

Contrôle continu - SIRIS

Gestion de Projet

Contact:

Yossi Gal, yossi.gal@galyotis.fr, Téléphone: 06 8288-9494



Contrôle 22-09-2011 / 15 minutes

1. Donnez la signification des sigles suivants et expliquez en une ligne ce qu'ils signifient.
 - ✓ PMI,
 - ✓ PMBOK,
 - ✓ PMP,
 - ✓ SEI,
 - ✓ CMM,
 - ✓ WBS,
 - ✓ ATP.
2. Quelle est la **définition d'un projet** selon PMI? Expliquez les caractéristiques principales.
3. Quels sont les éléments de la **triple contrainte**? Comment fonctionne la relation entre ces éléments?

Questions ?



Yossi Gal
Galyotis
IT Business Applications
yossi.gal@galyotis.fr

Les Activités de la méthodologie

- Comité de **Pilotage**
- Étapes de **Validation** (ATP)
- **Revues** de Projets, Minutes, Liste des Actions
- Gestion des **Risques** , Assurance **Qualité** , Gestion de **Configuration**, Méthodes d'**Estimation**
- **Jeux d'Essai**, Plans de **Test** et **Procédures** de Test
- **Coordination** avec d'autres projets
- **Documentation**: Utilisateur, Technique
- **Formation** pour les équipes de Développement, pour les Utilisateurs
- Collection des **Métriques**
- Documentation de la Démarche, **Leçons Apprises**



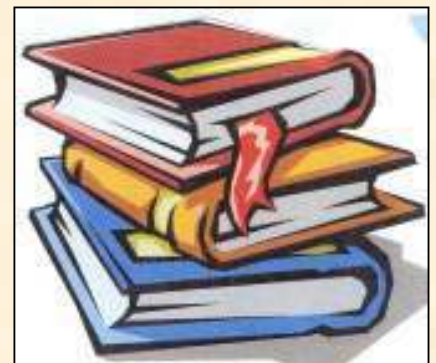
Les Documents de la Méthodologie

La Pré-étude

- Le Document de Pré-Étude
- Le Document de La gestion des Risques

La Planification:

- Le Document de La Gestion de la configuration
- Le Document de L 'Assurance Qualité
- Le Document de La Planification
- Le Document du plan des Tests
- Le Document du plan d'Installation et de Support
- Le Document du plan de Formation



L'analyse:

- Le Document de L 'analyse Fonctionnelle Détaillée
- Le Document de La Conception Technique

La Construction:

- Le Document de construction
- La documentation Utilisateur



Le Document de Pré-Étude

- Le document de **pré-étude** regroupe l'essentiel des éléments rassemblés au cours de la phase initiale du projet.
- Donne un **aperçu global du projet** et de l'objectif principal à atteindre
- Décrit les **besoins initiaux** exprimés par les utilisateurs
- **définit le produit à concevoir** en citant ses principales caractéristiques et ses fonctionnalités.
- Résume les **différentes approches étudiées** pour résoudre le problème, met en évidence les **points forts** et **les limites** de chaque solution étudiée, et **propose la solution à choisir**



Le Document de Pré-Étude (2)

- **Décompose le projet en éléments atomiques** facilement estimables, mesurables et gérables
- Établit les charges, les coûts et les délais en **appliquant les méthodes d'estimation** sur les composants de base du projet
- Établit la **répartition budgétaire** du coût total du projet sur la durée totale du projet
- Établit un **planning de livraison** de fonctions principales avec un ordre de priorité proposé et qui pourra être modifié par les utilisateurs.
- **Doit être validé par le commanditaire du projet** avant de passer à l'étape suivante du projet.
- En cas de **non acceptation**, le **projet s'arrête**.



Le Document de La gestion des Risques

- Le plan de **gestion des risques** est le document qui gère les risques attachés au projet.
- **Identifie les risques** et les **conséquences** qu'elles peuvent avoir sur le projet s'ils venaient à se réaliser.
- résume les **risques principaux** en citant pour chaque risque sa **probabilité**, son niveau d'**impact** sur le projet, la **priorité** avec laquelle il doit être traité, les **facteurs contribuant** à sa réalisation ainsi que des **solutions préventives** pour réduire la probabilité du risque ou au moins les conséquences de son impact sur le projet.
- **Faire prendre conscience** au chef du projet et à son équipe, des **obstacles** qui peuvent entraver le bon déroulement du projet.



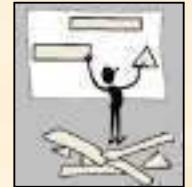
Le Document de La Gestion de la configuration

- Le plan de **la configuration** est le document qui se focalise sur le **contrôle du développement**, de la **sauvegarde**, des **mise à jours** des programmes.
- Il permet **d'identifier tous les objets du projet** (Programmes, Documents).
- Il établit des **standards** de **nomenclature** concernant les **releases**, les **objets**, le **langage de programmation** et autres outils utilisés.
- Il participe à la maintenabilité des produits du projet et permet de **constituer toutes les releases** du projet.
- Il décrit **l'organisation du projet**, c'est à dire l'équipe qui participe au projet et les rôles de chacun.



Le Document de La Gestion de la configuration (2)

- Il établit la liste des **outils utilisés** au cours du projet (Word, Excel, Lotus Note, PowerPoint etc....), les **méthodes d'analyse** et les langages de programmation.
- Au niveau du code il va définir des **standards d'écriture** que le développeurs doivent respecter.
- Il décrit la **procédure de sauvegarde** des Objets du projet, en constituant un **référentiel de base** (*Baseline*).
- Il décrit aussi la **procédure de mise a jour** des objets (*Check in/Check out*).
- Tout **changement** à l'un des objets du projet fait l'objet d'une **procédure écrite** et formelle.



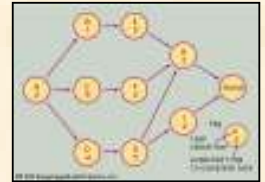
Le Document de L'Assurance Qualité

- Le plan **d'assurance qualité** établit les droits et les devoirs du projet en ce qui concerne la qualité.
- Il décrit les **vérifications (Audits)** que doit effectuer le responsable qualité pour s'assurer que le projet agit selon la méthodologie.
- Ce document décrit la liste des **taches** qui doivent être effectuées par le projet, les **dates d'audit** planifiées et la **procédure à suivre en cas de litige** entre le responsable qualité et le projet.
- Chaque **inspection** donne suite à un rapport avec la W3 liste (quoi, qui, quand) et au suivi des actions
- Le responsable qualité doit être **indépendant du projet**



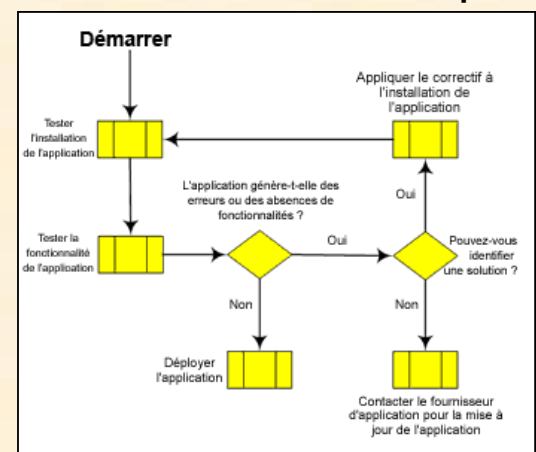
Le Document de La Planification

- Le dossier de **planification** contient le planning de tout le projet.
- Généralement il est documenté a travers un outil de planification (**MS Project**, Project Workbench,..)
- Il doit contenir au minimum:
 - ✓ Le **Gantt des tâches** (Tache/date début/date fin/délais).
 - ✓ Le récapitulatif des **ressources disponibles , utilisées**.
 - ✓ Le récapitulatif du **coût total du projet**.
 - ✓ Et éventuellement le **chemin critique**



Le Document du plan des Tests

- Le **plan des tests** décrit les différents types de tests qui doivent être effectués, quand ils doivent être effectués et par qui.
- 4 types de tests
 - ✓ Les tests **unitaires**
 - ✓ Les Tests **d'Intégration**
 - ✓ Les tests de **non-régression**
 - ✓ Les tests **systèmes**
 - ✓ Les Tests **d'acceptance** utilisateurs
- Les **Procédures de Tests** et les **jeux d'essais**
- Le rapport des tests et la liste des actions correctives



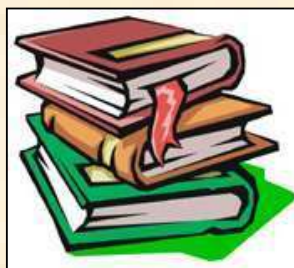
Le Document du plan d'Installation et de Support

- Le **plan d'installation et de support** décrit l'ensemble des **procédures** nécessaires à la **mise en production**, ainsi que les procédures nécessaires pour la **maintenance** du produit installé
- Il inclut la **configuration matérielle et logicielle** à mettre en place pour pouvoir installer le produit fini
- Il décrit les procédures de **conversion des données**
- il définit les **procédures d'intervention** en cas de problèmes
- il référence le plan de configuration pour les procédures de sauvegarde et de Baseline



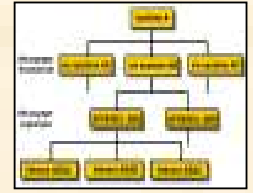
Le Document du plan de Formation

- Le **plan de formation** décrit l'ensemble du dispositif à mettre en place pour
 - ✓ la **formation de l'équipe du projet** aux techniques nécessaires au développement du projet,
 - ✓ Et la **formation des utilisateurs** sur le logiciel développé
- Ce plan détaille le **matériel nécessaire** à ces formations, ainsi que les **organismes** internes et externes qui doivent l'assurer
- Il doit également contenir le chiffrage des **coûts** de l'ensemble de la formation et du matériel



Le Document de L 'analyse Fonctionnelle Détaillée

- **L'analyse fonctionnelle détaillée** ou les **spécifications** détaillées font suite au document de pré-étude, le complètent et le détaillent
- Il est **destiné** directement aux **utilisateurs**.
- Son but est de décrire les **comportements fonctionnelles** visibles du logiciel sans se soucier des **techniques** qui vont être utilisées pour **l'implémentation**
- Il précise les spécifications du produit une fois terminé, il peut s'apparenter à un **cahier des charges**.
- Il décrit en détail les différents **menus**, **fenêtres**, **règles de gestion** en plus des **performances** et **contraintes** du produit fini.
- Ce document doit être **approuvé par l'utilisateur** avant le passage a l'étape suivante (Conception Technique)



Le Document de La Conception Technique

- Ce document couvre **l'architecture Technique** du projet
- Il fait référence à **l'architecture globale** du logiciel en décrivant les formes, des menus, les Contrôles, etc....
- Il décrit les **interfaces** internes et externes au projet
- Il décrit les **outils de programmation** utilisés et la façon dont ils sont utilisés pour implémenter les fonctions requises dans les spécifications fonctionnelles.
- Il décrit également la **structure des données** et la logique de programmation.
- Il a pour vocation de diriger l'équipe de développement dans les différentes étapes du codage.



Le Document de construction

- Le **document de construction** inclut la liste des objets du projet constituant la release
- Cette liste contient le nom des objets, leur version, la taille, la date de dernière mise a jour, etc. ...
- Par **Objet** du projet on entend les **documents**, les **formes**, les **modules** principaux et tous les **exécutables** (DLL, ActiveX, Les java beans, Les Applets et Servlets, ...) et les **communications** (fax, mails, ..)
- Il **ne contient pas le code des programmes** qui eux se trouvent dans les répertoires de développement.



