

Projet Informatique

Philippe Collet

Licence 3 Informatique – S5

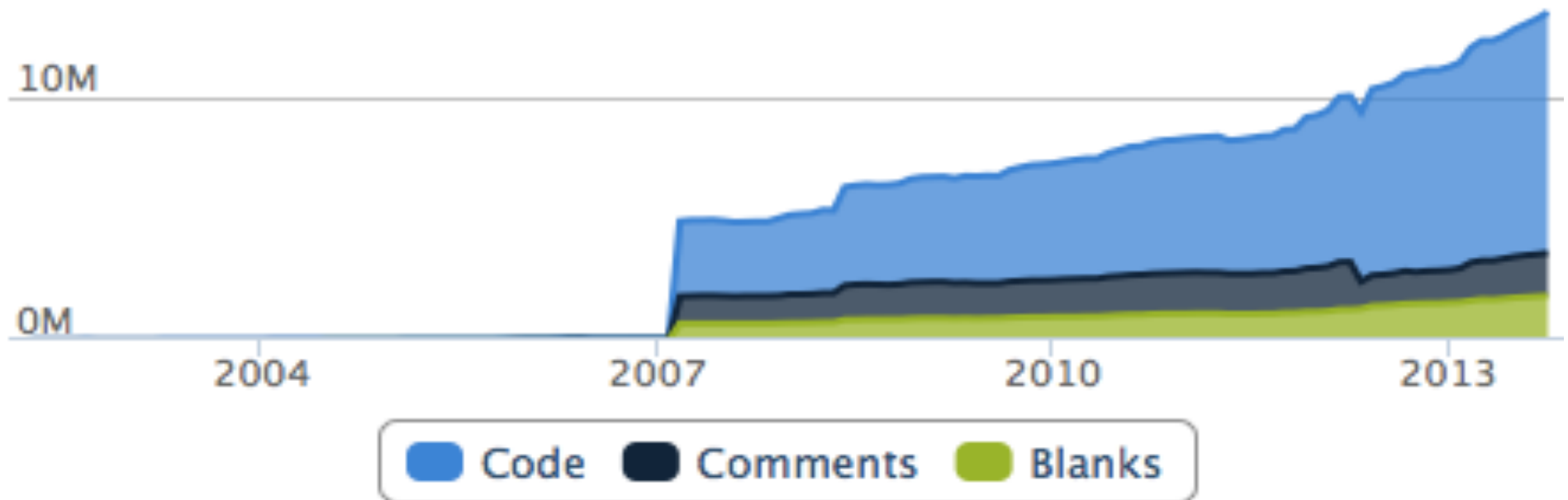
2014-2015

<http://deptinfo.unice.fr/twiki/bin/view/Linfo/ProjetInfo201415>

Réalisation d'un développement de taille conséquente ?

❑ Firefox ?

Lines of Code



Objectifs

- ❑ **Réalisation, en équipe,**
 - d'un développement de taille conséquente
 - à partir d'un cahier des charges et d'une architecture préétablis en Java

- ❑ **Donc : du développement !**
 - Sans (gros) problème de conception
 - Avec des problèmes de
 - ◆ Communication (a priori équipe de 4)
 - ◆ Techniques de programmation (API, etc.)
 - ◆ Fiabilité (tests unitaires indispensables)
 - En quasi-autonomie
 - Avec professionnalisme
 - ◆ Bonne réponse au cahier des charges
 - ◆ Efficacité, rapidité, qualité



Objectifs

❑ Travail demandé

- **Beaucoup, beaucoup (beaucoup !) de travail personnel**
- Surtout par rapport au faible volume des TD
- Pour fournir du code et de la gestion de projet (expliciter ce que vous faites et ce que vous allez faire)



