

Tests et recettes dans les projets Système d'Informations : une étape délicate

Toutes les directions métier ou informatique connaissent l'importance des tests. Rares sont celles qui sont réellement satisfaites de la place effectivement occupée par cette activité. On a le sentiment d'en faire trop ou pas assez, que personne ne se satisfait vraiment des résultats obtenus, que les intervenants sont peu motivés. L'image des tests reste mitigée. Aucune solution n'apparaît réellement satisfaisante, définitivement acquise.

Pourquoi ce déficit d'image, pourquoi ce sentiment ?

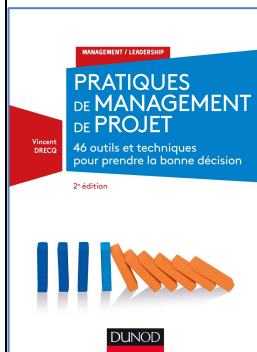
Les tests sont l'activité qui fâche par excellence alors que spécifier avec les métiers est passionnant, concevoir la solution est créatif, développer techniquement ou faire évoluer le logiciel est constructif et l'exploiter en production c'est rendre service tous les jours.

Et puis, tester arrive toujours au mauvais moment, puisque la qualification devient l'activité dominante à la sortie du tunnel de la réalisation, lorsque l'on est pressé de mettre en production.

Beaucoup d'interlocuteurs doivent se remettre autour d'une table : les métiers, les développeurs ou mainteneurs et les exploitants.

Pour finir, l'affaire n'est pas simple : la démarche de qualification fonctionnelle suppose une préparation importante, basée sur les exigences fonctionnelles initialement collectées. Des scénarios doivent être produits pour refléter le déroulement attendu des procédures informatisées, cas par cas. La fabrication technique des jeux d'essai ou des bases de test est une expertise en elle-même...

Vincent Drecq



Professionnel certifié PMP, il travaille depuis plus de 20 ans dans le management de projets. Sa vision du projet est très pragmatique et orientée vers les résultats.

Il est l'auteur de « Pratiques de management de projet, 46 outils et techniques pour prendre la bonne décision » aux éditions DUNOD

Cette fiche thématique est gratuite, n'hésitez pas à la diffuser largement à votre entourage.

SOMMAIRE

Qu'est-ce que la recette ?.....	2
La stratégie de recette	3
Le cahier de recette	5
Les recettes	6
Les vérifications	7
Les risques	8
Quelques définitions	9

Qu'est-ce que la recette ?

La « recette », c'est l'opération par laquelle le client reconnaît que le produit livré par le fournisseur est conforme à la commande passée, qu'il est exploitable dans le système d'informations de l'entreprise, et enfin qu'il est opportun de le mettre à la disposition des utilisateurs.

Pourquoi doit-on tester une application ?

- **Pour obtenir, tout d'abord, la satisfaction du client.** Mais cette dernière s'obtient progressivement et non pas le jour de la réception
- **Parce que le bon fonctionnement de l'application doit être garanti.** Les risques liés aux défaillances doivent être maîtrisés. La réception d'un logiciel n'est pas un exercice de corde raide mais est l'aboutissement d'une démarche maîtrisée.
- **Parce que le désordre coûte.** Les modèles itératifs, tel le Unified Process, permettent de démarrer les tests fonctionnels participant à la vérification, beaucoup plus tôt dans le déroulement du projet. La conception des tests peut ainsi démarrer dès que les exigences fonctionnelles d'une itération sont disponibles, et leur exécution dès qu'une version exécutable de l'application est disponible. La validation demeure à la fin du cycle, en raison de son caractère « contractuel ». On observe une croissance exponentielle des coûts d'une anomalie en fonction du moment où elle est découverte (plus on la découvre tard dans le projet, plus elle coûte cher).

