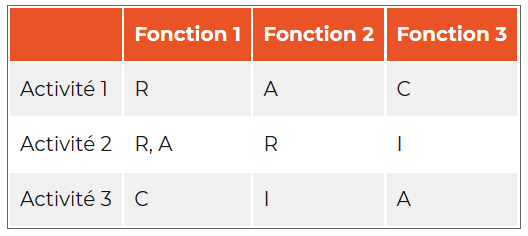
### Déterminez les rôles des intervenants

Comme je l’ai évoqué dans le chapitre 1, la matrice du RACI est utilisée pour définir les **rôles des acteurs dans un processus ou une activité**.

Pour chaque activité du processus, commencez par répondre aux questions ci-dessous pour déterminer les rôles des intervenants :

* **R–Qui réalise** de cette tâche ?
* **A–Qui est le décideur** ? Qui peut trancher en cas de problème ? Qui valide les résultats ?
* **C–Qui détient une expertise** dans le domaine ? Qui peut apporter son aide par de précieux conseils pour faire avancer la tâche ?
* **I–Qui est impacté par l'activité** ? Qui doit être nécessairement informé ?

Ensuite, établissez un tableau attribuant une responsabilité à chaque intervenant.



Toujours un seul A par ligne.

### En résumé

Voici un résumé de ce que vous avez appris **dans ce chapitre** :

* Une **procédure opérationnelle** de gestion des changements détaille les activités de votre processus de changement ;
* Elle va contenir **les processus et le RACI** de chaque type de changement : normaux, standard et urgents ;
* Les **activités essentielles** d’un processus de changement doivent être détaillées dans la procédure opérationnelle ;
* Vous devez établir une matrice RACI qui va **attribuer des rôles aux fonctions**.

### Réalisez l’ordinogramme d’un processus avec XMind

### Qu'est-ce qu'un ordinogramme ?

### L'ordinogramme va nous permettre de ****visualiser**** clairement et simplement ces activités, actions et décisions à prendre.

### Principes de réalisation d'un ordinogramme

Avant de vous lancer dans la réalisation d’un ordinogramme, il est nécessaire d’établir certaines règles :

* Utilisez une forme **ovale** pour représenter un événement qui intervient en entrée ou en sortie de processus ;
* Le **rectangle** représente une activité intervenant en cours de processus ;
* Utilisez le **losange** pour représenter une décision dans le processus ;
* Utilisez un **cercle** pour représenter une activité ou un point qui fait appel à un autre processus.

### Descriptif visuel des différents types d'éléments d'un ordinogramme et de leur form

### Il existe de nombreux outils pour réaliser un ordinogramme, le plus célèbre étant ****Microsoft VISIO****. Toutefois, vous pouvez utiliser des outils gratuits tels que [Présentations](https://docs.google.com/presentation/) de Google, ou [Keynote](https://itunes.apple.com/fr/app/keynote/id361285480?mt=8) de Apple pour réaliser vos ordinogrammes. Vous pouvez également utiliser des outils accessibles directement sur votre navigateur, comme le très puissant [Draw.io](https://www.draw.io).

### Préparez et organisez une mise en production grâce aux packages

### L’étude de cas

La DSI du site OPC a décidé de mettre en place une **solution de supervision** qui permettrait le contrôle en temps réel de **l’état de santé du SI et de ses composants**. Cette supervision permettrait la détection des situations à risque et ainsi éviter les incidents et les pertes liées à l’indisponibilité de la boutique en ligne.

La solution de supervision requiert le déploiement de :

* Un **serveur de supervision** traitant les alertes et évènements remontés par les agents de supervision ;
* Des **agents de supervision** installés sur chaque serveur de production afin de surveiller les services applicatifs, les éléments réseau, et l’état des disques ;
* Une **montée en version de l’OS** sur les serveurs devant recevoir les agents afin qu’ils puissent être opérationnels ;
* La mise en place de **scénarios de supervision applicative**.

Ces changements ont tous été approuvés via le **processus de gestion des changements**, et un planning de déploiement a été établi. Vous abordez maintenant les activités de conception, test et déploiement.

### Livrables, Packages et Mises en Production (MEP)

Un déploiement induit un **changement dans votre SI**, et donc présente des risques. Raison pour laquelle il doit être préalablement approuvé et validé dans le processus de gestion des changements.

Mettre en production la **solution de supervision** va consister à effectuer dans votre SI un ensemble de **changements**. Ces changements sont regroupés en **packages** ou *release* en anglais

### Le terme de ****Mise en Production (MEP)**** est très souvent utilisé pour décrire les changements applicatifs. Notez qu’une MEP s’utilise aussi bien pour du hardware que pour du software. La MEP est constituée d’un ensemble de changements déployés simultanément.

