

CMMI

CAPABILITY MATURITY MODEL INTEGRATION

Valentin Fumez | Quentin Lematte
SCIENCES U LYON | EII 2016-2018 GROUPE B

Table des matières

Qu'est-ce que CMMI ?	2
Principe de fonctionnement	2
Les modèles originaux	2
Les 5 niveaux de maturité	3
Niveau 1 – Initial, ou « ère des héros »	3
Niveau 2 – Géré	3
Niveau 3 – Défini	3
Niveau 4 – Quantifié	3
Niveau 5 – Optimisé	3
Les domaines de processus	4
Application de CMMI dans un projet d'entreprise	6
Au cœur d'un projet CMMI / Symphony	6
Quels sont les outils mis à disposition par CMMI ?	7
Les 5 phases d'un projet CMMI/Symphony	7
La phase Proposal	7
La phase Définition	8
La phase Design	8
La phase Construction	8
La phase Transition	9
Forces et limites du modèle	9
Forces du modèle	9
Limites du modèle	9

Qu'est-ce que CMMI ?

C.M.M.I. (pour **Capability Maturity Model Integration**) est un modèle de *référence*, un ensemble structuré de bonnes pratiques concernant la gestion de projet.

Elle a été développée par le Software Engineering Institute (SEI) en 1989 pour le Département de la Défense des Etats-Unis sous le nom **C.M.M.** (**Capability Maturity Model**).

Principe de fonctionnement

C.M.M.I. fonctionne grâce à un système de « *Modèle* ». Chaque modèle est décomposé en « Niveaux de maturité », qui eux-mêmes sont divisés en « Domaines de processus », puis en « Buts spécifiques ».

Pour définir le niveau de maturité d'un système, un certain nombre de buts spécifiques sont à valider.

Tout ceci débouche sur une certification, définie grâce aux niveaux de maturité. Cette certification est attribuée par le SEI.

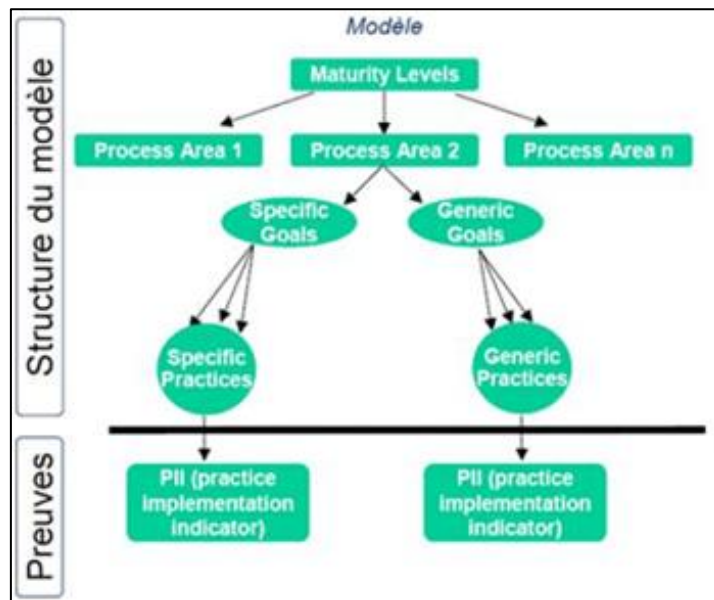


Schéma descriptif de C.M.M.I.

Les modèles originaux

Pour C.M.M., les modèles étaient au nombre de 6 :

- SW CMM – Software CMM _ Le modèle original
- SE CMM – System Engineering CMM
- SA – Software Acquisition
- IPD – Integrated Product Development
- People – Management des personnes
- Supplier Sourcing

En 2001, lorsque la refonte de CMM est débutée (qui deviendra, au lancement de cette dite refonte en 2006, CMMI), l'ensemble des modèles ont été refaits afin d'être simplifiés. Un socle « Commun » est instauré pour chaque projet / système souhaitant être évalué. Il ne restera plus que trois modèles à cette refonte :

- CMMI-DEV _ Dédié au développement
- CMMI-SVC – Dédié aux services

- CMMI-ACQ – Dédicée aux acquisitions

La partie management des personnes est complètement occulté lors de cette refonte.

