Global Scientific JOURNALS

GSJ: Volume 10, Issue 10, October 2022, Online: ISSN 2320-9186

www.globalscientificjournal.com

ETUDE COMPARATIVE DES DIFFERENTS CYCLES DE VIE DE LOGICIELS

Par

Stéphanie MONGOLO EKUTSHU

Licenciée en Informatique de Gestion, Option Conception des systèmes d'Information à L'ISS/Kin.

Résumé

Il s'agit d'une étude qui fait la comparaison des quelques cycles de vie de certains logiciels.

En effet, cette étude présente d'abord les logiciels et ensuite les avantages et les inconvénients de ces derniers.

Le socle de toute cette étude est les comparaisons faites et présentées pour chaque cas de figures reprises dans un tableau d'une manière détaillée pour une meilleure compréhension. Et, une conclusion a été tirée de notre part.

Abstract

This is a study that compares the few life cycles of some software.

Indeed, this study presents first the software and then the pros and cons of these.

The basis of this whole study is the comparisons made and presented for each case of figures included in a table in a detailed way for a better understanding. And, a conclusion has been drawn from us.

Mots clés

1. Le modèle en cascade,
2. Le modèle en V,
3. Le modèle en spirale,
4. Le modèle par incrément,

5. Le modèle Rad.
6. La qualité du logiciel,
7. Les délais de sa réalisation,
8. Les coûts associés,
9. Les risques

INTRODUCTION

Dans le domaine de l'informatique et notamment dans le domaine du développement des logiciels, on entend par **cycle de vie d'un logiciel** (en anglais *software lifecycle*) une procédure en vue du développement dudit logiciel de sa conception à sa disparition[1].

En effet, le cycle de vie d'un logiciel est la description d'un processus couvrant les phases de:

Création du logiciel, Distribution du logiciel sur un marché, Disparition du logiciel.

Ainsi, l'objectif du cycle de vie d'un logiciel est de ***définir des jalons intermédiaires permettant la validation du développement logiciel***, c'est-à-dire :

· La conformité du logiciel avec les besoins exprimés, La vérification du processus de développement (en d'autres termes l'adéquation des méthodes mises en œuvre).

Le cycle de vie d'un logiciel permet donc de détecter les erreurs au plus tôt et ainsi de maîtriser :

. La qualité du logiciel, les délais de sa réalisation, les coûts associés et les risques.

En tant que procédure, le cycle de vie d'un logiciel est composé de différentes étapes. Ainsi, en général, le cycle de vie d'un logiciel comprend au minimum les étapes suivantes :

Analyse des besoins, Conception globale, Conception détaillée, Codification, Tests, Recette, Intégration, Documentation, Déploiement, Maintenance et évolution

La séquence et la présence de chacune de ces activités (étapes) dans le cycle de vie dépend du choix d'un Modèle de Cycle de Vie (MCV) entre le client et l'équipe de développement.

[1] John McDermid et Knut Ripken. Life cycle support in the ADA environment. University Press, 1984.

En effet, il existe plusieurs MCV. Il est donc important pour une équipe de développement de logiciels de bien choisir son MCV[2]. Car, ce dernier prend en compte, en plus des aspects techniques, l'organisation et les aspects humains[3]. Ainsi, chaque équipe de développement de logiciels se doit de comparer les différents MCV en vue de choisir celui qui est adapté à leur contexte.

Comme nous l'avons déjà dit, il existe une multitude de MCV (en cascade, en V, par incrément, en B, en spirale, etc.). Face à cette multitude de modèles, nous avons fait le choix de faire la comparaison sur cinq (05) cycles de vie à savoir :

Le modèle en cascade, Le modèle en V, Le modèle en spirale, Le modèle par incrément, Le modèle Rad.

Le but de cette étude est de connaître et pouvoir comparer les impacts environnementaux d'un système tout au long de son cycle de vie, de l'extraction, par exemple, des matières premières nécessaires à sa fabrication à son traitement en fin de vie (mise en décharge, recyclage...), en passant par ses phases d'usage, d'entretien et, peut-être, de transport.

Sur ce voici les questions qui nous préoccupent :

- Comment faire pour quantifier les contributions aux impacts environnementaux d'un système (par étape de cycle de vie ou par sous-système : composants, matériaux utilisés, procédés) afin d'en dégager des pistes d'écoconception ou d'amélioration du bilan environnemental du système ?
- Comment faire pour comparer du point de vue environnemental deux systèmes ayant la même fonction, à quantité de service rendu égale (voir la notion d'unité fonctionnelle) ?

L'étude comparative de différents cycles de vie est une méthode d'évaluation normalisée (ISO 14040 et 14044) permettant de réaliser un bilan environnemental multicritère et multi-étage d'un système (produit, service, entreprise ou procédé) sur l'ensemble de son cycle de vie.

Elle va ainsi nous permettre :

- de quantifier les contributions aux impacts environnementaux d'un système (par étape de cycle de vie ou par sous-système : composants, matériaux utilisés, procédés) afin d'en dégager des pistes d'écoconception ou d'amélioration du bilan environnemental du système ;

[2] Philippe Kruchten. The Rational Unified Process: An Introduction. Addison-Wesley, Longman Publishing, Co., Inc. Boston, Massachusetts, 2000.

[3] Winston W. Royce. Managing the Development of Large Software Systems. IEEE Wescon, p. 1-9, 1970

- de comparer du point de vue environnemental deux systèmes ayant la même fonction, à quantité de service rendu égale (voir la notion d'unité fonctionnelle).

Aussi cette étude est :

- une procédure, c'est-à-dire une suite d'étapes standardisées ;
- un modèle mathématique de transformations permettant de faire correspondre des flux à leurs impacts environnementaux.

