

ECRIT PAR
Nicolas PEGUIN

DATE
Version révisée
Septembre 2020

SÉRIE : HOW TO

SAUVETAGE DE PROJET

Un projet en panne ? Un projet mal accepté par les utilisateurs ? Un planning qui dérape trop ? Les raisons du - ou des – problèmes peuvent être nombreuses : erreurs de conception, prestataire démissionnaire ou incompetent, imprécisions sur le besoin...

NOTES DE LA 2^{ème} VERSION :

Vous trouverez dans ce livre blanc quelques solutions que nous employons pour sauver les projets de nos clients !

SOMMAIRE

INTRODUCTION	4
1. POSER UN DIAGNOSTIC.....	6
1.1 Les dangers qui vous guettent !	6
1.2 Quelques risques en détail	7
1.3 Poser un diagnostic	9
2. METHODES DE SAUVETAGE.....	10
Étape 1 – D’abord, l’étape la pire et la plus simple... parfois il faut avoir le courage de supprimer le problème	10
Étape 2 – Ensuite gérer les problèmes périphériques : Quelles solutions pour gérer les problèmes humains ?	10
Étape 3 – Enfin, solutionner les problèmes techniques	12
Cas particulier – Les problématiques de performance	13
3 QUELQUES CAS PRATIQUES.....	14
Cas 1 – Une mauvaise architecture.....	14
Cas 2 – Un choix technique inadapté.....	15
Cas 3 – Abandon du projet par le prestataire	16
Cas 4 – Un périmètre à la dérive.....	17
CONCLUSION	18
QUELQUES CONSEILS POUR FINIR...	19

INTRODUCTION

Un sujet passionnant !

TheCodingMachine a souvent eu l'occasion d'aider des clients en difficulté sur leur projet. Ces situations critiques nous ont permis d'appréhender les origines de ces dérapages, de trouver des solutions et de les mettre en œuvre.

Le "sauvetage de projet" est un sujet passionnant à plusieurs titres :

- Il nécessite de **mobiliser une très large palette de compétences** : des compétences humaines pour distinguer objectivement les problèmes réels de ceux qui sont fantasmés et des compétences techniques afin de pouvoir plonger au cœur des applications, sans oublier la créativité pour imaginer ou trouver les bonnes solutions.
- C'est un **défi** : les attentes des clients sont fortes. Ces missions nécessitent de déployer une énergie importante au démarrage afin de convaincre les différentes parties et mettre en place les bonnes actions correctrices.

Différence entre « risque » et « échec »

Un risque est susceptible d'être géré. Il est possible de mener des actions préventives ou correctives. Un échec est souvent une accumulation de problèmes qui indépendamment les uns des autres pourraient trouver une solution. Autrement dit, de risques non gérés qui aboutissent souvent à un blocage global du projet. La différence entre les deux n'est pas évidente. La tendance naturelle est de mener des actions pour tenter de corriger les problèmes qui se présentent au fur et à mesure du temps. Parfois, en ayant la tête dans un projet, il n'est pas rare de perdre en lucidité. Un problème qui peut sembler très grave ne l'est peut-être pas et inversement. Alors, si vous vous posez la question de savoir si votre projet est raté ou si vous êtes simplement en train de gérer un risque, vous êtes certainement à une étape cruciale du développement de votre projet.

S'inquiéter est une bonne maladie. Car plus on se rend compte tard des problèmes, plus c'est grave. Autrement dit, plus on accumule les risques au fur et à mesure du projet, plus le sauvetage s'annonce difficile !

Ce livre blanc est là pour vous livrer une synthèse de nos expériences, les écueils à éviter et vous exposer les méthodes que nous avons appliquées.

Si vous estimez que votre projet est sur de mauvais rails ou en pleine tourmente ou si, plus simplement, vous souhaitez approfondir la notion de risques sur un projet, ce livre blanc est fait pour vous.

