

Annexes

Ci-après vous trouverez une série de liens et de slides supplémentaires sur Scrum et Agile.

Si vous avez des questions, si vous désirez des références sur un sujet en particulier, n'hésitez pas à me contacter: bruno.sbille@gmail.com

Vous pouvez également consulter cette page: <http://brunosbille.com/?p=1980> avec des liens sur Scrum et Agile. Pensez à changer de langue pour avoir les liens en anglais également

Le Standish Group a étudié plus de 40.000 projets en 10 ans.

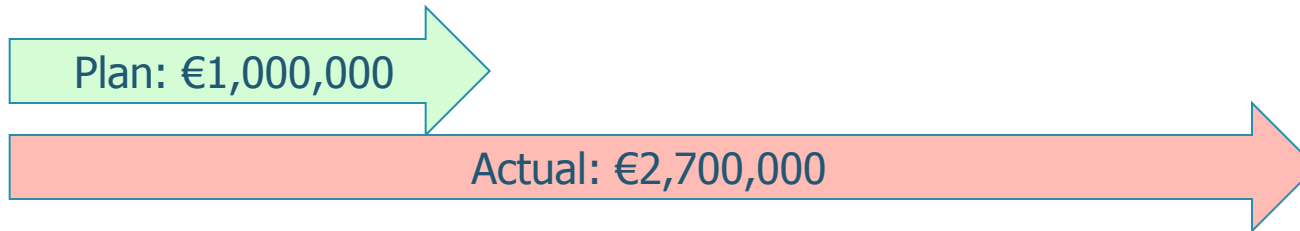
QUELQUES ENSEIGNEMENTS DE CES ÉTUDES

La plupart des projets IT échouent

Le Standish Group a étudié plus de 40.000 projets en 10 ans.

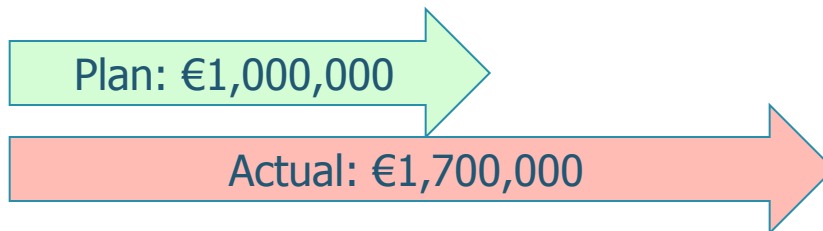
Taux de réussite des projets IT en 1994: **15%**

Dépassement moyen (coût et temps) : **≈170%**



Taux de réussite des projets IT en 2004: **34%**

Dépassement moyen (coût et temps) : **≈70%**



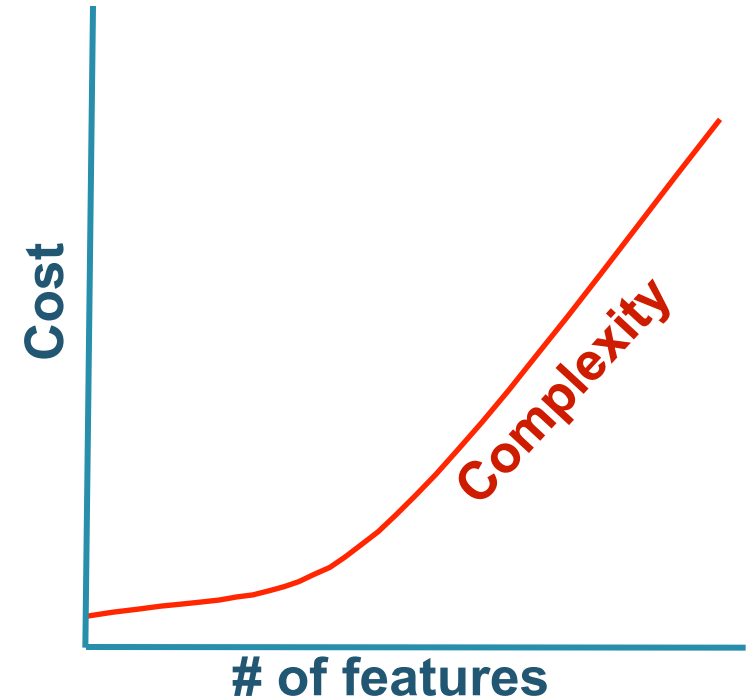
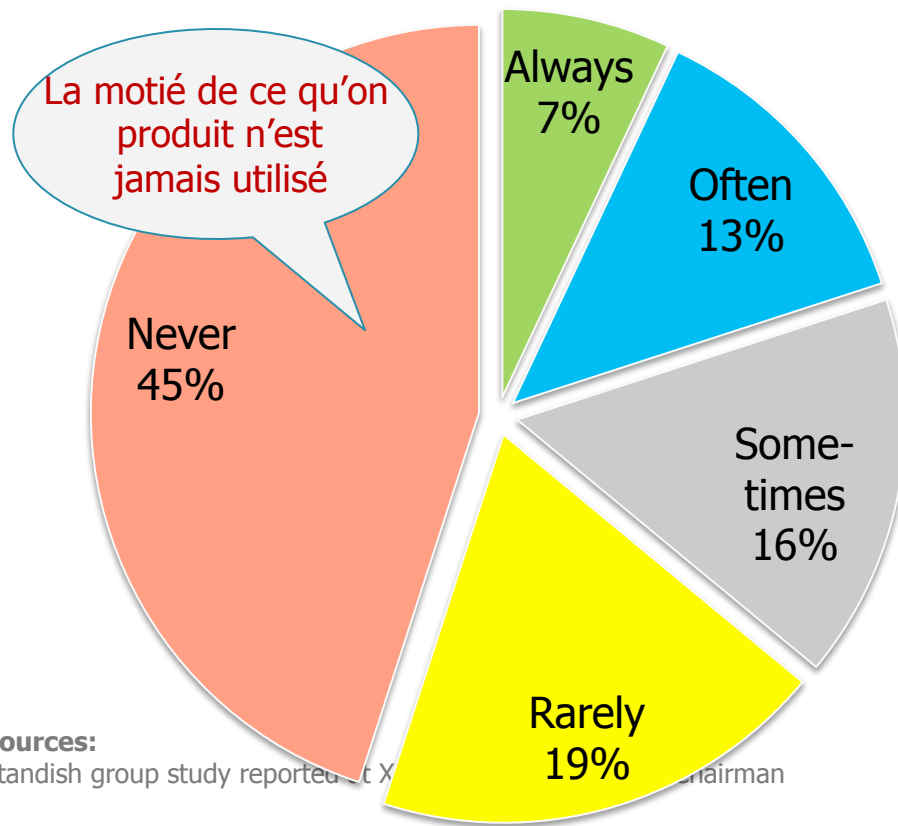
Sources:

<http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish>

<http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>

On a tendance à développer les "mauvaises" fonctionnalités

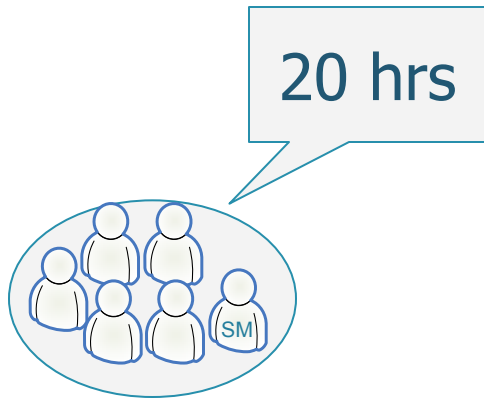
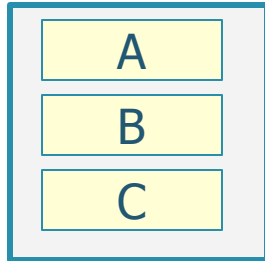
Features and functions used in a typical system



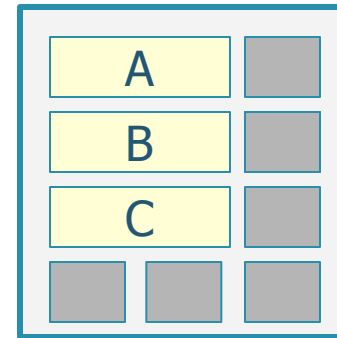
This graph courtesy of Mary Poppendieck

Comment les estimations sont affectées si on ajoute des informations "hors sujet"

Spec 1



Same spec
+ détails suferflus



Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

Le top 5 des raisons du succès

1. Impliquer l'utilisateur
2. Support du Senior Management
3. Objectifs Business Clairs
4. Optimiser le Scope du projet
5. Processus Agile

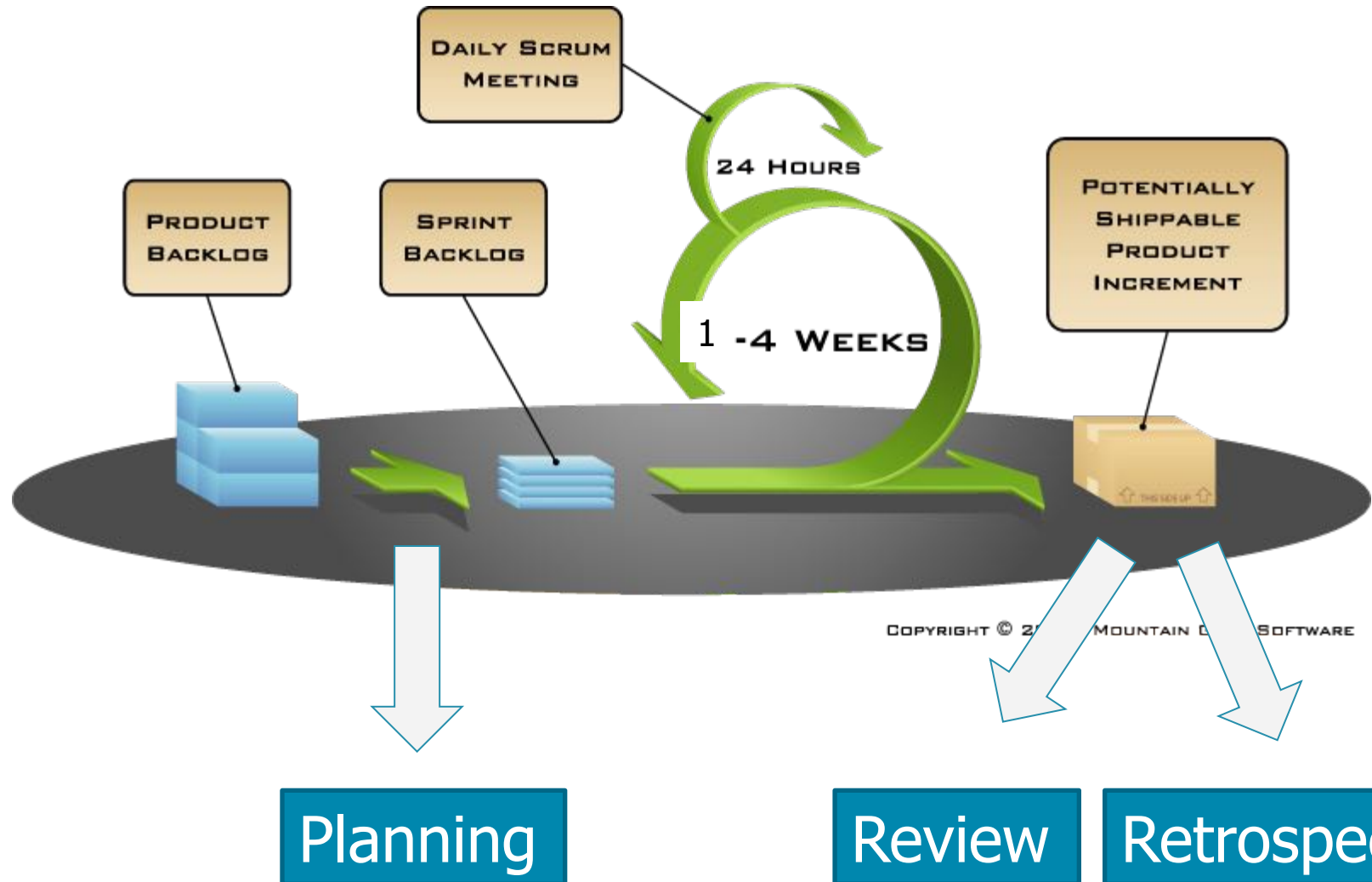
Sources:

<http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish>

<http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>

"My Life is Failure", Jim Johnson's book

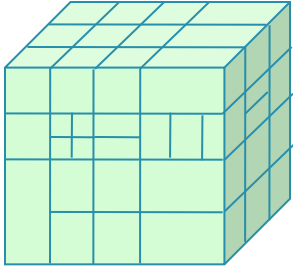
Scrum en Bref: les différentes réunions et les différents documents