Suivant les quatre ou cinq phases (suivant que l'on se réfère à l'ISO groupant objectifs et champ d'étude ou à l'ILCD les séparant), il faut d'abord définir l'objectif de l'étude, puis choisir en conséquence l'objet à étudier. Il faut ensuite étudier les systèmes impliqués par les produits à comparer, puis les flux de matières et d'énergie, puis les impacts environnementaux connus, pour chaque étape du cycle de vie. Il faut enfin pondérer ces impacts, habituellement sous la forme d'unités de charge écologique (UCE).

- **Revue critique :** Les bonnes pratiques impliquent d'intégrer une "*revue critique*", qui est un processus de vérification par un tiers que l' *« Analyse du Cycle de Vie satisfait aux exigences de méthodologie, de données, d'interprétation et de communication et si elle est conforme aux principes de la méthodologie tels qu'ils sont indiqués par les normes en vigueur »*. Le tiers peut être un expert interne ou externe ou un comité des parties intéressées. Il vérifie la cohérence des méthodes relativement aux normes en vigueur, leur validité scientifique et technique, et si les données sont pertinentes pour atteindre les objectifs de l'étude. Il vérifie aussi que le rapport d'étude soit transparent, vérifiable et cohérent, ainsi que « le reflet des interprétations au vu des limitations identifiées et des objectifs de l'étude ».

Les premiers travaux relevant de l'ACV datent des années 1970. Ces origines sont relatées dans les travaux de 1996 de Boustead et Hunt, respectivement au Royaume-Uni et aux États-Unis d'Amérique.

- Les premières méthodes d'évaluation des impacts environnementaux, telles que nous les connaissons toujours aujourd'hui sous la forme de l'ACV, remontent à 1992, soit environ vingt ans après les prémisses de la méthodologie. Mentionnons l'EPS (*Environmental Priority Strategy*), basée sur une modélisation orientée dommage exprimée en valeur monétaire, le modèle *Swiss Ecoscarcity Ecopoints* fondé sur le « principe de la distance à la cible » et la méthode de 92 du CML avec une orientation "problème".

Mais avant de faire une comparaison sur ces MCV, il est important de les décrire au préalable.

A. DESCRIPTION DES MCV

1. LE CYCLE DE VIE EN CASCADE (Waterfalls)

Dans ce modèle, le principe est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Les résultats sont définis sur la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés satisfaisants. Ce modèle, développé dans les années 1970 par W. ROYCE a servi pendant des années de modèle de référence.

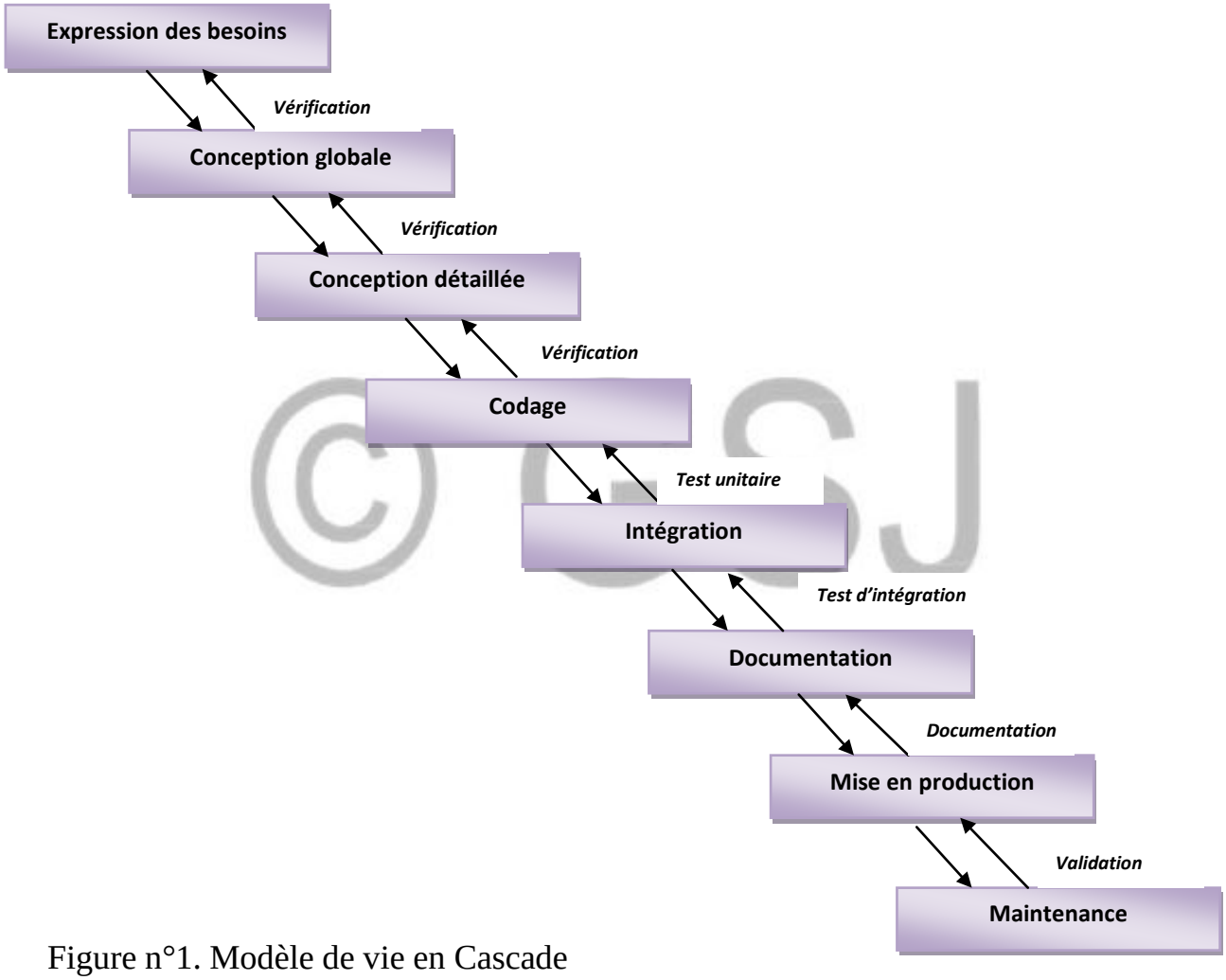

Figure n°1. Modèle de vie en Cascade

2. LE CYCLE DE VIE EN V

Il s'agit d'un modèle en cascade dans lequel le développement des tests et du logiciel sont effectués de manière synchrone.