Attention : Ce livre blanc est le fruit de notre expérience. Vous y trouverez peut-être des défauts ou en soulèverez des limites. N'hésitez pas à nous en faire part. D'autre part, nous sommes

convaincus que nous n'avons toujours pas fait le tour du sujet, vous avez le droit de faire preuve de créativité !

1. POSER UN DIAGNOSTIC

Votre projet évolue dans un environnement hostile, il faut d'abord poser un diagnostic précis

1.1 Les dangers qui vous guettent !

Réussir le sauvetage d'un projet nécessite de plonger au cœur du problème afin de dresser un tableau objectif de son état. Quels sont les problèmes ? Peuvent-ils être résolus rapidement ? Par qui et de quelle manière ?

Sur chaque facette (organisation, technique etc.) et à chaque étape, des dangers vous guettent mais pas de panique ! Il n'y a que ceux qui ne font rien qui ne prennent pas de risques. Quelques remarques en préalable :

- Les risques peuvent être endogènes ou exogènes : un mauvais suivi sera au moins en partie de la responsabilité du client donc un risque interne ; tandis qu'un prestataire incompetent sera un risque externe. Essayez d'être objectif sur la situation !
- Souvent les risques sont en cascade. Par exemple, le choix d'une mauvaise architecture technique lors du lancement de projet impliquera peut-être de mauvaises performances. Ainsi, l'étape de lancement c'est-à-dire la manière de structurer votre projet et le choix du prestataire constituent des étapes clés pour limiter les risques. Vous devrez y porter une attention particulière.

Passage en revue des 5 familles de risques :

BUSINESS		
LANCEMENT	DEVELOPPEMENT	PRODUCTION
<p>Recueil des besoins partiel ou non aboutis (ne pas consulter les utilisateurs finaux par exemple),</p> <p>Imaginer le projet parfait plutôt que le projet faisable.</p> <p>Rédiger un document (brief, cahier des charges...) incomplet ou trop complexe.</p>	<p>Peu d'implication des utilisateurs finaux (utilisateurs ou clients) dans le projet,</p>	<p>Fonctionnalités inadéquates et déceptions de vos utilisateurs/clients.</p> <p>UI/UX inadaptée.</p>

TECHNIQUE		
LANCEMENT	DEVELOPPEMENT	PRODUCTION
Mauvais choix techniques (pas adaptés aux besoins), Mauvaise architecture technique.	Accumulation d'une dette technique importante au fur et à mesure des développements, Bugs impossibles à résoudre dans un délai raisonnable, Nombre de bugs trop importants.	Bugs à répétitions, Nombre de régressions important, Instabilité de la production, Mauvaises performances techniques (temps de réponse).

GESTION DE PROJET		
LANCEMENT	DEVELOPPEMENT	PRODUCTION
Votre prestataire vous propose un prix irréaliste pour le projet (prix hameçon).	Mauvais/manque de suivi du projet, Non implication du management dans le projet, Désintérêt du prestataire pour le projet, Temps de développement trop long.	(Lorsque la gestion de projet est mauvaise, arriver jusqu'à ce stade est compliqué !)

RESSOURCES		
LANCEMENT	DEVELOPPEMENT	PRODUCTION
Manque d'expérience pour ce type de projets,	Compétences trop légères des équipes, Mauvaise montée en charge sur le projet.	Risque important de dette technique.

En Annexe : quelques conseils de bon sens pour maîtriser ces risques lors de votre projet (et d'éviter la mise en place d'un plan de sauvetage).

1.2 Quelques risques en détail

LE PERIMETRE A LA DERIVE

Symptôme très fréquent : dès l'origine du projet, un périmètre très vaste est défini ou des besoins de plus en plus complexes sont exprimés au fur et à mesure du projet.

C'est un cas qui se produit souvent lorsque le client souhaite lancer une nouvelle activité. Proposer de nombreuses fonctionnalités donne l'impression rassurante d'offrir un service plus complet au client final donc plus « vendable ». Dans la réalité, c'est souvent l'inverse. En général, il vaut mieux définir le périmètre le plus petit possible, tester rapidement et redéfinir le besoin en utilisant les retours utilisateurs ou bêta testeurs ("Pave the cow path").