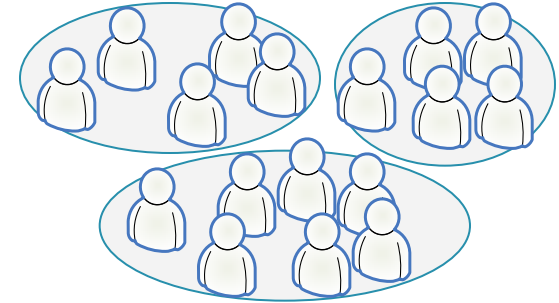
Divisez votre organisation en équipes

Scrum en bref

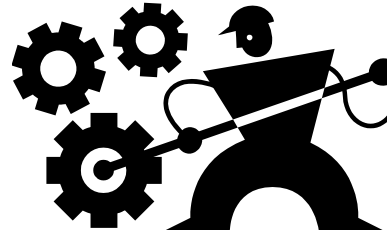
Divisez votre produit



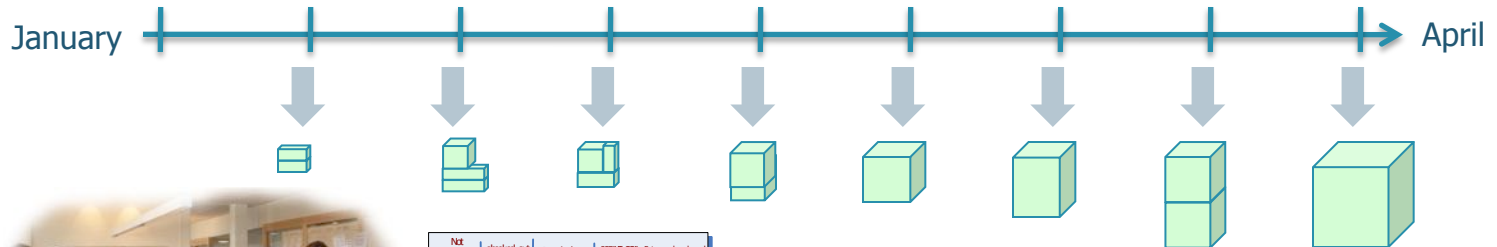
~~Un grand groupe passant beaucoup de temps à construire un gros truc~~
Une petite équipe passant un peu de temps à construire une petite chose
... Mais en intégrant rapidement pour voir l'ensemble



Optimisez le processus

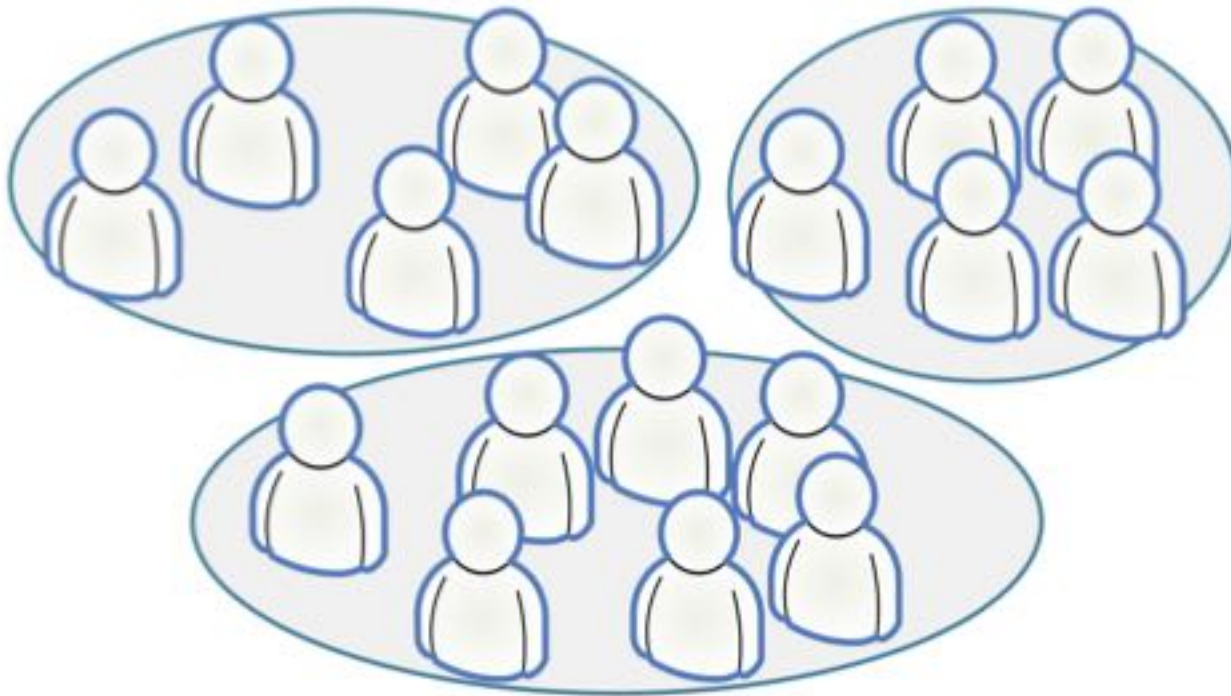


Court cycles de temps



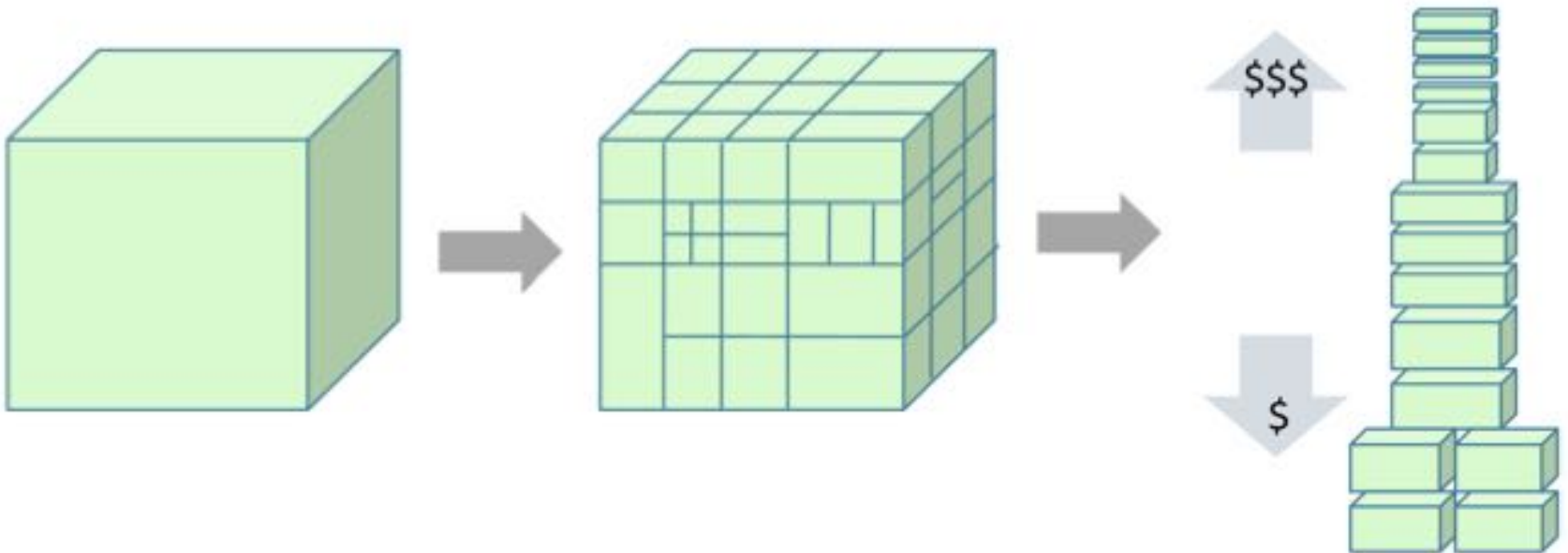
Mettre en place Scrum ? (1/4)

- **Divisez votre organisation** en petites équipes multidisciplinaires et auto-organisées.



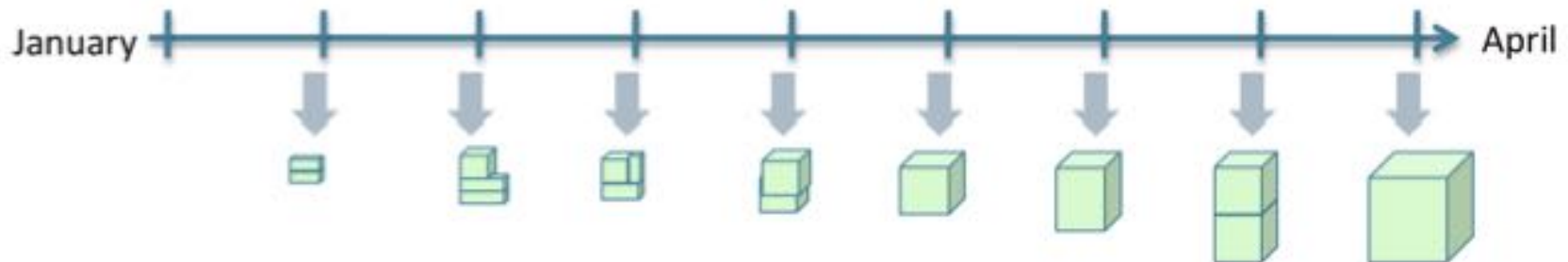
Mettre en place Scrum ? (2/4)

- **Divisez votre travail** en une liste de petits livrables concrets. Triez cette liste par priorité et estimez la taille relative de chaque élément.



Mettre en place Scrum ? (3/4)

- **Divisez le temps** en petites itérations de durée fixe (appelées des sprints et durant habituellement de 1 à 4 semaines) et faites une démonstration à l'issue de chaque sprint avec un produit potentiellement livrable.

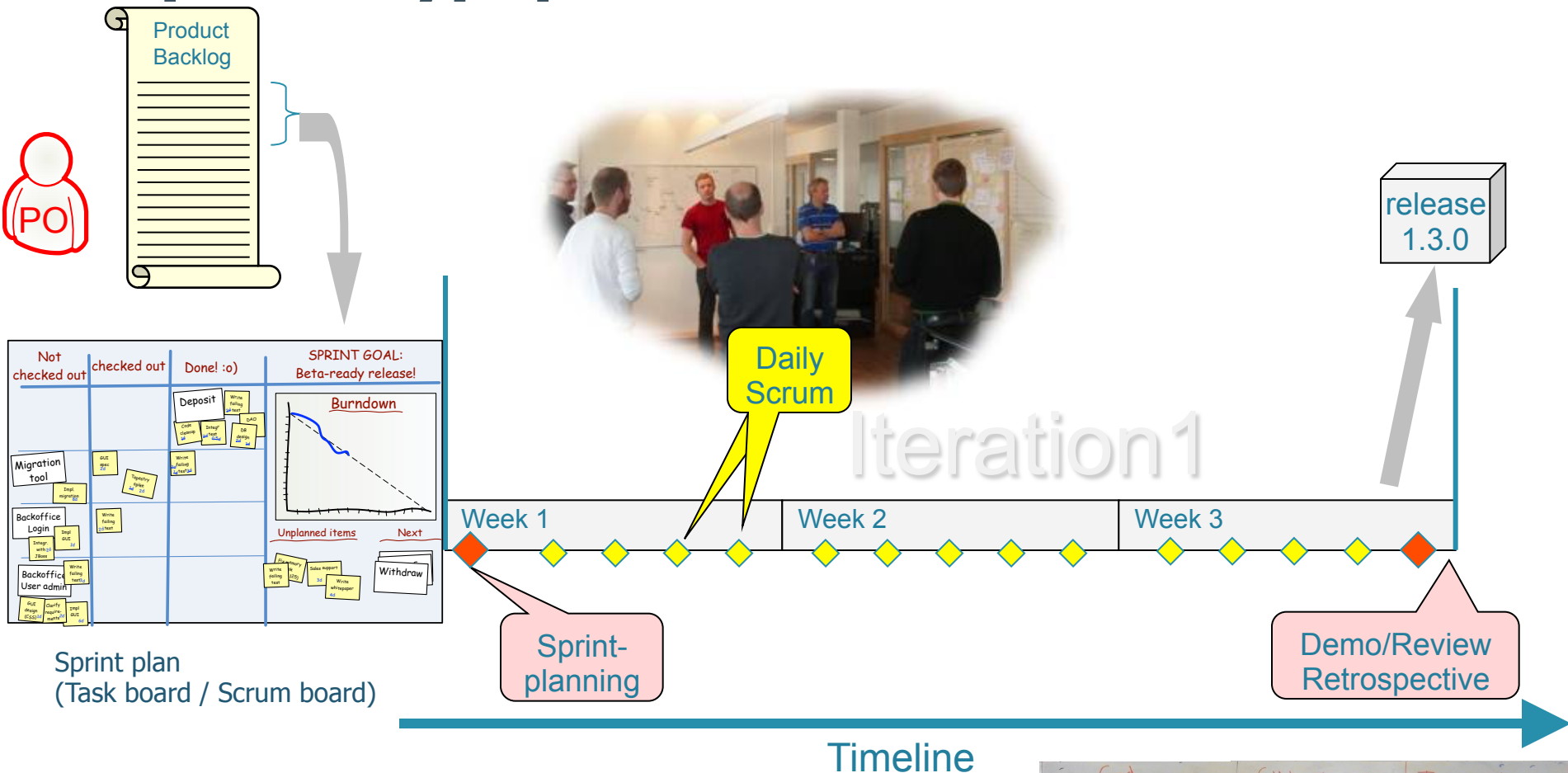


Mettre en place Scrum ? (4/4)

- **Optimisez le planning de la version** et mettez à jour les priorités en collaboration avec le client, sur la base de ce que vous avez appris après chaque sprint.
- **Optimisez le processus** en organisant une rétrospective après chaque sprint.

Ainsi, au lieu d'avoir un **grand groupe** passant **beaucoup de temps** sur la construction d'une **grande chose**, nous avons une **petite équipe** passant **un peu de temps** à construire une **petite chose**... mais **intégrant régulièrement** pour voir l'ensemble.

