

Cycle de développement de logiciel

Cours BDR - SI3

Nadia Abchiche-Mimouni

nadia.abchiche@univ-cotedazur.fr

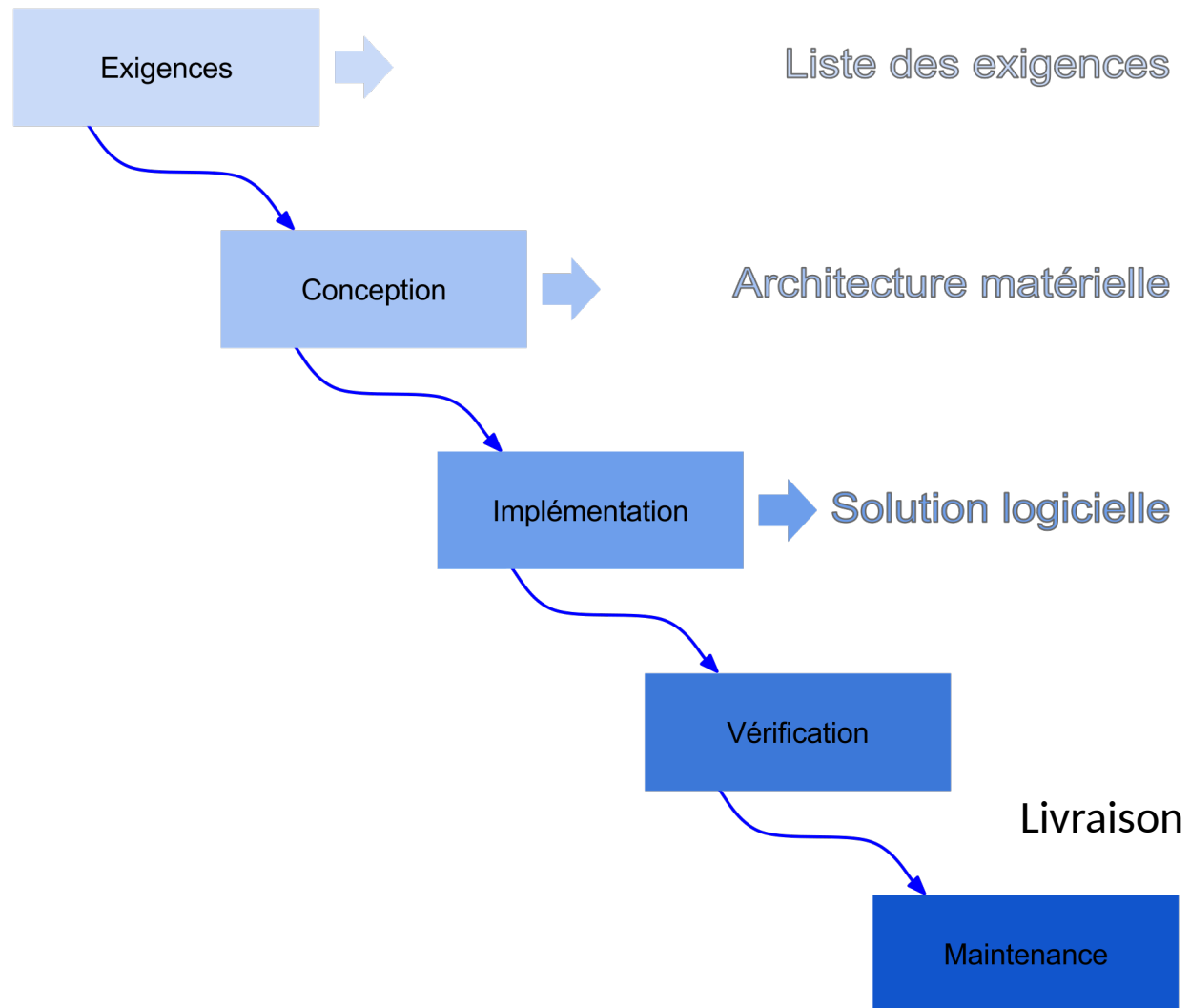
Objectif

Acquérir les notions de base, notamment, les phases de développement de logiciel

Méthode de développement de projets informatiques : pourquoi ?

