

[Livre blanc]



SERIOUS GAME : PRODUIRE AUTREMENT ?

BEATRICE LHUILLIER

Les méthodes traditionnelles de gestion de projet ont initialement été transposées du monde du bâtiment à l'industrie informatique. Sécurisantes mais rigides, ces méthodes ne sont pas adaptées pour tout type de projet. Bonne nouvelle : il est aujourd'hui possible de produire des serious games autrement, toujours dans le respect des budgets, plannings et délais dans un esprit de collaboration, de co-construction et d'ajustements successifs plus en phase avec les desiderata des clients et des équipes de production.

SOMMAIRE

Pourquoi ce livre blanc ?	3
Avertissement.....	6
Quelques definitions.....	7
Rôles du chef de projet	8
Serious game : de quoi parle-t-on ?	11
A qui s'adresser pour realiser un serious game ?	14
Méthodes prédictives	17
Méthodes agiles	21
Quelques conseils pour finir... ..	29
Annexes	33
Bibliographie.....	35
Présentation de l'auteure	36

POURQUOI CE LIVRE BLANC ?

Le développement de serious games fait couler beaucoup d'encre. Pour des raisons de coûts, ces nouveaux outils de communication ou de formation étaient jusqu'alors plutôt l'apanage des grandes entreprises qui étaient les seules à pouvoir débloquer les budgets nécessaires à leur développement. Cette tendance commence à s'inverser : les éditeurs peuvent aujourd'hui capitaliser sur leurs premières réalisations voire s'appuyer sur des « logiciels auteurs » pour faire des économies d'échelle et proposer sur le marché des serious games pour quelques dizaines de milliers d'euros (contre plusieurs centaines il y a encore deux ans...). Vous envisagez de lancer un projet de réalisation d'un serious game ? Mais savez-vous concrètement comment se passe la gestion et la production de ce type de projet ? Nombreux sont les éditeurs qui utilisent les méthodes dites « traditionnelles » de gestion de projet. Mais, ces méthodes sont-elles adaptées et efficaces pour tout type de projet serious game?

En France

Nous ne disposons pas d'étude sérieuse sur le nombre de projets informatiques menés chaque année en France et leur taux de réussite. Tout juste pouvons-nous nous appuyer sur une estimation du chiffre d'affaires généré par l'activité de production de serious games : moins de 4,5 millions d'euros¹ (3% des 144 millions d'euros de chiffre d'affaires engrangés par les acteurs e-learning) d'après la dernière étude du Cabinet Fefaur sur le sujet.

Notons que ce chiffre ne prend pas en compte TOUS les serious games produits mais uniquement ceux qui le sont par des éditeurs de e-learning. Or, comme nous le

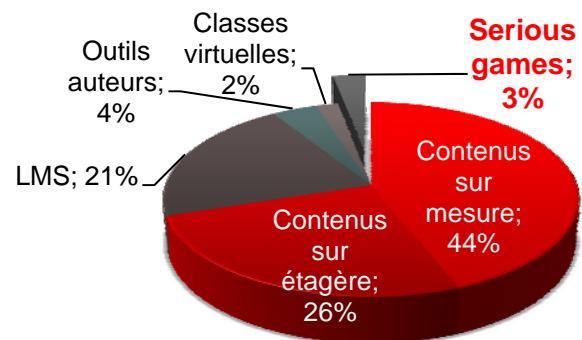


Figure 1 - Chiffre d'affaires 2009 du e-learning par segment de marché

verrons, il existe quantité de sociétés éditrices de serious games qui ne sont pas positionnées sur le créneau de la formation : celles qui développent notamment des serious games à usage de communication.

¹ L'offre professionnelle e-learning en France – Fefaur - 2010

Aux Etats-Unis

Appuyons-nous sur les résultats de nos confrères américains qui, eux, se penchent sérieusement sur la question depuis de nombreuses années. Ainsi, aux Etats-Unis le Standish Group² mène une enquête depuis 1994 sur le taux de réussite des projets informatiques menés sur le marché américain. En 2009, cette étude a fait apparaître que seulement 32% des projets pouvaient être considérés comme réussis, c'est-à-dire livrés à temps, dans le respect du budget et intégrant toutes les fonctionnalités prévues initialement. Plus de trois projets sur quatre sont des échecs complets ou partiels ; ils sont abandonnés ou finissent par aboutir mais au prix de dépassements budgétaires conséquents et/ou de décalages importants du planning.

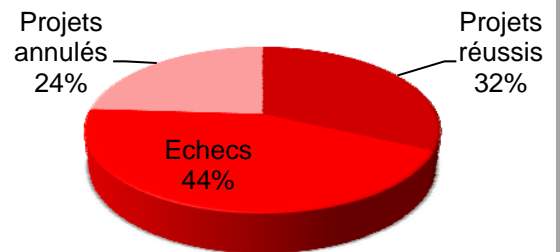


Figure 2 - Taux de réussite des projets.
Source : Standish Group

D'après Jim Johnson, Président du Standish Group :

« 44% des projets font état de retard, sont hors budget et/ou sont livrés avec moins de fonctionnalités que prévu initialement et 24% sont annulés car incomplets ou livrés mais jamais utilisés »

"44% were challenged which are late, over budget, and/or with less than the required features and functions and 24% failed which are cancelled prior to completion or delivered and never used."

Des méthodes remises en question

Les méthodes dites traditionnelles de développement sont aujourd'hui encore très utilisées mais remises en question autant par les éditeurs que par leurs clients. Elles sont fondées sur un principe linéaire de construction, de la rédaction de spécifications à la livraison et présentent des limites aujourd'hui bien identifiées :

- ▶ Il est difficile de spécifier complètement une application en début de projet sans risquer d'oublier certains détails qui peuvent avoir des conséquences importantes sur sa volumétrie
- ▶ Les besoins du client peuvent évoluer au cours du projet, parce que le contexte des projets est mouvant, que des événements internes à l'organisation peuvent survenir ou tout simplement parce qu'au moment des premières livraisons, le

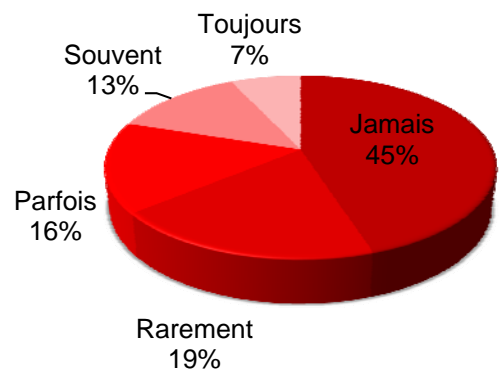


Figure 3 - Pourcentage de fonctionnalités implémentées réellement utilisées

² Standish Reports : www.standishgroup.com

client détecte des carences, des oublis ou au contraire des fonctionnalités redondantes ou finalement peu utiles.

A ce sujet, notons un autre chiffre intéressant³ (J.Johnson 2002) : plus de la moitié des fonctionnalités développées et livrées dans le cadre d'un projet informatique ne sont finalement jamais ou rarement utilisées par les utilisateurs finaux...

Tous ces constats questionnent la pertinence et l'efficacité des méthodes de gestion de projet et de développement des outils informatiques, et notamment des serious games...

³ Source : J.Johnson, Keynote speech, XP2002 (Sardaigne, Italie)

AVERTISSEMENT

Ce document ne prétend pas à l'exhaustivité quant à la présentation des méthodologies de travail utilisées aujourd'hui par les éditeurs de serious games. Chacun invente ses process, ses outils, ses documents en fonction de son historique, de son organisation, de la constitution même de ses équipes.

D'autre part, expliciter dans le détail chaque méthode une à une reviendrait à rédiger un ouvrage complet sur la question et tel n'est pas le propos. Il existe quantité d'ouvrages spécialisées sur les méthodes décrites dans le présent livre blanc. Si vous souhaitez creuser le sujet, vous trouverez plusieurs références dans la bibliographie à la fin de ce document.

Enfin, il convient de préciser qu'il n'existe pas de « bonnes » et de « mauvaises » méthodologies de développement. En revanche, certaines sont plus adaptées que d'autres selon la typologie de projet à mener...

QUELQUES DEFINITIONS

Projet

Selon le Project Management Institute⁴, organisation internationale de standardisation du management de projet :

Un projet est une entreprise temporaire décidée dans le but de créer un produit, un service ou un résultat unique.

Chef de projet

Le chef de projet (ou chargé de projet) mène un projet et contrôle son bon déroulement. De manière générale, il dirige ou anime une équipe pendant la durée du ou des projets dont il a la charge.

