



# Support de cours



## Parcours Agile (1) /

GKAG03: Ingénierie logicielle Agile  
(PSD Professional Scrum Developer)  
(3J)

GKAG04: De Chef de Projet à Manager Agile  
(PSM Professional Scrum Master)  
(2J)

GKAG05: Le rôle du Responsable Produit dans le Projet Agile (PSPO Professional Scrum Product Owner)  
(2J)

GKAG02: Travail en équipe Agile  
(2J)

GKAG01: Méthodes Agile: comprendre la démarche  
(2J)

Contenus et articulation des 5 cours tels que désignés et requis par le Fafiec

## Parcours Agile (2)

- Module GKAG01: fortement conseillé pour tous les parcours
- Pré-requis du module GKAG02 :
  - Module GKAG01 obligatoire
- Pré-requis du module GKAG03 :
  - Module GKAG01 fortement conseillé
  - Module GKAG02 obligatoire
- Pré-requis du module GKAG04 :
  - Module GKAG01 fortement conseillé
  - Module GKAG02 conseillé
- Pré-requis du module GKAG05
  - Module GKAG01 fortement conseillé
  - Module GKAG02 conseillé

Contenus et articulation des 5 cours tels que désignés et requis par le Fafiec

## Logistique / Organisation

- Horaires (repas/pauses)
- Toilettes, boissons
- Téléphone Mobile / Laptop
- P@ssw0rd, wifi4all
- Code cours, code session -> Signatures
- Accès eDoc
- CAT: Comprendre (/cours), Apprendre (/soir), Tester (/fin)
- Ouverture (si différent de vos pratiques, relax -> Scrum.org)
- Parole libre, mais un seul à la fois ;-)
- Exercices -> n'allez pas au slide suivant ;-)

© Global Knowledge Training LLC. All rights reserved.

Page 4

Pour rappel :

Login sur les PC des salles de cours GK : GK / mot de passe : P@ssw0rd

Code Wifi : Attention il dépend du site GK sur lequel vous vous trouvez.

Renseignez vous à l'accueil

Pour tous les points administratifs (Signatures, accès eDoc, évaluation de fin de cours ➔ Consultez le document dans votre dossier formateur)

## Présentation

- Nom, Prénom
- Parcours (éducation, jobs)
- Poste actuel
- Avez-vous déjà utilisé Agile/Scrum?
- Avez-vous choisi de venir ici?

Le formateur se présente (1 slide) après les participants

## Summary

- Origin of Agile Culture:
  - Agile Manifesto (4 values, 12 principles)
  - Difference between Culture, Framework, Methodology, Language, Tool
  - Serious Game: "IceBreaker Agile"
- Panorama of the main Agile approaches:
  - Lean, Kanban, Scrum, ScrumBan, XP, Crystal, DSDM, FDD, SAFe, Design Thinking, Lean Startup, LeSS, Nexus, etc.
- Scrum:
  - **3 Pillars:** Transparency, Inspection Adaptation - **5 Values:** Focus, Respect, Openness, Commitment, Courage - **3 Roles:** Development Team, Product Owner, Scrum Master - **5 Events:** Sprint Planning, Daily Meeting, Sprint Review, Sprint Retrospective, Sprint - **3 artifacts:** Product Backlog, Sprint Backlog, Increment - Definition of Done, Release

### Sommaire

#### Origine de la Culture Agile :

Agile Manifesto (4 valeurs, 12 principes)

Déférence entre Culture, Cadre de travail, Méthodologie, Langage, Outil

Serious Game: « IceBreaker Agile »

#### Panorama des principales approches Agiles :

Lean, Kanban, Scrum, ScrumBan, XP, DSDM, FDD, SAFe, Design Thinking, Lean Startup, LeSS, Nexus, etc.

#### Scrum :

**3 piliers :** Transparence, Inspection, Adaptation - **5 valeurs :** Focus, Respect, Ouverture, Engagement, Courage - **3 rôles :** Equipe de Développement, Product Owner, Scrum Master - **5 évènements :** Sprint Planning, Daily meeting, Sprint Review, Sprint Rétrospective, Sprint - **3 artifacts :** Product Backlog, Sprint Backlog, Incrément - Définition du Done (terminé), Release

## Summary (2)

- Change management in the company:
  - Clarification between Portfolio, Programs, Project, Production, Project Management, Product Management, Product Scope, Project Scope
- Project management :
  - Distinction between Classic Project and Agile Projects (Iterative, Incremental, Simple / Complex) - herding cats »video
- Criteria for choosing an approach (Classic / Agile), areas of application and specific risks
  - Serious Game "Select the good temperature in a room"
- Myths about choosing Agile as an approach, about the application of Agile and about its specific risks
- Key Success Factors for a Successful Implementation of Agile
  - Serious Game "Fatburger" - "Agile Family Management" video

### **Conduite du changement dans l'entreprise :**

Clarification entre Portfolio, Programmes, Projet, Production, Gestion de Projet, Gestion de Produit, Périmètre du Produit, Périmètre du Projet

### **Gestion de Projet :**

Distinction entre Projet Classique et Projets Agile (Itératif, Incrémental, Simple/Complexe

### **Critères de choix d'une approche (Classique/Agile), domaines d'applications et risques spécifiques**

Serious Game « Choisir la bonne température dans une pièce »

### **Mythes sur le choix d'Agile comme approche, sur l'application d'Agile et sur ses risques spécifiques**

### **Facteurs clés de succès d'une implémentation réussie d'Agile**

Serious Game « Fatburger » - « Agile Family Management » vidéo

## Origine of Agile Culture

- Agilemanifesto.org (4 values, 12 Principles)
- Agile → Practices → Frameworks (Scrum, XP, Kanban, SAFe)
- Differences among Culture, Framework, Methodology, Langage and Tools
- Agile Frameworks Map
- Agile (2001 - Software) derived from Lean (1950 - Manufacturing)
- Agile is...
- Why Agile ?
- Agile approach VS Classical approach
- Serious Game: « IceBreaker Agile »

## L'origine de la culture Agile

- Agilemanifesto.org (4 valeurs, 12 Principes)
- Agile : des pratiques vers des Frameworks (Scrum, XP, Kanban, SAFe)
- Différences entre Culture, Framework (cadre de travail), Méthodologie, Langage et Outils
- Cartographie des Frameworks Agile
- Agile (2001 - Software) dérivé de Lean (1950 - Manufacturing)
- Agile est ...
- Pourquoi Agile ?
- L'approche Agile VS l'approche classique
- Serious Game: « IceBreaker Agile »

## Agilemanifesto.org (established in 2001)

The screenshot shows the homepage of AgileManifesto.org. At the top, it says "The AGILE Manifesto" and "12 Principles of Agile Software". Below that is a list of 12 principles, each numbered and followed by a brief description. The principles are:

- 01** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 02** Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 03** Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
- 04** Business people and developers must work together daily throughout the project.
- 05** Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 06** Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 07** Working software is the primary measure of progress.
- 08** The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 09** Continuous attention to technical excellence and good design enhances agility.
- 10** Simplicity—the art of maximizing the amount of work not done—is essential.
- 11** The best architectures, requirements, and designs emerge from self-organizing teams.
- 12** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

At the bottom left, it says "© Global Knowledge Training LLC. All rights reserved." and at the bottom right, "Page 9".

<https://agilemanifesto.org/>

## Agile Manifesto : 4 values

- We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- That is, while there is value in the items on the right, we value the items on the left more.

## Manifeste Agile : 4 valeurs

Nous découvrons de meilleures façons de développer un logiciel en le faisant et en aidant les autres à le faire.

Grâce à ce travail, nous avons pu valoriser:

- Priorité donnée aux personnes et aux interactions, plutôt qu'aux processus et aux outils
- Priorité donnée à la production de logiciels opérationnels, plutôt qu'à une documentation exhaustive
- Priorité donnée à la collaboration avec le client, plutôt qu'à la négociation contractuelle
- Priorité donnée à l'adaptabilité et à l'accueil des éventuels changements, plutôt qu'au suivi d'un plan

C'est-à-dire que, s'il y a de la valeur dans les éléments « de droite », priorité donnée à ... , nous valorisons les éléments « de gauche » , plutôt que....

## Agile Manifesto : 12 Principles (1-5)

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development, Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

## Manifeste Agile : 12 principes (1-5)

- 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.

## Agile Manifesto : 12 Principles (6-12)

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity - the art of maximizing the amount of work not done - is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

© Global Knowledge Training LLC. All rights reserved.

Page 12

## Manifeste Agile : 12 principes (6-12)

- 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.

Agile → 4 values, 12 principles → Practices → Frameworks (Scrum, XP, Kanban, SAFe, etc.)

## Agile is a behaviour (Culture)

### 4 VALUES BEHAVIOURS

- Individus et interactions
- Collaboration
- Respond to change
- Respect for people

### 12 PRINCIPLES BEHAVIOURS

- Transparency
- Inspection
- Adaptation
- Empiricism
- Trust
- Eliminate waste
- Deliver for fast feedback
- Build in quality
- Build organisational knowledge
- Optimise for the whole

### MANY PRACTICES BEHAVIOURS

- Value-centred
- User-centred
- Optimised for fast feedback

### NUMEROUS FRAMEWORKS BEHAVIOURS

- Using a defined framework
- Disciplined in its execution
- Utilised for continuous improvement
- Utilised for continuous feedback loops



Agile → 4 valeurs, 12 principes → Pratiques → Frameworks (Scrum, XP, Kanban, SAFe, etc.)

Agile est un comportement (une culture)

**4 valeurs :**

- Individus et interactions
- Collaboration
- Réponse aux changements
- Respect des personnes

**12 Principes**

- |                |  |
|----------------|--|
| • Transparence | • Éliminer le gaspillage               |
| • Inspection   | • Fournir un retour rapide             |
| • Adaptation   | • Réaliser un travail de qualité       |
| • Empirisme    | • Construire une base de connaissances |
| • Confiance    | • Optimiser pour l'ensemble            |

**Plusieurs pratiques**

- Orientée valeurs
- Orientée utilisateurs
- Optimisée pour un retour rapide

**Nombreux frameworks (cadres de travail) :**

- Utiliser un cadre de travail défini
- Se discipliner sur son utilisation
- L'utiliser pour une amélioration continue

- L'utiliser pour des boucles de feedback continues

## Differences among Culture, Framework, Methodology Langage and Tools



© Global Knowledge Training LLC. All rights reserved.

Page 14

**Différences entre langages et outils liés à la culture, les frameworks, les méthodes**

[https://fr.wikipedia.org/wiki/Foxit\\_Reader](https://fr.wikipedia.org/wiki/Foxit_Reader)

[https://fr.wikipedia.org/wiki/VLC\\_media\\_player](https://fr.wikipedia.org/wiki/VLC_media_player)



### Cartographie des frameworks Agile

Agile : rester simple et léger

Axe des ordonnées : Étendue du cycle de vie

Axe des abscisses : Profondeur de guidage

## Agile (2001 - Software) derived from Lean (1950 - Manufacturing)



© Global Knowledge Training LLC. All rights reserved.

Page 16

Formateur : [https://en.wikipedia.org/wiki/Lean\\_manufacturing](https://en.wikipedia.org/wiki/Lean_manufacturing)

Stagiaire : **Agile (2001- Logiciel) dérivé de Lean (1950 – Industrie)**

## When am I Agile ?

⌚ 2 min.

- What have you heard ?
- Based on what we have reviewed, what do you think ?
- According to the Instructor.

© Global Knowledge Training LLC. All rights reserved.

Page 17

## Quand suis-je Agile ?

Qu'avez-vous entendu ?

D'après ce que nous avons examiné, qu'en pensez-vous?

D'après votre formateur ?

## Agile is...

- A philosophy/culture about software development
- A collection of processes and practices that uphold that philosophy
- A grassroots movement to change software development
- Key terms : Iterative, incremental, progressive elaboration, empowering, collaborative

## Agile est ....

- Une philosophie / culture issue initialement du développement de logiciels
- Une collection de process et de pratiques qui maintiennent cette philosophie
- Un mouvement populaire pour faire évoluer le développement de logiciels
- Mots clés : **Itératif, incrémental, élaboration progressive, autonomie, collaboration**

## Why Agile ?

### ➤ Expected Results from Agile

1. Manage rapidly changing priorities
2. Increase Productivity
3. Enhance Quality
4. Accelerate Time-to-Market

© Global Knowledge Training LLC. All rights reserved.

13 / 19

## Pourquoi Agile ?

- 1.Gérer la priorité des changements
- 2.Augmenter la productivité
- 3.Améliorer la qualité
- 4.Accélérer les délais de commercialisation et le temps de mise sur le marché



Formateur : Expliquer les avantages en 4 dimensions (si Agile est bien appliqué)

### Stagiaire : L'approche Agile VS l'approche classique

Développement Agile : Proposition de valeurs

- Visibilité
- Adaptabilité
- Valeur business
- Risque

## What is Agility

⌚ 3 min.

- Purpose: Working with Management



Hi,  
Heard lots of good things  
about your team.  
Want to know more on  
Agile.  
See you in 3 minutes.

Your CEO

Question: Explain to the CEO what « Agile » is about.

© Global Knowledge Training LLC. All rights reserved.

Page 21

## Qu'est ce qu'Agile

### But : Travail avec le management

Bonjour,

J'ai entendu beaucoup de bonnes choses sur votre équipe.

Je veux en savoir plus sur Agile.

On se voit dans 3 minutes.

Votre PDG

### Question : Expliquer au PDG ce qu'est « Agile »

## Serious Game: « IceBreaker Agile » ☰ 10 min.

- It is asked groups (3 to 4 participants) to choose from 24 images,
- 3 images each representing for them Agile Culture,
- Then quantify (1 to 10) their satisfaction to make their 3 choices.

Formateur : utiliser 24 images dans un dossier « Ice Breaker New »

Stagiaire : **Serious Game « Ice breaker Agile »**

- Il est demandé aux groupes (3 à 4 participants) de choisir parmi 24 images.
- 3 images représentant chacune pour eux la culture Agile.
- Quantifier ensuite (1 à 10) leur satisfaction de faire leurs 3 choix.

