



# Transformer la DSI

**XP-Days, Paris, 24 mars 2006**

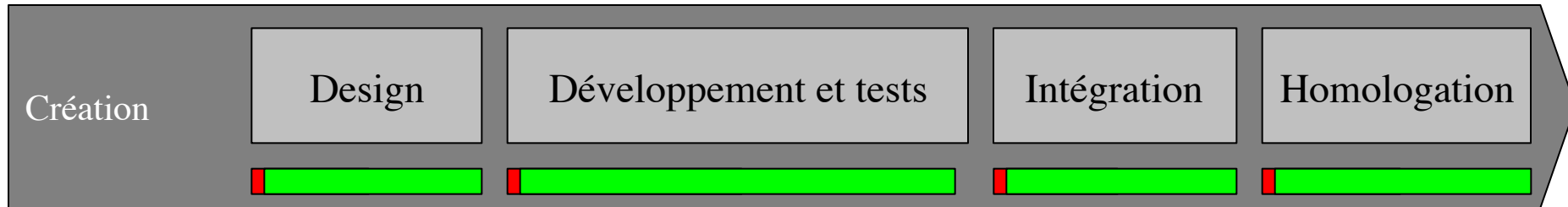
- **Indigestion dans les grands Systèmes d'Information**
  - Croissance des SI, diminution de la productivité
  - Tartes à la crème, mythes et refoulement
    - Processus normé/cascade, outils silver bullet, externalisation ...
- **Changer ?**
  - L'organisation : frontière, CCI, direction des applications
  - Les gens : culture du savoir, culture risque/valeur, culture commerciale, auto-organisation
  - Le système : diminuer le passif
- **Et demain matin ...**