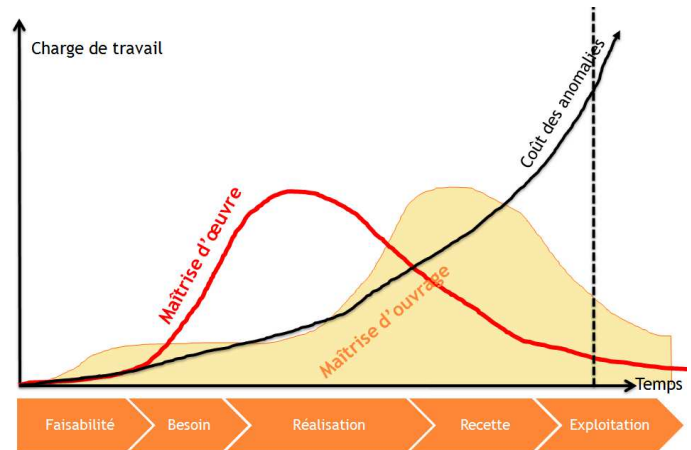


Fig. 1 : Charge pour traiter une anomalie

- **Pour éliminer le stress lié à l'incertitude**
- **Parce que l'on a tout à y gagner**
 - ✓ Diminution des coûts de maintenance,
 - ✓ Optimisation de la réception
 - ✓ Amélioration de la fiabilisation
 - ✓ Satisfaction du travail bien fait
 - ✓ Remotivation des équipes

La recette est réalisée selon des procédures qui ne s'improvisent pas. L'anticipation de cette activité permet d'améliorer l'efficacité des tests (simplification et optimisation des tests).

La recette se décompose alors en deux grandes phases :

Phase 1 : La préparation

Cette phase consiste à bâtir la stratégie de test. Il s'agit de planifier les différentes activités sans négliger la préparation logistique nécessaire à une réalisation dans de bonnes conditions.

Phase 2 : La réalisation

Les tests sont réalisés, les bogues sont remontés, le reporting informe le management et un bilan permet d'améliorer la prochaine série de tests.



Fig. 2 : Démarche de planification et réalisation de tests itératifs

La stratégie de recette

La stratégie de tests consiste à déterminer précisément les vérifications qu'il est indispensable de conduire. Elle permettra de définir la méthode à utiliser et les preuves à apporter. Le cahier de recette doit permettre de répondre à quelques questions :

- Qu'est-ce qui sera testé ?
- A partir de quand les tests débiteront ?
- Jusqu'à quelle profondeur les tests seront effectués ?
- Quel planning de tests est envisagé ?
- Quelle organisation sera mise en œuvre ?
- Est-ce que l'ensemble de l'application sera concernée par les tests ?
- Quelles preuves tangibles seront fournies pour justifier de la bonne réalisation des tests ?
- ...

La stratégie de test doit répondre à plusieurs exigences :

- Elle doit rester souple pour éviter les blocages

- Elle doit intégrer les risques techniques et prévoir des alternatives
- Elle doit rester centrée sur son but: atteindre ses objectifs et identifier les chemins pour les atteindre
- Elle doit être « collaborative » avec les parties prenantes - cela est encore plus vrai en approche itérative: les points de contacts entre équipes doivent être bien identifiés, les « règles de priorité » doivent être claires pour tous.

La gestion d'une campagne de tests doit donc répondre à deux intérêts qui peuvent paraître antinomiques: souplesse et rigueur. ♦

Ainsi, on s'attachera à garder ces valeurs à l'esprit lors du déroulement d'une campagne de tests afin de trouver le bon équilibre entre la rigueur nécessaire au suivi et à l'évaluation de l'exécution, et la souplesse nécessaire pour s'adapter aux perturbations inévitables.

La démarche globale des tests doit suivre les étapes clé suivantes :

- Définition des objectifs :
Prendre en compte l'ensemble des spécificités du projet : exigences clients, criticité, caractéristiques produit, ...
- Estimation et identification des moyens disponibles :
Humains, matériels, logiciel, ...
- Planning de mise en œuvre :
Ordonnancement, composants unitaires, responsabilités, environnements informatiques, jeux d'essais, ...
- Organisation du processus de réalisation des tests :
Traçabilité, ...
- Suivi de l'avancement des tests :
Tableaux de bord de suivi, % avancement, ...
- Gestion des écarts :
Résultats obtenus/résultats attendus
- Reporting :
Remontée des points bloquants, ...
- Réalisation d'un bilan :
Évaluation des tests