## Serious Game (debrief)

### ➤ Lessons after Serious Game:

- All agree first on process to use
- Take input from all (N smarter than N-1)
- Value Team ( $N \times (N-1)/2$ ) > Value Group (N)

## Debrief du Serious Game

### A retenir après le SG :

- Se mettre tous d'accord sur le process à utiliser
- Prendre en compte les inputs de toutes les personnes
- La valeur de l'équipe > La valeur du groupe

## Overview of Agile Frameworks

- Lean
- Kanban (Methodology)
- Scrum (ScrumBan)
- XP
- DSDM
- FDD
- SAFe
- LeSS
- Nexus Scrum
- Lean Startup
- Factors leading to the right Agile approach
- Main Agile certifications (unscaled)

© Global Knowledge Training LLC. All rights reserved.

Page 24

## Panorama des principales approches Agile

## Lean Software Development

- Adapted from Lean principles in manufacturing (applied first by Toyota in 1953).
- Focus on eliminating waste while increasing delivered value.
- Make improvements by focusing on the Value Stream.

### ➤ Principles

- Eliminate Waste
- Create Knowledge
- Build Quality In
- Deliver Fast
- Respect People
- Improve the System

## Lean embodied in Scrum

Formateur : [https://en.wikipedia.org/wiki/Lean\\_manufacturing](https://en.wikipedia.org/wiki/Lean_manufacturing)  
[https://en.wikipedia.org/wiki/Lean\\_software\\_development](https://en.wikipedia.org/wiki/Lean_software_development)

### Stagiaire : Développement de logiciels Lean

- Dérivé du Lean manufacturing (Appliqué la 1<sup>ère</sup> fois chez Toyota en 1953)
- Met l'accent sur l'élimination des gaspillages tout en augmentant la valeur ajoutée livrée.
- Apporte des améliorations en vous concentrant sur la chaîne de valeur du produit.
- Principes :
  - Éliminer les gaspillages
  - Créer de la connaissance
  - Renforcer la qualité
  - Livrer rapidement
  - Respecter les personnes
  - Améliorer le système

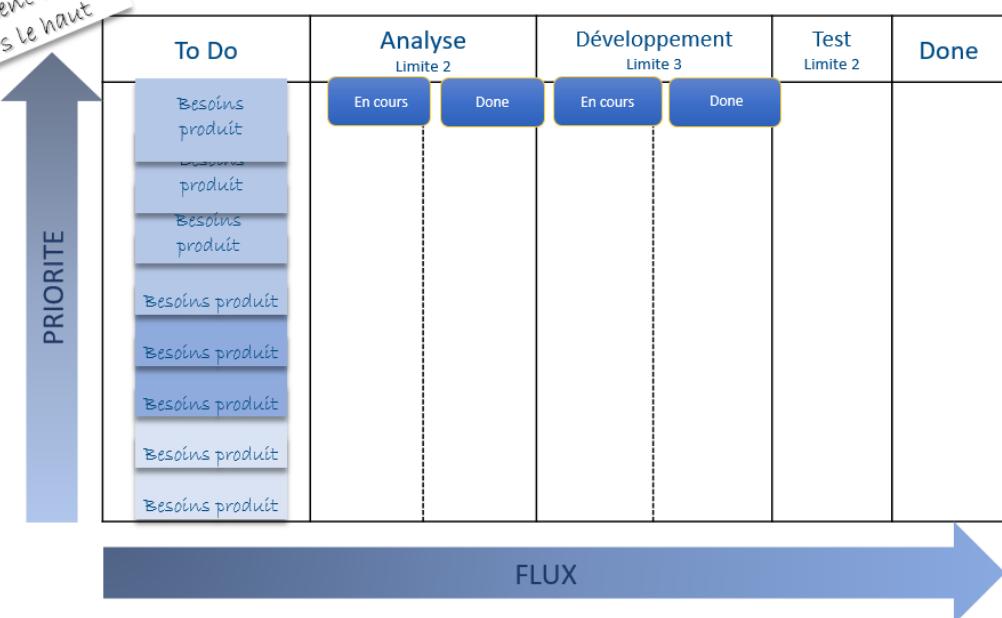
Le Lean est intégré dans Scrum

## Kanban (1)

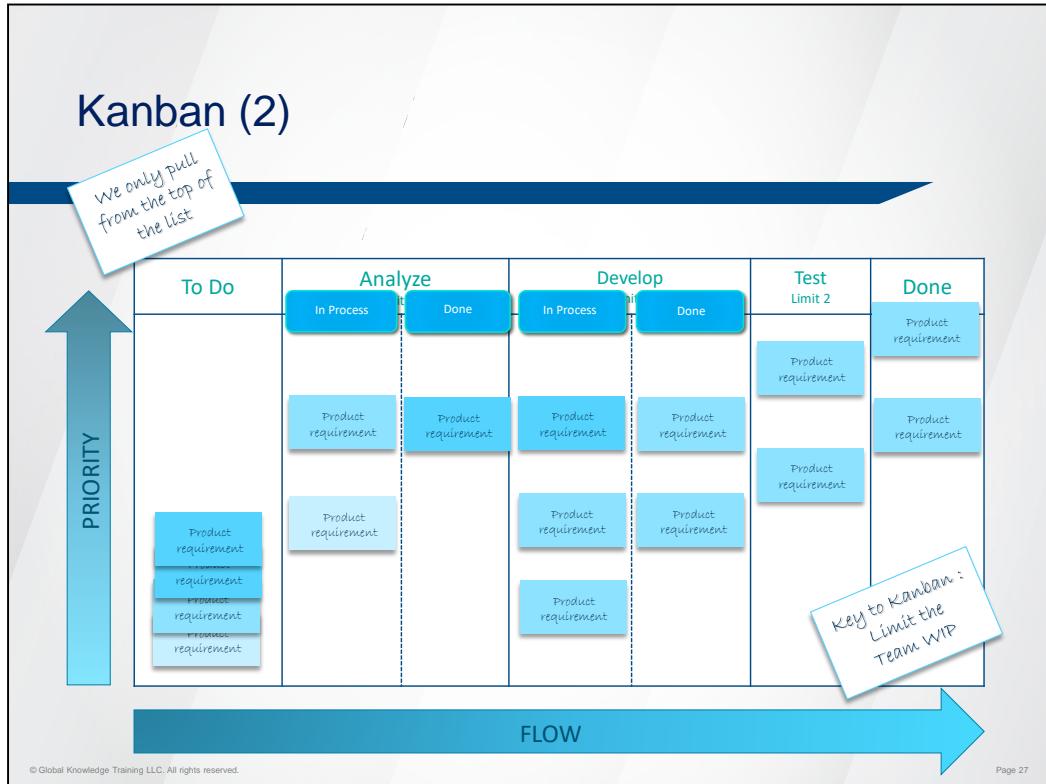


Formateur : <https://en.wikipedia.org/wiki/Kanban>

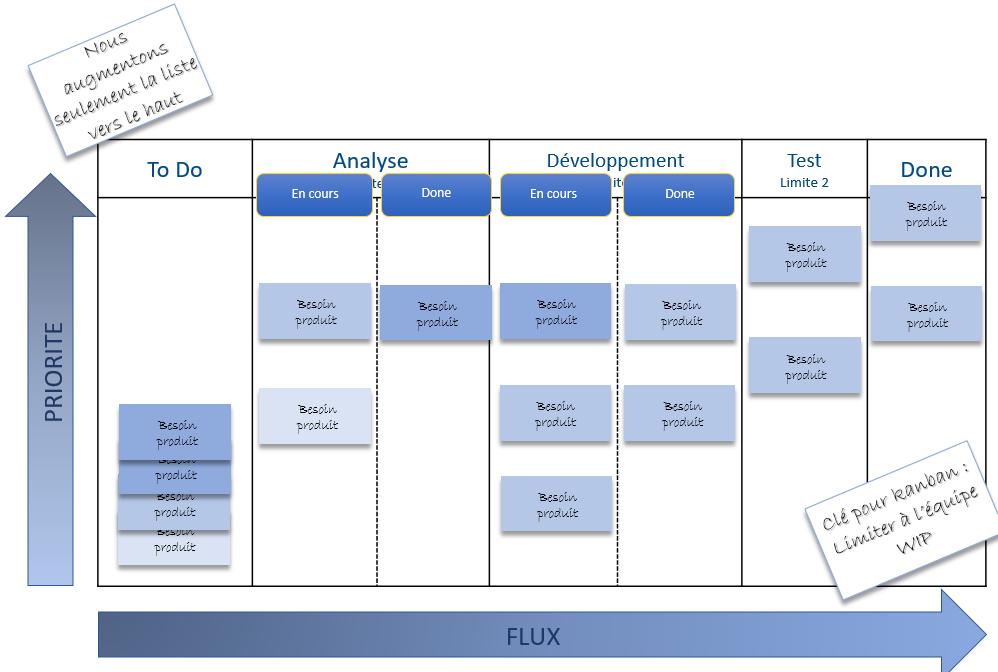
Stagiaire : **Kanban (1)**







## Kanban (2)



## Scrum Components (1)

- 3 PILARS
  - Transparency
  - Inspection
  - Adaptation
- 5 VALUES
  - Focus
  - Respect
  - Openness
  - Courage
  - Commitment
- 3 ROLES
  - Product Owner
  - Development Team
  - Scrum Master

© Global Knowledge Training LLC. All rights reserved.

Page 28

## Les composants Scrum (1)

**3 piliers** : Transparence, Inspection et Adaptation

**5 valeurs** : Focus, respect, ouverture, courage, engagement

**3 rôles** : Product Owner, équipe de développement et Scrum Master

## Scrum Components (2)

- 5 EVENTS
  - Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective
- 3 ARTIFACTS
  - Product Backlog
  - Sprint Backlog
  - Increment
- 1 Definition of Done (DoD)

© Global Knowledge Training LLC. All rights reserved.

Page 29

## Les composants Scrum (2)

**5 événements** : Sprint, Sprint Planning (la planification de Sprint), Daily Scrum (la mêlée quotidienne ), Sprint Review (la revue de Sprint), Sprint Retrospective (la rétrospective du Sprint)

**3 artefacts** : Product Backlog, Sprint Backlog et l'Incrément

Une définition du « **Done** »



Formateur :

[https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-04/2018%20Kanban%20Guide%20for%20Scrum%20Teams\\_0.pdf](https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-04/2018%20Kanban%20Guide%20for%20Scrum%20Teams_0.pdf)

## XP : Taking technical best practices to an extreme

### ➤ Value

- Simplicity
- Communication
- Respect
- Feedback
- Courage

### ➤ Key Practices :

- Planning Game
- Small Releases
- Acceptance Tests
- Simple Design
- Pair Programming
- Test-Driven Development
- Refactoring
- Continuous Integration
- Collective Ownership
- Coding Standards
- Metaphor
- Sustainable Pace

Formateur : [https://en.wikipedia.org/wiki/Extreme\\_programming](https://en.wikipedia.org/wiki/Extreme_programming)

Stagiaire : **XP : Mettre à l'extrême les meilleures pratiques techniques**

### **Valeur :**

#### **•Simplicité:**

- La chose la plus simple qui puisse marcher
- Éviter la complexité inutile dans le code
- Toute duplication doit être éliminée

#### **Communication:**

Développement = Effort collectif de création, d'où le besoin d'avoir une vision commune et pouvoir se synchroniser.

Qualité de la communication = Communication directe et contact humain afin d'augmenter la réactivité

**Respect:** cette valeur inclut le respect pour les autres, ainsi que le respect de soi. Les programmeurs ne devraient jamais valider les modifications qui cassent la compilation, qui font échouer les tests unitaires existants ou qui retardent le travail de leurs pairs. Les membres respectent leur propre travail en cherchant toujours la qualité et la meilleure conception pour la solution et cela grâce au refactoring.

## XP Roles

➤ Standard Roles :

- Customer
- Developer

➤ Extended Roles :

- Tracker
- Coach

## Les rôles XP

**Rôles standards :** Client, développeur

**Rôles avancés :** Suiveur, coach

## Dynamic Systems Development Method (DSDM)

- DSDM framework consists of three sequential phases:
- Pre-project
  - Funding is realized and project commitment is ensured
- Project life-cycle ; 5 stages that form an iterative step-by-step
  1. The feasibility Study
  2. The Business Study
  3. Functional Model Iteration
  4. Design and Build Iteration
  5. Implementation
- Post-project phases
  - Maintenance, enhancements and fixes

© Global Knowledge Training LLC. All rights reserved.

Page 34

Formateur :

[https://en.wikipedia.org/wiki/Dynamic\\_systems\\_development\\_method](https://en.wikipedia.org/wiki/Dynamic_systems_development_method)

Stagiaire : **Dynamic Systems Development Method (DSDM)**

Le framework DSDM se compose de 3 phases séquentielles :

- **Le pré projet**
  - Le financement est réalisé et l'engagement du projet est assuré

## Feature Driven Development (FDD)

### ➤ Five Activities

1. Develop Overall Model
2. Build Feature List
3. Plan by Feature
4. Design by Feature
5. Build by Feature

### ➤ Key Practices :

- Domain Object Modeling
- Develop by Feature
- Component/Class Ownership
- Feature Teams
- Inspections
- Configuration Management
- Regular Build
- Visibility of Progress

Formateur : [https://en.wikipedia.org/wiki/Feature-driven\\_development](https://en.wikipedia.org/wiki/Feature-driven_development)

Stagiaire : **Feature Driven Development (FDD)**

**Cinq activités :**

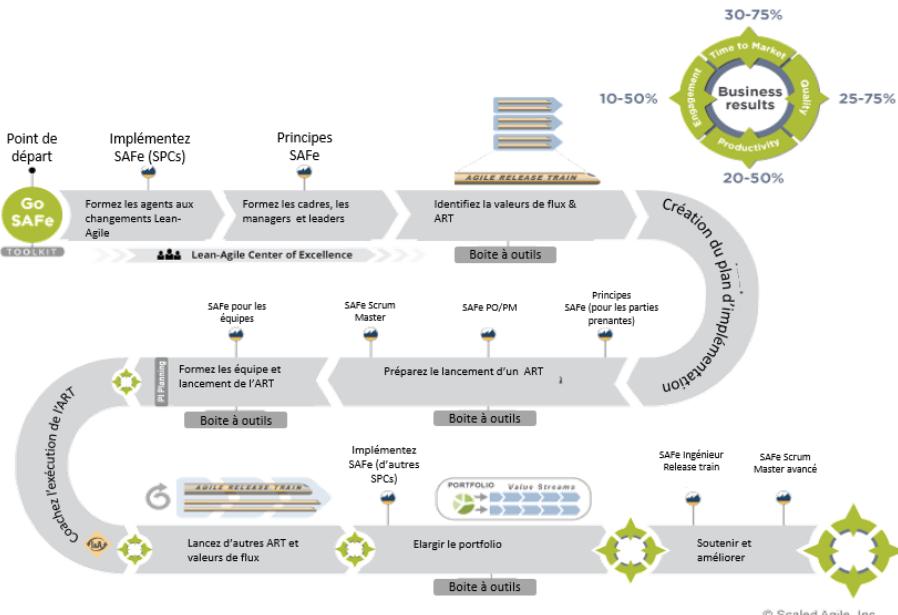
1. Élaborer un modèle global
2. Établir une liste de fonctions
3. Planifier par fonction
4. Concevoir par fonction
5. Construire par fonction