Le principe de ce modèle est qu'avec toute décomposition doit être décrite la recomposition et que toute description d'un composant est accompagnée de tests qui permettront de s'assurer qu'il correspond à sa description.

C’est le cycle de vie le plus connu et certainement le plus utilisé.

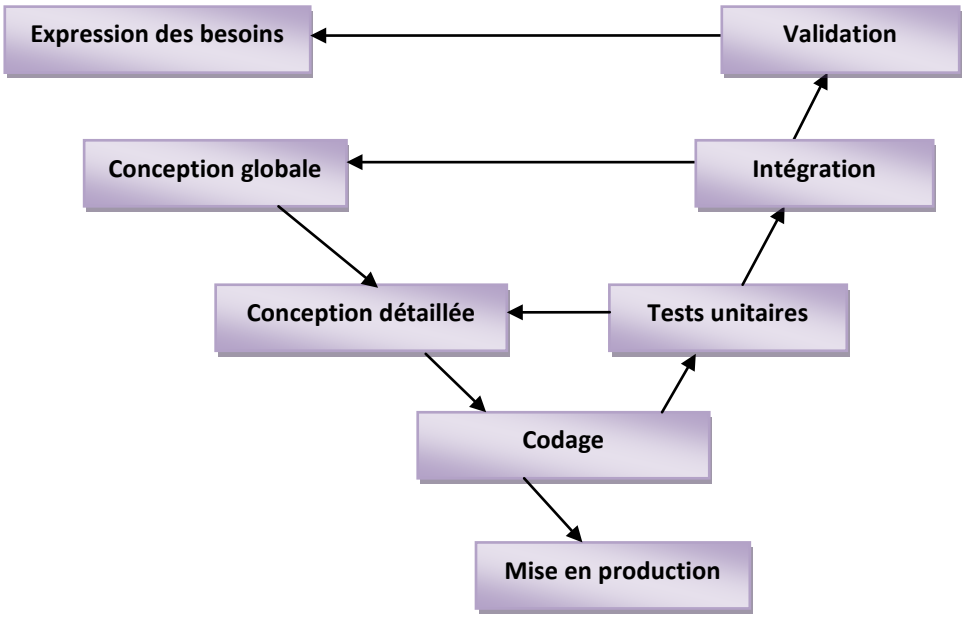

Figure n°2. Modèle de vie en V

3. LE CYCLE DE VIE EN SPIRALE[4]

Proposé par B. Boehm en 1988, ce modèle est beaucoup plus général que le précédent. Il met l’accent sur l’activité d’analyse des risques. Chaque cycle de la spirale se déroule en quatre phases :

· L’analyse préliminaire des besoins, des objectifs du cycle, des alternatives pour les atteindre et des contraintes ;
· La conception, l’analyse des risques, l’évaluation des alternatives et, éventuellement le maquettage[5] ;
· La réalisation, le développement et la vérification de la solution retenue, un modèle « classique » (cascade ou en V) peut être utilisé ici ;
· La validation et la revue des résultats et vérification du cycle suivant.

[4] Barry W. Boehm. A Spiral Model of Software Development and Enhancement. IEEE Computer, 21(5), p. 61-72, 1988.

[5] Kent Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, Longman Publishing Co., Inc. Boston, Massachusetts, 1999.

