

CYCLES DE VIE

Les cycles de vie les plus importants et les plus couramment utilisés. Après une présentation des cycles classiques comme ceux en Cascade ou en V, seront présentées des méthodes plus récentes basées sur l'agilité.

Les Classiques

Cycle en Cascade

Le cycle en cascade est typiquement un cycle de développement prédictif. Provenant du bâtiment, il part du principe que la construction nécessite, en général, un enchaînement logique ; la couverture d'une maison ne peut pas être effectuée sans avoir préalablement fait les fondations. Il définit une démarche de développement séquentiel (figure 2.9) où chaque phase conduit à la production d'un ou plusieurs livrable(s) qui doivent être validés avant d'être utilisés lors de la phase suivante.

Le modèle en cascade nécessite la définition d'un planning détaillé qui énonce toutes les étapes et réalisations attendues. Différentes activités d'analyse, de conception, d'implantation, de tests et d'intégration sont effectuées afin de converger vers l'obtention du système logiciel final.

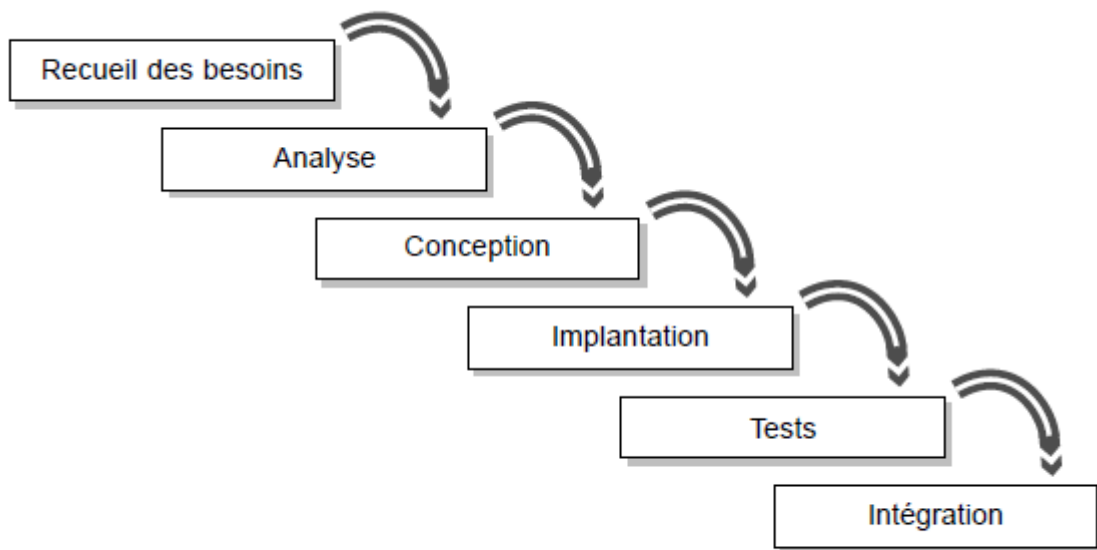


FIG. 2.9 – Modèle du cycle en cascade

Initialement, le modèle en cascade est un cycle de développement purement séquentiel, cependant, diverses possibilités d'itération ou de retour vers les phases amont ont ensuite été intégrées au modèle. Ces itérations permettent de vérifier les produits obtenus au fil du développement et ainsi fournir plus de souplesse à la conception.

Cycle en V

Le cycle en V est l'un des cycles les plus connus et utilisés (figure 2.10). C'est un cycle de type prédictif qui a été défini pour remédier aux lacunes du cycle en cascade qui manque de réactivité face aux erreurs découvertes lors de la conception, du développement ou encore de l'analyse. La structure en V du cycle a l'avantage de mettre en vis à vis les activités de développement et de tests permettant de mieux préciser les documents à partager entre ces phases, notamment les rapports de tests et les modifications qu'il est nécessaire d'apporter pour corriger les erreurs. Ainsi, lors de la phase montante du cycle, toutes les réalisations doivent être testées et validées.