Définir un périmètre très vaste donne aussi l'impression trompeuse de coûter moins cher qu'un développement en plusieurs étapes. C'est faux car si de nombreuses fonctionnalités ne servent pas, le coût de ces développements inutiles et de l'effort nécessaire pour les produire peuvent mener le projet vers l'échec.

La dérive d'un périmètre peut aussi être attribuée à une mauvaise gestion de projet. Un prestataire en mauvaise position, comme par exemple dans le cas d'un retard important ou d'un problème de ressources, peut accepter des développements complémentaires en espérant conserver de bonnes relations avec son client. Il lui rend un mauvais service. En cumulant retard et nouveaux développements, le projet risque d'être encore plus long à finaliser.

LE PRIX HAMEÇON

Le prix proposé par le prestataire est volontairement bas afin d'« hameçonner » le client. Un prestataire minimise volontairement le prix de la prestation afin de remporter le marché. Au fil du projet, deux situations se présenteront :

- Les développements continuent car vous acceptez de nombreux avenants,
- Le projet est bloqué car vous ne souhaitez pas (ou ne pouvez pas) continuer à payer.

Ce "faux prix" risque de fortement dégrader la relation que vous entretenez avec votre prestataire à mesure que le projet avance.

Il est cependant nécessaire de faire la différence entre un acte délibéré (un prestataire qui hameçonne un client) et un périmètre qui évolue et grandit anormalement en cours de route.

PROJETER SES ANGOISSES (OU FAIRE DES SUPPOSITIONS)

Ce n'est pas à proprement parler un risque. La méconnaissance des projets peut générer des angoisses irrationnelles. Il nous est déjà arrivé d'intervenir sur un projet qui allait plutôt bien ! Il est donc important de mettre en œuvre, dès le démarrage, un dispositif de gouvernance permettant de mesurer l'avancement, gérer les risques et d'arbitrer les évolutions. Tous les dispositifs qui permettront aux parties prenantes d'être à l'aise avec le projet et son avancement.

Ces mesures : temps consommé, avancement etc. doivent permettre de fournir à l'ensemble de l'équipe un avis objectif sur l'état de votre projet. Ils permettent de mettre en œuvre des plans d'actions pour gérer certains risques ou replanifier certaines parties du projet.

L'EFFET TUNNEL

L'effet tunnel consiste à développer pendant une longue période sans faire intervenir les utilisateurs. La solution développée risque, in fine, de ne pas convenir aux besoins/souhaits des utilisateurs. La recette, longue et donc fastidieuse, risque alors d'être bâclée, les utilisateurs ne disposant pas forcément du temps nécessaire pour la faire aboutir.

LE NOMBRE D'ANOMALIES

Voilà une question qui revient souvent : est-ce qu'un nombre important d'anomalies durant la phase de recette indique qu'un projet est en train de se planter ?

Dans la plupart des cas, non ! Il faut tordre le cou à cette idée reçue. Plus le nombre d'anomalies est important, plus vos utilisateurs testent la solution. C'est lorsque la solution n'est pas assez testée que l'on peut rencontrer des difficultés en production. Donc, de manière contre-intuitive, cet indicateur est plutôt un facteur de qualité.

La première recette technique doit permettre de régler la plupart des anomalies de base. Si cela n'est pas le cas, un recadrage peut s'avérer nécessaire pour éviter l'épuisement des utilisateurs en cours de recette.

L'autre nuance que l'on peut apporter est de déterminer si les corrections sont efficaces. Des problèmes liés aux performances qui ne peuvent être corrigés indiquent peut-être des problèmes importants liés à la conception.