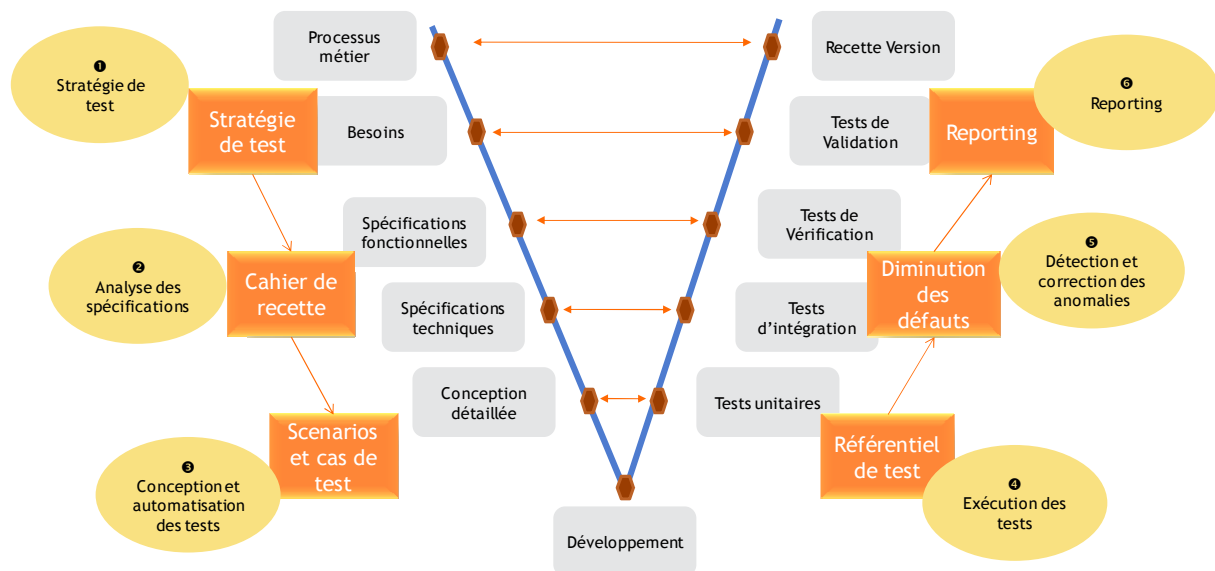


Fig. 3 : Les tests et le cycle de développement

On distingue ainsi sous le nom de **vérification** les activités de tests visant à s'assurer que le produit développé correspond en tout point à la solution décrite dans le dossier de spécifications ; et sous le nom de **validation** les tests visant à s'assurer que le produit correspond bien au besoin du donneur d'ordre (La MOA, Maîtrise d'Ouvrage).

La vérification relève donc généralement de la maîtrise d'œuvre et la validation de la maîtrise d'ouvrage. La vérification fait généralement apparaître des défauts de conception et des non conformités fonctionnelles, alors que la validation remonte le plus souvent des défauts dans le recueil des besoins ou des ambiguïtés dans la compréhension et la formalisation des exigences.

Le cahier de recette

La théorie de l'informatique, confirmée d'ailleurs par la pratique, enseigne qu'il est impossible de prouver qu'un logiciel ne comporte aucune erreur. L'application du théorème de Gödel nous permet d'affirmer qu'il est impossible de concevoir un programme capable de vérifier tous les programmes. Certes on peut définir des méthodes de vérification efficaces et les outiller de sorte qu'elles soient faciles à utiliser. Mais ces méthodes, aussi efficaces soient-elles, ne garantissent pas que tous les programmes qu'elles valident soient corrects.

[...] quel que soit le soin que l'on apporte à la définition des tests qu'elle comporte, on n'aura jamais la certitude que le logiciel ne comporte aucun bogue.

La procédure de recette ne peut donc pas être exhaustive : quel que soit le soin que l'on apporte à la définition des tests qu'elle comporte, on n'aura jamais la certitude que le logiciel ne comporte aucun bogue.