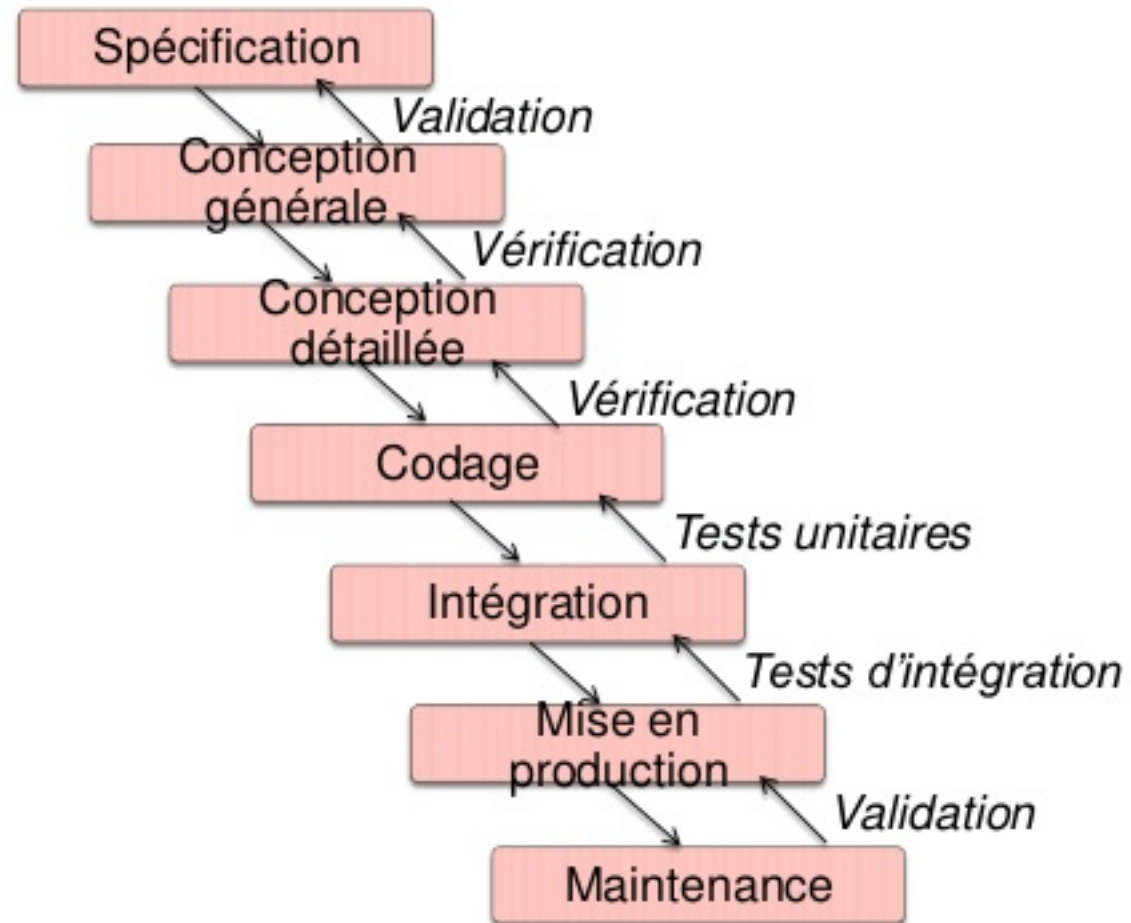
- Le projet est une aventure
- Besoin d'un cadre de travail commun
- Besoin de garanties :
 - Répondre à l'objectif
 - Garantir la qualité du résultat
 - Garantir la tenue des délais
 - Maîtrise des coûts

