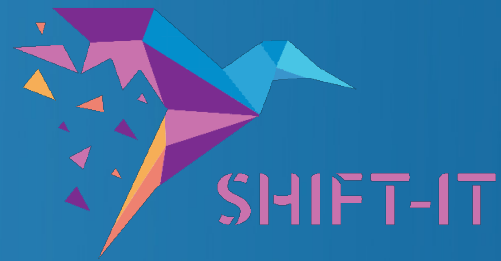
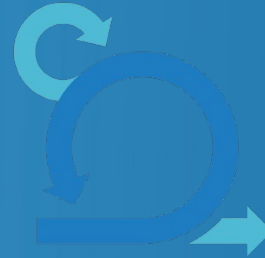


Rémy BROUSSET  
remy.brousset@aquilenet.fr



# Introduction à SCRUM

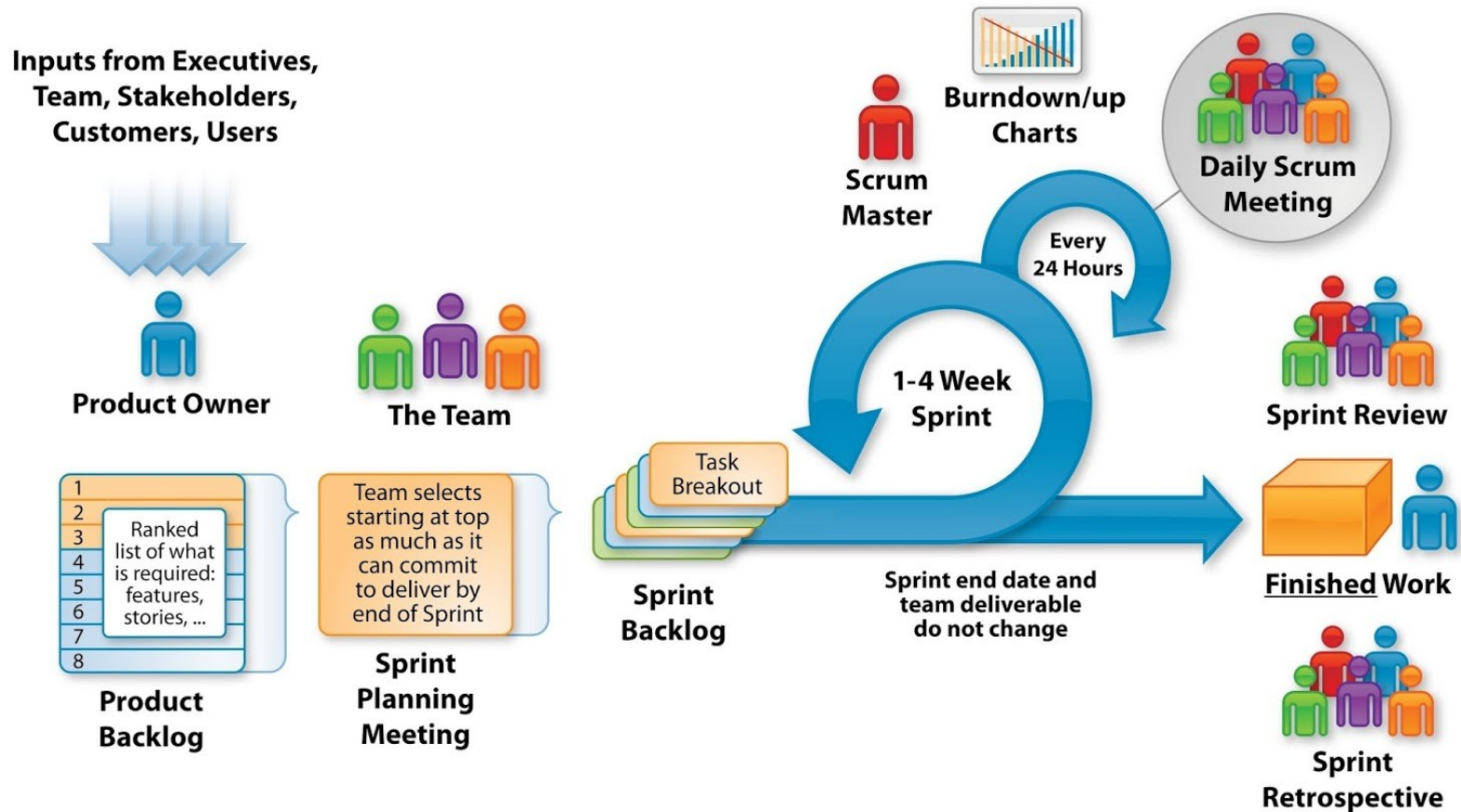




# SCRUM

- 1986 : Nouvelle approche qui augmenterait la vitesse et la flexibilité dans le développement de nouveaux produits.
- 1995 : Le terme Scrum est introduit par Ken Schwaber et provient du rugby (Mêlée).
- 2001 : Méthode décrite dans le livre « Agile Software Development With Scrum » (Ken Schwaber & Mike Beedle) .
- 2010 : **Guide Scrum** (Jeff Sutherland & Ken Schwaber).
  - Dernière révision **2020**.

