

«Management de projet: le Cycle en V »

Vincent Le Cam

Ingénieur de Recherche

**Responsable de la cellule Électronique et
Informatique Industrielle**

**Département MACS
(Mesure Auscultation Calcul Scientifique)**

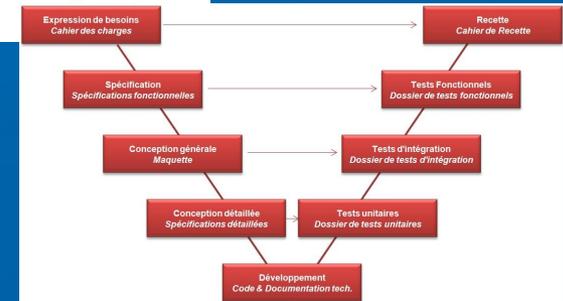
IFSTTAR

**Institut Français des Sciences et
Technologies du Transport de
l'Aménagement et des Réseaux**

**Version V1.0 - Déc. 2012
Tous droits réservés**

Contact

vincent.le-cam@ifsttar.fr



IFSTTAR

Sommaire



1) Qu'est ce qu' Ifsttar ?

2) Pourquoi une méthode ?

3) Le cycle en V

1) Approche générale

→ Positionnement dans **un projet**

2) Approché détaillée

→ analyse et illustration de chaque étape du **cycle en V**

4) Autour du « Cycle en V »

5) Illustrations sommaires par quelques exemples concrets

6) Bilan - conclusion



Qu'est ce qu'IFSTTAR ?



IFSTTAR

Cf. la présentation ci-dessous (6 diapos) :

/home/lecam/_Synchro/PRO/ENSEIGNEMENT/Enseignement_dispense/CoursFormationsVLC/For



2 - Management : notion de méthode

Pourquoi une méthode ?



- La vie de l'entreprise est faite de **projets**, d'affaires, de programmes, de livrables...
- Notion qu'à un moment donné un **besoin de l'entreprise doit être satisfait**,
- Il y a projet dès lors qu'une commande ou un besoin est formalisé entre 2 équipes d'une même entreprise ou de 2 entreprises distinctes
- On identifie alors **pour chaque projet** :
 - **La MOA : Maitrise d'Ouvrage ou le Client**
 - Entité porteur du projet et du besoin – détient le budget
 - Est ou représente l'utilisateur final
 - **La MOE : Maitrise d'œuvre le Fournisseur**
 - Entité responsable de la conduite du projet
 - Interface avec le client (voire avec sous-traitants)
- A priori vous débuterez plutôt coté MOE : SSII, prestation, BE...



2 - Management : notion de méthode

Pourquoi une méthode ?

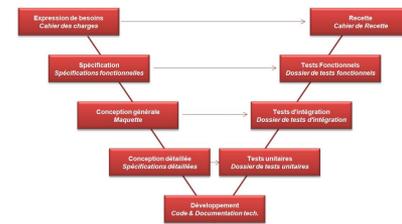


Pourquoi une méthode pour les projets ?

- A l'inverse, comment imaginez-vous un grand projet sans méthode ?
- Pensez-vous que Ariane puisse exister sans méthode ?
- Adaptons le dicton « Si vous trouvez que la culture coûte cher, essayez l'ignorance »...
- La méthode du « cycle en V » comme toute méthode a vocation à :
 - Apporter de la **visibilité**
 - Savoir où on va, permet d'**anticiper** les biais
 - Force à **formaliser** les idées, les équipes, les rôles
 - Permet de **maîtriser des délais, des coûts...**
- Une **méthode** peut être imposée par le client lors de la commande...
Ou partiellement - exemple : règle de codage, tests à effectuer...



3 – Le Cycle en V



Approche Générale :

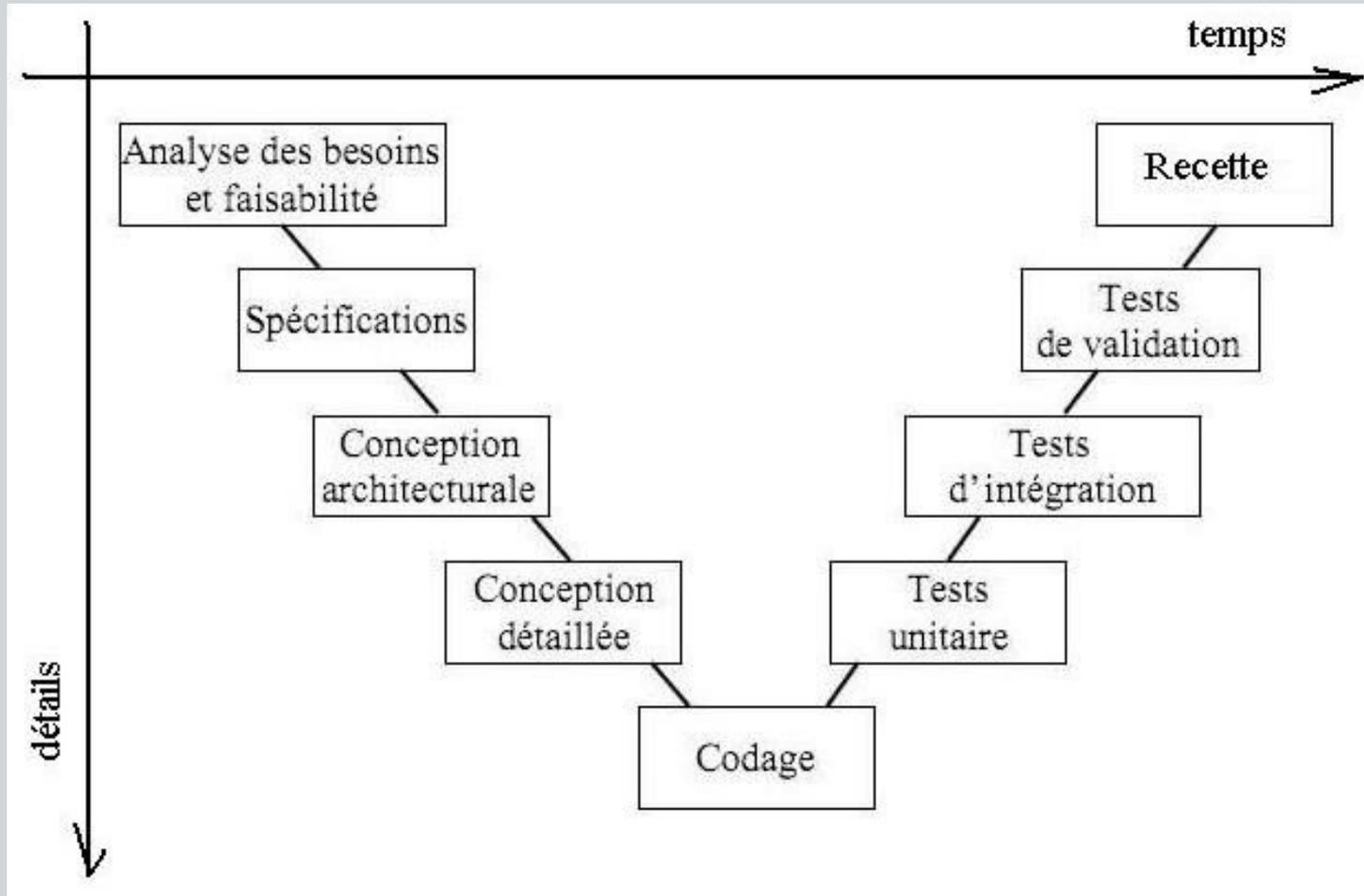
- Plus un **concept général**, du bon-sens formalisé qu'une recette de cuisine
Il est l'un des outils du chef de projet
- **Ensemble de phases** descendant du général (le besoin) vers le particulier (réalisation) et remontant vers le produit final (livrable) => d'où le V
- Contexte:
 - Né dans les **années 80 (a fait ses preuves)**... mais probablement avant
 - Issu essentiellement de l'industrie logicielle mais transposé et transposable à de nombreux autres domaines : électronique, mécanique...à la **conception de système en général**
 - **Objectif** : permettre la découpe d'un projet en un ensemble d'étapes claires permettant des A/R de l'une à l'autre, des itérations pour corriger les biais, être réactifs et garder la visibilité sur l'avancement globale