Un sprint "Typique"



















Le « Definition of Done »

- **A la fin d'un sprint, on délivre un incrément de produit qui correspond au définition of done.**
- **Le DoD est un document qui évolue avec le temps:**
 - On n'avait pas pensé à certaines choses...par exemple il faut écrire les releases notes. Pas de problème, on l'ajoute au DoD.
 - Certaines choses n'étaient pas possible avant, mais maintenant oui! Exemple: depuis qu'on fonctionne en intégration continue, on peut maintenant ajouter au Dod « déployé dans l'environnement d'acceptance ».

Exemple « Definition of Done »

- **Code: c'est développé en respectant nos standards**
- **C'est Testé (Unit, fonctionnel, acceptance)**
- **C'est documenté (Doc technique, user guide, release notes)**
- **C'est intégré (ça fonctionne sur les différents environnements: test, acceptance, pré-prod)**
- **C'est validé (par le PO et les documents de gouvernances sont à jour)**

Exemple de Definition Of Done (vieux mais dont j'ai l'autorisation de publication)

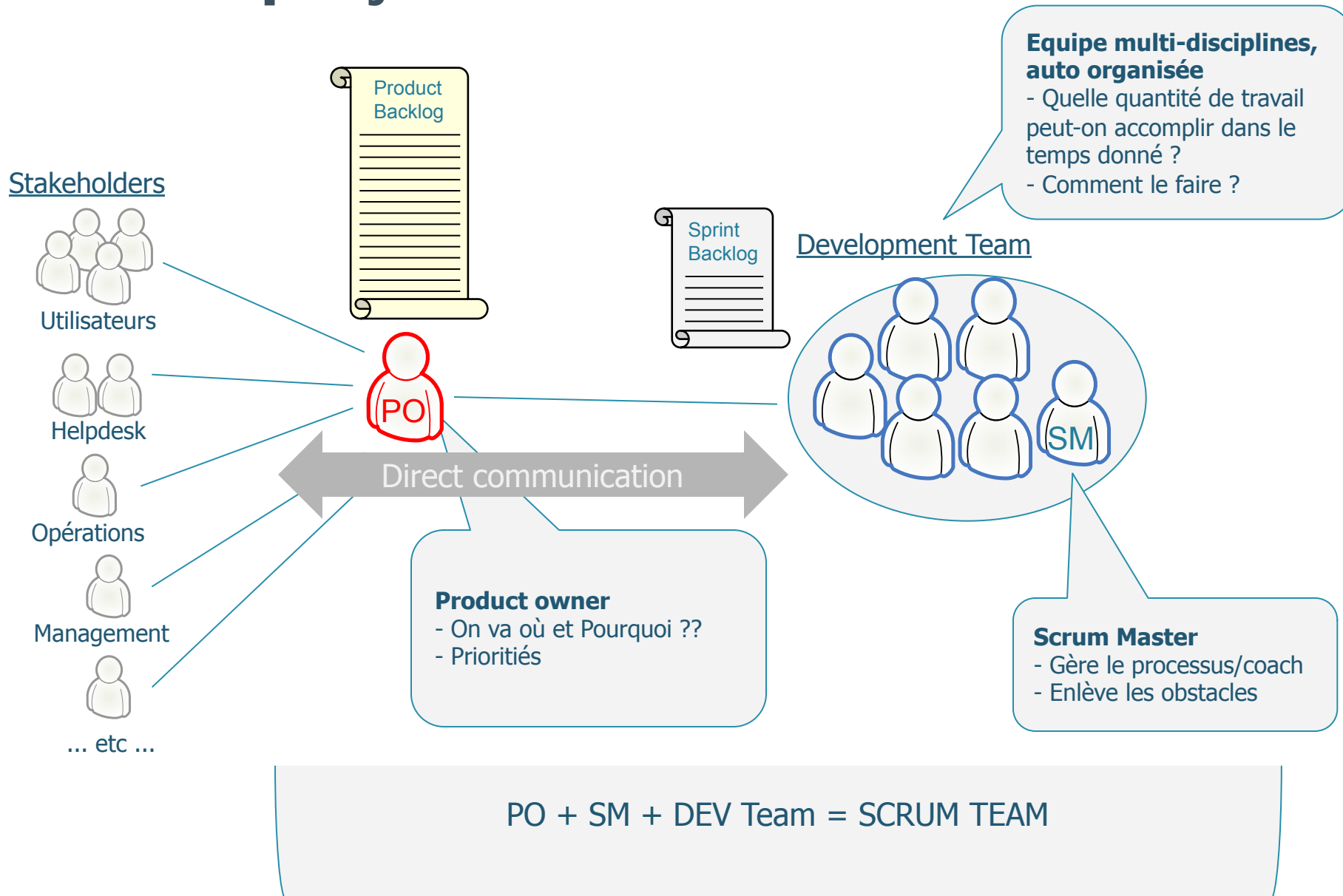
| | | | |
|--|---|--|---|
|  | Développement | Migration des données (structures + données) |  |
|  | Support IE7 + FF3 | Test Seleniums écrits |  |
|  | Support IE6 | Test Seleniums passé avec succès |  |
|  | Support "Navigateurs Home Page" | Test Unitaires écrits |  |
|  | Déployé sur Staging | Test Unitaires passé avec succès |  |
|  | Tests de régression ok (tous les tests passent) | Multilingue et traduction ok |  |
|  | Documentation (dossier d'hébergement,...) | Démarches à effectuer auprès de l'infrastructure (pour la Prod ou autres. Ex: url, connexion db,ftp,...) |  |
|  | Dépendance avec d'autres acteurs | Visualiser sur le mur |  |

A ajouter: Attribuer les droits aux utilisateurs, gestion des erreurs d'indisponibilités.

Aperçu des différents rôles

LES DIFFÉRENTS RÔLES

Scrum aperçu des différents rôles



Development Team

- **Ils réalisent le projet**
- **Les membres sont « multi-compétences »**
- **Polyvalent**
- **Autonome**
- **Auto-organisés**
- **Entre 5 et 9 personnes**

Le Scrum Master

- **Est responsable pour l'application de la méthode**
- **Son job: permettre à l'équipe d'avoir ce qu'il faut pour travailler**
- **Accompagne l'équipe, la «coache»**
- **Protège l'équipe des interférences**
- **Aide à la résolution des problèmes**
- **Connaît très bien Scrum**

Le Product Owner

- **Donne la direction**
- **Définit les priorités**
- **Valide au fur et à mesure**
- **Fait partie intégrante de l'équipe**
- **Peut aider à la résolution des problèmes**

Présentation du manifeste, des 4 valeurs et des 12 principes

LE MANIFESTE AGILE

Manifeste Agile

www.agilemanifesto.org

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.

Principes sous-jacents au Manifeste Agile

- Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
- Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.
- Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés.
- La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le dialogue en face à face.
- Un logiciel opérationnel est la principale mesure d'avancement.
- Les processus Agiles encouragent un rythme de développement soutenable. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- Une attention continue à l'excellence technique et à une bonne conception renforce l'Agilité.
- La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées.
- À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Principes sous-jacents au Manifeste Agile

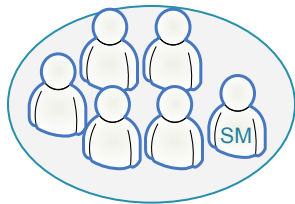
- Notre plus haute priorité est de **satisfaire le client** en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée.
- **Accueillez positivement les changements de besoins**, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.
- **Livrez fréquemment un logiciel opérationnel** avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.
- **Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble** quotidiennement tout au long du projet.
- Réalisez les projets avec des **personnes motivées**. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur **confiance** pour atteindre les objectifs fixés.
- La méthode la plus simple et la plus efficace pour transmettre de l'information à l'équipe de développement et à l'intérieur de celle-ci est le **dialogue en face à face**.
- Un **logiciel opérationnel** est la principale mesure d'avancement.
- Les processus Agiles encouragent un **rythme de développement soutenable**. Ensemble, les commanditaires, les développeurs et les utilisateurs devraient être capables de maintenir indéfiniment un rythme constant.
- Une attention continue à **l'excellence technique** et à une **bonne conception** renforce l'Agilité.
- La **simplicité** – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
- Les meilleures architectures, spécifications et conceptions émergent d'équipes **auto-organisées**.
- À intervalles réguliers, l'équipe **réfléchit aux moyens de devenir plus efficace**, puis règle et modifie son comportement en conséquence.

Scrum

LES ESTIMATIONS ET LE PLANNING

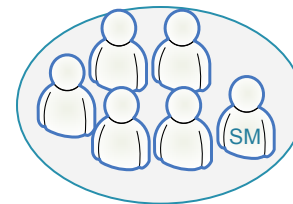
Comment les estimations sont affectées par la longueur des spécifications

Spécifications



117 hrs

Spec identiques – plus de pages



173 hrs

Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

Estimation relatives

- **Plus précis**
- **Indépendant que de la personne qui réalise**
- **Plus rapide**
- **Impact sur la motivation**

On n'estime pas la durée, on calcule la taille, la complexité avec des points relatifs ET on en déduit la durée.

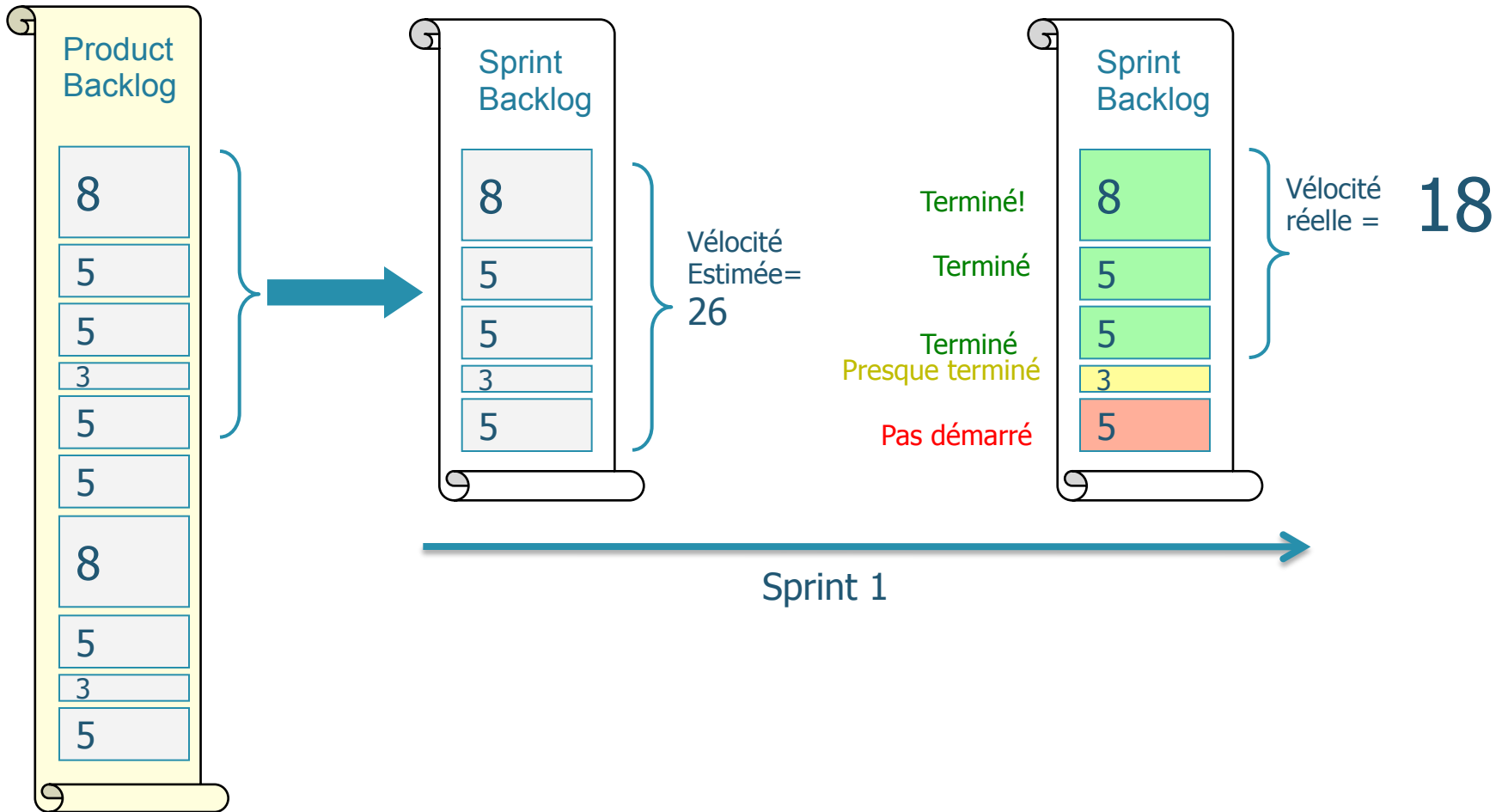
Poker planning

- **Explication détaillée du processus via ce lien:**
http://fr.wikipedia.org/wiki/Planning_poker