❑ Problématique

- Comment vous organiser en équipe ? Développer en équipe ? Coder/tester ?
Etre efficace ? Communiquer ?
- Passer du cahier des charges à une définition et en suivi :
 - ◆ Des objectifs généraux et du livrable principal (l'application finale)
 - ◆ Des jalons pour y arriver, de comment évaluer qu'on arrive bien à ces jalons
 - ◆ Des contraintes

Calendrier

A	B	C	D	E	F
	08h - 9h30	9h 45 - 11h 15	11h 30 - 13h	13h 15 - 14h 45	15h - 16h30
11/09/2014	Cours Projet Info	Cours Projet Info			
18/09/2014	CM COO - D. Ribouchon	CM COO - D. Ribouchon		CM COO - D. Ribouchon	
25/09/2014	TD COO - S. Urli	TD COO - S. Urli			
02/10/2014	CM COO - D. Ribouchon	CM COO - D. Ribouchon		TD COO - S. Urli	TD COO - S. Urli
09/10/2014	CM COO - S. Urli	CM COO - S. Urli		TD COO - S. Urli	TD COO - S. Urli
16/10/2014	CM COO - S. Urli	CM COO - S. Urli		TD COO - S. Urli	TD COO - S. Urli
23/10/2014	Cours Projet Info	Cours Projet Info		TD COO - S. Urli	TD COO - S. Urli
30/10/2014	Interruption pédagogique	Interruption pédagogique	Interruption pédagogique	Interruption pédagogique	
06/11/2014	Cours Projet Info	Cours Projet Info		TP Projet Info	TP Projet Info
13/11/2014	TD COO - S. Urli	TD COO - S. Urli		TP Projet Info	TP Projet Info
20/11/2014	Cours Projet Info	Cours Projet Info		TP Projet Info	TP Projet Info
27/11/2014	CM COO - D. Ribouchon	CM COO - D. Ribouchon		TP Projet Info	TP Projet Info
04/12/2014	TD COO - S. Urli	TD COO - S. Urli		TP Projet Info	TP Projet Info
11/12/2014		TD COO - S. Urli		TP Projet Info	TP Projet Info
18/12/2014	soutenance projet Info (journée continue)				

Evaluation

- ☐ une note de contrôle d'avancement en TD (25%),
- ☐ une note de soutenance (20%),
- ☐ une note relative au code livré (architecture, qualité et test) (25%)
- ☐ une note d'utilisation des outils de "forge" (ticket, gestionnaire de versions, documentation) (30%)

Principe de suivi : 6 semaines de TD de suivi

☐ Jeudi après-midi :

■ 13h15 – 15h15 : 2h en présence de l'enseignant

- ◆ Questions sur les fonctionnalités à réaliser
- ◆ Propositions / validation sur le découpage du travail
- ◆ Validation sur la conception de l'application
- ◆ Surveillance de l'avancement du projet
- ◆ Suivi et évaluation de chaque membre de l'équipe individuellement
- ◆ Aide technique sur le langage utilisé

■ Jusqu'à 18h15 : salle réservée pour continuer à travailler en équipe...

■ Et en dehors des horaires :

- ◆ En profiter pour continuer d'avancer sur les points durs
- ◆ Bien valider la répartition du travail, la charge de chacun
- ◆ Mettre en place des tests
- ◆ Finaliser un test d'intégration avec toute l'équipe

Le TD de suivi ne fait que le... suivi

- ☐ **Le développement dure 6 semaines et non pas 2 ou 4 heures**
 - Il faut organiser l'activité de ces 6 semaines avant et pendant
 - Chaque heure "perdue" compte
 - Il faut gérer l'information (documents, codes, tests...) en continu, particulièrement lorsqu'on travaille à 4 ou 5...
- ☐ **Avant le démarrage des TD de suivi, il vous est demandé :**
 - De former des équipes
 - De vous auto-former (un minimum) aux outils (gestion de tickets, versioning, eclipse)
- ☐ **Pendant les premiers TD, il vous sera demandé :**
 - De formaliser le cahier des charges de ce que vous avez à réaliser sous forme de jalons (objectifs intermédiaires)

