

Gestion des exigences dans le développement agile de logiciels pour dispositifs médicaux

Julien Zaegel - Stago

Présentations

- Développement agile pour l'industrie minière et le web
- 6 ans chez Stago à introduire les méthodes agiles et à les adapter au contexte médical
- AgileMedDev.com
- <https://fr.linkedin.com/in/julienzaegel>



Le problème

- Les méthodes agiles sont puissantes
- Mais contradictoires avec l'ingénierie des exigences et la philosophie qui va avec

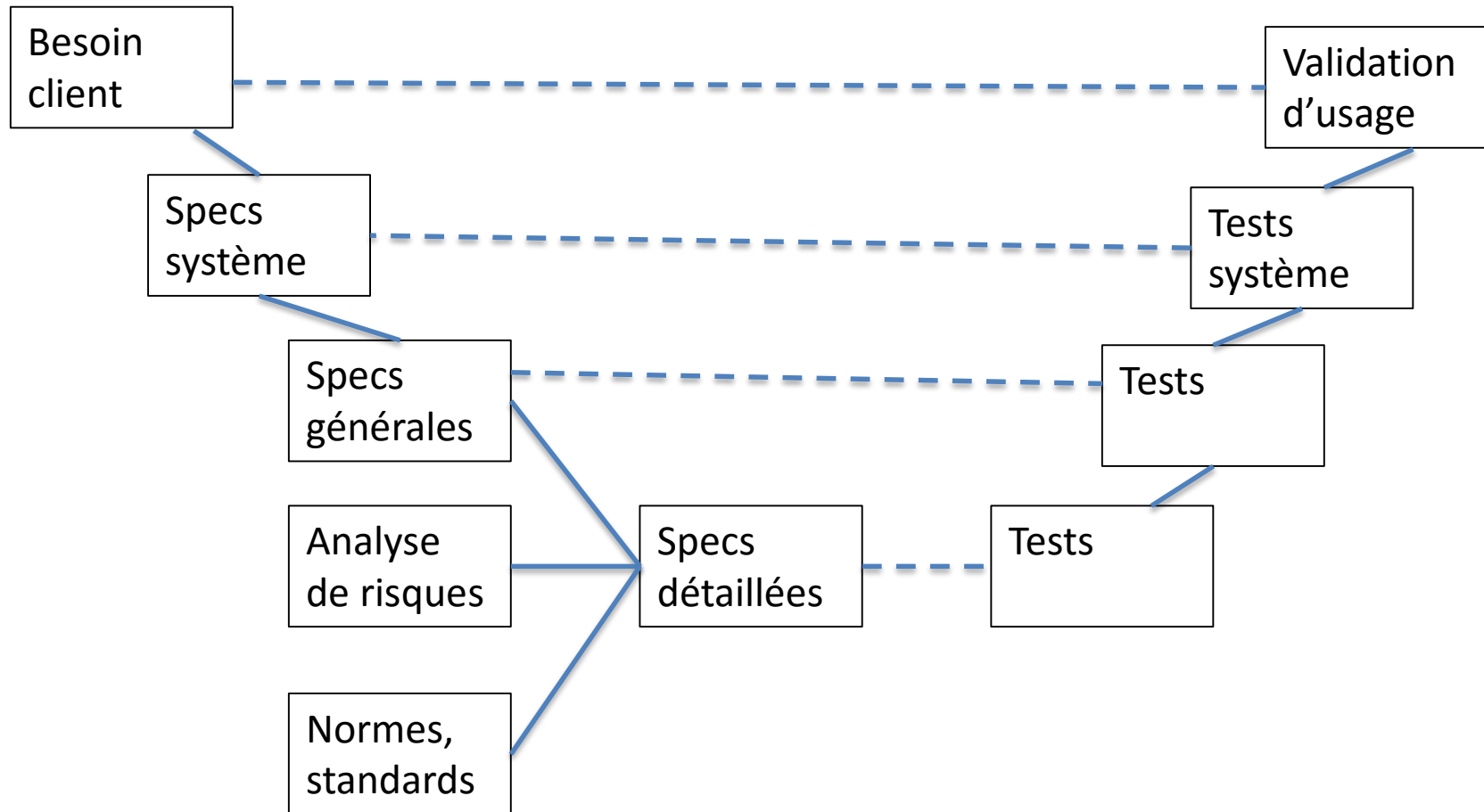


- 
- Ingénierie des exigences pour dispositifs médicaux
 - Analyse du besoin pour les méthodologies agiles
 - Unification des deux univers

Ingénierie des exigences pour les dispositifs médicaux: pourquoi ?