# SCRUM





# SCRUM

## ■ Des principes

- Série de Sprint où l'on conçoit, développe et teste la solution
- Fonctionnement empirique
- Equipe auto-organisée<sup>e</sup>

## ■ Des rôles

- Product Owner,
- Scrum Master,
- Development Team

## ■ Des cérémonies

- Daily Scrum
- Sprint planning
- Sprint review
- Retrospective

## ■ Des artifacts

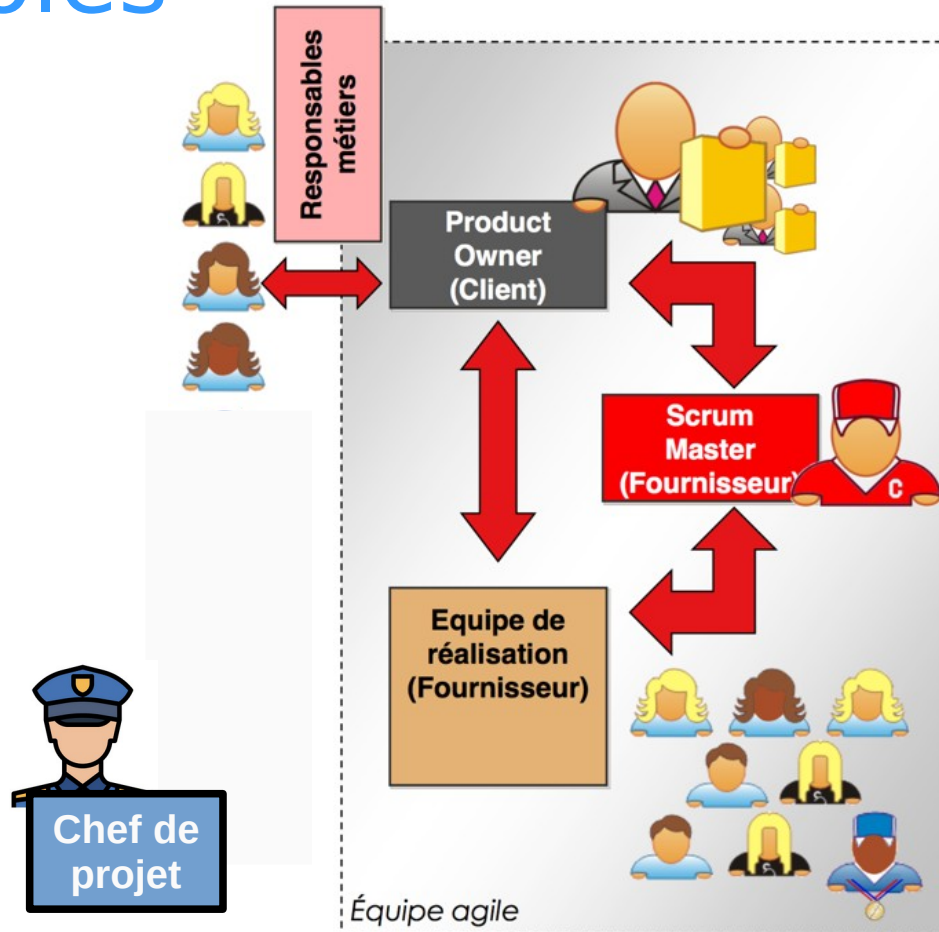
- Product backlog
- Sprint backlog
- Burndown charts