1.3 Poser un diagnostic

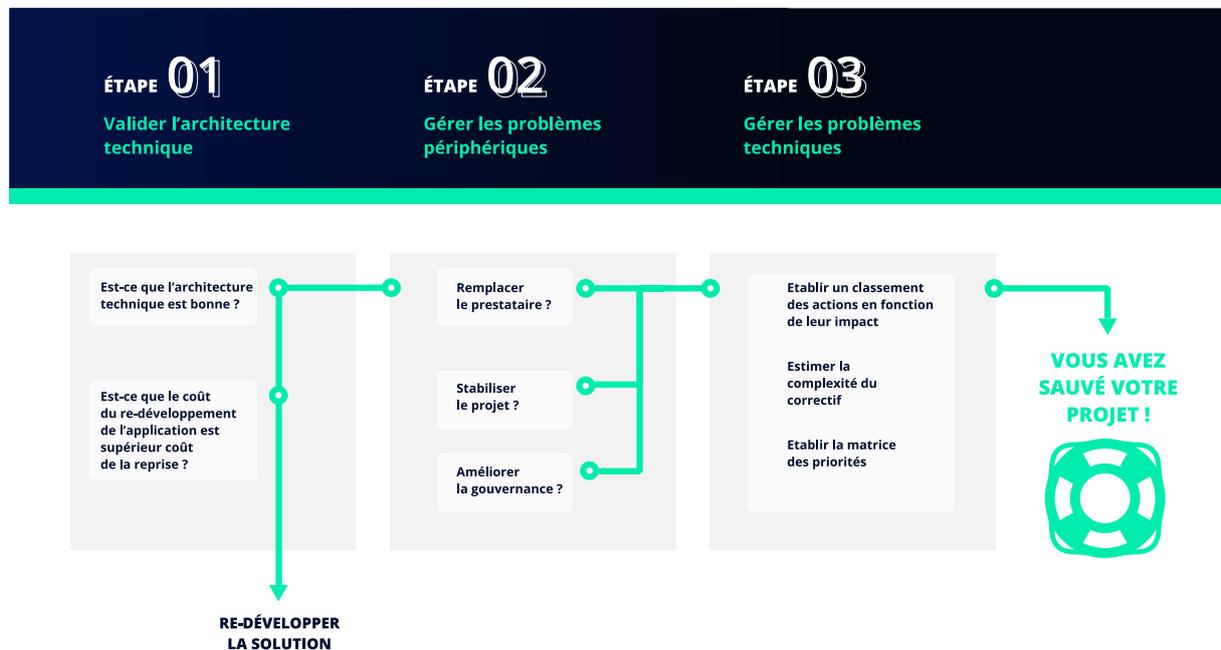
Les solutions dépendent étroitement de l'état du projet : le projet est-il encore en développement ou bien en production. Elles dépendent aussi de la nature des problèmes : est-ce un problème technique ? Un problème concernant le périmètre ? Une relation qui s'est dégradée avec votre prestataire ?

Pour garantir un diagnostic dépassionné, solliciter une société extérieure est souvent une bonne option. En effet, de nombreux problèmes périphériques qui accompagnent inévitablement la dérive d'un projet, parasitent notre vision. Faire appel à un tiers qui est indépendant et sans a priori permet de rationaliser le débat.

Poser ce diagnostic permet de résoudre la majorité des problèmes, car une fois identifiés, il est possible de mettre en place un plan d'actions efficace.

2. METHODES DE SAUVETAGE

Vous avez établi le diagnostic et votre projet risque de ne pas aboutir du tout. Pas de panique, TheCodingMachine vous apporte des solutions !



Étape 1 – D’abord, l’étape la pire et la plus simple... parfois il faut avoir le courage de supprimer le problème

Si les problèmes ont pour origine une mauvaise (voire très mauvaise) architecture technique, l'application développée a très peu de chance d'être "sauvée". Corriger ce type d'erreurs est souvent aussi coûteux que le redéveloppement complet de l'application.

L'analyse de l'existant doit donc être méticuleuse pour éviter de jeter une application qui pourrait être sauvée, mais également pour ne pas vouloir sauver cette application à tout prix et faire durer l'échec.

Ce que l'on peut récupérer est souvent une partie de la conception fonctionnelle : processus ou écran par exemple.

