

# Résumé approfondi et détaillé du cours - Études avant projets

## 1. DÉFINITIONS FONDAMENTALES DES SERVICES INFORMATIQUES

### 1.1 Fourniture de services informatiques

**Définition complète :** La fourniture de services informatiques peut être définie comme un ensemble de ressources organisées mises à disposition pour atteindre un objectif ayant une valeur ajoutée pour ses utilisateurs. Cette définition implique plusieurs éléments clés :

- Une **organisation structurée** des ressources
- Un **objectif précis** à atteindre
- Une **valeur ajoutée** mesurable pour l'utilisateur final

### 1.2 Classification des services informatiques

**Services simples :**

- Impression de documents
- Gestion des emails
- Numérisation (scans)
- Caractéristiques : monolithiques, limités à une fonction unique, facilement compréhensibles

**Services intermédiaires :**

- SaaS (Software as a Service)
- IaaS (Infrastructure as a Service)
- PaaS (Platform as a Service)
- Caractéristiques : niveau d'abstraction moyen, nécessitent une compréhension technique

**Services complexes :**

- Plateformes type leboncoin.fr
- Réseaux sociaux type facebook.fr
- Caractéristiques : multiples composants, architecture distribuée, milliers d'utilisateurs simultanés

### 1.3 Distinction Service vs Produit

**SERVICE :**

- **Définition :** Un moyen de permettre la co-création de valeur ajoutée en aidant les clients à se rapprocher de leurs résultats sans leur imposer des coûts et des risques spécifiques
- **Caractéristiques :**
  - Immatériel
  - Co-création de valeur (collaboration client-fournisseur)
  - Minimisation des risques pour le client
  - Mutualisation des coûts
  - Évolutif et adaptable

**PRODUIT :**

- **Définition :** Ressources agencées et configurées pour offrir de la valeur au consommateur
- **Caractéristiques :**
  - Tangible ou intangible mais défini
  - Configuration fixe à un instant T
  - Propriété transférable
  - Valeur intrinsèque

**Exemple de distinction :**

- **Produit :** Acheter Microsoft Office en licence perpétuelle (version 2019)
- **Service :** S'abonner à Microsoft 365 (accès continu, mises à jour automatiques)

## 2. LE CLOUD COMPUTING EN PROFONDEUR

### 2.1 Définition et concept

**Cloud Computing** : En français "informatique en nuage", correspond à l'accès à des services informatiques (serveurs, stockage, mise en réseau, logiciels) via Internet (le « cloud » ou « nuage ») à partir d'un fournisseur.

**Métaphore du nuage** : Internet est représenté comme un nuage car l'utilisateur n'a pas besoin de connaître l'emplacement physique exact des ressources.

### 2.2 Les trois modèles de services cloud (SaaS, PaaS, IaaS)

#### IaaS - Infrastructure as a Service

- **Niveau le plus bas** de l'abstraction cloud
- **Fourni** : Serveurs virtuels, stockage, réseaux
- **Géré par le client** : OS, middleware, applications, données
- **Exemples** : Amazon EC2, Google Compute Engine, Azure Virtual Machines
- **Cas d'usage** : Entreprises voulant contrôler totalement leur infrastructure logicielle
- **Avantages** : Flexibilité maximale, contrôle total
- **Inconvénients** : Nécessite expertise technique, responsabilité de maintenance

#### PaaS - Platform as a Service

- **Niveau intermédiaire** d'abstraction
- **Fourni** : Infrastructure + OS + environnement d'exécution
- **Géré par le client** : Applications et données
- **Exemples** : Heroku, Google App Engine, Azure App Service
- **Cas d'usage** : Développeurs souhaitant se concentrer sur le code sans gérer l'infrastructure
- **Avantages** : Gain de temps, déploiement rapide, scalabilité automatique
- **Inconvénients** : Moins de contrôle, dépendance au fournisseur

#### SaaS - Software as a Service

- **Niveau le plus élevé** d'abstraction
- **Fourni** : Application complète clé en main
- **Géré par le client** : Uniquement les données utilisateur et configuration
- **Exemples** : Gmail, Salesforce, Office 365, Dropbox
- **Cas d'usage** : Utilisateurs finaux, pas de compétences techniques nécessaires
- **Avantages** : Simplicité maximale, accessibilité immédiate
- **Inconvénients** : Peu de personnalisation, dépendance totale au fournisseur

### 2.3 Types de déploiement cloud

#### Cloud Public

- Infrastructure partagée entre plusieurs clients
- Géré par des fournisseurs tiers (AWS, Azure, Google Cloud)
- Coûts réduits par mutualisation
- Moins de contrôle sur la sécurité et localisation des données

#### Cloud Privé

- Infrastructure dédiée à une seule organisation
- Peut être hébergée sur site (on-premise) ou par un tiers
- Contrôle maximal sur sécurité et conformité
- Coûts plus élevés

#### Cloud Hybride

- **Définition clé** : Combinaison de cloud public et privé
- **Réalité terrain** : Il est très rare qu'une entreprise ait 100% de ses services hébergés en cloud
- **Pratique courante** : Architectures hybrides mélangeant ressources locales et ressources accessibles sur Internet

- **Stratégie** : Données sensibles en privé, applications standards en public
- **Flexibilité** : Permet de choisir le meilleur environnement pour chaque workload

## Multicloud

- Utilisation simultanée de plusieurs fournisseurs cloud
- Évite le vendor lock-in (dépendance à un seul fournisseur)
- Optimisation des coûts et performances

## 2.4 Changement de paradigme économique

### Modèle traditionnel (avant le cloud)

- **Acquisition** : Achat d'un droit d'usage pour période illimitée (ou quasiment illimitée, aux limites de durée de garantie près)
- **Modèle économique** : CAPEX (Capital Expenditure - dépenses d'investissement)
- **Caractéristiques** :
  - Coût initial élevé
  - Amortissement sur plusieurs années
  - Propriété du matériel/logiciel
  - Maintenance à charge de l'entreprise
  - Obsolescence programmée

### Modèle cloud (actuel)

- **Accès** : Location de services moyennant un loyer (abonnement)
- **Modèle économique** : OPEX (Operational Expenditure - dépenses opérationnelles)
- **Caractéristiques** :
  - Paiement à l'usage (pay-as-you-go)
  - Pas de coût initial important
  - Pas de propriété des ressources
  - Maintenance incluse par le fournisseur
  - Mises à jour automatiques
  - Scalabilité (ajustement selon les besoins)

### Exemple comparatif :

- **Avant** : Achat d'un serveur pour 10 000€, amortissement sur 5 ans
- **Maintenant** : Abonnement serveur cloud à 200€/mois, ajustable selon charge

## 2.5 Risques inhérents au cloud (CRITIQUE pour le test)

### Propriété des données

- **Question** : À qui appartiennent les données stockées dans le cloud ?
- **Problématique** : Clauses contractuelles parfois floues
- **Enjeu** : Droits d'utilisation, de modification, de revente des données par le fournisseur ?