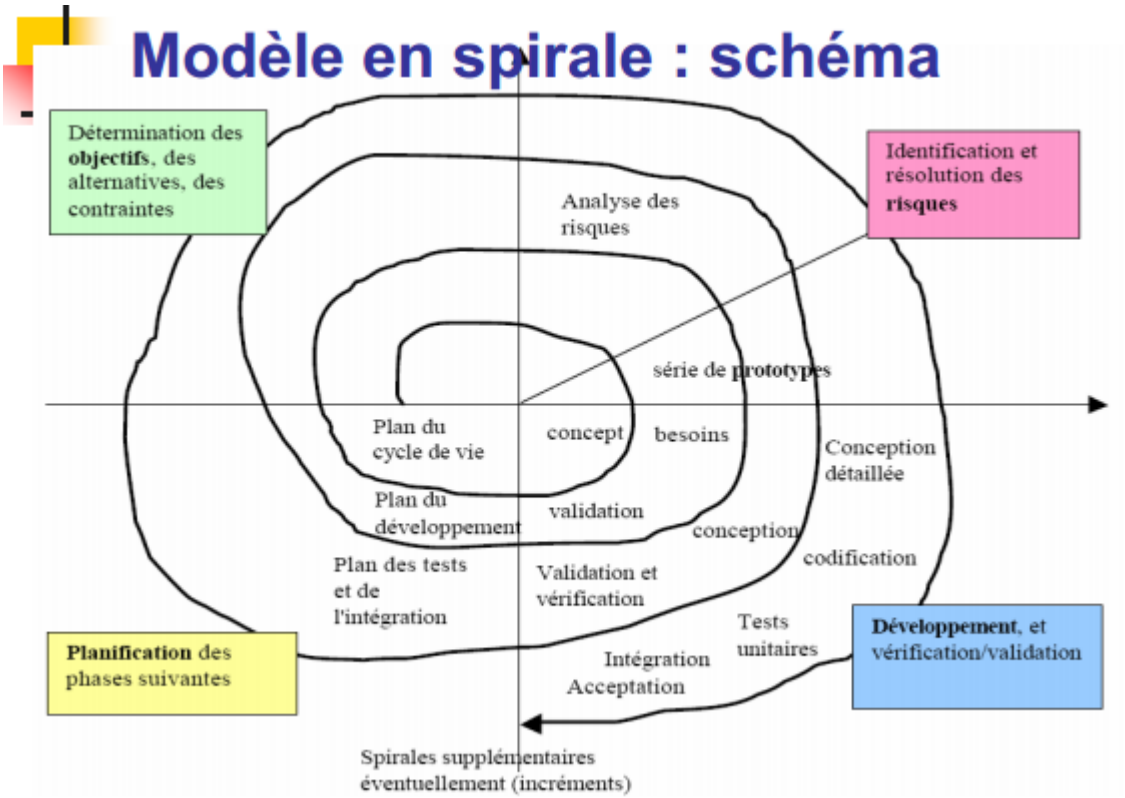

Figure n°3. Modèle de vie en Spirale

4. LE CYCLE PAR INCREMENT

Dans les modèles précédents, un logiciel est décomposé en composants développés séparément et intégrés à la fin du processus.

Dans les modèles par incrément un seul ensemble de composants est développé à la fois : des incréments viennent s'intégrer à un noyau de logiciel développé au préalable[6]. Chaque incrément est développé selon l'un des modèles précédents. En bref, le modèle par incrément découpe le système en domaines qui sont traités individuellement sur le modèle en cascade.

[6] Ken Schwaber et Mike Beedle. Agile Software Development with SCRUM. Prentice Hall, Upper Saddle River, New Jersey, 2001.

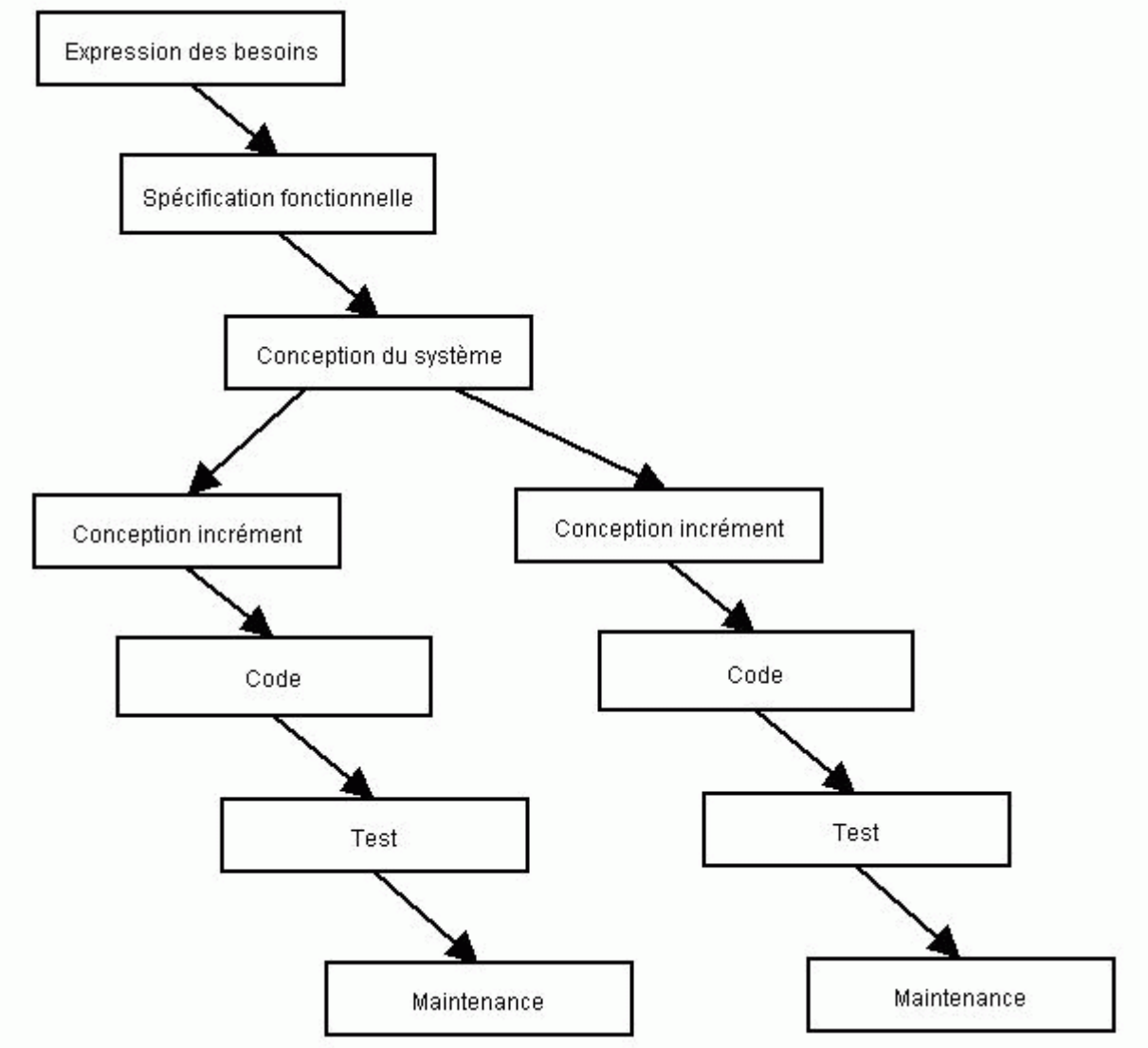

Figure n°4. Modèle de vie en Incrément

5. LE CYCLE DE VIE RAD

Le cycle de vie RAD (Rapid Application Development) est employé lorsque l'implication forte de l'utilisateur est nécessaire. Il permet de construire le système avec l'utilisateur. Il a été introduit par James Martin et comporte 3 phases[7] :

Cadrage, qui couvre l'analyse des besoins, le périmètre et la planification de l'itération

Conception c.à.d. la conception, description et organisation des données et des traitements avec les utilisateurs.