# Indigestion dans le SI

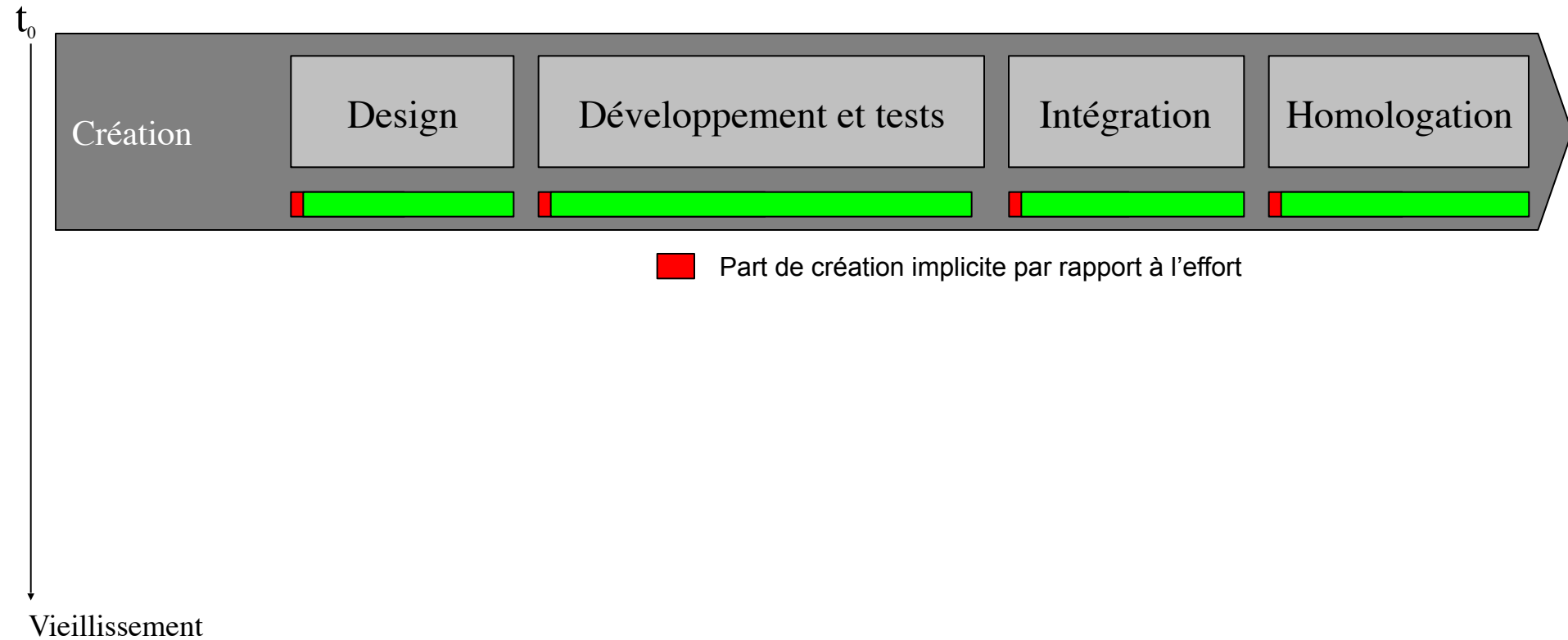


# La productivité se dégrade avec le temps !

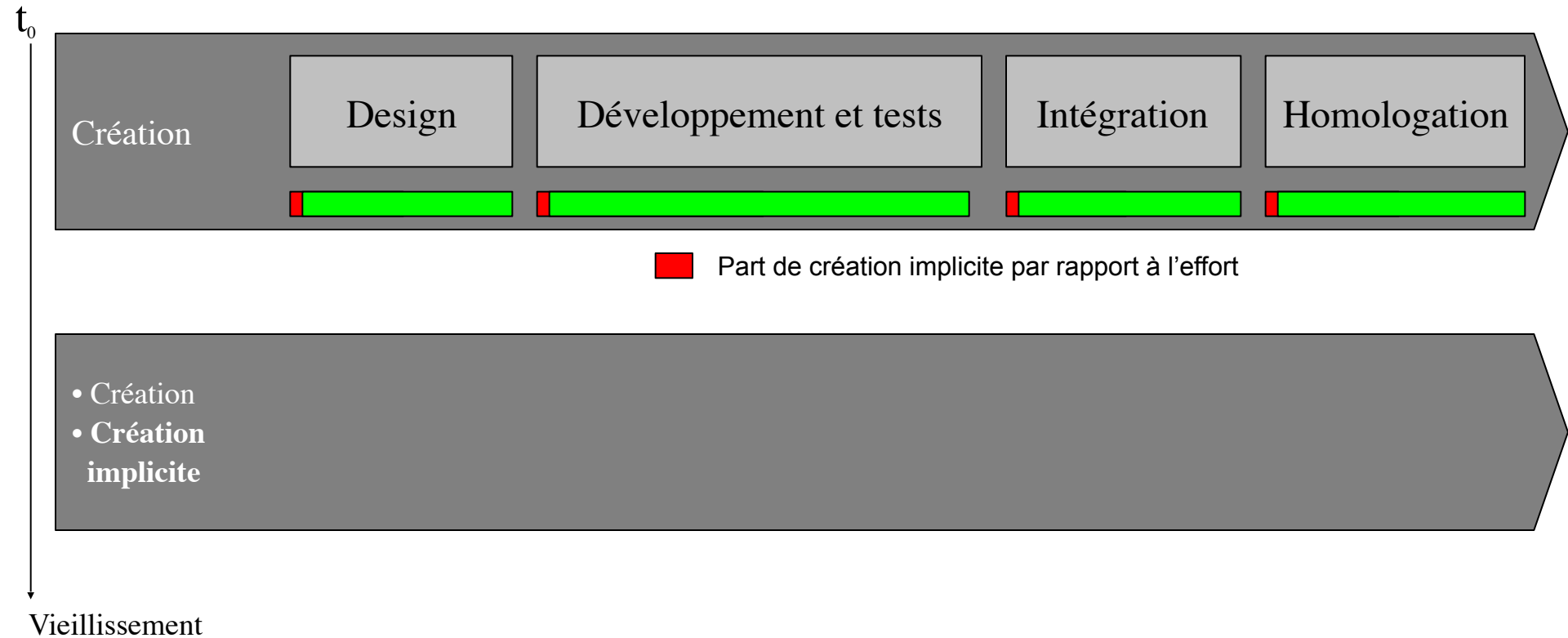


■ Part de création implicite par rapport à l'effort

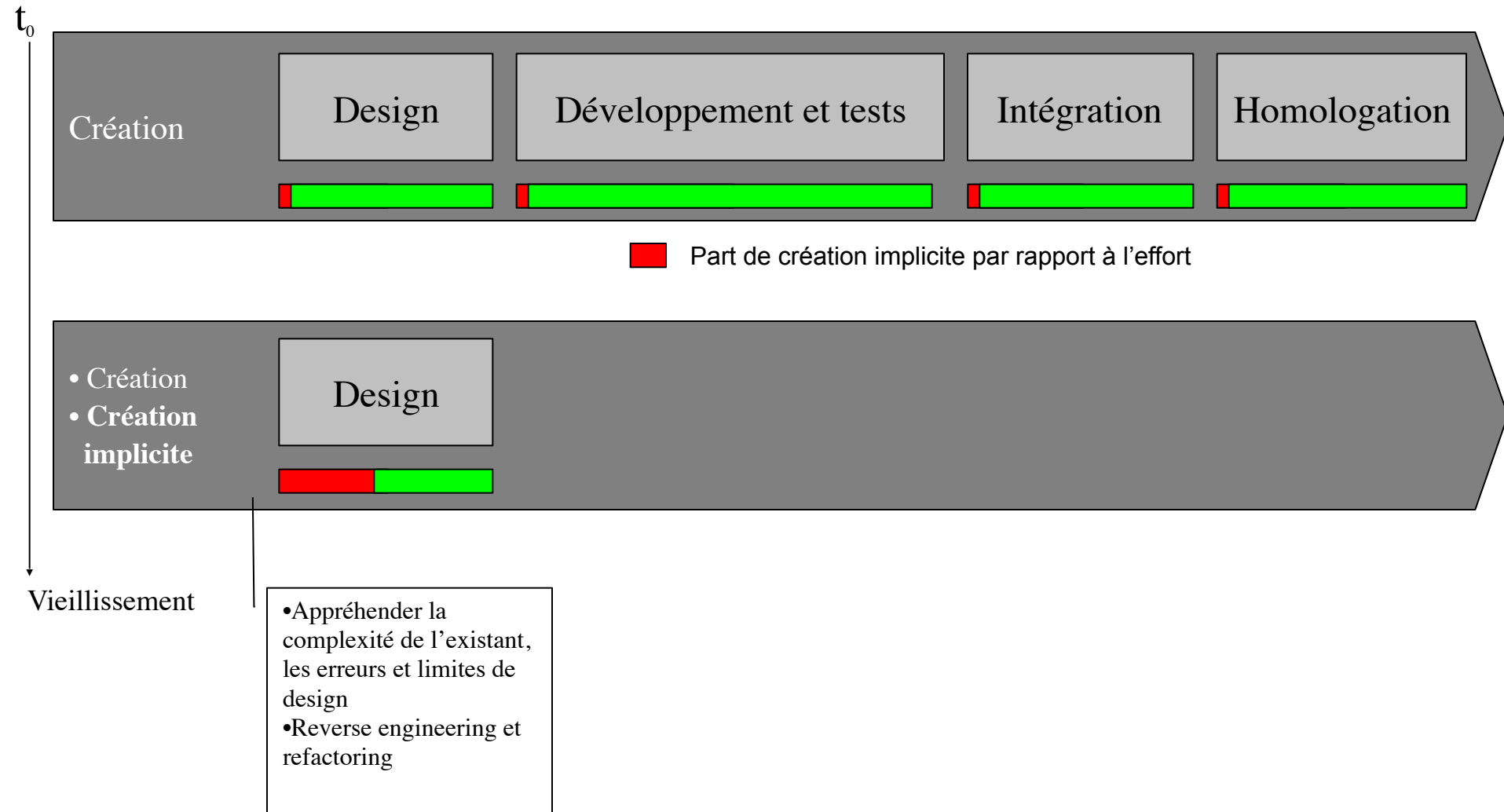
# La productivité se dégrade avec le temps !