Modèle en cascade

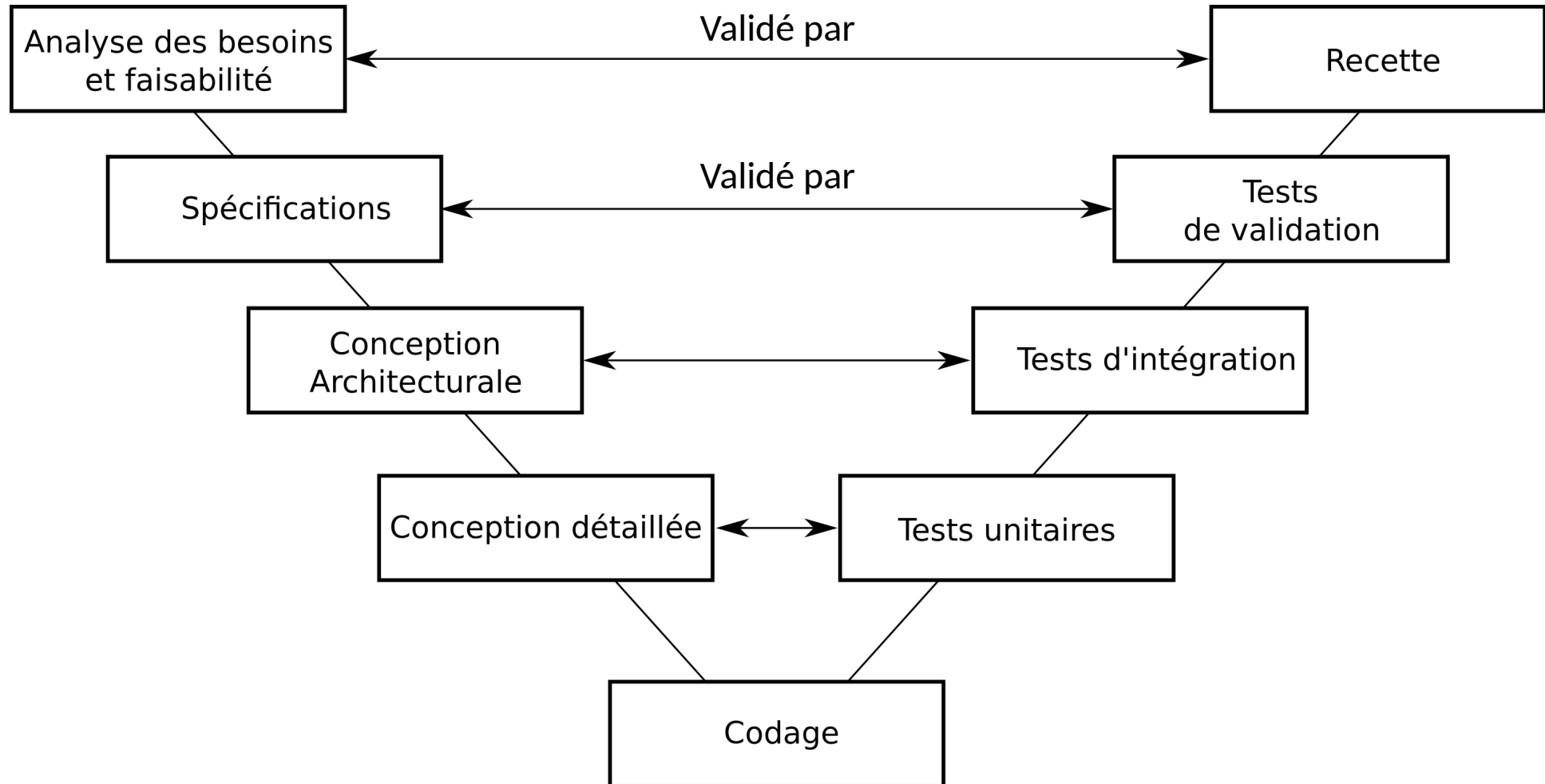


Modèle en cascade

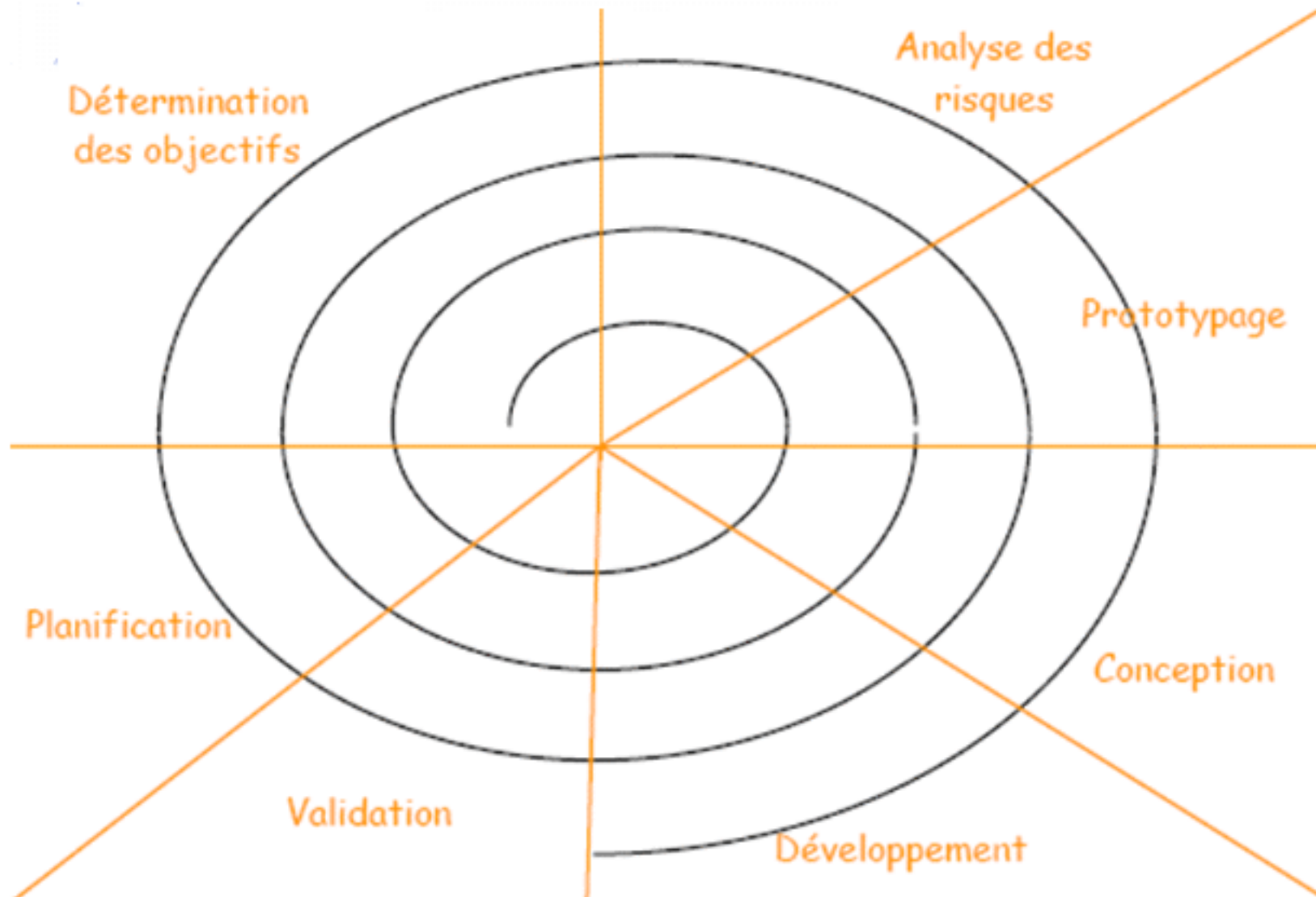
Cycles de vie classiques Modèle en Cascade