Serious game

Selon Julian Alvarez, chercheur français, docteur en sciences de l'Information et de la Communication, un serious game⁵ est :

Une application informatique, dont l'intention initiale est de combiner, avec cohérence, à la fois des aspects sérieux (Serious) tels, de manière non exhaustive et non exclusive, l'enseignement, l'apprentissage, la communication, ou encore l'information, avec des ressorts ludiques issus du jeu vidéo (Game). Une telle association, qui s'opère par l'implémentation d'un "scénario utilitaire", qui sur le plan informatique correspondrait à implémenter un habillage (sonore et graphique), une histoire et des règles idoines, a donc pour but de s'écarter du simple divertissement



⁴ <http://www.pmi.org> et <http://pmi-fr.org>

⁵ Introduction au serious game, Julian Alvarez, Damien Djaouti, 2010, éditions L>P Questions théoriques

ROLES DU CHEF DE PROJET

Certains l'oublient parfois tant cela peut ressembler à une lapalissade, mais l'objectif premier du chef de projet est...de mener le projet à son terme. En ce sens, il a une obligation de résultat. Quelle que soit la méthode de gestion de projet adoptée ce résultat se mesure toujours selon 3 axes : la **qualité**, le **budget** et le **délai**. C'est finalement sur ces 3 critères que le travail d'un chef de projet est jugé.



Mais, comme l'indique Véronique Messenger Rota dans son ouvrage *Gestion de projets - Vers les méthodes agiles*⁶, « *le périmètre des responsabilités du chef de projet est large mais variable* ».

En effet, selon le contexte, la typologie et/ou la volumétrie du projet, le chef de projet peut être amené à cumuler plusieurs fonctions. Ainsi, sur un « petit projet » e-learning, il est fréquent que le chef de projet se charge également de la conception pédagogique du contenu. Sur un projet ambitieux de serious game, la charge dévolue à la gestion de projet peut être telle qu'il est plus difficile de cumuler les rôles.

Concrètement, en quoi consiste le métier de chef de projet ?

Coordonner



Le chef de projet a en effet, en premier lieu, et quel que soit le type de projet concerné, un rôle de coordination. Tel un chef d'orchestre, il se doit d'organiser les actions des différents intervenants d'un projet. Dans le cadre d'un serious game, ce sera par exemple : les concepteurs pédagogiques et/ou serious-game designers, les développeurs, les graphistes etc. mais aussi les acteurs côté client.

Animer

Le chef de projet a un rôle d'animateur important. Souvent, il est en charge des aspects logistiques de l'organisation des réunions sur site ou à distance. Il doit créer un environnement favorable aux échanges et à la communication. Enfin, Il mobilise les intervenants, pratique l'écoute active, décèle les problèmes qui pourraient hypothéquer la



⁶ Véronique Rota Messenger, *Gestion de projet vers les méthodes agiles*, Eyrolles 2^{ème} éd, 2009, page 3 (MESSAGER ROTA 2009)

réussite du projet. Il a finalement un rôle de facilitateur, voire de coach d'une équipe dédiée à un objectif commun.

Planifier

Quelle que soit la méthodologie de gestion de projet adoptée, le chef de projet a toujours un rôle de planification et de suivi des plannings de réalisation. Parce qu'il est finalement « le maître du temps », il a un rôle important dans le séquençage et le tempo donné au projet. Or, cette tâche est rendue difficile par le fait qu'en matière de projets informatiques, la prévisibilité est limitée.



S'adapter



Qu'on se le dise une fois pour toutes : un projet ne se passe JAMAIS comme prévu ! Des aléas viennent systématiquement et régulièrement perturber son déroulement : C'est l'essence même du projet ! Le tout est de ne pas chercher à combattre cet état de fait, mais au contraire de faire face et de s'adapter aux changements. Le bon chef de projet n'est pas celui qui saura tout prévoir et tout anticiper (c'est tout simplement impossible), mais celui qui, en toutes circonstances, garde le cap et son calme. C'est aussi celui qui réagit positivement, en cherchant des solutions aux nouveaux problèmes qui se présentent à lui.

Suivre



Le chef de projet a un rôle important de suivi et de pilotage du projet. Et comme l'indique une fois encore Véronique Rota Messenger, « *un bon suivi de projet tient en la disponibilité d'informations fiables et à jour* ». En d'autres termes, plus le suivi sera rigoureux et les indicateurs pertinents, plus la prise de décision sera facilitée pour ramener le projet sur ses rails.

Tenir le budget

Cette tâche est liée à la précédente, de suivi du projet. Le chef de projet doit ainsi toujours être en mesure de répondre aux questions de sa hiérarchie ou du client : « qu'avons-nous dépensé ? », « que nous reste-t-il à consommer ? ».



Faciliter



Le rôle de facilitateur est souvent mal considéré par les chefs de projet juniors. Ils ont peur d'être abusés par l'équipe ou le client en confondant les notions de facilitation et de gentillesse. En réalité, un bon chef de projet est à l'écoute des autres et doit faire en sorte de faciliter le travail de chacun, toujours dans un esprit de coopération et pour le bien du projet.

Consolider

Nous l'avons vu précédemment, pour tenir correctement son rôle, le chef de projet doit à tout moment avoir une vue précise de son projet d'un point de vue du budget, du planning et de la qualité du produit réalisé. Pour cela, il doit utiliser un certain nombre d'outils qui lui permettent de consolider les informations, les données pour disposer d'indicateurs fiables.



Informer



Mener un projet c'est aussi et surtout communiquer avec son équipe, le client, sa hiérarchie, les éventuels intervenants extérieurs etc. Il s'agit de faire circuler l'information au juste assez et au juste à temps sans alourdir les process (ni les boites e-mails...). Transmettre aux bonnes personnes les bonnes informations au bon moment est l'une des tâches les plus difficiles. Cela demande une certaine intelligence et une très bonne connaissance des métiers et des rôles de chacun.

Alerter

C'est également l'un des rôles les plus difficiles à tenir en tant que chef de projet. L'exercice est délicat et souvent vécu comme un aveu d'échec alors qu'au contraire, c'est une preuve de professionnalisme et d'implication (la preuve, finalement que tous les autres rôles précédemment cités ont été correctement tenus...). En réalité, alerter c'est tirer à temps les conclusions de tableaux de bord et d'indicateurs correctement tenus qui permettent à un instant T d'annoncer, à partir de données factuelles et après avoir échangé avec différents acteurs, un glissement probable du projet d'un point de vue de la qualité, du délai ou du budget.



SERIOUS GAME : DE QUOI PARLE-T-ON ?

Le terme *serious game* est apparu récemment dans le paysage du multimédia et de la formation. Surtout utilisé à des fins marketing par les éditeurs du marché, il leur permet de faire connaître et valoriser leurs réalisations et leur savoir-faire. Ce flou sémantique est donc cultivé par les acteurs du marché eux-mêmes qui peuvent ainsi imposer leur vision et leur propre définition de ce qu'est ou n'est pas un serious game.

3 grandes catégories de serious games

Comme je l'indique dans mon ouvrage *Concevoir un serious game pour un dispositif de formation*⁷, on dénombre sur le marché des dizaines de termes censés se retrouver sous la dénomination commune de Serious Game : social game, business game, casual game, training game, art game, news games etc.



Figure 4 - Recensement non exhaustif des différents termes retrouvés sous l'appellation Serious game

Pour clarifier le discours, je propose de regrouper tous ces outils en 3 grandes catégories selon leurs objectifs :

- ▶ **Les learning games** : dédiés à l'apprentissage, la montée en compétences, l'acquisition de connaissances. Ils ont pour objectifs de : former, enseigner, transmettre du savoir, faire monter en compétences etc.

⁷ Concevoir un serious game pour un dispositif de formation, Béatrice LHUILLIER, Ed.FYP, 2011, p.22

- ▶ **Les persuasive games** : serious games « à message ». Message informatif (par exemple publicitaire), persuasif (par exemple communication institutionnelle) ou subjectif (telle que la propagande politique). Souvent financés par des marques qui cherchent à communiquer auprès des consommateurs, ou par des organisations, gouvernementales, politiques, associatives, qui souhaitent transmettre un message civique, politique ou syndical. Les objectifs des persuasive games : communiquer, convaincre, influencer, informer, suggérer, promouvoir, séduire, persuader etc.
- ▶ **Les simulateurs** : s'appuient sur l'entraînement et la répétition de gestes pour amener l'apprenant (ou le joueur) à acquérir un certain nombre de réflexes. L'environnement de jeu reproduit le plus fidèlement possible la réalité. Ces outils se distinguent des learning-games par le fait qu'ils entraînent non seulement les connaissances de l'utilisateur mais également sa dextérité, son aptitude à réagir dans certaines situations. L'effet attendu est au final une sorte de drill adouci.

Le « drill prussien » était un entraînement physique très sévère. L'enchaînement et la répétition d'exercices permettait de rendre les soldats, pompiers, etc. aptes à exécuter des manœuvres plus ou moins complexes sans hésitation, rapidement et sans faute sous des conditions de stress extrême (incendie, combat sous un tir nourri, utilisation d'un masque à gaz etc.)