### Localisation des données

- **Question** : Où sont physiquement hébergées les données ?
- **Problématique** : Impact sur la latence, la souveraineté numérique
- **Exemple** : Données de citoyens français hébergées aux USA = problème RGPD

### Juridiction applicable

- **Question** : Quelle loi s'applique en cas de litige ?
- **Problématique** : Cloud Act américain vs RGPD européen
- **Exemple** : Un fournisseur US peut être contraint de transmettre données au gouvernement US, même si hébergées en Europe

### Réversibilité

- **Question** : Peut-on récupérer facilement ses données et changer de fournisseur ?
- **Problématique** : Vendor lock-in (dépendance fournisseur)
- **Enjeux** :

- Format des données exportables ?
- Coûts de migration ?
- Compatibilité avec autres systèmes ?
- Temps nécessaire pour migration ?

## 2.6 Internalisation vs Externalisation

### Années 1990 : Infogérance

- Terme historique pour désigner l'externalisation de la gestion informatique
- Transfert de la gestion du SI à un prestataire externe

### Aujourd'hui : Gouvernance du cloud

- **Principe** : Choisir un niveau de cloud pour un service donné correspond à acter un niveau d'externalisation
- **Décision stratégique** :
  - Plus on monte dans l'abstraction (IaaS → PaaS → SaaS), plus on externalise
  - Plus on externalise, moins on a de contrôle
  - Trade-off entre contrôle et simplicité/coût

## 3. TENDANCES ET PROJETS ACTUELS

### 3.1 Projets demandeurs en cloud

#### Intelligence Artificielle et Machine Learning

- **Besoins** : Puissance de calcul massive (GPU/TPU)
- **Volume de données** : Téraoctets à pétaoctets
- **Raison du cloud** : Coûts prohibitifs d'infrastructure locale, scalabilité nécessaire
- **Exemples** : Entraînement de modèles de deep learning, ChatGPT, reconnaissance d'images

#### Big Data

- **Caractéristiques** : 5V (Volume, Vélocité, Variété, Véracité, Valeur)
- **Besoins** : Stockage distribué, traitement parallèle
- **Technologies** : Hadoop, Spark, bases NoSQL
- **Raison du cloud** : Élasticité pour gérer pics de charge

#### Micro-services

- **Architecture** : Décomposition d'applications en services indépendants
- **Avantages** : Déploiement continu, résilience, scalabilité fine
- **Technologies** : Docker, Kubernetes, API REST
- **Raison du cloud** : Orchestration, auto-scaling, haute disponibilité

#### Internet des Objets (IoT)

- **Caractéristiques** : Milliards d'objets connectés générant données en continu
- **Besoins** : Ingestion de flux de données, traitement temps réel
- **Exemples** : Capteurs industriels, smart cities, wearables
- **Raison du cloud** : Centralisation des données, traitement à grande échelle

### 3.2 Demain : L'informatique quantique

**Principe** : Utilisation de qubits exploitant superposition et intrication quantiques

**Contrainte physique majeure** : Nécessite des températures proches du zéro absolu (-273,15°C) pour fonctionner

- **Raison** : Éviter la décohérence quantique
- **Infrastructure nécessaire** : Systèmes de refroidissement cryogéniques extrêmement coûteux

**Conséquence** : Cloud obligatoire

- Impossible pour la plupart des organisations de maintenir cette infrastructure

- Accès via cloud uniquement (IBM Quantum, Google Quantum AI, Amazon Braket)
  - Modèle pay-per-use pour temps de calcul quantique
- 

## 4. LE CONCEPT ATAWADAC (ESSENTIEL)

### 4.1 Signification

- **AT : Anytime** = N'importe quand (24/7/365)
- **AW : Anywhere** = N'importe où (mobilité totale)
- **AD : Any Device** = N'importe quel appareil (smartphone, tablette, PC, montre connectée)
- **AC : Any Content** = N'importe quel contenu (documents, vidéos, applications)

### 4.2 Attente utilisateur moderne

Citation clé : "Je veux pouvoir accéder à tout, n'importe quand, depuis n'importe où, à partir de n'importe quel terminal"

### 4.3 Implications techniques

- **Synchronisation** : Données accessibles depuis tous devices
- **Responsive design** : Interfaces adaptées à toutes tailles d'écran
- **APIs** : Interconnexion des systèmes
- **Sécurité** : Authentification multi-facteurs, SSO
- **Performance** : Bande passante, CDN, caching

### 4.4 Implications business

- **Expérience utilisateur** : Standard moderne attendu
  - **Compétitivité** : Obligation pour rester pertinent
  - **Productivité** : Travail à distance, mobilité
  - **Cloud** : Seule solution permettant vraiment ATAWADAC
- 

## 5. OFFRE DE SERVICES

### 5.1 Définition

Les fournisseurs de services présentent leurs services aux consommateurs sous la forme d'offre de services décrivant un ou plusieurs services basés sur un ou plusieurs produits.

### 5.2 Composantes d'une offre de services

#### Éléments descriptifs

- Nom du service
- Description fonctionnelle
- Cas d'usage
- Bénéfices attendus

#### Éléments techniques

- Architecture (IaaS/PaaS/SaaS)
- Performances (temps de réponse, disponibilité)
- Capacités (stockage, bande passante, nombre d'utilisateurs)
- Sécurité et conformité

#### Éléments contractuels

- **SLA** (Service Level Agreement) : Engagements de disponibilité
- Support (niveaux, horaires, délais de réponse)
- Tarification (abonnement, usage, paliers)
- Conditions générales

## Éléments commerciaux

- Prix et modèles tarifaires
- Périodes d'essai
- Options et add-ons
- Conditions de résiliation

### 5.3 Exemples d'offres (selon le document)

**Microsoft** : Offres multiples combinant plusieurs produits et services **OVH** : Gamme complète IaaS/PaaS/SaaS pour marché européen avec souveraineté des données

---

## 6. PHILOSOPHIE D'AMÉLIORATION CONTINUE

### 6.1 Principe "Soyez acteurs !"

**Message clé** : Pour que l'amélioration continue d'un système fonctionne, chacun doit se sentir concerné par l'amélioration du système.