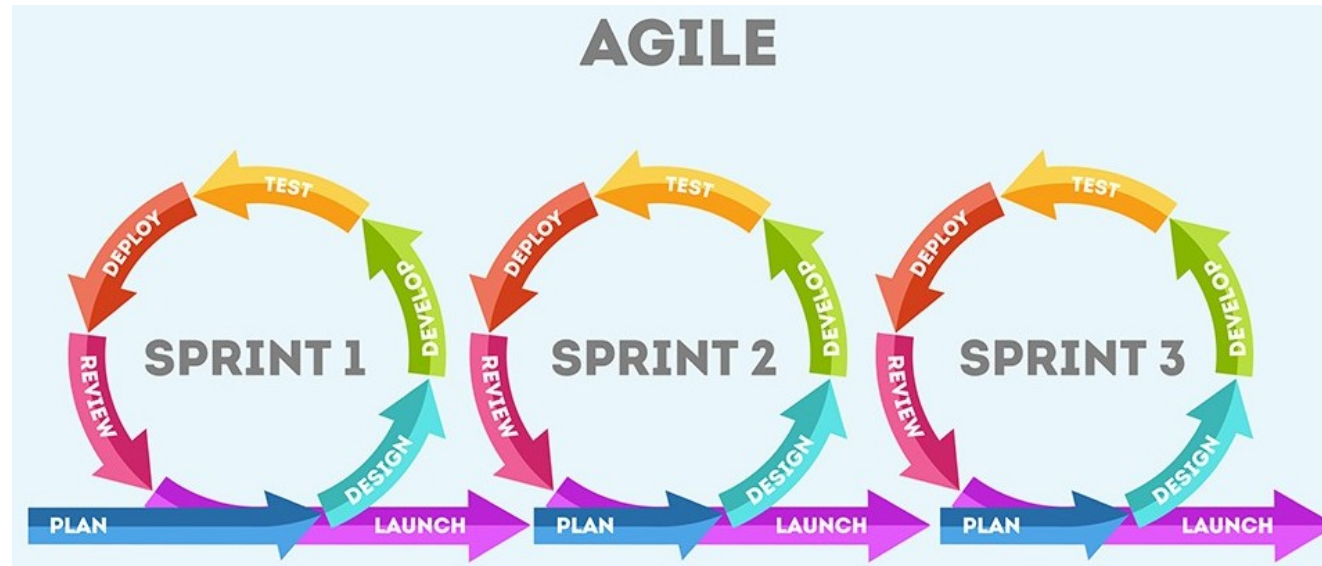
Cycle en V



Modèle en spirale (itératif)



La Méthodologie Agile



<https://fiches-pratiques.chefdentreprise.com/Thematique/gestion-1050/FichePratique/Les-raisons-utiliser-methodes-Agile-entreprise-354321.htm>

Quatre valeurs fondamentales

- Individus et interactions plutôt que processus et outils
- Avancement du projet plutôt que documentation exhaustive
- Collaboration et entraide plutôt que rigidité contractuelle