Les 5 niveaux de maturité

Niveau 1 – Initial, ou « ère des héros »

Le niveau le plus bas montre que l'organisation n'est pas prête, et le projet pas stable. Ce dernier dépend d'une poignée de personnes, qui ne font pas appel à des processus éprouvés. Il se peut cependant que le projet aboutisse, mais en dépassant certainement le budget et le temps alloués. Le projet ne construit pas sur les succès passés.

Niveau 2 – Géré

Le projet construit sur ce qui a été appris précédemment, en faisant appel à une certaine discipline et à une gestion de projet basique. De fait, le projet est géré selon les plans, avec étapes-clefs et vérification des coûts et des fonctionnalités.

Niveau 3 – Défini

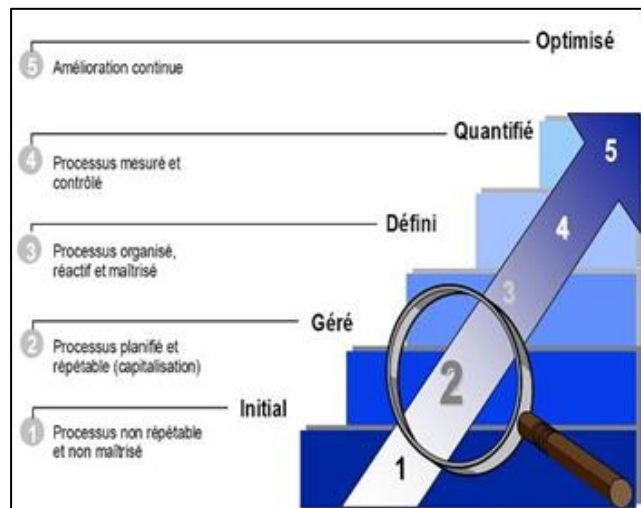
Ce n'est plus le projet qui dispose d'une bonne discipline, mais l'ensemble de l'organisation, de manière cohérente. Tous les projets s'en trouvent améliorés.

Niveau 4 – Quantifié

Les efforts de mesure et de gestion autorisent un contrôle sans effort du développement, avec capacité d'ajuster et adapter des projets précis sans troubler les autres. Les performances des processus sont prévisibles en quantité comme en qualité.

Niveau 5 – Optimisé

Les processus sont constamment améliorés de manière incrémentale et innovante. Les objectifs sont revus en permanence pour rester proches des besoins du marché. Les évolutions sont anticipées et gérées de bout en bout.



Les 5 différents niveaux CMMI

Les domaines de processus

Ci-dessous vous trouverez le détail des différents domaines de processus concernant CMMI :

CMMI-DEV pour le développement de systèmes

Processus spécifiques

RD – Requirements Development (m3)

Développement des exigences

Identifier et documenter les exigences du produit

TS – Technical Solution (m3)

Solution technique

Identifier, concevoir et réaliser une solution technique en réponse aux exigences

PI – Product Integration (m3)

Intégration produit

Assembler et livrer le produit

VER – Verification (m3)

Vérification

Contrôler que le produit répond aux besoins opérationnels

VAL – Validation (m3)

Validation

Contrôler que le produit est conforme aux exigences

CMMI-ACQ pour la maîtrise des activités d'achat

Processus spécifiques

AM – Agreement Management (m2)

Gestion de l'accord d'acquisition

Assurer l'exécution de l'accord par chacune des parties (donneur d'ordres et fournisseur)

ARD – Acquisition Requirements Development (m2)

Développement des exigences d'acquisition

Identifier et documenter les exigences client et contractuelles

SSAD – Solicitation and Supplier Agreement Development (m2)

Développement du dossier de consultation des fournisseurs et accords fournisseurs