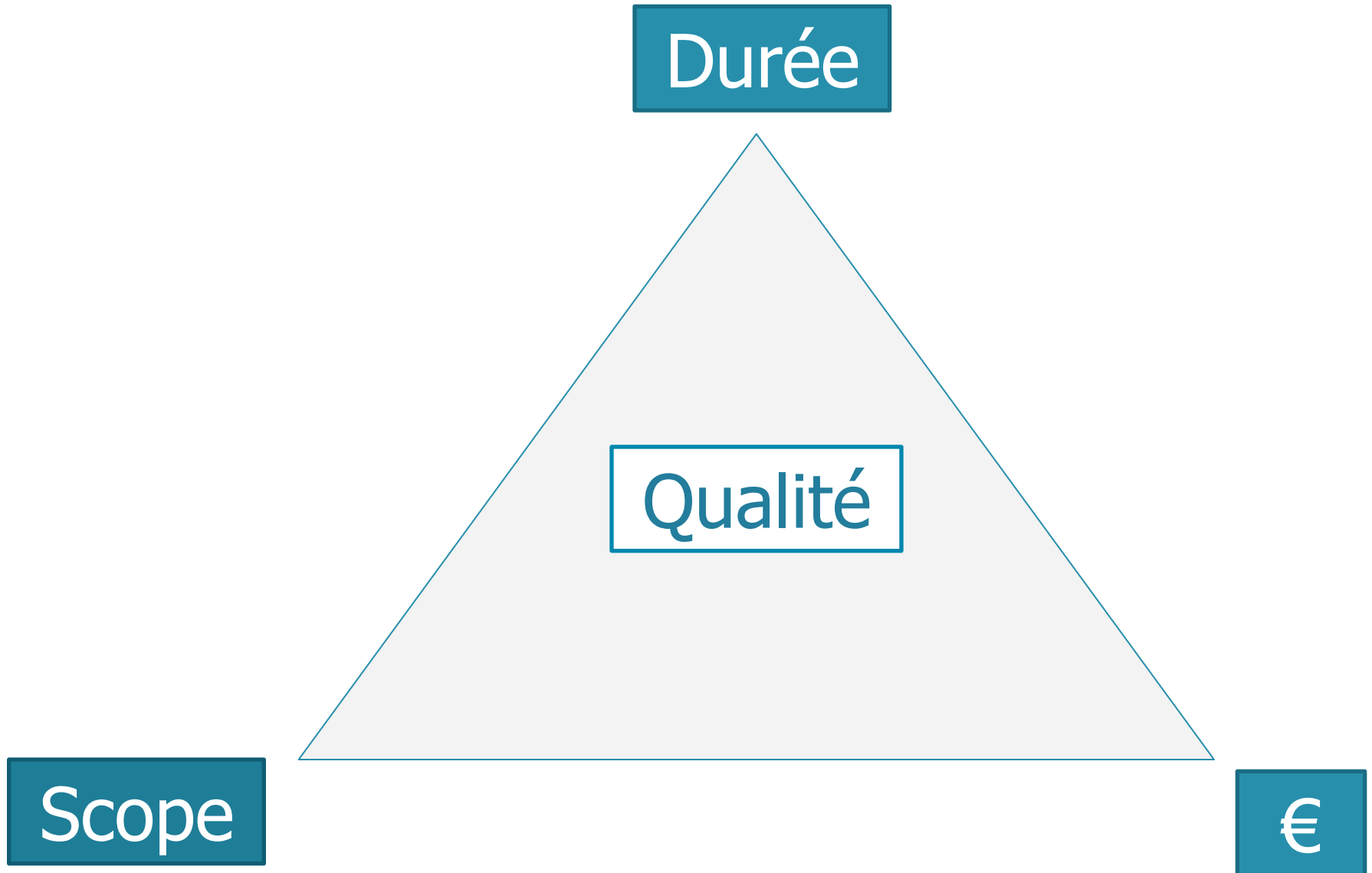
Mesurer la vélocité

Début du sprint 1

Fin du sprint 1



Les dimensions d'un projet



Release planning – durée limitée

- On est le 6 Août
- Sprint = 2 semaines
- Vitesse = 30 - 40

Qu'est-ce qui sera
terminé pour Noël ?

(10 sprints)

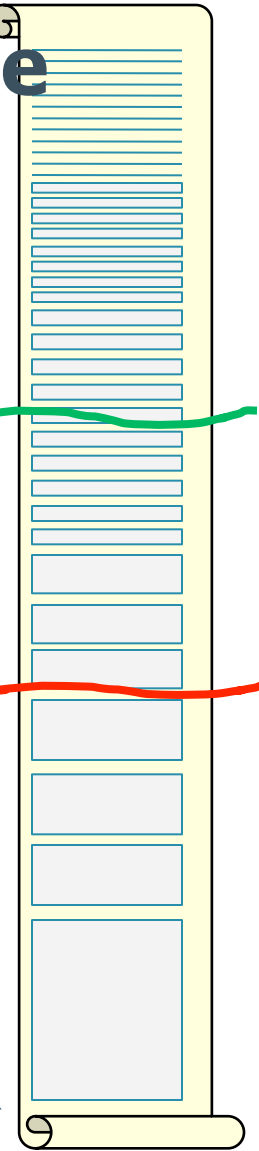
300



400

Release planning – budget fixe

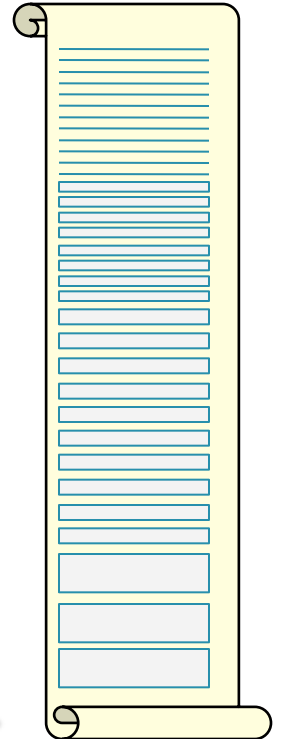
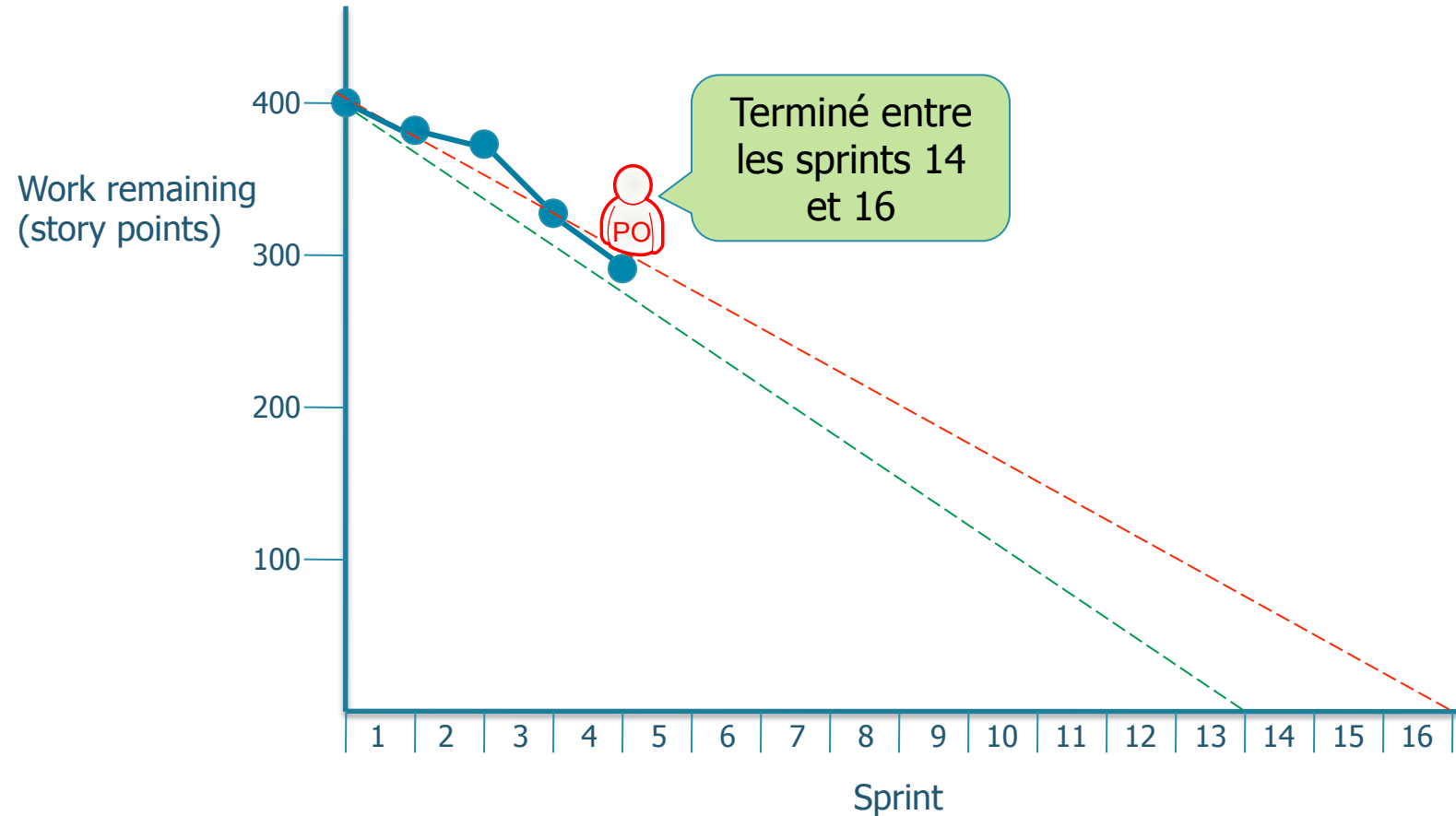
- Le budget est de 150.000 EUR
- Coût d'un sprint de 2 semaines ? => 15.000 EUR
- On peut réaliser 10 sprints
- Même problématique qu'une durée limitée



Release planning – Scope fixe

On aura
fini quand ?

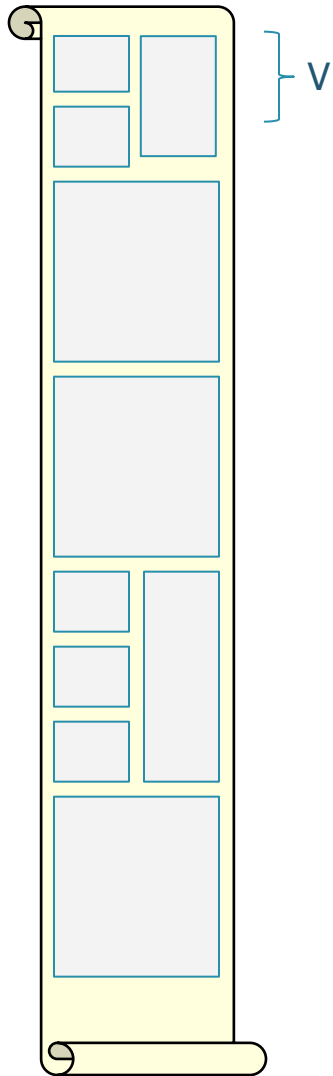
Release burndown chart



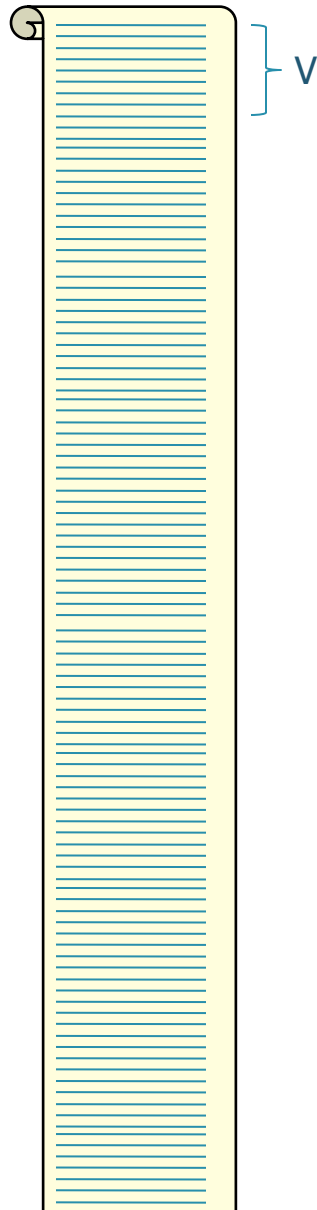
LE PRODUCT BACKLOG

Une juste balance pour product backlog

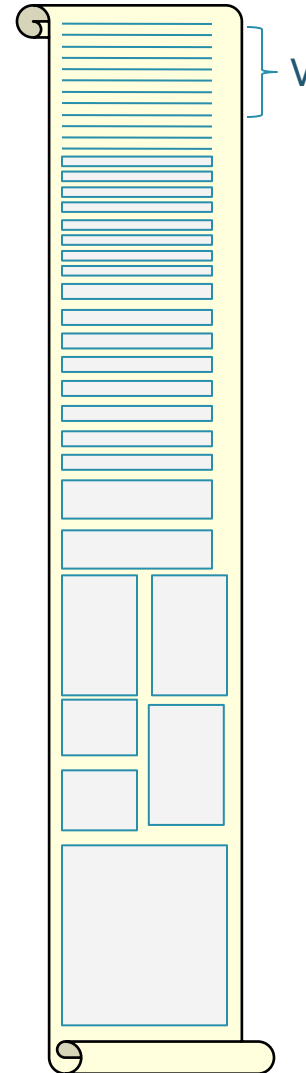
Too big items?



Too many items?



Balanced



http://www.youtube.com/watch?gl=BE&v=d_A-0ZNxgRs



🌈, Benoît Poelvoorde : La différence entre le cinéma belge et fra...

Petit intermède par Benoît Poelvoorde qui nous explique la différence entre faire du cinéma en France et en Belgique. Une belle métaphore d'un projet trop hiérarchisé versus un projet en mode Scrum.