La documentation Utilisateur

- La **documentation Utilisateur** a pour objet de guider l'utilisateur dans l'utilisation du logiciel.
- Elle se compose d'une **aide en ligne**, du **manuel d'utilisation** et du **guide de référence**
- Il est exclusivement **destiné à l'utilisateur** et son contenu ne fait qu'expliquer le fonctionnement du logiciel de telle sorte que l'utilisateur non informaticien puisse l'utiliser.
- Il **n'explique que les fonctionnalités implémentées** et non tout ce qui a été décrit dans les spécifications ou dans le document technique.
- Il peut être utilisé comme **support de formation** pour les utilisateurs du logiciel



Les Taches par activité

- Les Taches de la phase de planification
- Les Taches de la phase de Spécifications
- Les Taches de la phase de conception Technique
- Les Taches de la phase de Construction
- Les Taches de la phase des tests
- Les Taches de la phase d'Installation
- Les Taches pour la phase de Maintenance



Les Taches de la phase de planification (1)

- Adapter et dimensionner les activités clés de la méthodologie à la taille du projet
- Définir les méthodes d'estimation
- Définir la Coordination inter-projets
- Créer les Dossiers du Projet
 - ✓ Définir la gestion des Risques
 - ✓ Définir la gestion de la Configuration
 - ✓ Définir le plan Assurance Qualité
 - ✓ Définir le Plan des tests et les jeux d'essais
 - ✓ Définir le plan d'installation et de Support
 - ✓ Définir le Plan de Formation
- Définir les Métriques à collecter
- Définir le Référentiel de Base (Baseline Repository)
- Définir les Objectifs et les besoins des utilisateurs (SOW)



Les Taches de la phase de planification (2)

- Procéder a l'Analyse des **solutions alternatives**
- Préparer le WBS (Work Breakdown Structure) initial et **Estimer la taille, les charges, les délais et les coûts du projet.**
- Identifier le **coordinateur des utilisateurs**, le **Chef du projet** et l'équipe du projet
- Mettre en place les **groupes d'interface** avec d'autres projets, le **comité de pilotage**
- Mettre en place un **planning de rencontres** avec les utilisateurs. Définir les **revues de direction** (types, fréquences, format). Définir et mettre en place le plan des **revues du projet**
- Organiser **l'équipe des tests**
- Définir **l'environnement des tests**
- **Communiquer les plans** à l'équipe du projet



Les Taches de la phase de planification (3)

- Finaliser le Document de **Pré-Étude**
- Générer des rapports sur **l'état du projet**
- Obtenir **l'acceptation** du projet par la direction (**ATP0**)
- Collecter
 - ✓ la **taille** des objets du projet
 - ✓ la **charge** de travail,
 - ✓ Les **délais**
 - ✓ les **jalons**,
 - ✓ les **coûts**,
 - ✓ la **stabilité des changements** aux spécifications
 - ✓ Et autres indicateurs ...



Les Taches de la phase de Spécifications

- Commencer l'analyse des spécifications
- Identifier la documentation utilisateur
- Documenter les besoins en formation et l'approche générale de la formation
- Rédiger le cahier des charges (Spécifications)
- Conduire la revue de spécifications
- Conduire la revue de direction pour l'obtention de l'ATP1
- Collecter les métriques relative à
 - ✓ la taille des produits,
 - ✓ la Charge de travail,
 - ✓ les jalons importants,
 - ✓ les coûts
 - ✓ Et le niveau de la stabilité des spécifications



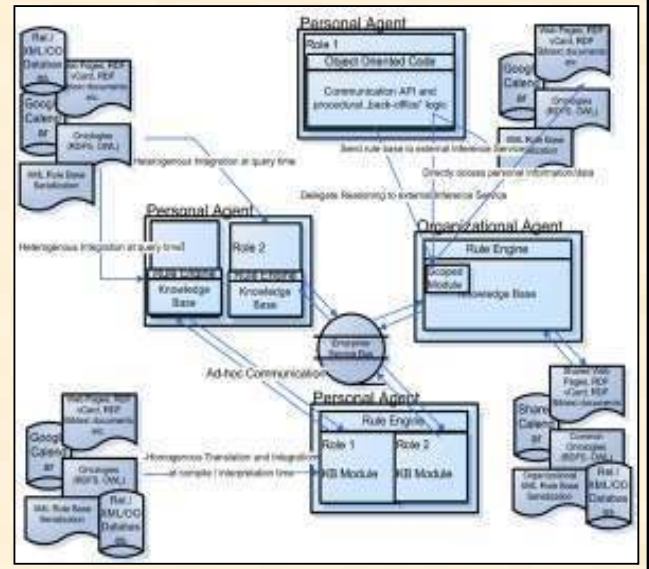
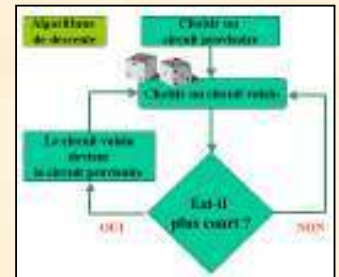
Les Taches de la phase de conception Technique (1)

- Définir et Documenter les charges nécessaires pour le Prototypage ainsi que sa faisabilité
- Revoir et mettre à jour les besoins utilisateurs
- Développer et démontrer un Model de l'application
- Définir les taches hors Prototypage
- Documenter les résultats du Prototypage
- Conduire la revue préliminaire de la conception Technique
- Rédiger le document la conception Technique
- Revoir et mettre a jour le plan de formation
- Définir la Documentation utilisateur préliminaire
- Revoir et mettre a jour le plan d'Installation et de Support
- Conduire la revue critique de la conception Technique



Les Taches de la phase de conception Technique (2)

- Demander la validation **ATP2**
- Collecter les métriques pour
 - ✓ Les délais,
 - ✓ les **spécifications déjà implémenter**,
 - ✓ la taille des produits,
 - ✓ la charge de travail,
 - ✓ Les jalons importants,
 - ✓ Les coûts constatés
 - ✓ Et la **stabilité des spécifications**



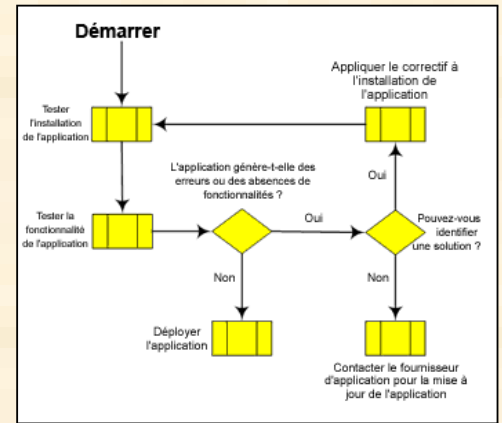
Les Taches de la phase de Construction

- Développer et revoir le **code**
- Définir les **procédures de tests**
- Rédiger la **documentation utilisateur**
- Constituer le matériel de **Formation** et conduire les séances initiales
- **Conduire les tests unitaires**, les tests de **non-Régression**
- Revoir et mettre à jour le **plan d'Installation et de Support**
- Collecter
 - ✓ **L'évolution des délais détaillés**,
 - ✓ Les spécifications déjà implémentées,
 - ✓ La taille des objets,
 - ✓ La charge de travail,
 - ✓ Les jalons importants,
 - ✓ Les coûts,
 - ✓ Et la **stabilité des spécifications**



Les Taches de la phase des tests

- Conduire les tests d'Intégration
- Conduire les tests de non-Régression
- Vérifier le matériel de Formation
- Conduire les tests systèmes
- Conduire les tests d'acceptance
- Documenter les résultats des tests
- Collecter les erreurs des tests et les autres métriques



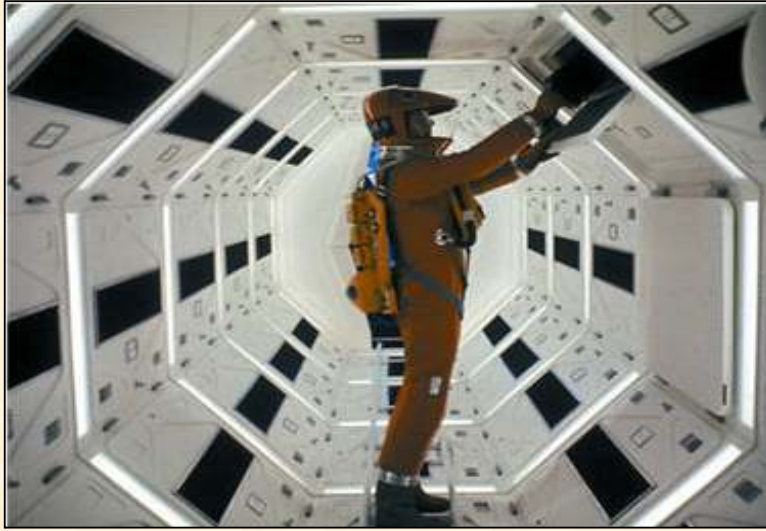
Les Taches de la phase d'Installation

- Finaliser la liste de vérification pour l'installation
- Conduire les validations de la configuration
- Conduire la revue de mise en production
- Demander l'autorisation pour la mise en prod (ATP3)
- Procéder à l'Installation
- Finaliser et Exécuter le plan d'installation et de Support
- Conduire la Revue de Poste-Installation
- Collecter les métriques



Les Taches pour la phase de Maintenance

- Revoir et mettre à jour le **plan d'installation et de Support**
- Définir et Exécuter les **procédures de maintenance**
- Revoir et prioriser la liste des **requêtes mises en attente pour la prochaine version**



Résumé (1)

- **La production de systèmes d'information pose des problèmes :**
- **Délais, Coûts, Qualité, Changements rapides des technologies** de l'information, **Restructuration des activités de l'entreprise, ...**
- Besoin d'avoir une **Démarche Méthodologique** afin de Produire des **systèmes de Qualité**.
- Pour **Comprendre, Prévoir, Contrôler et Maîtriser le cycle de développement** d'un Projet Informatique.
- Prendre les **Bonnes Décisions** relatives aux **coûts et bénéfices** d'un système d'information et **devenir plus Compétitif**.
- **Planifier, Communiquer, faire le Suivi, faire Valider** chaque étape de la définition, l'exécution et la mise en production.
- **Améliorer continuellement la démarche** pour faire **bon usage de la technologie au service de l'entreprise**.



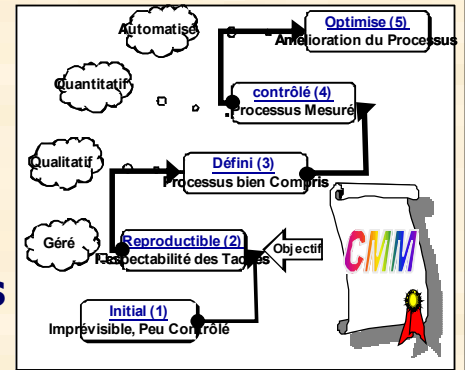
Résumé (2)

5 niveaux de Maturité:

- Niveau 2 - **Répérable, Intuitif**

Les Domaines Clés d'activité:

- **Planification**
- Gestion des Requêtes/**Besoins Utilisateurs**
- Le **Suivi de Réalisation**
- **L'Assurance Qualité**
- La **Gestion de Configuration**
- La Gestion des Ressources Externes
- Construction
- Tests
- Installation



Résumé (3)

- Comité de **Pilotage**
- Étapes de **Validation** (ATP)
- **Revues** de Projets, Minutes, Liste des Actions
- Gestion des **Risques**, Assurance **Qualité**, Gestion de **Configuration**, Méthodes d'**Estimation**
- **Jeux d'Essai**, **Plans** de **Test** et **Procédures** de Test
- **Coordination** avec d'autres projets
- **Documentation**: Utilisateur, Technique
- **Formation** pour les équipes de Développement, pour les Utilisateurs
- Collection des **Métriques**
- Documentation de la Démarche, **Leçons Apprises**



Résumé (4)

Phases

- Pré-Étude
Planification
- Spécifications
- Conception
Technique
- Construction
- Tests
- Installation
- Maintenance

Documents

- SOW
- SPP
- SRS
- SDD
- UDOC
- TDOC
- TPL/TPR
- ISP

Étapes de Validation

- ATP0
- ATP1
- ATP2
- ATP3

Revues

- CDR
- Code Review
- PRR
- PIR

Questions ?