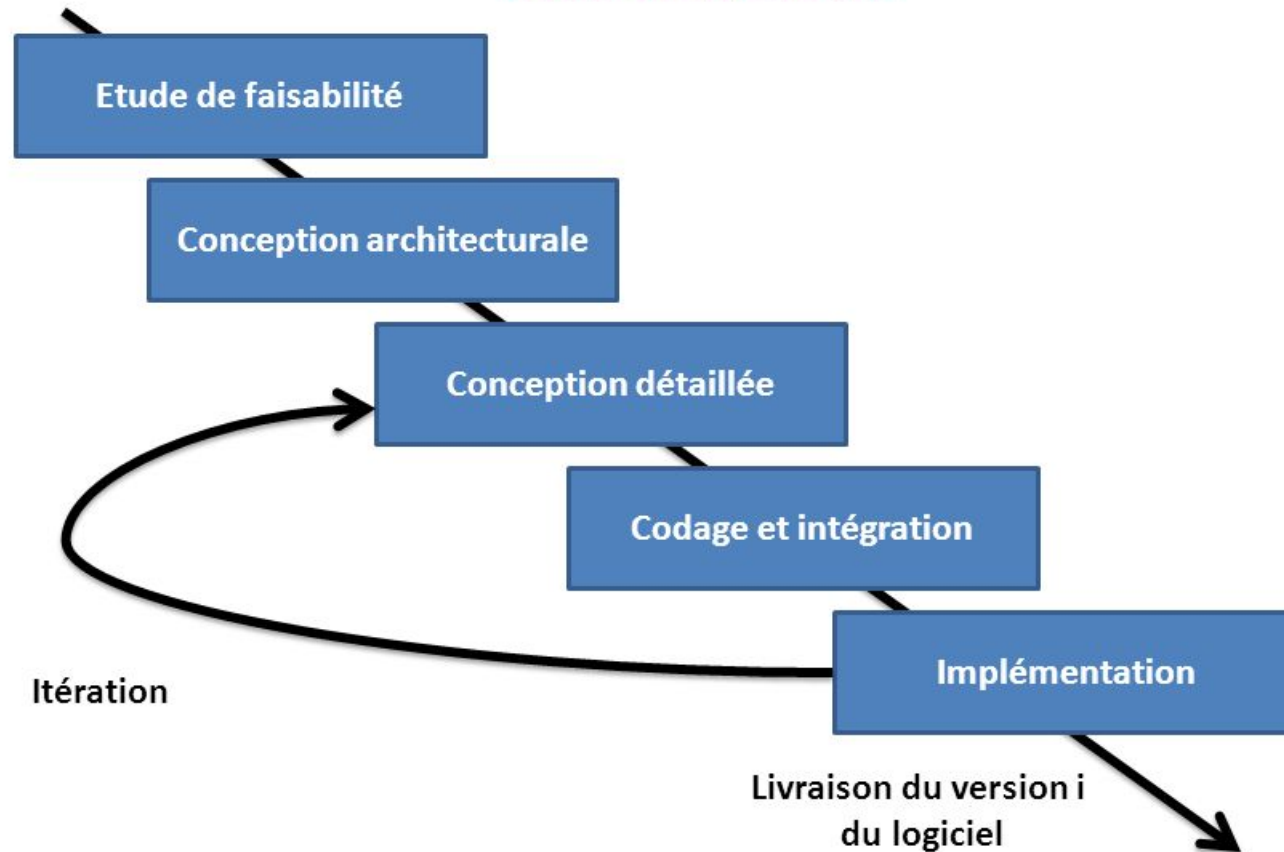
La Méthodologie Agile

Valeur ajoutée

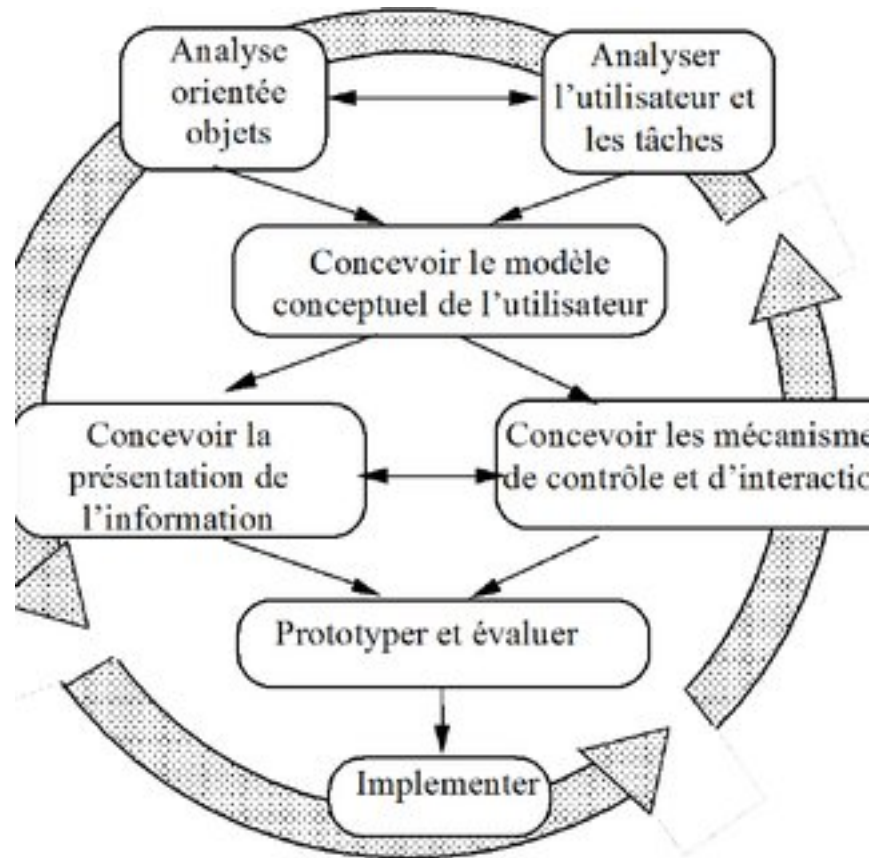
- Démarrer un projet sans avoir tous les éléments
- Livrer régulièrement sur des périodes courtes
- Capitaliser les erreurs et les succès
- Introduire constamment des améliorations
- Maîtrise des coûts, de la qualité et des délais

La Méthodologie Agile

Modèle de cycle de vie : Le modèle incrémental



Cycles enrichis sous l'angle des interactions homme-machine (travaux de recherche)



La méthode Agile SCRUM

La Méthode Agile Scrum

- Scrum = mêlée

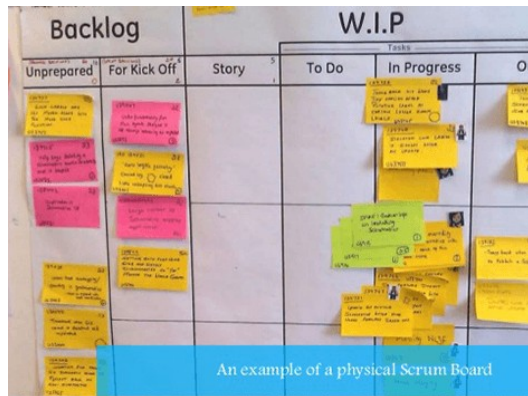


Définition : « Scrum est un cadre agile pour réaliser des projets complexes. A l'origine formalisé pour des projets de développement logiciel, Scrum fonctionne aussi dans tout environnement de travail complexe et innovant. Les possibilités sont infinies. La méthode Scrum est faussement simple »