Étape 2 – Ensuite gérer les problèmes périphériques : Quelles solutions pour gérer les problèmes humains ?

Il faut tout d'abord établir un constat d'échec. Cela semble évident aux chefs de projets dont le prestataire est démissionnaire. Pour celui qui aura un prestataire qui tente de le rassurer, lui

promet d'aboutir et qui tente de garder une bonne relation, établir ce constat sera en revanche plus difficile.

Si l'application mérite d'être sauvée, il est nécessaire de s'attaquer aux problèmes humains. C'est souvent délicat car les sentiments qui agitent les parties sont rarement bienveillants. Il est toujours douloureux de se tromper ou d'avoir le sentiment de s'être fait avoir par un prestataire peu scrupuleux. Se lamenter n'est pourtant pas une solution.

Impossible, malheureusement, d'indiquer ici une solution universelle. Nous vous recommandons bon sens et pragmatisme : êtes-vous victime de votre prestataire ou bien de vous-même ? Est-ce que changer le management du projet va vous permettre de porter un nouveau regard sur le projet, apporter une nouvelle motivation à l'équipe ? Est-il possible de créer un électrochoc, de déclencher une prise de conscience de la part des équipes ?

Dans ce domaine, la créativité n'a pas de limites. Voici une présentation des pistes les plus évidentes et les plus efficaces :

- **Le désengagement du prestataire** - et son éventuel remplacement, permet souvent de voir le projet avec un œil neuf afin d'évacuer les problèmes récurrents : Ce désengagement est-il souhaitable, possible ? Quels coûts supplémentaires va-t-il engendrer ? Quels bénéfices va-t-il rapporter ? Comment envisager la phase de transition ?
- **Changer les équipes peut redonner un second souffle au projet.** Pour cela il est nécessaire que l'image du projet ne soit pas trop dégradée afin de ne pas décourager les nouveaux collaborateurs : Ce changement est-il souhaitable, possible ? Quelle organisation mettre en place, qui conserver ?
- **Faire une pause, pour tenter de stabiliser le projet** et se remettre les idées en place pour se réorganiser : Cette pause est-elle souhaitable, possible ? Combien de temps (trop risquerait de mettre fin au projet) ? Qui et comment planifier la suite du projet ? Relancer les développements par étape, morceler les problèmes, permet d'en voir le bout et ainsi de se motiver : Peut-être est-il possible de découper le projet en sous-ensembles ? Le coût de cette réorganisation est-il supportable ? Les équipes sont-elles prêtes à ce changement ?

Parfois, le problème peut être interne. Certaines équipes souffrent du syndrome « NIH » (not invented here). A cette étape, il faut aussi se poser la question.

Étape 3 – Enfin, solutionner les problèmes techniques

Si le projet peut être sauvé, alors allons-y, n'attendons plus ! Ce qui n'empêche pas d'intégrer méthode et rigueur à notre opération de sauvetage.

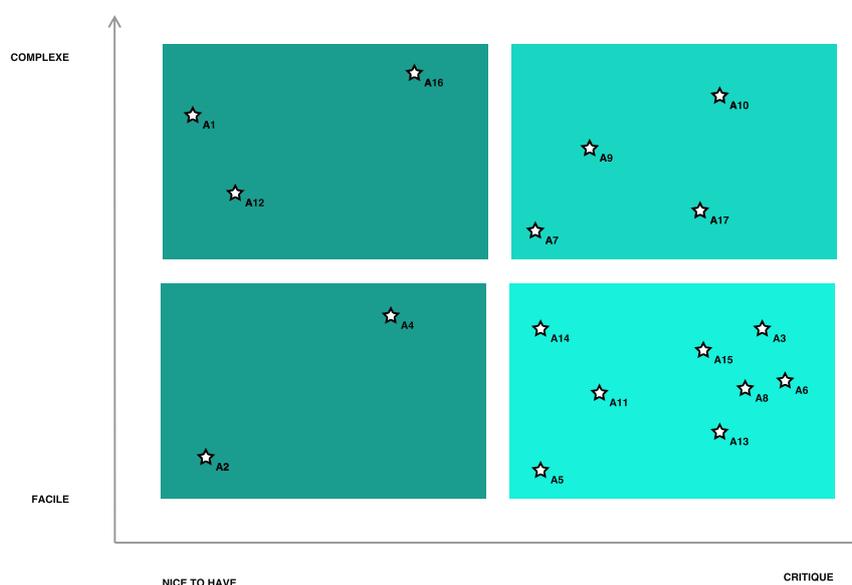
Pour identifier les risques et les actions associées à la reprise de votre projet, plusieurs techniques sont envisageables. Nous vous en proposons une qui a le mérite d'être à la fois simple à partager et facile à appliquer :