Yossi Gal
Galyotis
IT Business Applications
yossi.gal@galyotis.fr

2b-Risques

Gestion de Projet

Contact:

Yossi Gal, yossi.gal@galyotis.fr, Téléphone: 06 8288-9494



Risk Management Plan

Yossi Gal

Agenda

➤ Introduction

- ✓ L 'Objectif de cette Présentation et de la Gestion des Risques
- ✓ L Environnement dans lequel cette gestion se déroule
- ✓ Le Périmètre d 'utilisation

➤ La Méthodologie

- ✓ SEI/CMM (Software Engineering Institute/Capability Maturity Model)
- ✓ SPI/SPH (Software Process Improvement/ Handbook)
- ✓ RMP (Risk Management Plan)

➤ L'Application en Entreprise

- ✓ L 'Approche utilisée
- ✓ Identification des Risques, Facteurs Contribuants, Solutions
- ✓ La Documentation

➤ Conclusion/Questions

Introduction : L 'objectif de cette Présentation

- **Objectif: Décrire** le plan de la gestion des Risques dans le cadre de petits projets (3 à 4 personnes durant 6 à 12 mois)
- Le **Risk Management Plan** (RMP) est un composant essentiel dans la Méthodologie de gestion de projet. utilisée en entreprise
- Toute la gestion du projet tourne autour du concept **d'identification des risques majeurs**, et des conséquences que leur réalisation peut avoir sur le bon déroulement du projet.
- La gestion des risques fait **partie intégrante de la gestion de projet**. Elle permettra **d'identifier, très tôt** dans le cycle de vie, les **projets voués à l'échec** permettant à l'entreprise de minimiser les investissements à fonds perdus.

L 'objectif de la gestion des Risques

- L'objectif est de faire **Prendre Conscience** au chef du projet et à son équipe, des **Obstacles Potentiels** et l'inviter à réfléchir sur des solutions préventives.
- Le RMP commence par l'évaluation de l'importance du risque et de la **Priorité** avec laquelle il doit être traité,
- Ensuite par l'Identification des **Facteurs** pouvant **Contribuer** à ce que ce risque se réalise.
- Une fois ces éléments identifiés, le projet proposera des **Solutions Préventives** pour réduire l'influence de ces facteurs et faire de telle sorte que **le risque ne se produise pas** ou du moins que ces conséquences aient un **impact minimal** sur le déroulement du projet (**Mitigation Plan**).

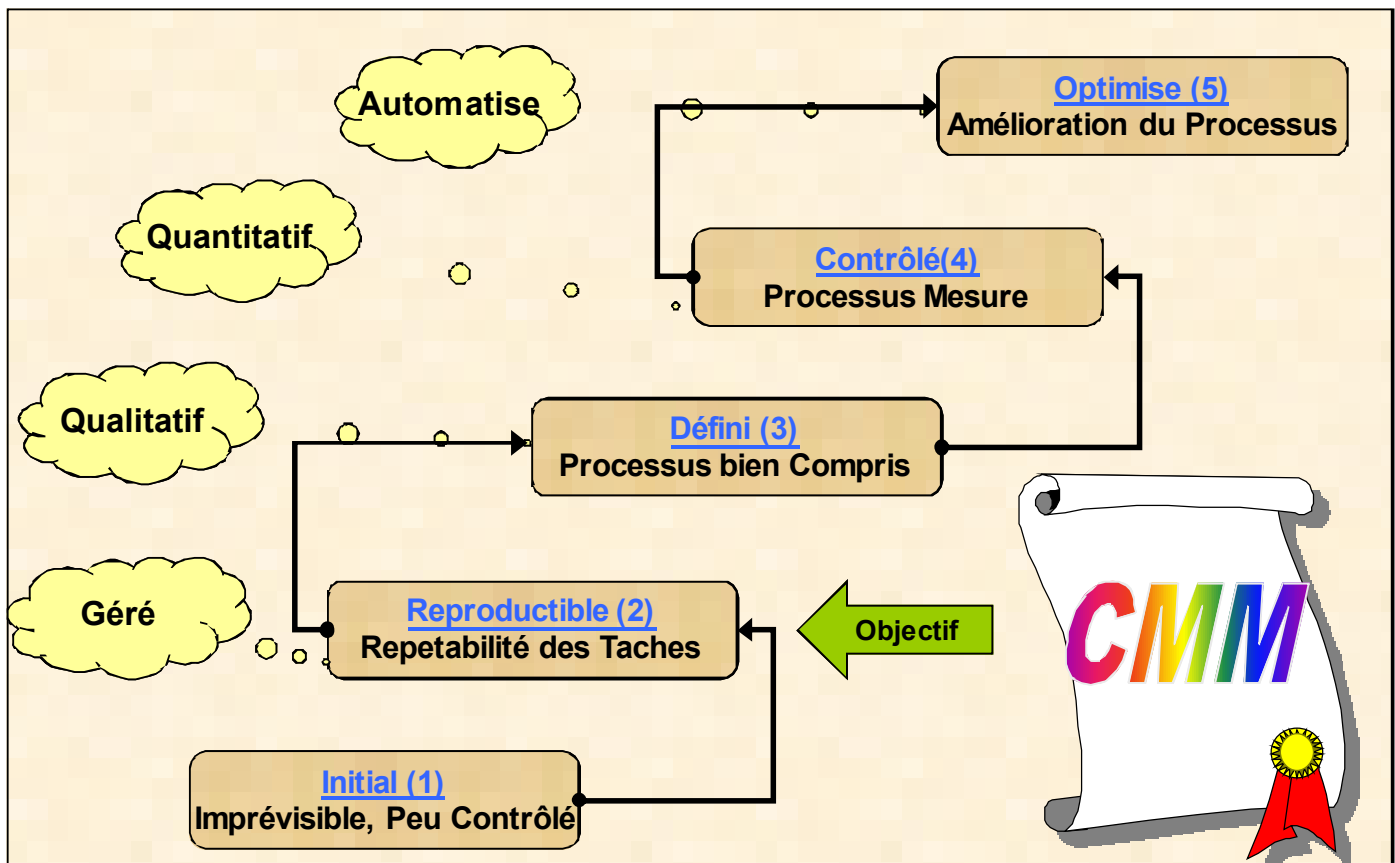
L Environnement de la Gestion des Risques

- Le **Risk Management Plan** est un document de base du '**SPH**' (**Software Process Handbook**) qui sert de référence pour la méthodologie de gestion de projet utilisée en entreprise, le '**SPI**' (**Software Process Improvement**).
- Le SPI est basé sur les concepts développés par l'institut '**SEI**' (**Software Engineering Institute**),
- Le **SEI** est un groupe de recherche à l'université Carnegie Mellon en Pennsylvanie, USA. Il a défini des **méthodes** pour **l'amélioration de la qualité**, et **5 niveaux de maturité** des processus informatiques
- La Méthodologie s'intéresse au **Processus** (**La Démarche**) et non au contenu du Projet (**La Qualité du Produit**).

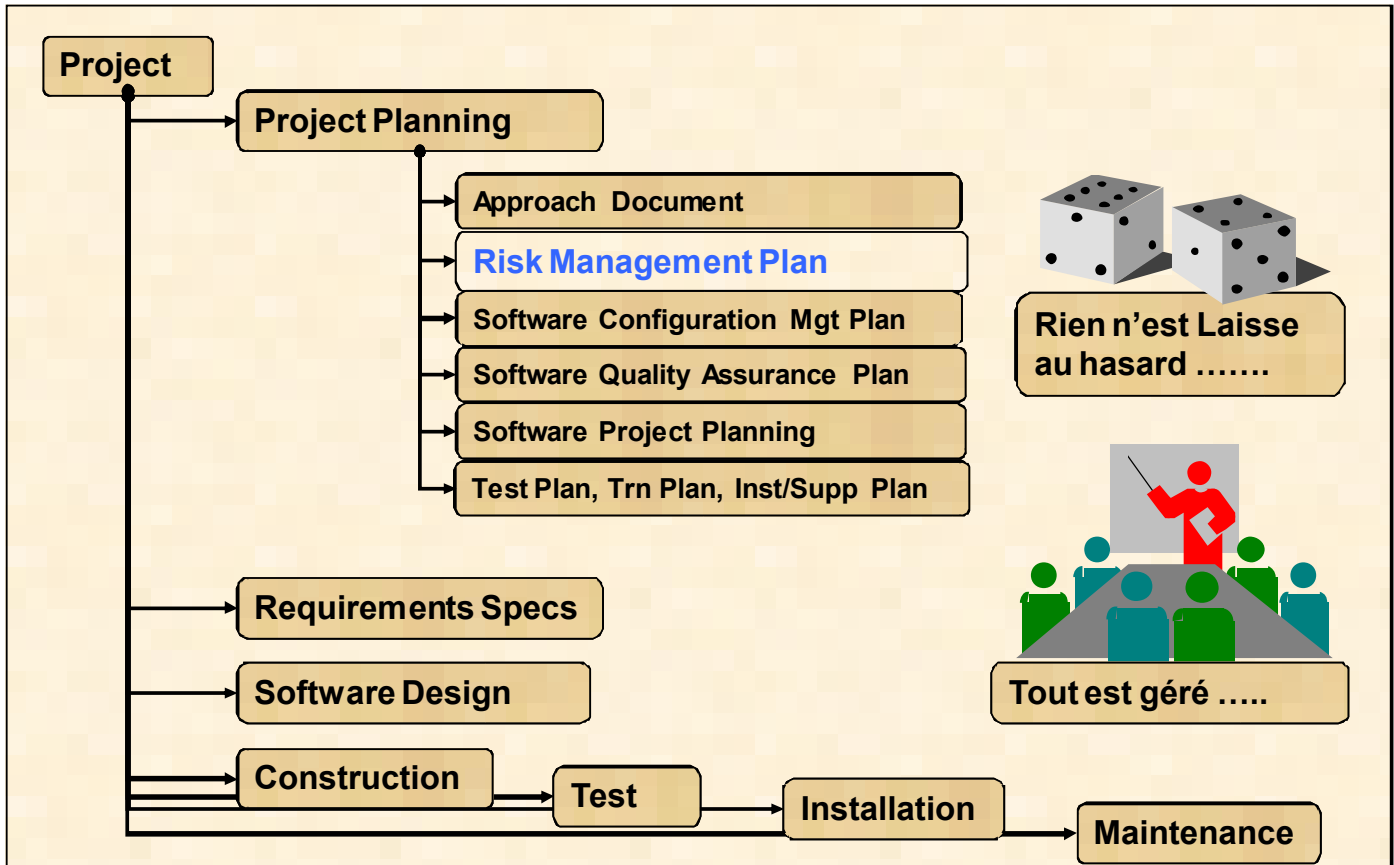
Le Périmètre d 'Utilisation

- Le **RMP** fait partie de la **planification du projet**. Il est initialisé au début du cycle de vie du projet et est **continuellement mis à jour** tout le long de ce cycle.
- Cette présentation mettra l'accent sur une **application pratique dans le cadre de petits projets** (3 à 4 Personnes pendant 6 à 12 mois).
- Elle s'intéressera aux risques **directement liés au projet** lui-même et ne traitera pas des **risques globaux** tel que la '**Disaster Recovery**' ou l'**abandon du projet** par le management
- Très souvent, ces petits projets ne s'occupent pas des risques liés au **matériel** ou à l'**infrastructure**.

La Méthodologie: CMM/5 Niveaux de Maturité



Le RMP dans le cadre du Projet



Le RMP dans le cadre du CMM

- Un **Risque** est tout **événement** entraînant la **possibilité d'échec** du projet quant à la **réalisation des objectifs** pour les
 - ✓ **Délais,**
 - ✓ **Coûts,**
 - ✓ **Fonctionnalités,**
 - ✓ **Qualité,**
 - Débit,
 - Performance,
 - Fiabilité,
 - Disponibilité.
 - ✓ **l'utilisation des ressources** informatiques critiques.
- Le **RMP** est un **ensemble de documents** décrivant les **activités de gestion des risques** se rattachant à un projet.
- Les risques sont **identifiés, évalués, documentés** et **gérés** conformément à une **procédure écrite**.