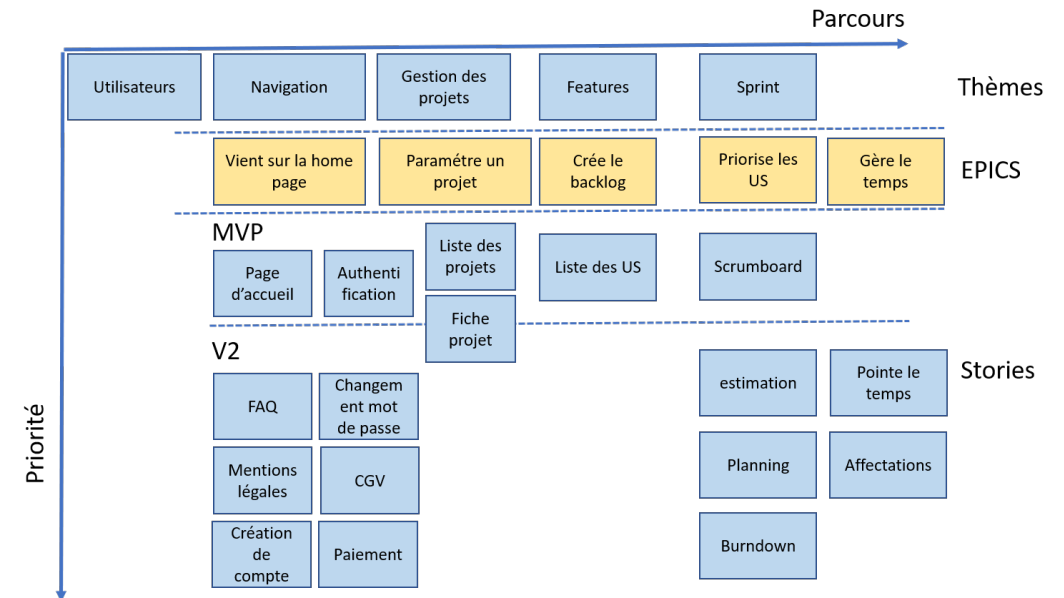
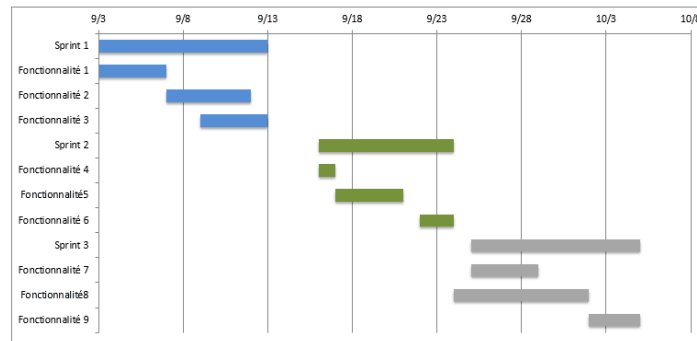
La Méthode Agile Scrum

Le fonctionnement 1/6

Le backlog : les **Actions** des Missions du projet sont collectées dans un **backlog du Projet**, tenu par le coordinateur du projet



Nom de la tâche	Responsable	Début	Fin	Jours	Statut
Sprint 1	Alex B.	9/3	9/13	10	Achevée
Fonctionnalité 1	Frank C.	9/3	9/7	4	Achevée
Fonctionnalité 2	Jacob S.	9/7	9/12	5	Achevée
Fonctionnalité 3	Jacob S.	9/9	9/13	4	En retard
Sprint 2	Jacob S.	9/16	9/24	8	En cours
Fonctionnalité 4	Alex B.	9/16	9/17	1	En cours
Fonctionnalité 5	Frank C.	9/17	9/21	4	Nom démarrée
Fonctionnalité 6	Shari W.	9/22	9/24	2	Nom démarrée
Sprint 3	Shari W.	9/25	10/5	10	Nom démarrée
Fonctionnalité 7	Alex B.	9/25	9/29	4	Nom démarrée
Fonctionnalité 8	Kennedy K.	9/24	10/2	8	Nom démarrée
Fonctionnalité 9	Jacob S.	10/2	10/5	3	Nom démarrée