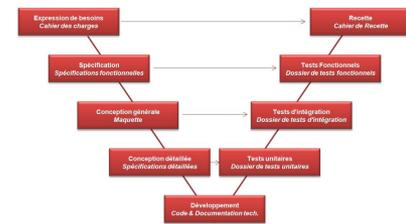
3 – Le Cycle en V



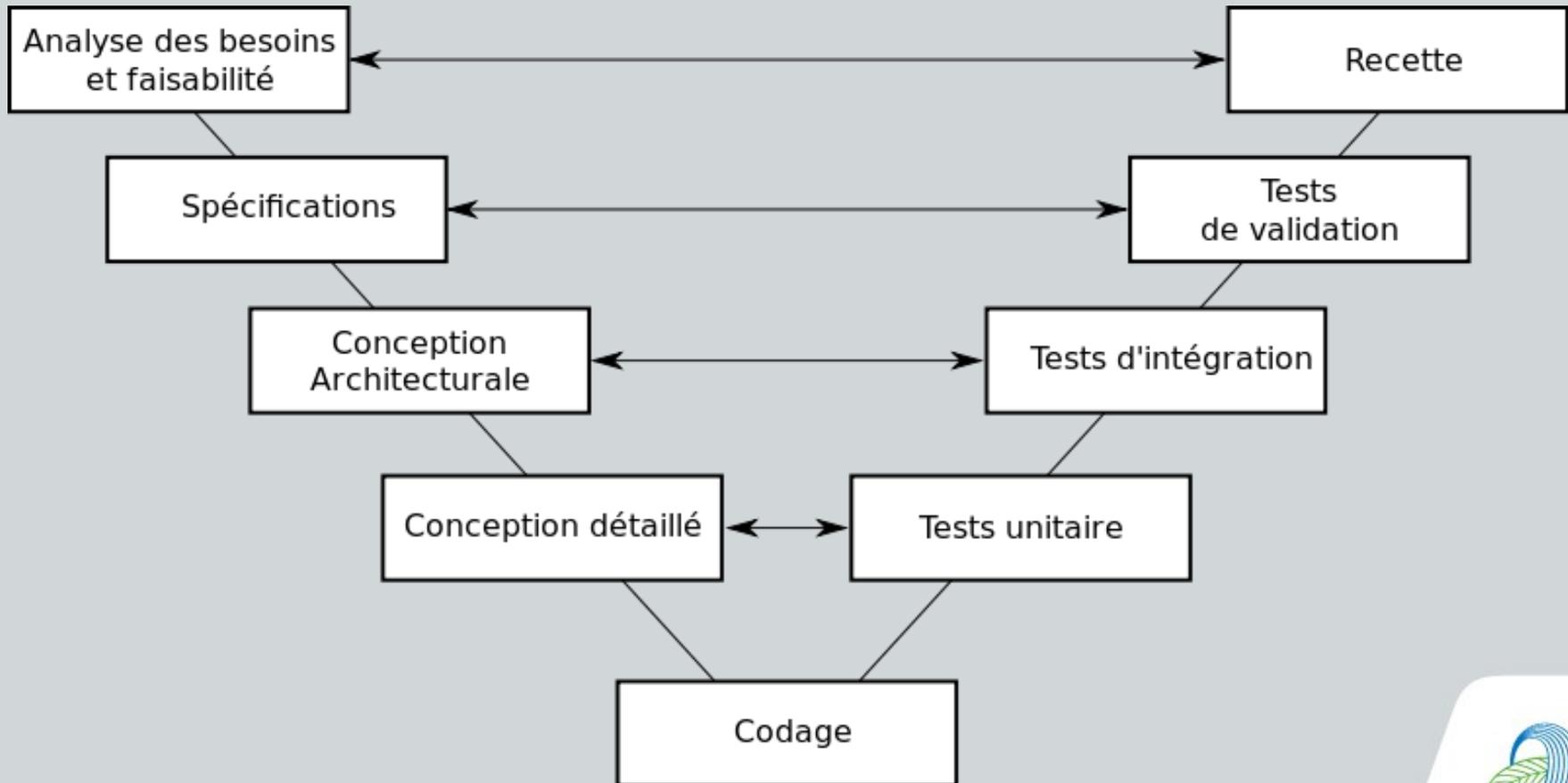
Approche et commentaires généraux : vue « temps / détails »



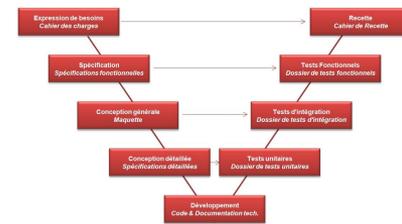
3 – Le Cycle en V



Approche et commentaires généraux : vue « correspondance des phases »



3 – Le Cycle en V

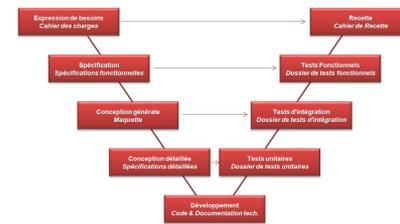


Parcourons chaque phase du Cycle en V :

- Chaque phase est (selon taille du projet) un **sous projet** à part entière
- A chaque phase, interviennent plus ou moins certains **acteurs** côté Client (MOA) ou Prestataire (MOE) – avec des rôles précis.
- A chaque phase, correspondent des entrées et (surtout) **des livrables de sortie** versionnés
- Notion **d'itération** : on itère dans une phase jusque aboutie quitte à remonter à phase précédent
- Le Cycle en V peut être facilement relié à la démarche Qualité de l'entreprise - ISO9001 = satisfaction client
→ appliquer ce type de méthode est une façon de démontrer que l'on se soucie de la satisfaction client



3 – Le Cycle en V



Phase 1 : l'expression du besoin

Description :

- Phase la plus importante
- Le client décrit le produit qu'il veut, selon sa vision
- Souvent négligée, l'expression du besoin doit être la plus détaillée possible.
Exemple ce que le produit doit faire ou pas (ex : imprimer)
- Détailler le contexte d'usage, les utilisateurs (surtout si différents du donneur d'ordre), les versions précédentes...
- Le développeur pose le max. de questions ! Prendre le temps, plusieurs rencontres...
- Phase qui souvent permet d'affiner les chiffrages : délais/étape, coût final

Qui :

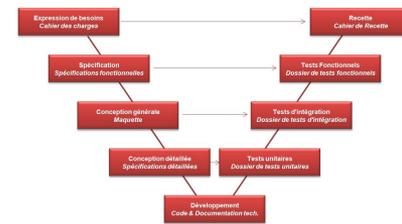
- Client ou délégation = assistance à MOA (*consulting*, SSII...)
- Fournisseur : écoute / questionnement. Éviter *a priori*

Livrable :

- le document « **Cahier des Charges** »
- Doit être **validé par les 2 partenaires !**



3 – Le Cycle en V



Phase 2 : Spécifications fonctionnelles

Description :

- Traduire le Cahier des Charges en « **Quoi** » (et pas en « Comment »)
- But : identifier **les fonctions du système**
- **Fonction** = service unitaire rendu à l'utilisateur ; entrée/traitement/sortie;...
- Phase négligée par les *jeunes développeurs* !
- Par diverses **méthodes**, on modélise le système à développer comme un ensemble de fonctions coopérantes (entre elles, avec l'utilisateur, des BDD,...)
En suivant des template de doc, en répondant à des questions...
- Voir illustration... (On peut aussi spécifier ce que le système ne fait pas)

Qui :

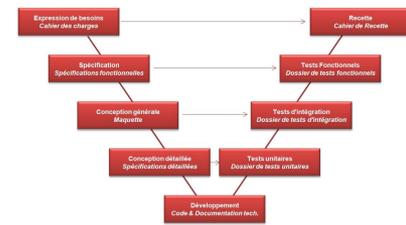
- **Fournisseur** : travail d'ingénieur ; rédaction; questionnement du client
- **Client** : répond aux questions, affine son besoin d'un pt de vue fonctionnel

Livrable :

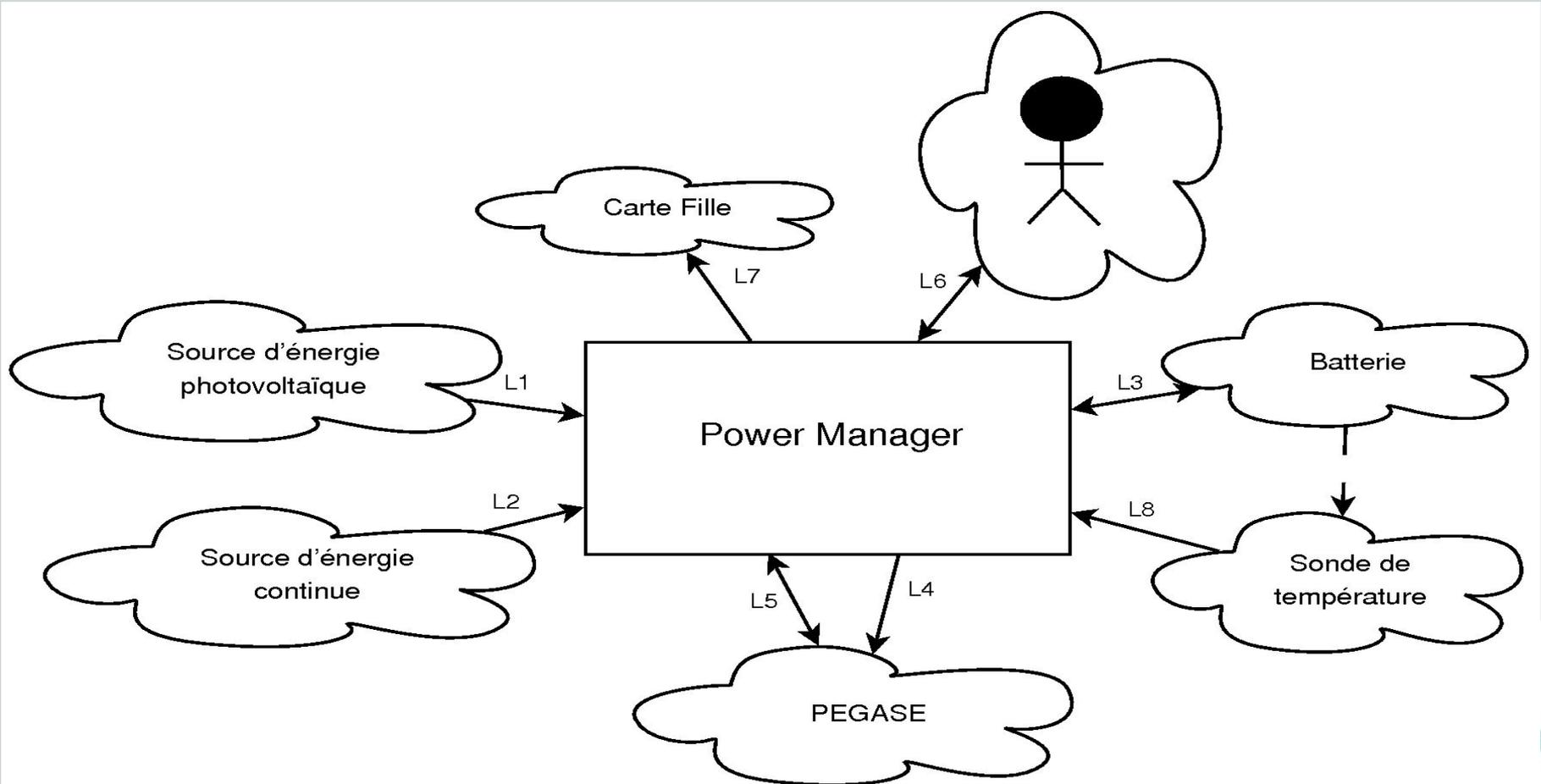
- le dossier des «Spécifications fonctionnelles »
- Doit être **validé par les 2 partenaires** !