Scrum

LES ACTIVITÉS

Le sprint planning

- **La première partie répond à la question « quoi ».**
Qu'est-ce qu'on va réaliser pour ce sprint ?
- **Cette partie est la responsabilité du Product Owner.**
- **La seconde partie répond à la question « comment ».** **Comment l'équipe va-t-elle réaliser ce que le PO demande.**
- **Cette partie est sous la responsabilité de l'équipe.**
- **Le résultat de cette réunion est le sprint backlog**

Sprint Review (demo)

- Répond à la question: qu'avons-nous accompli ?
- L'équipe réalise la démo aux stakeholders
- Seulement ce qui est terminé à 100% est montré
- Feedback direct de la part des stakeholders
- Le Feedback est inclus dans la product backlog

Sprint retrospective

Part 1: Que s'est-il passé ?

Part 2: Qu'est-ce qu'on veut faire de différent pour le prochain sprint ?





SCRUM

Le Daily Scrum

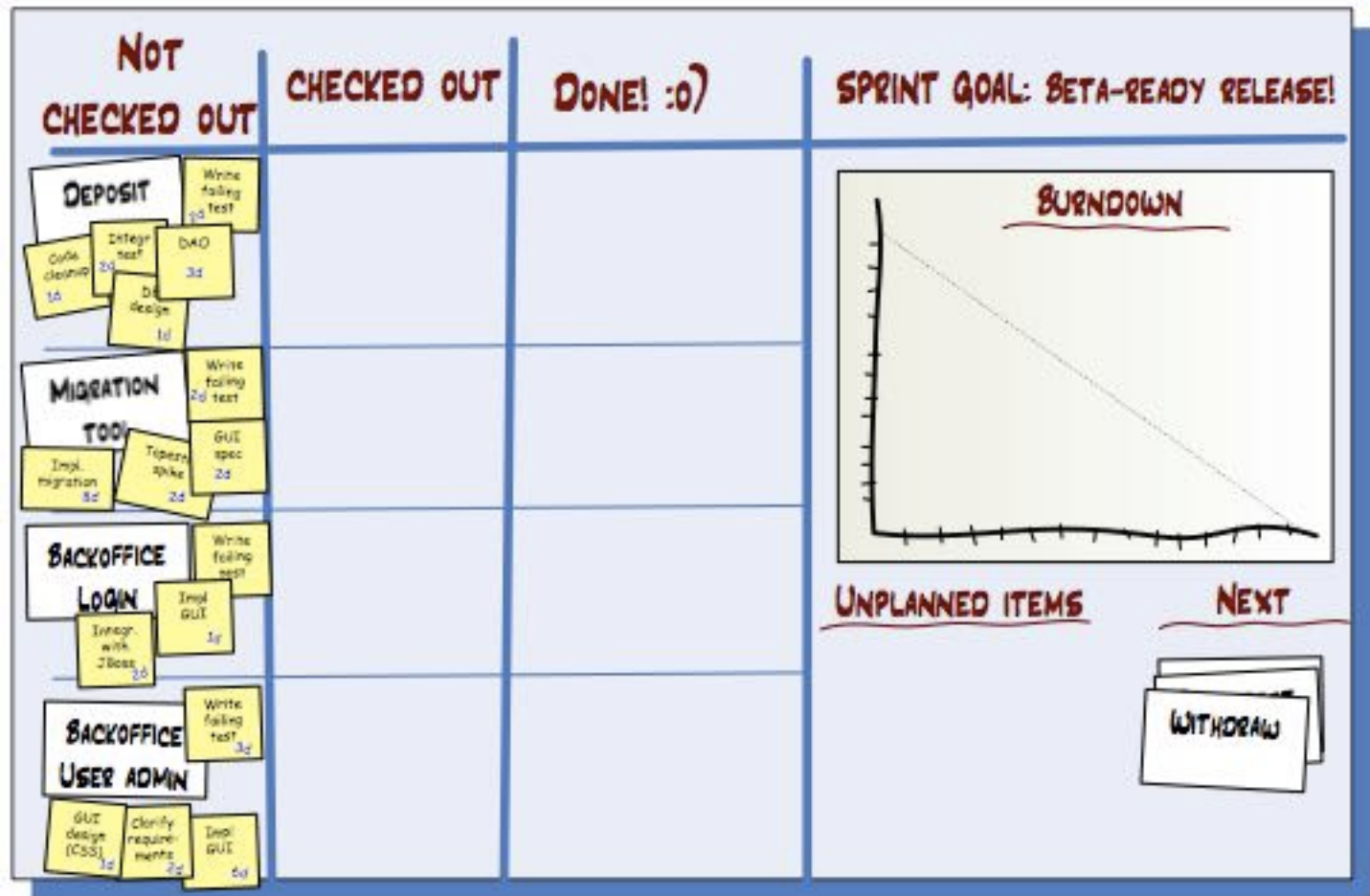
- Debout
- min 15 minutes
- (I) Ce que j'ai fait depuis la dernière meeting
- (II) Ce que je vais faire
- (III) Quelles difficultés je rencontre

Le sprint backlog

- **C'est la liste des choses à faire pour ce sprint**
- **Ces « choses » ont été sélectionnées dans le Product Backlog**
- **Souvent mais pas nécessairement, ces choses sont des user stories, d'autres formats peuvent être possible.**
- **Le sprint backlog vient avec un plan de réalisation.**
- **Le plan, c'est comment l'équipe pense réaliser les différentes features du sprint backlog**
- **Souvent, mais pas nécessairement, ce plan est composé d'une liste de tâches.**

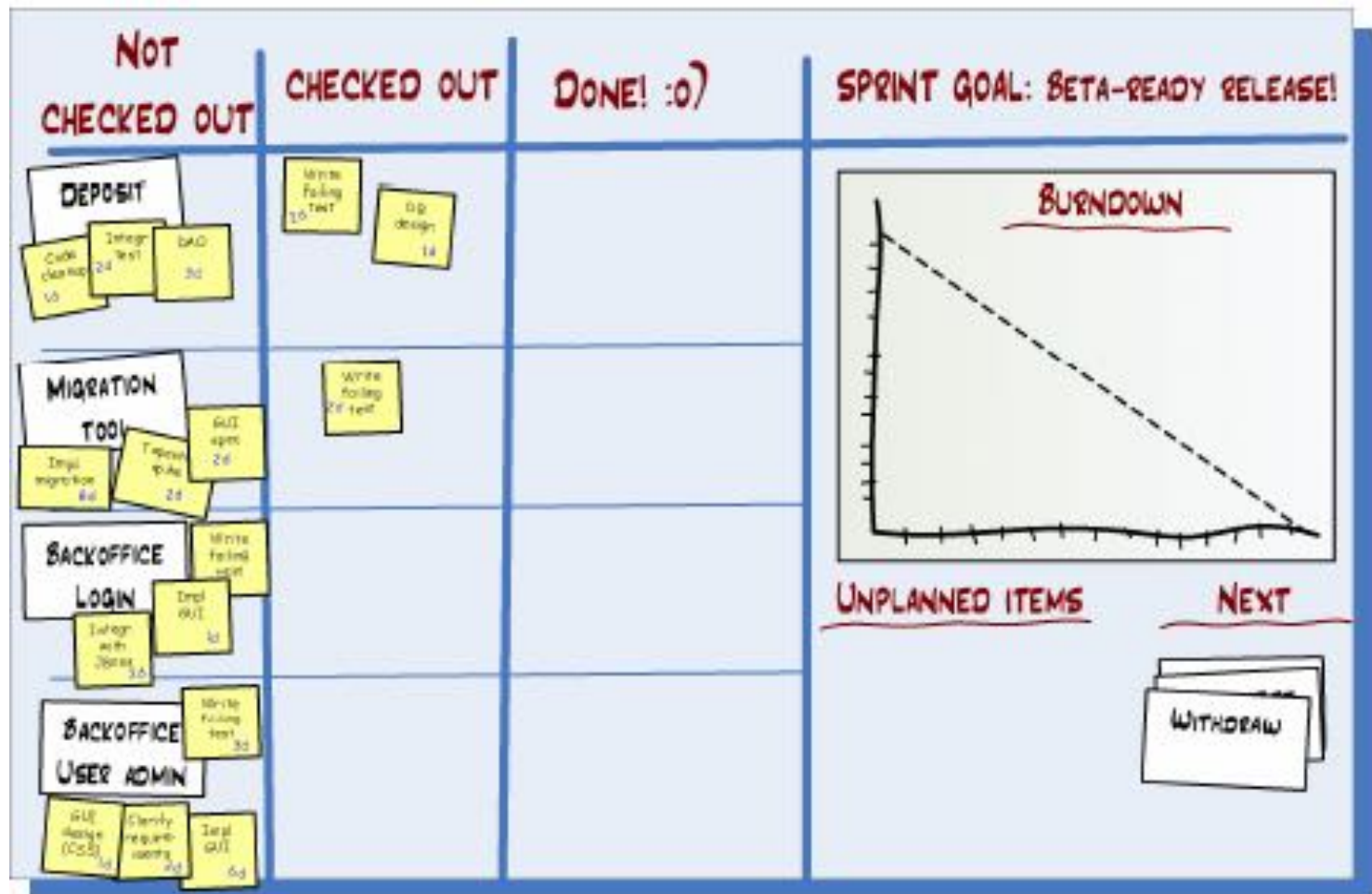
Exemple de sprint backlog, utilisant des stories (en blanc) et des tâches (en jaune)

Sprint 4 – Day 0,5 of 30



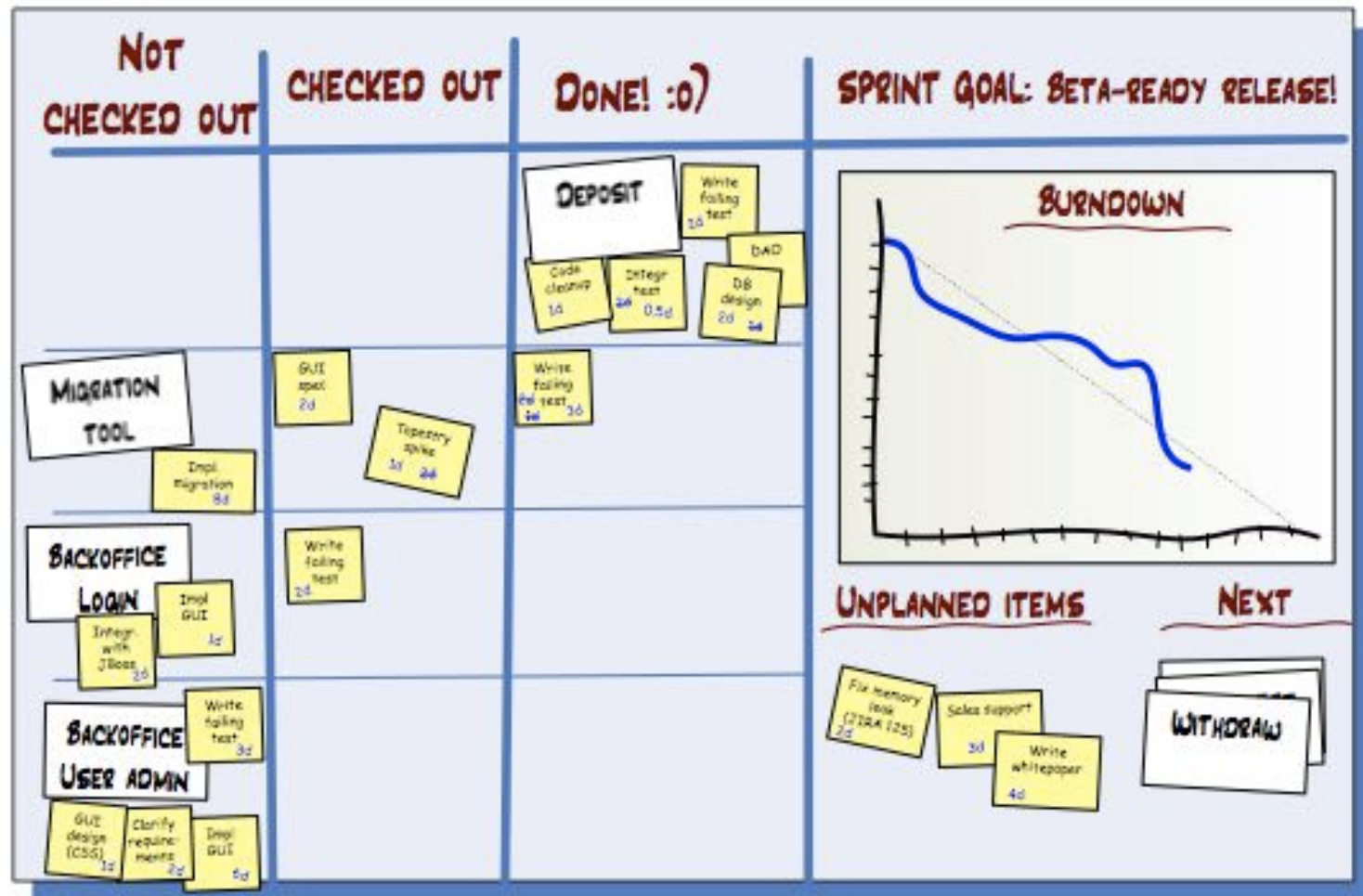
L'équipe commence à travailler par le haut du sprint backlog, les stories les plus prioritaires