Classement des risques et actions :

- Impact : le coût associé à la survenance du risque. Par exemple, la chute de la base de données principale a un coût supérieur à la chute d'un serveur de mail
- Probabilité : la probabilité de survenance du problème ;
- Effort de correction : le coût associé à la mise en place d'un correctif ;
- Catégorie : la catégorie associée au risque (architecture, sécurité du code, ...)

Description du risque et de l'action associée	
Impact	Faible/Moyen/Fort : description de l'impact
Probabilité	Faible/Moyen/Fort : probabilité de survenance de l'événement
Effort de correction	Faible/Moyen/Fort : description des tâches nécessaires pour corriger le problème

Les actions sont ensuite placées sur un quadrant, avec en abscisse : impact * probabilité, et en ordonnée la complexité de mise en place du correctif.



Les actions du quadrant en bas à droite sont évidemment les actions prioritaires.

Cas particulier – Les problématiques de performance

Les problèmes de performances passent souvent à la trappe dans les projets. Ils apparaissent souvent en fin de mission s'ils n'ont pas été anticipés. Par ailleurs, ils demeurent difficilement détectables avant la mise en production du projet. Ce n'est pas nécessairement très grave si l'architecture technique est bonne. Beaucoup de solutions d'optimisation existent !

TheCodingMachine a consacré un Livre Blanc à ce sujet. Nous y avons indiqué comment analyser et traiter ces problèmes particuliers.

Vous pouvez télécharger le document à cette adresse :

<http://www.thecodingmachine.com/fr/hauts-performances-web>

3 QUELQUES CAS PRATIQUES

Cas 1 – Une mauvaise architecture

LE PROJET :

Dans le cadre de la dématérialisation de ses correspondances avec ses clients, cette filiale d'un acteur majeur de la banque-assurance a demandé à un prestataire de créer un système de gestion électronique de documents (ou GED). Le projet comprenait l'externalisation de l'ouverture et de la numérisation du courrier, puis la distribution du courrier sous forme dématérialisée à travers une application qui gérait les workflows de demandes client.

NOTRE INTERVENTION :

Devant les retards accumulés et les dysfonctionnements constatés, TheCodingMachine a été mandatée pour conduire un audit, analyser finement l'application et proposer un plan de sauvetage.

LES PROBLEMES DETECTES :

L'architecture de la solution avait été très mal pensée. Les données étaient distribuées dans deux bases de données distinctes. La solution reposait sur un logiciel open-source que le prestataire ne maîtrisait pas. Les temps de réponse étaient parfois difficiles à supporter pour les utilisateurs et rendaient même parfois l'application inutilisable (avec des temps de réponse parfois supérieurs à 5mn). L'application était instable, avec parfois des arrêts de production de plusieurs jours.

LES SOLUTIONS APORTEES :

Le processus de build et l'utilisation systématique de caches et sessions ont permis de consolider l'application sans remettre en cause les fondements de l'outil.

L'AVIS DE TCM :

L'industrialisation des développements, la mise en place de bonnes pratiques ainsi que le lancement de tests automatisés pour garantir les non régressions assurent la stabilité de l'application dans la durée.

Cependant, nous nous demandons toujours s'il n'aurait pas mieux valu effectuer une refonte complète !