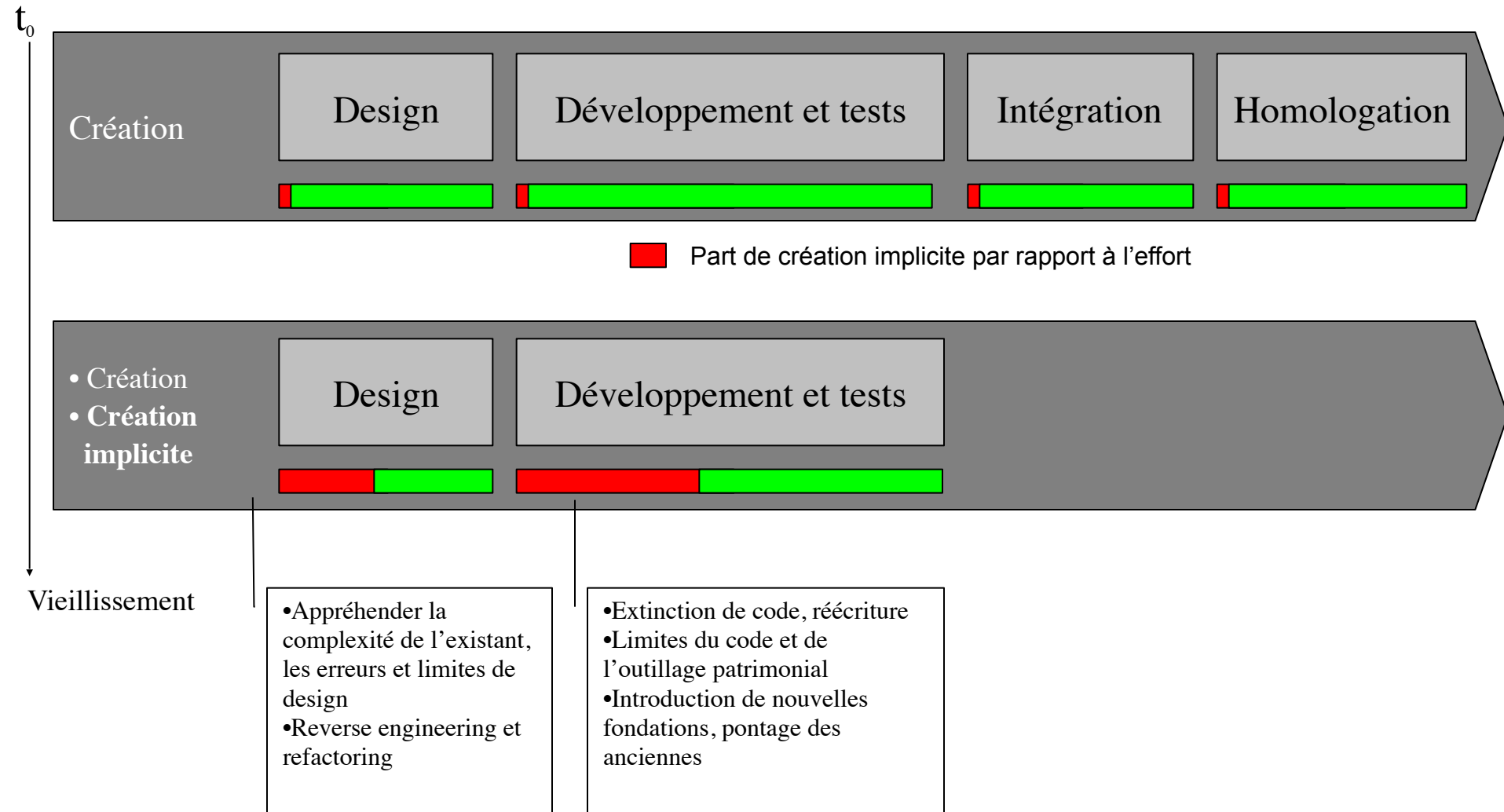
# La productivité se dégrade avec le temps !



