

|  |  |  |
| --- | --- | --- |
| *Le futur se dessine aujourd’hui* | | |
| Modèle de Cahier  Des Charges  Technique | | |
| Version V01  Direction Technique  Buildingmap | | |
| Résultat de recherche d'images pour "icon orange mail" | contact@futurmap.com | Date d’édition :  mercredi 3 septembre 2025 | |
| Résultat de recherche d'images pour "icon orange website" | www.futurmap.com |

# **Information Générales**

**Titre du projet :**

**Demandeur(s) :**

**Date de la demande :**

**Date limite souhaitée :**

**Version du document : 1.0**

# Objet et portée

* **Objet** : Présenter l’objet du document

Exemple : Ce document spécifie l’architecture, les exigences techniques, l’implémentation et les critères d’acceptation pour l’outil d’**automatisation de la dé-densification de nuages de points.**

* **Périmètre fonctionnel :** rappelle des principales fonctionnalités dans le CDCF

Exemple :   
- Interface graphique pour sélectionner type d’entrée (fichier/dossier) et sélectionner les fichiers/dossiers.  
- Paramétrage des options DeCap

1. **Hypothèses et contraintes**

Enoncer les conditions de fonctionnement du projet et les limites imposé (système d’exploitation, licence, plateforme, compétences de l’utilisateur…)

Exemple : L’outil s’exécute sous Windows avec DeCap.exe disponible sur la machine.

# Exigences techniques

## Exigences fonctionnelles détaillées (TR)

Lister ce que le système doit faire

Exemple : RF-01 : Parcourir et valider récursivement les dossiers pour détecter les fichiers d’entrée valides)

## Exigences non fonctionnelles (NFR)

Lister les caractéristiques du système (Robustesse, sécurité, interpolabilité…)

Exemple : RNF-01 : Performance – capacité à traiter par lots de 50 fichiers.

# Architecture et composants

## Choix technologiques recommandés

Préciser le ou les technologies retenues (langage, framework UI, gestion des processus, stockage, logging, packaging) et la raison du choix (interop Windows, performance, maintenance). Indiquer aussi une alternative si nécessaire.

Exemples :

- Langage / Framework principal : C# / Python

- Stockage et configuration : fichier JSON et profils enrégistrés dans AppData

## Schéma d’architecture (composants)

Présenter les grands blocs logiciels de l’application, leurs rôles et leurs interactions. Un schéma permettant de visualiser la séparation UI / moteur / exécution / stockage.

Exemple :

* **UI (WPF)** : gestion des profils, sélection des fichiers/dossiers, monitoring d’exécution, affichage des logs, tableau d’état par fichier.
* **Core Engine** : orchestration des traitements, validation d’entrée, génération de commandes DeCap, gestion des threads/queue.

# Description DU MODULE UI

Présenter la structure de l’interface graphique, les composants et les wireframes s’il y en a.

Exemple :

* Écran principal : sélection type d’entrée, zone de dépôt (drag & drop), liste des fichiers détectés.
* Pane paramètres : champs pour chaque option DeCap (champ numérique, dropdown pour unités/systèmes de coordonnées, checkbox pour booléens).
* Profil management : créer/enregistrer/supprimer/import/export profil.
* Console log temps réel + niveau de détail (info/debug/error).

# Tests et validation

## Tests unitaires

Vérifier le bon fonctionnement isolé des modules et fonctions (sans dépendances externes).

*Exemple* : tester le builder de commande DeCap (chaîne générée correcte), parser JSON des profils, parseur de sortie stdout simulée.

## Tests d’intégration

Vérifier que les modules fonctionnent ensemble, en interaction réelle avec DeCap.exe et fichiers.

*Exemple* : lancer un traitement sur un dossier avec 3 fichiers .e57 → vérifier que 3 sorties RCP sont générées et importables.

## Tests de validation utilisateur (UAT)

Vérifier en conditions réelles que le logiciel répond bien au besoin utilisateur final.

*Exemple* : un préparateur exécute un profil complet, obtient un RCP, l’importe dans Revit 2025 sans erreur.

# Critères d’acceptation (AC)

Conditions qui valident officiellement que le projet est réussi. Elles traduisent les besoins en résultats concrets et observables.

Exemple :

* AC-01 : L’application démarre et affiche l’UI sans erreur.
* AC-02 : Un profil peut être créé, sauvegardé et rechargé correctement.

# Annexes

Lien de documentation

Méthodes et fonctions réutilisables

Algorithmes

…

Exemples :

## Exemple de profil JSON (schématique)

{

"profileName": "Profil\_ProjetA",

"inputType": "folder",

"inputPaths": ["C:\\scans\\projetA"],

"minRange": 0.5,

"maxRange": 50,

"decimation": 5,

"unify": false,

"normaliseIntensity": true,

"currentCoordinateSystem": "EPSG:4326",

"targetCoordinateSystem": "EPSG:2154",

"inputUnitType": "mm",

"e57EnableImageImport": true,

"e57CreateScanPerImage": false,

"parallelJobs": 2,

"retryOnFail": 2,

"timeoutPerJobSeconds": 7200

}

## Exemple de ligne de commande (schématique)

DeCap.exe -i "C:\\scans\\projetA\\scan1.e57" -o "C:\\output\\projetA\_scan1.rdp" --decimation 5 --minRange 0.5 --maxRan