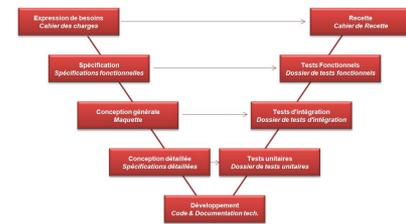
3 – Le Cycle en V



Phase 2 : Spécifications fonctionnelles – illustration sommaire
→ on *isole* au maximum le système des entités coopérantes

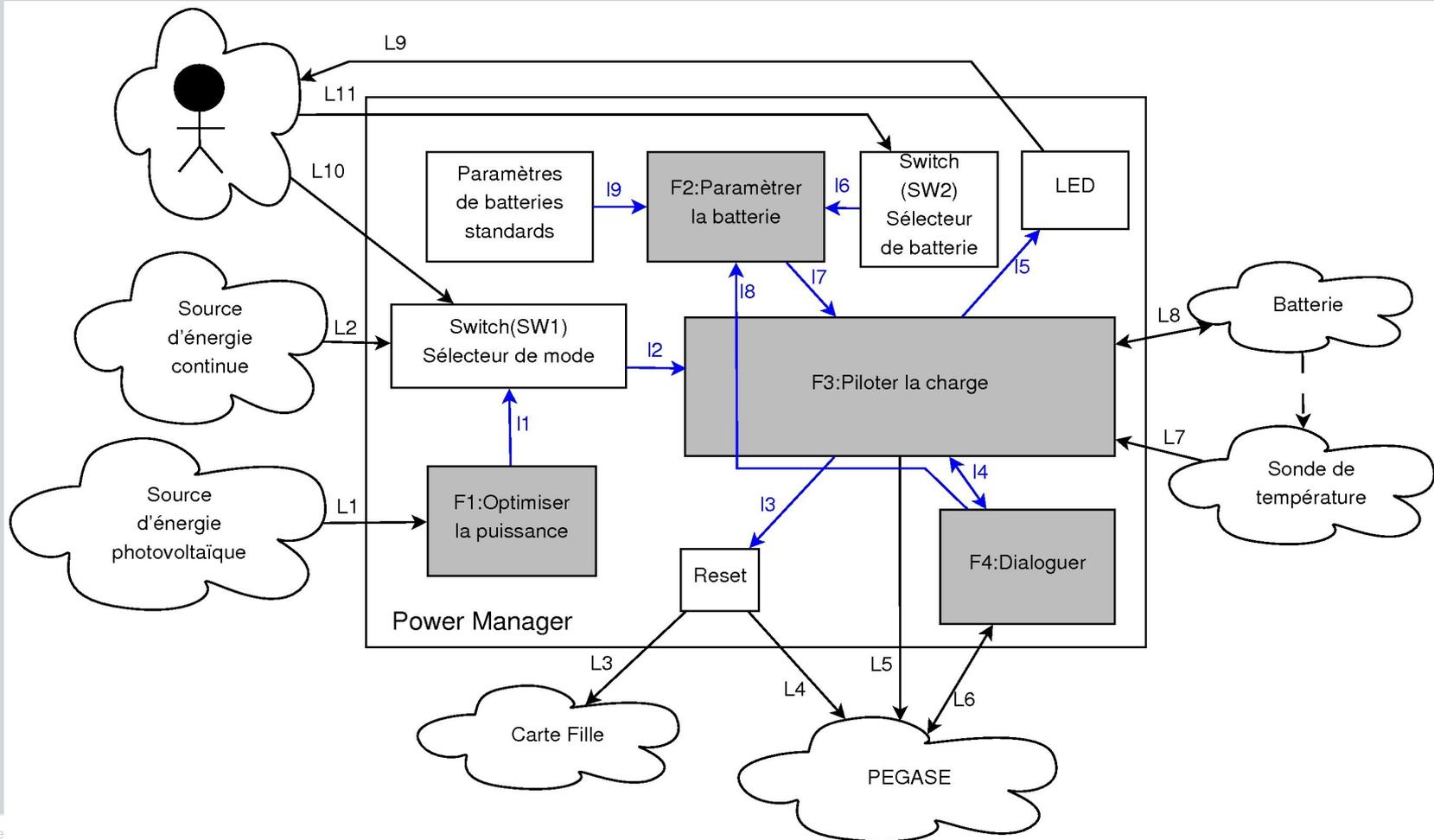


3 – Le Cycle en V

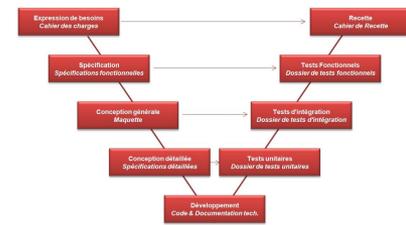


Phase 2 : Spécifications fonctionnelles – illustration sommaire

→ on identifie les fonctions et spécifie les données échangées, leurs liens,...

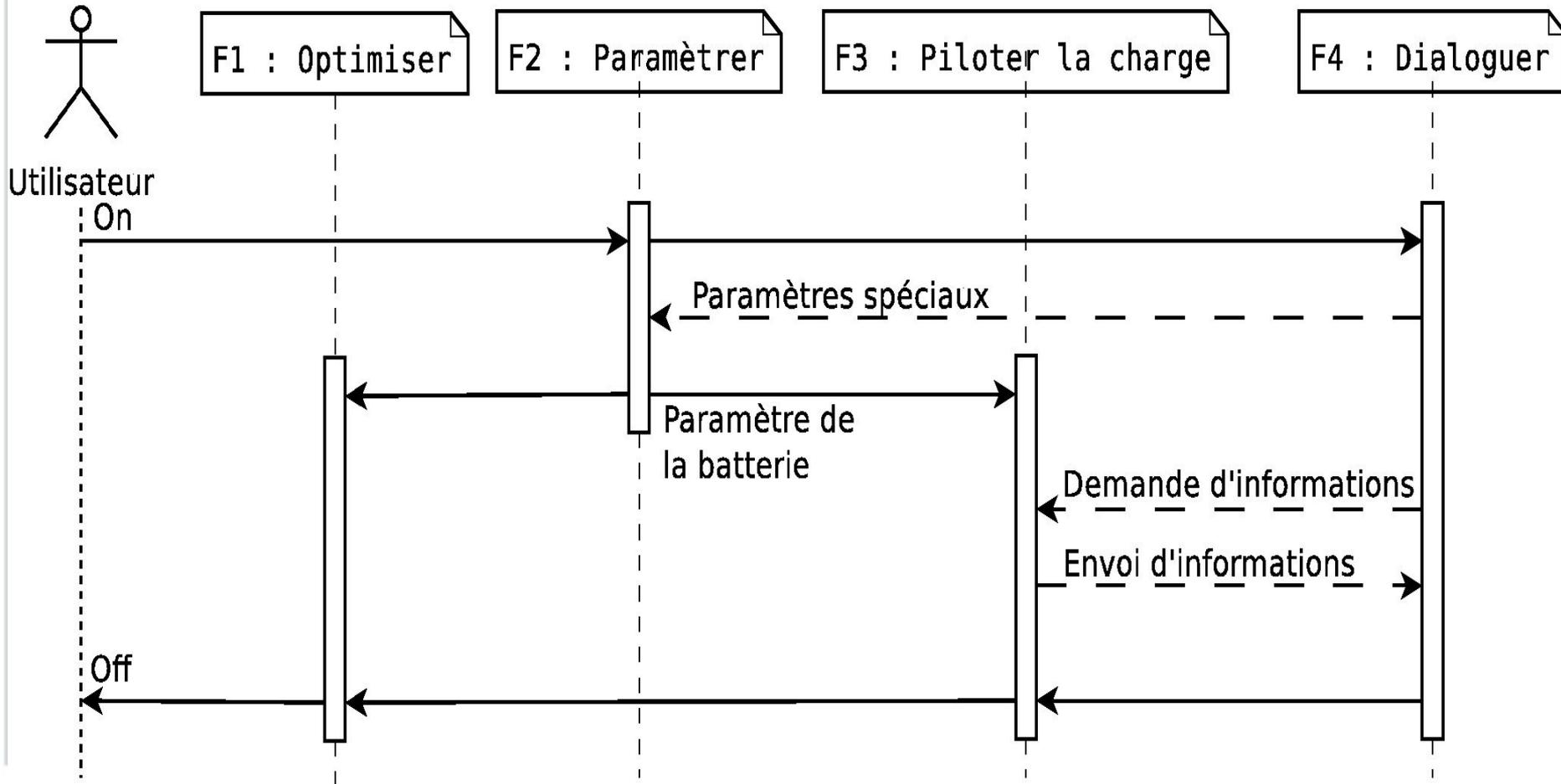


3 – Le Cycle en V

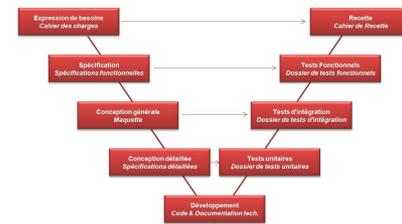


Phase 2 : Spécifications fonctionnelles – illustration sommaire

→ on spécifie les cas d'utilisation (use-case) des fonctions...



3 – Le Cycle en V



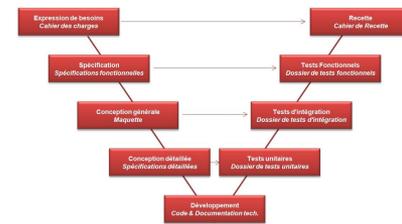
Phase 3 : Conception générale et détaillée

Description :

- Traduire les Spécifications Fonctionnelles (et le CDC) en « **Comment** »
- But : spécifier **comment les fonctions du système seront implémentées**
- **Des choix forts sont posés** : IDE, environnement du système (win/linux), générateur de code ou non, choix des composants du systèmes (microprocesseur, batteries, OS embarqué ...), dimension, conso, format des fichiers / des BDD,...
- Selon la nature du projet et des fonctions **plusieurs méthodes** peuvent être choisies :
 - Analytique pour si fonction soft plutôt orientée calcul
 - Automate à état fini / surtout pour des fonctions événementielles (IT)
 - Diagramme de classes sous outils UML : Rational Rose, Umbrello...
 - CAO pour fonctions électroniques : schéma de principe, plans,...
- Travail **réductionnel** et aussi plus **technique**



3 – Le Cycle en V



Phase 3 : Conception générale et détaillée

Description

- Garder le dialogue avec le client ! Surtout si équipes techniques
- Ne pas chercher à reproduire des solutions existantes ; éviter les *a priori* techniques...
- Travail d'autant plus complexe que l'intégration est forte : sur OS embarqué, cohabiter avec d'autres applis fortes...