Soutenance et évaluation

☐ Soutenance (en décembre) : 15 minutes

- Présentation (technique) de la réalisation
- Fonctionnalités réalisées (ou pas)
- Choix de conception (et justification)
- Mini-démo (attention, vraiment mini)
- Travail de chacun bien identifié
- Discussion sur les problèmes (de tout type) rencontrés et les solutions apportées

☐ Evaluation

- La forme (soutenance, documents,...)
- Le code (qualité, test, performance)
- Mise en œuvre des outils
- Gestion du projet : livrables réguliers

Organisation

- ❑ Organisation générale, notion de projet, V&V
- ❑ Versioning
- ❑ Système de tickets
- ❑ Tests unitaires, Junit, Test-driven development
- ❑ Environnement de développement, Eclipse
- ❑ Construction automatique, Maven
- ❑ Métriques, Sonar
- ❑ Documentation



Qu'est qu'un projet ?

❑ Définition

- Un effort temporaire
- qui est progressivement planifié, contrôlé et exécuté
- par des personnes travaillant avec des contraintes de ressources
- pour créer un produit, service ou résultat unique

❑ Temporaire

- Début et fin sont définies
- Pas forcément court, mais fini

❑ Planifié, contrôlé et exécuté

- Nécessité d'une planification initiale et d'un suivi
- Le travail s'organise pour accomplir des objectifs (exécution)
- Le travail nécessite des vérifications pour être correctement exécuté
- Et tout cela, progressivement, en étapes, en affinant au fur et à mesure

❑ Par des personnes

- La dimension humaine est primordiale

❑ Avec des contraintes de ressources

- Contraintes de temps, de coût
- Toute limitation ou frontière du projet est une contrainte

❑ *Gérer un projet, c'est essentiellement gérer continuellement ces contraintes, pour atteindre des critères de qualité prédéfinis*



☐ Pour créer un produit, service ou résultat unique

- Le projet crée quelque chose de nouveau
- Quelque chose de tangible (produit) ou non (service, résultat)
 - ◆ Exemple : Diminuer le temps d'attente au téléphone de 20 %

☐ Comment déterminer l'objectif du projet ?

- L'objectif du projet est quelque chose que l'organisation ne peut obtenir par son fonctionnement normal
- Exemple de fonctionnement normal : Produire les fiches de paie mensuelles

☐ Questions

- Pour un constructeur de maisons, chaque chantier est-il un projet ?

Caractéristiques du projet

☐ Livrables

- La partie la plus importante d'un projet, souvent multiples
- On parle parfois d'artefact, comme quelque chose qu'il est nécessaire de produire, sans que ce soit un livrable

☐ Portée du produit

- Caractéristiques et fonctionnalités du produit

☐ Portée du projet

- Comment les objectifs vont être atteints
- Donc, le travail, et uniquement le travail, pour réaliser les... livrables
- Donc, directement impacté par le temps et le coût

☐ Impossible de définir la portée du projet sans la portée du produit

❑ Il faut bien distinguer

- **Les qualités utiles à l'utilisateur, donc *a priori* souhaitées par le client**

- ◆ Phases d'exploitation

- **Les qualités utiles au développeur**

- ◆ Phases de construction et de maintenance

❑ **Fiabilité = Validité + Robustesse**

■ **Validité (Efficacité) ≡ correction, exactitude**

- ◆ *Efficacité : qualité d'une chose ou d'une personne qui donne le résultat escompté*
- ☞ Assurer exactement les fonctions attendues, définies dans le cahier des charges et la spécification, en supposant son environnement fiable
- ☞ Adéquation aux besoins

■ **Robustesse**

- ◆ Faire tout ce qu'il est utile et possible de faire en cas de défaillance: pannes matérielles, erreurs humaines ou logicielles, malveillances...