# Principes

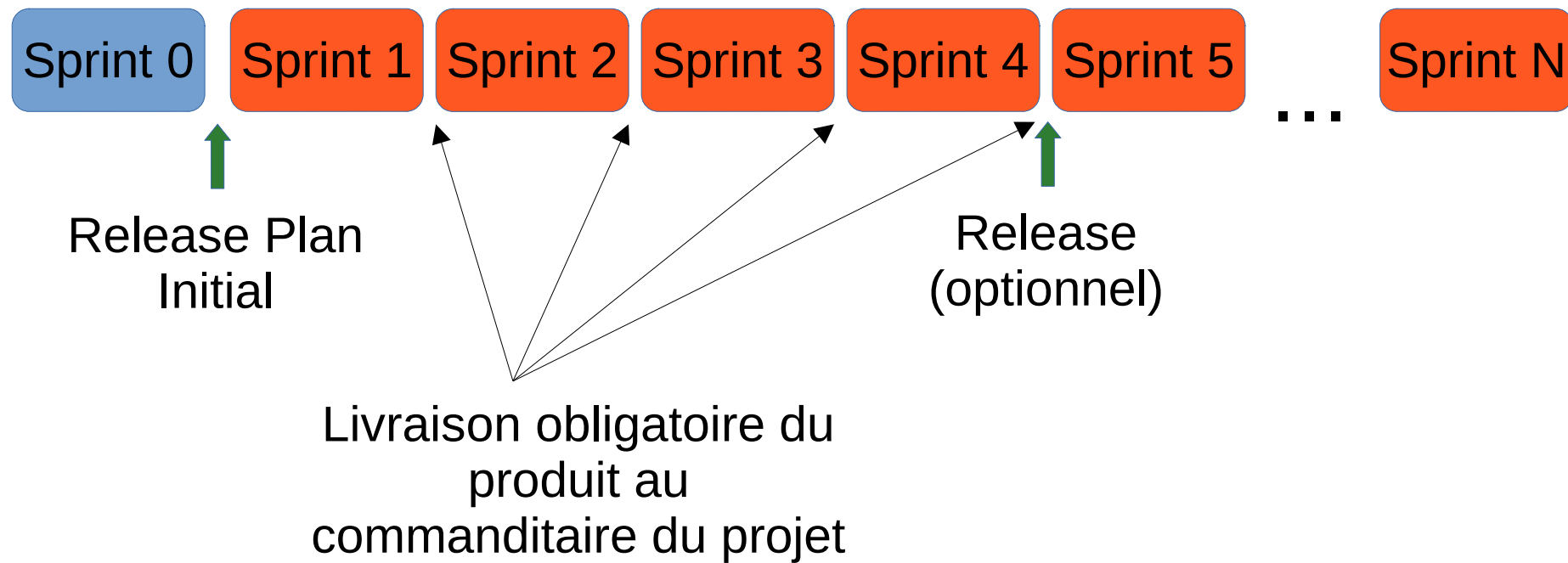
- Processus simples (**3 rôles**, **4 cérémonies**)
- Focalisé sur l'organisation
- Succession de Sprints (itérations de 2 à 4 semaines)
- Produit fonctionnel (même partiellement) à chaque fin de sprint
- Avancement quotidien
- Participation active du client nécessaire
- Chaque Cérémonie est **timeboxée**
  - La durée est connue en avance (spécifique)
  - On peut mandater un « **Gardien du temps** » (optionnel)
  - On peut ajuster les durées lors des rétrospectives

# Les rôles



- Product Owner
  - Définit le besoin
  - Priorise le besoin
  - Valide les livraisons
  - **N'est pas un supérieur !**
- Scrum Master
  - Garant de la méthode
  - Protège et soutient la team
  - **N'est pas le chef de projet !**
- Team
  - 5 à 10 pers.
  - Analyse et estime
  - Réalise
  - Auto-organisée
  - Fixe ses propres règles

# Planification





# Sprint

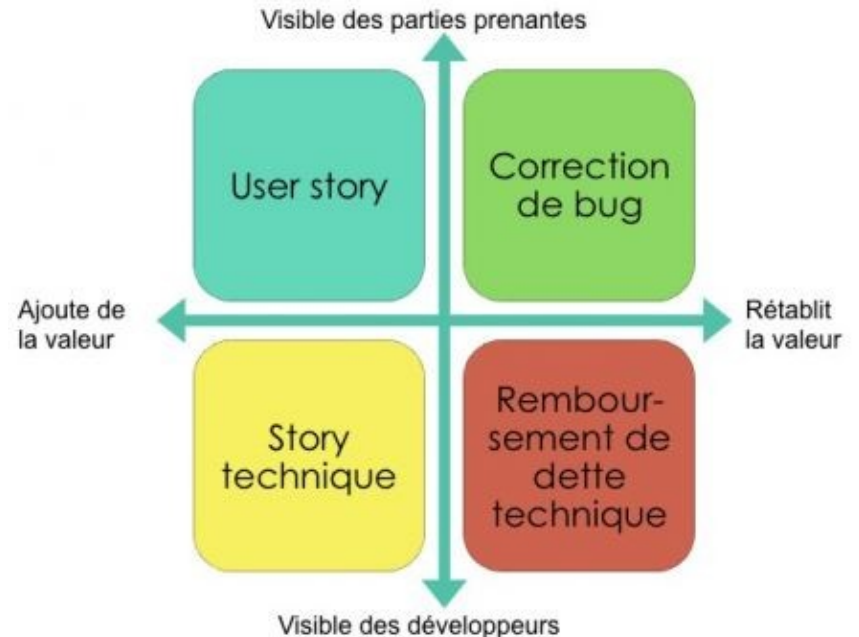
- **Itération de 2 à 4 semaines**
- Comprend suffisamment de User Stories pour occuper la Team pendant toute la période prévue.
- Les User Stories terminées et validées **sont livrées en production dès la fin du sprint** pour utilisation immédiate.
- La quantité de travail faisable (appelée la **capacité**) est calculée à partir de la **vélocité** de l'équipe (productivité moyenne des deux ou trois sprints précédents)