Qui :

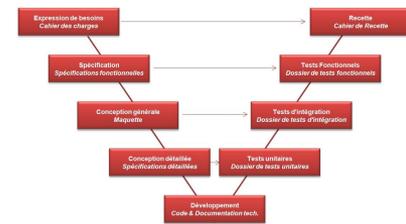
- **Fournisseur** : travail d'ingénieur ; rédaction
- **Client** : rien ! Sauf cas particulier: même métier, avis sur la façon de faire...mais devrait être précisée plus en amont ds le CDC

Livrable :

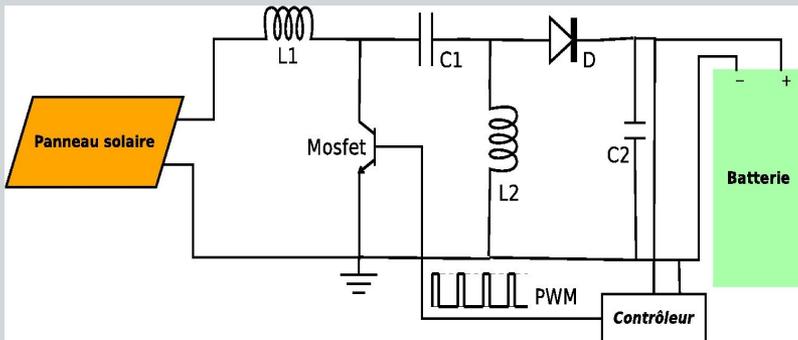
- le **dossier des «Spécifications de Conception »**
 - **Selon le projet** : un ou plusieurs docs
- Doit être **validé en interne par le fournisseur (chef de projet)**



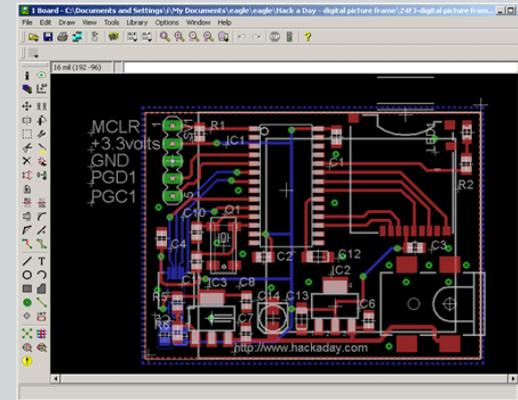
3 – Le Cycle en V



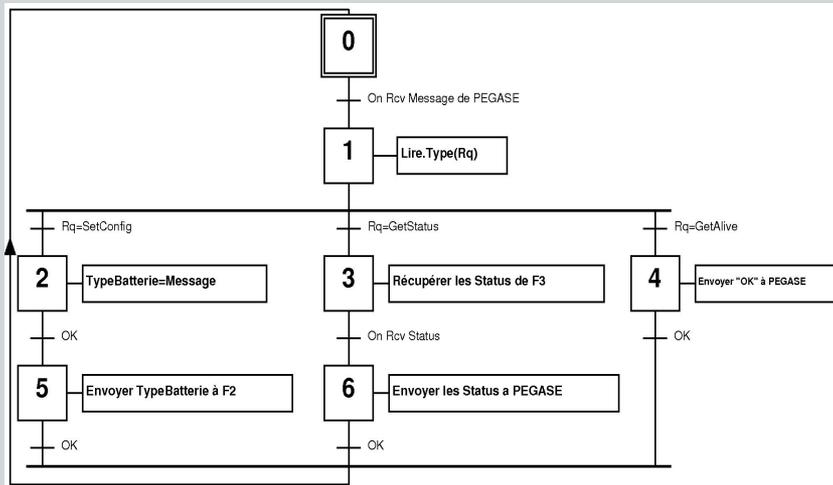
Phase 3 : Conception générale et détaillée : illustration sommaire



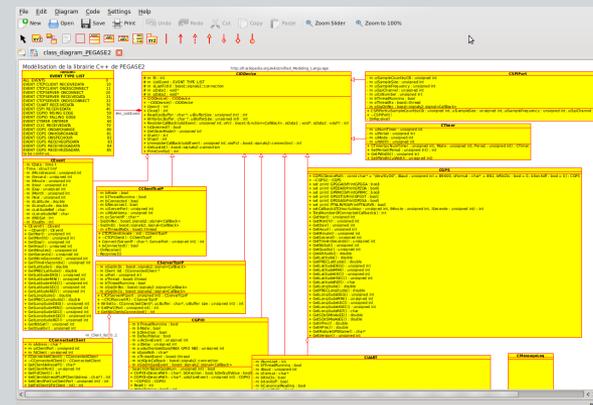
Schémas de principe en électronique



Choix de composants et CAO



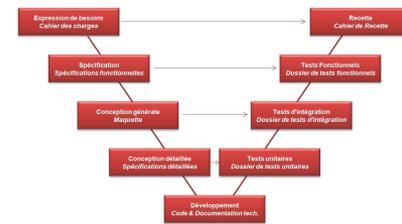
Automates à états finis



Modélisation UML



3 – Le Cycle en V



Phase 4 : Codage, réalisation

Description

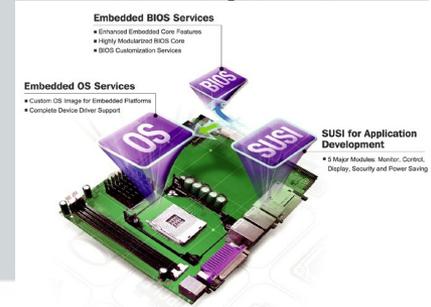
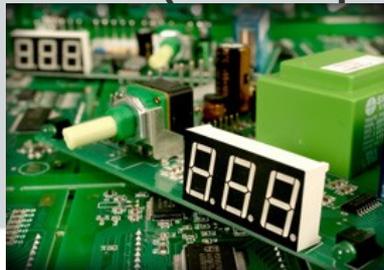
- Travail **technique** pur – **on réalise le quoi et le comment**
- On utilise des outils : IDE, compilateurs, machine à tirage des cartes,...
- On respecte des **méthodes** : règle de codage
- « Quand on code on ne réfléchit plus »...en théorie
- On donne vie à chaque fonction et sous-fonction unitairement

Qui :

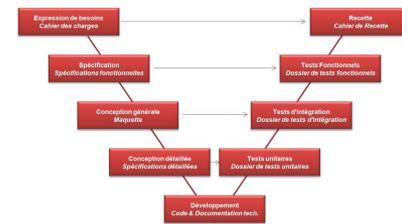
- **Fournisseur** : travail d'ingénieur, de techniciens, d'opérateurs...
- **Client** : rien ! Sauf cas particulier → **Exemple** : règles de codage imposées...

Livrable :

- L'ensemble (final) et chaque fonction intermédiaire est **validé en interne par le fournisseur (chef de projet)**



3 – Le Cycle en V



Phase 5 : Tests unitaires

Description :

- Objectif : **tester unitairement le caractère fonctionnel et opératoire d'une fonction ou sous-fonction**
- Parfois associés à la phase précédente... selon pratique de l'ets ou taille du projet
- Postulat : la (sous) fonction est réputée terminée
- Pas des tests poussés (a priori) : tests aux limites, modes dégradés...
- Notion d'itération unitaire : si bug trouvé sur ma fonction, alors je corrige ma fonction et re-test... pas de notion d'ensemble !



Qui :

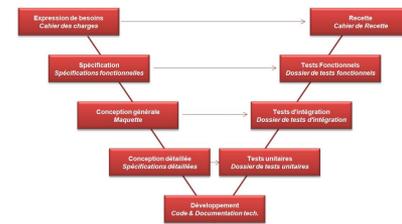
- **Fournisseur** : travail d'ingénieur, de techniciens, d'opérateurs...
C'est souvent le développeur qui se teste...et *qui est juge et partie !*
- **Client** : rien !

Livrable :

- En théorie : un formalisme libre, type check-list des tests à passer/passés, nb d'itérations nécessaires...
- En pratique : souvent rien, hélas !



3 – Le Cycle en V



Phase 6 : Tests d'intégration (TI)

Description :

- Objectif : **tester le caractère fonctionnel et opératoire d'un système dans son ensemble... toutes briques intégrées**
- Parfois associée (à tort) à la phase suivante : Tests de Validation
- Postulat : chaque développeur a fourni ses briques ; le système est réputé terminé / intégré. Idéalement : il y a 1 (des) intégrateur(s)
- Les TI n'ont pas vocation à vérifier les fonctions au sens des Spécifications Fonctionnelles mais au sens du Dossier de Conception (correspondance)
- Ce qui importe c'est le **bon fonctionnement de l'ensemble** dans une version et sur une cible égale ou proche de la livraison...au sens de la cohérence technique. Ex. : génère les fichiers, accède à la BDD, IHM cohérentes...



Qui :

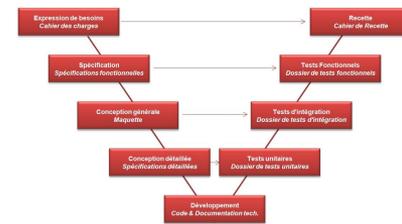
- **Fournisseur** : travail d'ingénieur, de techniciens, d'opérateurs...

Livrable :

- Un plan d'intégration spécifiant la version testée, son contenu, **scénarios de tests**
- Un bilan des TI : nb de passes, itérations,...