### 6.2 Application pratique

- **Quel que soit votre niveau hiérarchique** : VOUS devez suggérer des améliorations
- **Attitude** : Être acteur plutôt que spectateur
- **Responsabilité collective** : Impliquer les autres
- **Culture** : Kaizen (amélioration continue japonaise), Lean, Agile

### 6.3 Bénéfices

- Innovation bottom-up (de la base vers le sommet)
- Engagement des équipes
- Détection précoce des problèmes
- Optimisation continue des processus

---

## 7. MAÎTRISE D'OUVRAGE (MOA) ET MAÎTRISE D'ŒUVRE (MOE)

### 7.1 Origine des concepts

**Domaine d'origine** : Bâtiment et ingénierie civile **Adaptation** : Le monde des systèmes d'information s'en est fortement inspiré pour mettre en place sa propre organisation de projet **Raison** : Ces concepts ont été adoptés et adaptés au contexte des SI, notamment aux particularités de la relation qui lie le commanditaire et le réalisateur du projet

### 7.2 Maîtrise d'Ouvrage (MOA) - LE CLIENT

#### Définition complète

*"D'une manière générale, la maîtrise d'ouvrage d'un projet est constituée par la ou les directions fonctionnelles de l'entreprise qui seront soit commanditaires, soit propriétaires, soit futurs utilisateurs du produit livré à l'issue du projet."*

#### Composition

- Directions fonctionnelles (RH, Finance, Commercial, etc.)
- Direction Générale
- Futurs utilisateurs du système
- Sponsors du projet

#### Rôles et responsabilités

##### 1. Définir le besoin :

- Expression des besoins métiers
- Rédaction du cahier des charges
- Définition des objectifs

## 2. Fixer le cadre :

- Budget alloué
- Planning souhaité
- Périmètre fonctionnel
- Contraintes réglementaires

## 3. Valider les livrables :

- Recette fonctionnelle
- Acceptation des phases
- Validation de la conformité

## 4. Piloter stratégiquement :

- Priorisation des fonctionnalités
- Arbitrages budgétaires
- Décisions d'orientation

## Compétences requises

- Connaissance métier approfondie
- Vision stratégique
- Capacité à formaliser les besoins
- Communication avec utilisateurs finaux

## Livrables typiques MOA

- Cahier des charges fonctionnel
- Spécifications fonctionnelles détaillées
- Scénarios de tests fonctionnels
- Documentation utilisateur

## 7.3 Maîtrise d'Œuvre (MOE) - LE RÉALISATEUR

### Définition AFNOR

*"La maîtrise d'œuvre est la personne physique ou morale qui, pour sa compétence technique, est chargée par le maître de l'ouvrage, ou par la personne responsable du marché, de diriger et de contrôler les travaux et de proposer la réception et leur règlement."*

### Définition appliquée aux SI

*"La maîtrise d'œuvre est responsable de la réalisation des travaux qui lui ont été confiés par la maîtrise d'ouvrage."*

### Composition

- Direction des Systèmes d'Information (DSI)
- Prestataires de services (ESN - Entreprises de Services du Numérique)
- Équipes de développement
- Architectes techniques

### Rôles et responsabilités

#### 1. Concevoir la solution technique :

- Architecture applicative
- Choix technologiques
- Design patterns

#### 2. Réaliser les travaux :

- Développement/codage
- Intégration
- Tests techniques
- Déploiement

#### 3. Proposer des solutions :

- Conseil technique
- Alternatives technologiques
- Optimisations

#### 4. Garantir la qualité :

- Tests unitaires et d'intégration
- Performance

- Sécurité
- Maintenabilité

## Compétences requises

- Expertise technique (langages, frameworks, architectures)
- Méthodologies de développement (Agile, DevOps)
- Gestion de projet technique
- Capacité à traduire besoins fonctionnels en solutions techniques

## Livrables typiques MOE

- Spécifications techniques détaillées
- Code source et documentation technique
- Plans de tests techniques
- Dossier d'architecture technique (DAT)
- Guides d'exploitation

## 7.4 Relation MOA/MOE : Modèle Client-Fournisseur

### Principe fondamental

*"Il nous paraît plus sage de se référer au modèle de base de ce type de relation qui est en fait celui d'une relation de type client-fournisseur."*

### Caractéristiques de la relation

#### MOA = CLIENT

- Exprime le besoin
- Valide le résultat
- Paie la prestation
- Définit le "QUOI"

#### MOE = FOURNISSEUR

- Comprend le besoin
- Propose des solutions
- Réalise la prestation
- Définit le "COMMENT"

### Interfaces de communication

1. **Phase amont :**
  - MOA rédige cahier des charges
  - MOE répond par proposition technique et commerciale
2. **Phase projet :**
  - Réunions de suivi régulières
  - Comités de pilotage (COPIL)
  - Points d'étape et jalons
3. **Phase validation :**
  - Recettes fonctionnelles (MOA)
  - Recettes techniques (MOE)
  - Validation finale conjointe

### Points de vigilance

- **Clarté du besoin** : Cahier des charges précis évite malentendus
- **Contractualisation** : Formaliser engagements de part et d'autre
- **Communication** : Maintenir dialogue constant
- **Gestion du changement** : Procédure pour évolutions du besoin
- **Responsabilités** : Bien délimiter qui fait quoi

## 7.5 Cas particuliers

### AMOA (Assistance à Maîtrise d'Ouvrage)

- Consultant externe aidant la MOA
- Expertise fonctionnelle et méthodologique
- Traduction besoins métiers en spécifications

### AMOE (Assistance à Maîtrise d'Œuvre)

- Expert technique complémentaire
- Renfort sur compétences spécifiques
- Audit et conseil technique

### MOA déléguée

- MOA confie pilotage opérationnel à tiers
- Souvent la DSI fait MOA déléguée pour MOA métier
- Garde responsabilité finale