Construction qui couvre le développement et les tests.

[7] James Martin. Rapid Application Development. Macmillan Coll. Div., 1991.

Structure de la méthode

Le cycle RAD est en fait semi-itératif

Le RAD préconise la formation d'une équipe de développement particulière : le SWAT. Cette équipe est autonome, spécialement formée, concrètement motivée et outillée. Elle se compose essentiellement d'un profil unique de concepteurs-développeurs formés à des spécialités techniques complémentaires. Le rôle de chef de projet, n'est ni prohibé, ni obligatoire. Par contre, les décisions concernant l'organisation du projet sont consensuelles. L'équipe travaille avec les utilisateurs et, généralement avec un animateur, dans une salle dédiée, isolée, spécialement équipée dans le style war room, où les murs sont utilisés pour afficher un « radiateur d'information » (une forme de cockpit de gestion de projet)[8].

Sur le plan des principes de mise en opération, la méthode RAD implique :

Un cycle de développement sécurisant et court fondé sur un phasage simple : Cadrage, Design, Construction et l'absolu respect d'une dimension temporelle (90 jours optimum, 120 jours maximum) [Martin 1991] (*figure : Le cycle RAD est en fait semi-itératif*) ;

Une architecture de communication engageant des groupes de travail de structure et de composition variables selon les besoins des phases et respectant un mode opératoire précis structuré en trois étapes : pré-session, session, post-session [Mucchielli 1987]. ;

Des méthodes, techniques et outils permettant de définir et d'appliquer des choix portant sur quatre natures d'objectifs potentiellement contradictoires : budget, délais, qualité technique, qualité fonctionnelle et visibilité [Vickoff 1999] ;

Une architecture de conception s'appuyant sur les techniques de l'objet et particulièrement sur celles qui permettent une conception «en vue de modifications» [McCarty 1997] ;

Une architecture de réalisation qui impose, pour garantir la qualité technique, des normes minimales, des revues de projet, des jalons zéro-défaut et qui recommande, pour garantir la qualité fonctionnelle, le prototypage actif et les *focus* de visibilité [McConnell 1996].

[8] http://www.rad.fr/phasprin.htm

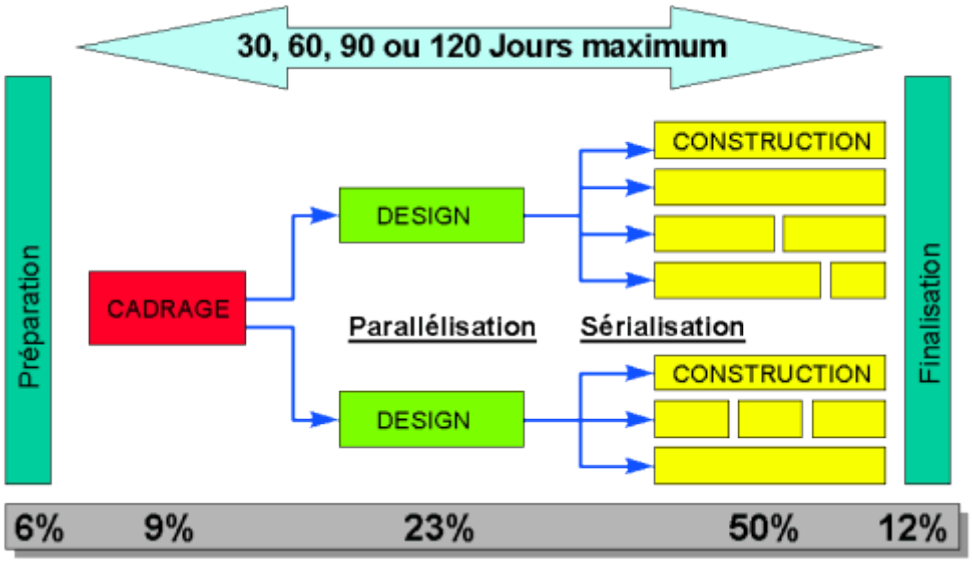

Figure n°5. Modèle de vie en RAD (Rapid Application Development)

Etude de cas : Exemples d'application

Cas de bâtiment en France

Tout élément bâti peut théoriquement faire l'objet d'une ACV, avec d'autant plus de complexité que la construction et ses usages sont complexes. En France 43 % de l'énergie finale (> 100 Mtep/an) est consommée par les bâtiments, dont la construction et démolition génèrent plus de 40 millions de tonnes de déchets, encore peu et mal recyclés, plaçant pour les émissions de « gaz à effet de serre » le résidentiel-tertiaire (24 %) se place devant les transports (23 %), l'industrie et l'agriculture. Suite aux demandes de performanciels et d'évaluation environnementale de la HQE, la réflexion a porté sur les bâtiments habités, surtout à partir des années 2000. La loi Grenelle 2 (2010) pousse le secteur du bâtiment vers les ECV, via de nouveaux articles du Code de la construction et de l'habitation, qui visent un label environnemental intégrant l'ensemble du cycle de vie du bâtiment.

