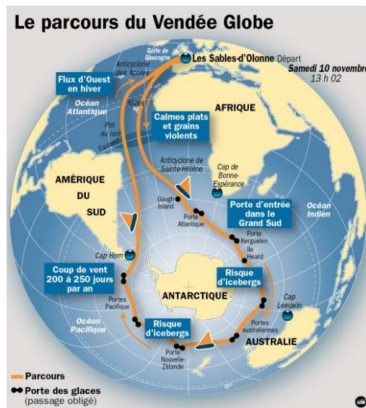


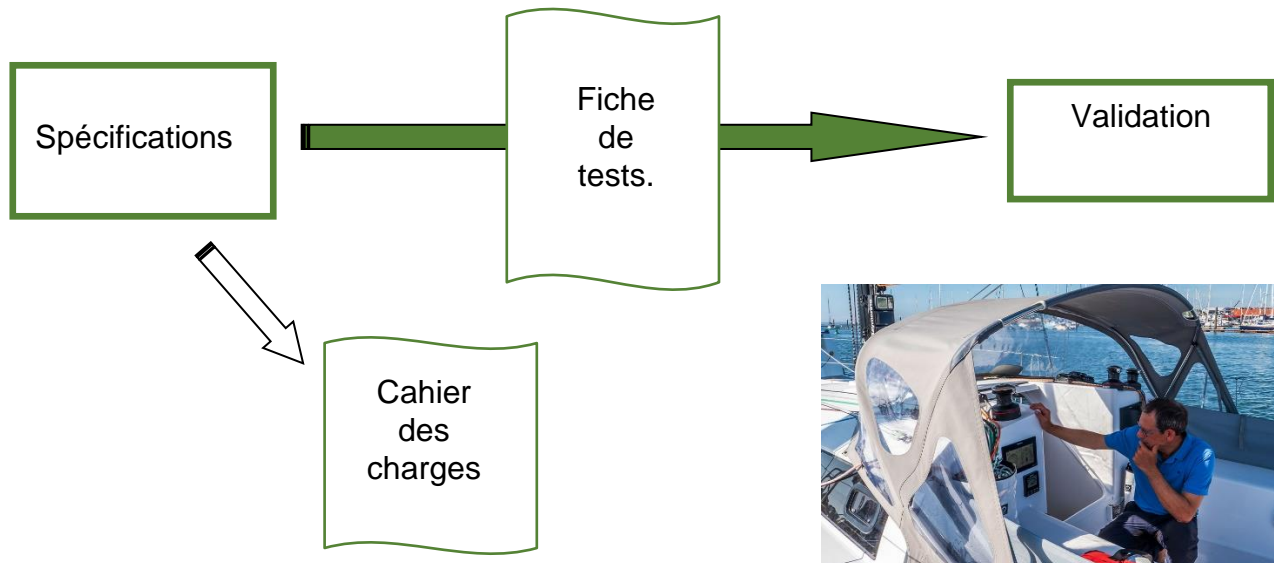
I. Exemple de projet :

1. Cahier des charges : expression du besoin

Expression du besoin :



Architecte : Marc Lombard
Chantier : MAG France
Date de lancement : 1998
Longueur : 18,28 m
Largeur : 5,30 m
Tirant d'eau : 4,50 m
Déplacement (poids) : 9 tonnes
Nombre de dérives : 2
Hauteur mât : 29 m
Voile quille : Acier mécano soudé
Surface de voiles au près : 260 m²
Surface de voiles au portant : 580 m²