[3 grandes catégories de Serious Games]

Learning Game

Persuasive Game

Simulateur

Composantes d'un serious game

Qu'il s'agisse d'un learning game, d'un persuasive game ou d'un simulateur, un serious game est toujours composé des quatre mêmes éléments : le fond, la forme, la mécanique, le tout supporté par un scénario (la dramaturgie).

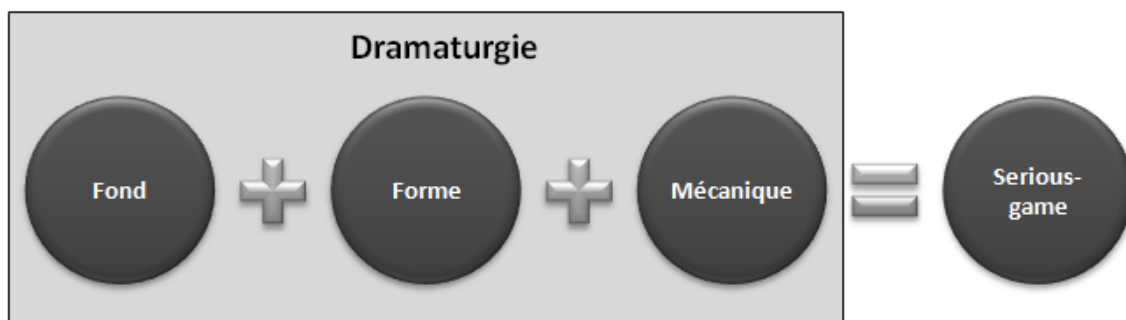


Figure 5 - Les composantes d'un serious game

- ▶ **Le fond** : ce sont les messages clés à faire passer et/ou les objectifs pédagogiques et opérationnels à atteindre. Finalement, c'est le contenu à transmettre.
- ▶ **La forme** : ce sont les éléments graphiques et sonores qui portent le produit. On parle souvent de *look and feel*.

- ▶ **La mécanique** : les anglophones parlent plutôt de *Gameplay*. Ce terme désigne l'ensemble des mécanismes et éléments servant à l'interaction : actions, contrôles, fonctionnement de l'interface, règles du jeu. Cela inclut la jouabilité, la maniabilité et la difficulté.
- ▶ **La dramaturgie** : le serious game va se caractériser par son type (action, aventure, stratégie etc.) et son genre (enquête, science fiction etc.)

Pour plus d'informations sur ces sujets, je vous invite à lire mon ouvrage *Concevoir un serious game pour un dispositif de formation*, (FYP Editions).

A QUI S'ADRESSER POUR REALISER UN SERIOUS GAME ?

Origine des éditeurs

Plusieurs dizaines d'éditeurs se partagent aujourd'hui le marché du serious game. Citons quelques noms parmi les plus connus⁸ : KTM Advance, Uni Learning, Haikara, Qoveo, Paraschool etc.

Dans leur ouvrage *Introduction au serious game*⁹, Julian Alvarez et Damien Djaouti ont recensé 13 secteurs d'origine des éditeurs de serious games.

[Les éditeurs de serious games sont issus de 13 secteurs d'activité différents]

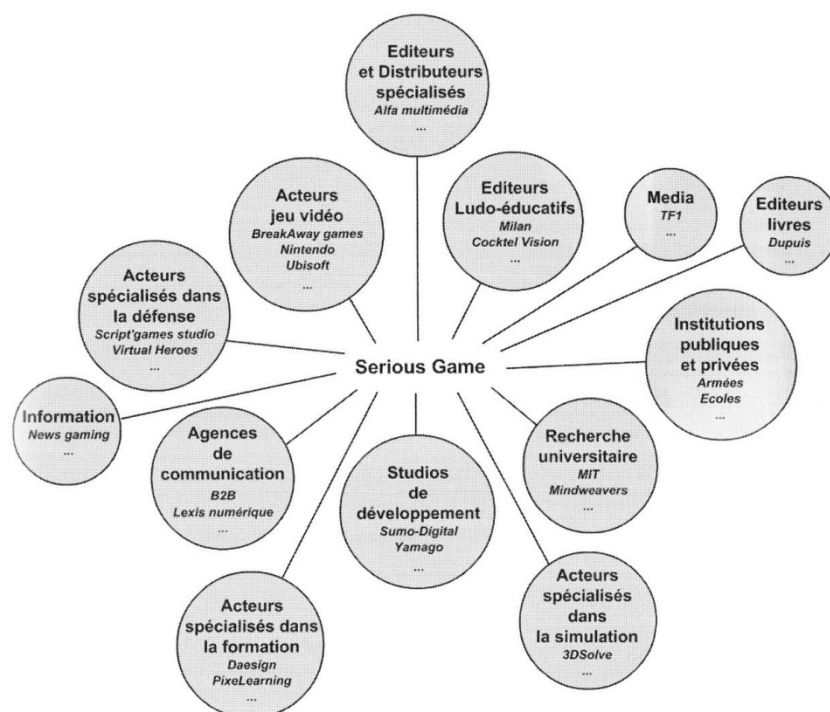


Figure 6 - Les différents secteurs d'origine des acteurs du Serious game (Alvarez / Djaouti)

Remarquez la variété des secteurs d'origine de ces acteurs : éditeurs de jeux vidéos, agences de communication, pédagogues, éditeurs de e-learning etc.

De par leur vécu et leur expertise, tous les éditeurs de serious games ne sont pas égaux face à une problématique donnée. **Tous les éditeurs ne sont pas en capacité de produire tous types de serious games.**

⁸ Pour accéder à une liste des acteurs du marché, rendez-vous sur : <http://learning-planet.fr> section Acteurs du marché

⁹ Introduction au serious game, Julian Alvarez, Damien Djaouti, 2010, éditions L>P Questions théoriques

La période que nous vivons aujourd'hui, en 2011, est encore une période de transition durant laquelle les éditeurs s'outillent, capitalisent sur leurs réalisations, continuent de s'organiser, d'affûter leur méthodologie de travail et de gestion de projets etc.

Il faut donc être prudent dans le choix de l'éditeur à qui vous allez confier votre projet.

Critères de différenciation

Voici 3 critères qui peuvent vous aider à départager les éditeurs :

1. Degré de maturité technique

- ▶ Pour développer ses serious games, l'éditeur utilise-t-il un outil-auteur ?
- ▶ Sinon, a-t-il développé un ou plusieurs moteurs adaptables ?

2. Nombre de projets réalisés

- ▶ Nombre de projets au total
- ▶ Nombre de projets du même type que celui attendu

3. Méthodologie adoptée

- ▶ Méthode prédictive
- ▶ Méthode agile
- ▶ Un mix entre méthode prédictive et méthode agile



Ces critères vont grandement influencer sur le coût global du projet. Le coût pour un même projet peut ainsi s'élever à 50k€ à plus de 250k€ selon les techniques et méthodes utilisées...

Check-list des questions à se poser

Voici une check-list, non exhaustive, des questions que vous pouvez vous poser avant de lancer votre projet :

Avant-projet

- ▶ Ai-je formalisé mon besoin dans un document clair, précis et compréhensible par tous les prestataires ?
- ▶ Quels sont les risques du projet ?
- ▶ Ai-je relevé des serious-games comparables à celui que j'imagine ? (en terme d'objectifs, de game play, de graphisme etc)
- ▶ Ai-je informé mon service Achats de ma démarche ? Un acheteur peut-il m'accompagner dans mon projet ?
- ▶ Un autre service a-t-il déjà expérimenté l'achat d'un produit semblable ? Peut-il nous faire profiter de ses retours d'expérience dans le cadre de ce nouveau projet ?
- ▶ Existe-t-il un produit sur étagère adapté à mon besoin ?
- ▶ Ai-je déterminé l'enveloppe budgétaire que je compte allouer à ce projet ?

[3 critères pour différencier les éditeurs]

Maturité technique

Expérience

Méthodologie

- ▶ Comment pourrai-je calculer le ROI du projet ?

Organisation

- ▶ Est-ce que je dispose du temps nécessaire pour la rédaction du cahier des charges ? pour le choix du prestataire ? pour le suivi du projet ? Ou vais-je nommer un chef de projet ?
- ▶ Mon équipe projet est-elle au complet ? Est-elle motivée ? Ai-je inclus des opérationnels à cette équipe ?
- ▶ Quelle méthodologie projet devra être utilisée par le prestataire ?
- ▶ Ai-je informé les syndicats de ce projet ?
- ▶ Mon service communication peut-il m'aider à communiquer en interne sur mon projet ? Si non, quel budget vais-je allouer à ces aspects ?

