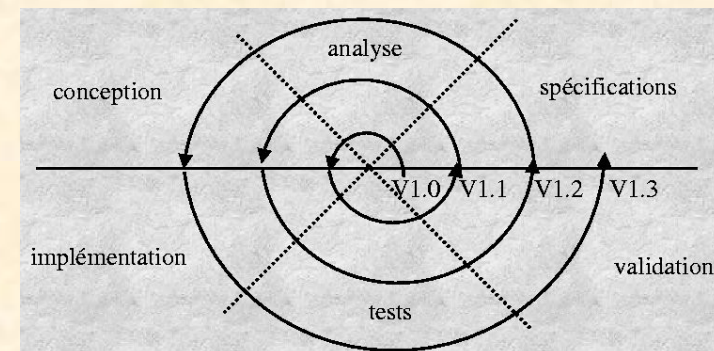




# Organisation des TD/TP

- Les TD et TP seront faits par groupe de 6 étudiants environ. La constitution d'un groupe ne peut changer qu'entre deux projets.
- **Méthodes Agiles**
  - ⇒ **Cycle de vie en Spirale**
  - ⇒ **itérations !**
  - **Fiches d'itérations : quoi, par qui, pour quand**
  - **Réunions, ordre du jour et compte rendu**
- Notation : note globale du projet + note personnelle
- Durée : 2 semaines





# Projet 2 : réunions

---

- 2 réunions client (R. externe) au début de chaque itération :
  - Analyse des besoins du client
  - Recette intermédiaire (validation maquette, spécifications, etc.)
- 1 réunion du comité de pilotage (R. interne) (fin d'IT1) :
  - Bilan et évaluation IT1 par le Directeur de Projets (qui a fait quoi, etc.)
  - Préparation de la réunion client RE2 :
    - *Validation des documents à présenter au client*
    - *Réflexion sur les points à éclaircir (spécifications)*
    - *Validation de l'ordre du jour*
  - Validation de la fiche d'itération 2
- Ordres du jour à envoyer au moins 24h avant
- Comptes-rendus à envoyer au plus tard 24h après
- **Une réunion se prépare !!**



# Projet 2 : rendu 1/3

---

- Deux document à rendre :
  - Rapport de conception (au format PDF)
  - Archive contenant code + doc + manuel utilisateur
- TOUS les documents doivent exister en fin d'IT1, même si certaines parties manquent d'exhaustivité ou de précision



# Projet 2 - rendu 2/3

---

## ■ Dossier allégé de conception (inclure ce qui semble pertinent) :

- ❑ *Introduction du projet*
  - Présentation générale du sujet, membres de l'équipe.
- ❑ *Spécifications allégées* (01 pp. 88-89, 02 pp.38-42)
  - **Analyse des besoins (cahier des charges)**, cas d'utilisation, maquette, etc.
- ❑ *Planification*
  - Découpage en lots / livrables
  - Planning prévisionnel
- ❑ *Analyse fonctionnelle allégée* (01 p. 90, 02 pp. 43-50)
  - Analyse statique : modèles métiers (définition des entités, diagramme de classes métier)
  - Analyse dynamique : interactions (entre acteurs et composants système) : diagrammes simplifiés de séquences
  - Navigation dans l'interface graphique : diagramme d'activités de navigation
- ❑ *Analyse technique allégée*
  - Outils, bibliothèques et méthodes de développement utilisés (version, justification) pour chaque composant, etc.
- ❑ *Conception* (01 p. 90, 02 pp. 51-53)
  - **Définition des classes + diagramme de classes**
- ❑ *Implémentation*
  - Diagramme de composants, diagramme de déploiement
- ❑ *Bilan de la planification*
  - Planification Réelle
  - Bilan
- ❑ *Annexes*
  - OdJ et CR des réunions (externes et internes), fiches d'itération



# Projet 2 - rendu 3/3

---

- Code source sous forme d'archive au format `.tar.gz`  
Lors du désarchivage, on doit obtenir un répertoire contenant :
  - Le fichier `.pro`
  - Éventuellement dans des sous-répertoires :
    - *Les sources : `.ui`, `.h`, `.cpp`*
    - *Les ressources éventuelles : images, etc.*
    - *La documentation générée avec Doxygen*
    - *Le manuel d'utilisateur (installation / utilisation)*



# Dates importantes

---

- Départ: Lundi 17/10 16h.
- RE1: Lundi 17/10.
  - 2 fois 15 minutes :
    - *Groupes A et D : 16h00 et 17h00*
    - *Groupes B et E : 16h20 et 17h20*
    - *Groupes C et F : 16h40 et 17h40*
- Fin IT1: Dimanche 23/10 16h.
- RI1: Lundi 24/10.
  - *Groupes A et D : 16h00*
  - *Groupes B et E : 16h30*
  - *Groupes C et F : 17h00*
- RE2: Mardi 25/10.
  - *Groupes A et D : 12h15*
  - *Groupes B et E : 12h45*
  - *Groupes C et F : 13h15*
- Rendu final: Vendredi 28/10 23h59.



# Projet 2

## Fouilles archéologiques

---

### ■ Objectifs :

Concevoir et réaliser un logiciel graphique d'archéologie permettant de conserver, classer, interroger les objets découverts dans différentes fouilles archéologiques ainsi que les campagnes de fouilles et les informations sur les personnes participant aux fouilles.

### ■ Fonctionnalités du logiciel :

- Ajouter de nouveaux sites archéologiques
- Ajouter des fouilleurs
- Ajouter les objets découverts pendant une fouille
- Ajouter de nouveaux types, matières et musées de destination
- Classer/Rechercher les objets selon certains critères et compositions possibles
- Afficher/Éditer des caractéristiques d'une liste d'objets choisis suite à une requête

### ■ Contraintes :

- Toutes les personnes ne participent pas à la fouille
- Recoupement d'objets entre type, matières et dimension
- Stockage des informations réalisée à l'aide d'un fichier texte. SGBD proscrit.
- Le langage de programmation sera le C++ couplé à la bibliothèque Qt 4.7. Les bibliothèques STL et boost interdites.