Haute École Léonard de Vinci
SCIENCES ET TECHNIQUES

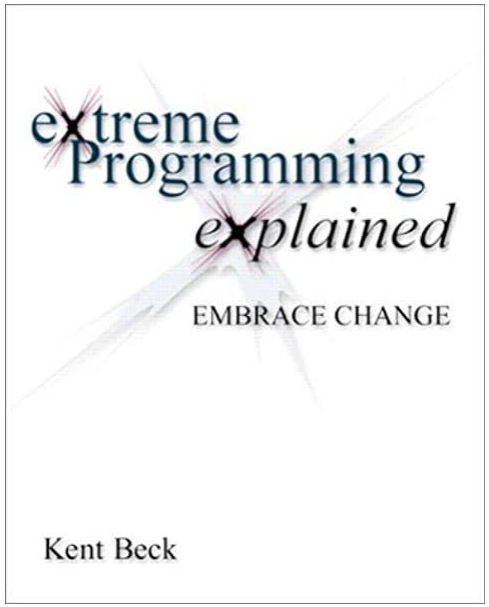
BINV2190-B DevOps : eXtrem Programming

Git les bases

Diego Legua – Olivier Choquet

L'histoire de L'XP

- Né en 1999 avec « Extreme Programming explained »



Source: Amazon.fr

Les valeurs

- La communication
- La simplicité
- Les feedbacks
- Le courage
- Le respect

Les pratiques

- Le client doit être sur site
- Le poker planning
- L'intégration continue (ca va revenir ;))
- Les petites releases
- Un rythme de travail tenable
- Les tests fonctionnels
- Les tests unitaires

Les pratiques

- Une architecture et une conception simple
- Le refactoring
- La métaphore
- L'appropriation collective du projet
- La mise en place de convention de nommage
- Le pair programming

remise dans le contexte

Ce que l'Agile a reprise à l'XP

- Les valeurs communes:
  - La communication:
    - Les individus et leurs interactions plus que les processus et les outils
  - Les feedbacks:
    - La collaboration avec les clients plus que la négociation contractuelle

Ce que l'Agile a reprise à l'XP

- Les valeurs qui sont devenues des principes:
  - La simplicité:
    - La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.
  - Les feedbacks:
    - À intervalles réguliers, l'équipe réfléchit aux moyens de devenir plus efficace, puis règle et modifie son comportement en conséquence.

Ce que l'Agile a reprise à l'XP

Les pratiques

XP	Agile	Scrum
Le client doit être sur site	Les utilisateurs ou leurs représentants et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.	Le rôle de PO
Le poker planning	X	Le poker planning
L'intégration continue	Un logiciel opérationnel est la principale mesure d'avancement.	DOD

Ce que l'Agile a reprise à l'XP

Les pratiques

XP	Agile	Scrum
Les petites releases	Livrez fréquemment un logiciel opérationnel avec des cycles de quelques semaines à quelques mois et une préférence pour les plus courts.	Les sprints
Un rythme de travail tenable	Les processus Agiles encouragent un rythme de développement soutenable.	Le burndown chart
Les tests fonctionnels	X	X
Les tests unitaires	X	X

Ce que l'Agile a reprise à l'XP

Les pratiques

XP	Agile	Scrum
Une architecture et une conception simple	La simplicité – c'est-à-dire l'art de minimiser la quantité de travail inutile – est essentielle.	DOR et DOD
Le refactoring	Accueillez positivement les changements de besoins, même tard dans le projet. Les processus Agiles exploitent le changement pour donner un avantage compétitif au client.	Les sprints planning

Ce que l'Agile a reprise à l'XP

XP	Agile	Scrum
La métaphore	X	X
L'appropriation collective du projet	Les meilleures architectures, spécifications et conceptions émergent d'équipes autoorganisées.	La responsabilité partagée
La mise en place de convention de nommage	X	X
Le pair programming	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.	X

Extreme programming

Le feedback

L'importance du feedback en XP

Les outils

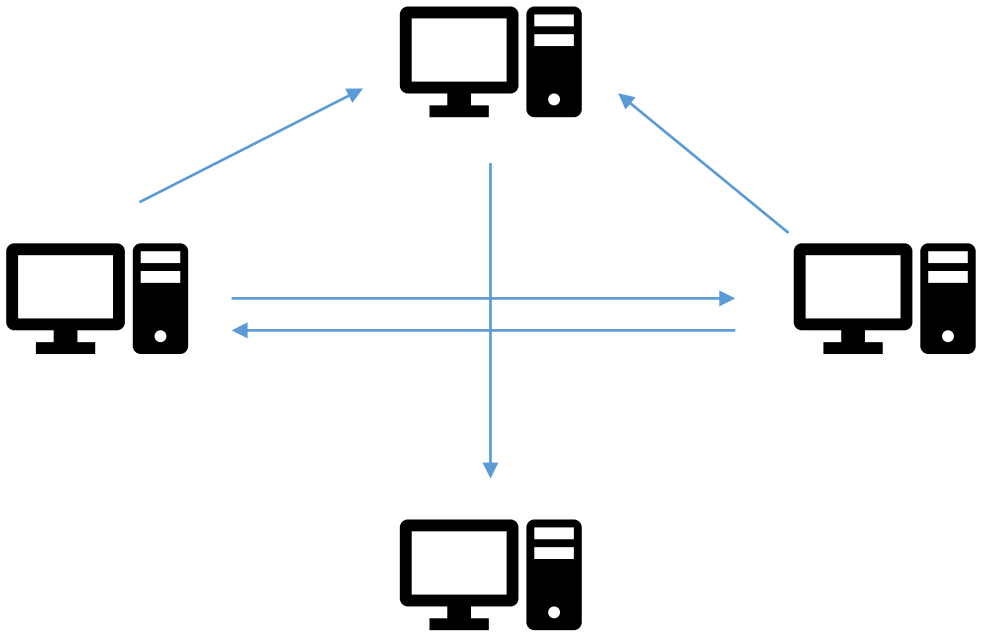
- Les Pull requests
- Le pair programming
- Les rétros
- …