- Parce que c'est obligatoire
- Parce que cela permet la rigueur (matrice de traçabilité)
- Pour maîtriser les risques
- Pour donner des points d'ancrage aux disciplines à temps de cycle long (hardware)
- Pour documenter les comportements du système
- Pour anticiper les problèmes
- Pour faciliter le test

Décomposition du besoin et traçabilité



Exemple de spécification

2.1.2 Fermer une session


PS_ST_000295-321	Le système doit permettre à tout opérateur identifié de fermer sa session manuellement.
PS_ST_000295-345	Le système doit rendre possible la fermeture manuelle de session depuis n'importe quelle interface.
PS_ST_000295-322	Le système doit fermer une session automatiquement au bout d'une durée paramétrable.
PS_ST_000295-336	Le système doit permettre de paramétrer un tempo d'inactivité opérateur au delà de laquelle la session est fermée automatiquement. Ce tempo est configurable parmi la liste suivante: 5 mn, 10 mn, 15 mn, 30 mn, 45 mn, 1h, 2h, 3h, 4h, 8h, 24h.
PS_ST_000295-360	Le système doit proposer par défaut une durée de fermeture automatique de 10 minutes.
PS_ST_000295-346	Que la session ait été fermée manuellement ou automatiquement, le système doit poursuivre tout événement en cours : - analyses (patients, CQ, calibration) en cours et présentés dans la liste de travail, - maintenances automatique, - chargement automatique d'échantillons, - remontées des alertes/erreurs.
PS_ST_000295-431	Toutes modifications du système effectuées pendant une session doivent être visibles pour les sessions suivantes quel que soit l'utilisateur. <i>Note: quelques exemples, changement de langue, méthodologie, demande d'analyse</i>
PS_ST_000295-347	Le système doit rendre accessible uniquement l'interface dashboard lorsqu'une session est fermée.

Dangers

- Dangers de la discipline: la tentation du cycle en V
 - Sur-ingénierie de la spécification abstraite parfaite
 - Retarde le feedback des activités aval
 - Retarde l'intégration système et bio (toujours risqués)
 - Retarde le feedback client
 - Rend difficile la prise en compte de nouveaux besoins
 - La spécification abstraite parfaite contractualisée
 - Avancement réel inconnu - effet tunnel



Plan

- 
- Ingénierie des exigences pour dispositifs médicaux
 - **Analyse du besoin pour les méthodologies agiles**
 - Unification des deux univers

Le manifeste agile

Manifeste pour le développement Agile de logiciels

Nous découvrons comment mieux développer des logiciels par la pratique et en aidant les autres à le faire.
Ces expériences nous ont amenés à valoriser :

Les individus et leurs interactions plus que les processus et les outils

Des logiciels opérationnels plus qu'une documentation exhaustive

La collaboration avec les clients plus que la négociation contractuelle

L'adaptation au changement plus que le suivi d'un plan

Nous reconnaissons la valeur des seconds éléments,
mais privilégions les premiers.