On peut toutefois, et cela relève du bon sens, vérifier qu'il remplit correctement les fonctions essentielles que l'on attend de lui que ce soit en termes de performances, de qualité des données ou d'interface homme-machine. La liste des tests à réaliser doit être établie à froid, avant la livraison du produit ; elle porte le nom de « **cahier de recette** ».

La combinatoire des tests possibles étant infinie, le cahier de recette constitue une sélection au sein de cette combinatoire. Certains tests sont coûteux en raison de la volumétrie ou des difficultés techniques qu'ils impliquent. Comme le budget accordé à la recette est limité, le cahier de recette doit idéalement comporter la liste de tests la plus efficace pour un coût donné. Il est nécessaire d'adapter la démarche de tests au cycle de vie et à la criticité du projet.

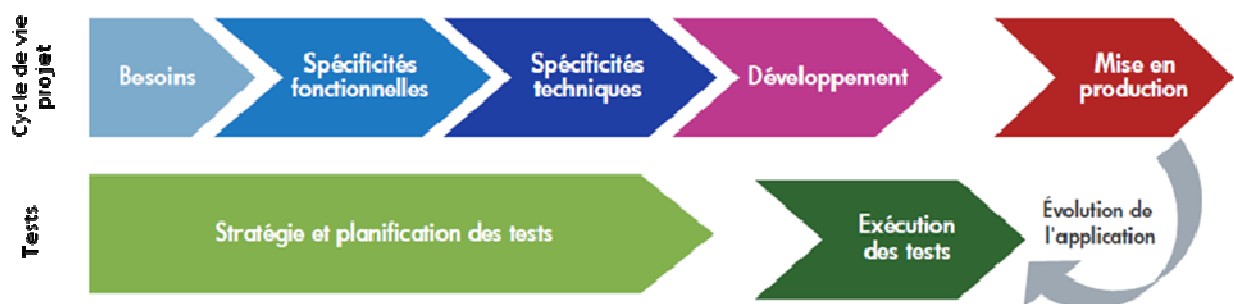


Fig. 4 : Positionnement des tests dans le déroulement d'un projet

Il faut que la recette soit effectuée en partant de « vraies données » et non de données de qualité parfaite. On sait en effet qu'en informatique de gestion, les données sont souvent, pour des raisons parfaitement compréhensibles, de qualité variable (codages imparfaits, données man-

quantes, vérifications insuffisantes). C'est par rapport à cet état de fait qu'il faut évaluer la qualité de l'opération, et non par rapport au monde idéal où les données seraient sans défaut.

Il faut préciser le protocole selon lequel la recette sera organisée :

quelles seront les tâches qui incomberont au client, celles qui incomberont au fournisseur ; quels seront les documents qu'ils devront se communiquer ; dans quel ordre les tests seront réalisés ; quels seront les seuils d'acceptation du produit. Si le protocole n'est pas défini à l'avance, le risque de conflits entre client et fournisseur sera élevé.

Les recettes

Recette « usine » et recette « utilisateur » :

On distingue deux étapes dans la recette : la « recette usine », faite avant la livraison du produit par le fournisseur, permet à celui-ci de vérifier que le produit est conforme à la commande reçue ; la « recette utilisateur » est faite par le client après livraison.

La recette usine

Il faut que le compte rendu de la recette usine soit livré par le fournisseur en même temps que le produit : ce compte rendu apportera au client la preuve que le produit a été sérieusement testé avant sa livraison, et permettra de gagner du temps en ne refaisant pas les tests déjà réalisés par le fournisseur. Il faut prévoir la livraison du compte rendu de la recette usine dans le protocole de recette, sinon le client aura du mal à l'obtenir.

Lors de cette phase, on réalise des tests unitaires et des tests d'intégration. Les tests unitaires permettent d'assurer les fondations de la construction logicielle. Ces tests permettent de limiter les écarts par rapport à toutes les spécifications du composant à tester et d'assurer la fiabilité de la phase de codage/paramétrage.