### Pour réaliser un package, vous devez regrouper les changements par **unité de mise en production**.

Le regroupement des changements par unité de mise en production permet de :

* **Faciliter** le nombre de changements nécessaires pour mettre en œuvre une MEP ;
* **Optimiser** les ressources humaines et le temps nécessaires pour concevoir, tester et déployer les changements.

### 

### Pour déployer notre solution, les changements vont être organisés en releases, en regroupant les ****RFC**** en fonction des unités de configuration précédemment établies, mais aussi en fonction des prérequis : il faut installer le hardware avant le software, par exemple.

### Conception du package de mise en production

Le package de mise en production est le résultat de la phase de conception.

Il va permettre de passer d’une situation actuelle à une situation désirée. Il contient tout ce qu’il faut pour effectuer les changements dans le SI.

Dans notre exemple, il est composé du matériel à installer pour le serveur de supervision, du software à installer ou à concevoir, de la documentation ainsi que de la formation.

La phase de conception de la Release 1 va consister à :

* Acquérir le matériel (hardware) qui hébergera le serveur de supervision ;
* Acquérir la distribution de l’OS à installer sur le serveur ;
* Acquérir ou développer la solution de supervision côté serveur.

Une fois la phase de conception achevée, il est temps de tester le package conçu afin de s’assurer qu’il répond au besoin du changement, mais aussi et surtout qu’il s’intègre parfaitement dans le SI.

### En résumé

* Un **package** est un groupe de changements à effectuer via une **mise en production** ;
* La **mise en production** (MEP) est un ensemble de changements autorisés apportés à un service et déployés ensemble ;
* Les éléments de configuration habituellement mis en production ensemble constituent une **unité de mise en production** ;
* Le découpage par unité de mise en production permet d’optimiser les ressources et le temps nécessaires pour construire, tester, distribuer et implémenter une MEP ;
* Un **package de mise en production** contient tout ce qui est nécessaire pour passer de la situation actuelle à la situation désirée. Il s’agit des logiciels, du matériel, de la documentation, et souvent de la formation. C’est le produit de la conception.

### Testez et validez vos packages

### Objectif

### Les **tests et validations** assurent que le package correspond à ses caractéristiques de conception, et qu’il répondra aux besoins du business.

### **Un environnement de test ou recette** est un environnement contenant du matériel, des instruments, des simulateurs, des outils logiciels et d'autres éléments de support nécessaires à l'exécution d'un test. En clair, c'est ce qui permet de **tester le service sans avoir à impacter la production**.

Les tests de validation vont permettre de valider les points suivants :

* La **déployabilité** du package ;
* **utilisation de l’interface** par les utilisateurs ;
* la **documentation** fournie ;
* Le plan de **retour arrière**.

### Validez le passage en environnement de production

### Dans notre exemple, une fois la solution de supervision déployée dans l’environnement de test, il faudra ****simuler des évènements**** (surcharge réseau, coupure, saturation) afin de s’assurer que les alertes remonteront et que les évènements importants sont remontés dans la console de supervision. Cela permettra de ****valider le fonctionnement de notre supervision****.

Le **plan de retour** arrière doit aussi être testé et validé.

Nous en profiterons pour effectuer des tests utilisateurs : l’équipe d’exploitation de l'infrastructure qui sera la première utilisatrice de l’outil de supervision pourra confirmer le bon fonctionnement de l’interface utilisateur, ainsi que les interfaces avec d’autres applications (dans le cas d’utilisation d’une API).

### Déployez une version pilote

### Une ****version pilote**** est une version testée en conditions réelles par un petit groupe d’utilisateurs. Il s’agit de déterminer les écarts entre l’environnement de test (aussi appelé la "recette") et l’environnement de production.

Dans le cas de OPC, vous pouvez constater que la release 2 déploie la supervision sur serveur de production secondaire afin d’effectuer des tests en situations réelles, sans pour autant perturber la production.

Le déploiement d’une version pilote ne peut intervenir qu'**après la validation en phase de test**. Durant la phase pilote, les équipes de déploiement doivent :

* **Prévoir un plan de retour arrière** s’il existe des écarts non acceptables ;
* S’assurer que les **utilisateurs** en phase pilote soient **formés** ;
* S’assurer que la **documentation initiale** soit disponible au service desk ;
* Diagnostiquer et **résoudre les incidents** qui pourraient survenir ;
* Documenter les **améliorations** à apporter au projet.

Une fois les tests validés, le déploiement en production des changements peut sereinement avoir lieu.