SCRUM

Product Owner



Scrum Master



Daily Scrum
15min

Daily

Sprint
1-4weeks



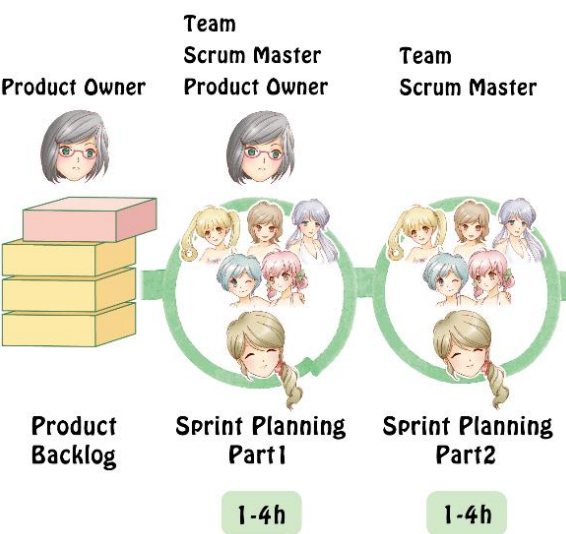
Product Backlog

Product Owner

Team Scrum Master
Product Owner

Sprint Planning Part 1
1-4h

Sprint Planning Part 2
1-4h



Sprint Backlog

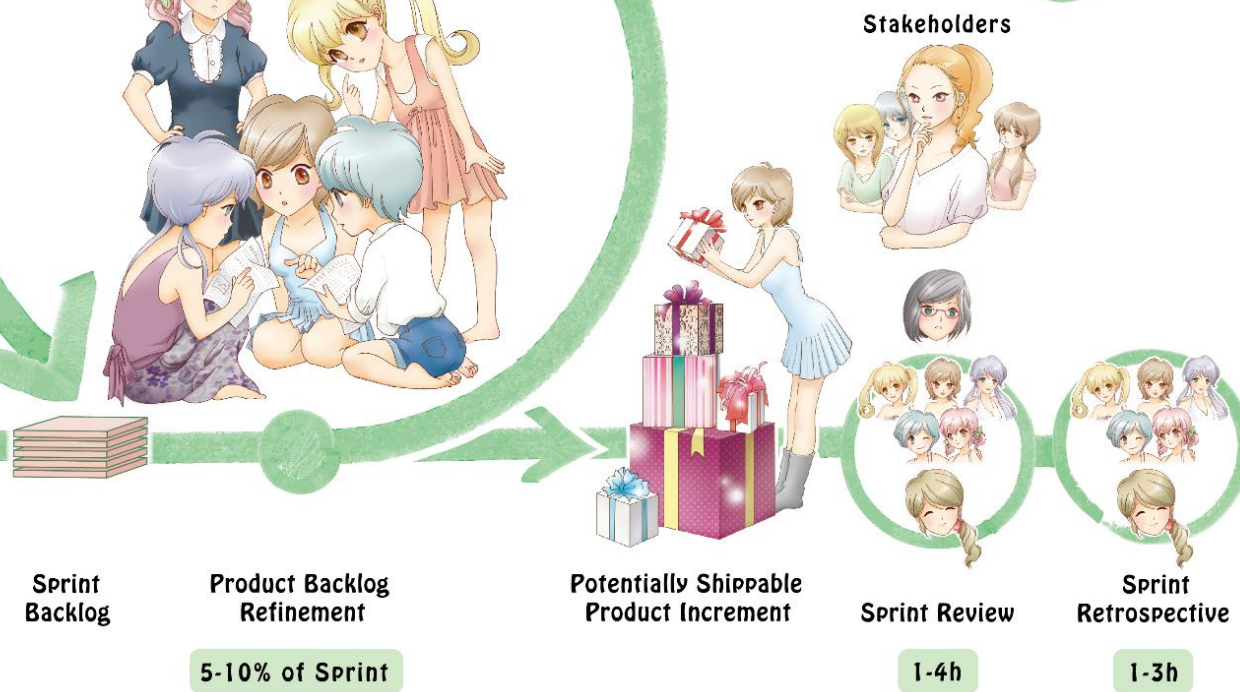
Product Backlog Refinement
5-10% of Sprint