Préparer un dossier de consultation des fournisseurs, sélectionner le ou les fournisseurs et documenter l'accord fournisseur

ATM – Acquisition Technical Mangement (m3)

Gestion technique dès l'acquisition

Evaluer la solution technique du fournisseur et gérer les interfaces entre l'acquisition et le système où elle s'intègre

AVAL – Acquisition Validation (m3)

Validation de l'acquisition

Contrôler que l'acquisition répond aux besoins opérationnels

AVER – Acquisition Verification (m3)

Vérification de l'acquisition

Contrôler que l'acquisition est conforme aux exigences

CMMI-SVC pour la fourniture de service

Processus spécifiques

SD – Service Delivery (m2)

Fourniture du service

Fournir les services conformément aux accords de service

CAM – Capacity and Availability Management (m3)

Gestion de la capacité et disponibilité

Assurer la disponibilité et la capacité nécessaire pour répondre aux exigences du service

IRP – Incident Resolution and Prevention (m3)

Prévention et résolution des incidents

Prévenir et résoudre les incidents de services

SC – Service Continuity (m3)

Continuité du service

Documenter et appliquer les plans des activités nécessaires à assurer la continuité de service

SSD – Service System Development (m3)

Développement du système nécessaire au service

Concevoir et mettre en place un système pour satisfaire aux besoins des accords de service

SSM – Service System Management (m3)

Gestion du système nécessaire au service

Mettre en place des services standards alignés sur les besoins et plans stratégiques

Application de CMMI dans un projet d'entreprise

Au cœur d'un projet CMMI / Symfony

Le modèle de maturité n'est pas une méthodologie projet et ne propose donc aucun livrable et aucun processus décrit de manière détaillée.

Voici un exemple d'application du modèle de maturité CMMI chez BNP Paribas qui a été décliné de la manière suivante :

Définition des rôles et responsabilités :

Rôles et responsabilités de domaine

- Le Responsable de Domaine doit gérer les activités dans le périmètre de son domaine, ce qui inclut : projets, maintenance releases, interfaces avec les comités de pilotage, etc. Il est responsable du plan annuel de domaine et reporte en général à l'Execution Manager.
- L'Application Manager gère l'activité de maintenance de son application, ce qui inclut à la fois les maintenance releases mais aussi les emergency patch. Il reporte en général au Domain Manager.
- Le Configuration Manager gère la bibliothèque de développement de l'application (code et documentation), construit les releases, qu'elles soient internes ou externes.
- Le Test Manager définit la stratégie de tests, planifie les tests, gère l'exécution des tests et des UAT (User Acceptance Tests).
- Le Supply Manager gère la relation fournisseur (sélection, contrats, audit, etc.).

Rôles et responsabilités de projet

- Le Project Manager définit le project plan, gère les risques, les problèmes et les changements. Il suit la réalisation du projet conformément à son plan, reporte et communique sur l'avancement et l'état. Il réalise la clôture du projet.
- Le Business Analyst élabore et gère les requirements, produit les cas de tests d'acceptation si nécessaire, conduit les tests nécessaires lors de la création de nouvelles fonctionnalités. Son rôle est très proche de celui de Project Manager.
- Le développeur IT fournit l'estimation des charges de son travail, développe les composants nécessaires, effectue les tests unitaires requis, écrit les spécifications fonctionnelles et réalise les révisions de code.
- Enfin, le coach CMMI assiste toute personne qui le demande dans la définition de son projet, ou dans le processus de releases, identifie et reporte les non conformités à CMMI et en suit la résolution.

Quel est le cycle de vie d'un projet dans un cadre CMMI ?



Chez BNP Paribas, le cycle de vie d'un projet CMMI a été rythmé des phases suivantes : **Proposal** > **Definition** > **Design** > **Construction** > **Transition** > **Operations**, phases qui couvrent l'ensemble de la démarche depuis « Je pense que je vais lancer un projet » à « Le Go Live du projet a eu lieu ».