**Points clés :** Modélisation des objets de domaine, développement par fonction, composante/classe de propriété, équipes par fonction, inspections, gestion de la configuration, construction régulière, visibilité du progrès



Formateur : <https://www.scaledagileframework.com/>

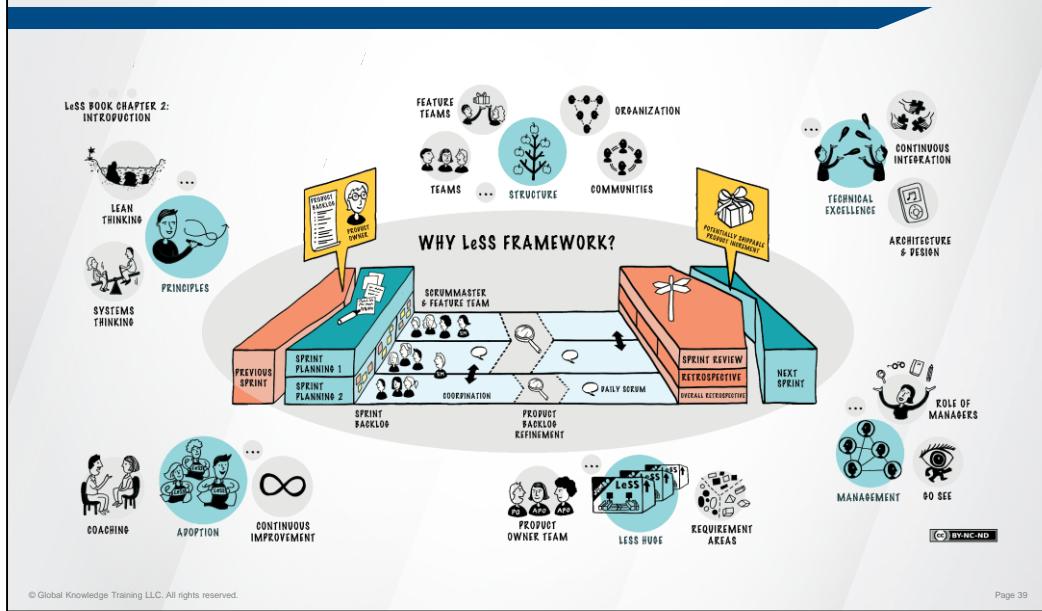
### Stagiaire : SAFe : Feuille de route d'implémentation





Formateur : [https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-01/2018-Nexus-Guide-English\\_0.pdf?nexus-file=https%3A%2F%2Fscrumorg-website-prod.s3.amazonaws.com%2Fdrupal%2F2018-01%2F2018-Nexus-Guide-English\\_0.pdf](https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-01/2018-Nexus-Guide-English_0.pdf?nexus-file=https%3A%2F%2Fscrumorg-website-prod.s3.amazonaws.com%2Fdrupal%2F2018-01%2F2018-Nexus-Guide-English_0.pdf)

## LeSS (Scaled Agile)



Formateur : <https://less.works/less/framework/introduction.html>  
 Stagiaire : LeSS



## Lean Startup

### Combine Design Thinking, Lean Startup and Agile



Gartner

© Global Knowledge Training LLC. All rights reserved.

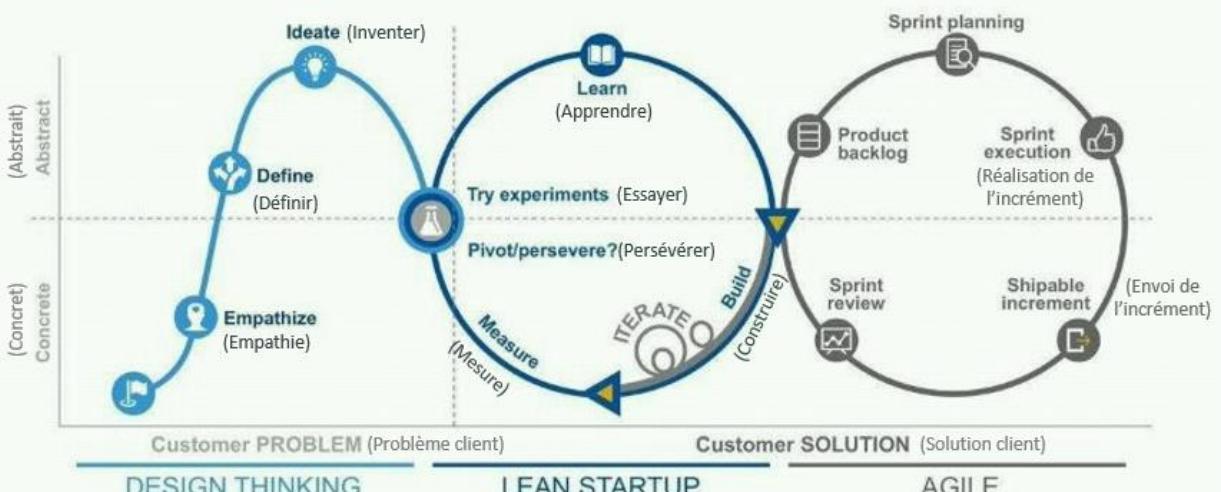
Page 40

<http://theleanstartup.com/principles>

### Stagiaire : Lean Startup

Combiner le « Design Thinking », le « Lean Startup » et Agile

### Combine Design Thinking, Lean Startup and Agile



Gartner

#GartnerSYM

© CONFIDENTIAL AND PROPRIETARY © 2014 Gartner, Inc. or its affiliate(s). All rights reserved. Gartner and ITsym are registered trademarks of Gartner, Inc. or its affiliate(s).

## Factors leading to the right Agile approach

| Factor  | XP                                   | Scrum                                | Scrum-Lean                                    | Kanban                                |
|---|--------------------------------------|--------------------------------------|---|---------------------------------------|
| Keeping team intact                             | Prescribed                           | Prescribed                           |   |                                       |
| Use time boxed intervals                        | Yes                                  | Yes                                  | Yes   | No                                    |
| Prioritize users stories across a team          | Yes                                  | Yes                                  | Yes   | No                                    |
| When to release completed work                  | At the end of the selected iteration | At the end of the selected iteration | At the end of the selected iteration          | Whenever at the discretion of team    |
| Works in Support Environment                    | No                                   | No                                   | No  | Yes                                   |
| Co-Located Teams                                | No guidance                          | No guidance                          | Use fast-flexible-flow to create optimal flow | Manage with appropriate WIP limits    |
| Support for the product management organization | No                                   | No                                   | yes   | Partial                               |
| Code Quality                                    | Yes                                  | Not discussed                        | Use a workflow that increases quality         | Use a workflow that increases quality |

© Global Knowledge Training LLC. All rights reserved.

Page 41

## Facteurs menant à la bonne approche agile

| Facteur  | XP                                   | Scrum                                | Scrum-Lean  | Kanban  |
|--|--------------------------------------|--------------------------------------|---|---|
| Garder l'équipe intacte                                | Prescrit                             | Prescrit                             |   |   |
| Utiliser des intervalles de temps prédéfinis           | Oui                                  | Oui                                  | Oui   | Non   |
| Prioriser les Items ou User Stories à travers l'équipe | Oui                                  | Oui                                  | Oui   | Non   |
| Définir quand le travail est terminé                   | A la fin de l'itération sélectionnée | A la fin de l'itération sélectionnée | A la fin de l'itération sélectionnée                              | A n'importe quel moment, suite à la demande de l'équipe |
| Travailler dans un support                             | Non                                  | Non                                  | Non   | Oui   |
| Equipes colocalisées                                   | Pas de directive                     | Pas de directive                     | Utilisation d'un flux adaptatif et rapide pour optimiser les flux | Gestion avec des limites d'en-cours appropriées         |
| Soutien de l'organisation pour la gestion de produit   | Non                                  | Non                                  | Oui   | Partiel   |
| Qualité du code  | Oui                                  | Pas discutée                         | Utilisation d'un workflow pour augmenter la qualité               | Utilisation d'un workflow pour augmenter la qualité     |

\*WIP = Work in progress

## Main Agile certifications (unscaled)

| Criteria              | PSM                                | CSM                        | PMI-ACP                       |
|-----------------------|------------------------------------|----------------------------|-------------------------------|
| <b>Knowledge</b>      | Scrum                              | Scrum                      | Overall Agile                 |
| <b>Popularity</b>     | High                               | High                       | Medium                        |
| <b>Pre-requisites</b> | NA                                 | NA                         | Previous Experience           |
| <b>Difficulty</b>     | Intermediate                       | Easy                       | Intermediate                  |
| <b>Duration</b>       | 1 hour                             | No limit                   | 3 hours                       |
| <b>Passing grade</b>  | 68/80 questions (85%)              | 24/35 questions (70%)      | 84/120 questions (70%)        |
| <b>Online</b>         | Yes                                | Yes                        | Local premise of PMI partners |
| <b>Cost</b>           | 150\$ (offert avec suivi de cours) | 1000\$ (Incluant le cours) | 495\$                         |
| <b>Renewal need</b>   | No                                 | Every 2 years              | Every 3 years                 |
| <b>Renewal Cost</b>   | No                                 | 100\$                      | 150\$                         |

© Global Knowledge Training LLC. All rights reserved.

Page 42

## Les principales certifications Agile (non mis à l'échelle)

| Critère                | PSM                                | CSM                        | PMI-ACP                    |
|------------------------|------------------------------------|----------------------------|----------------------------|
| Connaissance           | Scrum                              | Scrum                      | Tout Agile                 |
| Popularité             | Haute                              | Haute                      | Moyenne                    |
| Prérequis              | NA                                 | NA                         | Expérience dans le domaine |
| Difficulté             | Intermédiaire                      | Facile                     | Intermédiaire              |
| Durée                  | 1 heure                            | Pas de limite              | 3 heures                   |
| Score d'obtention      | 68/80 questions (85%)              | 24/35 questions (70%)      | 84/120 questions (70%)     |
| En ligne               | Oui                                | Oui                        | Centre local agréé PMI     |
| Coût                   | 150\$ (offert avec suivi de cours) | 1000\$ (Incluant le cours) | 495\$                      |
| Renouvellement ?       | Non                                | Tous les 2 ans             | Tous les 3 ans             |
| Coût du renouvellement | Non                                | 100\$                      | 150\$                      |

## Scrum

- Scrum :
  - 3 pilars : Transparency, Inspection, Adaptation
  - 5 values : Focus, Respect, Openness, Commitment, Courage
  - 3 roles : Product Owner, Development Team, Scrum Master
  - 5 events : Sprint Planning, Daily meeting, Sprint review, Sprint retrospective, Sprint
  - 3 artifacts : Product Backlog, Sprint Backlog, Increment
  - 1 Definition of Done, Release
- Scrum.org <https://www.scrum.org/>
- Scrum Guide  
<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
- Open Assessments <https://www.scrum.org/resources/scrum-guide>

© Global Knowledge Training LLC. All rights reserved.

Page 43

Formateur : <https://www.scrum.org/>

Stagiaire : **Scrum**

3 piliers: Transparency, Inspection, Adaptation

5 valeurs: Focus, Respect, Ouverture, Engagement, Courage

3 rôles: Chef de produit (Product owner), Equipe de développement, Scrum Master

5 événements: planification du sprint, réunion quotidienne, revue du sprint, rétrospective du sprint, sprint

3 artefacts: Product Backlog, Sprint Backlog, Incrément

1 Définition de Done, Livraison

Scrum.org <https://www.scrum.org/>

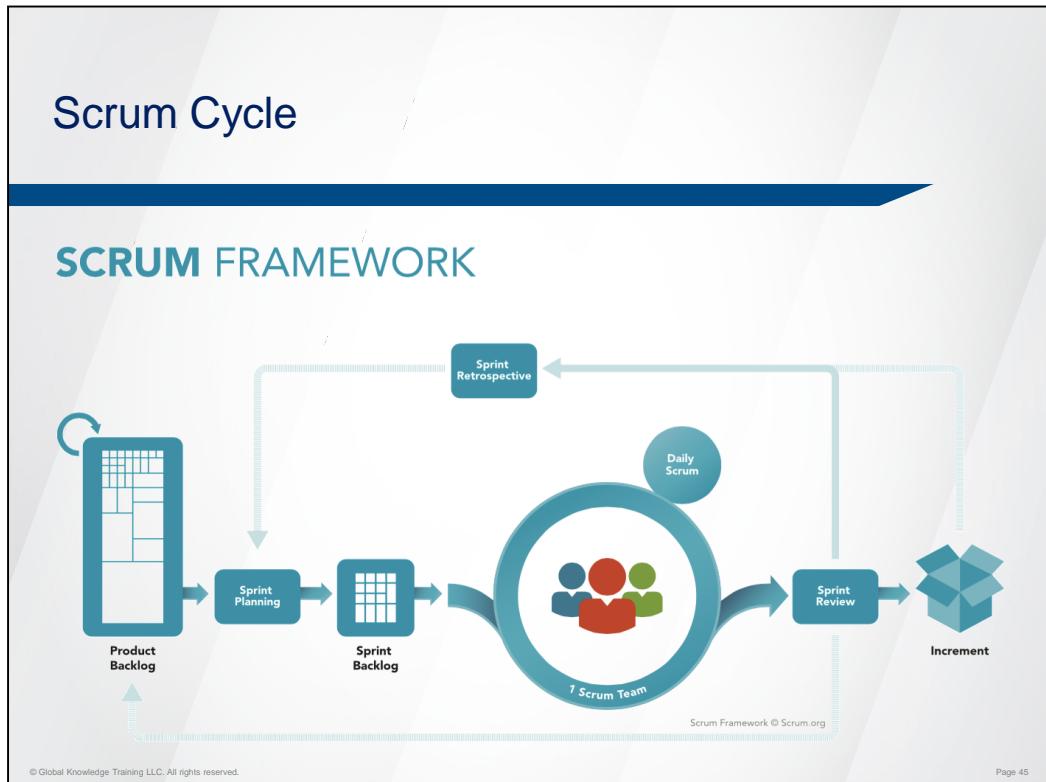
Scrum Guide <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>

Evaluations disponible : <https://www.scrum.org/resources/scrum-guide>

## Scrum (Origins)



[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))



## Le cycle Scrum

## Purpose of Scrum /

### ➤ Framework

1. Helping people address complex problems.
2. Productively and creatively deliver products of the highest possible value.