Potentially Shippable Product Increment

Stakeholders

Sprint Review
1-4h

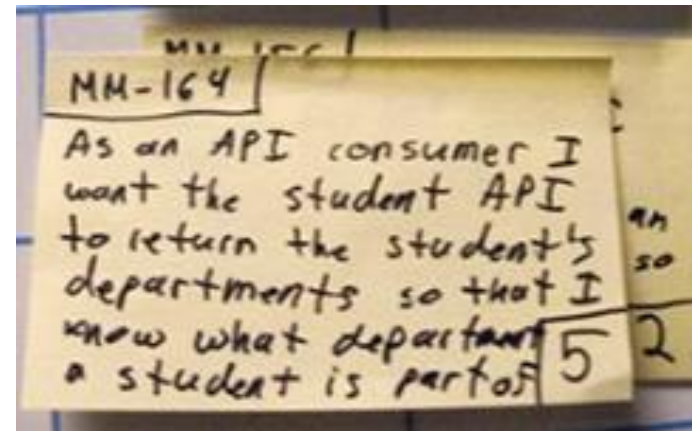
Sprint Retrospective
1-3h



L'analyse du besoin avec les User Stories

- Template: en tant que <qui>, je veux <quoi> afin de <pourquoi>
- Exemple: En tant qu'utilisateur connecté à l'application, je veux fermer ma session manuellement afin que les actions suivantes effectuées dans le logiciel ne me soient pas imputées dans la traçabilité

- Le formalisme importe peu
- N'importe qui peut écrire une user story
- Peut être précisée par un critère d'acceptation
- Ce qui compte n'est pas le texte, mais la discussion

