

Génie logiciel : du prédictif à l'adaptatif

Les cycles de vie et les méthodes

Jean-Pierre Vickoff

Dans le domaine du génie logiciel, l'arbre généalogique des méthodes se limite à deux branches. La première, dans l'ordre d'apparition chronologique, est fondée sur un cycle de levée du risque par la validation d'une succession de nombreuses étapes. L'objectif étant de tenter de réduire le risque de changement en cours de développement. Les méthodes issues de cette branche sont représentées par la métaphore d'une « cascade ». La seconde branche, plus récente, se base sur la mesure objective du changement afin de l'accepter. Elle se structure selon un cycle dit « itératif, incrémental et adaptatif ». Les méthodes issues de la branche « cascade » se réfèrent du « prédictif » et les autres de l'« adaptatif ». La dernière génération de méthodes adaptatives se qualifie « d'Agiles ».

Itératif et incrémental

Autant le principe de la cascade est simple à appréhender, autant les notions sous-jacentes aux principes incrémental itératif adaptatif sont fréquemment mal comprises.

La métaphore du tableau de la Joconde, choisie par Jeff Patton (Figure 1) pour matérialiser individuellement ces concepts est particulièrement pertinente.



Figure 1. — Itératif / incrémental (Jeff Patton)

Afin de visualiser le principe des méthodes Agiles combinant ces deux premiers concepts, je propose sur cette base, le montage suivant (Figure 2) qui

matérialise la dimension temporelle de l'itération, donc le pilotage du projet en regard de la dimension applicative de l'incrément.

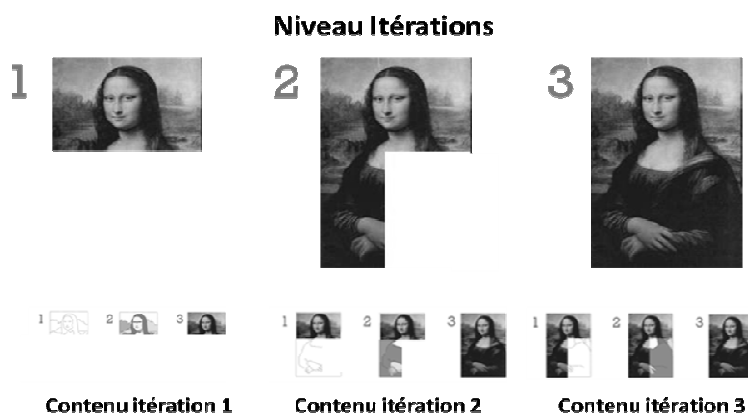


Figure 2. — L'itératif combiné à l'incrémental

À la fin de la première itération, un livrable utilisable est produit. En effet, si la production devait cesser à ce point, le client disposerait d'un portrait au lieu d'un buste, mais au moins il serait achevé.

La figure 3 illustre l'addition du concept « adaptatif » (sa réalité de mise en œuvre est beaucoup moins triviale que son illustration).

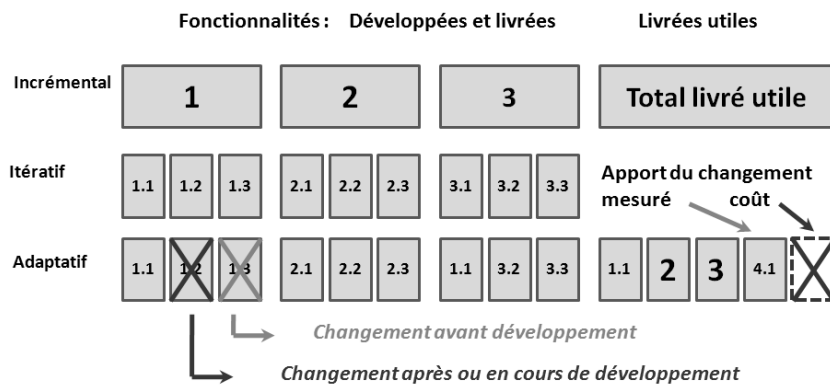


Figure 3. — Agile = Itératif +Incrémental+Adaptatif



L'aboutissement applicatif est parfaitement adapté aux besoins mais pourrait présenter un aspect artistique très avant-gardiste dans le cas de la métaphore de la Joconde.

Des débuts empiriques

Les concepts basiques posés, observons l'histoire. Certains prétendent faire remonter les origines de certaines méthodes aux années 60. À l'époque, l'informatique existait déjà, sur les premiers ordinateurs dits « scientifiques » pour l'armée et certaines branches de la recherche. Les entreprises se contentaient de matériel dit « classique » qu'il fallait câbler physiquement.

J'ai vécu professionnellement cette époque et je peux assurer qu'aucun chef de projet en informatique de gestion ne s'occupait de méthode.