Cas 2 – Un choix technique inadapté

LE PROJET :

Le client, dans le domaine de l'immobilier, souhaitait proposer à ses collaborateurs un Intranet : présentation des informations, gestion des formations, reporting etc.

NOTRE INTERVENTION :

En rupture avec son ancien prestataire, le client devait montrer que son projet n'était pas en échec. Une partie de la conception a donc été reprise pour conserver le schéma de la base de données relatif à la gestion des utilisateurs notamment.

LES PROBLEMES DETECTES :

La technologie était totalement inadaptée aux besoins, en raison de l'inexpérience du prestataire sur ce type de projet, plus habitué aux sites vitrines qu'aux d'Intranets.

LES SOLUTIONS APORTEES :

Refonte du projet : l'intégralité de la structure du code a été revue : framework MVC, Base de données etc.

L'AVIS DE TCM :

Vouloir sauver son projet à tout prix est parfois une erreur qui coûte cher !

Cas 3 – Abandon du projet par le prestataire

LE PROJET :

Grande franchise du secteur de l'esthétique, le client souhaitait proposer la vente de ses services en ligne. C'est une agence marketing qui avait remporté le projet, qu'elle sous-traitait, pour la partie technique, à une SSII installée en Tunisie.

Compte tenu du manque de marge de manœuvre en ce qui concerne le budget, le projet avait été abandonné par le prestataire. TheCodingMachine a été sollicité par l'agence pour reprendre ce projet.

NOTRE INTERVENTION :

Pour respecter les délais, priorité très forte du client, nous avons conduit le sauvetage du projet en deux temps :

- nous avons épuré le périmètre pour mettre en pré-production une version testable par le client et lui permettre ainsi d'avoir une visibilité sur les avancées du projet,
- en parallèle, nous avons finalisé le développement du périmètre initial.

LES PROBLEMES DETECTES :

Manque de compétences des équipes de développement.

LES SOLUTIONS APORTEES :

L'architecture initiale – utilisation de Magento – et le projet dans son ensemble ont été conservés. En revanche, nous avons redéveloppé et reconfiguré des modules plus adaptés : multi-boutique, thème, modules complémentaires...

L'AVIS DE TCM :

Effectuer un sauvetage de projet pour le compte d'un tiers, et non le client final, est extrêmement complexe. La communication et le ressenti client est primordial pour ce type de sauvetage. Il faut se positionner avec le client pour le rassurer au mieux, et ne pas partir avec une agence qui dénigre la technique ! (Nous aussi on commet des erreurs).

Cas 4 – Un périmètre à la dérive

LE PROJET :

Cette société de vente par correspondance et à domicile avait décidé de refondre son système d'informations (approvisionnement, vente, facturation, etc.). Le choix d'un prestataire off-shore et d'une organisation complexe (pas de communication directe, passage par un tiers, etc.) avait conduit le client à utiliser en production une application à moitié finie.

NOTRE INTERVENTION :

Nous avons commencé par un audit technique de la « solution » existante (bien que peu de fonctionnalités aient été abouties). Puis nous avons décidé, conjointement avec le client, d'une nouvelle organisation.

PROBLEMES DETECTES :

Deux problèmes majeurs ont été détectés :

1. Aucune spécification précise des fonctionnalités n'existait, les développements avaient tous été menés au fil de l'eau à partir d'un document général.
2. Quand les relations sont devenues tendues entre le client et le prestataire, les développements ont été accélérés au détriment de la qualité technique et des règles de codage (utilisation des outils installés).

LES SOLUTIONS APPORTEES :

Nous sommes repartis du point le plus avancé des développements afin de récupérer les fonctionnalités développées.

Nous avons ensuite réorganisé les développements en écrivant les spécifications manquantes, réduisant la dette technique à chaque fois que c'était possible tout en estimant puis en priorisant au préalable chaque tâche.

