

# **Guide Scrum**

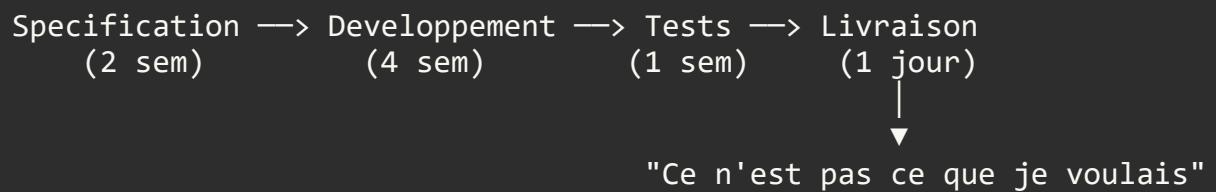
## **Projet Web - Mobilite Internationale**

Methodologie agile pour livrer de la valeur de maniere iterative

# Pourquoi Scrum ?

---

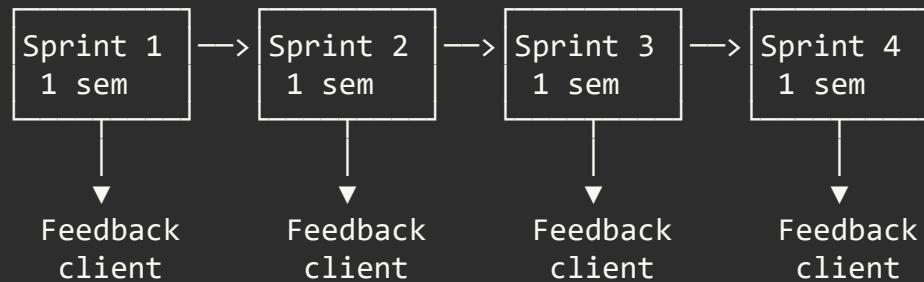
## Le probleme de la methode en cascade



Le client decouvre le resultat **7 semaines** apres le debut.

# L'approche Scrum

---



Le client voit le produit évoluer **chaque semaine**.

# Les 3 piliers de Scrum

---

Pilier	Description
Transparence	Tout le monde voit l'avancement
Inspection	On vérifie régulièrement le travail
Adaptation	On ajuste selon les retours

# Les roles Scrum

---

Product Owner → Définit le QUOI

Scrum Master → Facilite le COMMENT travailler

Developpeurs → Définit le COMMENT techniquement

# Organisation pour vos groupes

---

## Equipe de 3 personnes

Role	Responsabilite	Cumul
Product Owner	Priorites, validation	+ Dev
Scrum Master	Rituels, blocages	+ Dev
Developpeur	Realisation	Principal

# Organisation pour vos groupes

---

## Equipe de 2 personnes

Role	Responsabilite
PO + Developpeur	Priorites et code
SM + Developpeur	Rituels et code

Les deux personnes partagent les responsabilités.

# Le Product Owner (PO)

---

Représente le client - Définit le QUOI

Responsabilités :

- Réddiger et prioriser les User Stories
- Définir les critères d'acceptation
- Valider les livrables à chaque fin de sprint
- Arbitrer les choix fonctionnels

# Le Product Owner - Exemples

---

Dans votre projet :

- Decide quelles pages developper en priorite
- Valide que "Postuler a une offre" repond au besoin
- Tranche : recherche ou favoris d'abord ?