# La productivité se dégrade avec le temps !

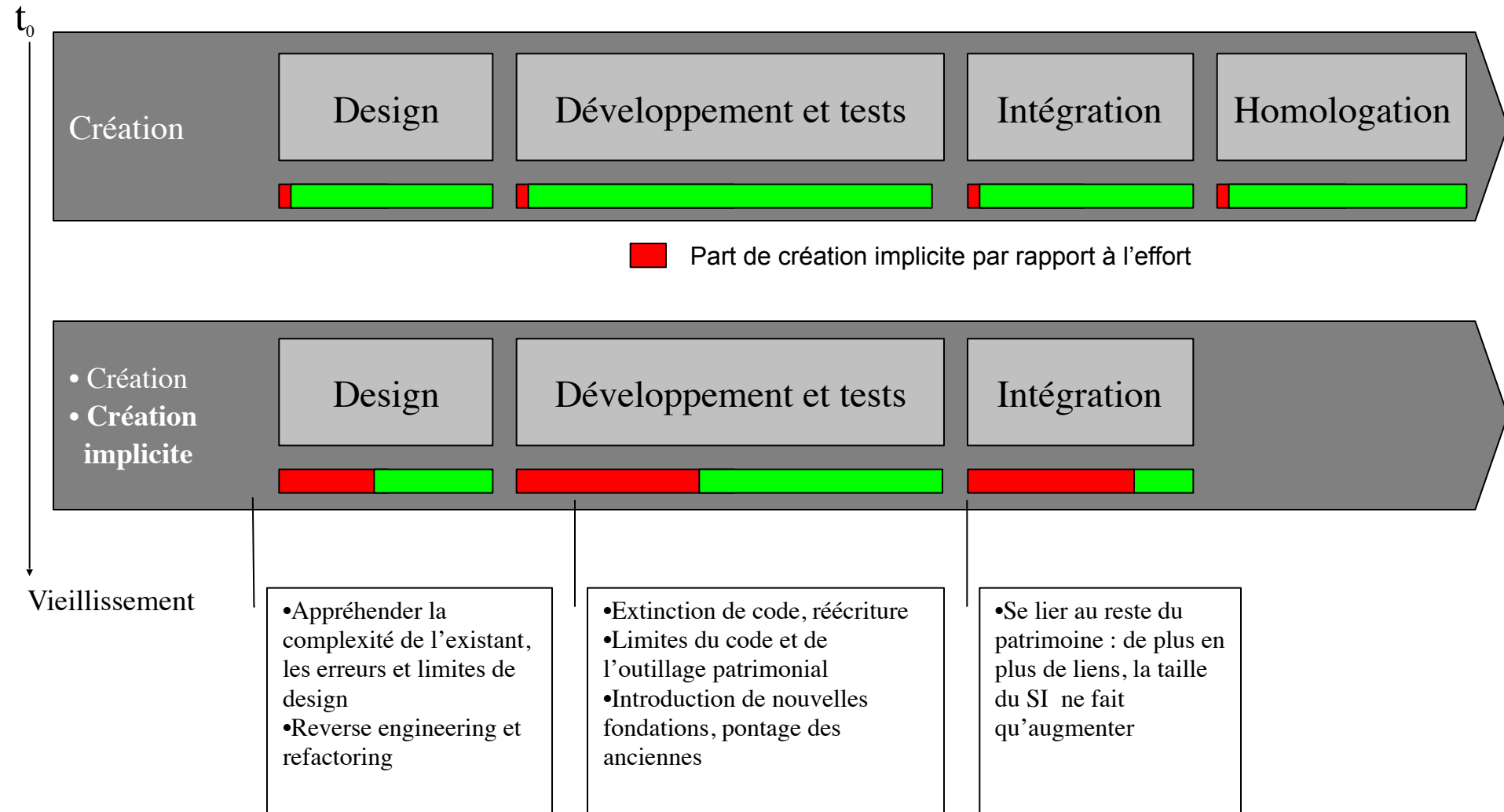


# La productivité se dégrade avec le temps !

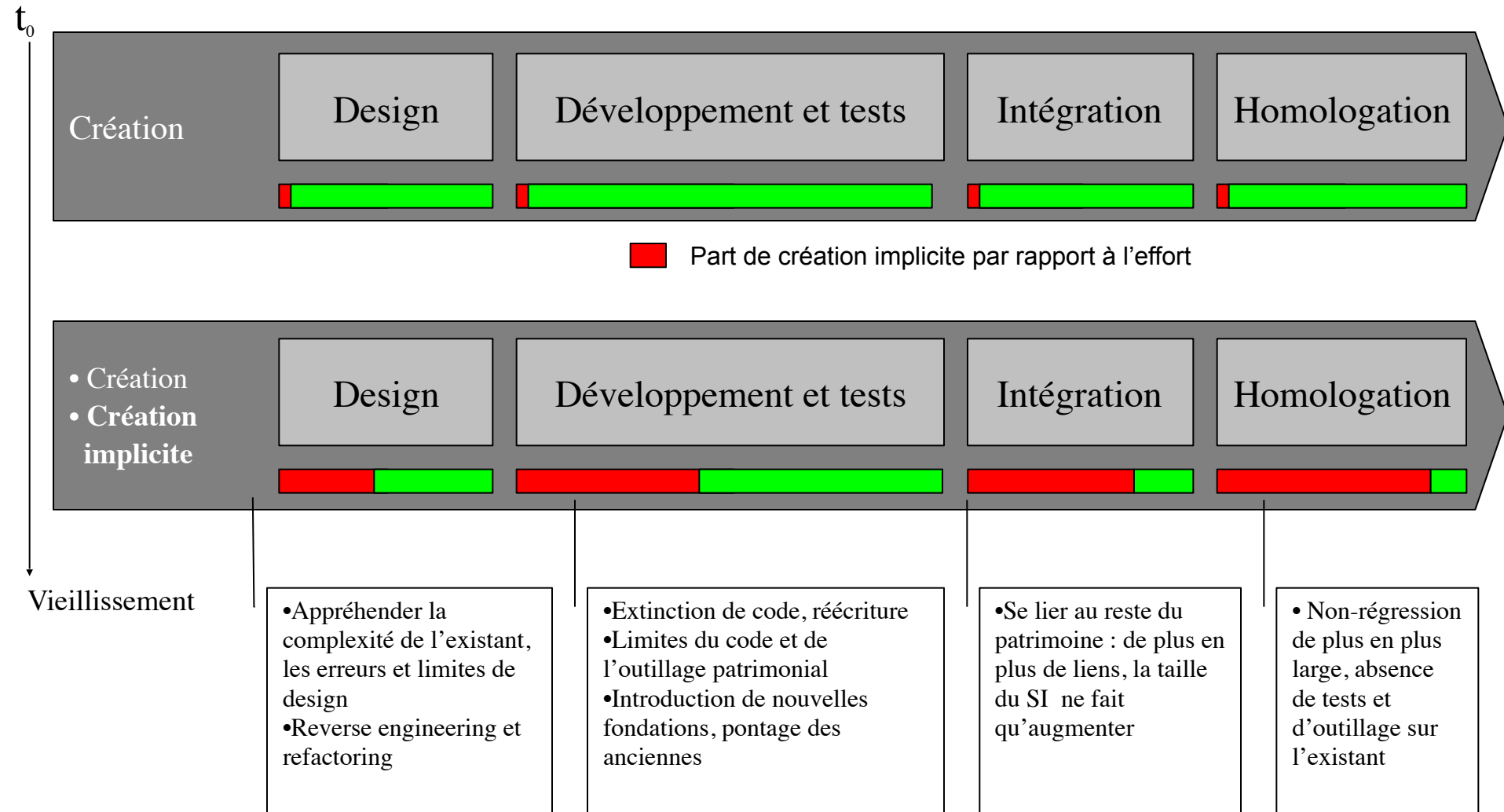




# La productivité se dégrade avec le temps !



# La productivité se dégrade avec le temps !



# Baisse de la productivité aux études .. un moindre mal

- **Développer = 10% budget IT**

- Etudes (*build*) = 1/3 budget, Production (*run*) = 2/3
- Dans les études, 2/3 = analyser, intégrer, tester, déployer



Direction des  
Systèmes d'Information



**Direction  
des Etudes (build)**

Projets

Architecture & infrastructure



**Direction de la  
Production (Run)**

Administration & Infrastructure

Exploitation

- **Ce tiers du budget cristallise pourtant toute la crispation sur l'informatique**

- C'est donc plus *l'adéquation* et les *délais* qui suscitent une légitime douleur\*

- **Pendant ce temps, derrière un apparent changement rapide, les rites réels n'ont pas évolué depuis 30 ans**

- Downsizing, PC, Unix, IP, Web, Java, XML, IDE ...