L'AVIS DE TCM :

Il ne faut jamais abandonner les règles techniques de base pour gagner du temps. Cela s'avère souvent contre-productif pour la bonne conduite d'un projet. Le projet en serait, au mieux, trop compliqué à maintenir, et dans le pire des scénarios, impossible à faire aboutir.

CONCLUSION

Notre expérience nous a montré que la plupart des problèmes peuvent trouver une solution technique même si elle est parfois coûteuse. Le point le plus délicat à gérer se situe bien souvent dans les relations entre les parties prenantes. Des relations conflictuelles induites par des échecs accumulés sur un projet s'avèrent parfois difficiles à pacifier. Il est pourtant important de dépassionner les débats pour permettre à des décisions réfléchies et rationnelles d'émerger, motivées par la réussite finale du projet.

Évidemment, de nombreux éléments mériteraient d'être approfondis. N'hésitez pas à nous contacter si vous souhaitez en discuter.

QUELQUES CONSEILS POUR FINIR...

Vous avez tout lu ! On ne va pas se quitter comme cela. Voilà quelques règles qui vous permettront d'éviter un sauvetage de projet (autrement dit : pour gérer votre projet au mieux) :

NE PAS AVOIR LES YEUX PLUS GROS QUE LE VENTRE

Vouloir trop de fonctionnalités est parfois risqué. Commencez petit, avec l'indispensable : plus vous élargirez le périmètre plus le projet sera difficile à gérer.

Le lotissement de la solution doit être logique et compréhensible. Lotir ne veut pas dire construire des briques non fonctionnelles.

CHERCHER A TOUT COMPRENDRE

Si vous organisez une consultation et que vous n'avez pas de compétences techniques, laissez l'opportunité au prestataire de concevoir la solution technique la plus adaptée et de répondre avec les technologies qu'il maîtrise le mieux. Cependant, ne vous désintéressez pas de la technique. Ne pas maîtriser cet aspect est très dangereux.

Et, ne laissez pas démarrer un projet sans avoir validé et donc parfaitement compris les spécifications fonctionnelles.

ETRE RIGoureux SUR L'AVANCEMENT

Des points d'avancement doivent être réalisés pour vous assurer de la bonne tenue du projet. Tout retard doit avoir une explication rationnelle ! L'information doit être transmise de façon régulière, les bonnes nouvelles comme les moins bonnes. Ces points doivent détailler les risques et la manière de les gérer.

PROCÉDER PAR ETAPE

Une période de développement trop longue sans implication des utilisateurs est problématique. Il est nécessaire de parer ce risque en prévoyant des livraisons intermédiaires.

S'IMPLIQUER DANS LES TESTS

Impliquez-vous dans la recette de votre projet. Réalisez des cahiers de tests, créez des jeux de données pour tester les éléments clés de votre projet.

Le dimensionnement et les environnements doivent permettre une utilisation optimale de l'application (gestion des serveurs, de l'infrastructure réseau, etc.).

Il vaut mieux séparer les développements de la production (partie du travail qui consiste à mettre en place les serveurs, les environnements, gérer les sauvegardes etc.).

(Et une dernière :) L'AGILE N'EST PAS MAGIQUE !

Croire que mettre en place une méthodologie agile vous évitera d'avoir à fournir ces efforts n'est pas du tout une bonne idée !

EN COMPLÉMENT / PROCHAINEMENT



N'hésitez pas à télécharger notre livre blanc **MODÈLE DE CAHIER DES CHARGES PLATEFORMES WEB** qui vous donnera les clés pour comprendre tous les aspects d'un projet digital.



Et, notre série **COMPRENDRE LA TECH** va se poursuivre, avec en cours de rédaction : **COMPRENDRE LES TENDANCES TECHNIQUES 2021**

Si vous voulez recevoir ces documents, n'hésitez pas à nous écrire un email contact@thecodingmachine.com ou via les réseaux sociaux



MERCI !

Si vous avez une idée, un nouveau projet ou même
un projet à rattraper
passez nous voir pour en discuter !

TheCodingMachine
56 rue de Londres – 75008 – Paris
01.85.08.34.98