Pédagogie

- ▶ Qui sont les apprenants ? (moyenne d'âge, maîtrise de l'outil informatique, curiosité pour les nouvelles technologies, attrait pour les jeux vidéos...)
- ▶ Ai-je identifié clairement les objectifs que je compte atteindre en mettant en place ce serious-game ?
- ▶ En quoi est-ce la meilleure solution pédagogique pour atteindre ces objectifs ?
- ▶ Dans quel dispositif pédagogique le serious-game va-t-il s'intégrer ?
- ▶ Ai-je imaginé le dispositif d'accompagnement des apprenants ? (tutorat ? à distance ? mise en place et animation de communauté d'apprentissage ?etc)
- ▶ Quels sont les objectifs pédagogiques à atteindre ?
- ▶ Est-ce que je dispose, en interne, de compétences en ingénierie pédagogique et en ingénierie de formation ?

Technique

- ▶ Ai-je une idée des technologies que je souhaite dans mon serious-game : 2D, 3D, temps réel, vidéos, sons, réalité augmentée etc ?
- ▶ Ai-je identifié les contraintes techniques à prendre en compte auprès de mon service technique ? (débit, postes des apprenants, carte son, présence de casques etc.)
- ▶ Ma société dispose-t-elle d'un LMS ? Le serious-game devra-t-il remonter des informations au LMS ?
- ▶ Mon serious-game doit-il être disponible dans plusieurs langues, plusieurs pays ? Qui se chargera de la traduction ?
- ▶ Le serious-game doit-il intégrer de la vidéo ? Sont-elles déjà disponibles ? Doivent-elles être doublées et/ou sous-titrées ? Si oui, par qui ?
- ▶ Le serious-game doit-il être sonorisé : musique d'ambiance, virgules sonores, voix ?
- ▶ Par la suite, mes équipes seront-elles autonomes pour mettre à jour le produit ? Ou faudra-t-il forcément repasser par le prestataire ?
- ▶ Ai-je prévu une phase de test ? Pour chaque pays et langue ?

METHODES PREDICTIVES

Issues de l'industrie du BTP, les méthodes prédictives sont également appelées *méthodes traditionnelles* ou *méthodes classiques*. Elles sont utilisées aujourd'hui par un grand nombre d'éditeurs de serious games, surtout ceux issus du e-learning ou de l'édition logicielle. Les deux plus connues s'appellent *méthode en cascade* et son adaptation la *méthode en V*.

Principes communs aux 2 méthodes

► Séquencement

Le projet est subdivisé en séquences : le recueil du besoin, la définition du produit (conception générale, conception détaillée), la production, les tests et la livraison.

► Planification

Dès le début du projet, on essaie de tout planifier et tout prévoir (d'où le nom « méthode prédictive »). Un document de cadrage délimitant le périmètre du projet est rédigé, un planning de réalisation comprenant tous les jalons et les étapes de validation est porté en annexe. Côté prestataire, les ressources nécessaires à l'ensemble du projet sont pré-réservées sur cette base.

Méthode en cascade

Chaque séquence fait l'objet d'un ou plusieurs livrables qui doivent être validés par le client avant de passer à la phase suivante. On ne passe à la phase suivante que lorsque la précédente est validée par le client.

Une fois un élément validé, on ne revient pas en arrière, sauf à rediscuter le périmètre, le délai de réalisation, la qualité et/ou le budget puisqu'une modification en apparence mineure peut avoir une conséquence importante sur le périmètre du projet.

Ainsi, l'ajout de quelques lignes dans le document de conception générale (ajout de nouveaux objectifs pédagogiques par exemple) peut générer des dizaines de pages de storyboards et/ou de scénarios en plus, soient plusieurs jours de travail d'un concepteur pédagogique...

Winston Royce est considéré comme le créateur de la méthode en cascade. Mais ce dernier désapprouvait l'utilisation qui en était faite !



Figure 7 - Méthode en cascade

Il expliqua un jour que ses travaux avaient été mal interprétés et même déformés. En effet, dans l'article qu'il rédigea en 1970, Royce insistait sur la nécessité de fonctionner par itération successive et sur la possibilité de revenir en arrière sur les étapes précédentes si besoin. Ce n'est d'ailleurs même pas lui qui a baptisé cette méthode *méthode en cascade*. Mais la méthode s'est diffusée, enseignée en université et en entreprise, en ne conservant finalement que le schéma linéaire au grand dame de Winston Royce.

Méthode en V

La méthode en V est une amélioration du modèle en cascade. Elle permet un retour aux étapes précédentes tout en limitant les impacts sur le planning, le budget et la qualité du produit final.

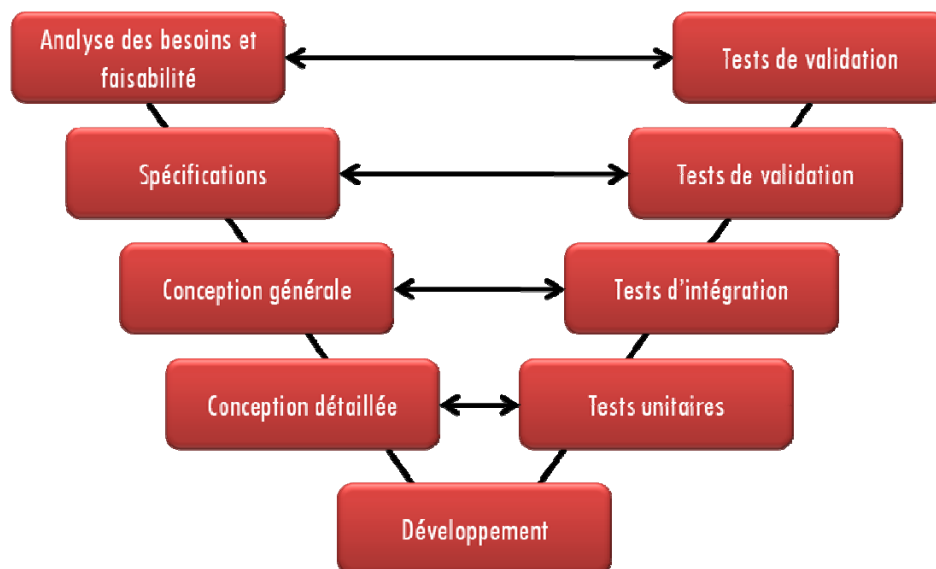


Figure 8 - Méthode en V

Les phases de la partie montante (à droite dans le schéma) renvoient de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer ce qui est produit.

Le cycle en V met en évidence la nécessité d'anticiper et de préparer dans les étapes descendantes les « attendus » des futures étapes montantes : ainsi les attendus des tests de validation sont définis lors des spécifications, les attendus des tests unitaires sont définis lors de la conception, etc.

La méthode en V est devenue un standard de l'industrie du développement de logiciels et de la gestion de projet depuis les années 1980.

Avantages

Le principal avantage des méthodes prédictives est qu'elles rassurent le client final. Le découpage en phases (ou séquences) facilite l'organisation côté prestataire et côté client, notamment pour la planification des phases de validation et de tests.

Le dossier de cadrage constitue le document de référence du projet, le chef de projet a donc pour objectif de coller au plus près à ce qui a été défini. De fait, il dispose d'une boussole tout au long de sa mission.

Les phases de cadrage et de conception permettent d'établir un langage commun entre le client et le prestataire. Elles permettent également de s'entendre sur un résultat à obtenir en fixant les besoins via des spécifications de plus en plus détaillées avant de lancer la production à proprement parler.

Enfin, ces méthodes ont été utilisées dans de nombreux domaines, elles sont donc aujourd'hui bien connues des acteurs des projets. Si ce n'est pas le cas, simples dans leur forme, elles sont faciles à expliciter à un client novice.

Inconvénients

Un client qui n'a jamais mené de projet de type e-learning ou de projet informatique au sens large, aura du mal à se projeter et à imaginer à quoi le produit final va ressembler. Il faut en effet, très tôt concevoir, c'est-à-dire, lister l'ensemble des besoins de manière exhaustive. Dans la méthode en cascade, le retour en arrière est théoriquement impossible (d'où le terme *méthode en cascade* qui implique un écoulement dans un sens unique sans possibilité de retour arrière). La méthode en V a justement été conçue pour palier ce problème de réactivité de la méthode en cascade.

De plus, ce type de méthodologie implique un *effet tunnel* perturbant et même stressant, surtout dans le cas où le coordinateur du projet, côté client, mène ce type de projet pour la première fois. En effet, tout doit être décrit en phase de conception (générale puis détaillée) puis, la phase de développement démarre durant laquelle le client n'a que très peu de vue sur la production en cours. Il faut souvent attendre que le moteur du jeu soit totalement développé et les premiers écrans et/ou scénarios intégrés pour avoir une vision du produit final. Cela peut, dans certains cas, prendre des semaines voire des mois dans le cas des projets les plus ambitieux ne s'appuyant pas sur un moteur existant.



