

Normes de développement logiciel mise en place au CRI de l'Université de Rennes1

Thierry Bedouin, Sylvain De Feo

Université de Rennes1



Plan de la présentation

- Contexte
- Pourquoi une démarche qualité ?
- Objectifs et méthodologie suivie
- Les résultats
- Bilan
- Obstacles et difficultés rencontrées
- Facteurs d'acceptation de la démarche
 - auprès des développeurs
 - auprès des usagers
 - auprès des responsables
- Gains obtenus en faisant de la qualité
- Contre-exemple et ses conséquences
- Pour conclure

Contexte

- Université pluridisciplinaire (~ 22000 étudiants, 4000 actifs dans la base de GRH)
- Centre des Ressources Informatiques (service commun) structuré en pôle (infrastructures, système d'information et services, proximité, administration) travaillant sur tous les domaines
- Une tradition de développement et souvent site pilote...
- Des projets locaux, régionaux et nationaux
- Des titulaires mais aussi des contractuels sur les projets
- Des besoins de qualité dans le suivi et l'exploitation
- Une organisation de pilotage du SI et des organes de concertation

Pourquoi une démarche qualité ?

- Produits développés sans unicité de technique et de méthode
- Pas de cahier des charges réels et parfois modifiés en cours de route
- Développeur en relation directe avec le client → dérapages possibles sur les fonctionnalités du produit
- Peu de documentation
- De part leur unicité, chaque produit affecté à son créateur pour toute sa durée de vie
- Multiplication des développements



Objectifs et méthodologie suivie

- Objectifs de cette normalisation :
 - permettre une meilleure gestion et évolution de nos applications (modularité des éléments du produit)
 - garantir au "client" que le produit fourni répondra à ses besoins exprimés
 - tenir les délais de livraison en cadrant les demandes d'évolutions du produit sur les prochaines versions

- Constitution d'un groupe de travail « normes »

- Constitution du groupe :
 - Chefs de projets
 - Développeurs
 - Exploitants



Les résultats

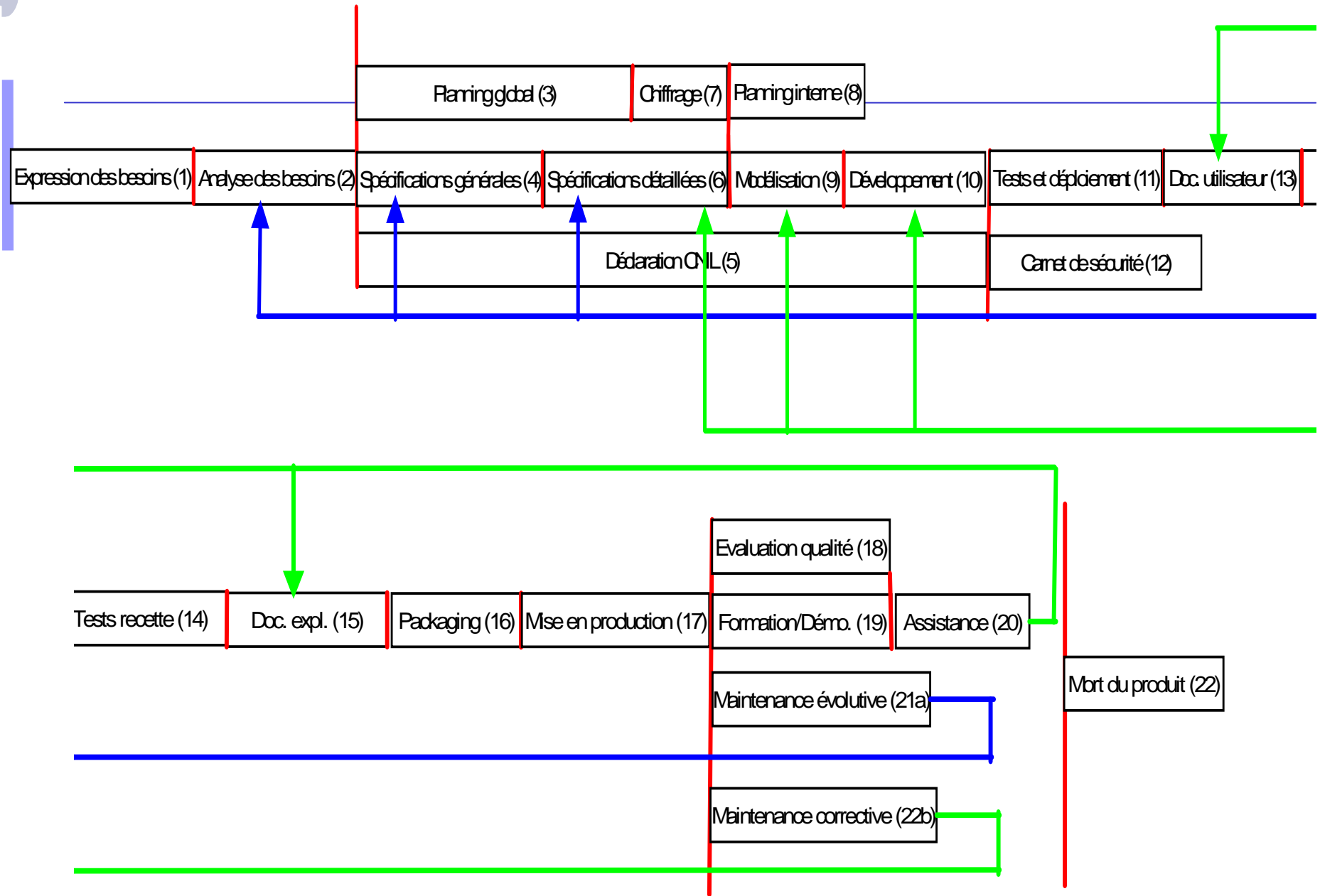
- Un document principal précisant toutes les étapes (22) de la vie d'un projet.