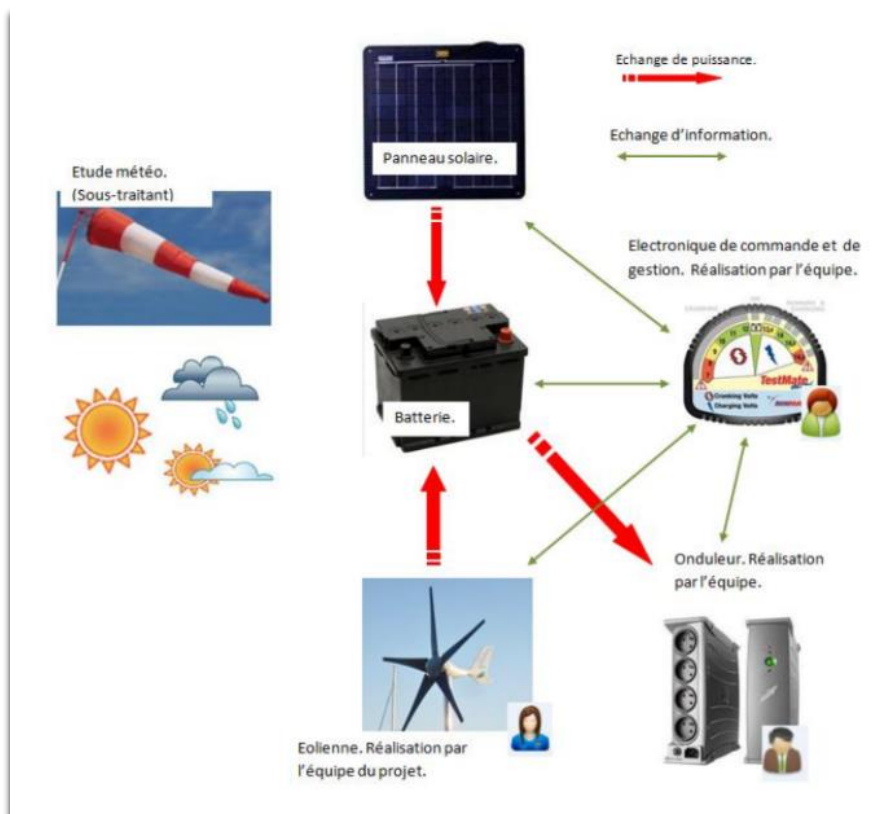
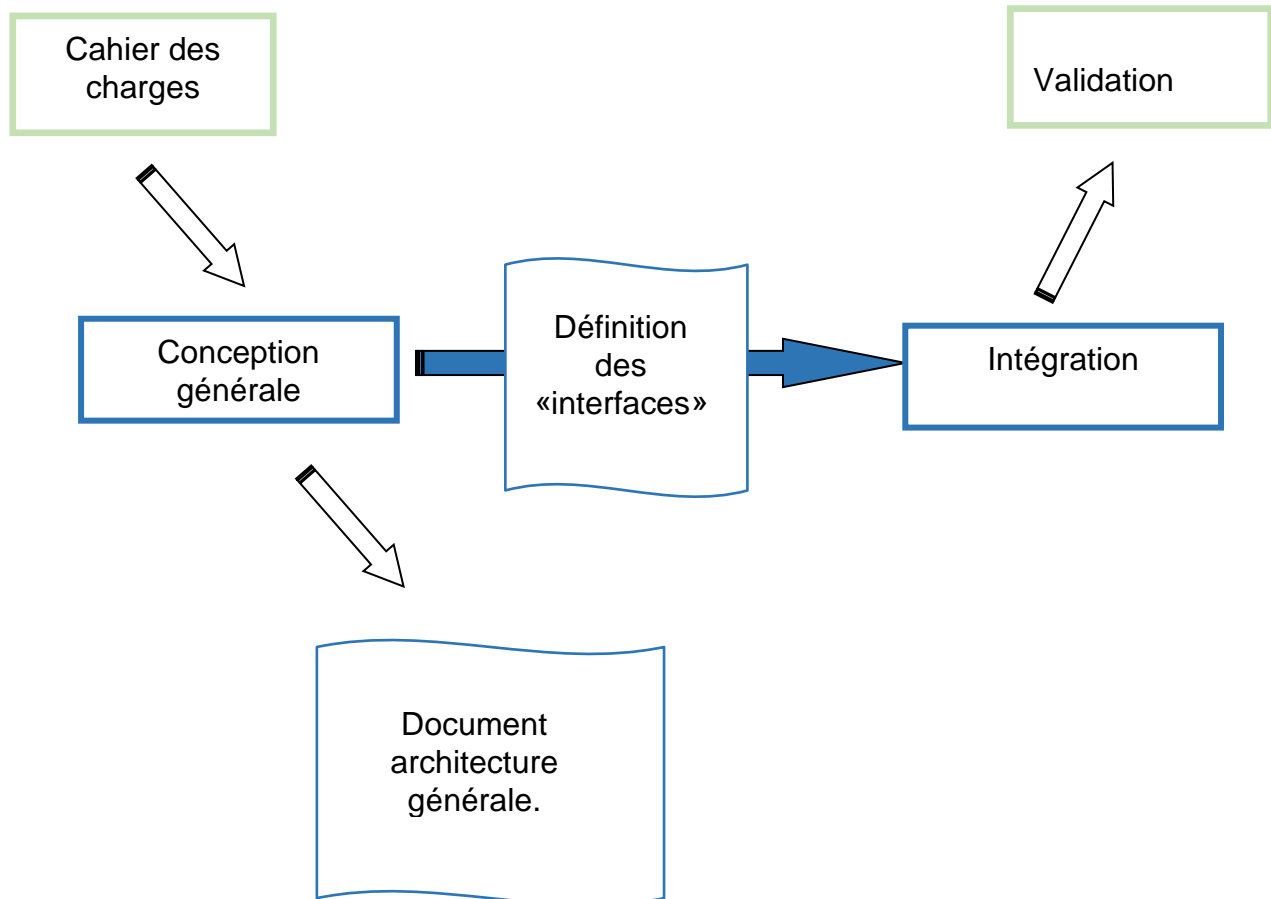