Sprint 4 – Day 01 of 30



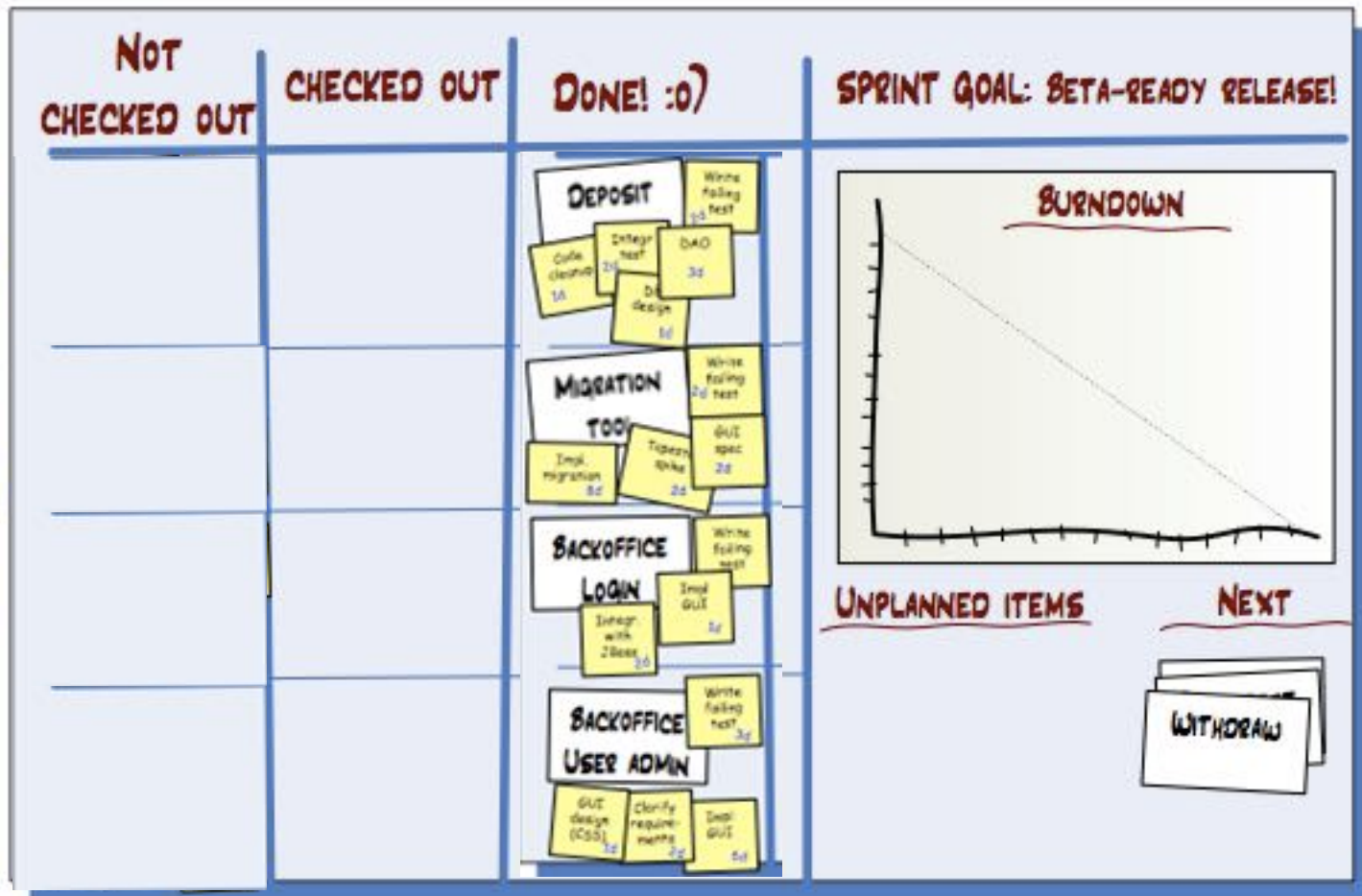
Une bonne pratique: terminer les stories complètement, avant d'en démarrer d'autres

Sprint 4 – Day 12 of 30

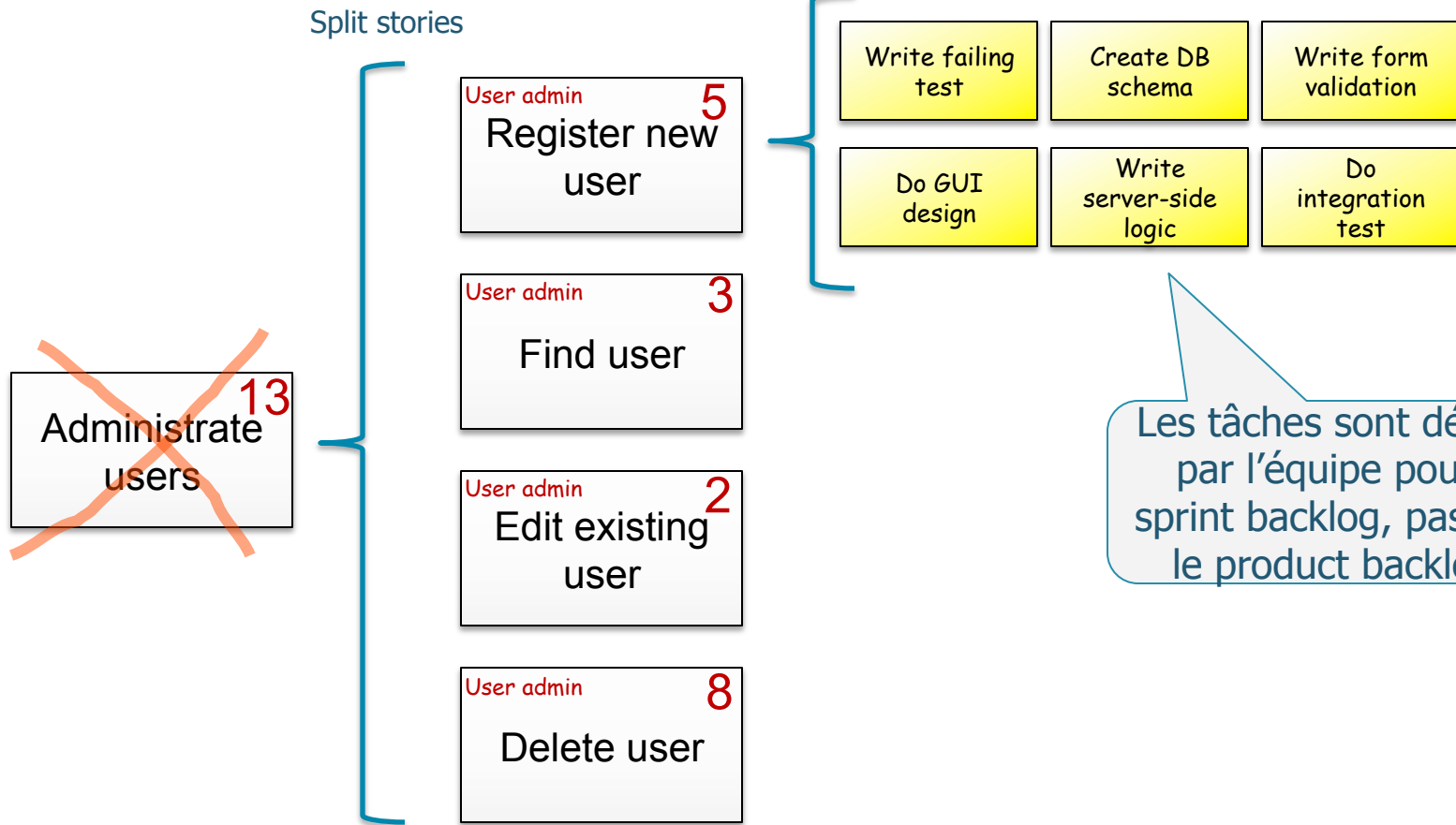


Fin du sprint

Sprint 4 – Day 30 of 30 (Happy Scenario 😊)



Découper une story en stories et en tâches



Les raisons d'échecs d'un projet Scrum

- **Ne pas vouloir voir (ou traiter) les problèmes**
- **Faire comme avant mais avec le vocabulaire "Agile"**
- **Customiser (trop tôt) Scrum**

Qui utilise Scrum ?



Exemple de backlog

ID

Planning + Historique

| ID | Categorie | Description | | | | | Fib | Planned in sprint# | Done in sprint # | |
|----|------------------|-------------|-------------|-----------|--|----------|--|--------------------|--------------------|------------------|
| 0 | Cat | As a | (user-role) | i want! | (action) | so i can | (business objective) | F | Planned in sprint# | Done in sprint # |
| 56 | Design | As a | | i want to | Mettre au point le design de la home page "mobilité" | so i can | put dynamic applications in the real web design | 1 | 1 | 1 |
| 57 | Ticker | As a | Internaut | i want to | be able to read the info traffic on the home page | so i can | be inform on major incidents in brussels, encoded later via the portal by mobiris team | 3 | 1 | 1 |
| 58 | Mobiris | As a | Internaut | i want to | see in the simplest way, some dynamic datas from Mobiris | so i can | be reassured on the fact that we can import datas from Mobiris database to our portal | 21 | 1 | 1 |
| 59 | Projet - Contenu | As a | Internaut | i want to | Pouvoir visualiser au moins 3 exemples de pages "projets" (p53 emakina) dont le contenu est déjà "stocké" dans la plateforme | so i can | afin de pouvoir, voir comment le contenu produit se présente sur l'écran. Le contenu est fourni par Bxl Mobilité | 21 | 1 | 1 |
| 60 | Projet - Contenu | As a | | i want to | Définir le Wireframe (layout) de la page "projets" document n.53/118 | so i can | pouvoir produire du contenu et visualiser le contenu dans la démo | 13 | 1 | 1 |
| 61 | Projet - Contenu | As a | | | | | | 8 | 1 | 1 |
| 63 | News | | | | | | | 8 | 3 | 3 |
| 52 | Works | | | | | | | 5 | 2 | 2 |
| 53 | Generic | | | | | | | 21 | 2 | 2 |
| 54 | Map | | | | | | | 8 | 3 | 3 |
| 50 | Generic | As a | | | | | | 8 | 5 | 5 |
| 65 | Camera | As a | Internaut | i want to | Show the simplest way pictures of a real time camera | so i can | be reassured on the fact that we can import datas from mobiris database to our portal | 2 | | |
| 66 | Mobiris | As a | Internaut | i want to | see on the map of the home page mobilité, the differents parkings available from mobiris | so i can | | 3 | | |
| 67 | Map | As a | Internaut | i want to | I want to see in differents colors the traffic jam on the map of brussels | so i can | | 5 | 2 | 2 |
| 69 | Mobiris | As a | Internaut | i want to | see on the map of the home page mobilité, the differents events from mobiris (evenements et travaux routiers) | so i can | | 8 | 2 | 2 |
| 70 | Mobiris | As a | Internaut | i want to | put my mouse on the map and have a pop-up with detailed info on the event | so i can | | 13 | 4 | 4 |
| 71 | Camera | As a | | i want to | see the different thumbnailed pictures of camera when i click "camera" on the home page (page with index of all cameras) | so i can | | 1 | | |

User Story:
Ligne bleue: faite, Ligne blanche: à faire

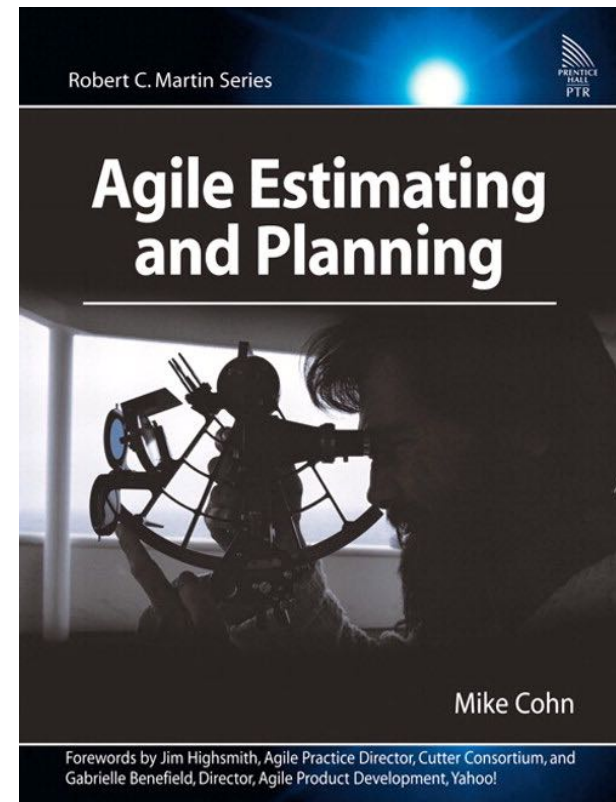
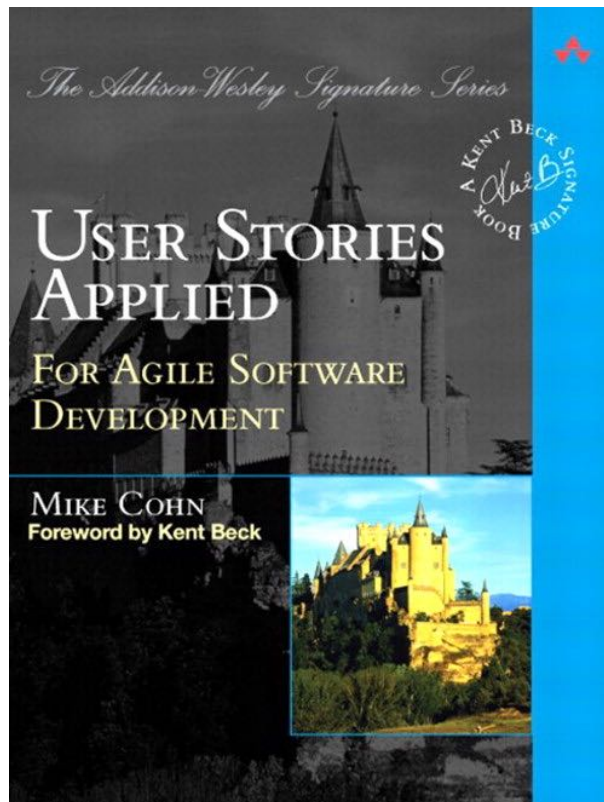
Catégories