---

## 8. MÉTHODOLOGIES D'ANALYSE DE PROJET

### 8.1 Analyse SWOT (du Cloud Computing)

#### Forces (Strengths)

- Élasticité et scalabilité
- Réduction CAPEX
- Mises à jour automatiques
- Accessibilité (ATAWADAC)
- Mutualisation des coûts
- Haute disponibilité
- Disaster recovery facilité

#### Faiblesses (Weaknesses)

- Dépendance à Internet
- Dépendance au fournisseur (vendor lock-in)
- Perte de contrôle
- Complexité de migration
- Coûts OPEX récurrents
- Performance parfois inférieure à on-premise

#### Opportunités (Opportunities)

- Innovation facilitée (IA, IoT, Big Data)
- Transformation digitale
- Nouveaux business models
- Time-to-market réduit
- Accès aux dernières technologies
- Expansion géographique facilitée

#### Menaces (Threats)

- Risques de sécurité et confidentialité
- Problématiques juridiques
- Évolution tarifaire imprévisible
- Faillite du fournisseur
- Espionnage industriel
- Non-conformité réglementaire

## 8.2 Analyse PESTEL (du Cloud)

### Politique

- Souveraineté numérique (clouds nationaux)
- Réglementations gouvernementales
- Relations internationales (Cloud Act)
- Politiques de cybersécurité nationale

### Économique

- Modèle OPEX vs CAPEX
- Crise économique → recherche d'économies
- Investissements dans infrastructures
- Compétitivité entre fournisseurs

### Social

- Transformation des métiers IT
- Besoin de formations
- Télétravail et mobilité
- Attentes utilisateurs (ATAWADAC)
- Fracture numérique

### Technologique

- Évolution rapide des technologies
- 5G et amélioration connectivité
- Edge computing
- Informatique quantique
- IA et automatisation

### Environnemental

- Consommation énergétique data centers
- Refroidissement (PUE - Power Usage Effectiveness)
- Green IT et cloud écologique
- Optimisation ressources
- Bilan carbone numérique

### Légal

- RGPD (protection données personnelles)
- Cloud Act américain
- Directive NIS2 (cybersécurité)
- Normes ISO 27001
- Certification HDS (Hébergement Données de Santé)
- Contrats SLA

---

## 9. ÉTUDES D'OPPORTUNITÉ ET LANCEMENT DE PROJET

### 9.1 Phase de questionnement initial

#### Contexte (selon atelier)

- Entreprise : TPE française "TMA Expert" (fictive)
- Spécialité : Tierce Maintenance Applicative
- Problématique : Process de planification RH
- Enjeu : Respect des engagements contractuels clients
- Contrainte : Interlocuteurs débordés

#### Méthodologie : Les 5 questions maximums

**Principe :** "Si vous posez les bonnes questions, vous avez des chances d'obtenir les bonnes réponses !"

### Stratégie de questionnement :

1. **Question ouverte stratégique** : Quel est le problème principal ?
2. **Question de contexte** : Quels sont les processus actuels ?
3. **Question quantitative** : Combien de ressources/contrats/incidents ?
4. **Question d'objectif** : Quel résultat attendu mesurable ?
5. **Question de contrainte** : Quelles limites (budget, délai, technique) ?

### Types de questions à privilégier :

- Questions ouvertes (Quoi, Comment, Pourquoi)
- Questions factuelles (données mesurables)
- Questions orientées solution
- Questions de priorisation

## 9.2 Modèle #1 : Bilan critique de la situation existante

### Structure

#### 1. Description de l'existant

- Processus actuels
- Outils utilisés
- Organisation en place
- Volumétrie (utilisateurs, transactions, données)

#### 2. Analyse des dysfonctionnements

- Problèmes identifiés
- Impacts métiers
- Coûts des dysfonctionnements
- Témoignages utilisateurs

#### 3. Causes racines

- Analyse des causes (5 Pourquoi, diagramme Ishikawa)
- Facteurs organisationnels
- Facteurs techniques
- Facteurs humains

#### 4. Opportunités d'amélioration

- Leviers d'optimisation
- Benchmarks sectoriels
- Bonnes pratiques
- Technologies disponibles

### Exemple appliqué à TMA Expert

#### Existant :

- Planification RH sur Excel partagé
- Communication par emails
- Pas de vision temps réel des disponibilités
- Gestion réactive des absences

#### Dysfonctionnements :

- Conflits d'allocation ressources
- Non-respect délais clients
- Surcharge de certains consultants
- Perte d'informations

#### Causes :

- Manque d'outil centralisé

- Processus non formalisé
- Données dispersées
- Pas d'alertes automatiques

## Opportunités :

- Outil de planification RH dédié
- Automatisation des alertes
- Indicateurs de performance
- Vision consolidée

## 9.3 Modèle #2 : Finalité et objectifs du projet

### Finalité (le "Pourquoi")

#### Vision stratégique :

- Raison d'être du projet
- Alignement avec stratégie entreprise
- Bénéfices attendus à moyen/long terme
- Transformation visée

**Exemple TMA Expert** : "Garantir le respect de nos engagements contractuels en optimisant l'allocation de nos ressources humaines, afin de maintenir notre qualité de service et notre compétitivité."

### Objectifs (le "Quoi")

#### Objectifs SMART (Spécifiques, Mesurables, Atteignables, Réalistes, Temporels)

#### Exemple TMA Expert :

1. **Réduire** de 30% les conflits d'allocation sous 6 mois
2. **Atteindre** 95% de respect des SLA clients sous 1 an
3. **Améliorer** de 50% la visibilité des disponibilités consultants
4. **Automatiser** 80% des alertes de planification
5. **Réduire** de 2h/jour le temps de gestion administrative

### Indicateurs de performance (KPI)

- Taux de respect SLA
- Temps moyen d'allocation ressource
- Nombre de conflits d'allocation
- Satisfaction clients (NPS)
- Taux d'utilisation consultants

### Parties prenantes

- **Sponsors** : Direction Générale
- **Utilisateurs finaux** : Planificateurs RH, managers
- **Bénéficiaires** : Consultants, clients
- **Contributeurs** : DSI, éditeur logiciel

## 9.4 Modèle #3 : Évaluation stratégique (Étude d'opportunité)

### Utilisation des analyses

#### SWOT du projet

- Forces : Quelle expertise interne ? Quelle motivation ?
- Faiblesses : Quelles lacunes ? Quels risques ?
- Opportunités : Quel marché ? Quelles synergies ?
- Menaces : Quelle concurrence ? Quels obstacles ?

