

Diplôme fédéral en informatique

Gestion des changements
techniques



Sommaire

1. Construction des versions.....	11
1.1 Composants d'une version.....	15
1.2 Versioning.....	20
1.3 Configuration nécessaire.....	21
1.4 Classes d'applications.....	24
2. Analyse de l'existant.....	29
3. Ressources.....	35
3.1 Humaines.....	36
3.2 Temporelles.....	36
3.3 Outils de déploiement.....	40
4. Stratégies de déploiement.....	43
5. Planification.....	49
5.1 Inventaire des tâches.....	50
5.2 Dépendances entre les tâches.....	50
5.3 Durées des tâches.....	52
5.4 Qualification des intervenants.....	53
5.5 Jalons.....	54
6. Formation.....	55
7. Tests.....	65
8. Annulation.....	75
9. Déploiement et contrôle.....	79
10. Documentation.....	85
11. Transfert.....	87
Annexe 1 : gestion des configurations.....	96
Annexe 2 : gestion des changements.....	113

Annexe 3 : fiche de scénario de test.....	119
Annexe 4 : fiche de résultat de test.....	120
Lexique.....	121
Bibliographie.....	127
Table des illustrations.....	129
Table des matières.....	131



Introduction

Il existe plusieurs phases particulièrement critiques dans le cycle de vie d'un système d'information. L'une d'entre elles se situe au moment de la spécification d'une nouvelle solution, car une erreur provoque la consommation d'importantes ressources pour obtenir un outil qui ne répondra pas complètement aux besoins de l'organisation. L'autre moment critique se situe au moment où une solution quitte la phase de réalisation pour entrer en phase de production.

C'est sur le déroulement de cette phase que porte le présent manuel, en considérant que les phases précédentes ont fourni les résultats auxquels on peut s'attendre lorsqu'on utilise une approche structurée, par exemple les bonnes pratiques ITIL¹ et la gestion de projet selon HERMES, dont les phases et les principales activités sont représentées sur la figure suivante.

¹ tous les termes soulignés en pointillés sont définis dans le lexique, page 121 et suivantes.