Cahier des charges : Assurer l'autonomie en énergie électrique d'un voilier.

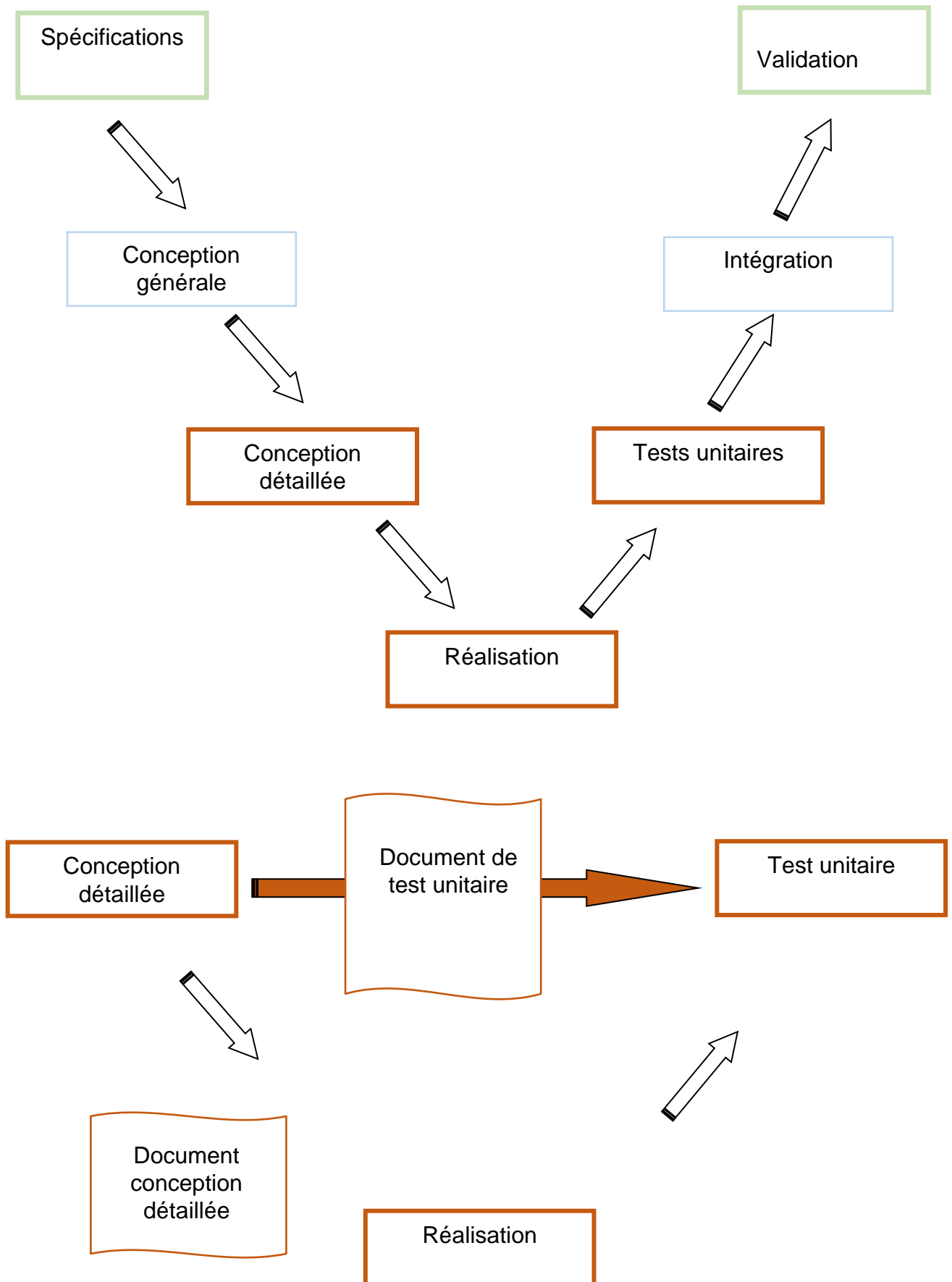
- Le système électrique doit être complètement autonome pendant 4 mois.
- Il doit fournir une énergie de 2kwh chaque les jours.
- Il doit fournir une alimentation 12V continu et 220V alternatif à 50 Hz.
- Il doit fournir une puissance maximale de 1500W.
- Le poids de l'ensemble sera au maximum de 80.