Le RMP dans le cadre du CMM (suite)

- **Les activités de gestion des risques comprennent :**
 - ✓ l'**Identification Précoce** des **objectifs à risque élevé** du projet;
 - ✓ l'**Identification** des événements pouvant introduire ces risques ou les augmenter (**Facteurs Contribuants**)
 - ✓ le **Suivi des Indicateurs Clés** de ces risques.
- **Les éléments du plan de gestion des risques comprennent**
 - ✓ les **Ressources Nécessaires** (y compris le personnel et les outils);
 - ✓ la **Méthode de Gestion** et la **Liste des Risques** (identification, analyse, évaluation, priorité, planification, suivi et résolution)
 - ✓ le **Calendrier** de gestion des risques, les **Responsabilités**, la méthode et la fréquence de la **Communication**
 - ✓ les **Mesures** collectées et leur analyse.

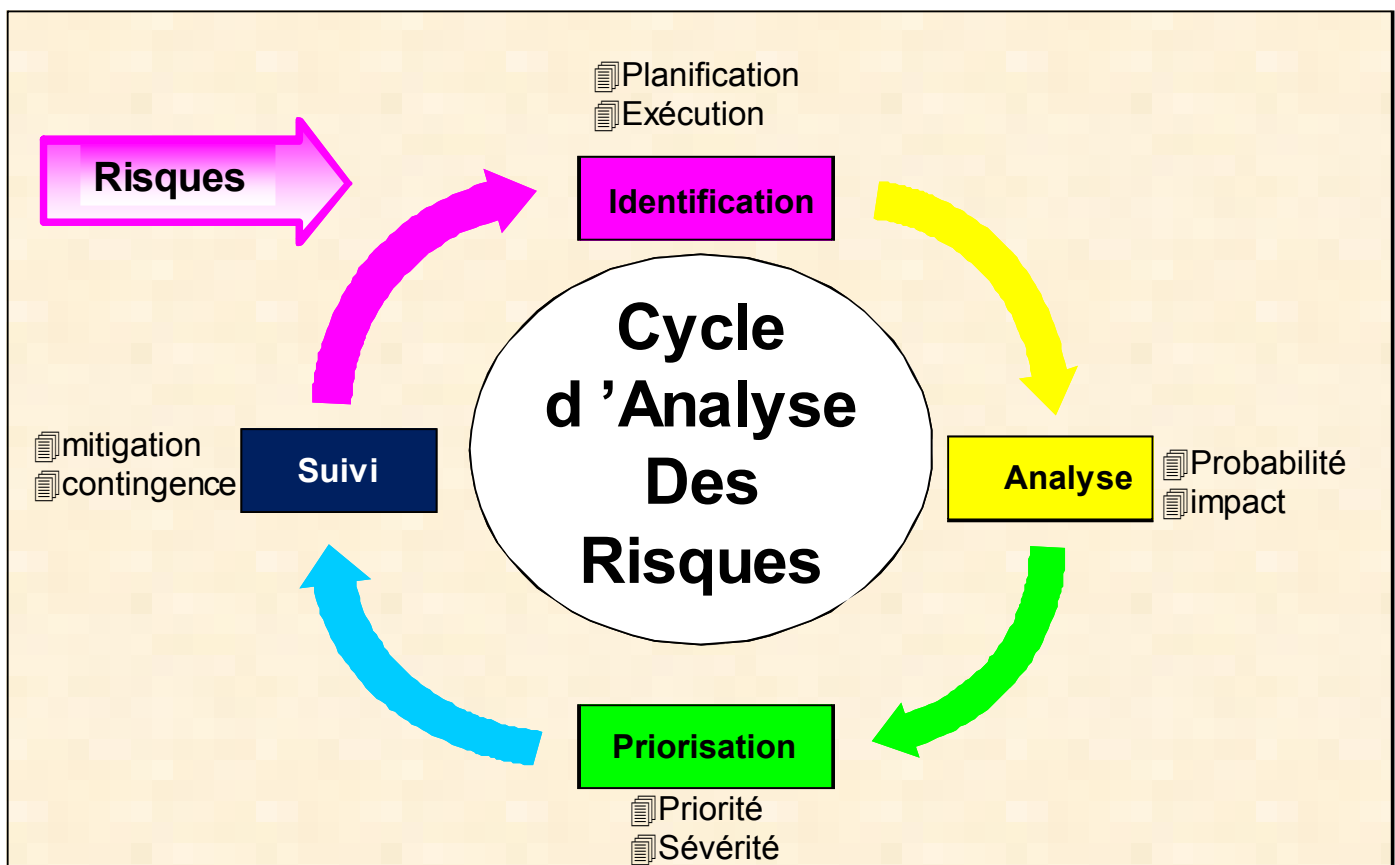
Le RMP dans le cadre du CMM (Suite)

- Le **lancement Initial** du plan de gestion des risques et les **Révisions** importantes à y apporter sont **soumis à des revues**.
- Les risques sont soumis à un **suivi** et à de **Nouvelles Évaluations** pouvant faire l'objet d'une **Re-planification**.
L'information obtenue au cours de ces suivis est utilisée pour **Affiner Les Évaluations**.
- **L'Équipe du Projet et la Direction** reçoivent les **Communications** appropriées sur les **Risques Potentiels**, le **Plans de Gestion** et les **Résultats des Actions Entreprises** pour la **Réduction des Risques** identifiées.
- pour chaque risque identifié, **L'impact Négatif Réel** est constaté par rapport aux **Pertes Estimées**; le nombre et l'ampleur de ces impacts font l'objet d'un **Suivi**.

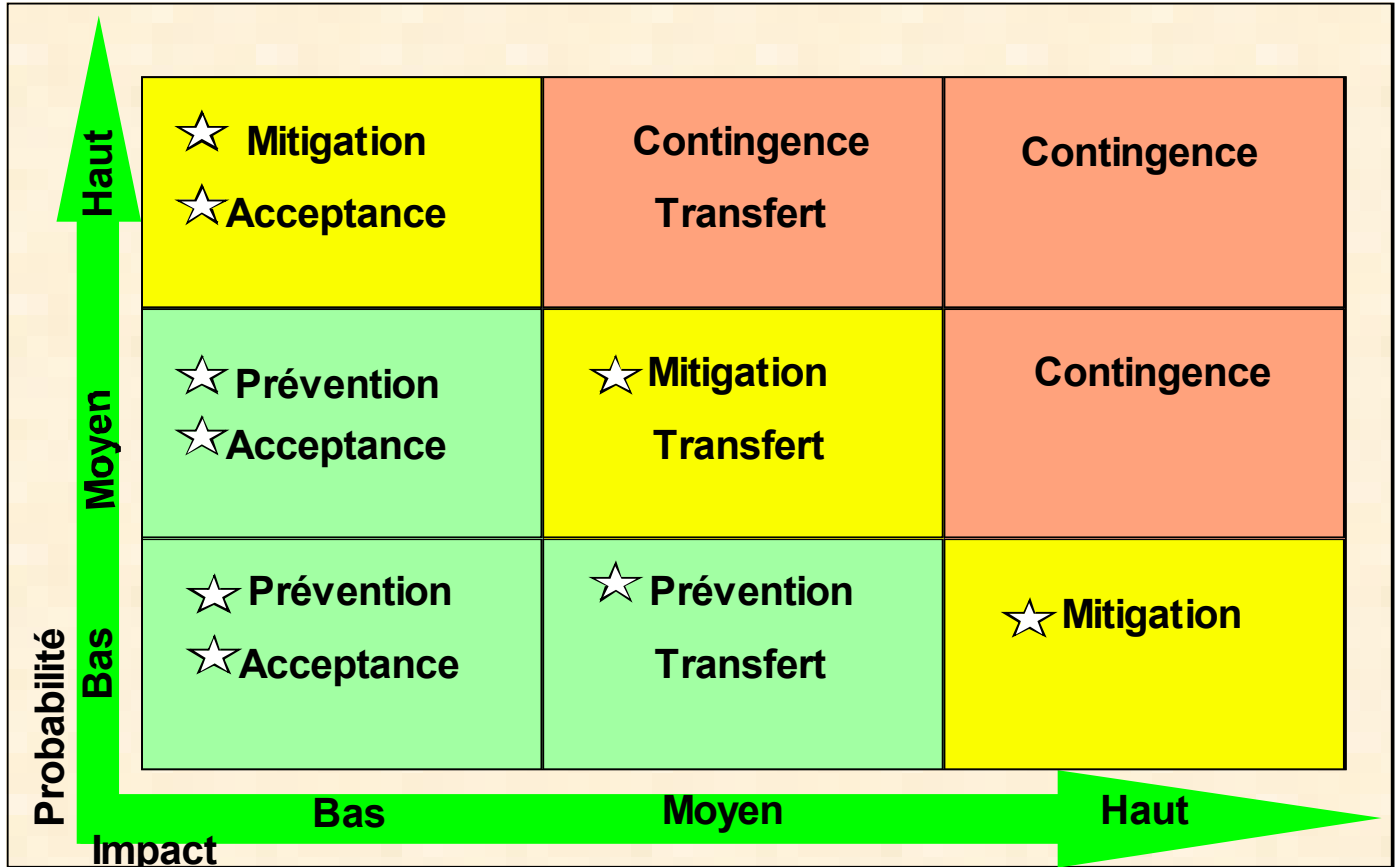
L'Application: Méthodes d'approche

- **Gestion par Événement** : Quel serait l'impact sur le projet si tel ou tel événement se produisait dans l'une des phases du projet
 - ✓ Un Changement important dans les Spécifications
 - ✓ Une Ressource deviendrait Manquante
 - ✓ Un Logiciel n'est pas reçu dans les temps prévus
 - ✓
- **Gestion par Risque/Facteurs Contribuants**: Quels sont les risques majeurs auxquels le projet peut être confronté et quels sont les facteurs qui peuvent contribuer à ces risques
 - ✓ Le risque de ne pas délivrer la totalité des Fonctionnalités spécifiées
 - ✓ Ne pas les délivrer dans les Délais prévus
 - ✓ Le niveau de Qualité spécifiée (débit, performance, disponibilité)
 - ✓ Dans le cadre du Budget alloué
 - ✓

Méthode D 'analyse des Risques



Matrice de catégorisation



Matrice de catégorisation (Définition)

Contingence(Contingency): Établissement d'un plan d'actions (W3 Liste) dans le cas où le risque venait à se réaliser

Mitigation: Revoir les objectifs du projet en ce qui concerne le budget, les délais, les fonctionnalités ou la qualité afin de réduire la probabilité de réalisation du risque

Prévention (Avoidance) : Prendre des actions préventives dès le début du projet dès que le risque est identifié pour faire de tel sorte que le risque ne se produise pas du tout

Transfert (Deflection) : Transférer le risque à une tierce partie (avec contrat)

Acceptance: Comprendre l'impact du risque sur le projet et accepter de vivre avec dans le cas où il venait à se réaliser

La Méthode Utilisée (Petits Projets)

Nous utiliserons la méthode: Risques/Facteurs

- **Identification** des Risques
- Leurs **Priorisation** (L 'ordre dans lequel ils seront traites)
- Le poids de leur **Impact** (Sévérité)
- Pour chaque risque recenser les **Facteurs Contribuants** potentiels
- Pour chaque Facteur Contribuant, Proposer des **Solutions Préventives** pour réduire l 'impact du Risque (**Mitigation Plan**)

Identification des Risques Majeurs

Les risques principaux sont:

- Ne pas délivrer la **Totalité des Fonctions** demandées
- Ne pas délivrer **Une Fonctionnalité Particulière** (f1, f2, f3,)
- Délivrer un Produit **Non Conforme aux Spécifications**
- Ne pas délivrer Le produit dans les **Délais Prévus**
- Ne pas délivrer Le produit avec le **Niveau de Qualité** requis
- Dépasser la **Charge de Travail** Prévue
- Dépasser le **Budget** alloué
- Autres risques spécifiques au projet
 - ✓ **Sécurité**
 - ✓ **Nouvelles Technologies**
 - ✓

Facteurs Contribuants

Quelques exemples de facteurs contribuant:

- Planification trop serrée (pas assez de marge)
- Instabilité des spécifications utilisateur (trop de changements)
- Dépendance vis à vis d'autres projets (attente de résultats ..)
- Expérience de l'équipe (Connaissances Business, Techniques)
- Motivation de l'équipe (Niveau de hiérarchie, Rôle, \$\$\$)
- Conditions matérielles dans lesquelles se déroule le projet
- Difficultés Techniques (Complexité du Projet, Infrastructure)
- Nouvelles Technologies (Serveurs Multi Media, Web, Wireless)
- Manque de Contrôle de Qualité (SQAA, Configuration, Backup)
- Problèmes de performances du produit (temps de réponse, délais des résultats, ..)