Fibonacci

En savoir plus

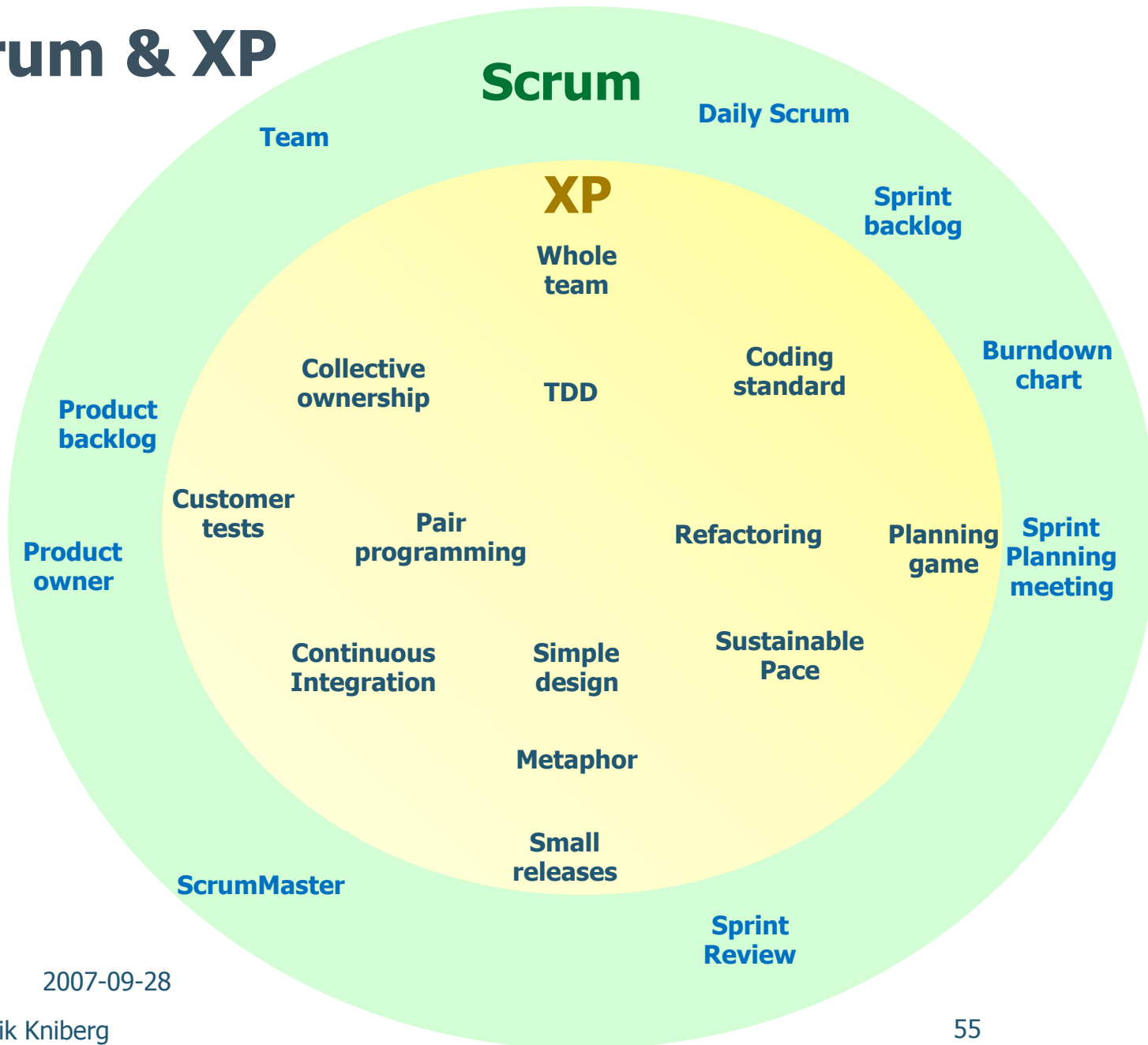
- www.mountangoatsoftware.com (EN)
- www.agile-software-development.com (EN)
- www.aubryconseil.com (FR)
- borisgloger.com/en/2008/07/01/product-backlog-templates-scrum-tools/ (EN - templates)
- www.xqa.com.ar/visualmanagement/ (EN)
- www.brunosbille.com (FR-EN)

Livres



TECHNIQUES D'INGÉNIERIE ET D'ARCHITECTURE

Scrum & XP



2007-09-28

Feedback loops



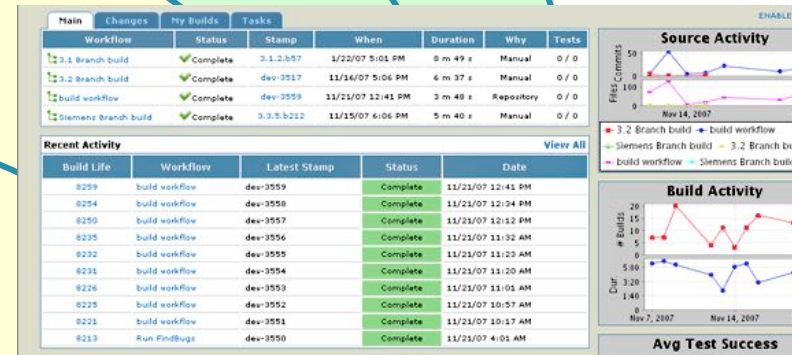
Sprint review



Daily Scrum



Continuous integration



Unit test

Pair programming



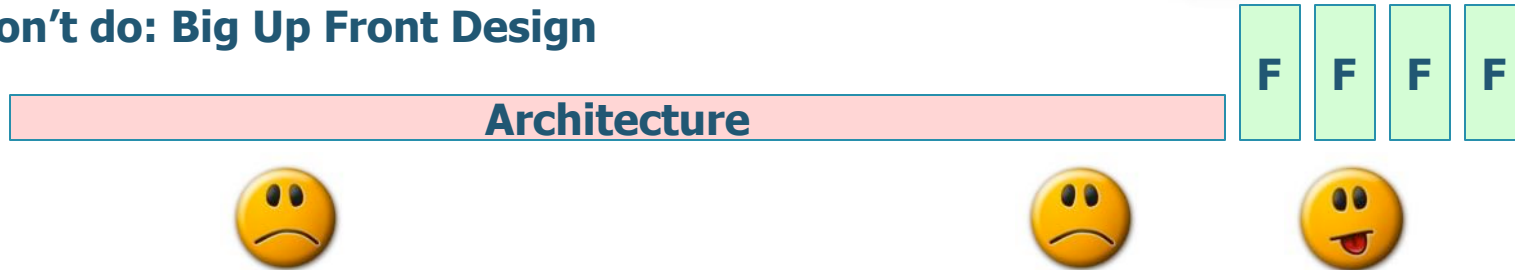
Agile architecture



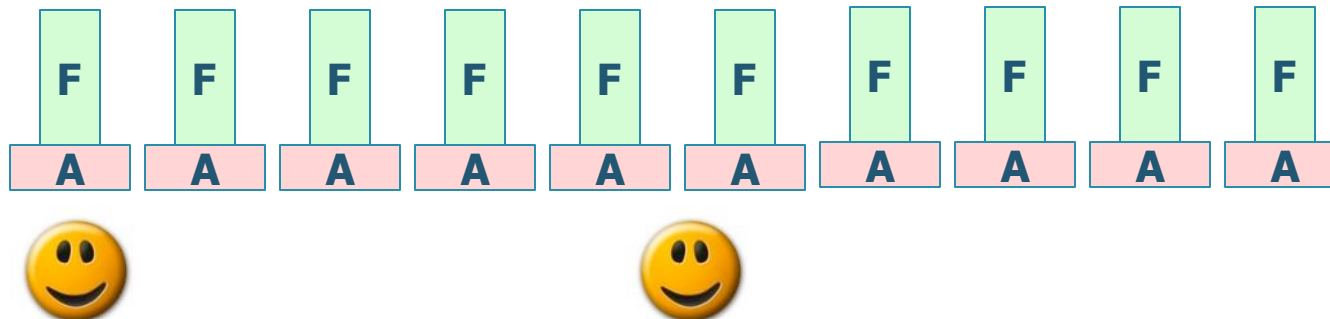
Don't do: Quick 'n dirty features => Slow 'n dirty => Entropy death? Big bang rewrite?



Don't do: Big Up Front Design



Do: Find a balance



Clean & simple code

Simple code:

1. Passes all tests
2. No duplication
3. Readable
4. Minimal

Simple is hard!

~~Code is an asset~~
All code is cost!
Some code is value.

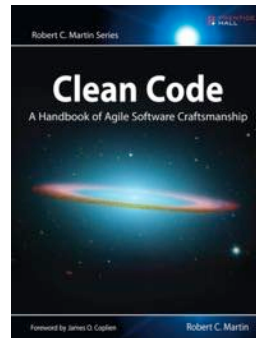
```
public class Dog {  
    private final String name;  
    private int woofCount = 0;  
  
    public Dog(String name) {  
        this.name = name;  
    }  
  
    public void woof() {  
        ++woofCount;  
    }  
}
```



```
import java.sql.Connection;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
public class Dog {  
    private Executor executor = Executors.newFixedThreadPool(18);  
    private int CACHE_SIZE = 50;  
  
    public Dog() {  
        try {  
            Class.forName("oracle.jdbc.ThinDriver");  
            connection = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin", "beefhead");  
            statement = connection.prepareStatement("insert into Dog values (?, ?, ?)");  
        } catch (ClassNotFoundException e) {}  
  
        new Thread().start();  
    }  
  
    public void woof(Person woofCaller) {  
        Connection connection = null;  
        PreparedStatement statement = null;  
        try {  
            connection = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin", "beefhead");  
            statement = connection.prepareStatement("insert into Dog values (?, ?, ?)");  
            statement.setLong(1, System.currentTimeMillis());  
            statement.setString(2, person.getName());  
            statement.setString(3, person.getPhoneNumber().getNumber());  
            statement.executeUpdate();  
        }  
        Connection a = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin", "beefhead");  
        b = a.prepareStatement("select * from Dog where name = '" + name + "'");  
        c = b.executeQuery();  
        if (c.next()) {  
            String foundName = c.getString("name");  
            PhoneNumber phoneNumber = new PhoneNumber(c.getString("woofCount"));  
            Person person = new Person(foundName, phoneNumber);  
            return person;  
        } else {  
            return new Person("", null);  
        }  
  
        catch (SQLException e) {  
            return null;  
        }  
        catch (IllegalArgumentException x) {  
            throw x;  
        }  
    }  
}
```



Embrace
Change!



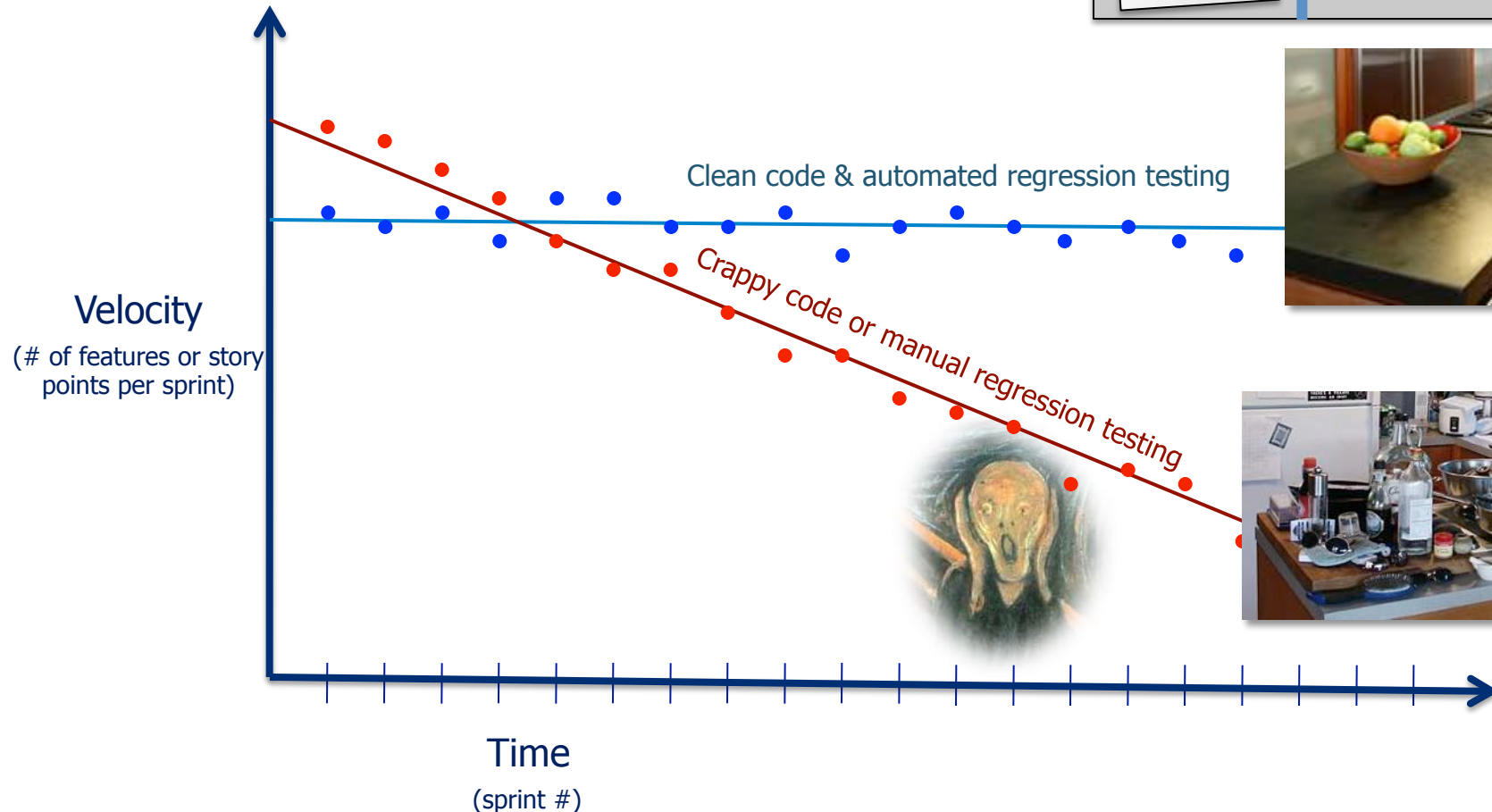
```
All() {  
    rManager.getConnection("jdbc:oracle:thin:@prod", "admin", "beefhead");  
    tion.prepareStatement("insert into Dog values (?, ?, ?)");  
    1, System.currentTimeMillis());  
  
    {  
        tString("name");  
        umber = new PhoneNumber(c.getString("woofCount"));  
        w Person(foundName, phoneNumber);  
    }  
}
```