En octobre 2011, un projet de décret et d'arrêté sur la déclaration environnementale des produits de construction devrait pouvoir aboutir à l'obligation d'ACV pour les acteurs voulant communiquer sur les impacts environnementaux de leurs constructions. Le décret est désormais publié au JO : Décret n°2013-1264 du 23 décembre 2013 relatif à la déclaration environnementale de certains produits de construction destinés à un usage dans les ouvrages de bâtiment. L'ACV est alors multi-échelle ; des produits et éléments de bases (fiche de déclaration environnementale et sanitaire ou FDES, PEP, EPD accessibles depuis la *Base nationale française de référence INIES*) aux bâtiments eux-mêmes (quantification des performances environnementales) et parfois à l'échelle d'îlots, d'éco quartier ou d'infrastructures. Cette fiche est remplie sous la responsabilité des fabricants (ou syndicat professionnel) du produit. La norme pr EN 15804 (qui a partiellement remplacé la NF P01-010 en

2012) donne la méthode d'obtention et le format de déclaration des informations environnementales et sanitaires.

Des outils informatiques d'aide au calcul devraient encore être améliorés, testés, validés et normalisés (ex : norme ISO 15392:2008 sur le développement durable dans la construction, norme européenne pr EN 15804 sur les déclarations environnementales des produits de construction qui devrait remplacer la NF P01-010 en 2012, norme EN 15978 devant remplacer la XP P01-020-3 en 2012 sur les indicateurs et méthodes de calcul des impacts environnementaux des bâtiments). En 2011, plusieurs approches devaient pouvoir donner des résultats différents.

Toutefois, la convergence des référentiels d'évaluation (échelle produit : NF EN 15804, PCR PEP v3, ISO 21930. Echelle bâtiment NF EN 15978) est en marche.

Les travaux d'articulation d'ACV avec l'échelle des quartiers et de la ville étaient également portés au sein de la commission AFNOR Aménagement durable et résilient : AFNOR/ADR et de l'institut Efficacity : l'institut de R&D pour la transition énergétique de la ville.

L'ACV se retrouve aujourd'hui être l'une des méthodes d'évaluation environnementale préconisée notamment par la Commission Européenne au sein de ses feuilles de route « Resource Efficiency » et « Circular Economy ».

Voici quelques chiffres comparés des impacts entre les infrastructures et les procédés (selon Éco-indicator 99, score unique (en pt. hiérarchisé).

	Impact infrastructures	Impact procédé utilisation
Fonte	8 %	92 %
Papier journal	13 %	87 %
Contreplaqué	3 %	97 %
Patate bio	96 %	4 %
Électricité éolienne	99 %	1 %
Hydro-électricité	92 %	8 %
Électricité charbon	5 %	95 %
Nucléaire	32 %	68 %
Énergie diesel	4 %	96 %

Énergie gaz naturel	1 %	99 %
Camion	33 %	67 %
Voiture	24 %	76 %
Incinération	3 %	97 %
Recyclage	19 %	81 %
Enfouissement	96 %	4 %

Tableau n°1 : Quelques chiffres comparés des impacts entre les infrastructures et les procédés

Nous venons de faire la description de ces cinq (05) MCV. Nous allons maintenant faire les comparer selon les critères ci-après :

· Forces, Faiblesses, et A quel moment utiliser tel ou tel MCV.

B. TABLEAU COMPARATIF DE DIFFERENTS CYCLES DE VIE

CYCLE DE VIE	FORCES	FAIBLESSES	QUAND UTILISER
CASCADE (WATERFALLS)	· Facile à comprendre et à utiliser · Adapté pour une équipe inexpérimentée · Les limites de chaque étape sont visibles · Facilite un management du projet · La définition des besoins est non-évolutive · La qualité prime sur le coût	· Tous les besoins doivent être bien spécifiés au départ · Donne une fausse impression de l'avancée des travaux · Pas d'interaction entre les phases de développement · L'intégration n'a lieu qu'à la fin du cycle · Le client peut se retrouver non satisfait · Pas de retour en arrière d'une phase à l'autre	· La phase de spécification a été très bien faite · La définition du produit est stable · Il s'agit d'une nouvelle version d'un produit existant · L'implantation d'un produit existant sur une nouvelle plate-forme · Une bonne maîtrise de la technologie
EN V	· Facile à utiliser · Les tests sont effectués à chaque étape · Le contrôle se fait progressivement à	· Une mauvaise prise en compte des évènements concurrents · Le processus n'est pas itératif · Une mauvaise prise en	· Les spécifications de besoins doivent être bien faites · La solution à développer et la technologie à utiliser

	chaque étape · Les phases de validation sont prises en main très tôt dans le processus de développement	compte des changements de la spécification des besoins · Ne contient pas les activités d'analyses de risques	doivent être parfaitement connues · Les changements doivent être faits avant l'analyse · Excellent pour les systèmes requérant une grande sûreté
EN SPIRALE	· Sans coût élevé, donne des indications sur les risques majeurs · Les fonctions critiques à haut risque sont développées en premier lieu · La conception ne doit pas forcément être terminée · Les utilisateurs finaux sont intimement associés à toutes les étapes du développement · Le développement se fait en interaction avec les clients · L'évolution du coût de développement est sous contrôle · Les utilisateurs ont dès le départ une vue globale du système	· Le temps consacré à l'évaluation des risques est trop élevé pour des petits projets · Le temps mis à planifier, évaluer les risques, fixer les objectifs, les prototypes peut être excessif · Ce modèle est complexe · Une expertise en évaluation des risques est nécessaire · La spirale peut être infinie · les développeurs travaillent par intermittence · il est difficile de définir les objectifs et les points de validation intermédiaires entre les différentes étapes	· les coûts et l'évaluation des risques est important · pour des projets à risque au moins moyennement élevé · pour des projets à long terme dont les financements peuvent varier · les utilisateurs ne définissent pas clairement leurs besoins · la spécification des besoins est complexe · il s'agit d'une nouvelle gamme de produits · des changements significatifs peuvent intervenir à cause par exemple de l'évolution de la recherche ou de l'exploration
PAR INCREMENT	· Le client peut valider chaque étape du processus · Utilise la méthode Diviser Pour Régner · La délivrance du produit est rapide · Le coût de lancement du projet est moindre · Un produit exploitable	· Requière une bonne planification et une bonne conception · Requière la définition complète des fonctionnalités du système pour une définition des différents incréments · Le coût total du développement du	· Sur un projet utilisant de nouvelles technologies · Sur des projets ayant une durée de développement assez longue · Le besoin pour les développeurs de connaître dès le départ les