- Cascade : mythe du logiciel sur plan & d'un processus de construction normé aux segments interchangeables



- **Le mal semble plus profond ...**

\* Standish Group : le taux de projets en échec ou mitigés, est passé de 84% à 71% en 10 ans ...

# La cascade



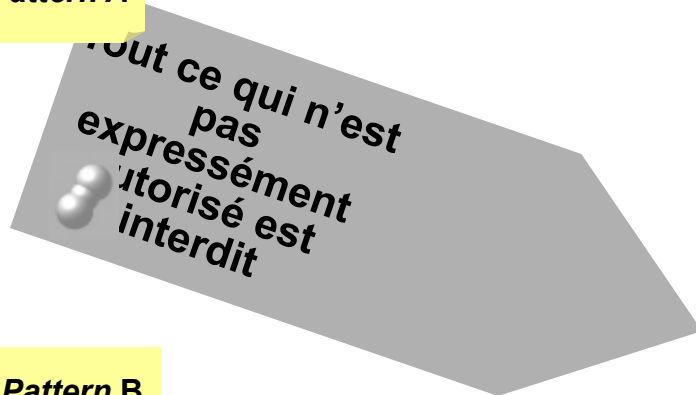
- **On peut en rire ... (extrait de Waterfall 2006)**
  - *Pair Managing: Two Managers per Programmer* by Jim Highsmith
  - *User Interaction: It Was Hard to Build, It Should Be Hard to Use* by Jeff Patton
  - *wordUnit: A Document Testing Framework* by Kent Beck
- **On peut en expliquer les causes visibles**
  - effet papillon - petite cause, grands effets -, coût du changement non linéaire (Barry W. Boehm, 1981)
- **On a du mal à en révéler les causes profondes**
  1. Le confort : *il ne sera pas dit que j'ai fait une erreur avant de transférer la charge, le risque, la responsabilité à l'équipe suivante en aval : ne jamais perdre, ne jamais apprendre*
  2. L'impossibilité du cahier des charges (Wittgenstein) : *les mots ne sont pas univoques*