Robert C Martin (Uncle Bob)

Kent Beck
Henrik Kniberg

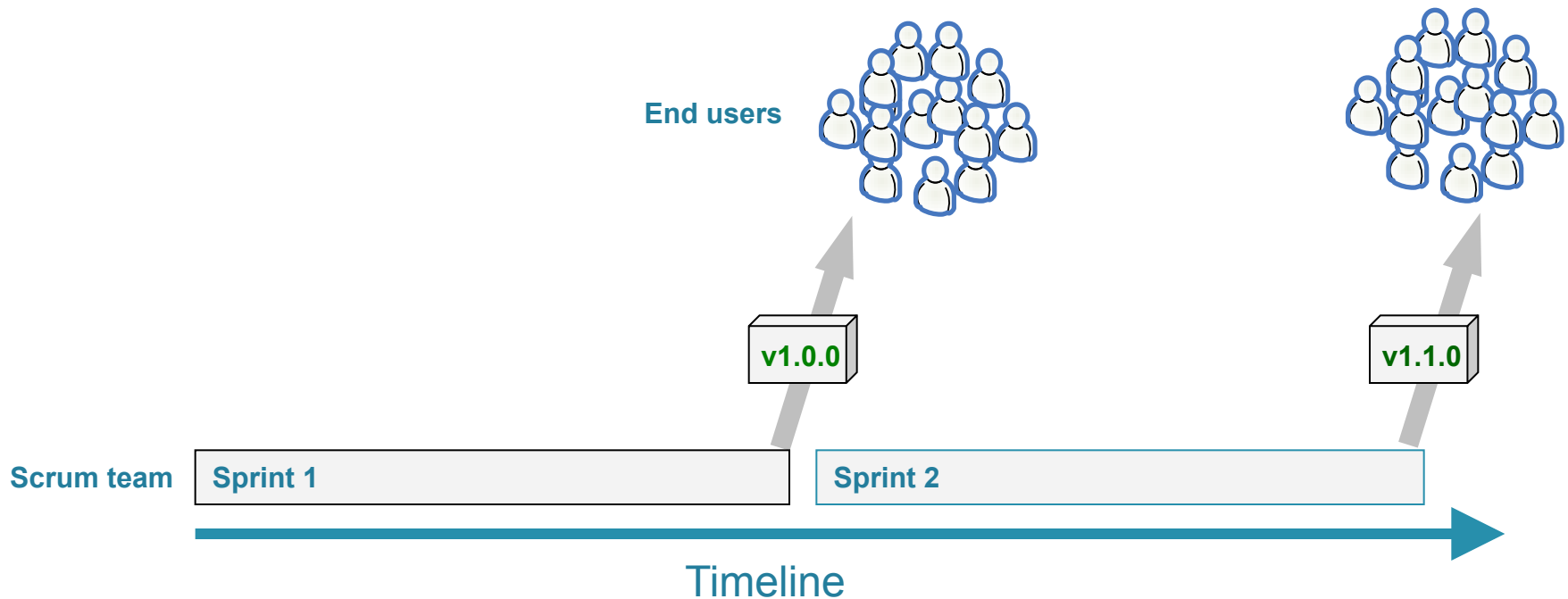
"Done" includes clean code & regression testing

| To do | Doing | Done |
|----------------------|---------|----------|
| Withdraw Transfer | Deposit | Register |



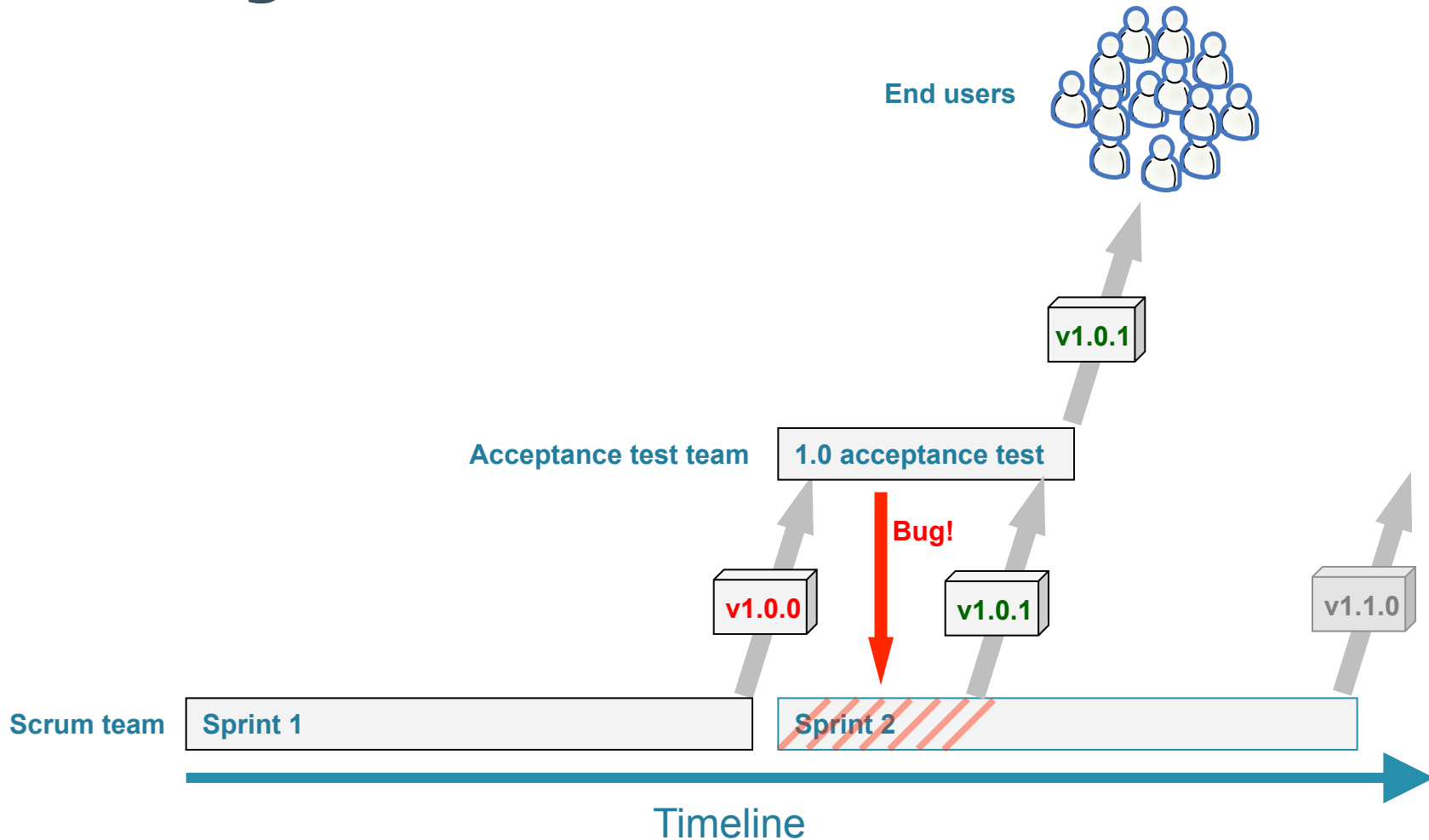
Testing

Testing – ideal case



19/11/14

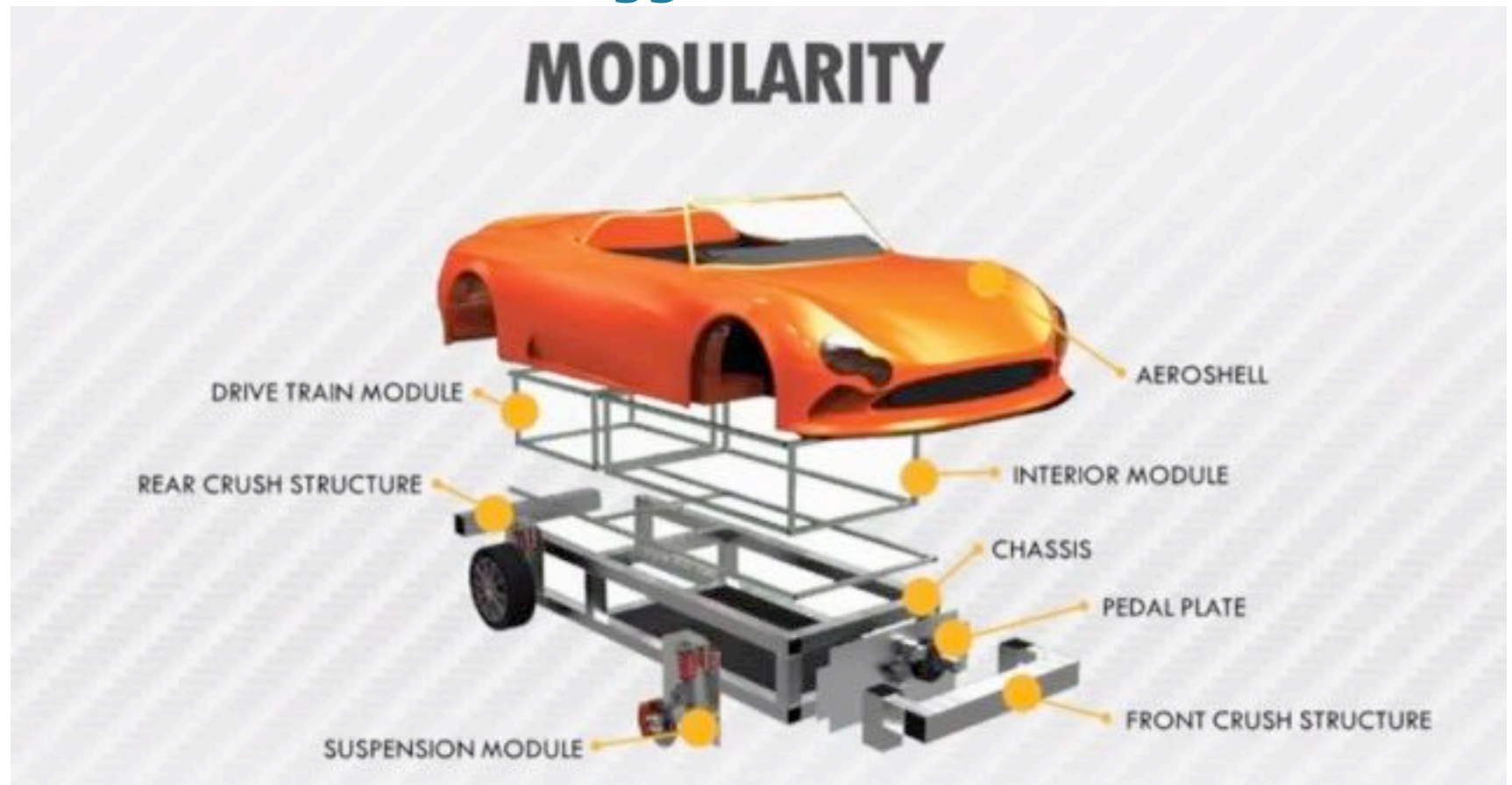
Testing – common alternative



19/11/14

Créer une voiture avec Scrum

- <http://ouishare.net/fr/2013/05/wikispeed-revolution-industrielle-open-source/#.UZnvi95hvNh.blogger>



Vidéo sur Scrum: 5 minutes sur la vie d'un projet en mode Scrum.

<https://vimeo.com/4587652>



Attention, la vidéo à été filmée en 2008, certaines choses ne se font plus nécessairement: par exemple estimer les tâches en heures.

Sources

- **The Essence of Agile (Keynote by Henrik Kniberg)**
- **<http://agileatlas.org/atlas/scrum>**