#### PESTEL du projet

- Analyser environnement macro du projet
- Identifier facteurs externes impactants
- Anticiper évolutions contextuelles

## Scénarios étudiés

### Scénario 1 : Développement interne

- Avantages : Personnalisation totale, propriété IP
- Inconvénients : Coûts élevés, délais longs, maintenance
- Budget estimé : XXX k€
- Délai : XX mois

### Scénario 2 : Achat solution du marché (SaaS)

- Avantages : Rapidité déploiement, pas de maintenance
- Inconvénients : Abonnement récurrent, moins personnalisé
- Budget estimé : XX k€/an
- Délai : X mois

### Scénario 3 : Solution hybride

- Avantages : Équilibre personnalisation/coûts
- Inconvénients : Complexité intégration
- Budget : XX k€
- Délai : XX mois

## Analyse de faisabilité

### Faisabilité technique

- Technologies disponibles ?
- Compétences en interne ?
- Intégration avec SI existant ?
- Risques techniques ?

### Faisabilité économique

- Budget disponible ?
- ROI (Retour sur Investissement) ?
- Coûts cachés ?
- Financement possible ?

### Faisabilité organisationnelle

- Adhésion des utilisateurs ?
- Ressources humaines mobilisables ?
- Conduite du changement ?
- Impact sur organisation ?

### Faisabilité temporelle

- Délai réaliste ?
- Contraintes calendaires ?
- Disponibilité des ressources ?
- Dépendances critiques ?

## Matrice de décision

Critère	Poids	Scénario 1	Scénario 2	Scénario 3
Coût initial	20%	2/10	8/10	6/10
Délai	25%	3/10	9/10	7/10
Personnalisation	20%	10/10	4/10	7/10
Maintenance	15%	3/10	9/10	6/10
Risques	20%	4/10	7/10	6/10
<b>TOTAL</b>	<b>100%</b>	<b>4,5/10</b>	<b>7,3/10</b>	<b>6,5/10</b>

## Recommandation

Sur la base de cette analyse, le **Scénario 2** (solution SaaS) apparaît comme le plus approprié pour TMA Expert :

- Déploiement rapide
- Coûts maîtrisés
- Maintenance externalisée
- Risques limités

# 10. EXPRESSION DES BESOINS UTILISATEURS

## 10.1 Importance de l'expression des besoins

**Citation clé :** "Si vous posez les bonnes questions, vous avez des chances d'obtenir les bonnes réponses !"

L'expression des besoins est une phase **critique** qui conditionne la réussite du projet :

- Un besoin mal défini = solution inadaptée
- Coûts des modifications exponentiels selon avancement projet
- Satisfaction utilisateurs dépendante de l'adéquation besoin/solution

## 10.2 Méthodes de recueil des besoins

### Entretiens individuels

- **Avantages** : Profondeur, confidentialité, personnalisation
- **Inconvénients** : Chronophage, vision individuelle
- **Quand** : Parties prenantes clés, besoins complexes

### Ateliers collectifs (workshops)

- **Avantages** : Consensus, créativité collective, gain de temps
- **Inconvénients** : Domination de certains participants, organisation complexe
- **Quand** : Définition vision commune, priorisation

### Questionnaires

- **Avantages** : Large audience, quantification, rapidité
- **Inconvénients** : Superficiel, faible taux de réponse
- **Quand** : Phase exploratoire, validation hypothèses

### Observation terrain

- **Avantages** : Réalité des usages, besoins implicites
- **Inconvénients** : Chronophage, effet observateur
- **Quand** : Processus mal connus, innovations

### Analyse documentaire

- **Avantages** : Compréhension contexte, pas d'impact utilisateurs
- **Inconvénients** : Documentation obsolète ou inexistante
- **Quand** : Phase préparatoire, compréhension existant

## 10.3 Techniques de questionnement

### Questions ouvertes

- **Forme** : Quoi, Comment, Pourquoi, Quand, Où, Qui
- **Objectif** : Exploration, compréhension profonde
- **Exemple** : "Comment gérez-vous actuellement la planification ?"

### Questions fermées

- **Forme** : Oui/Non, choix multiples
- **Objectif** : Confirmation, quantification
- **Exemple** : "Utilisez-vous Excel pour la planification ?"

## Questions de relance

- **Forme** : "Pouvez-vous préciser ?", "C'est-à-dire ?"
- **Objectif** : Approfondissement
- **Exemple** : "Vous avez mentionné des problèmes, lesquels précisément ?"

## Questions orientées problème

- **Objectif** : Identifier dysfonctionnements
- **Exemple** : "Quelles sont vos principales difficultés ?"

## Questions orientées solution

- **Objectif** : Recueillir attentes
- **Exemple** : "Comment imagineriez-vous la solution idéale ?"

## 10.4 Priorisation des besoins

### Méthode MoSCoW

#### Must have (Indispensable)

- Fonctionnalités critiques
- Sans elles, le système ne fonctionne pas
- Obligation légale ou réglementaire

#### Should have (Important)

- Fonctionnalités importantes mais non bloquantes
- Contournement temporaire possible
- Forte valeur ajoutée

#### Could have (Confort)

- Fonctionnalités souhaitables
- Améliorent l'expérience
- Non prioritaires

#### Won't have (Hors périmètre)

- Fonctionnalités explicitement exclues
- Reportées à version ultérieure
- Clarification du périmètre

### Matrice Valeur/Effort

	Effort faible	Effort élevé
Valeur élevée	QUICK WINS (priorité 1)	PROJETS MAJEURS (priorité 2)
Valeur faible	NICE TO HAVE (priorité 3)	À ÉVITER (priorité 4)

## 11. CAHIER DES CHARGES FONCTIONNEL (CdCF)

### 11.1 Définition et objectifs

**Définition** : Document contractuel décrivant de manière exhaustive les besoins fonctionnels et les contraintes d'un projet, rédigé par la MOA.