Il faut très tôt décrire le futur contenu dans un dossier de conception générale qui engage le client et le prestataire. En conséquence, les acteurs du projet en viennent à redouter tout changement ou oubli, de peur de mettre en péril le projet. Car, **par définition, pour garantir le délai, la qualité ou le budget il faut respecter les spécifications imaginées en début de projet.** En conséquence, comme l'indique Véronique Messenger Rota « *ces approches prédictives se sont révélées trop rigides parfois, exposant les organisations à trop peu de réactivité dans le contexte de nouveaux projets stratégique* ».

Enfin, la méthode en cascade implique que les tests aient lieu tardivement. Pour palier cela, certains éditeurs proposent bien sûr d'élaborer un prototype qui permet de visualiser, un peu plus tôt dans le process, le résultat de la production. La méthode en V permet de contourner, du moins en partie, cette carence.

Exemple de projet serious game avec la méthode en cascade

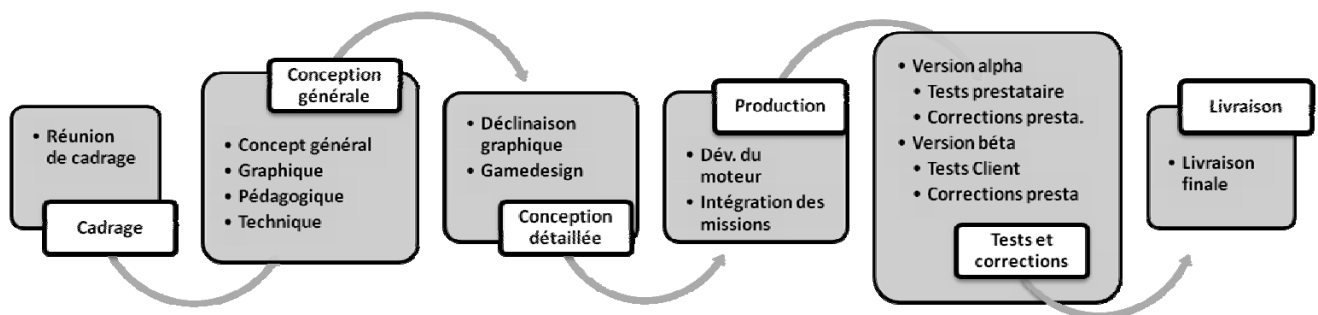


Figure 9 - Exemple de méthode prédictive appliquée à la gestion d'un projet de serious game

METHODES AGILES

Principes

Le mouvement des méthodes agiles est né au tout début des années 2000 aux USA. Ayant constaté un fort taux d'échecs des projets de développement logiciel, 17 experts du sujet ont décidé de mettre en commun leur savoir-faire et leur expérience pour inventer une nouvelle méthodologie de gestion de ce type de projet.

L'objectif de cette communauté était de penser et d'élaborer une nouvelle façon de concevoir et développer des logiciels, en créant un socle commun de valeurs et de bonnes pratiques. Ce travail a donné naissance au *Manifeste pour le Développement logiciel agile* et la création de l'*Agile Alliance*.

Depuis, de nombreuses déclinaisons ont vu le jour, toutes rangées sous la dénomination commune de *Méthode agile* : ASD, Chrystal, DSDM, RAD, Lean, Scrum, Unified Process, eXtreme Programming, Xbreed, EVO, FDD etc. Toutes ces méthodes ont en commun 4 valeurs et 13 principes de base.

Au final, on constate que toutes les méthodes ont un tronc commun de pratiques puis se différencient par leur degré de formalisme ; le nombre et la formalisation des différentes étapes, les revues, le rythme du projet, le nombre et la durée des itérations etc.

Les 4 valeurs du Manifeste :

- ▶ Les individus et leurs interactions avant les processus et les outils
- ▶ Des fonctionnalités opérationnelles avant la documentation
- ▶ Collaboration avec le client plutôt que contractualisation des relations
- ▶ Acceptation du changement plutôt que conformité aux plans



Attention, les valeurs mises en avant à gauche n'excluent évidemment pas les valeurs de droite. En revanche, en fonction des situations, du contexte, on privilégiera plutôt les valeurs de gauche.

Ces 4 valeurs sont déclinées en 13 principes :

- ▶ Notre priorité est de **satisfaire le client** en lui livrant très tôt et régulièrement des versions opérationnelles de l'application, source de valeur.
- ▶ **Accepter le changement dans les exigences**, même tard dans le cycle de vie, pour garantir la compétitivité du client.
- ▶ **Livrer le plus souvent possible des versions opérationnelles** de l'application, à une fréquence allant de 2 semaines à 2 mois.

[Sites références]

agilemanifesto.org

agilealliance.org

pmdoi.org

[Socle commun]

4 valeurs

13 principes

- ▶ Client et développeurs doivent **coopérer quotidiennement** tout au long du projet.
- ▶ Construire des projets autour d'**individus motivés**. Leur donner l'environnement et le support dont ils ont besoin et leur **faire confiance** pour remplir leur mission.
- ▶ La méthode la plus efficace pour communiquer des informations à une équipe et au sein de celle-ci reste **la conversation en face-à-face**.
- ▶ **Le fonctionnement de l'application est le premier indicateur** d'avancement du projet.
- ▶ Les méthodes agiles recommandent que le projet avance à un **rythme soutenable**.
- ▶ Sponsors, développeurs et utilisateurs doivent pouvoir **maintenir un rythme constant indéfiniment**.
- ▶ Porter une attention continue à l'**excellence technique et à la conception** améliore l'agilité.
- ▶ La **simplicité** – art de maximiser la quantité de travail non fait – est essentielle.
- ▶ Les meilleures architectures, spécifications et conceptions sont le fruit d'**équipes qui s'auto-organisent**.
- ▶ A intervalles réguliers, l'ensemble de l'équipe s'interroge sur la manière de **devenir encore plus efficace**, puis ajuste son comportement en conséquence.

Chaque méthode peut bien sûr enrichir ce socle de valeurs et de principes mais ne peut pas supprimer d'éléments pour continuer à pouvoir s'appeler *Méthode Agile*. Une méthode peut adopter sa propre terminologie et préconiser ses propres pratiques.

Détails de la méthode

Approche

Une méthode agile se caractérise, en premier lieu, par son **approche itérative et incrémentale**.

Le projet est découpé en plusieurs étapes de durée égale, de quelques semaines (souvent 15 jours) : les **itérations** aussi appelées *tranches de temps, boîtes de temps ou sashimis*.

Lors d'une itération, une version minimale du produit attendu est développée puis soumise au client pour validation. Les fonctionnalités sont ainsi intégrées au fur et à mesure du cycle de vie selon un **mode incrémental**.

Chaque itération constitue une sorte de mini-projet qui comporte toutes les tâches : analyse, conception, développement et test. L'objectif étant d'obtenir, à l'issue de chaque itération, un sous-ensemble opérationnel du logiciel cible et, au terme de la dernière itération, la version finale du produit attendu.

Les itérations sont regroupées au sein de **releases**. Le tout constituant la Road Map du projet.

Notons que l'on peut prévoir une itération par release, ce qui signifie une livraison à chaque fin de release, mais ce n'est pas systématique.

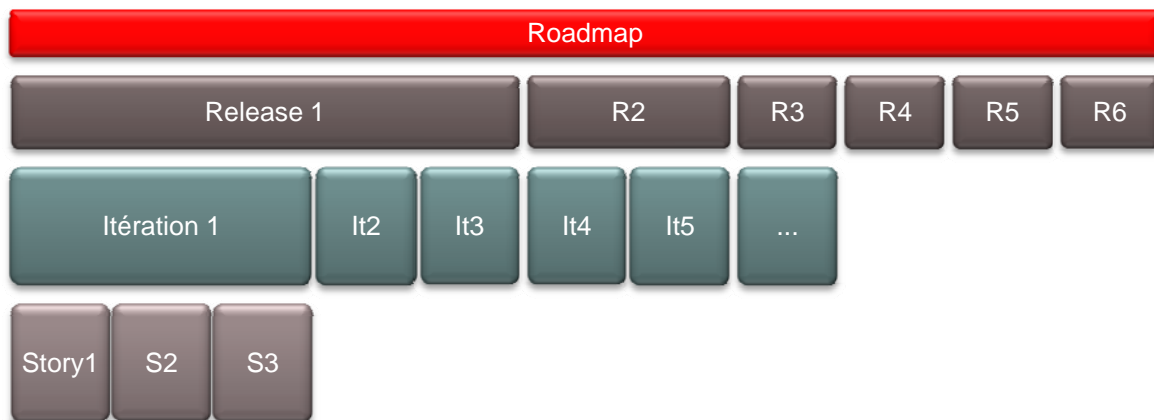


Figure 10 - Exemple de Roadmap

Mise en place du Product Backlog

Concrètement, en début de projet, un document est créé permettant de lister les **exigences** attendues par le client final. Ce document est appelé *Cahier d'exigences*, *Référentiel d'exigences* ou *Product Backlog*. Il peut s'agir d'exigences fonctionnelles (une règle de gameplay) ou non fonctionnelles (remontée d'un score sur un LMS). En début de projet, il n'est pas censé être exhaustif : il est enrichi au fur et à mesure de l'avancement du projet.