3 – Le Cycle en V



Phase 7 : Tests de validation (TV)

Description :

- Objectif : **tester le caractère fonctionnel et opératoire d'un système dans son ensemble... toutes briques intégrées**
- **Étape ca-pi-ta-le** synonyme de la qualité du produit livré
- Correspondance : les Spécifications Fonctionnelles
- Se référer à la présentation spécifique de V. Le Cam sur PTV :

La Méthode des Tests de Validation

/home/lecam/_Synchro/PRO/ENSEIGNEMENT/Enseignement_dispense/Cours

Qui :

- **Fournisseur** : travail de personnel technique...ou non ; voire autre équipe dédiée
- **Client** : fournit (evtlmt) des scénarios, valide le Plan de Tests, reçoit et valide le bilan des TV...

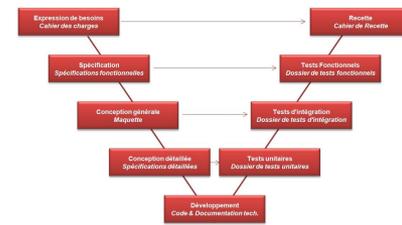
Livrable :

- Le Plan des TV dont : Matrice de Couverture et Cas de Tests
- Le bilan des TV

→ Cf . Illustration sur PEGASE



3 – Le Cycle en V



Phase 8 : La Recette

Description :

- Objectif : **les 2 acteurs valident ensemble la conformité du système aux attentes (CDC)**
- Correspondance : le Cahier des Charges
- On formalise des tests de recette qui généralement reprennent des TV ou des scénarios plus métier ; tests plus orientés « métier du clients »
- Tests réalisés dans le contexte technique du client ! « Chez lui », dans son SI
- Livraison des sources, plans, schémas, docs...conformément au contrat
- Phase finale : point global sur la complétude des livrables



Qui :

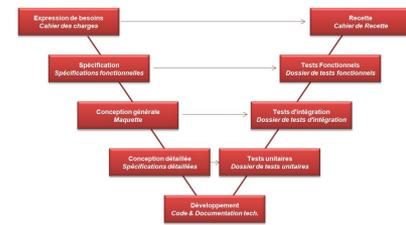
- **Fournisseur**
- **Client**
 - mènent (ou font mener par end-user) des tests de recette

Livrable :

- Procès Verbal de Recette (y compris restriction) => paiement
- Engagements du fournisseurs : vices cachés, garantie sur x ans...



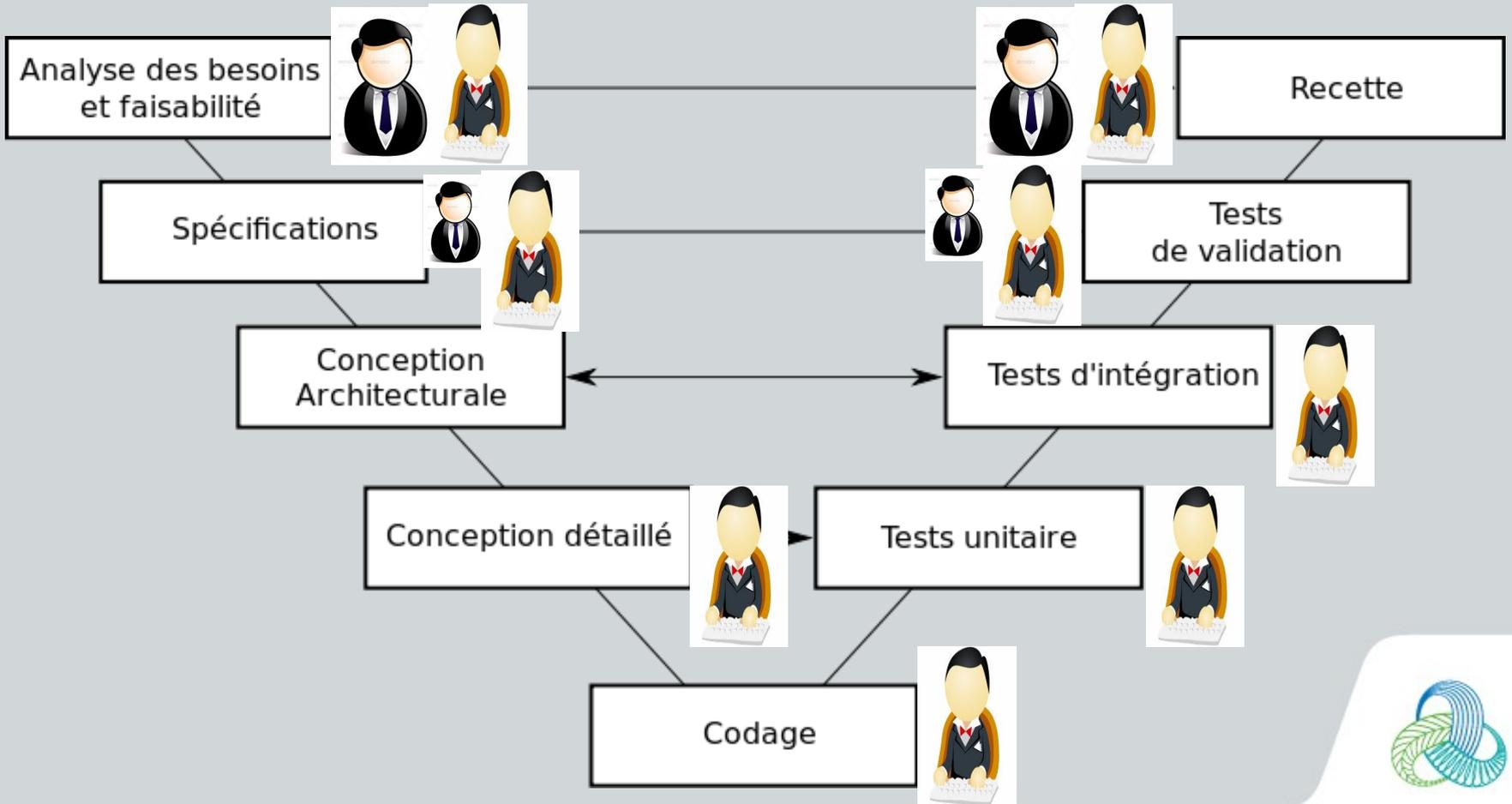
3 – Le Cycle en V



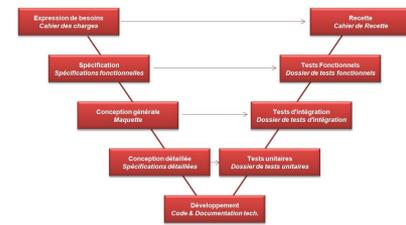
Synthèse interventions du client:



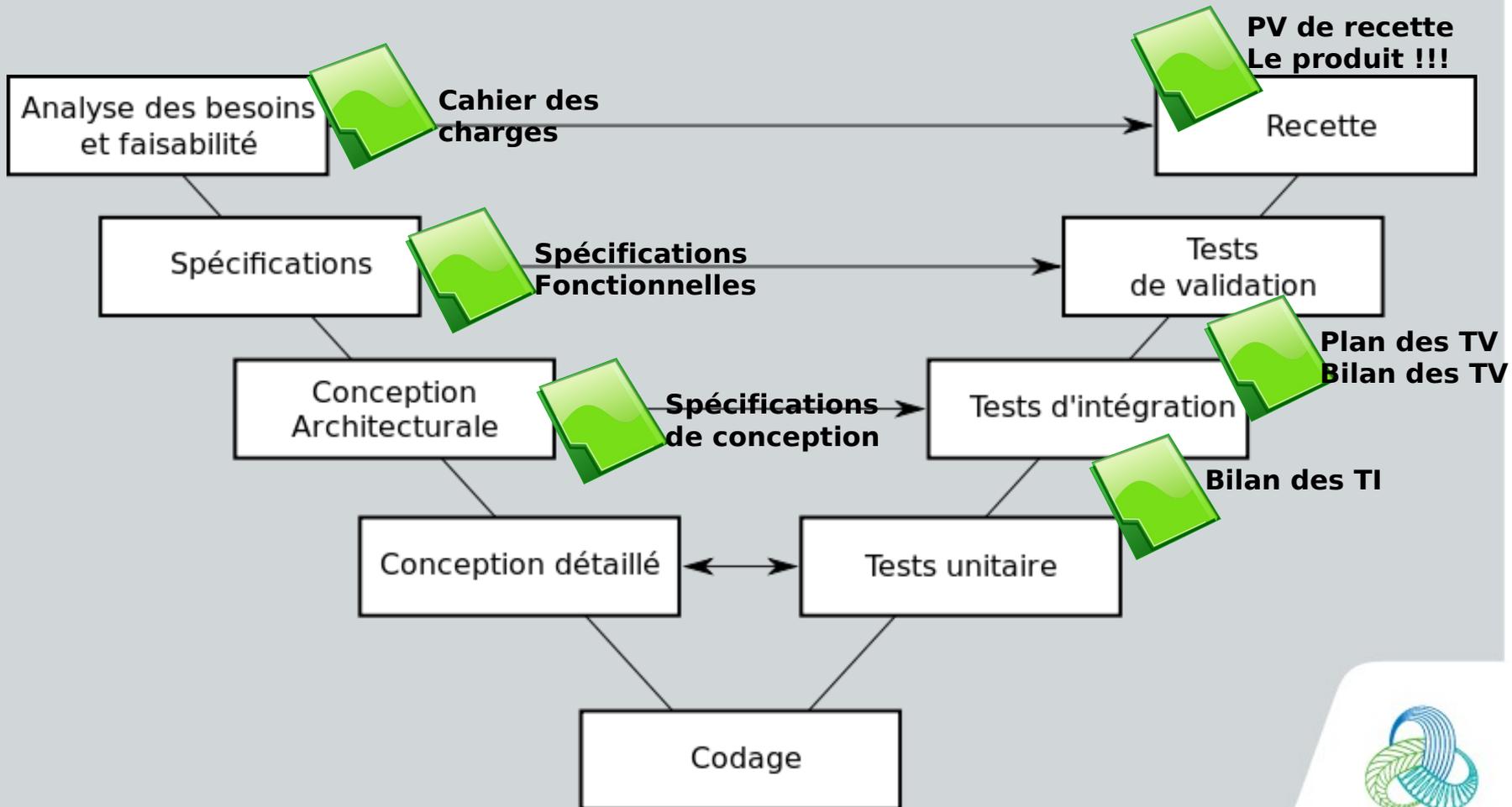
et du fournisseur:



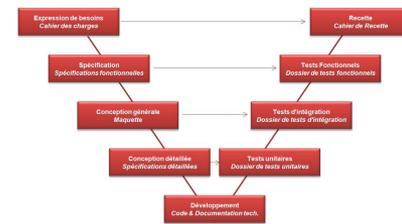
3 – Le Cycle en V



Synthèse des livrables:



4 – Autour du Cycle en V



Penser à la suite !