Solutions Proposées

Quelques exemples de solutions proposées:

- Produire un Prototype et le faire valider par les utilisateurs
- Mettre Plus de Ressources sur le projet (humaines, matérielles)
- Mettre en place un Coordinateur entre Projets
- Figer les Spécifications, planifier les nouvelles pour la prochaine Version
- Réutiliser du travail fait par d'autres projets (doc/programmes)
- Définir des Procédures Documentées, claires et précises
- Revoir L 'Organisation du Projet, respecter les rôles et les responsabilités.

Solutions Proposées (suite)

Quelques exemples de solutions proposées (suite):

- **Faire Participer** d'une façon plus importante les **Utilisateurs** et autres acteurs du projet
- Mettre en place un plan de **Formation Technique** pour l'équipe
- Mettre en place un plan de **Formation Utilisateurs**
- Mettre en place un **Démarche Qualité** (Configuration, suivi Qualité, Organisation, Backup,..)
- Mettre en place une **Procédure de Testes** avec grand volume de données
- Mettre en place des **Procédures d'inspection** (Revue de documents, Revue de Code, ..)
-

Tableau Récapitulatif

Tableau récapitulatif avec :

- **L'Identifiant** du risque (un numéro de séquence),
- Sa **Description** (Une ligne)
- **Priorité** (Haute, Moyenne, Basse ou un numéro d'ordre),
- **Sévérité** (Haute, Moyenne, Basse ou un numéro relatif),
- Les **Facteurs Contribuants**,
- et les **Solution Proposées** pour réduire le risque (**Mitigation**).
- L' **Actualisation du Risque** (Statut, Date)

Exemple de Tableau récapitulatif

R#	Description	Priorité	Severité	Facteurs	Solutions	Etat
1	Ne pas délivrer le produit.	Haute	Haute	• Mauvaise analyse de la situation	• Fixer des objectifs cohérent qui puissent être atteint	En action
2	Le produit ne répond pas aux besoins des partenaires (échoué)	Haute	Haute	• Pas assez de communication entre les utilisateurs et les développeurs	• Envoyer des mails réguliers et organiser des réunions pour bien comprendre le besoin utilisateur.	En action
3	Rendre le produit souhaité, mais hors délai	Haute	Moyenne	• Pas assez de temps à consacrer au projet	• Fixer des objectifs raisonnables qui puissent être atteint	En action
4	Rendre un produit incomplet.	Haute	Moyenne	• Ne pas avoir les bons produits de développement	• Faire des tests des produits reçus afin de connaître exactement leurs possibilités	En action
5	Ne pas arriver à développer le produit dans le budget planifié.	Moyenne	Moyenne	• Perdre trop de temps dans le développement (bugs, erreurs d'analyses)	• Utiliser des méthodes efficaces pour l'analyse ainsi que pour le développement. (SCMP ...)	En action
6	Rendre un produit qui n'a pas le niveau de qualité souhaité.	Moyenne	Moyenne	• Ne pas respecter les standards de "naming". Ne pas écrire assez de commentaires et documents sur le produit.	• Utiliser le document : • "Software Configuration Management Plan"	En action

État Détaillé des Risques

Un Risque par page avec:

➤ L'Identification du Risque:

- ✓ Identifiant: (un numéro d'ordre)
- ✓ Description: (2 à 3 lignes)
- ✓ Priorité (L'ordre dans lequel il sera traité)
- ✓ Sévérité (L'impact sur le projet si le risque se réalise)

➤ Les facteurs contributants:

- ✓ Facteur 1.1
 - Solution 1.1.1
 - Solution 1.1.2.
- ✓ Facteur 1.2
 - Solution 1.2.1.
 - Solution 1.2.2.

➤ État d'actualisation

Exemple de Risque détaillé

Risk 1 – Requirements

		Notes
Description	Outside requirements. Not to be able to meet user requirements as specified in the SRS.	
Contributing Factor 1	Not enough communication between users and developers.	
Mitigation 1.1	Set up coordinator between users and developers.	
Mitigation 1.2	Spent more time in requirement specification and prototyping.	
Contributing Factors 2	Lack of information – insufficient knowledge or data may be available when needed.	
Mitigation 2.1	Conduct meeting to inform all team members.	
Mitigation 2.2	Train users in the change control process.	

L'État d'Actualisation

L'actualisation des risques:

- **Référence** du risque (Nouvelle référence pour risques non planifiés)
- **État de réalisation** des facteurs contributants
- Les **Solutions effectivement mises en place**
- **Explications des écarts** avec les solution planifiées (si écart important)

Exemple d'Actualisation des Risques

➤ Risque N°1 : Ne pas délivrer le produit

Priorité : Haute --> Moyenne, **Sévérité** : Haute --> Moyenne

Risque toujours existant mais la réalisation de ce dernier est moins forte aujourd'hui car le projet a déjà atteint certains des objectifs fixés.

➤ Risque N°2 : Le produit ne répond pas aux besoins

Priorité : Haute --> Moyenne, **Sévérité** : Haute --> Moyenne

Risque toujours existant avant la présentation du projet aux partenaires.

➤ Risque N°3 : Rendre le produit souhaité, mais hors délais

Priorité : Haute **Sévérité** : Moyenne

Risque réalisé même si le projet a déjà atteint certains des objectifs fixés, prévisions optimistes.

Solution : Affecter plus de ressources car les ressources affectées n'ont pas eu le temps de mener à bout certaines fonctionnalités.

Exemple d'Actualisation des Risques (suite)

➤ Risque N°4 : Rendre un produit incomplet.

Priorité : Haute **Sévérité** : Moyenne

Risque réalisé même si le projet a déjà atteint certains des objectifs fixés, prévisions optimistes.

Solution : Affecter plus de ressources car les ressources affectées n'ont pas eu le temps de mener à bout certaines fonctionnalités.

➤ Risque N°5 : Produire le produit dans le budget planifié

Priorité : Moyenne **Sévérité** : Moyenne

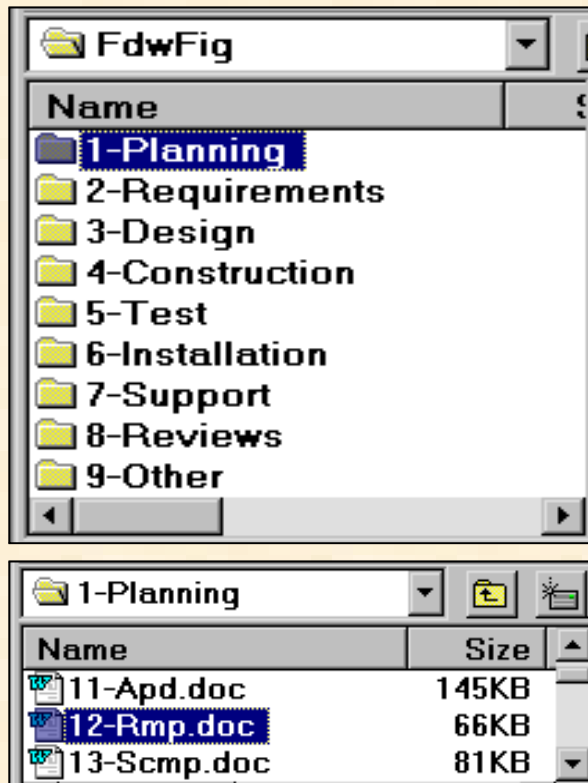
Risque réalisé autant au niveau des ressources financières (coût plus important) comme des ressources « temps disponible ».

➤ Risque N°6 : Rendre un produit sans le niveau de qualité requis

Priorité : Moyenne --> Faible, **Sévérité** : Moyenne --> Faible

Risque toujours existant mais la réalisation de ce dernier est moins forte aujourd'hui, car le projet a déjà atteint certains des objectifs fixés

La Documentation: Structure du Document RMP



0.3. Table of Contents

0.0. IDENTIFICATION	
0.1. DOCUMENT IDENTIFICATION	
0.2. CHANGE HISTORY	
0.3. TABLE OF CONTENTS	
1.0. INTRODUCTION	
1.1. SCOPE	
2.0. ACTIVITIES	
2.1. GLOBAL APPROACH	
2.2. RISKS DESCRIPTION AND MITIGATIONS: SUMMARY	
3.0. RISK DETAILS	
3.1. RISK 1 – REQUIREMENTS	
3.2. RISK 2 – SCHEDULE	
3.3. RISK 3 – DELIVERY (FUNCTIONALIES)	
3.4. RISK 4 – BUDGET	
3.5. RISK 5 – QUALITY LEVEL	
4.0. APPROVALS	
5.0. APPENDIX	

Structure du Document RMP(suite)

Document Identification

Document	RMP.DOC, Risk Management Plan
Project	Financial Data Warehouse/Figures
Project Release	1.1
Document Release	Jun/99
Document Security	TI Internal Data
Created by	Yossi Gal, On : 03/Jul/98

Change History

Who	When	What
Yossi Gal	03/Jul/98	Initialize Document
Laurent Plisson	10/Jul/98	Fill Documents
Yossi Gal	07/09/98	Update plan before Project review
Laurent Manchon	23/Jun/99	Review June Release

Approvals

Action	By	Role	On	Status
Created	Yossi Gal	FDW Project Manager	03/Jul/98	Initialized
Updated :	Laurent Plisson	FDW/RS SCM Manager	10/Jul/98	
Reviewed	Laurent Manchon	FDW/Figures SCM Manager	23/Jun/99	
Approved				

Conclusion

- La Gestion des Risques est un moyen efficace dans **l'amélioration de la gestion de projets** informatiques
- Les Projets sont **gérés et non plus subis**
- Une **Identification précoce** des Risques Potentiels et la préparation de **solutions anticipées** peuvent s'avérer très utiles pour **résoudre des situations de crises** dans le cycle de vie d'un projet.
- C'est un bon **moyen de communication** entre l'équipe du projet et un bon **dispositif d'alerte** pour le management et les utilisateurs.
- Malheureusement, la Gestion des Risques **n'est pas toujours utilisée et surtout pas par les petits projets ...**