# Le Scrum Master

---

## Garant de la méthode - Facilite le travail

Responsabilités :

- Animer les rituels (planning, daily, review, retro)
- Protéger l'équipe des perturbations
- Lever les blocages
- S'assurer du respect des engagements

# Le Scrum Master - Exemples

---

Dans votre projet :

- Organise le daily de 10 minutes
- Aide sur une config Apache bloquante
- Rappelle : pas de changement de scope en cours de sprint

# L'équipe de développement

---

## Realise le travail - Définit le COMMENT technique

Responsabilités :

- Estimer la complexité des tâches
- S'auto-organiser pour l'objectif du sprint
- Produire un increment fonctionnel et testé
- Respecter les standards (PSR-12, W3C)

## L'équipe - Exemples

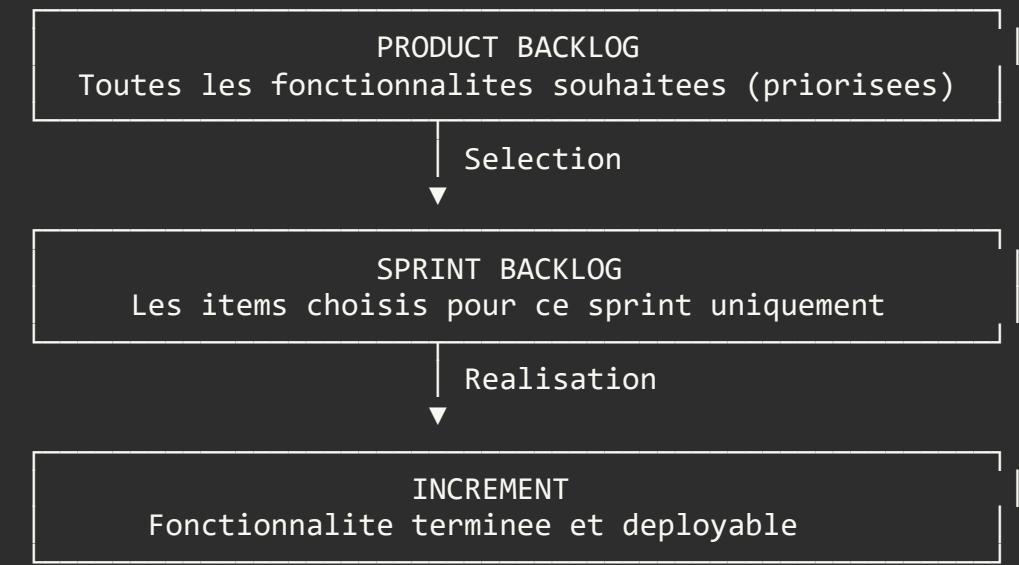
---

Dans votre projet :

- Decide d'utiliser Twig comme moteur de template
- Repartit : "Tu fais le front, je fais le back"
- S'engage à livrer 3 User Stories ce sprint

# Les artefacts Scrum

---



# Le Product Backlog

---

Liste ordonnee de tout ce qui doit etre fait

Structure d'une User Story :

```
En tant que [role utilisateur]  
Je veux [fonctionnalite]  
Afin de [benefice]
```

# Exemple de User Story

---

En tant qu'étudiant

Je veux rechercher des opportunités par pays  
Afin de trouver rapidement une destination

Critères d'acceptation :

- [ ] Filtre par pays fonctionnel
- [ ] Pagination (10 résultats/page)
- [ ] Message si aucun résultat

# Le Sprint Backlog

---

## Decomposition en taches techniques

User Story	Tache	Estimation
Recherche par pays	Formulaire HTML	2h
	Controleur PHP	3h
	Requete SQL	1h
	Pagination	2h
	Tests	2h