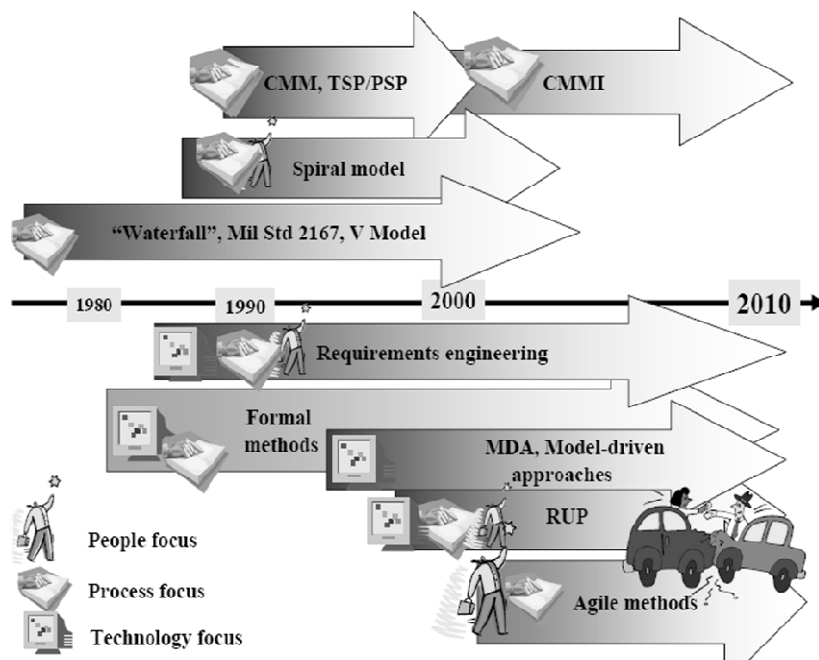


Figure 4. — Vision globale (source NCR 2007)

Le modèle « cascade » et le cycle en « V »

Ensuite, dans les années 70, il est exact que bien avant l'approche « cascade » caractérisée par un cycle en V, qui a marqué le début des grands développements de systèmes d'informations, les programmeurs utilisaient de manière totalement empirique le « codage-test-correction » (rien à voir avec le TDD).

Peut-on en conclure, comme je l'ai vu faire en conférence, que cette forme d'itération était une méthode et représentait les prémices de *Scrum* ? Non bien sûr.

Ce qui caractérise une méthode itérative, c'est sa capacité à planifier une itération de production en termes de fonctionnalités et d'interdépendances.

Premières communications sur l'itératif

C'est en 1986 que commencent vraiment les communications sur ce qui constitue aujourd'hui les bases du génie logiciel itératif, incrémental.

En janvier 1986, Hirotaka Takeuchi et Ikujiro Nonaka, professeurs à l'université de Hitosubashi au Japon, s'appuyaient sur les théories du Lean industriel (Toyota, Xerox, ...) et de l'ingénierie concurrente pour publier dans Harvard Business Review, en une dizaine de pages, sous le titre *The New New Product Development Game*, les fondements d'une métaphore du Rugby appliquée à la conduite de projets. L'idée semble en être restée là officiellement jusqu'en 1996.

Au mois d'août 1986, Barry W. Boehm, de South California University, communiquait officiellement avec *A spiral model of software development and enhancement* sur l'idée d'un nouveau modèle de développement permettant de réduire les risques des projets et d'accroître la qualité applicative : *La Spirale*. Ce n'était pas la première communication traitant du principe itératif incrémental mais c'était la première matérialisation formalisée de sa mise en œuvre.

Cycle Itératif – Incrémental - Adaptatif

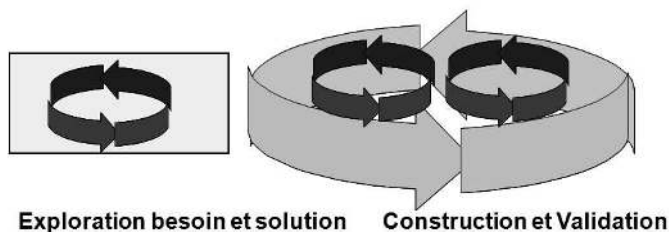


Figure 5. — Itératif, Incrémental, Adaptatif (origine : méthode RAD)

À la fin des années 80, les communications de James Martin (enseignant à l'Université d'Oxford), divulguaient les fondements du développement Itératif-Incrémental-Adaptatif (Figure 5), base des approches Agiles actuelles.

