**CHAPITRE 3 :**  **METHODOLOGIE**

* 1. **Définition des besoins fonctionnels et techniques**
     1. Identification des utilisateurs cibles (ingénieurs, techniciens, étudiants)
     2. Définition des fonctionnalités principales du progiciel :

1. Saisie des données géométriques et mécanique
2. Calcul automatique des sollicitations
3. Dimensionnement des sections en béton armé
4. Vérification des contraintes et des armatures
5. Génération de rapports
   1. **Dimensionnement théorique du dalot en béton armé**
      1. Hypothèses de calcul (type de sol, conditions de charge, appuis, etc.)
6. Dalot à une travée
7. Dalot à travée multiple
   * 1. Représentation schématique du dalot (structure isostatique ou hyperstatique)
     2. Détermination des charges :
8. Charges permanentes (poids propre, remblai)
9. Charges d’exploitation (trafic routier)
10. Surcharge de terre et poussée latérale
    * 1. Méthodes de calcul des efforts internes (moment fléchissant, effort tranchant, effort normal)
    1. **Développement des algorithmes de calcul**
       1. Formulation mathématique des équations de dimensionnement
       2. Application des formules de béton armé (section rectangulaire, vérification à l’ELS et ELU)
       3. Dimensionnement des armatures longitudinales et transversales
       4. Vérification des contraintes admissibles et des flèches
    2. **Conception et développement du progiciel**
       1. Choix des bibliothèques :
11. NumPy pour les calculs numériques
12. Matplotlib pour les graphiques
13. Tkinter pour l’interface utilisateur
14. ReportLab pour les rapports de calculs
    1. **Architecture du programme :**
       1. Modules de calcul
       2. Interface graphique
       3. Gestion des entrées/sorties

**CHAPITRE 4** : **RESULTATS ET INTERPRETATIONS**

4.1. Validation et tests

* + 1. Tests unitaires des modules de calcul
    2. Vérification des résultats par comparaison avec des calculs manuels
    3. Vérification des résultats par comparaison avec les résultats obtenus sur robot
    4. Étude de cas pratique : application du progiciel à un projet réel ou académique
    5. Analyse des performances (temps de calcul, précision, ergonomie)

~~6. 🔄 Amélioration continue~~

* ~~Recueil des retours d’utilisateurs (enseignants, professionnels)~~
* ~~Identification des limites du progiciel~~
* ~~Propositions d’améliorations futures :~~
* ~~Ajout d’autres types de dalots~~