	peut être délivré à tout moment · Les clients obtiennent les fonctionnalités majeures du système très tôt · Le risque du changement des besoins est minimal · Développe les fonctions primordiales dès le départ	système n'est pas négligeable · Les différentes interfaces doivent être bien définies	fonctionnalités majeures · La plupart des fonctionnalités doivent être connues au départ mais certaines ne seront utilisées que plus tard · Les risques, le financement, la complexité du logiciel, l'élaboration ou les besoins pour les bénéfices maximum dès le départ
EN RAD	. la spécialisation des rôles ; . l'instrumentation des communications ; . l'organisation des divers types de réunions ; . le groupe de facilitation et de rapport ; . les « raccourcis méthodologiques » de modélisation ; . l'architecture de réalisation (imbrication des itérations) ; . la formalisation de processus légers de mise en œuvre.	. L'auteur insiste sur la réutilisabilité. Or, pour qu'un composant soit réutilisable, il doit être utilisable dans des contextes différents par des applications différentes, ou par des modules différents. . Ceci implique soit une réflexion en amont de sa construction, soit une « généralisation » de ce composant à posteriori pour l'ouvrir à d'autres utilisations que celle pour laquelle il a été conçu. . Si les délais sont très courts, c'est bien ce travail de fond qui sera le premier sacrifié. . Si un développeur est pris par le temps, on ne voit pas quelle raison le pousserait à chercher à rendre un composant réutilisable à plus long terme, donc à accomplir une tâche dont il ne verra jamais les fruits. . Il n'y a pas de miracles. La réutilisabilité	

		demande un investissement à long terme. .Elle doit être gérée. . Elle doit faire partie des objectifs de l'équipe de développement. À moins que le développement consiste en un assemblage de composants préexistants, conçus et développés en dehors du contexte du RAD. Et nous retombons alors dans la problématique de la conception et de la réalisation par objets	

Tableau n°2. Tableau comparatif de différents cycles de vie

CONCLUSION

Il n'est pas facile de comparer les différents cycles de vie des différents logiciels. Chacun a ses forces, ses faiblesses et un cadre d'utilisation bien déterminé.

Nous avons démontré l'importance de mener cette étude pour connaître et faire connaître les avantages tout comme les inconvénients de certaines procédures dans le choix d'un ou l'autre logiciel pour le bon fonctionnement de son système d'information.

Tenant compte de l'immensité et de la profondeur de cet océan qu'est un logiciel suivant son cycle de vie (dès sa conception jusqu'à sa fin de vie) il était important pour nous de présenter cette petite réflexion pour éclairer la lanterne de plus d'un d'entre nous.

Il faut noter qu'il y a plusieurs cycles de vie dont on fait allusion dans cet article.

Malgré tout, nous constatons une plus grande utilisation du cycle en V par la plupart des équipes de développement.

L'analyse de cycle de vie offre une vision globale de l'impact environnemental d'une filière, permet de prévoir certains transferts de pollution, d'évaluer quel type d'impact environnemental est dominant dans la réalisation d'un produit et quelles étapes (étape de production, utilisation, mise au rebut) ou quels éléments

particuliers du produit y contribuent le plus. Ceci est obtenu par une démarche aussi exhaustive que possible et selon une démarche clairement documentée. Cette méthode permet également une mise en perspective des différents types d'impacts plutôt que de se limiter à un type d'impact particulier.

C'est également un outil utile pour faire des choix autant à portée globale (choix d'une politique environnementale, comme l'intérêt du recyclage de certains produits) que locale (choix de design et de production pour un produit).

Cependant nombre d'obstacles font que l'analyse du cycle de vie ne sera jamais un outil universel. D'abord il est quasi impossible d'obtenir l'intégralité des flux utilisés pour un produit, il faut donc se contenter de données parfois limitées et faire appel à des données génériques, donc manquant de précision.

Dans les logiciels d'analyse de cycle de vie actuels, les processus sont généralement régionalisés (contrairement aux impacts qui ont lieu de façon géographiquement indifférenciée et ne dépendent pas de la région ou des régions où a/ont lieu le cycle de vie): il existe généralement plusieurs instances pour chaque processus, en fonction du lieu d'utilisation. Par exemple, pour un même logiciel d'analyse de cycle de vie, il existe plusieurs processus de production d'électricité par le biais de centrales à charbon, pour différents pays, et ces centrales ont des profils d'émission sensiblement différents d'un pays à un autre. Cependant, toutes les régions du monde ne sont pas représentées pour un même type de processus. Il est donc souvent difficile, voire impossible, de réaliser une analyse de cycle de vie qui tienne complètement et parfaitement compte des particularités et du contexte de chaque pays. En revanche, tant que les processus représentatifs requis existent, il est facile et rapide de remplacer un processus par un autre dans une même analyse de cycle de vie. Cela rend aisé, dans la limite de la disponibilité des processus adéquats, la modification d'une analyse de cycle de vie en vue de l'adapter à la réalité contextuelle d'un autre pays ou d'une autre région.

Par ailleurs, plusieurs choix méthodologiques demeurent assez subjectifs comme les choix d'imputation et les méthodes de caractérisation des impacts, de normalisation et de pondération s'ils sont utilisés. Il n'est pas rare, dans le cadre d'une comparaison, de voir le classement entre plusieurs produits être inversé selon la méthode d'évaluation choisie et ce juste au niveau de la caractérisation.

Plusieurs auteurs plaident aussi pour une réévaluation de la notion de ressources naturelles dans l'ACV.