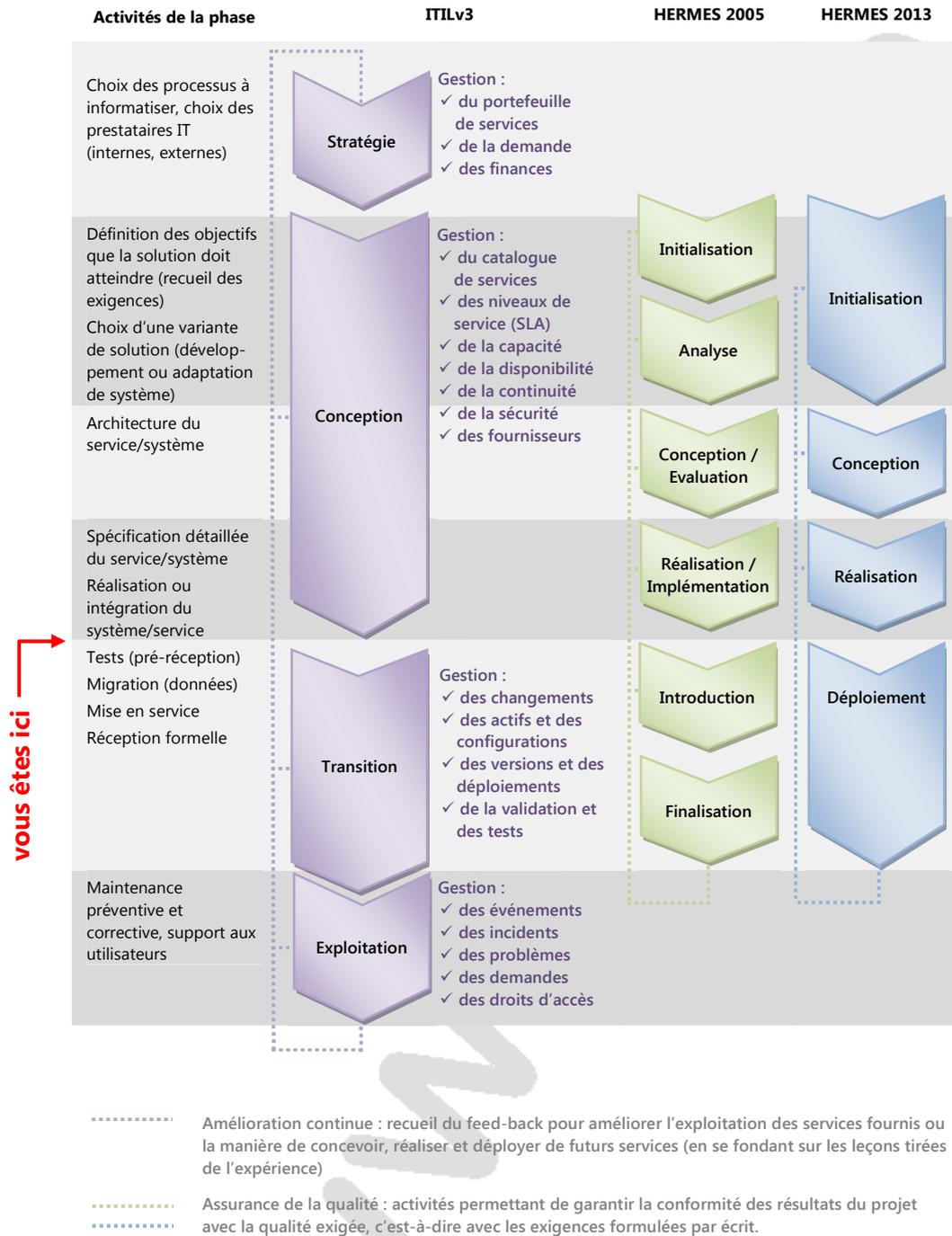


Figure 1 : phases du cycle de vie d'une solution IT selon ITIL et HERMES

L'étape de mise en production – également appelée transition, introduction ou déploiement – engendre de forts risques pour la stabilité, la disponibilité et la performance du système d'information (SI) et, par conséquent, pour l'activité de l'organisation qui l'utilise.

Parmi ces risques, citons :

- indisponibilité partielle ou totale du SI pour plusieurs heures voire plusieurs jours, entraînant une perte d'exploitation, des retards sur les activités et une perte d'image de marque auprès de la clientèle,
- impossibilité d'annuler certaines modifications non souhaitées du SI, entraînant une période d'exploitation en mode dégradé, et donc des coûts additionnels pour les opérations supplémentaires que doivent effectuer les utilisateurs et l'équipe IT, ainsi que des retards dans les opérations causés par ce surcroît de travail,
- perte de la confidentialité des informations, dans le cas où les modifications non souhaitées introduisent des failles dans la sécurité du système ou si les nouveaux composants introduits ne sont pas suffisamment sécurisés.

S'il s'agit de changements touchant aux applications, on risque aussi :

- la perte d'intégrité des données. Une perte d'intégrité qui ne serait pas détectée immédiatement est particulièrement grave car la restauration de sauvegardes n'est plus possible après une certaine durée d'exploitation. Il est alors nécessaire de procéder à des vérifications manuelles ou d'écrire un programme qui effectue ces contrôles.

Risque moindre en termes d'impact sur l'activité et l'image de marque mais notable en termes de coûts de maintenance, citons enfin :

- l'augmentation des incidents pendant plusieurs semaines voire plusieurs mois. Les incidents englobent aussi bien les dysfonctionnements que les demandes de support émanant d'utilisateurs déstabilisés par les nouveautés introduites.

Tout le défi de la mise en production consiste donc à introduire des éléments nouveaux dans le SI existant en préservant sa stabilité et ses fonctionnalités. Les caractéristiques de la version à déployer ainsi que celles du système cible doivent être parfaitement maîtrisées afin que leur conjonction ne génère aucun changement imprévu.