Une exigence :

- ▶ est rédigée dans un langage commun, non technique
- ▶ est orientée vers l'usage de l'utilisateur final (l'apprenant)
- ▶ est compréhensible par tous les acteurs du projet (voir extrait de Product Backlog en annexe).

Exemple d'exigence : Lorsque le personnage s'approche d'un élément cliquable, il se met à briller pendant 4 secondes.

Use case et User story

Selon le type de projet à réaliser (logiciel, site internet, serious game), l'éditeur peut opter pour l'une des deux méthodes suivantes : la méthode des *Use cases* (cas d'utilisation) ou la méthode des *User stories*. Dans les deux cas, ces UC ou US représentent les tests qui permettront de vérifier le produit répond bien à l'exigence qui avait été énoncée.

Le Use Case est une technique utilisée surtout dans le cadre de la méthode dite du *Processus Unifié*. Cela consiste à décrire, d'un point de vue strictement fonctionnel, les échanges entre le système à développer et les acteurs / éléments externes. On décompose alors l'usage qui est fait du produit, dans un contexte donné. Il s'agit donc finalement de décrire point par point le séquençement, les enchaînements de manipulations à réaliser pour arriver à un résultat. Il faut envisager et décrire tous les cas possibles (voir exemple de Use Case en annexe). Dans le cas du développement d'un serious game, on utilise plutôt la méthode des user stories, plus adaptée.

Une User Story est formulée en une ou deux phrases dans le langage de l'utilisateur. C'est finalement « se mettre dans la peau de l'utilisateur final » et imaginer les conséquences des actions de l'utilisateur et ce qui se passera à l'écran ou en backoffice. Assez précise, elle doit permettre à l'équipe de production d'estimer son coût. Elle doit pouvoir être réalisée entièrement lors d'une itération. On les note assez grossièrement dans le fichier de Backlog puis on les discute et on les détaille oralement et sur des fiches bristol lors des *sprint planning meeting* avec le client.

Planification des livraisons

Au début du projet, puis au début de chaque itération, le client et l'équipe se réunissent pour se mettre d'accord sur les stories qui vont être implémentées dans la prochaine itération. Cette réunion est appelée *Sprint Planning Meeting*.

1. Présenter les stories

Comme nous l'avons vu précédemment, une user story est, par définition, assez sommaire. Le client, aidé du chef de projet, est donc invité à commenter chaque user story à l'équipe qui pourra ensuite poser des questions si nécessaire pour affiner la demande.

2. Estimer les coûts

Il est difficile, surtout en début de projet, d'estimer le coût en temps de réalisation d'une story. C'est pourquoi, pour faciliter ce travail, il est préférable d'opter pour une unité de temps abstraite comme un système de points qui, au fil du projet, va s'affiner.

Pour ce faire, on choisit dans le lot de stories celle qui semble la plus petite et la plus simple à développer et on lui attribue la note de 1 point. Toutes les autres stories sont évaluées par rapport à cette story de base qui devient en quelque sorte l'« étalon ». On classe alors tous ensemble les stories selon leur poids approximatif « à la louche ». Puis, on détaille : telle story est 2 fois plus compliquée que celle-ci, on lui attribue 2 points, telle story vaut 30 points.



Dans les cas où l'équipe a du mal à procéder ainsi, il existe une méthode ludique, appelée *Planning Poker*. Chaque participant dispose de 13 cartes numérotées de 0 à 100. Les valeurs des cartes correspondent aux premiers nombres de la suite de Fibonacci : 1,2,3,5,8 et 13 auxquels sont ajoutées les valeurs 20, 40 et 100. Pour les mini bugs ou les évolutions mineures on dispose du 0 et du ½. Les deux dernières cartes sont : la carte « ? » qui permet d'indiquer que l'on est pas en mesure de procéder à l'estimation et la carte « café » qui permet de demander une pause et signifier ainsi qu'on n'a plus les idées assez claires pour procéder à l'estimation !



Figure 11 - Cartes de planning Poker (site Crisp.se)

Mon conseil : il est préférable de subdiviser une story dont l'estimation atteint 40 ou 100 en unités plus petites et plus « digestes ». De même, lorsque c'est possible et pertinent, il est préconisé de regrouper les plus petites stories en lot.

Notons enfin que l'estimation comprend toutes les tâches nécessaires à la livraison d'une story : y compris les tests et la documentation.

3. Prioriser les stories

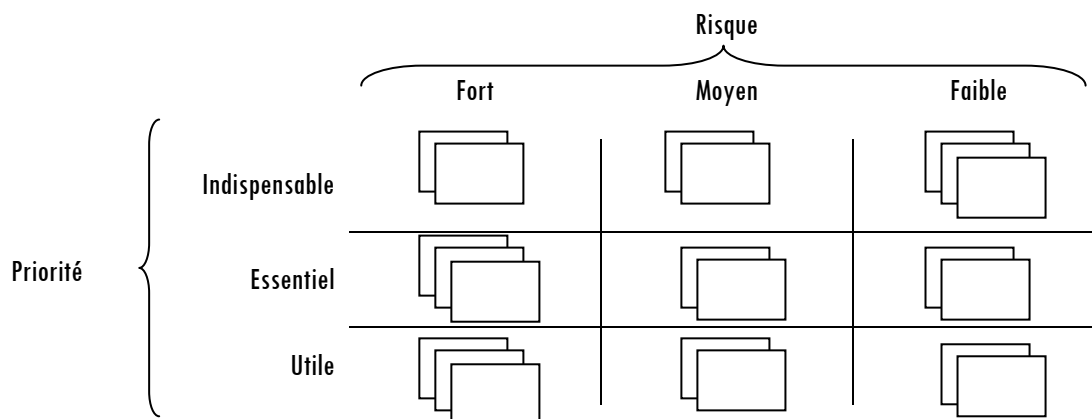
Au terme des deux premières étapes, l'équipe dispose d'un ensemble de stories détaillées et estimées. On procède alors à une priorisation des stories en vue de les répartir sur l'itération à venir et les suivantes.

Le tri se fait selon 2 critères :

- ▶ La priorité, déterminée par le client
- ▶ Le risque, déterminé par l'équipe de développement

En début de séance, le client est invité à classer les stories en trois tas : indispensable / essentiel / utile. Il s'agit du degré de priorité d'un point de vue fonctionnel.

Ensuite, les développeurs trient à leur tour chaque tas de stories en trois nouveaux tas en fonction du « risque technique » qu'ils représentent (il s'agit finalement de la complexité technique) : fort / moyen / faible.



Ainsi, on le comprend aisément, client et équipe sont autant responsables de la répartition qui suit.

Il existe d'autres méthodes de priorisation, telle que la méthode MOSCOW (FISPE en Français : Fonctions Indispensables, Souhaitables, Possibles, Eliminées), la méthode de Mike Cohn, le modèle de Kano ou encore la méthode, plus complexe, des poids relatifs.

4. Annoncer la vélocité

Comme l'indique le collectif d'auteurs de l'ouvrage *Gestion de projet Extreme Programming*¹⁰ :

La vélocité de l'équipe correspond au total des estimations en points des scénarios qui ont été totalement et correctement implémentés au cours de l'itération précédente.

La vélocité de l'équipe correspond donc au nombre de points de scénarios client que l'équipe peut traiter en une itération.

Par définition, lors de la toute première itération, l'équipe n'a pas de données sur lesquelles s'appuyer pour annoncer la vélocité. Il est donc nécessaire de procéder à une estimation.

Mon conseil : mieux vaut, lors de la première itération, surestimer la vélocité de l'équipe, c'est-à-dire annoncer volontairement un nombre plus important de points que l'équipe est en mesure de prendre en charge. Cela permettra de correctement réajuster lors de l'itération suivante quitte à constater que certaines stories n'ont pu être réalisées.

¹⁰ Jean-Louis Bénard, Laurent Bossavit, Régis Médina, Dominic Williams, *Gestion de projet eXtreme Programming* Ed.Eyrolles, 2004, page 144

Ne confondez pas vélocité et productivité. La vélocité est plutôt un indicateur qui doit permettre d'établir puis d'affiner la planification du projet.

5. Répartir les scénarios

On connaît à présent la vélocité de l'équipe, le client peut donc, en toute connaissance de cause, choisir, parmi les 9 tas de stories, un nombre de scénarios dont la somme des points est égale à ce total, en s'assurant que les plus importants soient bien traités en premier (ceux classés dans « Indispensable » ou « Essentiel »).