- Simple to understand
- Difficult to master



© Global Knowledge Training LLC. All rights reserved.

Page 46

## But de Scrum

Le framework :

1. Aider les personnes à traiter des problèmes complexes
  2. Livrer de façon productive et innovante des produits avec la plus haute valeur ajoutée possible
- 
- Simple à comprendre
  - Difficile à maîtriser

## AGILE - SCRUM Structure

- SCRUM ((tripod))

| <b>Product Owner</b> | <b>Scrum Master</b> | <b>Development Team</b> |
|----------------------|---------------------|-------------------------|
| WHAT                 | Framework           | HOW                     |
| Right Thing          | Right Process       | Thing Right             |

- Scrum is a Framework: like chess, it exists different ways to play...

© Global Knowledge Training LLC. All rights reserved.

Page 47

Formateur : Right thing specified, Thing developed right, using right Framework -> Right Value

### Stagiaire : Agile – Structure Scrum

|                |              |                        |
|----------------|--------------|------------------------|
| Product Owner  | Scrum Master | Equipe de développeurs |
| QUOI           | Framework    | COMMENT                |
| Bonne décision | Bon process  | Bon développement      |

Scrum est un framework : Comme aux échecs, il existe plusieurs façons de jouer.

## Scrum is a Tool Used to Become Agile

### ➤ Short, high value iterations :

- To deliver valuable opportunistic pieces of functionality
- By self-organizing, cross-functional teams.



Scrum Team

© Global Knowledge Training LLC. All rights reserved.

Page 48

## Scrum est un outil pour devenir Agile

- Court, avec des itérations à valeurs élevées :
  - Fournir des fonctionnalités à valeur
  - Par des équipes auto-organisées et multi-fonctionnelles





Formateur : An Agile project can be seen as several Classical projects

Stagiaire : **Scrum livre plus vite que les projets en « cascade » (classiques)**

- Logiciel fonctionnel est disponible

## Change management in the company

- Clarification between :
  - Portfolio
  - Program
  - Project, Project / Operation
- Product Scope / Project Scope
- Project / Change ➔ AGILE Scrum implementation
- Project = Change

© Global Knowledge Training LLC. All rights reserved.

Page 50

### **Conduite du changement dans l'entreprise**

Clarification entre:

- Portfolio,
- Programme,
- Projet, Projet / Opération

Périmètre du Produit / Périmètre du Projet

Projet / Changement ➔ Implémentation de Scrum Agile

Projet = Changement

## Portfolio, Program, Project

|                  | <b>Business Analysis</b>   | <b>Project Management</b>  | <b>Program Management</b>  | <b>Portfolio Management</b>  |
|------------------|--|--|--|--|
| Definition       | The set of activities performed to support the delivery of solutions that align to business objectives and provide continuous value to the organization                                      | The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements           | The application of knowledge, skills, and principles to a program to achieve the program objectives and obtain benefits and control not available by managing program components individually    | The centralized management of one or more portfolios to achieve strategic objectives   |
| Focus            | <b>Solution:</b> Something that is produced to deliver measurable business value to meet the business need and expectation of stakeholders (e.g. new products and enhancements to products). | <b>Project :</b> A temporary endeavor undertaken to create a unique product, service, or result                              | <b>Program :</b> A group of related projects, subsidiary programs, and program activities that are managed on a coordinated way to obtain benefits not available from managing them individually | <b>Portfolio :</b> A collection of projects, programs, subsidiary portfolios, and operations managed as a group to achieve strategic                       |
| Scope definition | Product scope: The features and function that characterize a solution  | <b>Project scope :</b> The work performed to deliver a product, service, or result with the specified features and functions | <b>Program scope :</b> The scope that encompasses program components and the interaction and synergy between them  | <b>Portfolio scope:</b> The organizational scope that changes with the strategic objectives of the organization  |
| Roles            | Those who identify business needs, and recommend and describe solutions through the definition of product requirements   | Those who manage the project team to meet the project objectives,  | Those who ensure that program benefits are delivered as expected by coordinating the activities of program's components  | Those who coordinate portfolio management staff, or program and project staff that may have reporting responsibilities into the aggregate of the portfolio |
| Success          | Measured by a solution's ability to deliver its intended benefits to an organization, degree of customer satisfaction and achievement of business objectives.                                | Measured by product and project quality, timelines, budget, compliance and degree of customer satisfaction.                  | Measured by program's ability to deliver its intended benefits to an organization, and by the program's efficiency and effectiveness in delivering those benefits.                               | Measured in terms of the aggregate investment performance and benefit realization of the portfolio   |

## Portfolio, Programme, Projet

|                     | <b>Analyse du business</b>  | <b>Gestion de projet</b>  | <b>Gestion du programme</b>  | <b>Gestion du portefeuille</b>   |
|---------------------|---|---|--|--|
| Définition          | Ensemble d'activités réalisées pour appuyer la prestation de solutions qui cadrent avec les objectifs opérationnels et procurent une valeur continue à l'organisation   | Application des connaissances, des compétences, des outils et des techniques aux activités liées aux projets pour répondre aux exigences du projet        | Application de connaissances, compétences et principes pour atteindre les objectifs du programme et obtenir des avantages et un contrôle qui ne sont pas disponibles en gérant individuellement les composantes du programme   | La gestion centralisée d'un ou de plusieurs portefeuilles pour atteindre les objectifs stratégiques  |
| Focus               | <b>Solution:</b> Produit réalisé pour offrir une valeur opérationnelle mesurable afin de répondre aux besoins opérationnels et aux attentes des différentes parties prenantes (ex., nouveaux produits et améliorations des produits). | <b>Projet :</b> Un effort temporaire entrepris pour créer un produit, un service ou un résultat unique  | <b>Programme :</b> Groupe de projets connexes, de programmes subsidiaires et d'activités de programme qui sont orchestrés comme un ensemble, de sorte à obtenir des avantages qui ne seraient pas réalisables si chaque élément était coordonné de manière individuelle. | <b>Portefeuille :</b> Un ensemble de projets, de portefeuilles subsidiaires de programmes et d'opérations gérés en groupe  |
| Définition du Scope | <b>Périmètre du produit :</b> Les caractéristiques et la fonction qui caractérisent une solution  | <b>Périmètre du projet :</b> Le travail effectué pour livrer un produit, un service ou un résultat répondant aux caractéristiques et fonctions spécifiées | <b>Périmètre du programme :</b> Portée qui englobe les composantes du programme et l'interaction et la synergie entre elles  | <b>Périmètre du portefeuille :</b> Portée organisationnelle qui change en fonction des objectifs stratégiques de l'organisation  |
| Rôles               | Ceux qui détermineront les besoins de l'entreprise, et qui recommanderont et décriront des solutions en définissant les exigences du produit  | Ceux qui gèrent l'équipe de projet pour atteindre les objectifs.  | Ceux qui veillent à ce que les prestations du programme soient offertes comme prévu en coordonnant les activités des composantes du programme  | Les personnes qui coordonnent la gestion du portefeuille, ou le responsable projet qui peuvent avoir des responsabilités en matière de rapports sur l'ensemble du portefeuille |
| Réussite            | Mesurée par la capacité d'une solution d'offrir les avantages escomptés à une organisation, le degré de satisfaction de la clientèle et l'atteinte des objectifs opérationnels  | Mesurée par la qualité des produits et des projets, les échéanciers, le budget, la conformité et le degré de satisfaction de la clientèle.                | Mesurée par la capacité du programme d'offrir les avantages escomptés à une organisation, et par l'efficience et l'efficacité du programme dans la prestation de ces avantages.  | Mesurée en termes de rendement global des placements et de réalisation des avantages du portefeuille   |

## Project / Operation



Formateur : Introduce DevOps

## Stagiaire : Projet / Opération

## Gestion de projet

## Activités temporaires dans le cadre du projet

### Sortie unique

Terminé quand l'objectif est atteint  
ensemble d'objectifs quand le précédent est terminé

## Gestion des opérations

## Activité au jour le jour

## Sorties répétitives

### Adopter un nouveau

## Product Scope / Project Scope

### Project Scope vs Product Scope

| Project scope                                    | Product Scope  |
|--|--|
| The work needed to create product of the project | Features and functions of the product of the project         |
| Completion of a project against the plan         | Completion of product scope is measured against requirements |

## Périmètre produit / Périmètre projet

### Périmètre projet vs périmètre produit

| Périmètre Projet   | Périmètre Produit  |
|--|--|
| Le travail nécessaire à la création d'un produit du projet | Fonctionnalités et fonctions du produit du projet                      |
| L'achèvement du projet est mesuré par rapport au plan      | L'achèvement de l'étendue produit est mesuré par rapport aux exigences |

## Project / Change -> AGILE Scrum implementation

### The Kübler-Ross change curve



© Global Knowledge Training LLC. All rights reserved.

Page 54

Formateur : Lifecycle Change (Project) // Lifecycle Change (Mourning)

Stagiaire : La courbe du changement Kübler-Ross



## Project = Change

### ➤ Criteria for choosing a Project Management approach

- Vision for Change
- Cycle time (Frequency of breaks)
- Scale of disruptions (type 1/ type 2)

## Projet = Changement

Critère pour choisir l'approche de gestion de projet

- Vision du changement
- Cycle de vie ( Fréquence des ruptures)
- Échelle de perturbations (type 1 / type 2)

## Project Management

- What is Project Management ?
- « Cowboys herding cats » video
- Project Management
- The triple constraints
- Paradigm Change
- Agile = Iterative + Incremental
- Predictive - Iterative - Incremental - Agile

### Gestion de projet

- Qu'est ce que la gestion de projet
- Vidéo « Cowboys herding cats » (lien : diapo 56)
- Gestion de projet
- La triple contrainte
- Le paradigme du changement
- Agile = Itération + Incrément
- Prédicatif - Itératif - Incrémental - Agile

## What is Project Management ?

- The process of bringing a project to fruition in as effective a manner as possible
- A set of tools and techniques to describe, organize and monitor the work of project activities
- The practice of bringing visibility to the project
- Project managers are responsible for carrying out these activities

## Qu'est-ce que la gestion de projet ?

- Processus de réalisation d'un projet de la manière la plus efficace possible
- Un ensemble d'outils et de techniques pour décrire, organiser et surveiller le travail des activités du projet
- La pratique consistant à faire connaître le projet
- Les gestionnaires de projet sont responsables de la réalisation de ces activités

## « Cowboys herding cats » video

<https://www.youtube.com/watch?v=Pk7yqlTMvp8>



## Project Management

- Project managers have a wide variety of skills :
  - General management
  - Leadership
  - Coaching
  - Technical
  
- Every project must work within « triple constraints »
  - Scope
  - Time
  - Cost

© Global Knowledge Training LLC. All rights reserved.

Page 59

## Gestion de projet

Les gestionnaires de projet possèdent une grande variété de compétences :

- Management
- Leadership
- Coaching
- Technique
  
- Chaque projet doit fonctionner avec « La triple contrainte » :
  - Pérимètre
  - Temps
  - Coût



Formateur : Project = Scope, Time, Cost + (Quality, Risk, Benefit, Ressources, Communication)

### Stagiaire : La triple contrainte

Temps (Planification) : Calendrier de création des produits livrables

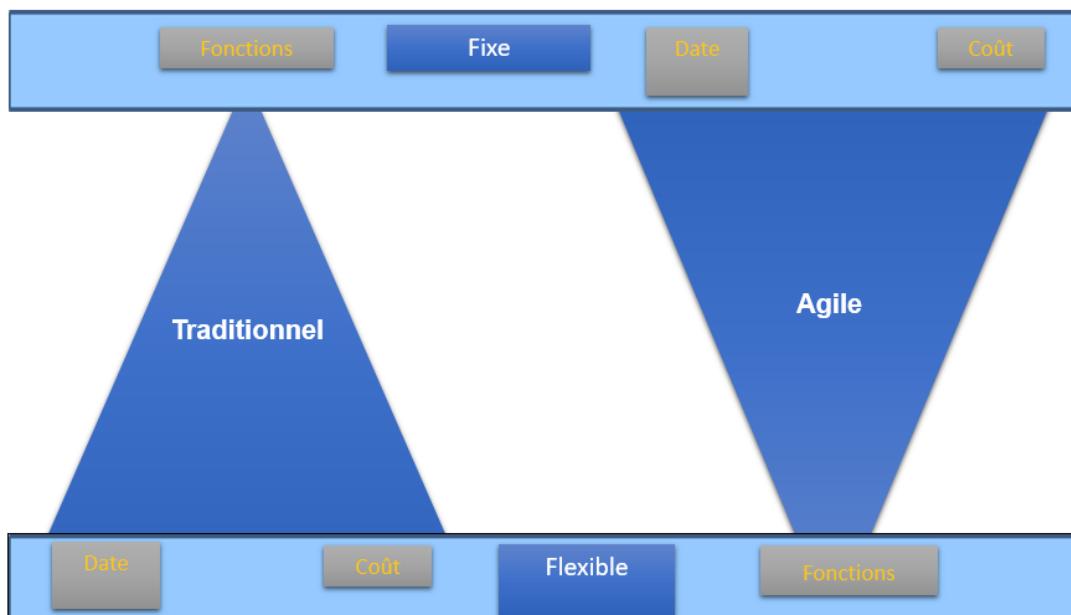
Coût : Personnel, matériel et équipement utilisés pour créer les produits livrables

Périmètre: Caractéristiques et fonctionnalités livrables pour le projet



Formateur : Agile = open Scope, eventually fixed Cost and/or fixed Time  
 Classical = fixed Scope, eventually sliding Time and/or Cost

### Stagiaire : Le paradigme du changement



Agile = Iterative + Incremental

Not like this....

Like this!

© Global Knowledge Training LLC. All rights reserved.