# Changer ?

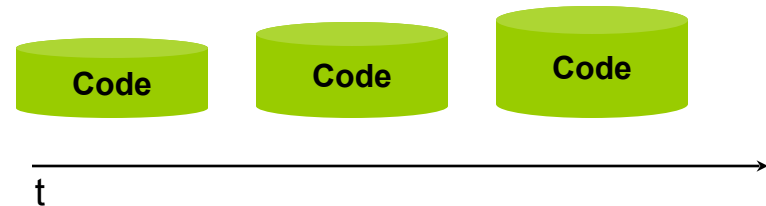
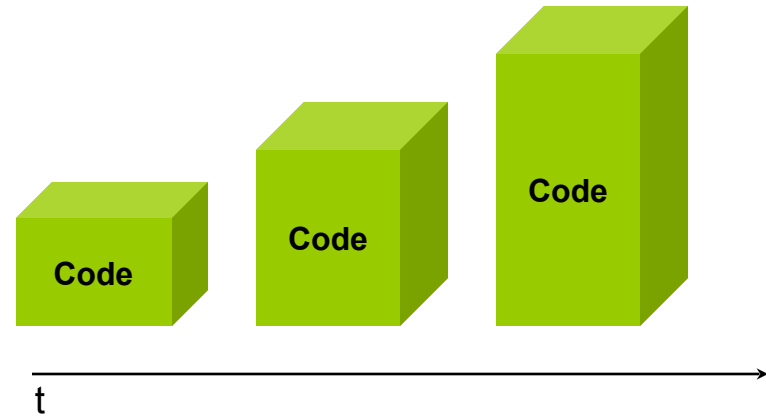
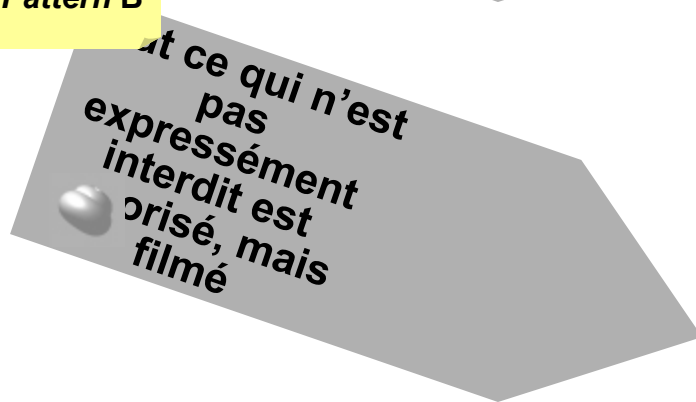