### En résumé

* Les tests et la validation de service assurent que le service **correspond à ses caractéristiques** de conception et qu’il **répondra aux besoins** du business ;
* Il est impératif de déployer le package de mise en production dans un **environnement de test** ;
* L’environnement de test doit être **maîtrisé, protégé et maintenu** pour éviter les écarts avec l’environnement de production ;
* Une **version pilote** est une **version testée en conditions réelles** par un petit groupe d’utilisateurs. C’est une version dite de **préproduction** qui permet de déployer en production le package, sans perturber la production.

### Comprenez les grandes étapes des phases de conception, test et déploiement

Lors des phases de conception, test et déploiement, vous devez :

* Mettre en place une **identification unique** : une numérotation et une convention de nommage (ex: V1.3.4) ;
* Vérifier la **fréquence** prévue pour chaque MEP ;
* Vérifier l’approche **d’acceptation et de regroupement des modifications** dans une MEP (création des packages) ;
* Gérer les **rôles et responsabilités** à chaque étape de conception, test et déploiement ;
* Assurer des **critères d’acceptation en fin de déploiement** qui permettent la sortie de la phase de transition vers la phase d’exploitation du service.

En fonction du type de service à déployer et de ses potentiels impacts, vous devrez choisir une méthode de déploiement adaptée à la situation.

### Les différentes approches de mise en œuvre du déploiement

#### Approche phasée

Dans une **approche phasée**, le service est déployé pour **une partie des utilisateurs au départ**, puis cette opération est répétée pour les parties suivantes de la base des utilisateurs, avec un planning de déploiement. Ce type de déploiement permet d’effectuer un **retour arrière** et de minimiser les coûts d’un déploiement raté à grande échelle.

#### Approche Big Bang

Dans une **approche Big Bang**, le service est déployé pour **l’ensemble des utilisateurs en une seule fois**. Cette option est utilisée pour les changements sur les applications, et lorsque la cohérence du service sur l’ensemble des utilisateurs est primordiale.

#### Approche push

L'approche de type **push** est utilisée lorsque le composant de service est déployé **depuis l’espace de distribution et poussé vers les utilisateurs** cibles par l’administrateur du service.

#### Approche pull

L'approche de type **pull** est utilisée pour les productions de logiciels, lorsque le logiciel est **mis à disposition en centrale de distribution** (du type App store). Un bon exemple est celui des signatures de virus, qui sont téléchargés pour mettre à jour le PC ou le serveur, au moment où l’utilisateur le désire.

### Fonctions, rôles et responsabilités

Plusieurs rôles et responsabilités peuvent être identifiés lors des phases de conception, test et déploiement.

Il s’agit entre autres de la construction des packages de MEP, avec les responsabilités suivantes :

* Établir la **configuration de la mise en production finale** ;
* **Concevoir le livrable** ;
* **Tester le livrable** final avant les tests indépendants ;
* Établir les **rapports** sur les erreurs connues en suspens et les solutions de contournement à communiquer ;
* Participer à la **validation** de la mise en œuvre finale.

Le personnel de déploiement aura les responsabilités suivantes :

* **Délivrer** physiquement la MEP des services ;
* **Coordonner la communication et la documentation** relatives à la mise en production, la formation et les notes aux équipes techniques ;
* Fournir un **guide d’application** technique et un support tout au long du processus de mise en production, y compris les erreurs connues et les solutions de contournement ;
* Fournir un **feedback** concernant l’efficacité de la mise en production.

Le personnel de **support** de début de vie participe à la phase de déploiement et de mise en production et aussi à la **première période d’exploitation**. Par exemple, pour résoudre les incidents avec le service modifié ou nouveau, ou pour parfaire la documentation.

### En résumé

* Il existe **4 méthodes de déploiement** d’un service ;
  + Approche **phasée** : le service est déployé pour une partie des utilisateurs au départ, puis l’opération est répétée pour les autres groupes d’utilisateurs ;
  + Approche **Big Bang** : le service est déployé pour l’ensemble des utilisateurs en une seule fois ;
  + Approche de type **push** : lorsque le composant de service est déployé et poussé vers les utilisateurs par l’administrateur du service ;
  + Approche de type **pull** : lorsque le logiciel est mis à disposition en centrale de distribution ;
* Il existe **2 fonctions principales** dans la mise en oeuvre des phases de conception, test et déploiement ;
  + La fonction de **construction des packages de MEP** a pour rôles de **construire et tester** le livrable, mais aussi de **participer à la validation** de la mise en œuvre finale ;
  + Le personnel de déploiement aura les responsabilités principales de **délivrer physiquement la MEP**, **coordonner la communication et la documentation** et **assurer la formation** et les notes aux équipes techniques.