#### Objectifs :

- Formaliser les besoins de manière non ambiguë

- Base contractuelle entre MOA et MOE
- Référence pour validation (recette)
- Communication avec toutes parties prenantes

## 11.2 Structure type d'un CdCF

### 1. Présentation du projet

#### 1.1 Contexte

- Environnement de l'entreprise
- Problématique métier
- Origine du projet
- Historique

#### 1.2 Objectifs

- Objectifs stratégiques
- Objectifs opérationnels
- Résultats attendus (SMART)
- Indicateurs de succès (KPI)

#### 1.3 Périmètre

- Périmètre fonctionnel (inclus/exclus)
- Périmètre organisationnel (services concernés)
- Périmètre géographique
- Périmètre temporel

#### 1.4 Parties prenantes

- Sponsors
- Utilisateurs finaux
- Contributeurs
- Bénéficiaires
- Rôles et responsabilités

## 2. Expression des besoins fonctionnels

### 2.1 Besoins généraux

- Vision d'ensemble de la solution
- Grandes fonctionnalités
- Workflows principaux

### 2.2 Exigences fonctionnelles détaillées Pour chaque fonctionnalité :

- **Identifiant unique** : EXF-001
- **Titre** : Nom explicite de la fonction
- **Description** : Détail du besoin
- **Acteurs** : Qui utilise cette fonction ?
- **Déclencheur** : Qu'est-ce qui initie l'action ?
- **Préconditions** : Conditions nécessaires
- **Scénario nominal** : Déroulement normal
- **Scénarios alternatifs** : Variantes possibles
- **Scénarios d'erreur** : Gestion des erreurs
- **Postconditions** : État résultant
- **Priorité** : Must/Should/Could/Won't
- **Règles de gestion** : Business rules
- **Données manipulées** : Inputs/Outputs

**Exemple - EXF-001 : Créer une planification**

Champ	Valeur
ID	EXF-001
Titre	Créer une planification consultant
Description	Le planificateur doit pouvoir affecter un consultant à un contrat client pour une période donnée
Acteur	Planificateur RH
Déclencheur	Nouvelle demande client ou disponibilité consultant
Priorité	Must have
Règles	- Consultant doit être disponible - Compétences doivent correspondre - Charge max 100% - Respect congés

## 2.3 Cas d'usage (Use Cases)

- Diagrammes de cas d'usage
- Descriptions détaillées
- Acteurs et interactions

## 2.4 Maquettes / Wireframes

- Écrans principaux
- Navigation
- Ergonomie souhaitée
- Charte graphique (si applicable)

## 3. Exigences non fonctionnelles

### 3.1 Performance

- Temps de réponse maximum (ex: < 2 secondes)
- Nombre d'utilisateurs simultanés
- Volume de données
- Disponibilité (ex: 99,9%)

### 3.2 Sécurité

- Authentification (SSO, MFA)
- Autorisation (RBAC - Role Based Access Control)
- Chiffrement des données
- Traçabilité (logs d'audit)
- Conformité (RGPD, ISO 27001)

### 3.3 Ergonomie / UX

- Accessibilité (WCAG)
- Responsive design
- Navigateurs supportés
- Langues supportées
- Formation utilisateurs

### 3.4 Compatibilité / Intégration

- Systèmes à interfaçer
- Formats d'échange (API REST, fichiers)
- Synchronisation temps réel / batch
- Middleware

### 3.5 Maintenabilité

- Documentation technique requise
- Standards de code
- Tests unitaires requis
- Gestion des versions

### 3.6 Évolutivité / Scalabilité

- Croissance anticipée (utilisateurs, données)
- Modularité
- Architecture extensible

### 3.7 Fiabilité

- Taux de disponibilité
- MTBF (Mean Time Between Failures)
- MTTR (Mean Time To Repair)
- Sauvegarde et restauration
- Plan de continuité (PCA/PRA)

## **4. Contraintes**

### **4.1 Contraintes techniques**

- Technologies imposées ou interdites
- Infrastructure existante
- Standards techniques
- Normes de sécurité

### **4.2 Contraintes réglementaires**

- RGPD
- Normes sectorielles (bancaire, santé...)
- Accessibilité numérique
- Archivage légal

### **4.3 Contraintes organisationnelles**

- Disponibilité des utilisateurs
- Périodes interdites (gel des changements)
- Process de validation
- Organisation support

### **4.4 Contraintes budgétaires**

- Enveloppe globale
- Répartition CAPEX/OPEX
- Modalités de paiement

### **4.5 Contraintes temporelles**

- Date de livraison impérative
- Jalons intermédiaires
- Phases de déploiement
- Fenêtres de maintenance

## **5. Livrables attendus**

### **5.1 Livrables applicatifs**

- Application fonctionnelle
- Code source (si prévu)
- Packages d'installation
- Scripts de déploiement

### **5.2 Livrables documentaires**

- Documentation utilisateur
- Documentation administrateur
- Documentation technique (DAT, DEX)
- Supports de formation
- FAQ

### **5.3 Livrables projet**

- Spécifications techniques
- Plans de tests
- Procès-verbaux de recette
- Bilan de projet

## **6. Modalités de recette**

## **6.1 Recette fonctionnelle**

- Qui : MOA
- Quoi : Validation conformité fonctionnelle
- Environnement de recette
- Jeux de données de test
- Scénarios de tests

## **6.2 Critères d'acceptation**

- Taux d'anomalies acceptable
- Classification des anomalies (bloquante, majeure, mineure)
- Conditions de levée de réserves

## **6.3 Procédure de recette**

- Planning de recette
- Participants
- Modalités de reporting
- Procès-verbal de recette

# **7. Modalités de mise en œuvre**

## **7.1 Déploiement**

- Stratégie (big bang, pilote, progressif)
- Planning de déploiement
- Procédure de rollback
- Communication

## **7.2 Formation**

- Public cible
- Contenu formation
- Modalités (présentiel, e-learning)
- Support pédagogiques

## **7.3 Accompagnement au changement**

- Plan de communication
- Support utilisateurs (hotline, FAQ)
- Référents métiers
- Retour d'expérience

# **8. Maintenance et support**

## **8.1 Garantie**

- Durée de garantie
- Périmètre garanti
- Conditions d'intervention

## **8.2 Support**

- Niveaux de support (N1, N2, N3)
- Horaires de support
- Canaux de contact
- Engagements de délai (SLA)

## **8.3 Maintenance évolutive**

- Procédure de demande d'évolution
- Priorisation
- Planning prévisionnel

# **9. Planning et jalons**

## Macro-planning

- Phases du projet
- Jalons majeurs
- Dates clés
- Dépendances critiques

Exemple :