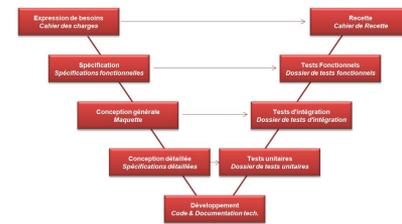
Le Cycle en V terminé et le projet... n'est pas fini:

- En cas d'erreur (bug, incompréhension...), **le client rappelle !**
- En cas de grand succès, **le client rappelle...**
 - Pour une évolution fonctionnelle du système
 - Pour une autre version ...

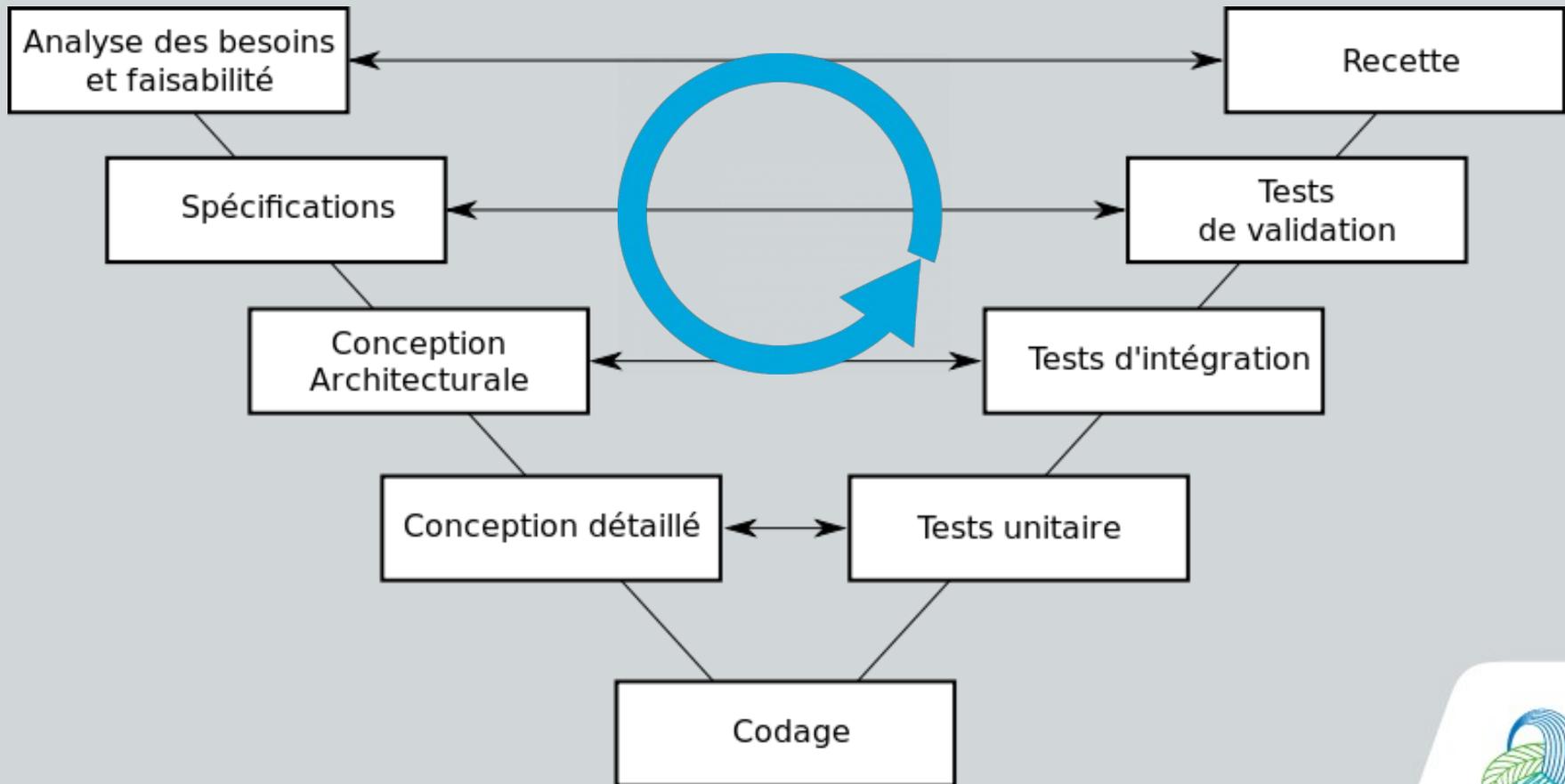
« Le client content le dit à 3 personnes, le déçu le dit à 10 personnes »
- D'autres phases démarrent :
 - Nouvelle version : nouvelle itération globale du Cycle en V
 - Importance de bien formaliser le projet ; du suivi de version.



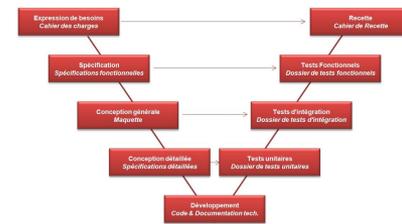
4 – Autour du Cycle en V



Notion d'itération, point fort du Cycle en V :



4 – Autour du Cycle en V

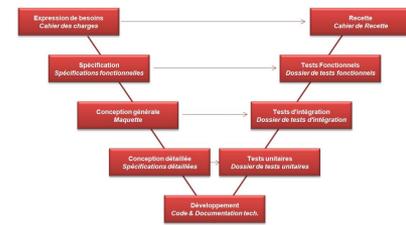


Et voici le cycle en W :

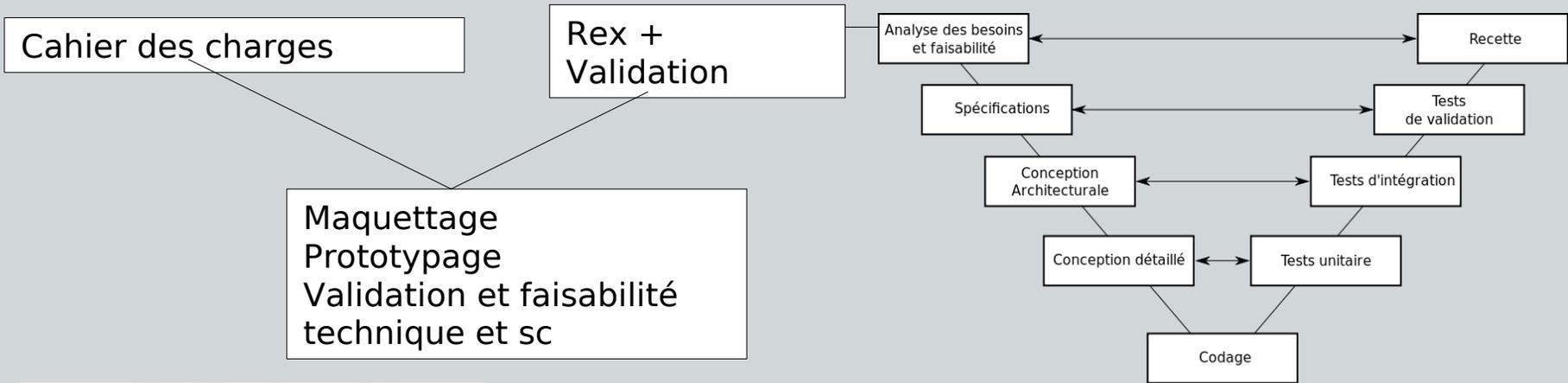
- **Idee** : palier l'un des défauts du Cycle en V classique où un temps important existe entre le CDC et les premières réalisations techniques (maquettage, prototype) et levées de verrou techniques / scientifique
- Après le CDC (pas forcément finalisé), introduction de deux phases :
 - Une phase de conception / maquettage pour validation incluant :
 - De la conception
 - Du maquettage
 - Du prototypage
 - De la faisabilité techno
 - Une phase (client/fournisseur) de Retour d'Expérience pour validation
- ...puis démarrage d'un Cycle en V classique
- **Nota bene** :
 - la maquette/proto ressort souvent de base
 - cohérent avec le monde open-source, re-utilisation, intégration,...