Page 62

Formateur : Explain the 2 lines (first down, second up) and that, once stabilized, Agile products become Classical products

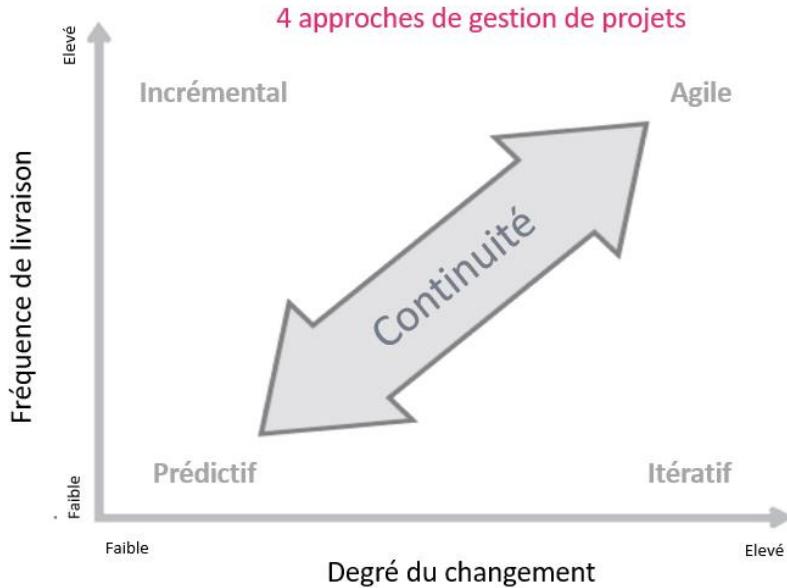
Stagiaire : Agile = Itératif + Incrémental

Agile = Iterative + Incremental



Formateur : Degree of Change and Frequency of Delivery most often linked to Innovation

Stagiaire : **Agile = Itératif + Incrémental**



## Predictive - Iterative - Incremental - Agile

| Characteristics    |              |                                       | 4 Project's Management approaches |   |
|--------------------|--------------|---------------------------------------|-----------------------------------|---|
| Approach           | Requirements | Activities                            | Delivery                          | Goal  |
| <b>Predictive</b>  | Fixed        | Performed once for the entire project | Single delivery                   | Manage cost   |
| <b>Iterative</b>   | Dynamic      | Repeated until correct                | Single delivery                   | Correctness of solution                             |
| <b>Incremental</b> | Dynamic      | Performed once for a given increment  | Frequent smaller deliveries       | Speed   |
| <b>Agile</b>       | Dynamic      | Repeated until correct                | Frequent small deliveries         | Customer value via frequent deliveries and feedback |

## Prédicatif – Itératif – Incrémental – Agile

| Caractéristiques |           |                                   |                            |   |
|------------------|-----------|-----------------------------------|----------------------------|---|
| Approche         | Exigence  | Activités                         | Livraison                  | But   |
| Prédicatif       | Fixe      | Fait une fois pour tout le projet | Une seule livraison        | Gestion des coûts                                       |
| Itératif         | Dynamic   | Répétition jusqu'à correction     | Une seule livraison        | Exactitude de la livraison                              |
| Incrémental      | Dynamic   | Fait une fois pour l'incrément    | Petite livraison fréquente | Rapide  |
| Agile            | Dynamique | Répétition jusqu'à correction     | Petite livraison fréquente | Valeur client via des livraisons rapides et corrections |

## Criteria for choosing an approach (Classic / Agile)

- Change = Project
- Classical – Agile, History
- Select the good temperature in a room
- It Is A Complex Situation
- Empirical processes help control complexity
- Simple, Complicated, Complex, Chaotic
- The Complexity of Software Development
- Relating Complexity to Management Style

## Critères de choix d'une approche (Classique/Agile)

Changement = Projet

Classique - Agile, Histoire

Sélectionnez la bonne température dans une pièce

C'est une situation complexe

Les processus empiriques aident à contrôler la complexité

Simple, compliqué, complexe, chaotique

La complexité du développement de logiciels

Relier la complexité au style de gestion

## Criteria for choosing an approach (Classic / Agile)

- Process Control Models
- Situation Dictates Defined or Empirical
- Scrum Team Multi-Tasking Consequences
- Basic Truths About People Motivation/Composition
- What Does Scrum Call for?
- Sprinting at a Sustainable Pace

## Critères de choix d'une approche (Classique/Agile)

Modèles de contrôle de processus

La situation dicte le choix de l'approche projet : définie vs. empirique

Conséquences du travail multi-tâches sur la productivité de l'équipe Scrum

Vérités fondamentales sur la motivation des personnes/ la composition des équipes

Qu'est ce qui est demandé par Scrum?

« Sprinter » à un rythme soutenable

## Change = Project (1)

- Classical VS Agile
- priority 1, priority 2, priority 3
  - Classical ➔ + Efficient (Scope, Time, Cost)
  - Agile ➔ + Effective (Scope, Time, Cost)
- Innovation
  - Effect « Time to market »
  - sometimes, Effect « First takes all »
- 2 necessary conditions for success :
  - Direction (strategy/plan - Right Thing)
  - Speed (realization -Thing Right)

Formateur : Classical can appear more efficient than Agile but can be more effective than Agile. Agile -> effect « Time to market » and sometimes, effect « First takes all » (GAFAM)

Best Direction +best speed = success

Stagiaire : **Changement = Projet (1)**

- Classique vs Agile
- Priorité 1, Priorité 2, Priorité 3
  - Classique ➔ Plus efficient (Périmètre, Temps, Coût)
  - Agile ➔ Plus efficace (Périmètre, Temps, Coût)
- Innovation
  - Effet sur le délai de mise sur le marché
  - Parfois, effet « C'est le premier qui récupère tout le marché»
- Deux conditions nécessaire au succès :
  - Direction (Stratégie/Plan – penser à faire ce qui est bien)
  - Rapidité (Réalisation – bonne approche pour faire des choses)

## Change = Project (2)

- Difference in Perceived Value
  - Waterfall : How much was consumed?
  - Agile : How much remains to be done?
  
- If unnecessary work or done nothing ➔ Perception of:
  - Progression ➔ Don't Fail
  - Regression ➔ If Fail, Fail Fast

Formateur: Classical: error is a fault; Agile: error is an opportunity to learn (in order not to repeat it, and create value through sharing)

### Stagiaire : **Changement = Projet (2)**

Différence dans la valeur perçue :

- Gestion de projet en cascade : Combien de ressources avons-nous consommé ?
- Agile : Combien de travail nous reste-t-il pour réussir ?

Si travail inutile ou « Ne rien faire » ➔ Perception de :

- Progression ➔ Ne pas échouer
- Régression ➔ Si échec, échouons vite

## Classical / Agile → different visions

|                                 | Cartesian rationalism   | Pragmatic empiricism  |
|---------------------------------|---|---|
| <b>Paradigm</b>                 | Predictability  | Adaptability  |
| <b>Methods</b>                  | Conventional or "complete"  | New or « Agile »  |
| <b>Cycle project</b>            | Cascade (without feedback)  | Incremental and iterative (adaptive)  |
| <b>Waiver form risk</b>         | Descriptive and documentary   | Action - research - experimentation   |
| <b>Reasoning</b>                | Discursive (premises, conclusions)                                    | Systemic and heuristic  |
| <b>Underlying vision</b>        | Isolate to structure a world frozen part                              | Run to understand the dynamics of interactions  |
| <b>Thought</b>                  | Reductionism and mechanistic hypotheses                               | Holistic view of events (HR, communication, environment, .)                               |
| <b>Analytical philosophy</b>    | Considers the nature of interactions                                  | Considers the effects of interactions   |
| <b>Structuring method</b>       | On the basis of fixed insulation levels of abstractions and concerns. | Based on a simple and flexible phasing taking into account the constraints of the project |
| <b>Research Axis</b>            | Analysis of the structure   | The outcome of actions  |
| <b>Limits and possibilities</b> | Reduction of simple and complex systems analysis                      | Understanding of complex systems by their purposes  |
| <b>To systems</b>               | High entropy  | High retroactivity, "cyber"   |
| <b>Outcome</b>                  | Search completeness of the solution                                   | Accept a "satisfactory performance"   |
| <b>Philosophy of Action</b>     | Leads to a fully detailed action planned                              | Leads to a flexible action and driven by goals  |
| <b>Validation</b>               | Test each element of the test cases or copy of reality                | Permanent confrontation of the model with reality   |

© Global Knowledge International Inc. 2019

Page 69

## Classique / Agile → Des visions différentes

|                                 | Rationalisme Cartésien   | Pragmatique empirique   |
|---------------------------------|--|---|
| <b>Paradigme</b>                | Prédictibilité   | Adaptabilité  |
| <b>Méthodes</b>                 | Conventionnel ou "complète"  | Nouvelle ou « Agile »   |
| <b>Cycle de projet</b>          | Cascade (sans feedback)  | Incrémental and itérative (adaptive)  |
| <b>Gestion des risques</b>      | Descriptive et documentée  | Action - recherche - expérimentation  |
| <b>Motivation</b>               | Discursive (locaux, conclusions)   | Systémique and heuristique  |
| <b>Vision sous-jacente</b>      | Isoler pour structurer les éléments gelés  | Agir pour comprendre la dynamique des interactions                            |
| <b>Pensée</b>                   | Réductionnisme et hypothèses mécanique   | Vue holistique d'événements (RH, communication, environnement, .)             |
| <b>Philosophie d'analyse</b>    | Considération de la nature des interactions                                      | Considération de l'effet des interactions                                     |
| <b>Procédé de structuration</b> | Sur la base de niveaux d'isolation fixes des abstractions et des préoccupations. | Basé sur une phase simple et flexible tenant compte des contraintes du projet |
| <b>Axe de recherche</b>         | Analyse de la structure  | Le résultat des actions   |
| <b>Limites et possibilités</b>  | Réduction de l'analyse des systèmes simples et complexes                         | Compréhension des systèmes complexes par leurs objectifs                      |
| <b>Systèmes</b>                 | Entropie élevée  | Haute rétroactivité, "cyber"  |
| <b>Résultats</b>                | Perfection de recherche de la solution   | Accepter une « performance satisfaisante »                                    |
| <b>Philosophie d'actions</b>    | Mène à une action prévue entièrement détaillée                                   | Dirige une action flexible et motivée par des objectifs                       |
| <b>Validation</b>               | Tester chaque élément des études de cas ou de la réalité                         | Confrontation permanente du modèle avec la réalité                            |

## Classical / Agile → History

| Project Management | Classical               |       | Agile                                     |       |
|--------------------|-------------------------|-------|---|-------|
| Year               | Prehistory              |       | 2001                                      |       |
| Approach           | Waterfall or Predictive | scope | Iterative/Incremental or Change driven    | scope |
|                    | PMP (PMI)               | all   | Scrum                                     | IT    |
|                    | Prince2 (Axelos)        | IT    | Extreme Programming (XP)                  | IT    |
|                    |                         |       | Acceptance Test Driven Development (ATDD) | IT    |
|                    |                         |       | Agile Modeling                            | IT    |
|                    |                         |       | Agile Unified Process (AUP)               | IT    |
|                    |                         |       | Continuous integration (CI)               | IT    |
|                    |                         |       | Crystal clear                             | IT    |
|                    |                         |       | Crystal Methods                           | IT    |
|                    |                         |       | Dynamic Systems Development Method (DSDM) | IT    |
|                    |                         |       | Feature Driven Development (FDD)          | IT    |
|                    |                         |       | Graphical System Design (GSD)             | IT    |
|                    |                         |       | Kanban                                    | IT    |
|                    |                         |       | Lean software development                 | IT    |
|                    |                         |       | Scrumban                                  | IT    |
|                    |                         |       | Story-driven modeling                     | IT    |
|                    |                         |       | Test-driven development (TDD)             | IT    |
|                    |                         |       | Velocity tracking                         | IT    |
|                    |                         |       | Software Development Rhythms              | IT    |

© Global Knowledge Training LLC. All rights reserved.

Page 70

## Classique / Agile → Histoire

| Gestion de projet | Classique             |           | Agile   |           |
|-------------------|-----------------------|-----------|---|-----------|
| Année             | Préhistorique         |           | 2001  |           |
| Approche          | Cascade ou Prédictive | Périmètre | Itérative/Incrémental ou conduite du changement | Périmètre |
|                   | PMP (PMI)             | Tout      | Scrum   | IT        |
|                   | Prince2 (Axelos)      | IT        | Extreme Programming (XP)                        | IT        |
|                   |                       |           | Acceptance Test Driven Development (ATDD)       | IT        |
|                   |                       |           | Modèle Agile                                    | IT        |
|                   |                       |           | Agile Unified Process (AUP)                     | IT        |
|                   |                       |           | Continuous integration (CI)                     | IT        |
|                   |                       |           | Crystal clear                                   | IT        |
|                   |                       |           | Crystal Methods                                 | IT        |
|                   |                       |           | Dynamic Systems Development Method (DSDM)       | IT        |
|                   |                       |           | Feature Driven Development (FDD)                | IT        |
|                   |                       |           | Graphical System Design (GSD)                   | IT        |
|                   |                       |           | Kanban  | IT        |
|                   |                       |           | Lean software development                       | IT        |
|                   |                       |           | Scrumban  | IT        |
|                   |                       |           | Story-driven modeling                           | IT        |
|                   |                       |           | Test-driven development (TDD)                   | IT        |
|                   |                       |           | Velocity tracking                               | IT        |
|                   |                       |           | Software Development Rhythms                    | IT        |

## Select the good temperature in a room 5 min.

Purpose: Explore how variables lead to complexity



Question: What variables will you take into account?

© Global Knowledge Training LLC. All rights reserved.

Page 71

### Sélectionnez la bonne température dans une pièce

Objectif : Explorer comment les variables mènent à la complexité

Question : Quelles variables devez-vous prendre en compte ?

## It Is A Complex Situation

- Number of people in room
- Metabolism of each person
- Activity of each person
- Opening/closing of doors
- Weather: including sun, clouds, and outside temperature
- Temperature of adjoining rooms
- Construction material of the building
- Floor of the room
- Will food be served, when, what type, and how much?
- Temperature of food brought into room

## C'est une situation complexe

- Nombre de personnes dans la pièce
- Métabolisme de chaque personne
- Activité de chaque personne
- Ouverture / Fermeture des portes
- Météo : Incluant le soleil, les nuages, et la température extérieure
- Température des pièces voisines
- Matériau de construction du bâtiment
- Sol dans la salle
- Quelle nourriture va être servi, quand, quel type et quelle quantité ?
- Température des aliments apportés dans la pièce

## Empirical processes help control complexity

- Variables can be ignored. Actual temperature drives setting of heating, sun blinds, and air conditioning.