Quant aux tests d'intégration, ils permettent d'assurer un bon fonctionnement de l'assemblage logiciel/logiciel et logiciel/matériel.

La recette utilisateur comporte deux étapes :

- ✓ **une recette technique**, réalisée par la direction informatique du client, vérifie que le produit est exploitable sur la plate-forme informatique de l'entreprise (compatibilité avec ses matériels, systèmes d'exploitation et logiciels) et que la performance physique est acceptable (volumétrie des bases de données et des flux de messages, délais d'affichage sur les écrans des utilisateurs, robustesse en exploitation [évaluation du système aux cas limites]). Des tests de respect de l'ordonnancement sont également réalisés. L'objectif étant de vérifier la dynamique des tâches temps réel (synchronisation, priorités, parallélisme, précedence,...);
- ✓ **une recette fonctionnelle**, réalisée par la maîtrise d'ouvrage, vérifie que le produit fournit les fonctionnalités demandées par le cahier des charges et qu'il est acceptable par les utilisateurs.

La stratégie de tests doit également comporter une stratégie de tests de non régression. ♦

Les anomalies détectées :

Les tests détectent des anomalies ; chacune d'elles fait l'objet d'une «fiche d'anomalie» envoyée au fournisseur. Celui-ci corrige les anomalies jusqu'à convergence des tests. Il arrive parfois que la correction d'une anomalie provoque d'autres anomalies, ce qui peut obliger à refaire des tests qui avaient auparavant donné un résultat acceptable. On parle alors de **tests de non régression** permettant de s'assurer que les modifications effectuées dans le logiciel ne dégradent

pas son comportement validé antérieurement. Ces tests sont à demander au fournisseur mais le client se doit de vérifier la non régression. Cela suppose alors de pouvoir comparer avec des résultats passés :

- Scénarios de tests
- Base de données d'essais
- Environnements identiques

La stratégie de test doit également définir une stratégie de non régression.

Il est normal, inévitable même, qu'un logiciel d'une certaine importance contienne des erreurs : la détection d'anomalies au début de la recette n'a donc rien de scandaleux, même si elle suscite toujours une tension entre client et fournisseur. Le véritable critère de qualité réside dans le délai de correction des anomalies : si ce délai est long, si les corrections suscitent d'autres anomalies de telle sorte que le volume d'anomalies à traiter ne diminue pas, le client doit s'interroger sur la qualité du logiciel et sur son évolutivité.

Les vérifications

Lorsque les tests de recette ont convergé, le client prononce une «**Vérification d'Aptitude au Bon Fonctionnement**» (VABF). Il est d'usage d'associer ce jalon à un jalon de facturation partielle. Puis le produit est mis en exploitation sur un site pilote. On détectera alors d'autres anomalies, puisque le cahier de recettes ne pouvait pas être exhaustif. Elles devront elles aussi être corrigées. La convergence de ces corrections peut demander quelques mois.

Cette étape terminée, le client prononce la «**Vérification de Service Régulier**» (VSR), ce qui est souvent associé au dernier jalon de facturation du fournisseur. Le produit peut alors être déployé sur tous les sites de l'entreprise, et l'on passe à l'étape du déploiement.