❑ **Performance (parfois appelée efficacité)**

- Utiliser de manière optimale les ressources matérielles : temps d'utilisation des processeurs, place en mémoire, précision...

❑ **Convivialité**

- Réaliser tout ce qui est utile à l'utilisateur, de manière simple, ergonomique, agréable (documentation, aide contextuelle...

❑ **Documentation** = Tout ce qu'il faut, rien que ce qu'il faut, là où il faut, quand il faut, correcte et adaptée au lecteur : **crucial** !

❑ **Modularité** =

- **Fonctionnalité**

- ◆ Localiser un phénomène unique, facile à comprendre et à spécifier

- **Interchangeabilité**

- ◆ Pouvoir substituer une variante d'implémentation sans conséquence fonctionnelle (et souvent non-fonctionnelle) sur les autres parties

- **Évolutivité**

- ◆ Facilité avec laquelle un logiciel peut être adapté à un changement ou une extension de sa spécification

- **Réutilisabilité**

- ◆ Aptitude à être réutilisé, en tout ou en partie, tel que ou par adaptation, dans un autre contexte : autre application, machine, système...

❑ Description d'un processus pour :

- la création d'un produit
- sa distribution sur un marché
- son retrait

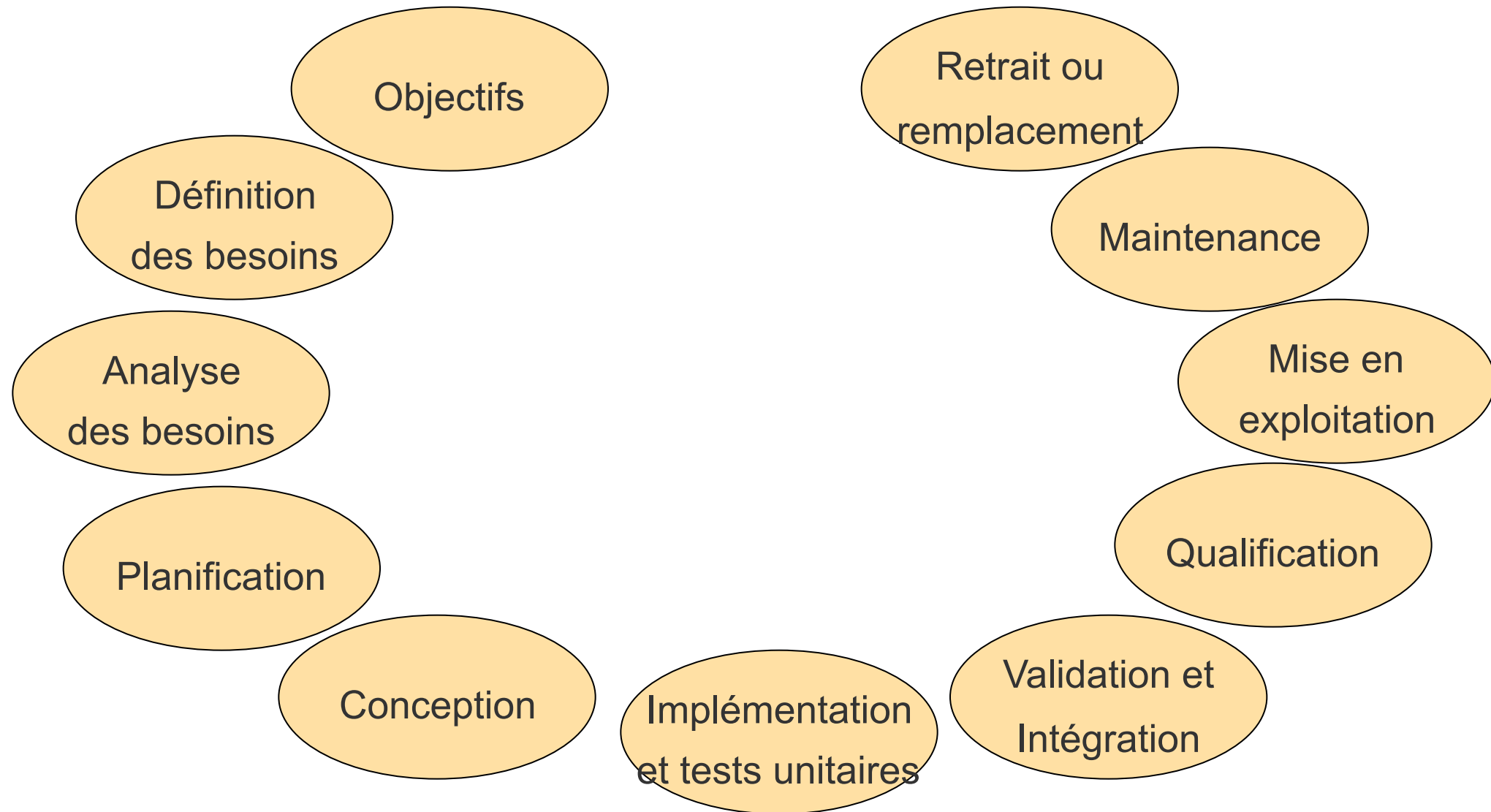
❑ Cycle de vie et assurance qualité

- Validation : le bon produit ?
- Vérification : le produit correct ?

❑ L'organisation des tâches peut être différente, suivant différents *modèles*

- *On en verra plus au 2nd semestre...*

Les phases du cycle de vie



Validation et Vérification

□ Deux aspects de la notion de qualité :

■ Conformité avec la définition : VALIDATION

- ◆ Réponse à la question : faisons-nous le bon produit ?
- ◆ Contrôle en cours de réalisation, le plus souvent avec le client
- ◆ Défauts par rapport aux besoins que le produit doit satisfaire

■ Correction d'une phase ou de l'ensemble : VERIFICATION