- An empirical process requires:
- To inspect the room temperature frequently
- To adapt the systems that drive the temperature (heating and air cooling)
- Transparency to make sure the real temperature is inspected with a common understanding

### Formateur : Réglage de la température d'une pièce

Les participants doivent lister les paramètres qui rentrent en compte dans le réglage de la température d'une salle de cours, appréhender la complexité d'un algorithme qui intègrerait ces paramètres pour mettre à la bonne température la salle de cours, et évaluer la fiabilité et le coût d'une telle solution. Par rapport à cette solution complexe, couteuse et fragile, le formateur leur demande de proposer une autre approche et présente alors l'approche du thermostat réglable par l'utilisateur, qui est simple, fiable et économique, et fait ressortir le parallèle de cette approche avec une approche Classique ou Agile dans la gestion de projet.

### Stagiaire : Les process empiriques aident à contrôler la complexité

- Les variables peuvent être ignorées. La température réelle détermine le réglage du chauffage, des stores solaires et de la climatisation.

### Un processus empirique exige :

- d'**inspecter** fréquemment la température ambiante
- d'**adapter** les systèmes qui alimentent la température (chauffage et refroidissement d'air)

- la *transparence* pour s'assurer que la température réelle soit inspectée avec une compréhension commune

## The complexity of Software Development

⌚ 3 min.

- List the variable and parameters that have to be considered in software development.
- How predictable are they?
- What would you do to control them?

### La complexité du développement logiciel

- Énumérer les variables et les paramètres qui doivent être pris en compte dans le développement de logiciels.
- Dans quelle mesure sont-ils prévisibles?
- Que feriez-vous pour les contrôler?

## The Predictability of Software Development

⌚ 1 min

➤ On a scale of 1-10, put a slider at the level of unpredictability of 3 major variables in software development:

- Requirements
- Technology
- People

©1993-2013 Scrum.Org

13

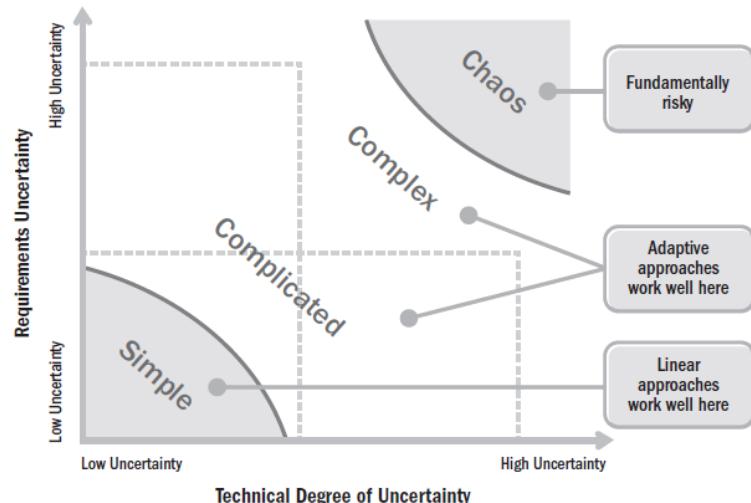
Page 75

## La prévisibilité de développement logiciel

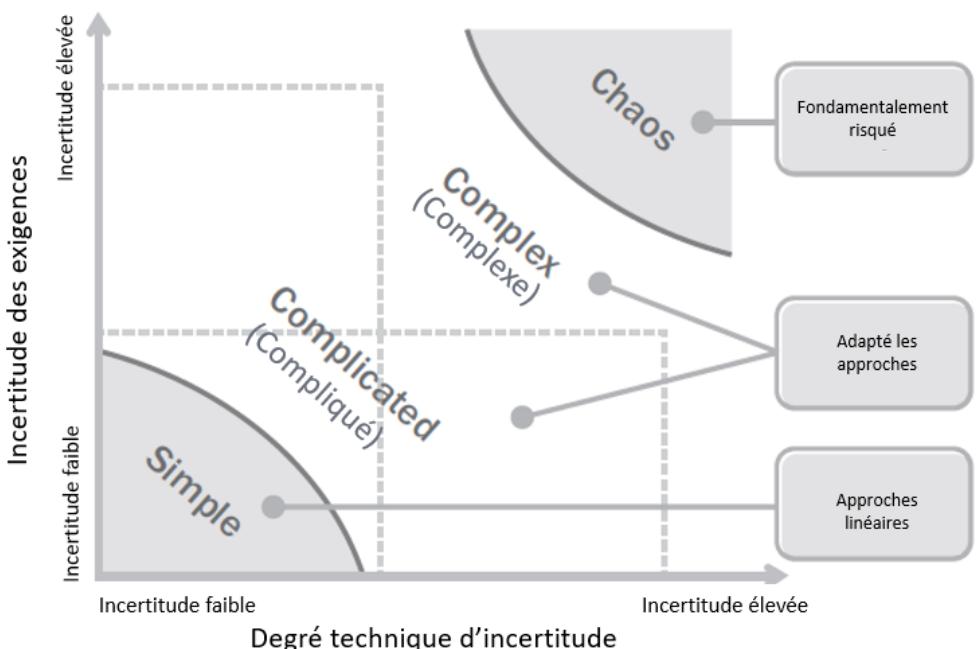
Sur une échelle de 1 à 10, mettre un curseur au niveau de l'imprévisibilité de 3 variables majeures dans le développement de logiciels:

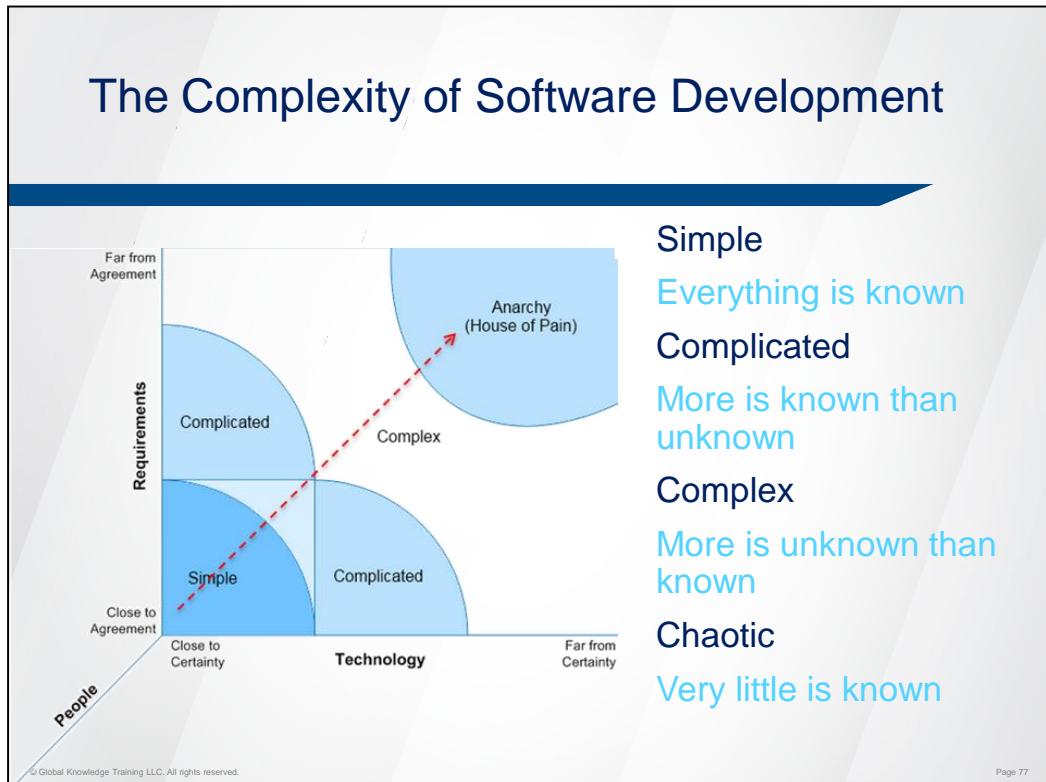
- Exigences
- Technologie
- Personnes

## Simple, Complicated, Complex, Chaotic



## Simple, compliqué, complexe, chaotique





Simple (Classical), Complicated (Agile), Complex (Agile), Chaotic (nothing, don't move until you know at least the direction)

### Stagiaire : La complexité du développement logiciel

Simple : Tout est connu

Compliqué : Plus de connu que d'inconnu

Complex : Plus d'inconnu que de connu

Chaotique : Beaucoup d'inconnu



## Relating Complexity to Management Style (1)

| Project type | Characteristics  | Leader's job  |
|--------------|--|---|
| Chaotic      | High turbulence<br>No clear cause-and-effect<br>Unknowns<br>Many decisions and no time                               | Immediate action to re-establish order<br>Prioritize and select actionable work<br>Look for what works rather than perfection<br>Act, sense, respond        |
| Complex      | More unpredictability than predictability<br>Emergent answers<br>Many competing ideas                                | Create bounded environments for action<br>Increase levels of interaction and communication<br>Servant leadership<br>Generate ideas<br>Probe, sense, respond |
| Complicated  | More predictability than unpredictability<br>Fact-based management<br>Expertd work out wrinkles                      | Utilise experts to gain insights<br>.....metrics to gain control<br>.....e, analyse, respond<br>Command and control   |
| Simple       | Repeating patterns and consistent events<br>Clear cause-and-effect<br>Well establish knowns<br>Fact based management | Use best practices<br>Extensive communication not necessary<br>Establish patterns and optimize to them<br>Command and control                               |

©1993-2013 Scrum.Org

Source: « Leadership Framework for Decision Making », Snowden and Boone, HBR, 2008

Page 78

## Rapport de complexité au style de management (1)

| Type de projet | Caractéristiques   | Rôle du Leader  |
|----------------|--|---|
| Chaoitique     | Turbulence élevée<br>Pas de cause à effet clair<br>Inconnus<br>Beaucoup de décision et pas de temps                                  | Mesures immédiates pour rétablir l'ordre, hiérarchiser et sélectionner les travaux donnant lieu à des mesures correctives.<br>Cherchez ce qui fonctionne plutôt que la perfection<br>Agir, sentir, réagir |
| Complexe       | Plus d'imprévisibilité que de prévisibilité<br>Réponses émergentes<br>De nombreuses idées concurrentes                               | Créer des environnements bornés pour l'action<br>Accroître les niveaux d'interaction et de communication<br>Leadership<br>Générer des idées<br>Sonde, sens, réponse                                       |
| Compliqué      | Plus de prévisibilité que d'imprévisibilité<br>Gestion factuelle<br>Expertd élimine les <b>rides</b>                                 | Utiliser des experts pour obtenir des renseignements<br>.....mesures pour prendre le contrôle<br>..... analyse, réponse<br>Commandement et contrôle   |
| Simple         | Modèles répétés et événements constitutifs<br>Cause à effet clairs<br>Bien établir les connaissances<br>Gestion fondée sur les faits | Utiliser les pratiques exemplaires<br>Une communication étendue n'est pas nécessaire<br>Établir des modèles et les optimiser<br>Commandement et contrôle  |

## Relating Complexity to Management Style (2)

| Project type | Characteristics  | Leader's job  |
|--------------|--|---|
| Chaotic      | High turbulence<br>No clear cause-and-effect<br>Unknowns<br>Many decisions and no time                               | Immediate action to re-establish order<br>Prioritize and select actionable work<br>Look for what works rather than perfection<br>Act, sense, respond        |
| Complex      | More unpredictability than predictability<br>Emergent answers<br>Many competing ideas                                | Create bounded environments for action<br>Increase levels of interaction and communication<br>Servant leadership<br>Generate ideas<br>Probe, sense, respond |
| Complicated  | More predictability than unpredictability<br>Fact-based management<br>Expert handles wrinkles                        | Utiliser des experts pour gain insights<br>.....mesures pour gain control<br>Sense, analyse, respond<br>Command and control                                 |
| Simple       | Repeating patterns and consistent events<br>Clear cause-and-effect<br>Well establish knowns<br>Fact based management | Use best practices<br>Extensive communication not necessary<br>Establish patterns and optimize to them<br>Command and control                               |

©1993-2013 Scrum.Org

Source: « Leadership Framework for Decision Making ». Snowden and Boone, HBR, 2008

© Global Knowledge Training LLC. All rights reserved.

Page 79

Formateur : Project Manager has to adapt its Management Style to the right Project approach to use

### Stagiaire : Rapport de complexité au style de management (2)

| Type de projet | Caractéristiques   | Rôle du Leader  |
|----------------|--|---|
| Chaotique      | Turbulence élevée<br>Pas de cause à effet clair<br>Inconnus<br>Beaucoup de décision et pas de temps                                  | Mesures immédiates pour rétablir l'ordre, hiérarchiser et sélectionner les travaux donnant lieu à des mesures correctives.<br>Cherchez ce qui fonctionne plutôt que la perfection<br>Agir, sentir, réagir |
| Complexe       | Plus d'imprévisibilité que de prévisibilité<br>Réponses émergentes<br>De nombreuses idées concurrentes                               | Créer des environnements bornés pour l'action<br>Accroître les niveaux d'interaction et de communication<br>Leadership<br>Générer des idées<br>Sonde, sens, réponse                                       |
| Compliqué      | Plus de prévisibilité que d'imprévisibilité<br>Gestion factuelle<br>Expert élimine les <i>wrinkles</i>                               | Utiliser des experts pour obtenir des renseignements<br>.....mesures pour prendre le contrôle<br>..... analyse, réponse<br>Commandement et contrôle   |
| Simple         | Modèles répétés et événements constitutifs<br>Cause à effet clairs<br>Bien établir les connaissances<br>Gestion fondée sur les faits | Utiliser les pratiques exemplaires<br>Une communication étendue n'est pas nécessaire<br>Établir des modèles et les optimiser<br>Commandement et contrôle  |

## Process Control Models

- Defined
  - Predict the future
  - Initial information and assumptions are valid throughout the complete planning horizon.
  - Examples: Construction, Payrol/Accounting software, stabilized Websites/Mobile Apps (# 80% of projects, will last forever)
- Empirical
  - Adapt to the future
  - Frequent inspection to adapt to new circumstances
  - Examples: GAFAM, NATU, IoT, AI, Blockchain, Autonomous driving car, etc. (# 20% of projects, growing share)

### Modèles de commande de processus

- Défini
  - Prévoit le futur
  - L'information initiale et les hypothèses sont valides tout au long de la planification.
  - Exemples : la construction, les logiciels de paie, les sites internet stabilisés /applications mobiles (#80% des projets, perdureront pour toujours)
- Empirique
  - S'adapter au futur
  - Inspection fréquente pour s'adapter aux nouvelles circonstances
  - Exemples : GAFAM, NATU, IoT, AI, Blockchain, voitures autonomes, etc. (# 20% des projets, part croissante)

## Situation Dictates Defined or Empirical

- « When the process is too complicated for the defined approach, the empirical approach is the appropriate choice »



© Global Knowledge Training LLC. All rights reserved.