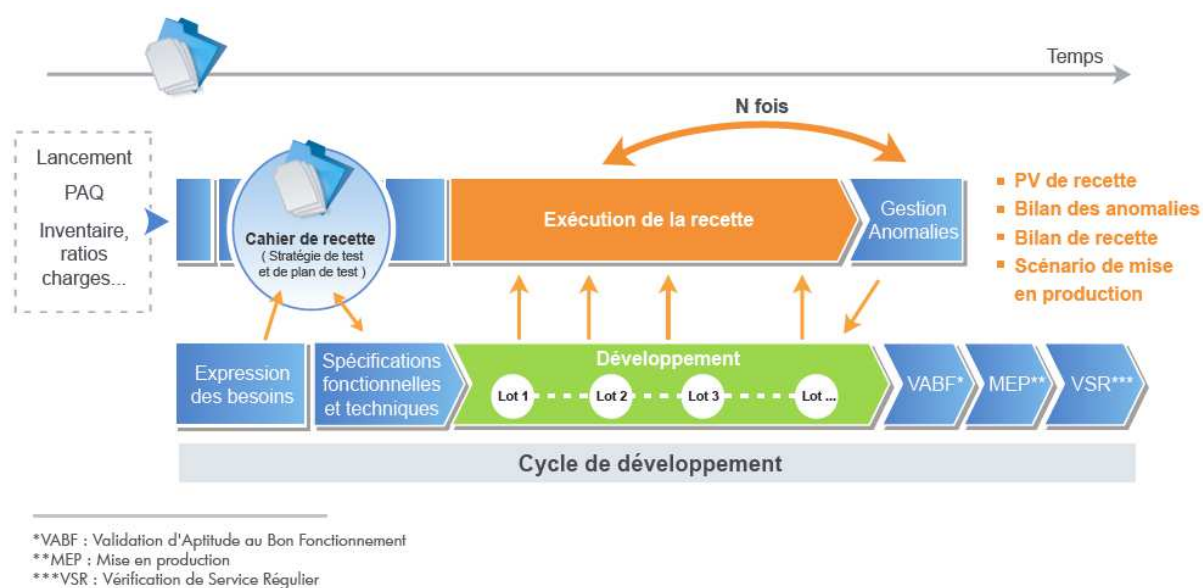


Fig. 5 : La réception d'un projet

Les risques

Toute recette présente des risques. Le test est l'application concrète de la gestion des risques. Évaluer le risque est donc la raison d'être des tests, le corolaire pouvant se formuler de la manière suivante : tester c'est choisir. Une bonne stratégie de tests devra donc s'intéresser aussi au processus de production du logiciel afin de détecter les points faibles qui ont pu causer l'introduction de défauts.

Certaines anomalies n'apparaîtront donc que lors du test en site pilote. Il est donc préférable que le passage entre la vérification d'aptitude et la mise en exploitation soit clairement défini: si le produit est de grande taille, et livré sous forme de lots successifs, on s'efforcera de définir ces lots de sorte que chacun soit un « module exploitable », qu'il soit possible de le mettre en exploitation sur un site pilote dès sa livraison afin d'éviter l'« effet tunnel » qui se produit lorsque la mise en exploitation ne peut se faire qu'après la réception de l'ensemble des lots.

L'application de la stratégie de tests reste le point délicat du processus et conditionne la bonne réussite d'une campagne de tests fonctionnels. L'activité de tests a en effet pour particularité de ne jamais être « finie » : on peut tester une application à l'infini et continuer de trouver des défauts. Cette particularité impose une organisation très différente des autres activités d'ingénierie logicielle. Il convient de déterminer avant le démarrage des tests, les conditions d'arrêt de la campagne de tests, conditions qui déterminent les critères permettant de conclure à la réussite ou à l'échec de la campagne. Lors du déroulement d'une campagne, ces critères doivent être suivis précisément et régulièrement afin de mesurer à tout instant l'écart par rapport à l'objectif, et prendre les décisions nécessaires à son atteinte ou bien ar-

rêter la campagne. On peut noter parmi les critères typiques :

- le taux de couverture des fonctions ou des processus métier
- le taux de couverture des règles de gestion
- le taux de réussite des tests
- le nombre de défauts mineurs, majeurs, et bloquants

Le test est l'application concrète de la gestion des risques. Évaluer le risque est donc la raison d'être des tests, le corolaire pouvant se formuler de la manière suivante : tester c'est choisir. ♦

Il convient, lors de l'élaboration du cahier de recette, de ne pas trop hiérarchiser les tests: on risquerait de bloquer longtemps certains tests en l'attente de la correction des anomalies détectées en amont.

Le délai nécessaire à la convergence des tests est aléatoire : on ne peut pas évaluer a priori la difficulté des corrections. Il faudra gérer la crise entre client et fournisseur quand ce délai semble s'allonger indéfiniment : cette crise peut aboutir soit (finalement) à une livraison de qualité acceptable, soit au refus du produit.