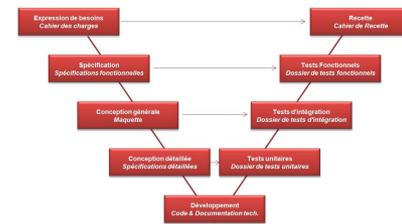
4 – Autour du Cycle en V



Et voici le cycle en W :



4 – Autour du Cycle en V

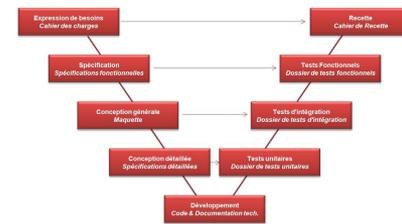


Les outils du Chef de Projet et de l'équipe :

- Le « Cycle en V » est un outil conceptuel pour le CdP et son équipe
- Concrètement l'ets. et/ou le CdP met en place des outils +ou- intégrés pour :
 - Le suivi global du projet :
 - Suivi des livrables
 - Tenue des délais (Gantt)
 - La rédaction partagée des docs : CDC, specs,...
 - La modélisation UML (soft) les schémas (hard)
 - Le co-développement à plusieurs (!= sites) développeurs
 - Le suivi des versions : sécurité, log, merge, tag,...
 - Piloter la phase des Tests de Validation → système de Tickets
 - ...
- Passage en revue de qqs outils connus et répandus



4 – Autour du Cycle en V



Les outils du Chef de Projet et de l'équipe : les plateformes de pilotage intégrée (comprenant de nombreuses fonctions)

guitone Mtn Source Tree -

http://127.0.0.1/~tommyd/indefero/www/index

Welcome, **Thomas Keller**. [Sign Out](#) | [Project List](#) | [Forge Management](#) | [Help](#)

[Project Home](#) [Downloads](#) [Documentation](#) [Issues](#) [Source](#) [Code Review](#) [Project Management](#) **guitone**

[Source Tree](#) | [Change Log](#) | [How To Get The Code](#)

guitone Mtn Source Tree

Branches:

- biz...t.mail_notification
- biz...ler.guitone.website
- biz...tcampus.qt-workshop
- biz...er.monotone-nightly
- biz.thomaskeller.test
- biz...est.restricted_diff
- biz...ler.trac.i18nstatus
- biz...eller.vincents-site
- biz...maskeller.worktimer
- net...ge.monotone.guitone
- net...uitone.releases.0_6
- net...uitone.releases.0_7
- net...uitone.releases.0_8
- net...ne.guitone.renaming
- net...parser_optimization
- net...e.trials.fs_watcher
- net...rials.graph_display

Tags:

- foo
- goo-1.2.3
- guitone-0.2

Root /

| File | Age | Message | Size |
|-------------------------------|--------------------|--|-----------|
| .mtn-ignore | 1 month 13 days | me@thomaskeller.biz: * add autobuild client code | 124 bytes |
| COPYING | 1 month 14 days | me@thomaskeller.biz: * add the license file | 35.15 kB |
| README | 1 month 14 days | me@thomaskeller.biz: * add a basic README | 1.08 kB |
| autobuild | | | |
| docbuild | | | |
| pull.sh | 1 month 14 days | me@thomaskeller.biz: * initial setup for nightly builders | 211 bytes |
| vars.template | 1 month 14 days | me@thomaskeller.biz: * bail out if no database is found * configure OBS_PROJECT and put this into autobuild/work | 297 bytes |



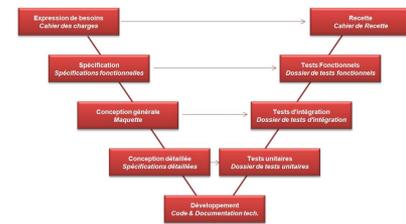
Serveur web :

- multi-sites, multi-user
- Doc, Wiki,
- Suivi de version : Git
- Tâche / tickets
- ...

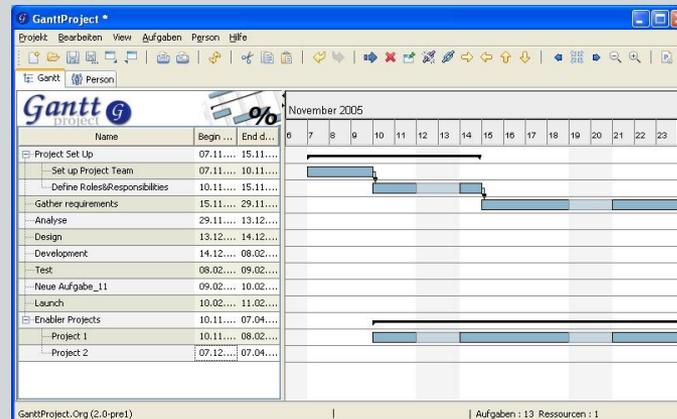
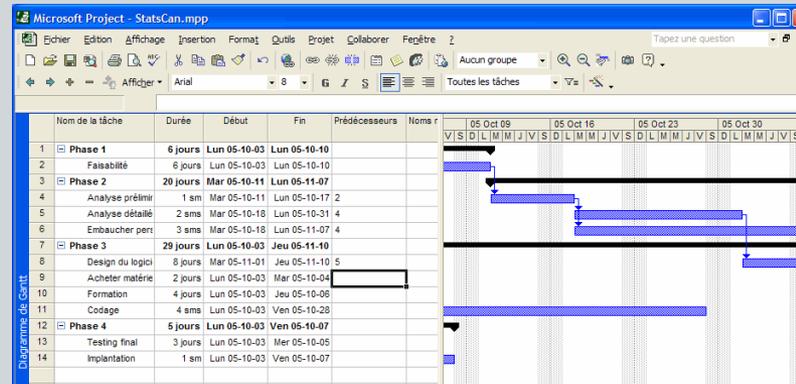
→ plus orienté développement.



4 – Autour du Cycle en V



Les outils du Chef de Projet et de l'équipe : les plateformes de pilotage intégrée (comprenant de nombreuses fonctions)



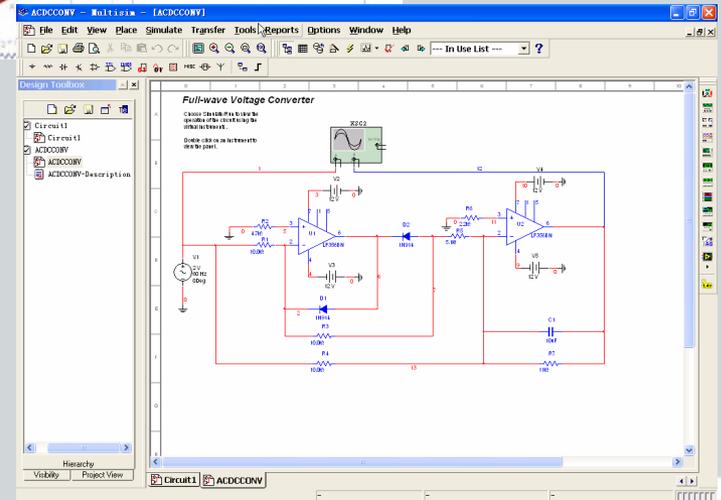
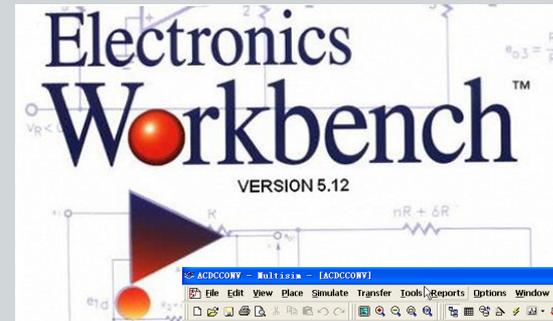
TITRE DIAPO

Les outils du Chef de Projet et de l'équipe : modélisation matérielle

Eagle...saisie, simulation, gestion commande des composants



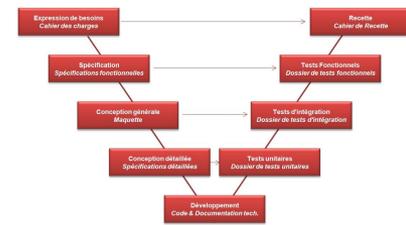
Eagle



Electronic Workbench
Multisim ...

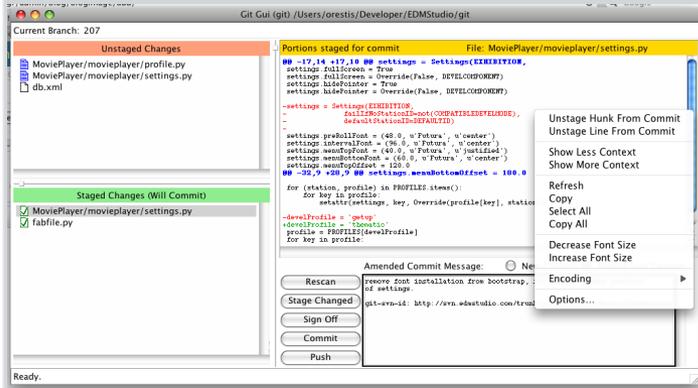


4 – Autour du Cycle en V

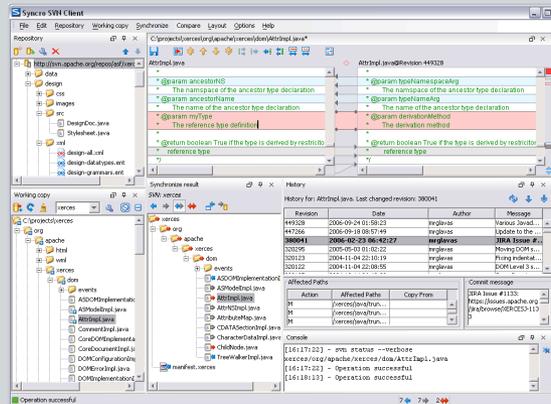


Les outils du Chef de Projet et de l'équipe :

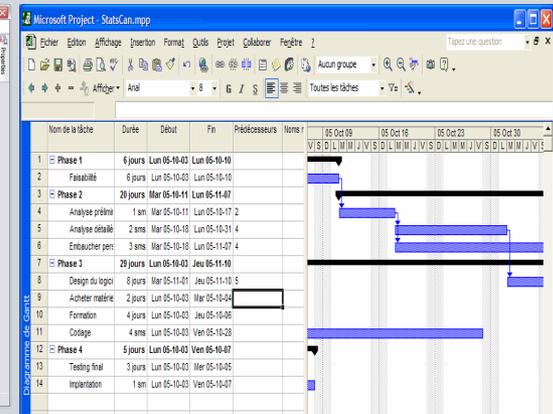
→ travail à plusieurs, merge, suivi de version, tag, sauvegarde... outil important !



