

# Séance 7 – Préparer l'industrialisation du projet

## Objectif de la séance

L'objectif de cette séance est de **préparer l'industrialisation du projet fil rouge**, sans entrer dans l'outillage technique (Maven, CI/CD, scripts). Vous allez travailler **au niveau architectural et organisationnel**, afin de définir une structure claire, cohérente et prête à être industrialisée lors des séances suivantes.

À l'issue de cette séance, vous devez être capable de répondre clairement à la question :

*« Si je devais industrialiser ce projet demain, que faudrait-il builder, dans quel ordre, et selon quelles règles ? »*

## Pourquoi cette séance est importante

Avant d'utiliser un outil de build, il est indispensable de **savoir ce que l'on construit**.

Un outil comme Maven :

- n'invente pas l'architecture,
- ne corrige pas un mauvais découpage,
- ne clarifie pas des dépendances floues.

Il ne fait que **formaliser** des choix déjà faits.

Cette séance sert donc à **figer ces choix**, pour éviter que le build devienne un problème technique au lieu d'un simple automatisme.

## Ce que vous ne ferez pas pendant cette séance

Il est important d'être très clair sur le périmètre :

- vous **ne codez pas**,
- vous **n'écrivez pas de fichiers Maven**,
- vous **ne configurez aucun outil**.

Le travail attendu est un **travail de conception et de structuration**, directement lié à votre projet fil rouge.

## Partie 1 – Découpage final du projet en modules

Vous devez identifier les **modules définitifs** de votre projet.

Un module correspond à :

- une responsabilité claire,
- un périmètre fonctionnel ou technique cohérent,
- une stabilité dans le temps.

Pour chaque module, vous devez préciser :

- son rôle principal,
- ce qu'il contient,
- ce qu'il ne doit jamais contenir.

Exemples (à adapter à votre projet) :

- module-domain : règles métier, entités, invariants
- module-application : orchestration des cas d'usage
- module-infrastructure : persistance, accès technique
- module-api : points d'entrée, contrôleurs

## Partie 2 – Dépendances entre modules

Une fois les modules identifiés, vous devez expliciter **les dépendances entre eux**.

Pour chaque module, indiquez :

- de quels autres modules il dépend,
- pourquoi cette dépendance est nécessaire.

Règles à respecter :

- pas de dépendance circulaire,
- le métier (domain) ne dépend de rien,
- l'infrastructure dépend du reste, jamais l'inverse.

Vous pouvez présenter ce travail sous forme :

- de schéma simple,
- ou de liste hiérarchisée commentée.

## Partie 3 – Arborescence cible du projet

Vous devez maintenant proposer **l'arborescence logique finale** de votre projet.

Il ne s'agit pas d'une arborescence Maven réelle, mais d'une **projection** de ce que sera le projet industrialisé.

Exemple de forme attendue :

project-root

- domain
- application
- infrastructure
- api
- documentation

Chaque dossier doit correspondre à un module identifié précédemment.

## Partie 4 – Règles de versioning

Vous devez définir des règles simples de versioning pour votre projet.

Le format attendu est le versioning sémantique :

- MAJOR.MINOR.PATCH

Vous devez préciser :

- la version actuelle du projet (ex. 0.1.0),
- quand incrémenter PATCH,
- quand incrémenter MINOR,
- ce qui constituerait un MAJOR.

L'objectif n'est pas la précision extrême, mais la **cohérence et la justification**.

## Livrable attendu

Vous devez rendre **un document unique** (PDF, Markdown ou Word) contenant :

1. La liste des modules avec leur rôle
2. Les dépendances entre modules
3. L'arborescence cible du projet
4. Les règles de versioning définies

Aucun code n'est attendu.

## Lien avec la suite du module

Le travail réalisé pendant cette séance servira **directement de base** à la prochaine séance consacrée à l'industrialisation réelle du build.

Les choix que vous figez aujourd'hui seront ceux que l'outil de build viendra simplement traduire.

Prenez donc le temps de privilégier la clarté, la cohérence et la justification de vos décisions.