Certaines des fiches d'anomalie seront interprétées par le fournisseur comme des demandes d'évolution et il demandera pour les corriger une rallonge au contrat. Il en résultera alors des négociations entre client et fournisseur.

Enfin la recette est toujours pour le client un moment délicat, puisque le produit qu'il découvre résulte à la fois des spécifications qu'il a fournies et de la réalisation bâtie par le fournisseur sur la base de ces spécifications. ♦

Nous contacter :

contact@conseilorga.com

1

C'est l'élément le plus important et qui consiste à organiser les tests. Une campagne de test décrit quels sont les scénarios qui seront testés et donc enchaînés les uns aux autres. Dans le cadre de la recette fonctionnelle, il s'agira de décliner les scénarios à exécuter permettant par exemple de créer une situation applicative et ensuite dans un second scénario de faire subir des traitements applicatifs à ces données, pour enfin vérifier les résultats obtenus. Ce sont les règles de gestion qui font référence et constituent le résultat attendu. L'exécution de la fonction dans l'application permettra de constituer le résultat obtenu.

C'est un composant du scénario, et représente une variante du scénario. Par exemple pour le test d'un changement d'adresse, le cas numéro un prévoira un changement d'adresse en France, alors que le cas de test 2 prévoira un changement d'adresse dans un pays étranger.

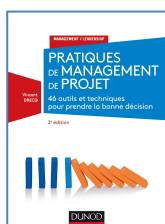
C'est un jeu de données, inscrit dans un document dans lequel figurera de manière précise quelles sont les données qui seront saisies dans l'application. Ce document inclura aussi le résultat attendu au niveau des données rendues par le système, et ce en fonction des règles de gestion.

C'est un document qui précise ce que sera le test, quelles fonctionnalités de l'application seront testées en précisant leur ordre ainsi que leur enchaînement. Le plan de test décrit quelles sont les règles de gestion du dossier de conception qui seront testées.

C'est l'ensemble des actions qui permettent de s'assurer que les développements/paramétrages réalisés lors du projet répondent convenablement aux besoins exprimés et fonctionnent correctement au sein de l'environnement cible.

C'est le concept qui précise la fonction à tester, ce document est lié à un document de jeux de données de test. La description y est précise et se réfère pour la recette fonctionnelle aux dossiers de conception. Un scénario de test est composé de un ou plusieurs cas de tests.

Cette fiche thématique est gratuite, n'hésitez pas à la diffuser largement à votre entourage.



Des techniques concrètes pour le management de projet !

PRESTATIONS DE CONSEILORGA



CONSEIL EN GESTION DE PORTEFEUILLE PROJET

—

Aider à faire des choix dans l'ensemble de vos projets afin qu'ils délivrent les bénéfices attendus.



CONSEILS & MÉTHODES PROJET

—

Améliorer l'efficacité de la mise en œuvre des projets.



FORMATION

—

Former des directeurs de projet et chefs de projets et des candidats à la certification PMP®.



COACHING OPÉRATIONNEL

—

Accompagner les dirigeants dans leur prise de décisions et les pilotes de projets dans leur organisation.

POINTS FORTS



ADAPTABILITÉ

Nos interventions s'adaptent à votre contexte et votre environnement



ACCOMPAGNEMENT

Vous avez la garantie d'un accompagnement sur mesure, d'une expertise forte procurant une valeur ajoutée élevée



EXPERTISE

En s'appuyant sur des méthodes agiles, nous mettons en œuvre un grand nombre d'outils et techniques appropriés au contexte de nos clients



PÉRENNITÉ

Nous facilitons la montée en compétences des collaborateurs par une transmission de notre savoir de manière pédagogique



EFFICACITÉ

Nous apportons des résultats ciblés par une appropriation rapide des problématiques de l'entreprise



SITUATIONS COMPLEXES

Nous vous aidons à définir un cadre, un périmètre, des scénarios dans des environnements en profonde transformation avec une forte incertitude.



HUMAIN

Nous avons une capacité à interagir avec tout type de personne, nous respectons le rythme de chacun et possédons une capacité d'engagement pour impliquer les acteurs