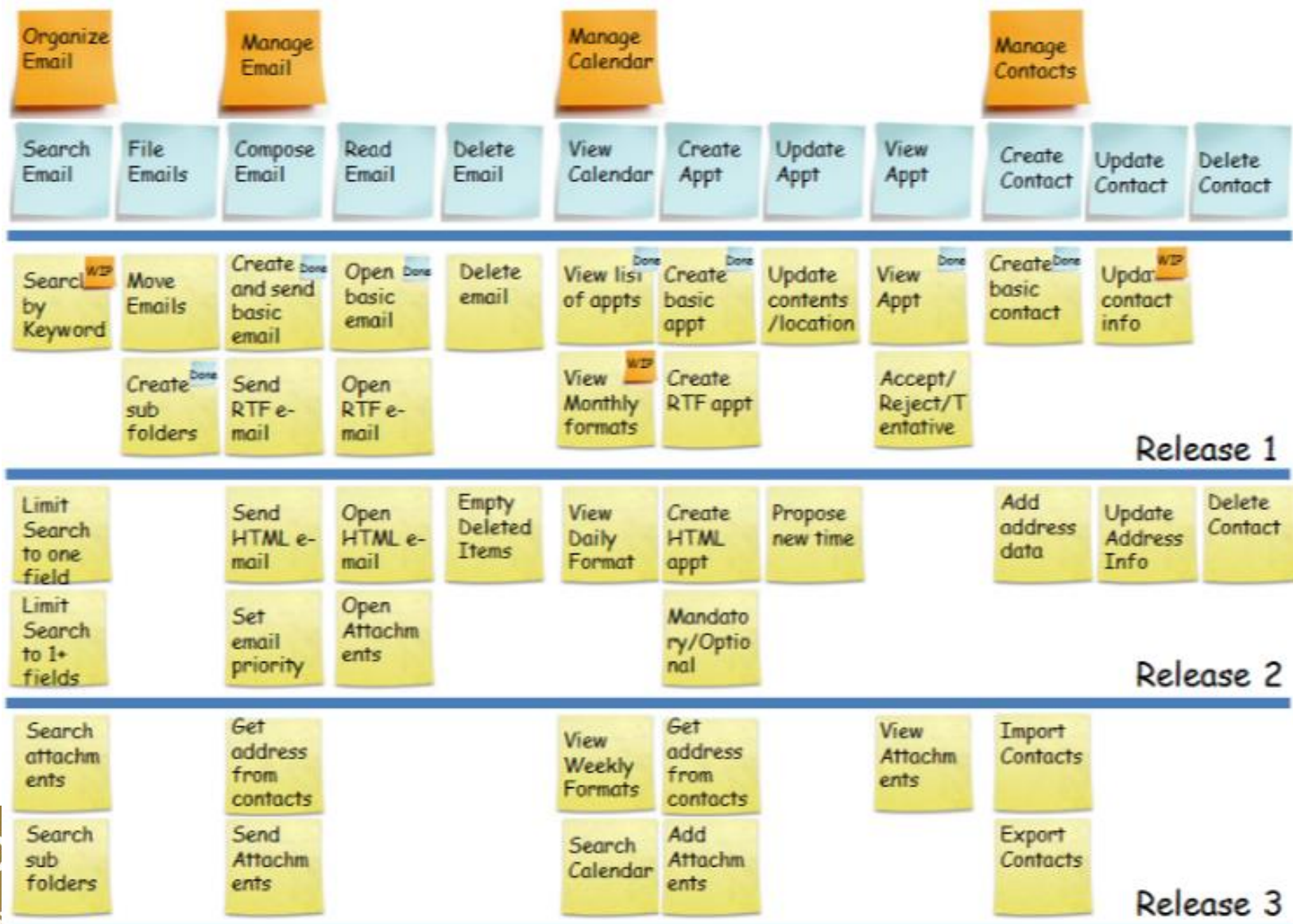


Le rôle du Product Owner

- Responsabilités:
 - Avoir une vision de ce que doit être le produit
 - Transmettre cette vision à l'équipe
 - Maintenir le Product Backlog et la Product Map
 - Gère le budget et le temps, arbitre la taille du produit
- Expert du domaine et du marché
- Disponible à plein temps pour l'équipe

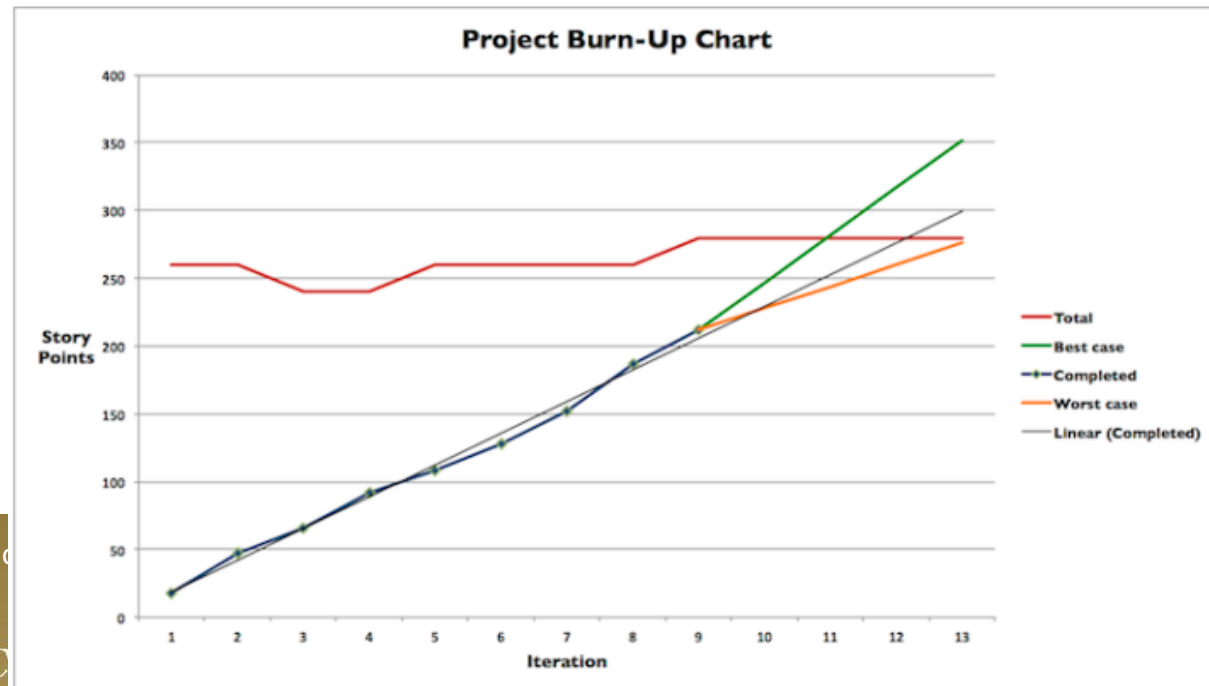


Product Map



Intérêts de la méthode

- Opportunités d'intégration
- Retours rapides des clients sur le produit
- Gestion des risques projet et produit
- Contrôle sur le déroulé du projet (stop ou encore)
- Motivation
- Productivité
- Prédicibilité