Analogie

Un SI fonctionne comme une usine : il utilise des matières premières (les données) pour fournir des produits aux clients (des informations, des traitements, des documents). Il stocke la matière première et les produits finis et exploite plusieurs lignes de production dédiées à différents produits. Tout dysfonctionnement dans les systèmes de stockage ou de production entraîne des retards de livraison, l'inactivité d'employés et, au pire, la destruction de matières premières et la fabrication de produits défectueux. Tout modification apportée à un SI comme à une usine doit être faite avec précautions.

En outre, l'introduction nécessite des ressources (temps, personnes) et doit se dérouler le plus rapidement possible afin de minimiser le dérangement causé aux utilisateurs, ce qui implique un important travail de planification et une gestion de projet rigoureuse.

L'aspect humain doit être pris en compte, tant au niveau des utilisateurs finaux que du personnel IT : le gain attendu de la nouvelle solution dépend fortement de l'adoption par les premiers et de la réussite de la migration effectuée par les seconds. Dans les deux cas, une formation appropriée doit être dispensée.

Enfin, le processus de déploiement doit être parfaitement contrôlé : les erreurs sont prévues et des procédures d'annulation permettent de rétablir l'état initial. Les points de contrôle et les responsabilités individuelles sont précisés.

Gérer les versions consiste à élaborer un processus formalisant toutes ces activités, comme représenté par exemple sur le schéma ci-contre, de manière à ce que toutes les modifications du SI soient réalisées selon des procédures garantissant leur succès ou, tout au moins, préservant l'existant.