RAD première méthode Agile publiée

Le support de mise en œuvre de la méthode RAD fut publié en 1991 sous la forme d'un guide complet de mise en œuvre : *Rapid Application Development*, édité par McMillan et largement diffusé à l'échelle mondiale. C'est sans conteste l'ouvrage le plus complet qu'il m'a été donné de lire traitant d'une méthode de génie logiciel. Il en justifiait les tenants et les aboutissants, en détaillait les conditions d'utilisation, en précisait les impacts de tous types.

C'est au Canada, entre 1989 et 1993, que je réalisais mes premiers projets avec cette approche hautement coopérative et adaptative, pour Hydro-Québec, puis les laboratoires Abbott et enfin Bell Mobilité Cellulaire.

Ce qui caractérise une méthode adaptative, c'est sa capacité à accepter les changements dans le cadre d'une métrique formelle de ceux-ci.

RAD et DSDM, les évolutions

À partir de cette époque, sous la plume de Jean-Pierre Vickoff, des articles et publications commencèrent à en diffuser les principes en français. Le premier rapport de mise en œuvre fut publié par MGI en 1994 et le premier livre en 1996 par Macmillan l'éditeur de James Martin. Signe des temps, l'évolution des méthodes s'accéléra, mais cela n'autorise pas à en bricoler l'histoire.

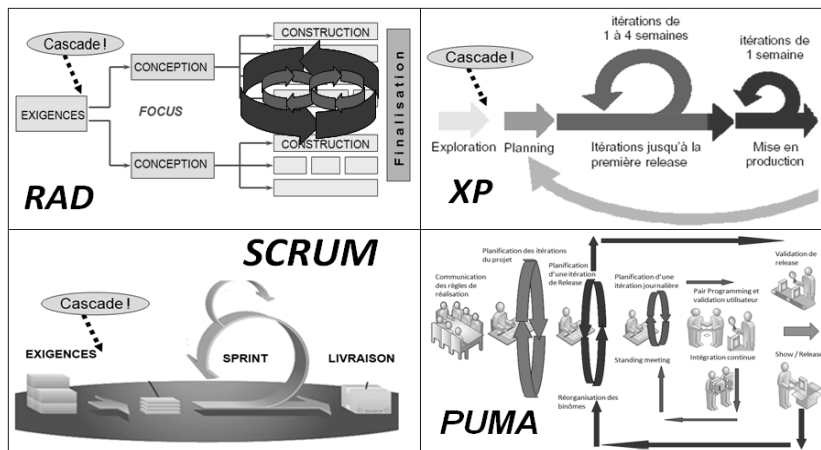


Figure 6. — Les méthodes Agiles sont semi-itératives

La méthode RAD repose sur un cycle de vie semi-itératif, une construction incrémentale et des techniques adaptatives (à l'instar de XP ou de *Scrum* d'ailleurs Figure 6). Cet aspect est souvent mal compris des agilistes néophytes qui pensent que le RAD ne serait pas itératif et que *Scrum* ou XP le serait totalement.

Le mode semi-itératif est indispensable à la plupart des projets. Le mode semi-itératif préserve en début de projet une réflexion minimum sur les contraintes du projet, l'expression globale des exigences, les impacts organisationnels, l'architecture ainsi que l'estimation initiale et la planification des itérations.

En 1995 Jennifer Stapleton publia DSDM la version anglaise du RAD. Un peu plus tard, Yan Stoke en assura la promotion en France.

Scrum, RAD2 et eXtrême Programming

En 1996 Ken Shwarber et Jeff Sutherland reprirent les principes de *The New New Product Development Game* et publièrent les bases de *Scrum*, une méthode générique de conduite de projets n'incluant pas de pratique de génie logiciel.

En mai 1999, la méthode RAD parvenait à la maturité d'une deuxième génération¹. Commandé et publié par le Gartner Group à Jean-Pierre Vickoff, RAD2 s'appuyait sur la formalisation de modes opératoires des techniques utilisées et d'un processus léger et adaptable recherchant la compatibilité avec le CMM niveau 5.

Merise disparue, en face, le seul opposant à défendre l'aspect prédictif et la complétude d'un processus lourdement outillé était RUP (*Rational Unified Process*) dont un des principaux contributeurs était le sympathique Philippe Kruchten passé depuis du « côté Agile de la force ».

RAD2 détaillant, étape après étape, la conduite de projet adaptative. Cette formalisation couvrait les aspects :

- diagnostic de l'organisation et du domaine ;
- mode opératoire des communications ;
- techniques de modélisation adaptative ;
- construction itérative incrémentale adaptative ;
- techniques structurées de qualité du logiciel.

De plus, ce n'était pas seulement le périmètre applicatif qui pouvait servir de variable d'ajustement, mais la modulation de 4 facteurs distincts permettant une planification stratégique fine adaptée aux exigences du projet et de l'application envisagés. La méthode RAD avec son minimum de formalisation des besoins et ses techniques de génie logiciel structurées mais non extrêmes, reste toujours dans ses principes et techniques une voie adaptée aux développements raisonnablement maîtrisés.