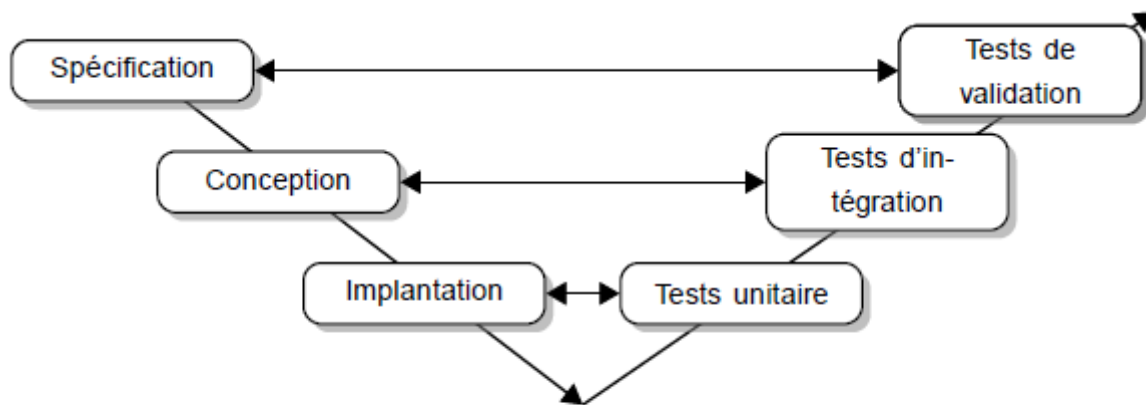


FIG. 2.10 – Cycle en V

Depuis les années 80, le cycle en V est considéré comme un standard du développement logiciel et de la gestion de projet dans les industries. A la suite du cycle en V sont apparues diverses variantes telles que, par exemple, le cycle en W qui propose d'effectuer deux cycles en V successivement, le premier servant à la conception d'un prototype de l'application, le second à construire l'application finale.

Cycle en Spirale

Défini par Barry Boehm, le cycle en spirale est une approche itérative du cycle de développement en V. Ainsi, il reprend l'essentiel des concepts en s'articulant autour de quatre phases importantes (figure 2.12) : la détermination des objectifs, la détermination des risques, le développement et les tests et enfin la planification de l'itération suivante.

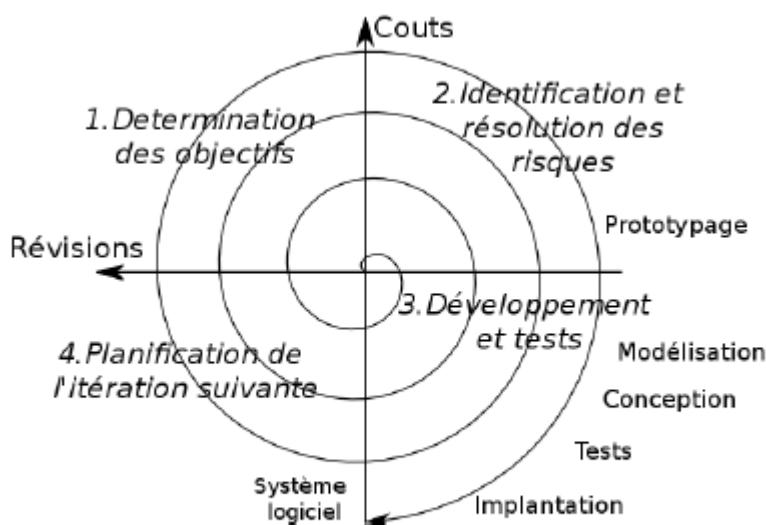


FIG. 2.12 – Modèle du cycle en spirale

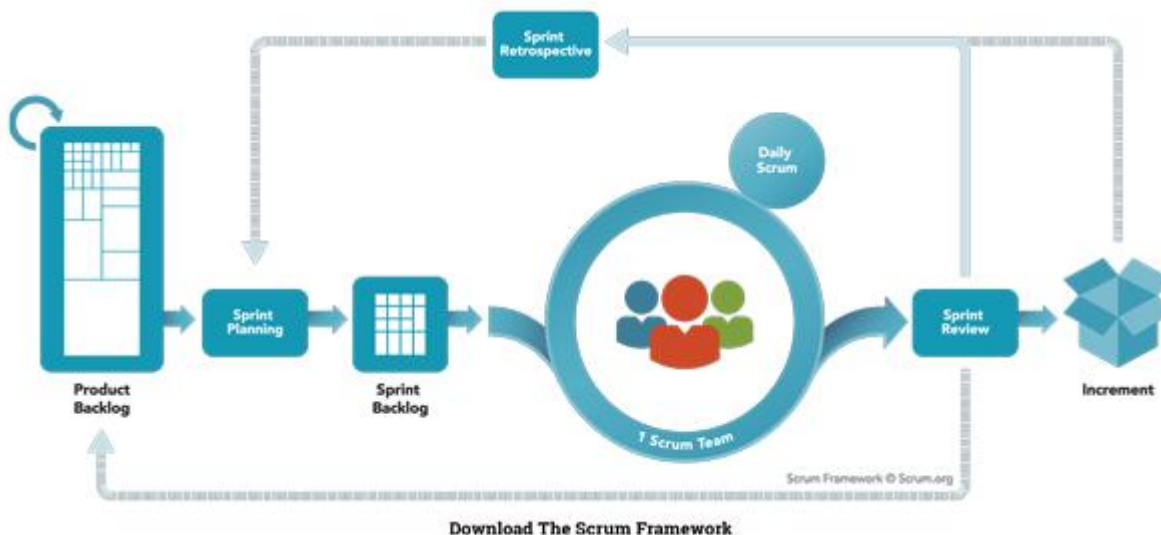
Par contre, à l'inverse du cycle en V, le modèle en spirale met un focus plus important sur l'analyse et la résolution des risques. Ceci est nécessaire car au fur et à mesure des itérations, la réalisation devient de plus en plus conséquente. Il est donc important d'évaluer correctement le risque à chaque itération sachant que toute erreur sera d'autant plus difficile à corriger que le développement sera avancé.