Questions



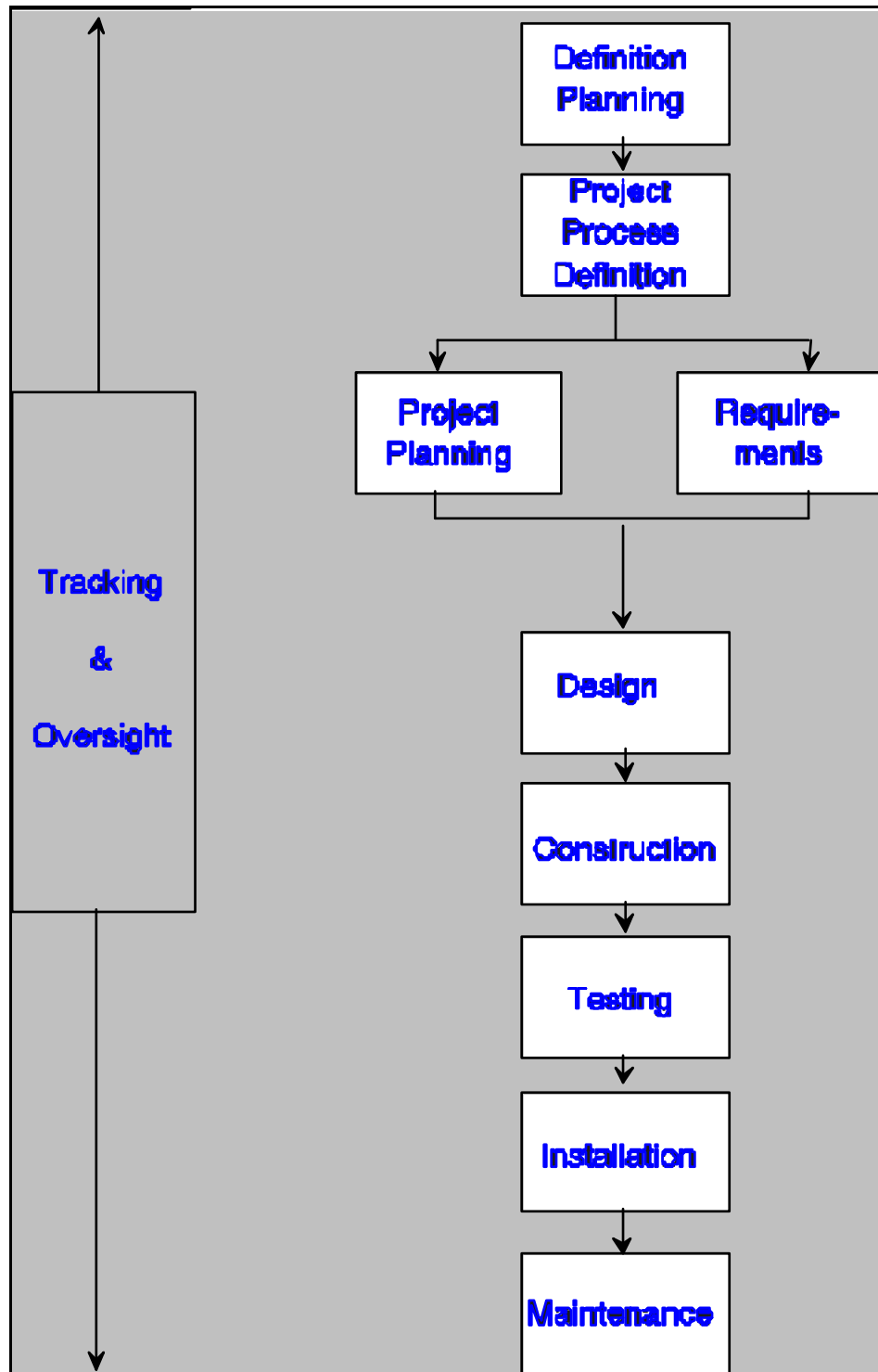
Yossi Gal
Galyotis

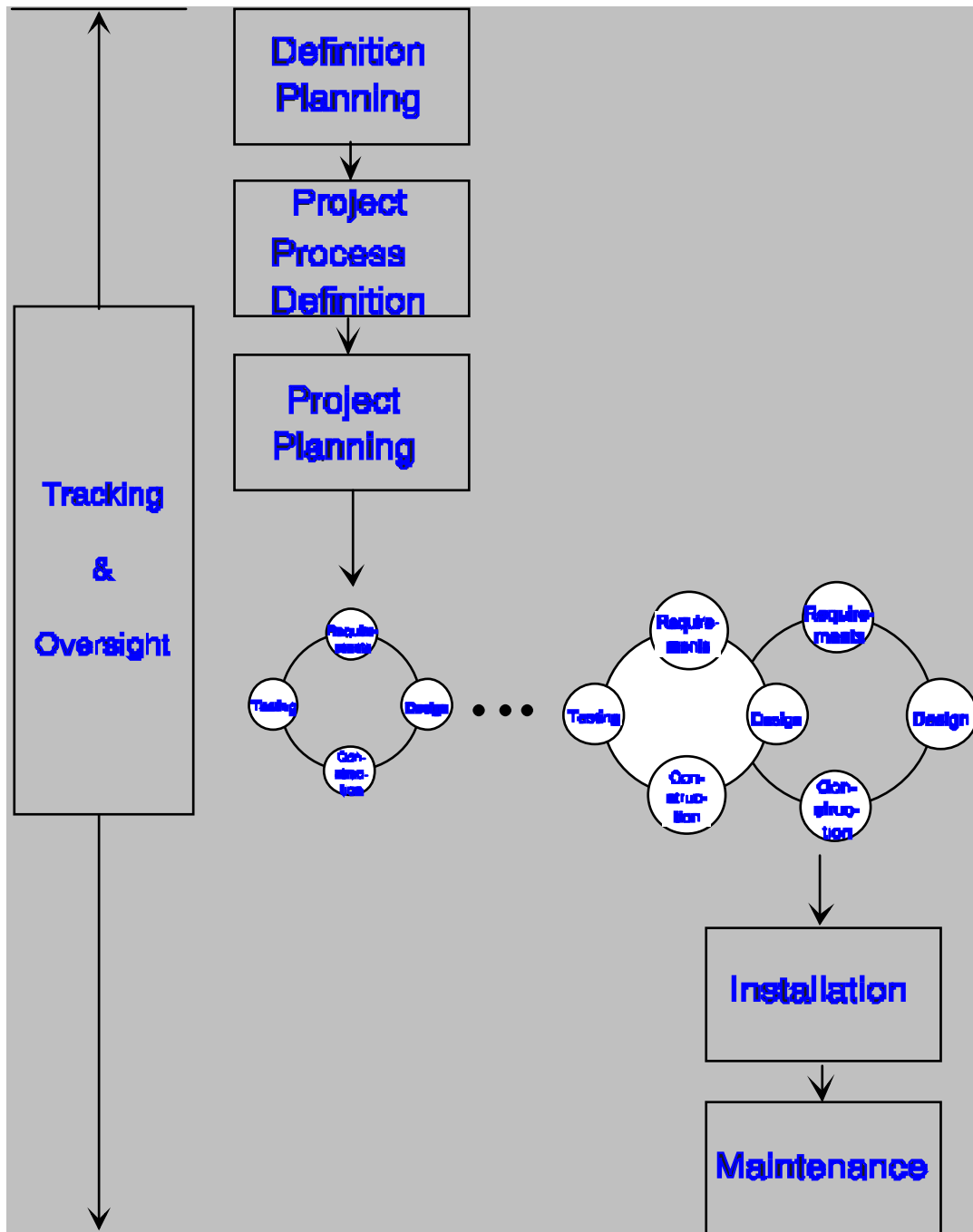
IT Business Applications

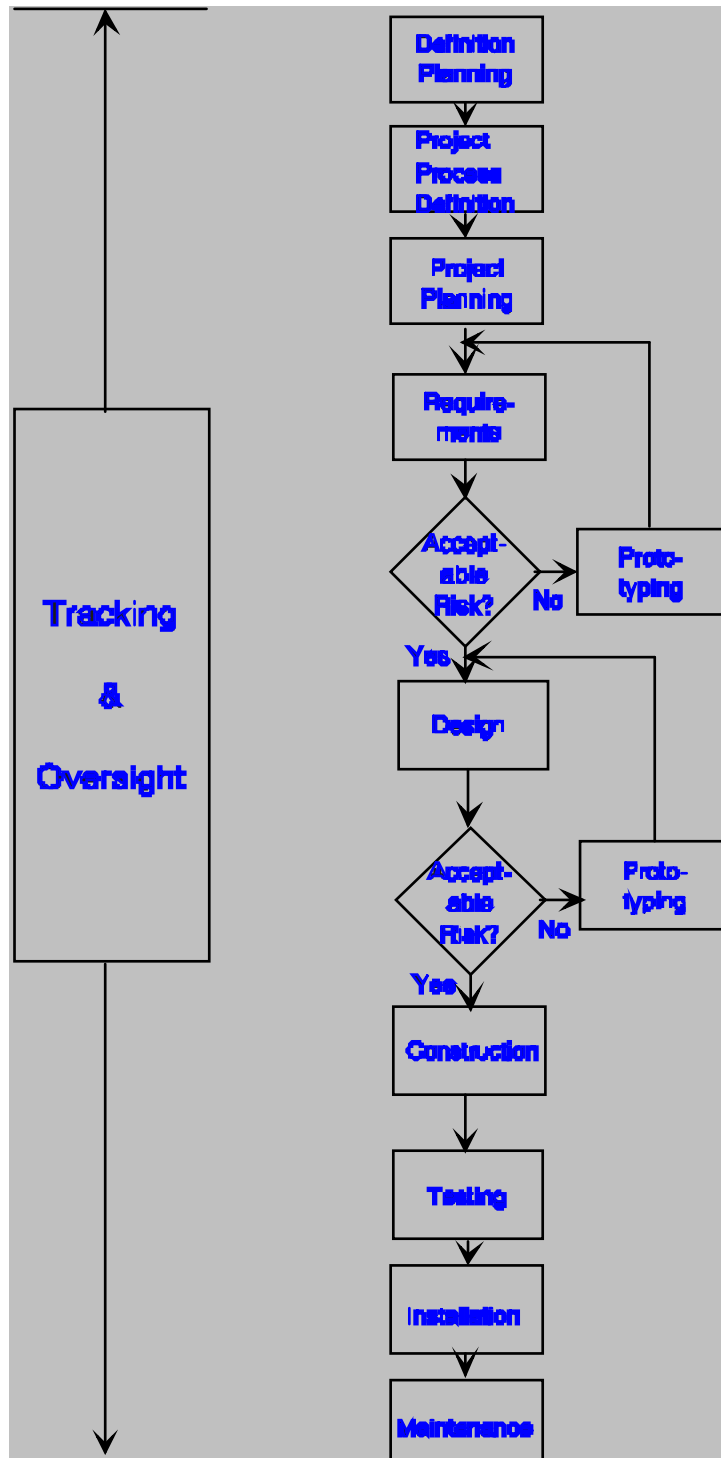
yossi.gal@galyotis.fr

Models

Waterfall Model



Tornado Model

Spiral Model

Tasks**Tasks**

Seq	Task	Responsibility
1	Estimate the Project	PM
2	(Define and) Execute Project Plans (SPP) and Procedures	PM+
3	(Create and) Maintain a Project Notebook	PM SCMA+
4	Analyze Requirements	PT Cust
5	(Define and) Execute Metrics Plan	PM
6	(Define and) Execute Risk Management Plans	PM
7	(Define and) Execute the Test Plan and Test Cases	TT
8	(Define and) Execute SQA Plan and SQA Procedures	SQAA
9	(Define and) Execute SCM Plan and SCM Procedures	SCMA
10	Establish Configuration Identification Procedures	SCMA
11	(Establish and) Control the Baseline Repository	SCMA
12	Conduct Management Reviews	PM
13	Conduct Peer Reviews	PT
14	(Define and) Execute Installation and Support Plans	PM
15	Define the Objectives and Customer Needs (SOW)	Cust PM+
16	Identify Customer Champion(s)	Cust
17	Identify Project Planner and Project Manager	Mgmt
18	Determine Definition Planning Resource Needs and Obtain Commitment (ATP with Project Approach (ATP0) Review)	PM+ Cust
19	Analyze Preliminary Requirements and Alternatives	Cust PM+
20	Identify Risks	PT
21	Analyze Requirements	PT Cust
22	Prepare Initial Breakdown of Work (WBS) and Estimates (size, effort, schedule, project cost, software operations cost)	PM+
23	Estimate the Project	PM
24	Document Preliminary Plans (Approach Document)	PM+
25	Review Preliminary Plans and Obtain Commitment (ATP with Project Planning (ATP1) Review)	Mgmt PM+ Cust
26	Establish Project Team and Interfacing Groups	PM+
27	Establish a baseline of the estimates, SOW, Approach Document, Communication Map	SCMA
28	Capture staffing, milestones, cost, and requirements stability metrics	PM+
29	Characterize the Project	PT
30	Tailor the key activities to meet intent	PT SQAA
31	Define the Project Software Process	PT SQAA
32	Define (and Execute) Project Plans (SPP) and Procedures	PM+
33	Create (and Maintain) a Project Notebook	PM+ SCMA
34	Analyze Requirements	PT Cust
35	Define (and Execute) Metrics Plan	PM