Validation : Avant de partir il faudra s'assurer du bon fonctionnement du système d'alimentation. Pour cela il faudra qu'il passe une série de tests, définis au préalable.

2. Conception générale : architecture de la solution :

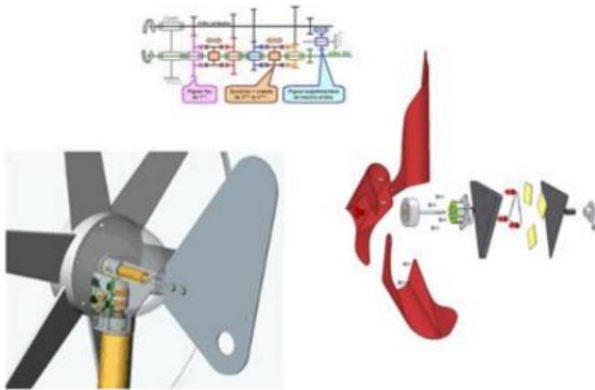


3. Conception détaillée et réalisation



a. Réalisation de l'éolienne :

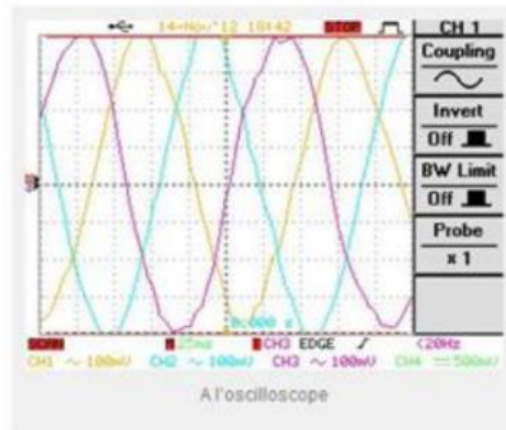
- Document de conception unitaire:



- Réalisation :