La Méthode Agile Scrum

Le fonctionnement 2/6

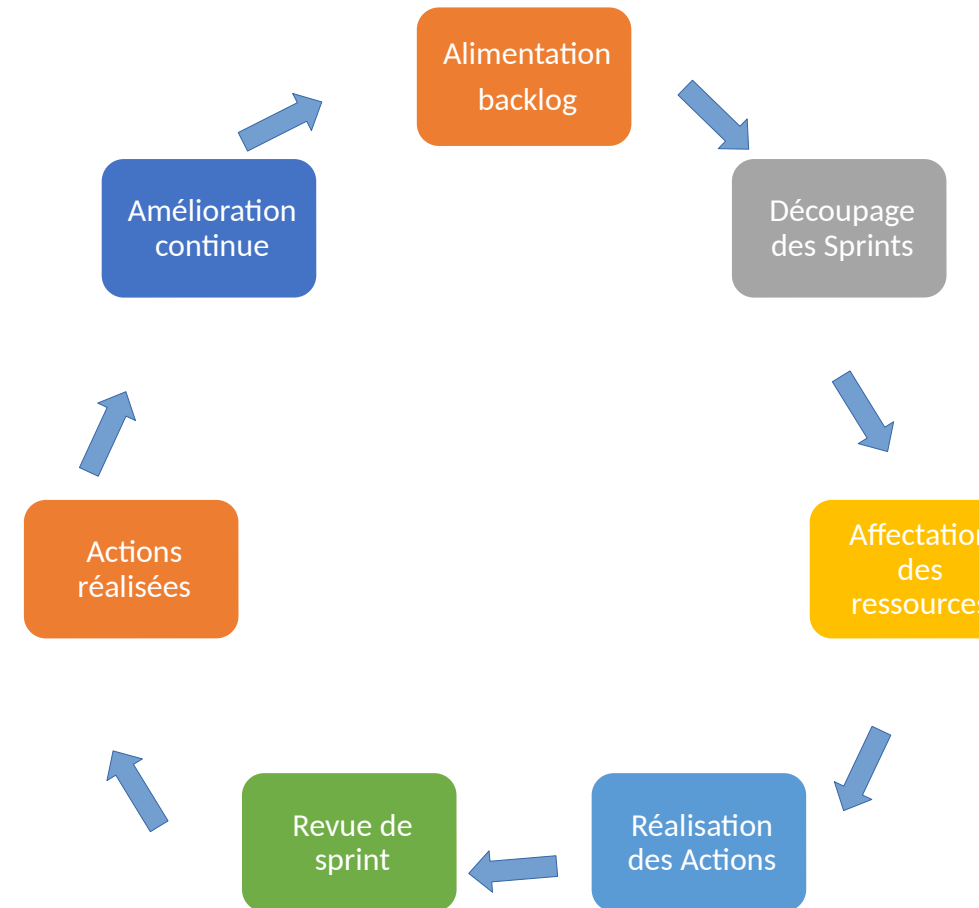
Les Releases

- Le projet est livré en plusieurs Versions (***releases***)
- Chaque ***release*** est le produit d'une série de ***sprints***

Le fonctionnement 3/6

Les Sprints

- Un **sprint** est une itération de 1 semaine à 1 mois pendant laquelle une série d'**Actions** sont intégralement **réalisées**.
- Ces Actions sont déterminées par l'**équipe** au début de chaque sprint (**blacklog**) lors d'une réunion de **planification des sprints**



Le fonctionnement 4/6

L'inspection

- ***Pendant le sprint*** l'avancement est contrôlé quotidiennement lors de la ***mêlée quotidienne (daily scrum)***, animée par le ***scrum master***, et d'éventuels ajustements sont décidés pour assurer le succès du sprint.

Le fonctionnement 5/6

La Revue du Sprint

- ***A la fin de chaque sprint***, l'équipe obtient un avancement partiel du projet, et des Missions potentiellement livrables. Une réunion de ***revue de Sprint*** apporte un ***feedback*** qui permet d'ajuster l'organisation du projet et le backlog pour les sprints suivants. C'est une démarche d'amélioration continue.

Le fonctionnement 5/6

La Revue de Sprint

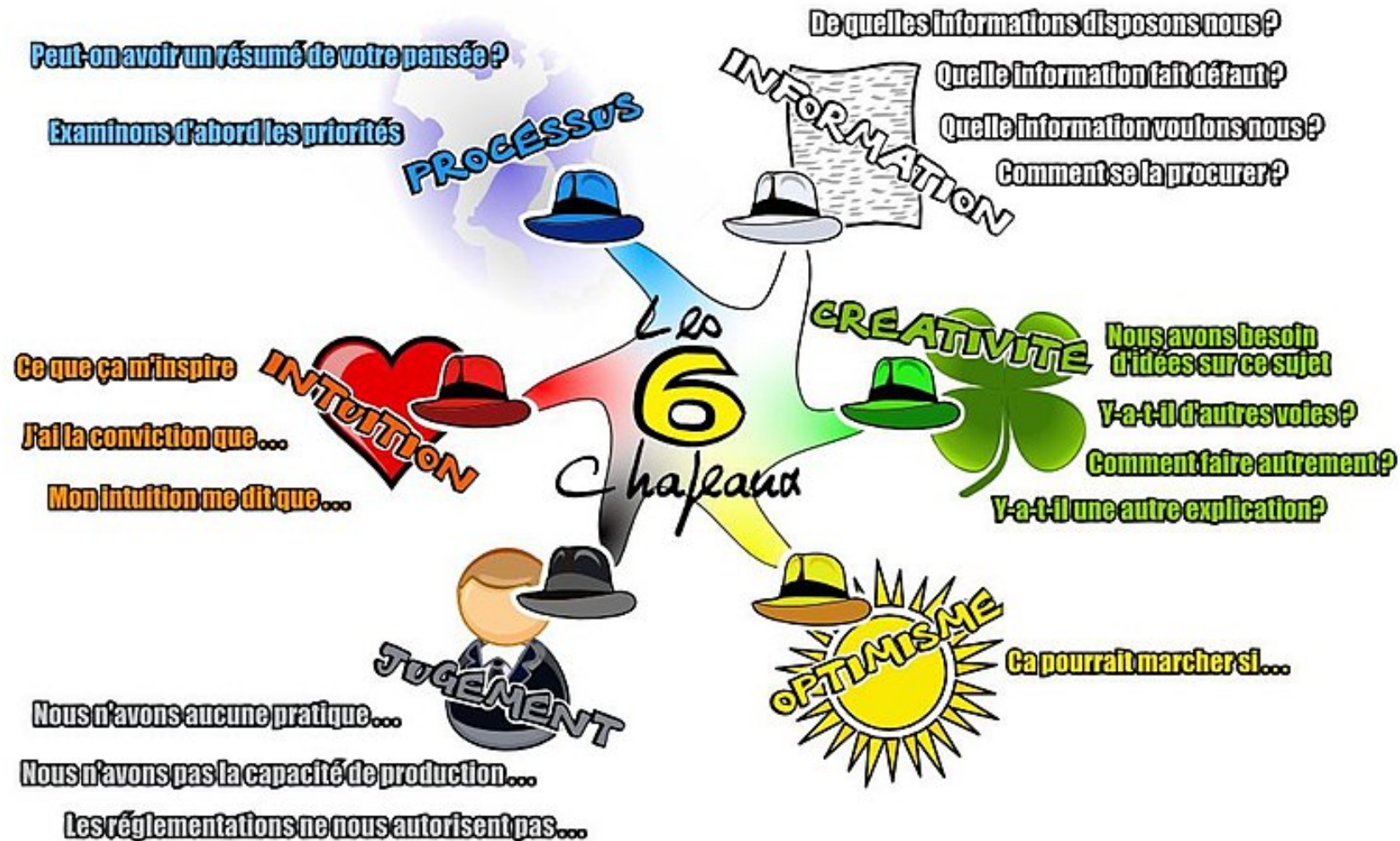
- Etapes
 - Collecte des informations, **en continu**
 - Identification des idées d'amélioration, **en continu**
 - Définition d'une amélioration prioritaire, **pendant la Revue de Sprint**
 - Adaptation de la méthode pour le sprint suivant, **pendant la Revue de Sprint**
- **Réunion auto-organisée**
 - Trois points :
 - Ce qui a été fait
 - Ce qui va être fait
 - Identifier les obstacles et les améliorations à apporter
- >> Mise à jour du plan de sprint