Phase	Début	Fin	Jalon
Spécifications techniques	J+0	J+30	Validation STD
Développement	J+31	J+90	Livraison V1
Recette	J+91	J+120	PV de recette
Déploiement	J+121	J+135	Mise en production
Garantie	J+136	J+226	Fin de garantie

## 10. Budget prévisionnel

### Estimation des coûts

- Licences logicielles
- Développement / paramétrage
- Infrastructure (si applicable)
- Formation
- Accompagnement
- Maintenance annuelle

### Répartition

Poste	Coût (k€)	%
Licences	15	20%
Développement	40	53%
Formation	10	13%
Déploiement	5	7%
Réserve	5	7%
<b>TOTAL</b>	<b>75</b>	<b>100%</b>

## 11. Glossaire

### Définition des termes métiers et techniques

- Acronymes
- Vocabulaire spécifique
- Concepts métiers

Exemple :

- **TMA** : Tierce Maintenance Applicative
- **SLA** : Service Level Agreement (Accord de Niveau de Service)
- **Sprint** : Itération de développement en méthode Agile (2-4 semaines)

## 12. Annexes

- Règles de gestion détaillées
- Modèles de données
- Matrices de flux
- Études préalables (SWOT, PESTEL)
- Comptes-rendus d'ateliers
- Benchmarks

### 11.3 Bonnes pratiques de rédaction

#### Clarté et précision

- Phrase courte et simple

- Un besoin = une phrase
- Éviter ambiguïtés ("si possible", "idéalement")
- Utiliser vocabulaire métier

## Vérifiabilité

- Chaque exigence doit être testable
- Critères d'acceptation mesurables
- "Le système doit..." plutôt que "Le système pourrait..."

## Traçabilité

- Identifiants uniques pour chaque exigence
- Possibilité de suivre depuis besoin jusqu'à test
- Versioning du document

## Complétude

- Tous les besoins exprimés
- Tous les cas (nominal + alternatifs + erreurs)
- Toutes les contraintes identifiées

## Cohérence

- Pas de contradiction entre exigences
- Terminologie homogène
- Niveaux de priorité logiques

## Pièges à éviter

- ✗ Spécifier la solution technique (rôle MOE)
- ✗ Être trop vague ou trop détaillé
- ✗ Oublier les exigences non fonctionnelles
- ✗ Ne pas impliquer les utilisateurs finaux
- ✗ Ne pas prioriser les besoins

# 12. MÉTHODOLOGIES DE GESTION DE PROJET

## 12.1 Cycle en V (traditionnel)

### Principe

Modèle séquentiel où chaque phase de conception correspond à une phase de validation

### Phases descendantes (spécification)

1. **Analyse des besoins** → Tests d'acceptation
2. **Spécifications générales** → Tests d'intégration
3. **Spécifications détaillées** → Tests unitaires
4. **Codage** (point bas du V)

### Phases montantes (réalisation/validation)

5. **Tests unitaires**
6. **Tests d'intégration**
7. **Tests système**
8. **Recette / Tests d'acceptation**

### Avantages

- Cadre structuré et rigoureux

- Documentation exhaustive
- Chaque phase validée avant passage suivante
- Adapté projets à risque élevé (aéronautique, médical)

## Inconvénients

- Rigidité (changements coûteux)
- Effet tunnel (livraison tardive)
- Tests en fin de projet
- Nécessite besoins stables et bien définis

## Quand l'utiliser

- Projet avec exigences stables
- Domaines réglementés
- Équipes expérimentées en V
- Faible incertitude technique

## 12.2 Méthodes Agiles

### Manifeste Agile (4 valeurs)

1. **Individus et interactions** > processus et outils
2. **Logiciel fonctionnel** > documentation exhaustive
3. **Collaboration avec client** > négociation contractuelle
4. **Adaptation au changement** > suivi d'un plan

### 12 principes Agiles (sélection)

- Satisfaction client par livraison rapide et continue
- Accueillir changements même tard dans développement
- Livrer fréquemment (semaines plutôt que mois)
- Collaboration quotidienne métier/développeurs
- Face-à-face pour transmettre informations
- Logiciel fonctionnel = mesure principale d'avancement
- Rythme soutenable pour durabilité
- Excellence technique et bonne conception
- Simplicité (maximiser travail non fait)
- Équipes auto-organisées
- Réflexion régulière pour s'améliorer

## 12.3 Scrum (framework Agile)

### Rôles

#### Product Owner (PO)

- Représente utilisateurs/clients
- Définit et priorise Product Backlog
- Valide les incrément
- Maximise valeur produit

#### Scrum Master (SM)

- Facilitateur
- Garant de la méthode Scrum
- Supprime obstacles
- Coach de l'équipe

#### Development Team (Équipe de développement)

- 3-9 personnes
- Pluridisciplinaire
- Auto-organisée
- Responsable collectivement de livraison

## Artefacts

### Product Backlog

- Liste priorisée de toutes fonctionnalités souhaitées
- Géré par Product Owner
- Évolutif (raffinement continu)
- Items sous forme de User Stories

### Sprint Backlog

- Sous-ensemble Product Backlog pour le Sprint
- Engagement de l'équipe
- Découpé en tâches

### Incrément

- Somme de tous items terminés dans Sprint
- État "potentiellement livrable"
- Définition of Done (DoD) respectée

### Événements (Time-boxed)

#### Sprint (1-4 semaines, généralement 2)

- Itération de développement
- Durée fixe
- Objectif de Sprint défini

#### Sprint Planning (8h pour Sprint de 1 mois)

- Début de Sprint
- Définition objectif Sprint
- Sélection items Product Backlog
- Plan de réalisation

#### Daily Scrum / Stand-up (15 min quotidien)

- Synchronisation équipe
- 3 questions par personne :
  - Qu'ai-je fait hier ?
  - Que vais-je faire aujourd'hui ?
  - Quels obstacles ?

#### Sprint Review (4h pour Sprint de 1 mois)

- Démonstration incrément aux parties prenantes
- Feedback
- Adaptation Product Backlog

#### Sprint Retrospective (3h pour Sprint de 1 mois)

- Amélioration continue
- Ce qui a bien fonctionné
- Ce qui peut être amélioré
- Plan d'action

### User Stories

**Format :** En tant que [rôle], je veux [fonctionnalité], afin de [bénéfice]

**Exemple :** "En tant que planificateur RH, je veux consulter les disponibilités des consultants sur un calendrier, afin d'affecter rapidement la bonne personne aux demandes clients."