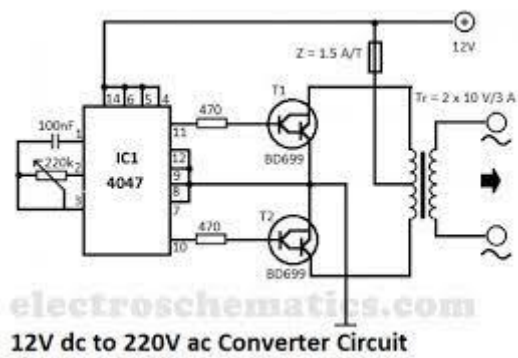


- Test :



b. Onduleur

- Conception



- Réalisation



- Test



c. Tous les autres sous-systèmes sont conçus suivant la même méthode

II. Jeu de briques

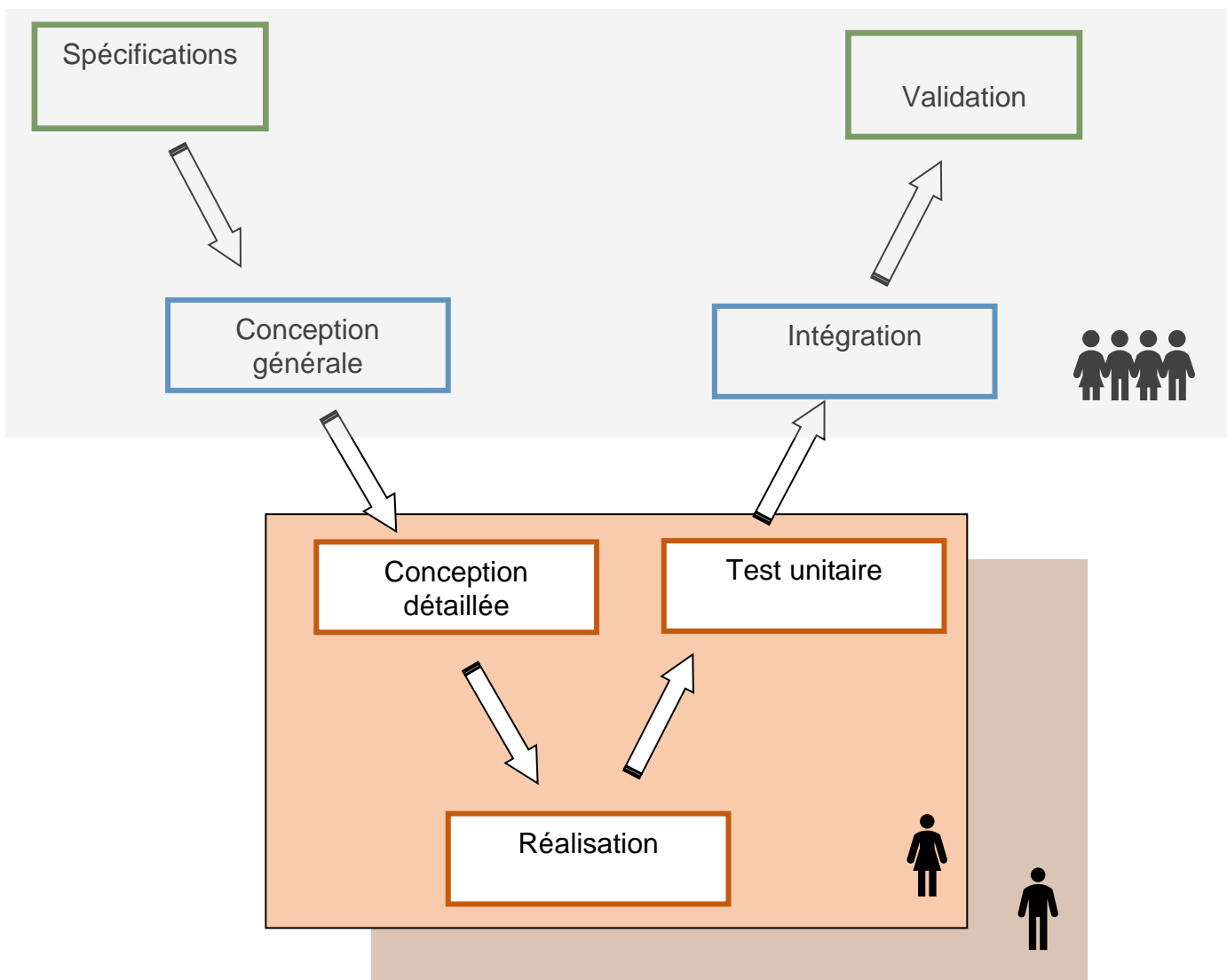
1. Présentation (wikipédia) :

Le **casse-briques** est un [genre de jeu vidéo](#) souvent classé dans la catégorie [arcade](#), apparu en 1975 avec le jeu [Breakout](#). Il est directement inspiré de [Pong](#). Le principe général est de détruire, au moyen d'une ou plusieurs balles, un ensemble de briques se trouvant dans un niveau pour accéder au niveau suivant.