Page 81

### Les situations décident « Défini » ou « Empirique »

« Quand le process est trop compliqué avec une approche définie, l'approche empirique est le choix le plus approprié »

## Scrum Team Multi-Tasking Consequences

- When people task switch amongst work:
  - Appearance: A greater amount of work underway
  - Reality: Less work accomplished
  - Every piece of work is many times more expensive



© Global Knowledge Training LLC. All rights reserved.

Page 82

## Les conséquences d'une équipe Scrum multi-tâches

Lorsque les tâches des personnes changent pendant le projet :

- Apparence : Plus de travail en cours
- Réalité : Moins de travail accompli
- Chaque élément de travail coûte beaucoup plus cher

### Equipe 1

Toutes les personnes de l'équipe travaillent à temps plein sur le projet A

Vélocité = 17.

### Equipe 2

Toutes les personnes de l'équipe travaillent à 50% du temps sur le projet A et 50% du temps sur le projet B

Vélocité = 12

### Equipe 3

Toutes les personnes de l'équipe travaillent à 25% du temps sur les projets A,B,C et D

Vélocité = 4

## The Cindy Problem

⌚ 3 min.

Purpose: How to deal with scarce skills

- You are scrum Master for three Scrum Teams. They work from the same Product Backlog, have the same Product Owner, and share a common code base.
- The Development Teams report that in the next three Sprints they will all be working in one area of the database. Cindy is the only DBA that knows that subschema well. The Teams will need Cindy full-time for their Sprints.

Question: What do you suggest?

© Global Knowledge Training LLC. All rights reserved.

Page 83

Formateur : Prioritize for the best value or the less worst loss

Stagiaire : **Le problème de Cindy**

**Objectif :** Comment gérer les compétences rares

Vous êtes Scrum Master pour trois équipes Scrum. Elles travaillent à partir du même Product Backlog, elles ont le même Product Owner et partagent une base de code commune.

Les équipes de développement signalent que, dans les trois prochains Sprints, elles travailleront toutes dans le même domaine de la base de données.

Cindy est le seul DBA qui connaît bien ce subschéma. Les équipes auront besoin de Cindy à temps plein pour leurs Sprints.

**Question :** Que suggérez-vous?

## Basic Truths About Team Performance

- Teams and people do their best work when they aren't interrupted;
- Teams improve most when they solve their own problems;
- Broad-band, face-to-face communications is the most productive way for teams to work together.

### Vérités de base sur la performance des équipes

Les équipes et les individus font de leur mieux lorsqu'ils ne sont pas interrompus;

Les équipes améliorent plus lorsqu'elles résolvent leurs propres problèmes;

La communication en face à face est la façon la plus productive pour les équipes de travailler ensemble.

## Basic Truths About People Motivation (1)

- External rewards like money (carrot-and-stick) work only for simple, mechanical work (Maslow)
  - It has opposite effects in cognitive, complex or creative work
- The secret to high performance lies in going beyond money (Herzberg)
  - Autonomy – Organizing my own work
  - Mastery – Becoming better at my work
  - Purpose – Making a contribution

©1993-2013 Scrum.Org

102

Page 85

## Vérités de base sur la motivation des équipes (1)

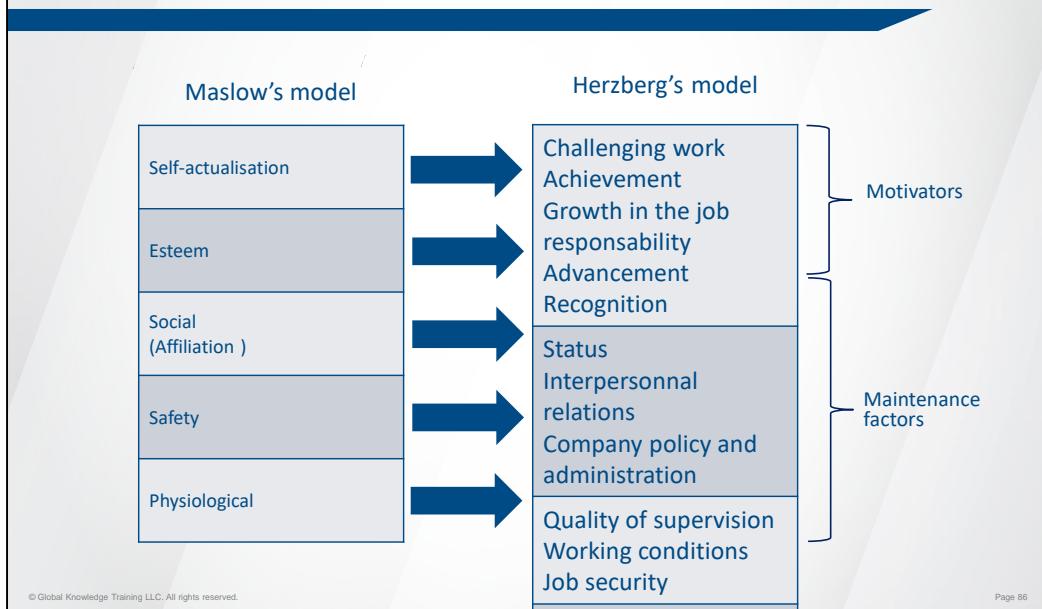
Les récompenses externes comme l'argent ne fonctionnent que pour un travail mécanique simple (Maslow)

- Il existe des effets opposés dans le travail cognitif, complexe ou créatif

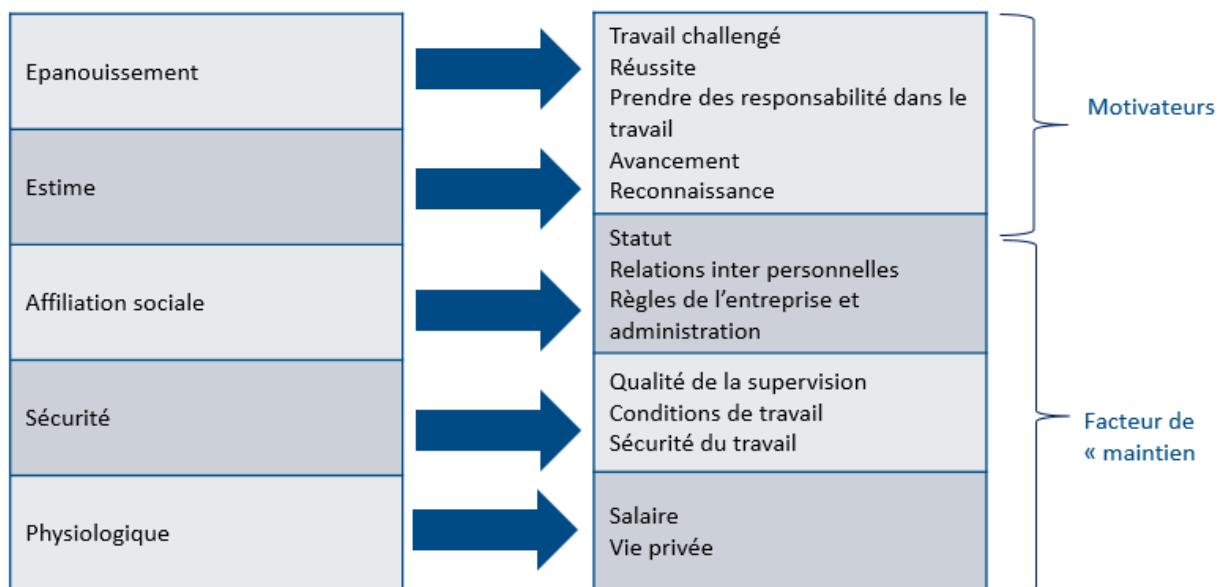
Le secret de la haute performance réside dans le dépassement de l'argent (Herzberg)

- Autonomie – Organisation de mon propre travail
- Maîtrise – S'améliorer dans mon travail
- Objectif – Contribuer

## Basic Truths About People Motivation (2)



## Vérités de base sur la motivation des équipes (2)



## Basic Truths About Team Motivation (3)

- People are most productive when they manage themselves
- People take their commitment more seriously than other people's commitment for them
- People have many creative moments during down time
- People always do the best they can (Jerry Weinberg)
- Under pressure to « work harder », developers automatically and increasingly reduce quality

## Vérités de base sur la motivation des équipes (3)

Les personnes sont plus productives lorsqu'elles se gèrent seules

Les personnes prennent leurs propres engagements plus au sérieux.

Les personnes ont de nombreux moments créatifs pendant le « temps mort »

Les personnes font toujours de leur mieux (Jerry Weinberg)

Sous la pression de « travailler plus », les développeurs réduisent automatiquement et de plus en plus la qualité

## Dispersed, Distributed Teams

⌚ 1 min.

- You are on a Scrum Team.
  - You and two other members live in Boston, MA.
  - One in St. Petersburg, Russia
  - Two in Beijing, China
  - One in San Jose, California
  
- How might you conduct the daily Scrum?

Formateur : General rule VS Exception rule. Smaller (1 or 2 ) could be acceptable with Skype, but here, too much...

### Stagiaire : Equipe dispersée

Vous êtes dans une équipe Scrum :

- Vous et deux autres membres de l'équipe vivez à Boston, MA.
- Un à St. Pétersbourg, Russie
- Deux à Pékin, Chine
- Un à San Jose, Californie

Comment pouvez-vous gérer le Daily Scrum ?

## Basic Truths About Team Composition

- Teams are more productive than the same number of individuals (1+1=3; synergy team/group)
- The optimum size team is around seven people, and no more than nine;
- Products are more robust when a team has all of the cross-functional skills focused on the work;
- Changes in team composition often lower productivity for a time.

## Vérités de base sur la composition des équipes

Les équipes sont plus productives que le même nombre de personnes (1+1=3; équipe/groupe synergique)

L'équipe de taille optimale est d'environ sept personnes, et pas plus de neuf;

Les produits sont plus robustes lorsqu'une équipe possède toutes les compétences interfonctionnelles axées sur le travail;

Les changements dans la composition de l'équipe réduisent souvent la productivité pendant un certain temps

## What Does Scrum Call for?

- Scrum does not require team co-location
- Scrum recommends, but does not mandate, team size (3 to 9)
- Inspect and adapt

« A co-located, self organizing Scrum Team is 100% more efficient than otherwise ».

(Source: Boston Consulting Group, 2005)

Optimize the productivity of the development organization by measuring the costs and benefits of different approaches to development

### A quoi fait appelle Scrum ?

- Scrum ne nécessite pas d'équipe co-localisée
- Scrum recommande, mais n'oblige pas, la taille de l'équipe (3 à 9)
- Inspection et adaptation

« Une équipe Scrum auto-organisée et au même endroit est 100 % plus efficace que d'autres équipes ayant un fonctionnement différent » (

Optimiser la productivité de l'organisation de développement en mesurant les coûts et les avantages des différentes approches de développement.

## Team Conflicts (1)

 3min.

Purpose: Focus on keeping teams performing

- You are the Scrum Master entering the team room.
  - The analyst runs past you crying and the lead engineer runs you enraged, both on the way to their functional managers' offices.
  - You can cut the tension with a knife it is so thick.
  - The analyst has been writing specs and giving them to the engineers, who then change them as they see fit. Anger over this has been building for three weeks.

Question: What do you suggest?

Formateur : Conflict resolution techniques -> next slide (but Scrum Master intervenes as coach not solver)

Stagiaire : **Gestion des conflits d'équipe (1)**

**Objectif : Focus sur le maintien de la performance des équipes**

Vous êtes le Scrum Master et vous entrez dans la pièce :

- L'analyste passe devant vous en pleurant et l'ingénieur en chef vous rend furieux, tous les deux en direction des bureaux de leurs managers fonctionnels.
- Vous pouvez stoppez la tension avec un « couteau » tellement elle est forte
- L'analyste a écrit des spécifications et les a données aux ingénieurs, qui les ont changées ensuite comme bon leur semble. La colère à propos de cela a été alimentée pendant trois semaines

**Question : Quelle est votre proposition ?**

## Team Conflicts (2)

- Problem Solving: Involves the use of logical thinking to understand the problem (root cause).
- Compromise: This technique involves reaching an agreement by each of the conflicting parties giving something. Therefore, both parties don't get exactly what they want.
- Forcing: Forcing, a win-lose situation, involves using your authority or power to decide on the outcome of a conflict regardless of the opinion of the participants.
- Smoothing: This conflict resolution technique involves playing down the issue. The key focus is simply on the positive aspects.
- Withdrawal: Withdrawal simply means to avoid the problem altogether.

Formateur : Problem Solving = best one

### Stagiaire : Gestion des conflits d'équipe (2)

- Résolution de problèmes : Implique l'utilisation de la pensée logique pour comprendre le problème (cause racine).
- Compromis : Cette technique consiste à parvenir à un accord par chacune des parties en conflit. Par conséquent, les deux parties n'obtiennent pas exactement ce qu'elles veulent.
- Forçage : Forcer, une situation où tout le monde est gagnant-perdant, implique d'utiliser votre autorité ou votre pouvoir pour décider de l'issue d'un conflit, peu importe l'opinion des participants.
- Lissage : Cette technique de résolution des conflits consiste à minimiser le problème. L'accent est mis sur les aspects positifs.
- Retrait : Retrait simplement pour éviter le problème.

## Sprinting at a Sustainable Pace

- People work at a sustainable pace of 8 hours per day
- On a self-organizing Development Team that commits to delivering working software every Sprint, team members are working 12 hours per day...
- If the Development Team members have to work more than 8 hours per day, the quality and creativity drop

High Moon Studios went from 8 to 12 hours days. There was a corresponding increase in defects of 60%. The cost of defects more than offset the additional productivity of 4 hours per day.