Il s'agit aussi de jalons importants, telles que *checkpoint*, *reviews*, etc.

Au sein de chaque phase, plusieurs activités doivent être réalisées, chacun comprenant un certain nombre de livrables et de tâches à effectuer, avec des outils prédéfinis. Tous les *templates* nécessaires dans le cadre de la méthodologie CMMI/Symphony sont mis à disposition de tous.

Quels sont les outils mis à disposition par CMMI ?

- Un outil d'estimation des charges, fortement apprécié des développeurs
- Un outil Diagramme de Gantt
- Une gouvernance, avec définition des rôles de chacun des acteurs du projets

Les 5 phases d'un projet CMMI/Symphony

La phase Proposal

Cette phase ne dure pas plus de 2 à 3 semaines. Elle répond à la question : Dois-je décider de lancer un projet ?

La réponse est oui si :

- Le projet est pertinent d'un point de vue business
- Le projet est réalisable d'un point de vue IT
- Les demandes en provenance du business sont les inputs de cette phase, et elles parviennent à l'IT sous des formats divers (tickets dans le système de remontée d'incidents, documents Word détaillés, demande de nouvelle fonctionnalité ou de modification, etc.).

Pour des projets de taille significative, il peut être obligatoire de faire valider les dépenses engendrées.

Si l'approbation du projet est obtenue, on commence alors la phase de définition.

La phase Définition

Il s'agit ici de définir suffisamment précisément le projet afin d'obtenir les ressources nécessaires à sa réalisation.

On aura notamment besoin de :

- Spécifications fonctionnelles détaillées
- Plan de gestion des livrables
- Matrice de traçabilité
- Du compte-rendu du Kick-Off
- Des estimations de quantité de travail, de charge, de ressources et de coûts
- D'un plan de gestion de projet
- D'un planning détaillé (Gantt)
- D'un plan de gestion de la configuration
- D'un tableau de bord du développement

Si l'approbation technique du projet est obtenue, et le budget validé, on commence alors la phase de design.

La phase Design

Il s'agit ici de définir suffisamment précisément la solution afin d'être capable de l'implémenter.

On aura notamment besoin de :

- Révision technique complémentaire
- Révision éventuelle du budget
- Design de la stratégie de tests du système
- S'assurer que les développements et les environnements de tests sont complètement installés

Si l'approbation du design de la solution est obtenue, tant en termes techniques que de périmètre et de budget (si revu), on commence alors la phase de construction.

La phase Construction

Il s'agit ici de construire et tester la solution afin qu'elle soit acceptée par l'utilisateur final.

On aura notamment besoin de :

- Livrer la version avec un code cohérent,
- Se préparer pour la phase transition (SLA, trainings, manuel utilisateurs, etc.)

Si la validation de la solution est obtenue, on commence alors la phase de transition.

La phase Transition

Le système a été installé, il faut donc contrôler sa bonne implémentation et sa stabilité d'un point de vue opérationnel.

Le projet étant sur le point de se terminer, il est important d'enregistrer toute recommandation d'amélioration (lessons learned). On prendra également la décision d'effectuer une Post-Implémentation Review ou non.

Forces et limites du modèle

Forces du modèle

- Il permet d'aborder les compétences de l'entreprise, de son organisation et non pas de l'individu.
- Il décrit les processus qu'il est opportun de maîtriser.
- Il répond de manière précise et efficace.
- Il limite les conflits par anticipation.
- Il demande l'identification et l'assignation d'une tâche à une ressource.
- Il permet l'uniformisation des process et des documents.
- Il améliore les conditions de travail.

Limites du modèle

- La méthode de qualification est différente d'une conduite de projet.
- Il ne regarde ni vers l'amont, ni vers l'aval du projet.
- Il s'attache au processus et non à la bonne utilisation des ressources.
- Il ne garantit aucun résultat.
- Il ne répond pas à toutes les attentes (modifications inattendues faites par des clients par exemple).