Git



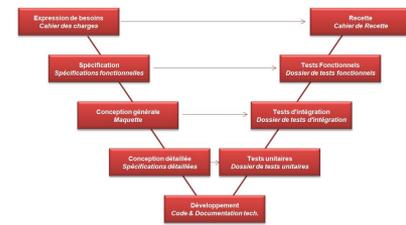
SVN



MKS

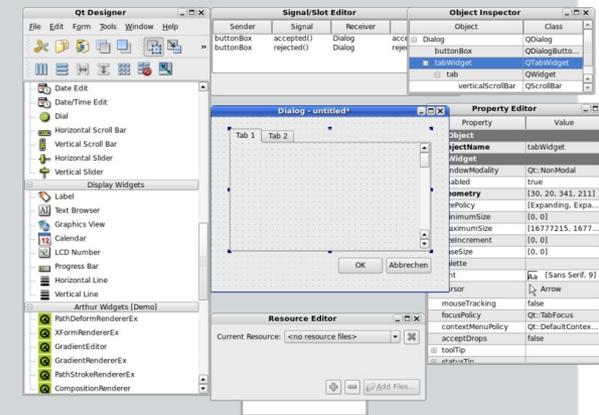
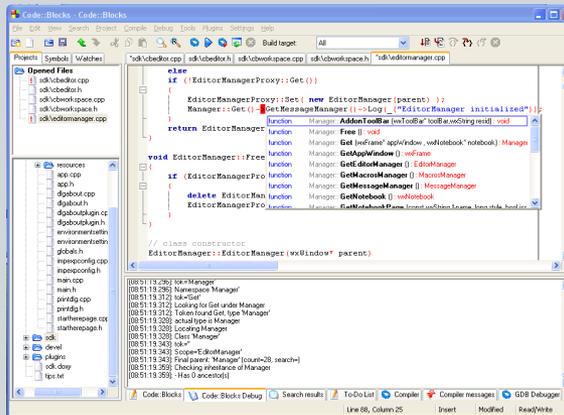
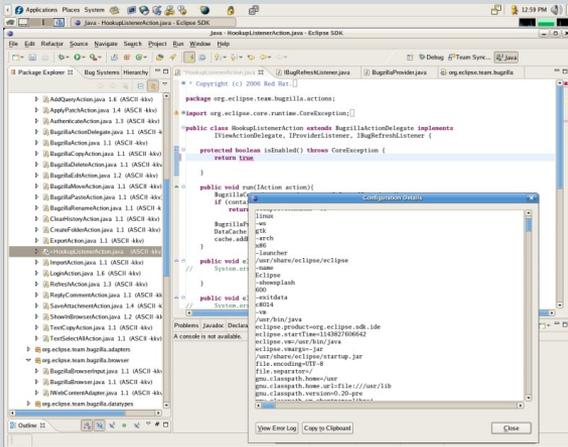
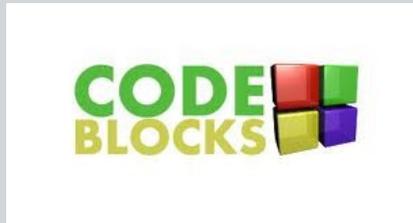


4 – Autour du Cycle en V



Les outils du Chef de Projet et de l'équipe :

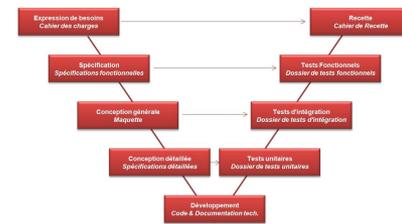
→ les IDE : environnement de développement intégré



... **Critères** : configurable (compilo, debugger, cross-development,...), multi-PF, multi-langages, liés à un suivi de version, simple d'utilisation...



4 – Autour du Cycle en V

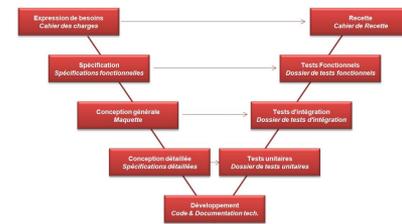


Qualité RH de l'ingénieur Chef de Projet : (qui met en oeuvre le Cycle en V)

- **Savoir faire** : expertise technique
 - Savoir faire pour pouvoir faire-faire (légitimité)
- **Savoir être** :
 - Avec son équipe, sa hiérarchie
 - Dans sa relation avec le client, les sous-traitants
- **Savoir communiquer** :
 - Parler, rédiger (!), réagir (lever / anticiper les pbs)
 - Ecouter, se mettre « à la place de »
 - Motiver, impliquer son équipe
 - à tout instant chacun sait où on en est
 - Mieux vaut 1 peu trop de com' que 1 peu pas assez !



4 – Autour du Cycle en V



Transposition à la Recherche (R&D) :

Sujet de recherche

Ex. : sujet de thèse

Publications,

Médaille Fields, prix nobel

Bibliographie

Identification des verrous sc

Généralisation,

Rédaction de Guides,
rapport de thèse..

Modélisation

- Analytique / Matlab
- Elements finis / Femlab

Ajustement et

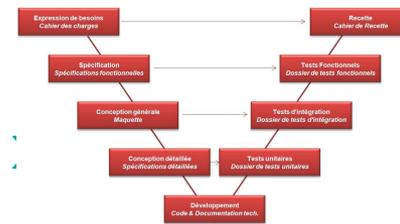
recalage du modèle

Validation du modèle

- Manip en labo
- Simulation

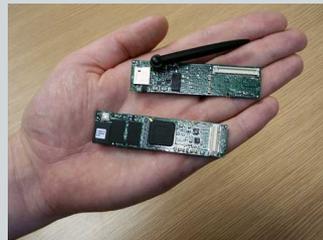


Recul sur la méthode du « Cycle en V » :

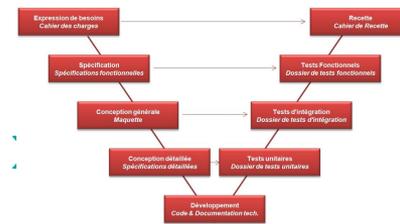


Critiques :

- Les méthodes de type « Cycle en V » ont tendance à enfermer les acteurs dans leur rôle (MOA/MOE) Donneur d'ordre / Sous-traitant un peu trop français...
- Une application trop stricte de la méthode ne favorise pas la créativité, la prise (saine) de risque...
- Le cycle en W : surtout si expression du besoin pas clair ou compréhension Client/Fournisseur pas parfaite (ou si métiers très différents)
- Théorie vs pratique... anticiper la conception qui peut arriver trop tardivement (surtout dans un monde actuel d'intégration) → intérêt du Cycle en W
 - Seuls 16% des projets respectent le temps prévus initialement
 - Tendance actuelle : des solutions en briques, être **Architecte Système**
- Ici 80% de votre solution existe:



Recul sur la méthode du « Cycle en V » :

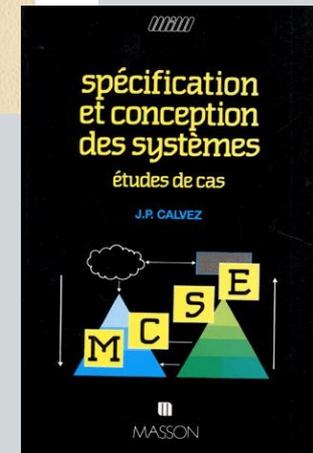
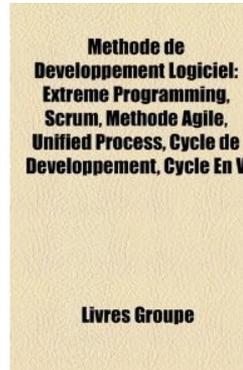


Bibliographie :

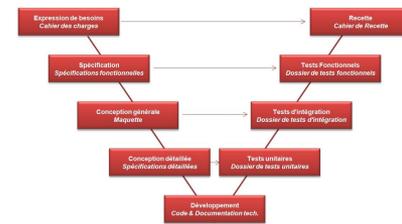
- Net, Wikipedia, wikibooks, chez Amazon
- **MCSE** : Méthode de Conception pour les Systèmes Electroniques – JP Calvez
- Méthode du Plan de Test de Validation – F. Barbot ; Alcatel
- ...

Autres méthodes:

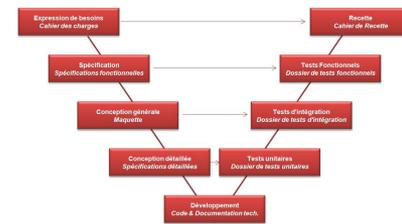
- **Agile** (très en vogue) : Les méthodes agiles sont itératives et participatives, on développe des petits morceaux d'application qui apportent de la valeur, qu'on fait valider par l'utilisateur avant de passer aux suivants. C'est l'utilisateur qui pilote au fur et à mesure le développement à travers le choix des fonctionnalités qu'il souhaite voir développer. Un cycle de développement, une itération, prend 1 à 3 semaines.
- **Comparaison** : <http://fleid.net/2012/01/10/gestion-de-projet-decisionnel-methodes-agiles-ou-cycle-en-v/>



Conclusion / perspectives



Conclusion / perspectives



- **Pas de projet réussi sans une méthode !** Cycle en V ou autre...
- Notons qu'il y a des « méthodes dans la méthode » : Modèle de Spec Fonctionnelle, règles de codage, MCSE, méthode des Tests de Validation, ...
- Savoir **prendre du recul** :
 - La Méthode ne fait pas le projet, c'est un concept, un outil parmi d'autres du CdP
 - **Savoir adapter** : appliquer le Cycle en V à la taille du projet (montant, équipe,...), pas d'application brutale, **rester critique**... passer par du Cycle en W (validation préalable)
- Avantage du Cycle en V (et des outils : Redmine) : **on formalise, on écrit...et ainsi on capitalise** ...sur ce qui marche et ce qui n'a pas marché. On progresse.
- Intégrer le contexte actuel : intégration d'autres briques existantes et d'autres entités interrégionales : SDK, OS Linux, reprise de code *open-source*... vers de nouveaux métiers « **Architecte système** » => savoir trouver, **identifier et qualifier des briques existantes**.
- Relier la méthode au syst qualité de l'ets ISO 9001 (préformattés qualités ; utilisation de Redmine...)
- **Méthode utilisée par les SSII** (80% du recrutement des ingés) => adaptée au client / fournisseur
- Le chef de projet doit avoir et développer **des compétences : savoir faire, savoir être, communiquer**...qui ne sont pas si courantes. On peut se distinguer sur le management technique- Compétence recherchée.



MERCI...à votre écoute !