**Critères INVEST (bonne User Story) :**

- Independent (indépendante)
- Negotiable (négociable)

- Valuable (valeur pour utilisateur)
- Estimable (estimable en effort)
- Small (petite)
- Testable (testable)

### Definition of Done (DoD) :

- Code écrit et reviewé
- Tests unitaires passés
- Tests d'intégration OK
- Documentation à jour
- Déployé en environnement de recette
- Validé par PO

### Avantages Scrum

- Adaptabilité au changement
- Feedback rapide et régulier
- Transparence et visibilité
- Engagement équipe
- Livraisons fréquentes de valeur
- Amélioration continue

### Inconvénients Scrum

- Nécessite discipline et rigueur
- Engagement fort Product Owner
- Peut être difficile avec clients traditionnels
- Estimation complexe au début
- Risque de "Scrum but" (Scrum superficiel)

### Quand utiliser Scrum

- Projet avec besoins évolutifs
- Innovation, R&D
- Product development
- Équipe colocalisée (idéal)
- Client impliqué disponible

## 12.4 Comparaison Cycle en V vs Agile

Critère	Cycle en V	Agile/Scrum
Approche	Séquentielle	Itérative
Flexibilité	Faible	Élevée
Documentation	Exhaustive	Juste nécessaire
Implication client	Début et fin	Continue
Livrasons	Unique (fin projet)	Fréquentes (chaque Sprint)
Gestion changement	Coûteuse	Intégrée
Visibilité	Tardive	Permanente
Risques	Détectés tard	Détectés tôt
Adapté si	Besoins stables	Besoins évolutifs
Prédicibilité	Élevée (délai/coût)	Variable

## 12.5 Approche hybride

De nombreuses organisations adoptent approches hybrides combinant :

- Cadrage et architecture en mode V
- Développement et livraisons en mode Agile
- Exemple : SAFe (Scaled Agile Framework)

### Critères de choix :

- Nature du projet
- Culture de l'organisation

- Contraintes réglementaires
- Maturité de l'équipe
- Degré d'incertitude

## 13. SYNTHÈSE ET POINTS CLÉS POUR LE TEST

### Définitions essentielles à maîtriser

#### Service vs Produit

- Service = co-création valeur, immatériel, sans transfert coûts/risques
- Produit = ressources configurées, valeur intrinsèque

#### IaaS / PaaS / SaaS

- IaaS = Infrastructure (serveurs, stockage)
- PaaS = Plateforme (environnement développement)
- SaaS = Software (application complète)

#### Cloud Computing

- Accès services informatiques via Internet depuis fournisseur
- Modèle OPEX (location) vs modèle traditionnel CAPEX (achat)

#### MOA / MOE

- MOA = Client, définit besoin (QUOI), valide
- MOE = Fournisseur, réalise solution (COMMENT), livre
- Relation client-fournisseur inspirée du bâtiment

#### ATAWADAC

- Anytime, Anywhere, Any Device, Any Content
- Attente utilisateur moderne d'accessibilité totale

### 4 risques cloud à connaître

1. **Propriété données** : À qui appartiennent-elles ?
2. **Localisation données** : Où sont-elles hébergées ?
3. **Juridiction** : Quelle loi s'applique ?
4. **Réversibilité** : Peut-on changer de fournisseur facilement ?

### Analyses stratégiques

**SWOT** : Forces, Faiblesses, Opportunités, Menaces     **PESTEL** : Politique, Économique, Social, Technologique, Environnemental, Légal

### 3 modèles d'études de projet

1. **Bilan critique existant** : Processus actuels, dysfonctionnements, causes, opportunités
2. **Finalité et objectifs** : Vision stratégique, objectifs SMART, KPI, parties prenantes
3. **Étude d'opportunité** : SWOT/PESTEL, scénarios, faisabilité, recommandation

### Cahier des Charges Fonctionnel (structure)

1. Présentation projet (contexte, objectifs, périmètre, parties prenantes)
2. Besoins fonctionnels (exigences détaillées, cas d'usage, maquettes)
3. Exigences non fonctionnelles (performance, sécurité, ergonomie, compatibilité)
4. Contraintes (techniques, réglementaires, budgétaires, temporelles)
5. Livrables, recette, mise en œuvre
6. Maintenance, planning, budget, glossaire

## Méthodologies projet

**Cycle en V** : Séquentiel, rigoureux, adapté besoins stables, livraison unique  **Scrum/Agile** : Itératif, flexible, adapté besoins évolutifs, livraisons fréquentes

## Formules et concepts à retenir

**Objectifs SMART** : Spécifiques, Mesurables, Atteignables, Réalistes, Temporels

**MoSCoW** : Must have, Should have, Could have, Won't have

**User Story** : En tant que [rôle], je veux [fonctionnalité], afin de [bénéfice]

**SLA** : Service Level Agreement (engagement disponibilité/performance)

**CAPEX** : Capital Expenditure (investissement) **OPEX** : Operational Expenditure (fonctionnement)

## Tendances actuelles

- IA/ML, Big Data, Micro-services, IoT → demandeurs en cloud
- Informatique quantique → cloud obligatoire (températures extrêmes)
- Architectures hybrides (mix local/cloud) → réalité des entreprises
- Infogérance (années 90) → Cloud (aujourd'hui)

---

## CONSEILS POUR LE TEST DE 2H

### Organisation

1. **Lire toutes les questions** (10 min) : Identifiez celles que vous maîtrisez le mieux
2. **Commencez par les questions sûres** : Sécurisez les points acquis
3. **Gérez votre temps** : 2h = 120 min. Si 10 questions, ~12 min/question
4. **Soignez la présentation** : Titres, paragraphes, schémas si pertinent

### Réponse aux questions

#### Définitions :

- Définition précise
- Caractéristiques clés
- Exemple concret
- Distinction avec concepts voisins si pertinent

#### Questions de comparaison (ex: MOA vs MOE) :

- Tableau comparatif si possible
- Points communs puis différences
- Exemples pour chaque

#### Questions d'analyse (ex: SWOT, PESTEL) :

- Structure claire (4 quadrants SWOT, 6 dimensions PESTEL)
- 3-4 éléments minimum par catégorie
- Justification de chaque élément

#### Cas pratiques (ex: rédiger partie CdCF) :

- Respectez structure attendue
- Soyez concret et précis
- Utilisez vocabulaire