- ◆ Réponse à la question : faisons-nous le produit correctement ?
- ◆ Tests
- ◆ Erreurs par rapport aux définitions précises établies lors des phases antérieures de développement

- ❑ **Les spécifications fonctionnelles définissent les intentions**
 - Elles sont créées lors de la phase d'analyse des besoins
- ❑ **La vérification du produit consiste à vérifier la conformité vis-à-vis de ces spécifications fonctionnelles**
 - Revues, inspections, analyses, tests fonctionnels et structurels en boîte blanche
- ❑ **La validation du produit consiste à vérifier par le donneur d'ordre la conformité vis-à-vis des besoins**
 - Le plus souvent, tests fonctionnels en boîte noire
 - Théoriquement, la validation devrait être plutôt faite par les utilisateurs, sans tenir compte du cahier des charges
 - En pratique, la validation s'appuie sur le cahier des charges pour créer des tests d'acceptation...

- ❑ **Portent sur des documents (plutôt des programmes), sans exécuter le logiciel**

- ❑ **Avantages**

- contrôle systématique valable pour toute exécution, applicables à tout document

- ❑ **Inconvénients**

- Ne portent pas forcément sur le code réel
- Ne sont pas en situation réelle (interaction, environnement)
- Vérifications sommaires, sauf pour les preuves
- Ces preuves nécessitent des spécifications formelles et complètes, donc difficiles

Techniques dynamiques

❑ Nécessitent une exécution du logiciel, une parmi des multitudes d'autres possibles

❑ Avantages

- Vérification avec des conditions proches de la réalité
- Plus à la portée du commun des programmeurs

❑ Inconvénients

- Il faut provoquer des expériences, donc écrire du code et construire des données d'essais
- Un test qui réussit ne démontre pas qu'il n'y a pas d'erreurs

👉 ***Les techniques statiques et dynamiques sont donc complémentaires***

