



# Agilité vs Méthodes classiques

## Quelques questions et réflexions

Yves Ledru

Univ. Grenoble Alpes/CNRS, LIG, UMR5217, 38000, Grenoble,  
France

[Yves.Ledru@imag.fr](mailto:Yves.Ledru@imag.fr)



# Yves Ledru



- Professeur à l'Université de Grenoble
  - Enseignant de Génie Logiciel en Master 1 et 2
  - Chercheur:
    - GL, méthodes formelles et test
    - Responsable de l'équipe VASCO du LIG (Test, Modélisation et Sécurité)
    - Ancien Directeur du GDR GPL
- Désolé, je ne suis pas Scrum Master.



asco

# Succès des méthodes agiles

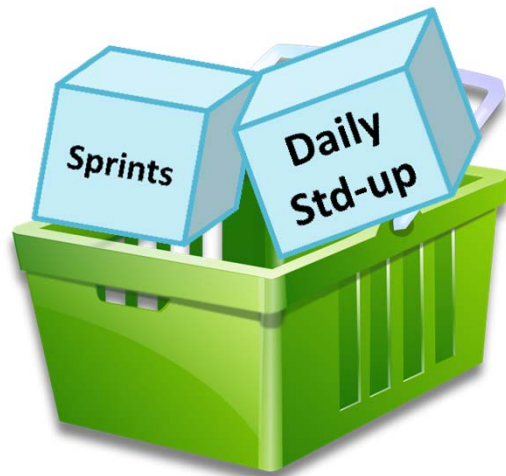
On ne peut pas ignorer le succès avéré des méthodes agiles:

- Démarche largement adoptée par les **développeurs** et leurs **clients**.  
(voir le succès des journées agiles)
- Des statistiques favorables du Standish Group [manifesto 2013]
  - Les **petits projets** (< 1M\$) réussissent mieux que les gros: 76% de réussite
  - L'utilisation des méthodes agiles identifiée comme **facteur de succès** (en phase avec l'implication de l'utilisateur).



# Chacun fait son marché!

- Beaucoup disent avoir adopté l'agilité...
- ... mais n'appliquent que quelques techniques!



- Attention! Il n'est pas interdit d'importer des pratiques agiles dans une approche plus conventionnelle!



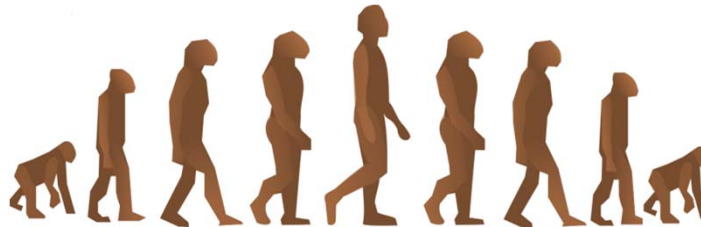
# A boire et à manger...

- Pour:
  - La possibilité de revoir les besoins en cours de projet
  - La disponibilité d'un logiciel exécutable à chaque incrément
  - Le développement piloté par les tests
  - L'utilisation extensive d'outils qui boostent la productivité
- Plus difficile:
  - Quand terminer le projet? Difficilement compatible avec une approche au forfait! (des chefs de projet en SSII n'acceptent cette option que dans un contrat en régie)
  - La gestion d'un projet multi-sites/ ou d'une grande équipe de développement
  - L'architecture d'un grand projet peut-elle vraiment émerger d'une suite de cycles incrémentaux?
  - La documentation sur papier ou dans la tête des membres du projet?
  - Qui est responsable en cas d'échec? Le chef de projet est plus un animateur qu'un chef...



# Une solution de facilité?

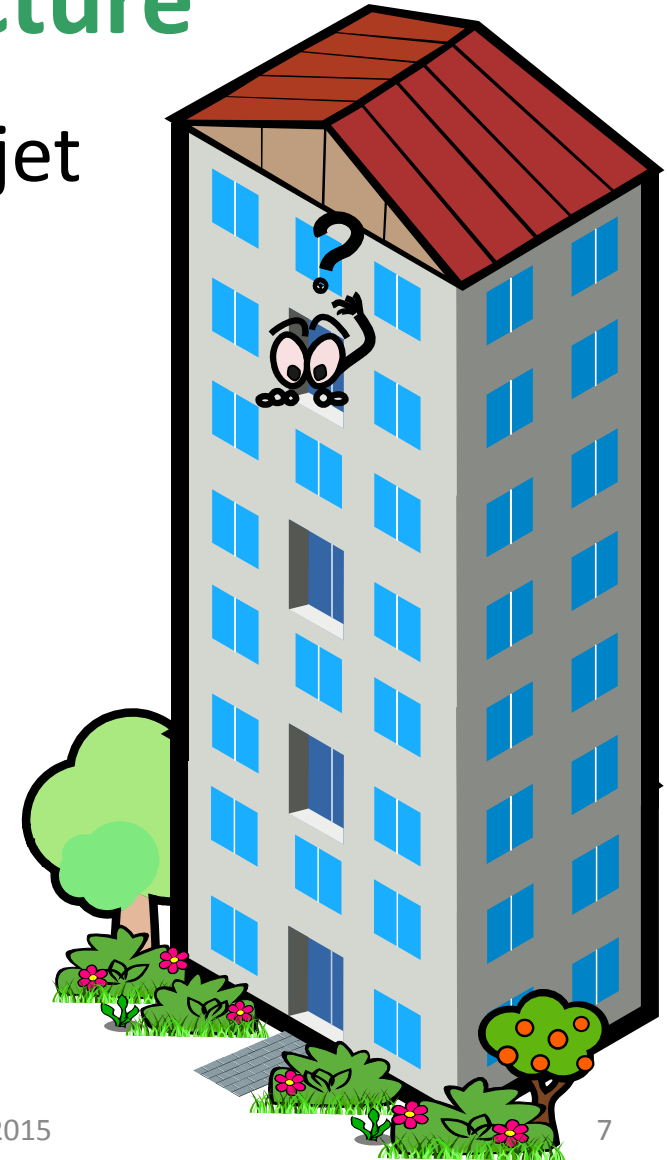
- Au début, la programmation déstructurée
- 30 années de progrès, difficiles à mettre en oeuvre :
  - Structured programming
  - Processus de développement structurés (identification de phases d'analyse des besoins, spécification, architecture, codage, test,...)
- Méthodes agiles : un retour vers la programmation qui encourage la vision à court terme
- E.W. Dijkstra (1930-2002) – Méthodes Agiles (2001-...)