En octobre 1999 Kent Beck et Ron Jeffrie qui travaillaient sur un projet de refonte de la paie de Chrysler, officialisèrent la méthode *eXtrem Programming*. La particularité d'XP était de systématiser et de pousser à l'extrême 12 pratiques de qualité du génie.

La phase de Construction du RAD était identique à XP dans son principe, mais moins eXtrême dans la mise en œuvre de techniques de qualité du code. Si les tests systématiques et l'intégration continue étaient exigés, la programmation en binôme par exemple n'était pas imposée sauf pour les parties « sensibles » ou complexes de l'application.

¹ Comme XP et Scrum sont largement publicisés, les rappels sur RAD ont pour but d'apporter une vision historique.

C'est d'ailleurs cette similitude qui m'a autorisé à publier dès 2001 sous le titre de *RAD Construction Agile et XP* « Portant la programmation au rang d'une discipline collective, l'eXtrem Programming propose un ensemble cohérent de techniques apportant des solutions à la grande majorité des problèmes de performance et de qualité en matière de développement d'application. XP peut donc enrichir ou se substituer à la phase Construction (Figure 7) de la méthode RAD ». Par contre, je ne trouvais rien de nouveau dans *Scrum* par rapport à RAD, à l'exception de la systématisation des rétrospectives. Au printemps 2001, Thierry Cross, le fondateur d'XP France et moi-même présentions RAD et XP dans le grand « amphi » de l'UNIL à Lausanne et je modifiais « *on the fly* » ma conférence pour faire état de cette

complémentarité. Comme, à l'époque XP, ne disposait pas de références européennes à présenter, durant la période de question ce furent celles du RAD à la SG et à la Seita, entre autres, qui furent « partagées ».

Une itération RAD comprend sa planification priorisée en terme de fonctionnalités à produire, ses validations permanentes (fonctionnelles et techniques), sa livraison et sa rétrospective (*Focus* ou *Show*).

La notion de prototypage « actif » recouvre la production et la « validation permanente » d'un produit livrable. Les itérations « Jalons ZD » (pour Zéro Défaut) correspondent aux tests, validations et intégrations journalières.

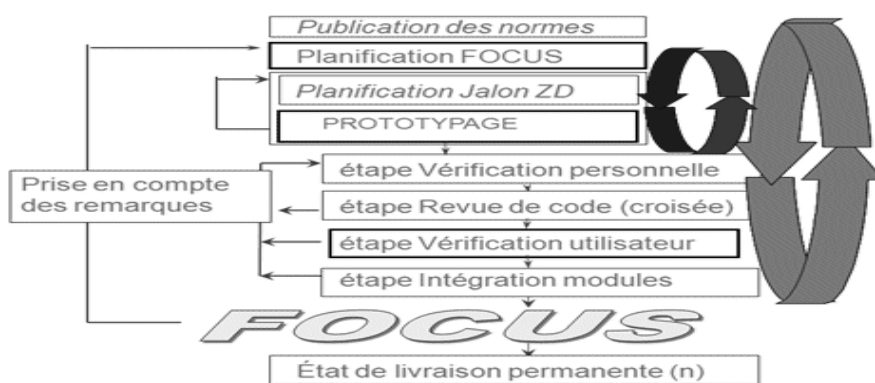


Figure 7. — Méthode RAD (Construction)

PUMA les origines

En septembre 2001 je rédigeais la communication initiale traitant de PUMA (Proposition pour l'Unification des Méthodes Agiles). Sa traduction fut alors expédiée aux tenants du mouvement Agile et aux universités américaines. Initialement, PUMA consistait en une étude de toutes les méthodes Agiles publiées. Le but était d'isoler leurs pratiques communes et surtout de mettre en évidence leurs pratiques différenciatrices.

C'est sur ces bases qu'en 2008, j'ai proposé PUMA Essentiel (*Processus Urbanisant les Méthodes Agiles*), la première méthode Agile de troisième génération, urbanisée et optimisée dont les pratiques sont détaillées dans l'ouvrage « Méthode Agile, Les meilleures pratiques Compréhension et mise en œuvre ».

Conclusion

Les années 80 ont vu chuter IBM qui n'avait pas saisi la translation des préoccupations du matériel vers le logiciel. Les années 90 ont bousculé Microsoft qui n'avait pas anticipé l'importance de l'Internet. Dans le même temps, la mondialisation impose aux entreprises un challenge mortel. Avec le rachat par IBM de Rational, pour citer l'exemple informatique, ce sont les années méthodes qui s'initialisent. L'objectif de PUMA, dernière méthode en date, est d'instrumenter un des futurs possibles. ▲

vickoff@noos.fr