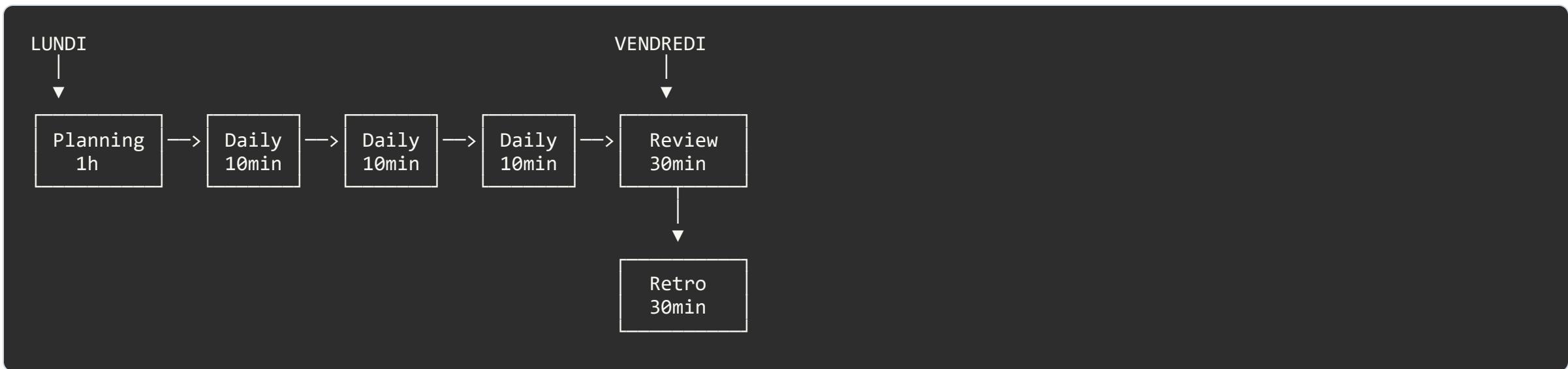
# Definition of Done (DoD)

---

Quand une fonctionnalite est-elle "terminee" ?

- Code valide W3C (HTML)
- Code conforme PSR-12 (PHP)
- Fonctionnalite testee manuellement
- Responsive design verifie
- Code pousse sur Git

# Les evenements Scrum



# Sprint Planning

---

**Objectif :** Definir ce qu'on va realiser

Deroulement (1h) :

1. Le PO presente les items prioritaires (15 min)
2. L'équipe pose des questions (15 min)
3. L'équipe estime et selectionne (20 min)
4. Décomposition en tâches (10 min)

# Sprint Planning - Resultat

---

## Objectif de sprint clair et mesurable

Exemples :

- "Un etudiant peut s'inscrire et se connecter"
- "La liste des opportunites est affichee avec pagination"
- "Un etudiant peut postuler avec CV"

# Daily Scrum (Stand-up)

---

**Objectif :** Synchroniser et identifier les blocages

Chaque membre répond à 3 questions :

Question	Exemple
Hier ?	"Formulaire d'inscription terminé"
Aujourd'hui ?	"Validation côté serveur"
Blocage ?	"Config sessions PHP..."

## Daily Scrum - Regles

---

- **15 minutes maximum**
- Debout (pour rester concis)
- Pas de resolution de probleme pendant
- Les discussions techniques : apres, en apart

# Sprint Review (Demo)

---

**Objectif :** Montrer le travail et recueillir du feedback

Déroulement (30 min) :

1. Demo live des fonctionnalités (20 min)
2. Le PO valide ou invalide (5 min)
3. Discussion sur la suite (5 min)

## Sprint Review - Conseils

---

- Preparer un scenario realiste

*"Je suis un etudiant qui cherche une mobilite..."*

- Montrer le site fonctionnel, pas des slides
- Accepter les retours sans se justifier

# Sprint Retrospective

---

**Objectif :** Ameliorer le fonctionnement de l'équipe

Keep	Stop	Start
Ce qui a marche	Ce qu'on arrete	Ce qu'on commence
"Bonne répartition"	"Coder sans sync"	"Code reviews"

# Sprint Retrospective - Action

---

Choisir **1 a 2 améliorations** concrètes

Exemples :

- "On fera un point rapide à 14h en plus du daily"
- "On utilisera des branches Git pour chaque feature"
- "On testera sur mobile avant de valider"

# Tableau Kanban

---

A FAIRE	EN COURS	A TESTER	TERMINE
Story 4 Story 5	Story 2 (Alice) Story 3 (Bob)	Story 1	

# A vous de jouer

---

Lundi 09 fevrier - Sprint 1

1. Attribuer les roles
2. Creer le Product Backlog
3. Sprint Planning (1h)
4. C'est parti !