Seq	Task	Responsibility
36	Define and Execute Risk Management Plans	PM
37	Communicate project plans to team	PM
38	Define (and Execute) Test Plan (and Test Cases)	TT
39	Define the Testing Environment	PM
40	Define and Execute SQA Plan and SQA Procedures	SQAA
41	Define and Execute SCM Plan and SCM Procedures	SCMA
42	Establish Configuration Identification Procedures	SCMA
43	Establish a Configuration Control Board (CCB)	PM
44	Establish Change Request Procedures	SCMA
45	Establish and Control the Baseline Repository	SCMA
46	Set up regularly scheduled communication meetings with the customer(s)	PM
47	Define Peer Review Plans	PM
48	Conduct Peer Reviews	PT
49	Establish a baseline of the SPP, Test Plan, Metrics Plan, Risk Management Plan, SQA Plan, and SCM Plan	PM+
50	Capture software size, peer review, staffing, milestones, cost, and requirements stability metrics	PM+
51	Analyze Requirements	PT Cust
52	Identify user documentation, testing, installation and support requirements in requirements analysis	PM+
53	Document Training Requirements and Approach (TAD)	PM
54	Represent Requirements (SRS)	PT
55	Define (and Execute Test Plan and) Test Cases	TT
56	Review Requirements (Conduct SRR)	PM Cust+
57	Establish a baseline of the SRS	SCMA
58	Conduct Management Reviews (Authorization to Proceed with Project Execution (ATP2) Review)	PM+ Cust
59	Establish a baseline of the SRS, Requirements Trace Map, test cases	PM+
60	Capture software size, peer review, staffing, milestones, cost, and requirements stability metrics	PM+
61	Develop a Model	PT
62	Demonstrate the Model	PT
63	Analyze and Review Preliminary Design (PDR)	PM
64	Define the Design Details (SDD)	PT
65	Define Training Design Document (TDD)	TRT
66	Define preliminary User Documentation	TRT
67	Define preliminary Installation and Support Plan	PM
68	Review the Design Details (CDR)	PT
69	Conduct Management Reviews (Authorization to Proceed with Project Execution (ATP2) Review)	PM+ Cust
70	Establish a baseline of the SDD, TDD and test cases	SCMA
71	Capture detailed schedule progress, requirements implemented, software size, peer review, staffing, milestones, cost, and requirements stability metrics	PM+
72	Develop and Review Code	PT
73	Define Test Procedures	PT
74	Establish a baseline for unit testing	PT SCMA

Seq	Task	Responsibility
75	Define and Document User Documentation	PM
76	Create Training Material	TRT
77	Conduct Unit Testing	PT
78	Conduct Regression Testing	PT
79	Define (and Execute) Installation and Support Plans	PM
80	Establish a baseline for integration testing	PT SCMA
81	Establish a baseline of the user documentation and Installation and Support Plan	SCMA
82	Capture detailed schedule progress, requirements implemented, software size, peer review, staffing, milestones, cost, and requirements stability metrics	PM+
83	Conduct Integration Testing	PT
84	Conduct Regression Testing	PT
85	Establish a baseline for system and acceptance testing	PT SCMA
86	Establish a baseline of the Training Materials	SCMA
87	Determine Test Readiness (TRR)	PM
88	Conduct System Testing	TT+
89	Conduct Regression Testing	PT
90	Conduct Acceptance Testing	TT Cust+
91	Document Testing Results (TRP)	TT
92	Establish a baseline of the TRP	SCMA
93	Capture test error, detailed schedule progress, requirements implemented, software size, peer review, staffing, milestones, cost, and requirements stability metrics	PM+ TT
94	Complete Pre-installation Checklist	SQAA
95	Conduct SCM Configuration Validations	SCMA
96	Determine Production Release Readiness (PRR)	PM Cust
97	Conduct Management Reviews (Authorization to Proceed with Project Installation (ATP3) Review)	PM+ Cust
98	Install the Product	PM
99	Conduct Training	TRT
100	(Define and) Execute Installation and Support Plan	PM
101	Review Lessons Learned (PIR)	PM
102	Establish a baseline of the TRP	SCMA
103	Capture test error, detailed schedule progress, requirements implemented, software size, peer review, staffing, milestones, cost, and requirements stability metrics	PM+ TT
104	(Define and) Execute Installation and Support Plan	PM
105	Define and Execute the Maintenance Process	PM

Acronyms

Acronym	Signification
AT	Acceptance Testing
ATP	Authorization to Proceed
ATP0	Authorization to Proceed with Project Approach Review
ATP1	Authorization to Proceed with Project Planning Review
ATP2	Authorization to Proceed with Project Execution Review
ATP3	Authorization to Proceed with Project Installation Review
CCB	Configuration Control Board
CDR	Critical Design Review
CMM	Capability Maturity Model
INP	Installation and Support Plan
IS&S	Information Systems and Services (Today IT)
IT	Integration Testing (Also Information Technology)
KLOC	Thousand Lines of Code
PDR	Preliminary Design Review
PIR	Post Installation Review
PLC	Product Life Cycle
PRR	Production Readiness Review
SCM	Software Configuration Management
SCMA	Software Configuration Management Analyst
SDD	Software Design Document
SEI	Software Engineering Institute
SPH	Software Process Handbook
SOW	Statement of Work
SPP	Software Project Plan
SQA	Software Quality Assurance
SQAA	Software Quality Assurance Analyst
SRR	Software Requirements Review
SRS	Software Requirements Specification
ST	System Testing
STEP	Software Engineering Process
TAD	Training Analysis Document
TBD	To Be Determined
TDD	Training Design Document
TPL	Test Plan
TPR	Test Procedure
TRP	Testing Report
TRR	Test Readiness Review
USD	User Documentation
UT	Unit Testing
WBS	Work Breakdown Structure

Glossary

Term	Explanation
Acceptance Testing (AT)	Formal testing conducted by the customer to determine whether or not a product meets the requirements imposed on it and to determine its acceptability.
Activity Table	Tables listed in Appendix D: Activity Table, that lists the activities and the primary responsible person.
Alternative Analysis	A process for generating, evaluating, and selecting the optimal or best combination or selection of reuse, buy, and build techniques to achieve a customer's need.
Approach Document	A document that discusses the analysis that was performed on the selected approach for the project and information on necessary resources and procedures needed to perform the chosen alternative. It includes or references a Statement of Work and a Work Breakdown Structure.
Authorization to Proceed (ATP)	Designated points in the life cycle of a project where decisions are made as to whether the project should continue.
Authorization to Proceed with Project Approach (ATP0) Review	Conducted after the Statement of Work to secure funding for alternative analysis and the approach definition.
Authorization To Proceed with Project Planning (ATP1) Review	Conducted after the alternative analysis and approach definition to secure funding for project development planning, requirements analysis, and preliminary design (before writing the Software Project Plan).
Authorization To Proceed with Project Execution (ATP2) Review	Conducted during project execution to secure approval to continue with the project. This review can be conducted multiple times.
Authorization to Proceed with Project Installation (ATP3) Review	Conducted after completion of design, coding, and testing to secure approval to install and support the product.
Baseline	A. The official source for software work products (e.g., code, supporting documents, technical documentation, and user documentation) to which only authorized changes or additions are made. Baselines are maintained by the Software Configuration Management analyst. B. A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. [IEEE]
Change Request	A. A request submitted to change baselined software and documentation.
Configuration Control Board	A centralized committee authorized to control changes to baselined items (software and documentation) for a project or set of projects.
Configuration Identification	A. Unique labeling of configuration elements that are tracked in software work products. B. Identifies the elements to be included in a development configuration or baseline. [STEP]
Construction	Stage of the life cycle that includes producing and reviewing the code and data.

Term	Explanation
Critical Design Review (CDR)	A. A meeting to evaluate the detailed design, performance, and test characteristics of the design solution, to evaluate the adequacy of the preliminary user documentation and the Installation and Support Plan, and to determine readiness for construction activities.
	B. (1) A review conducted to verify that the detailed design of one or more configuration items satisfies specified requirements; to establish the compatibility among the configuration items and other items of equipment, facilities, software, and personnel; to assess risk areas for each configuration item; and, as applicable, to assess the results of producibility analyses, review preliminary hardware product specifications, evaluate preliminary test planning, and evaluate the adequacy of preliminary operation and support documents. (2) A review as in (1) of any hardware or software component. [IEEE]
Cycle Time	A. The time to complete a cycle. B. The time interval between the start of one cycle and the start of the next cycle; the interval of time during which a sequence of events is completed.
Design	A. A stage of the software life cycle in which the goal is to produce a design that will satisfy the requirements specified in the SRS and to describe Test Plan that will verify that the code meets the design.
	B. (1) The process of defining the architecture, components, interfaces, and other characteristics of a system or component. (2) The result of the process in (1). [IEEE]
Development	The process of provisioning a new software product using one or a combination of the following methods: software reuse, supplier evaluation and purchase (buy), or software development (build).
Document Baseline	An approved, uniquely identified and numbered version from which a project must track all future changes.
Documentation	A. A collection of documents on a given subject. B. Any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results. [IEEE]
Implementation	A. The process of translating a design into hardware components, software components, or both. [IEEE] B. The result of the process in (A). [IEEE]
Installation	A. The stage of the software life cycle where the key activities are to conduct a PRR, establish a production baseline, install the product, hold a PIR, and train the users.
Installation and Support Plan (INP)	A plan that describes the criteria necessary to install and support a product. It includes maintenance and diagnostic tools, installation information, conversion information, system modification procedures, and disaster recovery information.
Integration	A. An orderly progression of combining software components (modules and subsystems) and software work products (documentation) until the entire system product has been built.
	B. The process of combining software components, hardware components, or both into an overall system. [IEEE]
Integration Testing (IT)	Testing usually performed by the software project team as the modules are linked together to ensure error free interaction.

Term	Explanation
Intergroup Coordination	An established means for a software project team to participate actively with the customer and other software project teams in order to satisfy the customer's needs effectively.
Key Activity	An activity that is a basic building block in the software development and maintenance process as described in the Software Process Handbook.
Lines of Code (LOC)	A measure of program size for a software project. It is a physical line on an input screen minus comments and blank lines. LOC counted includes executable lines, data definitions and JCL, new LOC, reused LOC, and modified LOC. When estimating, LOC must be recorded by language type and must comprehend all code to be developed. See Source for a definition of reused.
Maintenance	A. The stage of the software life cycle that focuses on addressing changes and additions to existing production systems. B. The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment. [IEEE]
Method	A reasonably complete set of rules that establishes a precise and repeatable way of performing a task and arriving at a desired result. [CMM]
Milestone	A. A key, measurable event that occurs during a software project life cycle release. It is specified as a calendar date. A milestone may be established by a customer or by the software project itself. A milestone represents a commitment to the achievement of the key event and therefore must be approved and included in the SPP and other appropriate project scheduling and status documentation. Revisions to planned milestones must be approved and also maintained in appropriate project documentation.
Mitigation Peer Review	See Risk Mitigation. A review of a software product following defined procedures by peers of the producers of the product for the purpose of identifying defects and improvements. [CMM]
Post-Installation Review (PIR)	A meeting held after production installation and release to review and analyze the major successes and problems of the project, to evaluate the metrics collected, and to identify action items for future improvement.
Preliminary Design Review (PDR)	A. A meeting to evaluate the progress, technical adequacy, consistency, and risk resolution of the top-level design and testing approach.
Procedure	A written description of a course of action to be taken to perform a given task. [IEEE] and [CMM]
Process	A sequence of steps performed for a given purpose; for example, the software development process. [IEEE]
Product	A. The software, hardware, documentation, and training associated together as a single entity. A product may consist of one system or several systems.