2. Conduite de projet :

La conduite de projet suivra la description ci-dessous :



Toutes les étapes du projet seront évaluées.

3. Spécifications :



a. Cahier des charges

- Définir les règles du jeu et le comptage des points,
- Définir le nombre de rangées de briques, le nombre de briques par rangée,
- La taille des briques, de la raquette et de la balle.

b. Fiche de validation

Décrire l'ensemble des tests que doit vérifier le jeu pour fonctionner en respectant le cahier des charges.

c. Evaluation de l'activité et documents à rendre :

Rendre le cahier des charges rédigé ainsi qu'une fiche de validation. Ces deux documents pourront être dans le même fichier mais il est impératif qu'ils soient rendus avant la suite du projet.

4. Conception générale :

a. Document Architecture générale

- Choisir le package qui supportera l'interface graphique du jeu,
- A partir du package choisi, déterminer un algorithme général,
- Proposer une organisation des données,
- Définir les objets (ne définir que les méthodes de l'objet) qui seront utilisés,
- Répartir le travail entre chaque membre de l'équipe,
- Choisir le nom des « éléments » commun (package, objet, classe, variable partagées, ...)



b. Définition des interfaces (signatures)

- Pour chaque objet et leurs méthodes définir les prototypes (paramètres d'appel) et les « doc string ». Pour certaines méthodes on définira les « asserts » pour les tests.



Remarque : dans le cas d'une conception objet, c'est façon de procéder est parfois nommée HOOD (*Hierarchic Object-Oriented Design*).

c. Evaluation de l'activité et documents à rendre :

- Rendre le document d'architecture générale,
- Définitions des interfaces.



Remarque : on peut procéder suivant une méthode dite « **Agile** » c'est-à-dire réaliser une intégration du projet avant le développement complet de tous ses composants. Et ainsi vérifier un fonctionnement dégradé.

Par exemple dans ce cas faire fonctionner le projet :

- Avec une seule brique.
- Avec la balle qui déplace sans détruire les briques.
- Qui ne compte pas les points.

5. Conception détaillée



- Proposer un algorithme simple,
- Définir les classes (Attributs, commenter les méthodes avec « # »
 - Définir le corps de : « if __name__ == '__main__': »

6. Réalisation, codage test unitaire



- Réaliser le codage
- Mettre au point en utilisant le débogueur .
- Vérifier les tests : Exécuter le package et vérifier son bon fonctionnement.

7. Intégration

Réaliser l'intégration : package par package.

8. Validation.

Passer les tests définis dans la partie spécification.

Evaluation du projet :

L'ensemble des compétences suivantes seront évaluées :

- Analyser et modéliser un problème
- Décomposer un problème en sous-problèmes
- Concevoir des solutions algorithmiques
- Mobiliser les concepts et les technologies
- Traduire un algorithme dans un langage de programmation
- Développer des capacités d'abstraction et de généralisation

Les compétences seront notées de la manière suivante :

0 pts absences , 2 pts Réalisé de manière incorrecte, 3 pts Réalisé correctement.

Spécification	<ul style="list-style-type: none">• Analyser et modéliser un problème
Conception générale	<ul style="list-style-type: none">• Décomposer un problème en sous-problèmes,• Concevoir des solutions algorithmiques,• Développer des capacités d'abstraction et de généralisation• Mobiliser les concepts et les technologies
Conception détaillée	<ul style="list-style-type: none">• Concevoir des solutions algorithmiques
Codage	<ul style="list-style-type: none">• Traduire un algorithme dans un langage de programmation
Test unitaire	<ul style="list-style-type: none">• Développer des capacités d'abstraction et de généralisation
Intégration	<ul style="list-style-type: none">• Développer des capacités d'abstraction et de généralisation