Si une liste quasi exhaustive des stories a été rédigée, il est possible par cette méthode, de répartir l'ensemble des stories sur toutes les itérations à venir. On obtient alors un planning complet du projet mais il s'agit bien sûr d'un plan provisoire puisque les exigences, et donc les stories peuvent évoluer, au fil du projet.

Mon conseil : utilisez cette méthode uniquement pour planifier une ou deux itérations à venir et non le projet complet car, par définition, les exigences et donc les stories peuvent évoluer.

A noter : même si toutes les stories n'ont pu être implémentées, la livraison est effectuée, tout en sachant que les stories qui sont reportées sont les moins importantes de l'itération.

Avantages

Les méthodes agiles permettent une meilleure communication au sein de l'équipe mais également entre le client et l'équipe. Les malentendus et incompréhensions sont rapidement soulevés et traités pour une meilleure efficacité.

Le client visualise très régulièrement l'avancée des travaux, tout au long du projet, sans subir d'effet tunnel ou boîte noire.

Les tests unitaires et les tests de recette sont effectués à chaque itération, les anomalies sont donc corrigées au fur et à mesure.

Le système de calcul par points permet de fiabiliser les prévisions.

Le client peut ajuster ses exigences au fur et à mesure du projet, en fonction des livraisons ou tout simplement parce que son environnement a changé, ses contraintes etc. Le changement n'est plus une menace pour le projet, au contraire, c'est l'occasion de mieux faire et de mieux coller aux besoins du client, le tout sans mettre à mal le budget et le délai.

Les risques liés aux coûts sont limités à l'itération en cours.



Inconvénients

Le client est sollicité tout au long du projet, il a un rôle plus important, plus impliquant que dans le cadre d'une méthode traditionnelle de gestion de projet. En conséquence, il peut avoir l'impression de passer plus de temps sur le projet mais c'est véritablement pour le bien de son projet.

Sauf exception, on ne planifie pas l'ensemble du projet dès le début. Cela peut donner l'impression au client d'un flou artistique, heureusement de courte durée puisque la première itération a pour objectif de caler les différents indicateurs de suivi et de calculer la vélocité de l'équipe.

Cette méthode est connue pour être efficace avec des équipes à taille humaine. L'instaurer sur des projets multisites avec un nombre important d'acteurs nécessite des ajustements et un outillage adapté (outil de travail collaboratif notamment).

Mettre en place une méthode agile dans une équipe nécessite un travail d'adaptation, de formation et d'accompagnement du changement.

QUELQUES CONSEILS POUR FINIR...

Si près de 75% des projets informatiques connaissent des décalages planning et des dépassements de budget, la faute ne peut pas entièrement être rejetée sur la méthodologie de travail adoptée.

Nombreux sont les éditeurs qui parviennent, avec des méthodologies traditionnelles à tenir les objectifs. Cela dépend de quantité de facteurs : le type de projet, l'expérience du chef de projet, la maturité de l'équipe en charge de la réalisation, l'organisation générale de l'entreprise, la volumétrie de l'équipe de production, la possibilité de paralléliser les flux de production, la possibilité de mobiliser une équipe sur un projet donné, l'utilisation ou non d'outils de production stables permettant des gains de productivité, la réutilisation de moteurs de jeux proprement développés et facilement adaptables etc.

Au final, d'autres raisons peuvent expliquer les nombreux échecs :

Des délais démentiels...

La "deadline" du projet est souvent décidée avant même que le projet n'ait démarré et n'est pas négociable. Elle est mentionnée comme une contrainte, un impératif dans un cahier des charges souvent flou, sans possibilité pour l'éditeur de faire valoir la nécessité d'un tempo différent pour obtenir un résultat qualitatif. Dans certains cas, le cahier des charges prévoit une date de début de projet qui n'est pas respectée mais la deadline, elle, n'est pas modifiée... : tout le monde a à perdre avec ce genre de pratique. Le prestataire qui, dès le début, est engagé dans un planning qui est déjà obsolète. Le client qui, dès le début, met en danger son projet sur les deux autres aspects : qualité et budget.

Au final, l'éditeur a la tentation de commencer à coder au plus tôt pour espérer finir dans les temps. Le plus souvent, c'est une mauvaise approche du problème, sauf à utiliser une méthode agile qui permet, effectivement, de commencer à développer très tôt.

Or, en utilisant les méthodes traditionnelles, cela revient à précipiter et à raccourcir la phase de conception du produit, en croisant les doigts pour ne rien oublier et en priant pour que le client final n'ait pas de nouvelles idées une fois que la production aura commencé...

Mon conseil : Si c'est une méthodologie projet traditionnelle qui est utilisée, ne précipitez jamais les phases de conception : elles sont nécessaires et même indispensables.



Autre conseil : Plutôt que de précipiter la phase de conception, prévoyez tôt dans le projet un prototype pour visualiser à quoi va ressembler le produit final. Cela rassure l'équipe cliente et permet de procéder à des ajustements si nécessaire.

Un budget insuffisant

Bon nombre de projets ne sont pas correctement calibrés en amont. Les cahiers des charges sont plus ou moins précis mais le prestataire est tenu de rédiger une offre qui, elle, doit être très détaillée ! Sa réponse engage l'entreprise alors même que les caractéristiques du projet ne sont pas encore connues...

Pire, c'est sur cette base que le client final va choisir un prestataire en considérant en premier lieu le budget annoncé (et on le comprend, surtout lorsque l'on sait qu'un projet de serious game démarre à quelques dizaines de milliers d'euros pour atteindre plusieurs centaines de milliers d'euros dans certains cas...)

Finalement, beaucoup de projets se décident sur la seule base du prix le plus bas. Mieux vaut-il choisir ce prestataire à 50k€ qui vous réclamera 15k€ d'avenants négociés âprement et dans la douleur tout au long du projet ? Ou celui-ci qui, certes annonce 65k€ mais dont la méthodologie de travail vous offre une certaine latitude pour ajuster le produit tout au long du projet ? A la fin, le budget consommé est identique mais posez-vous la question de la qualité des relations entretenues ensuite entre le client et son prestataire, et de la réputation de ce dernier...

Mon conseil : soyez réaliste et faites-vous accompagner dans le choix de votre prestataire. Tous ne se valent pas, tous ne sont pas aptes à développer tout type de produit.

Une communication défailante

Histoire vécue cent fois par cent chefs de projet et clients différents dans toutes les entreprises informatique du monde : Le client peste, le produit n'est pas conforme. Il met en cause le prestataire. Le chef de projet incrimine le testeur qui lui-même accuse le développeur...Celui-ci précise que le concepteur n'était pas clair dans ses spécifications. Le concepteur enrage et se retourne contre le chef de projet qui finalement s'en prend à ce fichu commercial qui a vendu n'importe quoi à son client...

Toute ressemblance avec votre histoire serait purement fortuite...



Une bonne communication entre l'équipe du prestataire, le client, les utilisateurs est primordiale pour la réussite d'un projet. Une mauvaise communication peut générer des confusions, des incompréhensions et au final des complications plus tard dans un projet. Il ne faut jamais partir du principe que « tout le monde a compris ». Prenez le temps qu'il faut, toujours, pour vous assurer que les messages importants ont bien été transmis aux bonnes personnes. Les outils de travail collaboratif ne font pas tout : ils sont très utiles mais ne se suffisent pas à eux-mêmes. Les contacts entre les êtres humains sont primordiaux : qu'il s'agisse de réunion sur site ou à distance, peu importe, faites en sorte que les gens se parlent.

Cessons de cloisonner les rôles : trop de développeurs travaillent dans leur coin sans JAMAIS rencontrer le client, c'est aujourd'hui une aberration, d'autant que permettre ce rapprochement crée de la valeur pour le projet. Le client prend connaissance des contraintes des développeurs (ce ne sont pas des bêtes de somme et ils ne ressemblent pas tous à des geeks...) et les développeurs ont l'occasion d'échanger avec le client, sans filtre, sans téléphone arabe, de comprendre le contexte d'un projet et les contraintes du client.

Mon conseil : quelle que soit la méthodologie de travail adoptée, faites en sorte de favoriser les échanges entre l'équipe du prestataire et le client.

La phase de tests est négligée

Quand la pression est forte pour livrer dans des délais déraisonnables, c'est souvent la phase de test qui en pâtit. En effet, souvent reléguée en toute fin de projet, cette phase, pourtant primordiale est négligée.

Certains éditeurs (trop peu nombreux...) ont mis en place une cellule de tests, chargée de tester tout ce qui sort du pôle développement, dans différents environnements de tests (différents systèmes d'exploitation, différents navigateurs internet etc.)

D'autres font appel à des spécialistes du test (les sociétés Bug Tracker ou Enzyme par exemple).