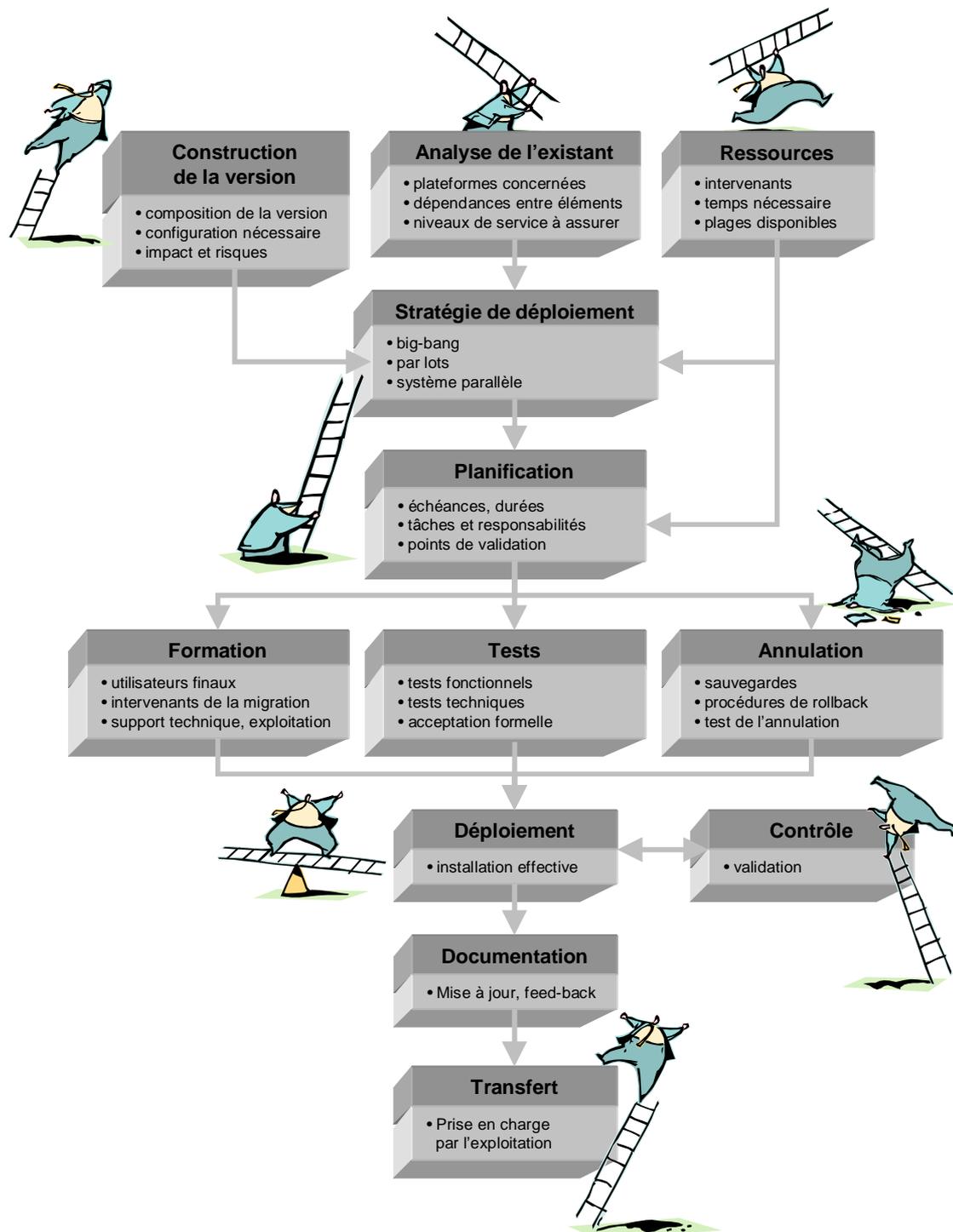
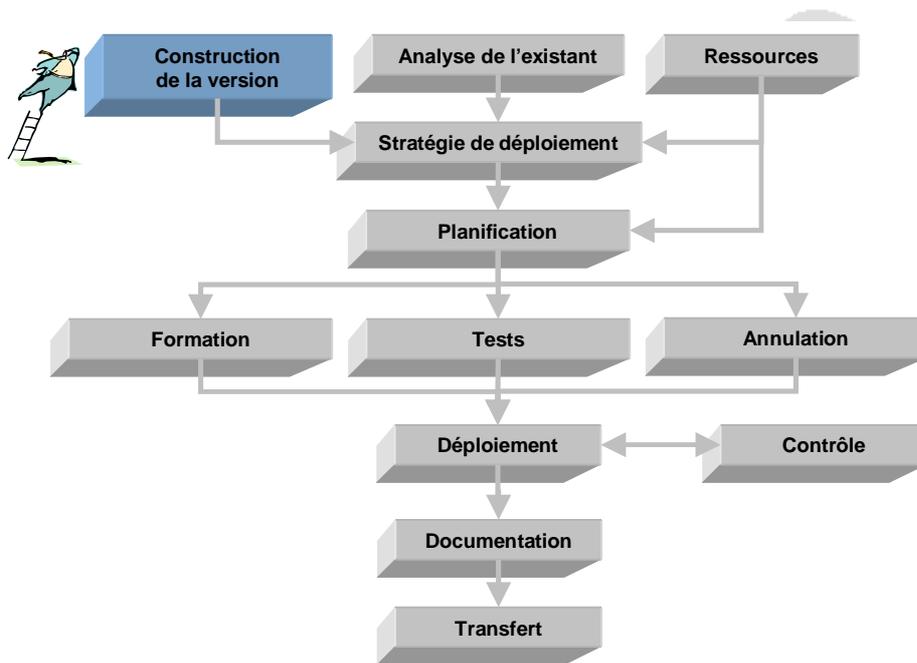


Figure 2 : organigramme de déroulement d'une mise en production

Pour illustrer les activités à mener et les choix à faire à chaque étape de la gestion des versions et de leur déploiement, quatre organisations fictives mais représentatives nous accompagneront de chapitre en chapitre :

- ⊕ Architech, cabinet lausannois employant une vingtaine d'architectes et d'ingénieurs en génie civil, qui remplace les trop nombreux logiciels administratifs utilisés jusqu'ici par un unique progiciel de gestion intégré (PGI) et en profite pour renouveler la moitié la plus ancienne de ses postes clients, ce qui implique une nouvelle version d'OS et de suite bureautique,
- ⊕ Ecobioclim, leader mondial des systèmes de climatisation écologiques fabriqués de manière équitable et presque biologique, qui a décidé de munir ses forces de vente de smartphones gérés par une solution MDM, ce qui nécessite une version « mobile » de ses principales applications métiers ainsi que l'intégration des smartphones existants,
- ⊕ l'Office fédéral des initiatives farfelues mais inventives (OFIFI), qui doit assurer une protection de l'information exemplaire et doit donc déployer immédiatement les correctifs urgents sur les navigateurs de son parc client,
- ⊕ Vendratu, important cabinet d'analyse des parts de marché et d'e-réputation, qui a développé son propre logiciel pour la collecte et la synthèse d'informations – Bigsoop – dont les agiles développeurs livrent des mises à jour quasiment chaque mois.