Pre-agilité

Scrum

Développée en 1993, Scrum est une approche agile qui met en avant l'intérêt des petites équipes de développement. Dans ces petites équipes, sont avant tout recherchées les compétences multidisciplinaires et la capacité d'intégration sociale des acteurs. Dans Scrum, les itérations appelées Sprint sont planifiées sur quatre semaines maximum. Ces itérations sont axées sur les besoins définis par le client qui constituent alors le référentiel de travail.

Les besoins sont hiérarchisés par degré d'importance et développés selon les priorités définies par le client. Chaque jour constitue une itération pendant laquelle une réunion est organisée pour établir l'état d'avancement du projet et de vérifier que les fonctionnalités et les délais sont bien respectés.



À la fin de chaque Sprint, une réunion établissant le bilan des réalisations est menée afin d'établir l'efficacité de l'équipe et les éventuelles améliorations qui doivent être apportées. La finalité d'un Sprint est de présenter un démonstrateur au client afin que celui-ci puisse valider les réalisations. Scrum définit un certain nombre d'acteurs qui vont intervenir lors de la mise en œuvre du processus :

1. Le ScrumMaster a pour rôle d'aider les acteurs du développement à communiquer au sein de l'équipe ainsi qu'avec le client. Il doit s'assurer que la philosophie et les pratiques de Scrum sont correctement suivies. Par contre son rôle n'est pas à confondre avec celui du chef de projet dont il est plutôt un conseiller.
2. Le Client ou Product Owner est lui aussi acteur du développement, il a pour rôle de définir les besoins. Il doit également définir les priorités dans les fonctionnalités à réaliser. Ainsi, il participe activement à l'élaboration du produit en suivant les étapes de sa réalisation afin de pouvoir en valider la finalité.
3. Enfin, l'Équipe qui est constituée de l'ensemble des corps de métiers nécessaires à l'élaboration du produit. Ces métiers sont classiquement ceux rencontrés lors des développements logiciels classiques (Développeur, Analyste, Testeur, etc.)

Pour finir, la méthode Scrum s'appuie sur la notion de Visibilité pour qualifier et quantifier les résultats de l'équipe. Des critères de validation doivent exister afin de définir si une fonctionnalité a été complètement réalisée ou non. Des critères d'Inspection doivent être définis afin de déterminer l'existence d'écarts entre la réalisation concrète et l'objectif final. Enfin, la notion d'Adaptation permet, lors d'écarts trop importants détectés pendant des Inspections, de modifier la gestion interne de l'équipe afin d'éviter que ces écarts ne s'amplifient. Ces critères permettront alors de gérer de la dette techniques ou encore d'évaluer la vélocité de l'équipe de développement afin de planifier au mieux les contenus des livraisons futurs

On peut noter que Scrum et XP sont deux approches très proches (on note d'ailleurs une forte proximité des valeurs véhiculées) qui cherchent l'une comme l'autre à remettre la vie du développeur au centre du processus. Cependant Scrum a une volonté plus marquée de fournir un processus formellement établi pour gérer le cycle de vie du logiciel alors que XP s'intéresse plus sur les bonnes pratiques. Généralement les équipes de développement pratiquant l'une ou l'autre pratiquent la seconde.

Conclusion

Les processus de développement sont nombreux et si historiquement l'approche en V a été adoptée majoritairement, les nouvelles approches comme les approches agiles tendent à s'imposer comme étant des solutions plus dynamiques et adaptées aux changements actuels. Je reviendrais plus dans le détail des approches telles que Scrum et Lean qui méritent au vu de leur présence actuelle d'être approfondies. De la même manière, ces dernières années, au-delà des processus de développement, on a pu voir émerger le DevOps dont le but est de regarder au-delà des équipes de développement en prenant en considération l'ensemble des besoins des acteurs agissant de près ou de loin dans la livraison du produit au client. Il me semble que même si cela déborde du cadre des processus de développement de l'équipe logicielle, celle-ci doit prendre en compte ces nouvelles préoccupations et cela a forcément un impact sur leur activité.