**Capacité** (en PE) = **Vélocité** (en PE/j) x **durée** du sprint (en j)



# User Story

- Unité de base de Scrum
- Courte phrase décrivant l'ajout de valeur pour le client
- Rédigée par le Product Owner
- Généralement fonctionnelle et nominale
- Plus rarement (à éviter) :
  - Bug (ou Defect Story)
  - Tache Technique
  - Refactoring

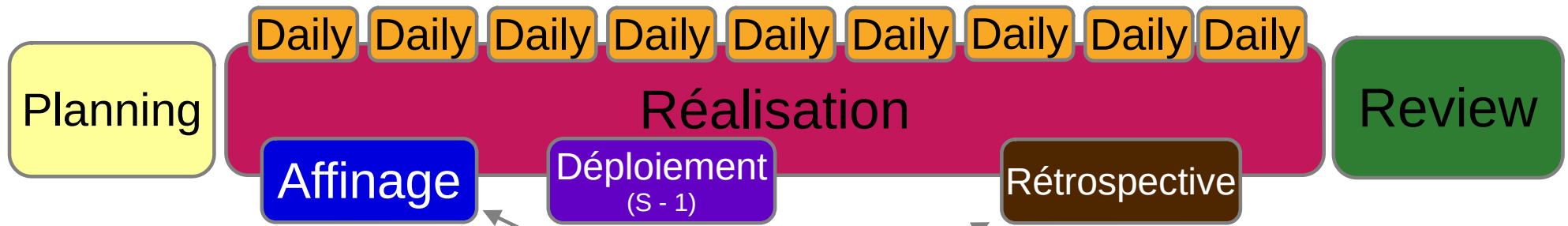




# Product Backlog

- Ensemble des besoins métier, exprimés sous la forme d'**une liste de User Stories** (et/ou Epics), **estimés** et **priorisés**.
- Fait au début du projet...
- ... mais évolue en permanence !
- Responsabilité du Product Owner
- Souvent produit à l'issue de séances de Product Vision, Personas, User Journeys, Story Mapping, et Refinement...
  - <https://www.infoq.com/fr/presentations/art-maniement-exigences-agiles>

# Anatomie d'un sprint



*Optionnel –  
Recommandé, mais pas  
« officiellement » SCRUM*

Placés  
librement !



# Sprint 0

- Mise en place de l'environnement
- Teste la méthode
- Environ 1 semaine
  - Le temps nécessaire à la réalisation de quelques US simples.
- Permet de mesurer la **vélocité** initiale (PE/j)
- Permet le Release Plan Initial
- On « teste » le DoD
- ***La vélocité d'une équipe ne se calcule pas, elle se mesure !***



# Sprint 0

- A l'issue du 1<sup>er</sup> sprint du projet (sprint 0) :
- Le SM calcule la vélocité initiale (nombre de PE réalisés / nombre de jours consacrés)
- On planifie les durées des prochains sprints (fonction du calendrier et des disponibilités de la team)
- On en déduit la capacité (en PE) de chaque sprint :  
$$\text{Capacité} = \text{Vélocité} \times \text{Durée du sprint}$$

# Résumé !

```
productBacklog = po.createBacklog()
devs.estimate(productBacklog)
po.priorize(productBacklog)
velocity = devs.sprint0()
sm.releasePlan(velocity,productBacklog)
while(client.isHappy()){
    let sprintBacklog = sm.sprintPlanning(po,devs,productBacklog)
    for(let day = 1; day <= 15; day++){
        sm.dailyMeeting(po,devs,sprintBacklog)
        feature = devs.dev(sprintBacklog)
        if(po.validate(feature)){
            product += feature
        }
    }
    sm.sprintReview(po,devs,client,sprintBacklog)
    ops.deliver(product)
    productBacklog = po.refineBacklog(sm,devs)
    sm.retrospective(po,devs)
}
```

A decorative graphic in the top-left corner consisting of a cluster of overlapping squares in various shades of blue, some with small white dots.

MERCI ! ...