En conclusion, l'étude comparative de différents cycles de vie des logiciels présente de nombreux intérêts. Toutefois les résultats à eux seuls peuvent toujours être contestables selon les choix méthodologiques réalisés. Par conséquent les valeurs obtenues peuvent difficilement être utilisées par le grand public et nécessitent d'être étudiées en détail.

BIBLIOGRAPHIE SOMMAIRE

1. Barry W. Boehm. A Spiral Model of Software Development and Enhancement. IEEE Computer, 21(5), p. 61-72, 1988.
2. Chen, I. C., Fukushima, Y., Kikuchi, Y., & Hirao, M. (2012). *A graphical representation for consequential life cycle assessment of future technologies*, Part 1: methodological framework. The International Journal of Life Cycle Assessment, 17(2), 119-125.
3. Dandres, T., Gaudreault, C., Tirado-Seco, P., et Samson, R. (2011). *Assessing non-marginal variations with consequential LCA: Application to European energy sector*. Renewable and Sustainable Energy Reviews, 15(6), 3121-3132 * Earles, J., et Halog, A. (2011). *Consequential life cycle assessment : a review*. The InternationalJournal of Life Cycle Assessment, 16(5), 445-453.
4. Dandres, Thomas (2012) Université de Montréal, http://publications.polymtl.ca/881/1/2012_ThomasDandres.pdf *Développement d'une méthode d'analyse du cycle de vie conséquentielle ; Prospective macroscopique : évaluation d'une politique de bioénergie dans l'union européenne à l'horizon 2025*, Département de génie chimique, école polytechnique de Montréal (Thèse), PDF 241 p.
5. Ekvall, T., et Andrae, A. S. G. (2006). *Attributional and consequential environmental assessmentof the shift to lead-free solders*. International Journal of Life Cycle Assessment, 11(5), 10.
6. Ekvall, T., et Weidema, B. P. (2004). *System boundaries and input data in consequential life cycle inventory analysis.International* ; Journal of Life Cycle Assessment, 9(3), 11.
7. European Commission - Joint Research Centre (JRC) - Institute for Environment and Sustainability: International Reference Life Cycle Data System (ILCD) Handbook - General guide for Life Cycle Assessment - Detailed guidance. First edition March 2010. EUR 24708 EN. Luxembourg. Publications Office of the European Union; 2010.
8. Hamelin, L., Wesnaes, M. S., Wenzel, H., et Petersen, B. r. M. (2011). *Environmental Consequences of Future Biogas Technologies Based on Separated Slurry*. Environmental Science & Technology, 45(13), 5869-5877.
9. http://www.rad.fr/phasprin.htm
10. James Martin. Rapid Application Development. Macmillan Coll. Div., 1991.
11. John McDermid et Knut Ripken. Life cycle support in the ADA environment. University Press, 1984.
12. JRC, E., 2011. ILCD Handbook: Recommendations for Life Cycle Impact Assessment - based on existing environmental impact assessment models and factors. Publications Office of the European Union.

13. JRC, E., 2012. JRC Reference Report ILCD Handbook - Towards more sustainable production and consumption for a resource-efficient Europe. Publications Office of the European Union.
14. Ken Schwaber et Mike Beedle. Agile Software Development with SCRUM. Prentice Hall, Upper Saddle River, New Jersey, 2001.
15. Kent Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, Longman Publishing Co., Inc. Boston, Massachusetts, 1999.
16. Lesage, P., Deschenes, L., et Samson, R. (2007). *Evaluating holistic environmental consequencesof brownfield management options using consequential life cycle assessment for different perspectives*. Environmental Management, 40, 15.
17. Olivier Jolliet, Myriam Saadé-Sbeih, Pierre Crettaz, Nicole Jolliet-Gavin et Shanna Shaked, *Analyse du cycle de vie : Comprendre et réaliser un écobilan*, Lausanne, Presses polytechniques et universitaires romandes, coll. « Science & ingénierie de l'environnement », 2017, 352 p. (ISBN 9782889151356, présentation en ligne
18. Pehnt, M., Oeser, M., et Swider, D. J. (2008). *Consequential environmental system analysis of expected offshore wind electricity production in Germany*. Energy, 33, 13.
19. Philippe Kruchten. The Rational Unified Process: An Introduction. Addison-Wesley, Longman Publishing, Co., Inc. Boston, Massachusetts, 2000.
20. Reinhard, J., et Zah, R. (2009). *Global environmental consequences of increased biodiesel consumption in Switzerland: consequential life cycle assessment.Journal of Cleaner Production*, 17 (Supplement 1), S46-S56.
21. Reinhard, J., et Zah, R. (2011). *Consequential life cycle assessment of the environmental impactsof an increased rapemethylester (RME) production inSwitzerland. Biomass andBioenergy*, In Press, Corrected Proof.
22. Sanden, B. A., et Karlstrom, M. (2007). *Positive and negative feedback in consequential life-cycle assessment.* Journal of Cleaner Production, 15, 13* Schmidt, J. H. (2008). *System delimitation in agricultural consequential LCA. Outline ofmethodology and illustrative case study of wheat in Denmark*. International Journal ofLife Cycle Assessment, 13(4), 15.
23. Thomassen, M. A., Dalgaard, R., Heijung s, R., et Boer, I. d. (2008). *Attributional and consequential LCA of milk production*. International Journal of Life Cycle Assessment, 13 (4), 11
24. Weidema, B. P. (2010). *LCA masterclass (Discussion about the use of general equilibrium modelin consequential life cycle assessment* ed.). Montreal.
25. Weidema, B. P., Frees, N., et Nielsen, A.-M. (1999). *Marginal production technologies for life cycle inventories*. International Journal of Life Cycle Assessment, 4(1), 9.
26. Winston W. Royce. Managing the Development of Large Software Systems. IEEE Wescon, p. 1-9, 1970.