→ 1 étape = 1 document normatif + modèles de document

Ce document normatif décrit pour l'étape :

- les actions à réaliser,
- les acteurs de ces actions
- les documents à fournir (basés sur des modèles)

- Un document pointant l'avancée du projet
- Un développement basé sur Esup-Commons





Bilan

- Remarque : tous les documents ne sont pas encore finalisés.
- La théorie :
 - interlocuteur principal du "client" : le chef de projet.
 - cahier des charges et spécifications générales et détaillées validés par le client
 - Complétude du document d'avancement du projet
 - rédaction de la documentation fonctionnelle et technique
 - produit développé sur un modèle identique à l'ensemble des développements du service
- La pratique :
 - interlocuteur primaire du "client" : le chef de projet. Ensuite c'est le développeur .
 - cahier des charges et spécifications générales et détaillées **en général** validés par le client.
 - documentation fonctionnelle et technique : **presque** toujours présente
 - produit développé sur un modèle identique à l'ensemble des développements du service : oui depuis l'arrivée d'une version stable de Esup-Commons

Obstacles et difficultés rencontrées

- résistance au changement
- le fait d'écrire ce que l'on sait déjà est ressenti comme une perte de temps
- Tâtonnement sur les outils, méthode et forme des documents : a servi de prétexte pour ne pas s'y mettre
- Il a fallu monter en compétence technique
- Les clients ont parfois du mal à accepter ces nouvelles exigences de validation de documents.



Facteurs d'acceptation de la démarche

- auprès des développeurs
 - Le premier développeur du produit : mitigé
 - les développeurs suivant : apprécient d'être en terrain connu et déjà balisé par une technique commune et de la documentation.
- auprès des usagers
 - les usagers ne se sont pas rendus compte de cette évolution ?
- auprès des responsables
 - La production de documents à chaque étape du projet permet de suivre son état d'avancement. Et le fait de savoir que le développeur n'est pas seul mais peut s'appuyer sur une communauté active est rassurant.



Gains obtenus en faisant de la qualité

- Produits plus modulaires et plus simple à faire évoluer
- Des modules qui peuvent s'échanger d'un produit à l'autre --> module de gestion de droit par exemple.
- Documentation présente (fonctionnelle et technique)
- Sur la technique pure : quasiment tous les développeurs sont interchangeables
- La contrepartie :
 - temps de production plus long
 - mode de développement plus complexe et donc plus long que du développement sans séparation des couches logicielles.

Contre-exemple et ses conséquences

L'application de gestion interne des congés

- un des premiers développement réalisé en java. Il n'a donc pas suivi de normes particulière.
- application monolithique avec un mélange des différentes couches logicielles à tous les niveaux du produit.
- Actuellement :
 - impossible de migrer ce produit sur un serveur d'application à jour car l'application plante aussitôt.
 - diagnostic facile au vu des messages d'erreur
 - très difficile d'identifier les bout de code fautifs à cause de la conception
- Résultat :
 - application fonctionnelle et qui répond en grande partie à la demande
 - impossible à faire évoluer
 - tourne sur un serveur d'application obsolète et plus maintenu.



Pour conclure

- Il faut impérativement dégager des moyens humains pour cette démarche
- La mise en place d'un tel projet nécessite de recueillir l'assentiment des personnes impactés
- Projet qui n'est jamais fini car ne peut pas (et ne doit pas) être figé au cours du temps. En effet tout évolue très vite, surtout dans les domaines techniques.
- Démarche longue,
 - Pénalisante sur du court terme
 - Payante sur du moyen-long terme.