1. Construction des versions

Avant toute chose, il est nécessaire de définir le sens que nous donnerons au terme « version », qui varie selon que l'on se trouve dans le domaine du développement de logiciel, dans la terminologie ITIL ou encore dans le langage courant.

Lors d'une conversation avec un éditeur ou un développeur de logiciel, « version » indiquera une application et ses caractéristiques techniques et fonctionnelles à un moment précis dans le temps. Dans le cadre de l'exploitation de systèmes d'information, phase qui succède à celle du développement logiciel, « version » prend un sens plus large : on inclut non seulement le logiciel qui est au centre de la solution attendue, mais

également tous les éléments qui doivent être installés dans l'environnement de production pour assurer qu'il fonctionne conformément à ses spécifications et fournit bien le service que l'utilisateur attend.

Cette définition orientée service s'inspire notamment des bonnes pratiques structurées au sein de l'ITIL, qui recommande de confier à plusieurs processus complémentaires les opérations touchant à l'évolution du SI :

- 📄 la gestion des changements, qui **organise** le changement : réception de la demande (RFC), évaluation des risques et coûts, acceptation, planification du changement, contrôle de l'implémentation,
- 📄 la gestion des configurations, qui **documente** les composants du système d'information ajoutés, modifiés ou impactés par le changement,
- 📄 la gestion des versions qui **réalise** le changement : élaboration de la version à mettre en production (assemblage des changements à apporter simultanément), tests et définition d'un plan de restauration en cas d'échec du changement, réalisation du changement.

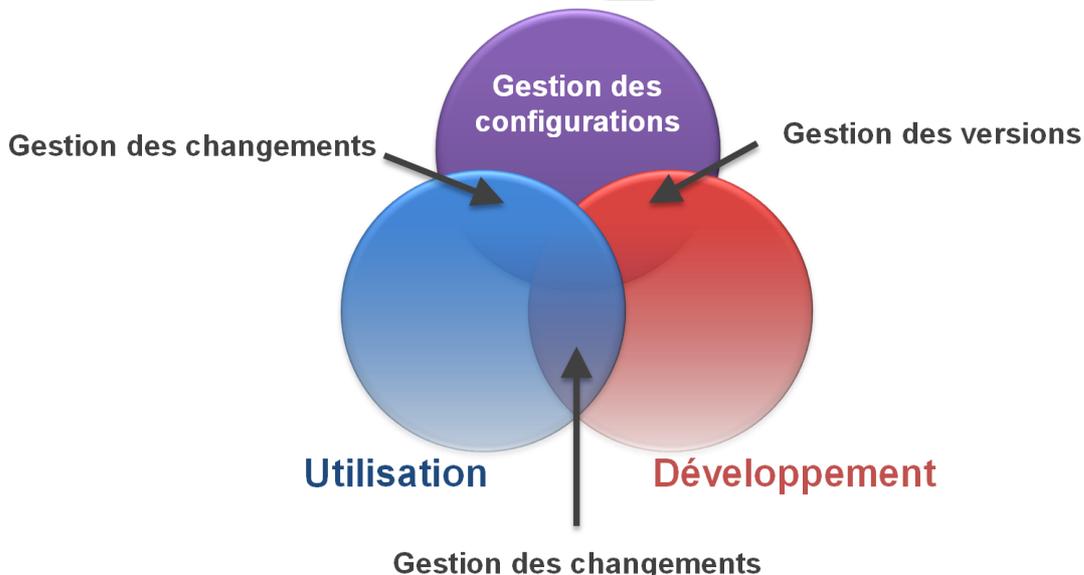


Figure 3 : processus ITIL de transition et domaines IT