# Vision à court terme : Architecture

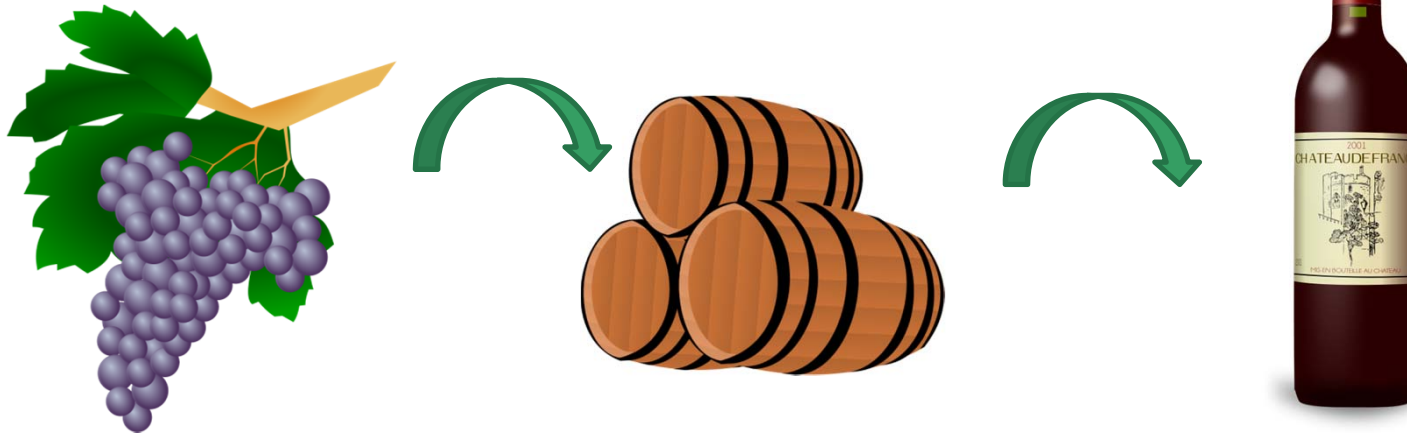
- L'architecture d'un grand projet peut-elle vraiment émerger d'une suite de cycles incrémentaux?
- Un gratte-ciel n'est pas un empilement de maisons...





# Vision à court terme : exécutable dès le premier sprint

- Maturer une solution prend du temps et suppose de passer par des étapes de compréhension du problème et de conception de la solution.
- Pour faire un bon vin, il faut de la patience et de la méthode.

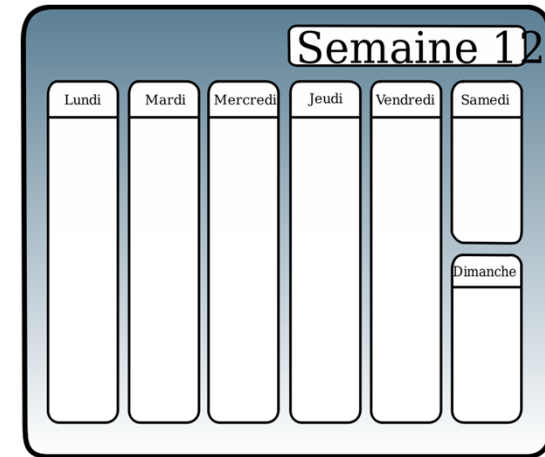






# Vision à court terme : gestion de projet/planification

- Le planning poker et la priorisation du backlog (par le client!) gèrent l'organisation du travail dans les prochains jours/semaines.
- Quel est la carte la plus haute du planning poker?
- L'organisation d'un projet d'envergure demande une planification sur plusieurs mois/sprints.





# Comment concilier une vision à long-terme des processus et des produits avec une approche agile?

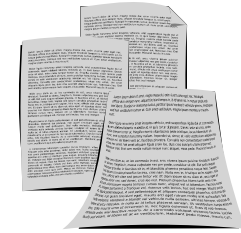


# Questions?

- Clipart from [openclipart.org](http://openclipart.org)



# Vision à court terme : documentation



- Principe agile :
  - *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*
- On peut aussi mal interpréter les principes suivants :
  - *Working software is the primary measure of progress.*
  - *Simplicity--the art of maximizing the amount of work not done--is essential.*
- En pratique,
  - beaucoup de projets bannissent la documentation “up front”,
  - La documentation est réduite à des “user stories” et à de la javadoc.
  - se reposent sur la programmation par paires pour diffuser et mémoriser l’information.
- Problème: comment fait-on pour les projets qui durent plus longtemps que la durée de vie de l’équipe?