### Sprint a un rythme soutenable

- Les gens travaillent à un rythme soutenable de 8 heures par jour
- Au sein d'une équipe de développement auto-organisée qui s'engage à livrer des logiciels opérationnels à chaque sprint, les membres de l'équipe travaillent 12 heures par jour...
- Si les membres de l'équipe de développement doivent travailler plus de 8 heures par jour, la qualité et la créativité diminuent

High Moon Studios est passé de 8 à 12 heures/jour. Il y a eu une augmentation correspondante des défauts de 60%. Le coût des erreurs compense la productivité supplémentaire de 4 heures par jour.

## Product Owner Availability

⌚ 3min.

- The Product Owner and Development Team are trying to work out their working relationship.
- How much time should the Product owner spend with the Development Team?
  - Full time?
  - Just for the Sprint Planning and Review?
  - Is he/she needed at the Retrospective?
  - Should he/she attend the Daily Scrum?

Formateur : As needed; Yes; Yes; No, or silently

### Stagiaire : Disponibilité du Product Owner

Le Product Owner et l'équipe de développement essaient de mettre au point leur relation de travail.

Combien de temps le Product Owner doit-il passer avec l'équipe de développement?

- A temps plein
- Uniquement les Sprint planning et les Reviews ?
- Est-il/elle nécessaire à la Rétrospective ?
- Doit-il/elle suivre le Daily Scrum ?

## Agile Myths and Legends (1)

1. Organization are not succeeding with Agile
2. Agile works better than traditional approaches (e.g Waterfall)
3. Traditional (Waterfall) works better than distributed/offshore teams
4. Agile teams waste a lot of time testing that traditional teams don't
5. Agile teams don't produce documentation
6. Daily stand-ups are just glorified status meetings
7. Without detailed records, I don't know that my people are really working all the time!
8. If we convert to Agile, that means we can « do more with less » right?

### Mythes sur le choix d'Agile comme approche, sur l'application d'Agile et sur ses risques spécifiques (1)

1. Les organisations ne réussissent pas avec Agile
2. Agile fonctionne mieux que les approches traditionnelles (ex. cascade)
3. Les approches traditionnelles (cascade) fonctionnent mieux que les équipes à distance/offshore
4. Les équipes Agile perdent beaucoup moins de temps à tester que les équipes traditionnelles
5. Les équipes Agile ne produisent pas de documentation
6. Les Daily stand-ups ne sont que des réunions de statut glorifiées
7. Sans enregistrements détaillés, je ne sais pas si mes employés travaillent vraiment tout le temps!
8. Si nous nous convertissons en Agile, cela signifie que nous pouvons « faire plus avec moins », n'est-ce pas?

## Agile Myths and Legends (2)

- Common myths still abound, although agile methods have been around for ~15 years :
  - Agile is only for software engineering
  - Agile doesn't scale to large systems
  - Agile doesn't use project management
  - Agile doesn't have any requirements
  - Agile requires a traditional system architecture
  - Agile doesn't have any documentation
  - Agile isn't disciplined or measurable
  - Agile has low quality, maintainability and security

### **Mythes sur le choix d'Agile comme approche, sur l'application d'Agile et sur ses risques spécifiques (2)**

Les mythes courants sur les méthodes agiles persistent depuis 15 ans :

- Agile est seulement pour l'ingénierie logicielle
- Agile ne s'applique pas aux grands systèmes
- Agile n'utilise pas la gestion de projet
- Agile n'a aucune exigence
- Agile requiert une architecture système traditionnelle
- Agile n'a aucune documentation
- Agile n'est pas carré ou mesurable
- Agile a une faible qualité, maintenabilité et sécurité

## Agile Myths and Legends (3)

|  |   |  |
|--|---|--|
| <b><i>“Agile development is not disciplined”</i></b> | <b><i>“Agile development is just another fad / hype”</i></b>        | <b><i>“Agile means teams cannot be controlled by management”</i></b>   |
| <b><i>“Agile teams do not plan”</i></b>              | <b><i>“Agile means you never have to write documentation”</i></b>   | <b><i>“Agile works only for small projects; it does not scale”</i></b> |
| <b><i>“Agile development is not predictable”</i></b> | <b><i>“Agile means I can change my mind whenever I want to”</i></b> | <b><i>“Agile teams do not work hard, they just play foosball”</i></b>  |

## Mythes sur le choix d’Agile comme approche, sur l’application d’Agile et sur ses risques spécifiques (3)

|   |   |  |
|---|---|--|
| Le développement Agile n'est pas discipliné | Le développement Agile est juste une « nouvelle mode »                    | Agile signifie que les équipes ne peuvent pas être contrôlées par un manager     |
| Les équipes Agile ne planifient pas         | Agile signifie que vous ne rédigerez jamais de documentation              | Agile fonctionne uniquement avec des petits projets. Il n'est pas évolutif       |
| Le développement Agile n'est pas prévisible | Agile signifie que je peux changer mon état d'esprit quand je le souhaite | Les équipes Agile ne travaillent pas dures, elles jouent uniquement au baby-foot |

## Key Success Factors for a Successful Implementation of Agile

- Forming the Scrum Team
- Team Roles
- Team Best Practices
- Communication
- Your Phases of Team Formation
- "Fatburger" Serious Game
- Tips for Working with Scrum Teams
- Keys to a Great Scrum Master
- "Agile Family Management" video

### Facteurs clés de succès d'une implémentation réussie d'Agile

- Former une équipe Scrum
- Rôles de l'équipe
- Bonnes pratiques de l'équipe
- Communication
- Les phases de création de l'équipe
- Le Serious Game « Fatburger »
- Astuces de travail de l'équipe Scrum
- Les clés pour être un bon Scrum Master
- La vidéo « Agile Family Management »

## Forming the Scrum Team

### ➤ Customer Unit

- Customer
- Product Manager
- Marketing
- Executives
- Business Analyst
- More...

### ➤ Developer Unit

- Developer
- QA
- IT/IS
- UI/ Creative
- DBA
- More...

© Global Knowledge Training LLC. All rights reserved.

Page 99

## Former une équipe Scrum

### Equipe client

- Client
- Gestionnaire du Produit
- Marketing
- Cadres
- Analyste business
- Autres...

### Equipe de développeurs :

- Développeurs
- QA
- IT/IS
- UI/Créatif
- DBA
- Autres...

## Team Roles

### ➤ Customer Unit

- Product Vision and Roadmap
- Manage the Product Backlog
- Be prepared with details at the right time
- Set clear expectations for acceptance
- COMMUNICATE

### ➤ Developer Unit

- Estimate
- Plan and commit for the iteration
- Execute the iteration
- Demo
- COMMUNICATE

## Les rôles de l'équipe

### Equipe client :

- Vision produit et feuille de route
- Gérer le Product Backlog
- Etre prêt à donner les détails au bon moment
- Fixer des attentes claires en matière d'acceptation
- COMMUNIQUER

### Equipe de développeurs :

- Estimer
- Planifier et s'engager pour l'itération
- Exécuter l'itération
- Démo
- COMMUNIQUER

## Team Best Practices

- Start as a team, finish as a team
- Empowerment
  - Decision making pushed down to lowest level (subsidiarity)
- Come together
  - Do you have a team space for collaboration
  - If not, what are the alternatives ?
  - Osmotic communication (open space)
- Open and Honest Communication
- Inspect and adapt



© Global Knowledge Training LLC. All rights reserved.

Page 101

### Bonnes pratiques de l'équipe

Commencez comme une équipe, terminez comme une équipe

Autonomisation

- Pousser la prise de décision au niveau le plus bas

Se réunir

- Avez-vous un espace pour la collaboration?
- Si ce n'est pas le cas, quelles sont les solutions alternatives ?
- Communication en osmose (open space)

Communication ouverte et honnête

Inspecter et s'adapter

## Communication /

### ➤ Forums for communication

- Daily Scrum Meeting
- Sprint Planning
- Sprint Review
- Sprint Retrospectives

© Global Knowledge Training LLC. All rights reserved.

Page 102

## Communication

### Moments de communication

- Daily Scrum Meeting
- Sprint Planning
- Sprint Review
- Sprint Retrospectives



### Les phases de formation d'une équipe



## Multiple Projects, one Team

⌚ 2 min.

- The Scrum Team is formed.
  - It has to work on several projects at once to get them done in time.
  - Organizational pressures and conventional wisdom assert the team work on all projects at once.
- What would you advise in this situation? Why?

©1993-2013 Scrum.Org

97

© Global Knowledge Training LLC. All rights reserved.

Page 104

Formateur : Task switching.... is never good

Stagiaire : **Plusieurs projets, une équipe**

Une équipe Scrum est formée :

- Elle doit travailler sur plusieurs projets à la fois pour les réaliser à temps.
- Les pressions organisationnelles et la sagesse conventionnelle affirment que l'équipe travaille sur tous les projets en même temps.

Que conseillerez-vous dans cette situation? Pourquoi?

## « Fatburger » Serious Game



© Global Knowledge Training LLC. All rights reserved.

Page 105

[https://youtu.be/W\\_CL7QOg8T0](https://youtu.be/W_CL7QOg8T0)

## Is it Customer Service?

⌚ 5 Min.

Purpose: Explore the impact of transparency

- You are a student working your way through college. You work at FatBurger on the 2pm to 11pm shift, and are the only person on duty. You are cleaning up at 10:30 when a customer approaches and orders a Double FatBurger Deluxe, with onions, cheese, and bacon and an order of fat fries. You ring up the order.
- The price is \$5.65. The customer informs you that he only has \$1.20
  - FatBurger is high quality. Everything is cooked from scratch.
  - There is no pre-cooked food you were planning on throwing out.
  - FatBurger uses strict inventory control. Anything you take to give to the customer will be charged to your paycheck.
  - The clerk has not yet entered the order.

Question: What do you do? What do you tell the customer?

© Global Knowledge Training LLC. All rights reserved.

Page 106

### Stagiaire : Objectif : Explorer l'impact de la transparence

Vous êtes un étudiant à l'université. Vous travaillez à Fatburger de 14 h à 23 h, et vous êtes la seule personne en service. Vous nettoyez à 22h30 quand un client s'approche et commande un Double Fatburger Deluxe, avec des oignons, du fromage et du bacon ainsi que des frites. Vous saisissez la commande.

Le prix est de 5,65 \$. Le client vous informe qu'il n'a que 1,20 \$.

- Fatburger est de haute qualité. Tout est cuit à la commande.
- Il n'y a pas de nourriture pré-cuite .
- Fatburger utilise un contrôle strict des stocks. Tout ce que vous prenez pour donner au client sera retenu sur votre fiche de paie.
- La commande n'a pas encore été enregistrée.

Question : Que faites-vous ? Que dîtes-vous au client ?

## Tips for Working with Scrum Teams

- Meet individually with each Team Member prior to Scrum project kickoff
  - Discuss ground-rules for Scrum planning sessions
  - Make clear each individual's role as a participant (identify dominant or timid ?)
  - Discuss benefits of participation
  - Discuss your role as Scrum Master and how you can help
- Be prepared for each level of planning, work with Product Owners to help them prepare
- Establish and maintain transparent communication to Management, Stakeholders, other teams, etc...

© Global Knowledge Training LLC. All rights reserved.

Page 107

### **Recommandations pour travailler avec des équipes Scrum**

Rencontrer individuellement chaque membre de l'équipe avant le lancement du projet Scrum

- Discuter des règles de base pour les séances de planification Scrum
- Énoncer clairement le rôle de chaque personne en tant que participant (identifier la personne dominante ou timide?)
- Discuter des avantages de la participation
- Discuter de votre rôle de Scrum Master et de la façon dont vous pouvez aider

Être prêt pour chaque niveau de planification, travailler avec les Product Owners pour les aider à se préparer

Établir et maintenir une communication transparente avec la direction, les parties prenantes, les autres équipes, etc.

## Keys to a Great Scrum Master

- Learn to be an active listener
- Be “emotionally intelligent” (identify, assess, and control the emotions of oneself and others)
- Encourage collaboration
- Always be on the look out for pain points
- Practice negotiation skills
- Be ready for conflict resolution

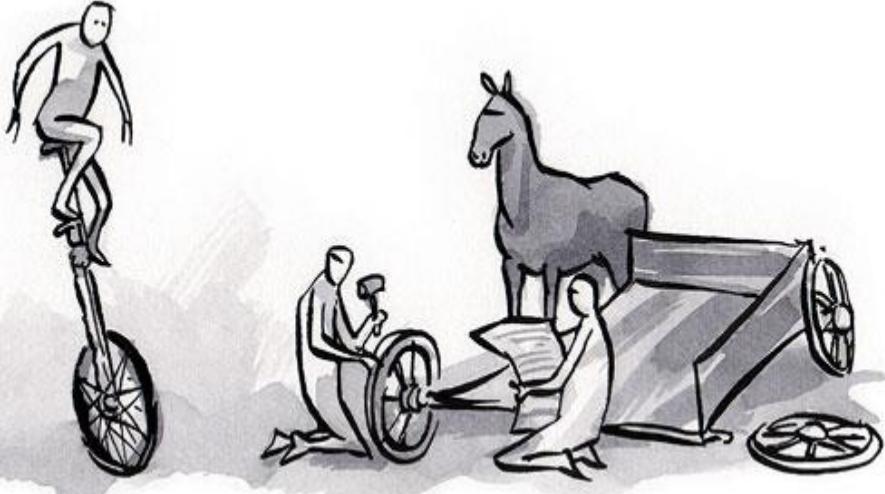
© Global Knowledge Training LLC. All rights reserved.

Page 108

### Les clés pour être un bon Scrum Master

- Apprendre à être en écoute active
- Être « émotionnellement intelligent » (identifier, évaluer et contrôler les émotions de soi-même et des autres)
- Encourager la collaboration
- Etre toujours à l'affût des points douloureux
- Compétences en techniques de négociation
- Être prêt pour la résolution des conflits

## Agile: The best tool for Innovation



© Global Knowledge Training LLC. All rights reserved.

Page 109

Formateur : A mono wheel is lighter and more sophisticated than a sleigh but can not be used for all purposes (individual VS piano)

Stagiaire : Agile : Le meilleur outil pour l'innovation

## « Agile Family Management » video

- Agile Family Management: Bruce Feiler
- [https://www.ted.com/talks/bruce\\_feiler\\_agile\\_programming\\_for\\_your\\_family](https://www.ted.com/talks/bruce_feiler_agile_programming_for_your_family)



© Global Knowledge Training LLC. All rights reserved.

Page 111