# Le SI se développe organiquement autour de *patterns*

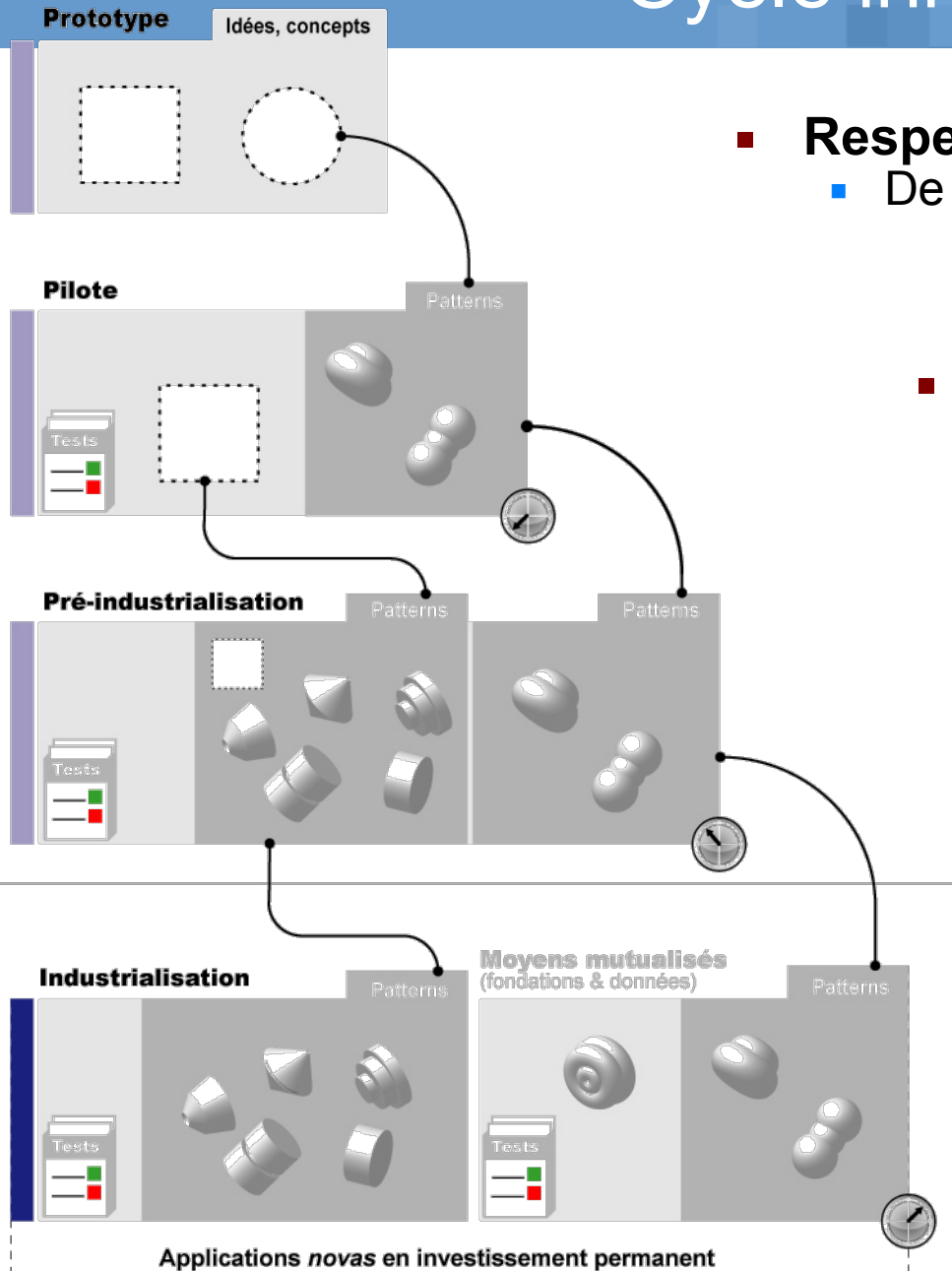
Pattern A



Pattern B



# Cycle innovation - rationalisation



- **Respect du cycle de vie d'un produit**
  - De la niche sur-mesure à l'offre de masse

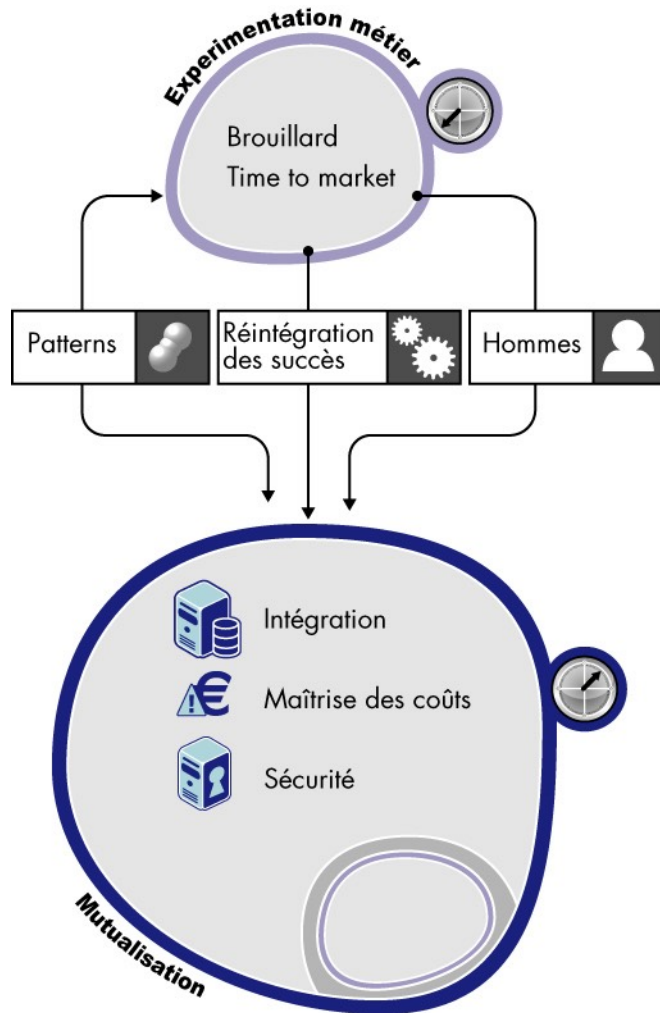
- **Le *cahier des charges* impossible initialement se matérialise progressivement**

- Sous la forme d'un système réellement utilisé ...
- Sous la forme de tests ...
- Sous la forme de *patterns* ...

- **Le cycle se termine par l'*intégration marginale* de l'innovation dans la zone mutualisée**