Le fonctionnement 6/6

La Rétrospective

- ***A la fin du projet***, il s'agit de procéder au Bilan général du projet et à une **revue du projet**, à l'image de la **revue de Sprint**.

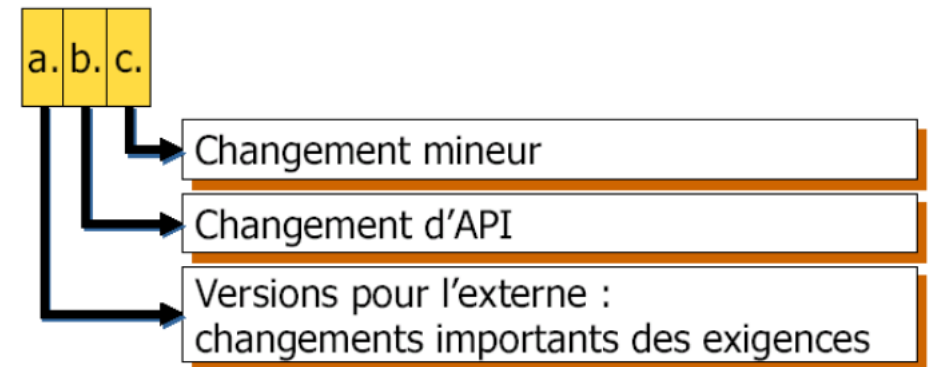
Courte mise en pratique : les six chapeaux de Bono



Gestion des versions et des évolutions

Numérotation à trois chiffres :

- a. 1^{er} chiffre : Numéro de versions majeures du produit, dont la sortie s'accompagne de progrès importants au niveau des fonctionnalités, et/ou changement notable d'environnement d'utilisation ou de portabilité
- b. 2^{ème} chiffre : numéro des version mineures. L'incrément est réalisé à chaque fois que l'équipe de développement libère une version du produit qui corrige des bugs attendus par les clients (mais non bloquants), et apporte des modifications légères
- c. 3^{ème} chiffre : numéro des corrections, versions résultant de la maintenance



- ❖ Version Alpha : version terminée en cours de test et de revue de qualité
- ❖ Version Béta : version alpha validée en test auprès d'un panel de clients privilégiés

Gestion des versions et partage de code

Avantages des outils de versionning

- Améliorer le travail collaboratif (travailler en simultané sur les mêmes fichiers)
- Si des modifications ont lieu en parallèle, la fonction merge (fusion) fera le travail.
- Aider à la communication générale et au partage des informations.
- Permettre le développement sur plusieurs versions en même temps grâce au système de branche.
- L'archivage améliore la sécurité (pertes de données) et la maintenance (rollback en cas de problèmes).
- L'historique permet la statistique : qui a fait quoi, quand, comment et pourquoi.
- Ils prônent la centralisation de l'information et l'utilisation d'un socle commun.
- Ils obligent la validation des changements par leurs visibilités au reste de l'équipe.

Inconvénients

- En centralisant l'information, ils améliorent la sécurité interne mais pas externe (vigilance !)
- Au premier abord, la mise en place et la configuration d'une telle solution peut en rebuter plus d'un.
- Il existe pléthore de fonctionnalités et leurs prises en main peut nécessiter du temps.
- Ils peuvent rapidement prendre de la place en fonction des projets et du nombre de collaborateurs.
- Ils nécessitent un logiciel client pour s'y connecter et valider son travail. Il s'agit de trouver le bon !

Les moyens les plus célèbres : Git et SVN

<https://www.youtube.com/watch?v=hPfgekYUKgk>