Les pull requests

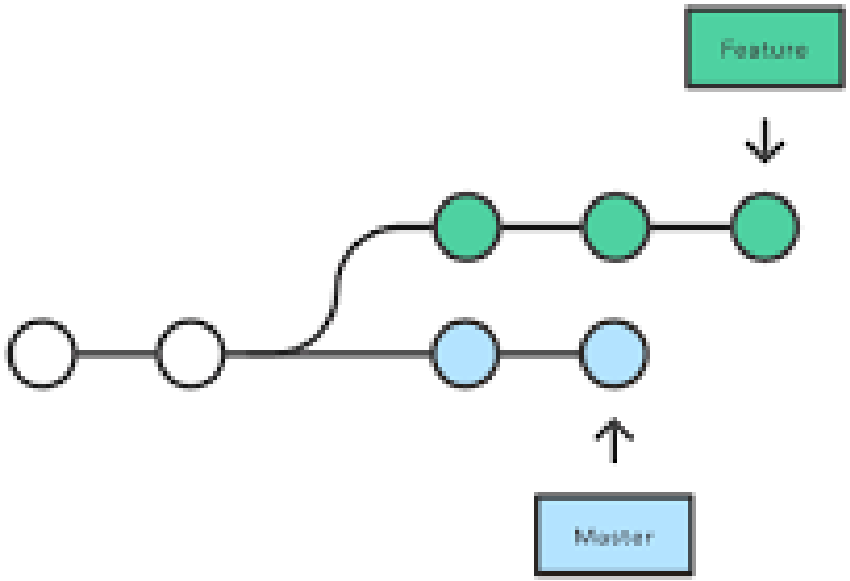
- Git:
  - Protocol de gestion de versions
  - Généralisé dans l'industrie

Git

- Plusieurs versions du codes
- Décentralisé



Les branches



Source: https://oer.gitlab.io

On parlait pas de feedbacks?

La pull request

- Une alternative au merge qui inclus un processus de feedback
- Se crée sur le serveur git « défini comme central »
- Permet à l'équipe de relire le code avant le merge
- Permet de discuter de la meilleur solution pour résoudre un problème

Code review

- Pull request
  - Proposition d'ajout d'une fonctionnalité
  - Proposition de refactoring
  - Proposition de correction de bug

Code review

- Relecture de la pull request
  - Par les autres développeurs
  - Toujours bienveillantes
  - Tout le monde est responsable de la qualité du code et va devoir travailler avec
  - Se fait par ajout de commentaires
  - Une discussion face à face est souvent utile
  - **Le but est d'arriver à la meilleur solution possible**

Le pair programming

- Deux développeurs pour un clavier
- Rôles:
  - Driver: tape le code (ce sont les mains)
  - Navigator: review le code(c'est le cerveau)
- Les statistiques ont démontré que le résultat est un code de meilleur qualité.

Stratégies possibles

- Changer de rôle toutes les 20 minutes
- Un dev écrit un test unitaire, l'autre implémente le test (TDD)

Le mob programming

- Une équipe, un clavier
- Convient pour des phase délicates du développement

Exercice intégrateur - Jeu de rôle

Enoncé :

En tant que développeur junior (**Mr Choquet jouera ce rôle**), vous rejoignez une équipe de développeurs (**tous les étudiants**).

- Celle-ci suit la plupart des principes de l'eXtreme Programming.
- L'équipe désire former ce développeur junior aux pratiques de l'équipe en utilisant le mob programming.
- Elle va demander à ce nouveau développeur de créer un projet basé sur le site des exoplanètes, partager ce projet avec l'équipe et d'ajouter une fonctionnalité toute simple

Exercice intégrateur - Jeu de rôle
Détails des étapes

1. Utilisation du mob programming
   - Mr Choquet est le clavier / les étudiants le cerveau
2. Mise en place du projet
   - Création du projet GitHub
   - Incorporation du site exoplanètes (zip sur MooVin)
   - Partage du projet
3. Ajout d'une simple route permettant l'ajout d'une exolune
   - Quelle stratégie de branche l'équipe emploie ?
   - Quelles sont les conventions de nommage ?
   - Mr Choquet est encore junior -> il faudra sûrement remanier (refactor) le code
4. Feedback sur le code produit par le développeur junior
   - Code Review via pull request