Dangers

- Risque sur la pérennité des savoirs
- Rétro-doc: activités vidées de leur sens
- Points d'ancrage difficiles à garantir pour la gestion des risques et le système
- Estimation de coûts et de temps pour un projet global optionnelles
- Risque réglementaire



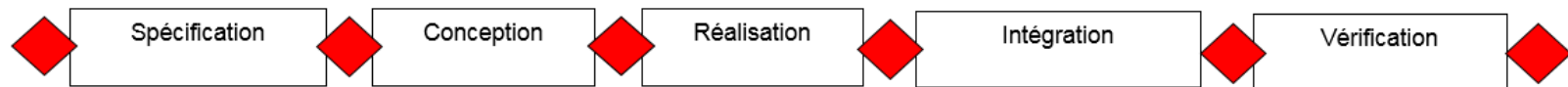
Plan

- 
- Ingénierie des exigences pour dispositifs médicaux
 - Analyse du besoin pour les méthodologies agiles
 - **Unification des deux univers**

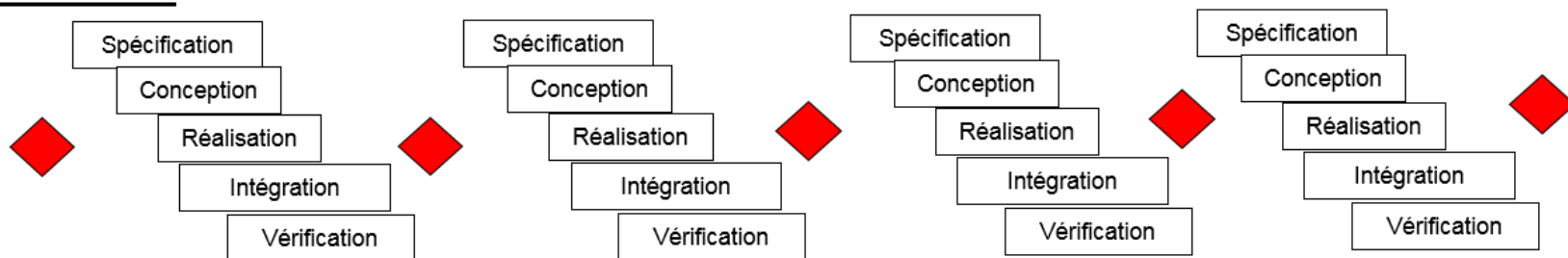
Principe de l'unification

- Adoption du cycle de développement agile avec la documentation dans la « Definition of Done »
- Jalons d'itération, pas de phase