Les autres comptent sur le fait que le chef de projet testera les produits avant de les envoyer à son client (c'est évidemment trop tard), ou tentent de mettre en place un système de tests tournants (un membre de l'équipe est chargé de tester le contenu d'un projet sur lequel il n'a pas travaillé par exemple).



Mon conseil : dans le cadre des méthodologies agiles, les tests font partie intégrante du processus et ne sont pas relégués en fin de production, ce qui permet d'ajuster rapidement le produit et empêche de détecter en toute fin de projet un énorme bug bloquant que personne n'avait vu jusqu'alors... Dans le cadre des méthodologies traditionnelles, assurez-vous que le prestataire a prévu une organisation adaptée pour minimiser les risques à ce niveau.

Le test est une mission très particulière, qui nécessite de la rigueur, beaucoup d'organisation et des moyens. Si le contenu est prévu pour fonctionner sur tous types de navigateurs web par exemple, il faut que le contenu soit testé dans tous ces environnements. Sinon, cela peut donner des échanges du type « Je ne vois pas telle interactivité ». Réponse « Chez moi ça marche... » qui bien entendu, ne font pas avancer le problème... (toute ressemblance avec des situations vécues, une fois encore, serait purement fortuite...).

Pour finir...

Agiles ou traditionnelles, il n'y a pas de bonnes ou de mauvaises manières de gérer un projet de serious game. Il y a des méthodologies de travail qui sont plus ou moins adaptées selon le type de projet que vous allez mener.

Pour terminer, je vous invite à me contacter pour échanger sur ces sujets passionnants.

ANNEXES

Extrait d'un Product Backlog

ID	Exigences	Catégorie	Type	UC ou US	Priorité	Risque	Coût
ID001	Les déplacements doivent se faire via une map de la région Bourgogne	Navigation	F	Quand on clique sur une ville, on voit un mini personnage se déplacer sur la carte en marchant.			
ID002	La carte de la région comprend n points. Chaque point correspond à une ville ou un village	Navigation	NF	Quand on ouvre la carte, les points accessibles sont d'une couleur différente.			
ID003-1	Chaque ville est matérialisée par un ou plusieurs décors	Univers	NF	Quand on clique sur une ville, le personnage se déplace sur la carte PUIS le décor apparait avec notre avatar au premier plan			
ID003-2	Chaque ville est matérialisée par un ou plusieurs décors	Univers	NF	Quand on a terminé une mission dans un décor donné, si on y revient, il est possible que l'on accède à un nouveau décor (non systématique).			
ID004	La carte de la région est accessible via un bouton de l'interface	Navigation	F	Quand on clique sur le bouton carte de l'interface, on accède à la carte de la région.			
ID005	Dans un décor, plusieurs personnages sont présents, certains sont fixes d'autres mobiles.	Univers	NF	Quand on accède à un décor, des personnages sont présents, certains bougent d'autres non.			
ID006	Pour parler à un personnage, l'apprenant doit cliquer dessus : un boîte d'actions possibles s'ouvre	Interactivité	F	Quand on clique sur un personnage, une boîte d'action s'ouvre			
ID007	Une boîte d'actions peut contenir au minimum 2 items : Voir et Ecouter et au maximum 4 : + Parler et Envoyer un sort (prévoir possibilité d'autres actions non encore envisagées)	Interactivité	NF	Quand on clique sur un personnage, une boîte d'actions s'ouvre et offre différentes possibilités d'interactions (différentes selon les personnages et leur rôle dans le jeu)			
ID008	Dans un décor, un ou plusieurs objets sont cliquables.	Interactivité	F	Quand on est dans un décor, les éléments cliquables brillent.			

Exemple de Use Case

Titre : demander de l'aide à une fée

Résumé : l'apprenant doit répondre à une énigme. Le système doit lui permettre de demander de l'aide à une fée en cliquant dessus. Il peut cliquer jusqu'à 3 fois par énigme.

Acteurs : apprenant, fée, LMS

Scénario nominal

- ▶ **étape 1** : l'apprenant ne parvient pas à répondre à une énigme
- ▶ **étape 2** : l'apprenant clique sur la fée
- ▶ **étape 3** : la fée donne un indice à l'apprenant pour l'aider à répondre à l'énigme
- ▶ **étape 4** : l'apprenant répond à l'énigme
- ▶ **étape 5-1** : l'énigme est considérée comme réussie, le score est augmenté de 10 points
- ▶ **étape 5-2** : l'énigme est ratée, le score reste inchangé.

Scénario alternatif

A1 : l'apprenant a déjà demandé de l'aide 10 fois à la fée précédemment.

- ▶ **étape 3 -1** : la fée indique à l'apprenant qu'il a déjà fait appel à elle 10 fois et lui précise que s'il demande un nouvel indice, il ne recevra que 8 points au lieu de 10.
- ▶ **étape 3-2** : la fée demande si il souhaite OUI ou NON accéder à l'indice dans ces conditions
- ▶ **étape 4** : l'apprenant répond à l'énigme
- ▶ **étape 5-1** : l'énigme est considérée comme réussie, le score est augmenté de 8 points
- ▶ **étape 5-2** : l'énigme est ratée, le score reste inchangé.

Scénario d'exception

E1 : l'apprenant ne réussit pas à répondre à l'énigme. Solution de contournement.

- ▶ **Etape 2** : l'apprenant clique pour la 3ème fois sur la fée
- ▶ **Etape 3** : la fée lui donne la réponse
- ▶ **Etape 4** : l'énigme est considérée loupée, le score est décrétementé de 3 points

BIBLIOGRAPHIE

www.agilealliance.org.

www.agilemanifesto.org.

ALVAREZ Julian, DJAOUTI Damien. *Introduction au serious game*. Édité par L>P Questions théoriques. 2010.

AZEAU, Olivier. <http://agiliteur.azeau.com>.

BENARD Jean-Louis, BOSSAVIT Laurent, MEDINA Régis, WILLIAMS Dominic. *Gestion de Projet eXtreme Programming*. Eyrolles, 2004.

Declaration of Interdependence. <http://www.pmdoi.org/>.

Fefaur. *Etudes e-learning*. <http://www.fefaur.com/fr/etudes.html>.

J.Johnson. "Keynote speech." Sardaigne (Italie), 2002.

LAVERGNE BOUDIER Valérie, DAMBACH Yves (KTM Advance). *Serious Game Révolution pédagogique*. Paris: Hermes Science - Lavoisier, 2010.

LHUILIER, Béatrice. *Concevoir un serious game pour un dispositif de formation*. FYP Editions, 2011.

MESSAGER ROTA, Véronique. *Gestion de projet - Vers les méthodes agiles*. 2è édition. EYROLLES, 2009.

Planning Poker. <http://www.crisp.se/planningpoker/>.

PRAT, Marie. *E-learning - Réussir un projet : pédagogie, méthodes et outils de conception, déploiement, évaluation*. Editions ENI, 2008.

Standish Group. *Standish Newsroom - Chaos 2009*.
http://www1.standishgroup.com/newsroom/chaos_2009.php.

PRESENTATION DE L'AUTEURE

Béatrice conseille les organisations dans :

- ▶ le choix de prestataires de solutions e-learning et serious game
- ▶ l'ingénierie de formation de leur dispositif
- ▶ la mise en place de leur cellule interne de gestion de projets e-learning et/ou serious game

Béatrice commence sa carrière en tant que **formatrice – chef de projets** chez [Oxalys technologies](#). Elle forme les utilisateurs et gère plusieurs dizaines de projets d'implémentation de solutions progiciels.

Passionnée de pédagogie et de nouvelles technologies, elle intègre ensuite la société [iProgress](#). **Conceptrice pédagogique et chef de projets**, elle conçoit et met en place des dispositifs de formation à distance, pour des PME et Grands comptes dans presque tous les secteurs d'activités .

Puis, elle intègre la société [Paraschool](#) en qualité de **Directrice de Production**. Elle organise tout le Pôle de gestion de projets, conception et production e-learning et serious games et continue, en parallèle, à piloter les projets les plus complexes et/ou stratégiques.

Devenue **consultante indépendante**, elle fonde le portail [learning-planet.fr](#) et réunit sous cette bannière des consultants, tous indépendants, dans le domaine de la formation à distance.

Béatrice est diplômée de l'**IFAG** (Institut de Formation aux Affaires et à la Gestion) et d'un Master **CAFEL** (Chef de projet Apprentissage et Formations en Ligne).

En 2011, elle publie *Concevoir un serious game pour un dispositif de formation* aux Editions FYP.

Pour la contacter

www.beatrice-lhuillier.fr
contact@beatrice-lhuillier.fr
06 98 10 32 51

Toutes photos et illustrations : Fotolia®

Sauf schéma « les secteurs d'origine des acteurs du marché » : J.Alvarez et D.Djaouti

Et photo du jeu de cartes de planning poker issu du site www.crisp.se