Term	Explanation
Production Readiness Review (PRR)	A meeting held before production installation and release to review the outcome of the Pre-Installation Checklist, action items from testing, updated project documentation, training plan and installation plans, and the Configuration Identification List of the product.
	C. To design, write, and test computer programs.
Program Manager	The individual designated as responsible for several projects feeding into a single project.
Project	A. A system or a group of systems. Typically, a software grouping designated by the project manager.
Project Manager	A. The individual designated as responsible for the software development from the start of the initial planning activities through the software development life cycle. B. Any manager, at a project or organizational level, who has direct responsibility for software development or maintenance.
Project Notebook	A record of important documents, events, and communication that occur as the project evolves.
Project Planning	The effort in the software life cycle that documents early planning definition and development tasks to help allocate resources to prepare the project for the later stages of the life cycle.
Project Software Process	A process that is used by a project to develop and maintain software. This process is the result of characterizing the project, mapping the key activities to a life cycle model, and creating procedures to meet the intent of the key activities.
Project Team	All personnel working on the project and involved in the planning, requirements definition, design, coding, test, documentation, SCM, and SQA. Includes junior and senior level programmers, analysts (systems, business), systems engineers, key technical contributors, technicians, support personnel (e.g., reference team for XXX project), technical writers, and management personnel.
Prototyping	A. A procedure that augments the software engineering process by helping to define and analyze requirements or design, reduce development time, and reduce design risk. B. A hardware and software development technique in which a preliminary version of part or all of the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process. [IEEE]
Rapid Application Development	A technique for applying the Tornado Model for software development (build). It is characterized by high customer involvement, group development, and frequent feedback.
Regression Testing	A. The reuse of selected test cases or test procedures to verify that modifications to one part of a system did not cause unintended side effects to other parts. B. The selective re-testing of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements. [IEEE]
Release	A. The installation, distribution, and delivery of approved, baselined software or documentation.

Term	Explanation
Requirement	A. The detailed items within an approved SRS that identify "what" a product is to do. Areas of requirements include external interface, user interface, security, functional, hardware interface, operations, and others. A requirement is typically identified by paragraph number or statements containing the verb "shall" (will or must) within the SRS.
	B. (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2). [IEEE]
Requirements Analysis	A. The stage of the software life cycle where preliminary requirements are refined, analyzed, and restated using precise language and quantifiable terms. B. In software engineering, the process of studying user needs to arrive at a definition of software requirements. Sometimes synonymous with analysis, software analysis, or software requirements engineering.
Requirements Trace Map	A technique to provide evidence that each requirement is addressed in design and test documents.
Resources	The term used to describe labor, material, and other effort needed to accomplish identified tasks.
Reusability	The characteristic of software or documentation that allows it to be used multiple times (i.e., can serve a common purpose again and again).
Revision	A modification to a configuration element.
Risk	An event, element, or factor that can negatively affect a process or product and that possesses both a possibility of occurring and a resulting effect.
Risk Analysis	A. Produces assessments of the severity (technical, cost, schedule, organizational, subcontractor, or quality) associated with each of the identified risk items and the constituent risk factors comprising each risk item. B. In system and software engineering, the methodical process of identifying areas of potential risk, the associated probability of occurrence, and the seriousness of the consequences of the occurrence. [STEP]
Risk Contingency	A plan, task, or set of conditions that can be initiated and followed in the event a predetermined risk reaches a certain effect level.
Risk Identification	Produces lists of the project specific risk items likely to compromise a project's satisfactory outcome. [Boehm]
Risk Management Plan	A plan that describes the aspects of risk identification, risk assessment, risk reduction, and risk management to be performed.
Risk Mitigation	Identifies actions and alternatives to eliminate or reduce risk items before they become either threats to successful software operation or sources of software rework.
Risk Monitoring	Establishes mitigation and contingency plans for risk items, monitors progress toward resolving each risk item, and reports risk mitigation status.

Term	Explanation
Risk Prioritization	Produces a prioritized ordering of the risk items identified and analyzed. [Boehm]
Schedule	A procedural plan that indicates the start, completion, and sequence of a series of tasks, events, or elements.
Senior Management	A manager(s) at a high enough level that his or her primary focus is expected to be the long-term vitality of the company and organization, rather than short-term project and contractual concerns or presence. In general, senior management for engineering would have responsibility for multiple projects. [CMM]
Software Configuration Management (SCM)	The process of identifying and defining the configuration elements in a system, controlling the release and change of these elements throughout the system life cycle, recording and reporting the status of configuration elements and change requests, and verifying the completeness and correctness of configuration baselines.
Software Configuration Management Analyst	The person primarily responsible for implementing configuration management procedures and activities.
Software Configuration Management Baseline Audit	A configuration validation activity that is conducted to validate the integrity, completeness and correctness of baseline contents.
Software Configuration Management Plan	The document that defines how SCM activities will be executed on a specific project. It includes descriptions of the SCM analyst's tasks and SCM procedures.
Software Design Document (SDD)	A. A document that describes the detailed design description of interfaces, software products, and unit and integration testing.
Software Engineering Process Group (SEPG)	A group of specialists who facilitate the definition and improvement of the software process used by the organization.
Software Integration	A. The process of combining two or more applications (possibly including supplier software) to create a unique software product. B. A process of putting together selected software components to provide the set or specified subset of the capabilities the final software system will provide.
Software Life Cycle	A. Software development and maintenance activities that include project planning, analysis, design, construction, test, installation, and support.
	B. The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. [IEEE]
Software Maintenance	See Maintenance.
Software Product	The complete set or any of the individual items of the set, of computer programs, procedures, associated documentation, and data designated for delivery to a customer or end user. See Software Work Product for a contrast.

Term	Explanation
Software Project	A. The activities in producing new software systems or subsystems or enhancement or modification of existing software systems or subsystems. B. An undertaking requiring concerted effort that is focused on analyzing, specifying, designing, developing, testing, or maintaining the software components and associated documentation of a system. A software project may be part of a project building a hardware and software system.
Software Project Plan (SPP)	A. A concise, detailed document describing the specific aspects of project development, management, control, and maintenance. B. The collection of plans that describes the activities to be performed for the software project. It governs the management of the activities performed by the software project team for a software project. It is not limited to the scope of any planning standard.
Software Provisioning	A process for providing software to meet business objectives, including reuse, buy, and build (development).
Software Quality Assurance (SQA)	A. A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements. [IEEE] B. A set of activities designed to evaluate the process by which an item or product is developed or maintained. [IEEE]
Software Quality Assurance Analyst	A person who is primarily responsible for implementing software quality assurance procedures and activities.
Software Quality Assurance Manager	A manager who has responsibility for overseeing the software quality assurance function.
Software Quality Assurance Plan	A document that defines how SQA activities will be executed on a project. It includes descriptions of the SQA analyst involvement in the project's software process activities and describes details of SQA procedures.
Software Requirements Review (SRR)	A. Meeting(s) to review the requirements of a product as specified in the SRR and establish project and customer approval. B. (1) A review of the requirements specified for one or more software configuration items to evaluate their responsiveness to and interpretation of the system requirements and to determine whether they form a satisfactory basis for proceeding into preliminary design of the configuration items. (2) A review as in (1) for any software component. [IEEE]
Software Requirements Specification (SRS)	A. A document that consists of a specified set of detailed requirements that satisfies the customer's needs and can be met with a reasonable design suitable for independent development, testing, and acceptance. B. A document that specifies the requirements for a system or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.
Software Work Product	Any item created as part of defining, maintaining, or using a software process, including process descriptions, plans, procedures, computer programs, and associated documentation, that may or may not be intended for delivery to a customer or end user. See Software Product for a contrast.

Term	Explanation
Source	A. LOC (traditional), OBJECTS (XXX), or function point.
Spiral Model	A. A model of the software development process in which the constituent activities, typically requirements analysis, preliminary and detailed design, coding, integration, and testing, are performed iteratively until the software is complete. [IEEE] B. The primary purpose of this is to mitigate risk in each stage before proceeding to subsequent stages.
Statement of Work (SOW)	A preliminary list of deliverables and requirements written from the customer's perspective that defines what will and will not be done.
Subcontractor Software	Software written under contract with an external organization.
System Testing (ST)	A. Testing performed by the testing team to show that the product meets the requirements imposed on it by the requirements specification.
Tailoring	Modifying a process, standard, or procedure to meet customer needs and business objectives.
Testing Activity	A. The stage of the software life cycle that verifies that the product meets the requirements as stated in the SRS. It includes unit, integration, system, and acceptance testing. B. The period of time in the software life cycle during which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied.
Test Case	A. A set of test data or a testing scenario to be used for a particular test objective.
Test Procedure (TPR)	A. Specifies the steps necessary to set up, execute, record, and measure a test case or a set of test cases. B. (1) Detailed instructions for the set up, execution, and evaluation of results for a given test case. (2) A document containing a set of associated instructions as in (1). (3) Documentation specifying a sequence of actions for the execution of a test. [IEEE]
Test Readiness Review (TRR)	Management and technical review of the product readiness for full system and acceptance testing.
Test Repeatability	An attribute of a test indicating that the same functional results are produced each time the test is conducted and that variations in performance response times are within acceptable tolerances.
Test Report (TRP)	A. A summary of the test results. It identifies where testing deviated from the Test Plan, test cases, and test procedures. It also includes the disposition of the release based on testing results. B. A document that describes the conduct and results of the testing carried out for a system or component. [IEEE]
Testing Team	A team responsible for conducting system and acceptance testing.
Test Baseline	Controlled source of all code and applicable documentation to be used during testing.
Test Plan (TPL)	A. A document that identifies and plans a systematic and consistent approach for testing specific software and documentation.

Term	Explanation
	B. (1) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, and any risks requiring contingency planning. (2) A document that describes the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, responsibilities, schedules, and required resources for the testing activity. [IEEE]
Tornado Model	A model of the software development process that supplements the Spiral Model with milestones and goals. It consists of one or more series of requirements, design, construction, and test until the software is complete.
Traceability	The characteristics of software and documentation that ensure requirements have been thoroughly addressed in design and testing activities. The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another.
Training Analysis Document (TAD)	A document that defines the training requirements, target audience, course outlines, and a schedule for the training design.
Training Design Document (TDD)	A document that describes the design necessary to satisfy the training needs that are analyzed in the Training Analysis Document.
Training Plan	A. A section included in the SPP that outlines the staff training needs for a given plan. B. The documented and approved plan for providing the resources and schedule to implement the training program. Training plans may address the organization's training needs or apply to a single individual.
Unit Testing (UT)	A. Testing usually performed by the developer(s) on software units. B. Testing of individual hardware or software units or groups of related units. [IEEE]
User Documentation (USD)	A. The document(s) that describes software requirements from the user's perspective and shows a step-by-step process to operate the system with examples of expected input and output. B. Documentation describing the way in which a system or component is to be used to obtain the desired results. [IEEE]
Version	An initial or subsequent release of a computer software configuration item or document associated with a complete compilation of the item.
Walkthrough	A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems. [IEEE]
Waterfall Model	A model of the software development process in which the constituent activities, typically a concept phase, requirements phase, design phase, implementation phase, test phase, and installation and checkout phase, are performed in that order, possibly with overlap but with little or no iteration. [IEEE]

Term	Explanation
Work Breakdown Structure (WBS)	A. A product-oriented family tree division of hardware, software, services and other work tasks that organizes, defines, and graphically displays the products to be produced, as well as the work to be accomplished to achieve the specific product. The formalized WBS that is developed to the lowest meaningful level illustrates the interrelationship of tasks to be accomplished, distribution of responsibilities and assignment of resources which permits scheduling of task start and completion dates. The structure also provides a logical cost summarization capability for each level from the lowest work package to the highest summary level.