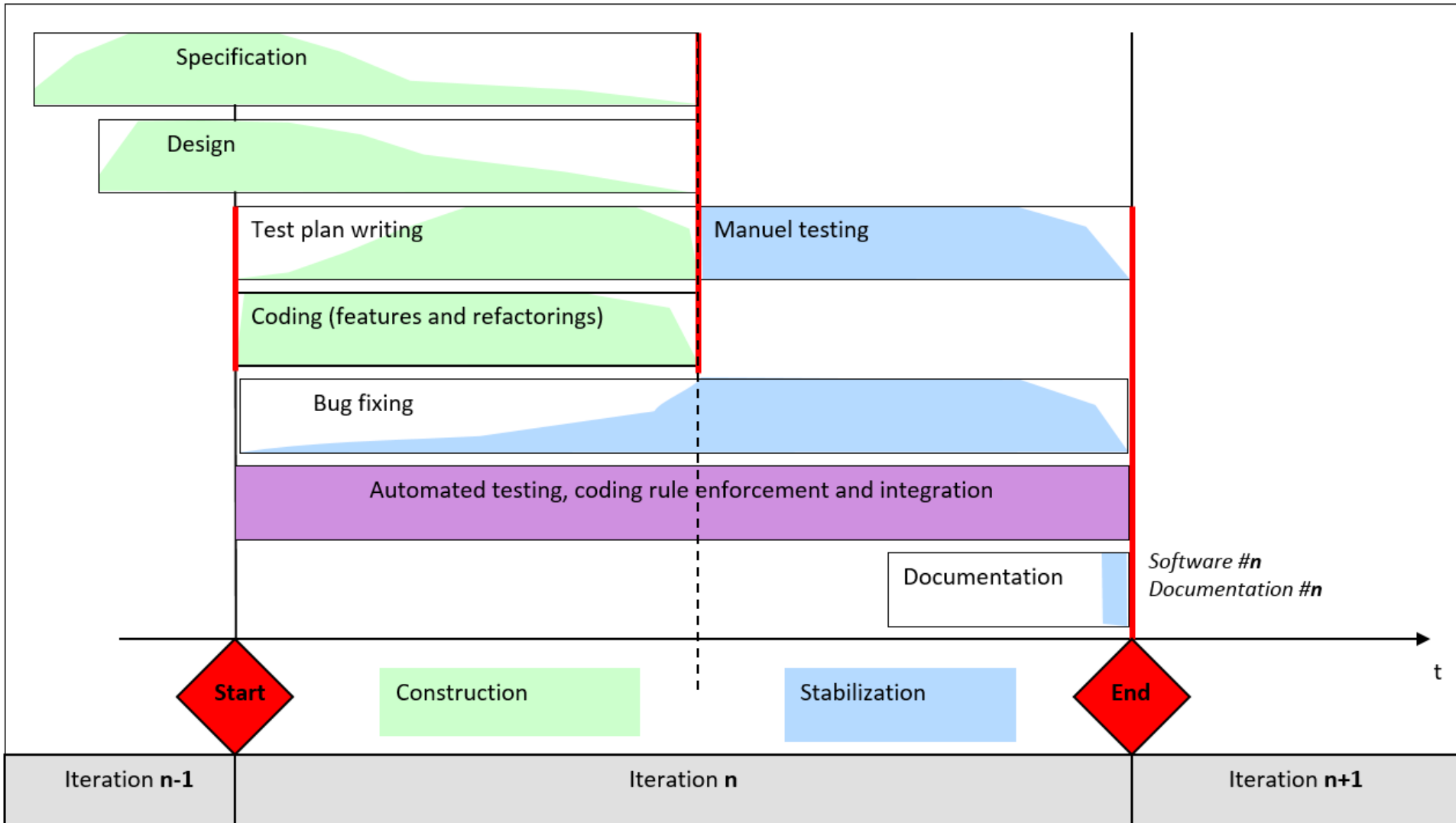
Cycle en V:



Cycle itératif:



Le cycle agile réglementarisé



Points d'attention

- User stories et exigences
 - Product Map initial
 - Spec détaillée au fur et à mesure du projet, avec légère avance sur le dév
 - Lien entre user stories et exigences
- Automatiser la production de documentation d'itération au maximum
 - Toutes les specs, plan de tests et analyses de risques dans des outils
 - Rapports de tests automatique
 - Génération de matrices de traçabilité

Exemple d'application monde réel

- 4000 exigences
- 3000 tests automatiques
- Analyse de risques, cybersécurité incluse
- 20 000 jours homme
- 30+ itérations de logiciel intégrable

Pour aller plus loin

- <https://www.mountangoatsoftware.com/agile/scrum/product-owner>
- <http://blog.palo-it.com/2014/02/11/rediger-un-user-story-les-bonnes-pratiques/>
- <https://www.mountangoatsoftware.com/agile/user-stories>
- User Stories Applied: For Agile Software Development (livre de Mike Cohn)
- Agile Estimating and Planning (livre de Mike Cohn)
- AAMI TIR45: 2012 - Technical Information Report Guidance on the use of AGILE practices in the development of medical device software
- Harvard Business Review, Embracing agile
<https://hbr.org/2016/05/embracing-agile>

Questions?