# Changer l'organisation du *build* : la frontière



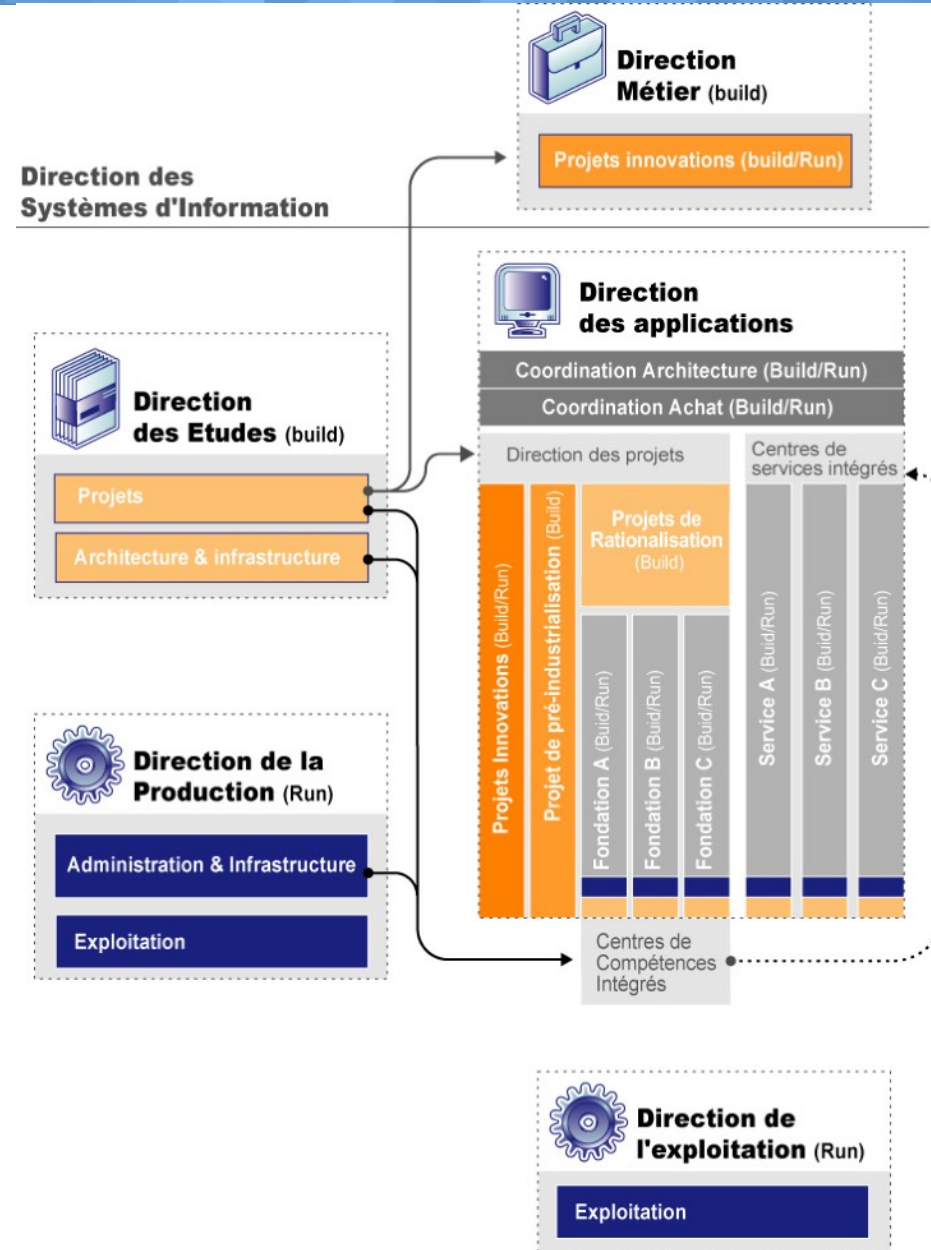
- **Des orientations différentes pour des domaines différents ...**
  - Domaines peu connus : nouvelles niches, nouvelles approches du client, rupture technique ouvrant de nouveaux horizons ..
  - Domaines connus : comptabilité, facturation, GRH ...
- **Plutôt qu'une seule offre de la DSI, proposons deux offres**
  - Une méthode adaptée au « peu de contraintes / temps court »
  - Une méthode adaptée au « beaucoup de contraintes / temps long »



# Changer l'organisation : dépasser la frontière *build/run*

## ■ Pour des frontières plus responsabilisantes

- Direction des applications
  - Responsable du patrimoine applicatif (« MOA »+ « MOE »)
  - Jugé sur l'actif (valeur métier) et le passif (dettes techniques)
  - Pas sur la documentation ou des coûts annuels partiels
- Centre de compétences intégré
  - Responsable de fondations (Linux/J2EE/Oracle) sous l'angle *build* ET *run*
  - Jugé sur le service global rendu à la DirApp et à la DirExp
  - Pas sur la beauté des concepts ou la raideur des normes



# Changer les gens ...

## ■ Culture du savoir

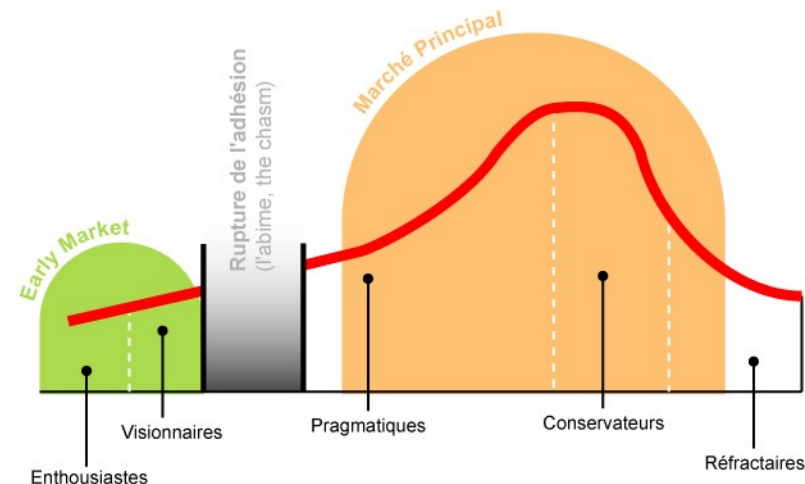
- Passer du repli identitaire/refuge de l'expertise à l'échange de savoirs
- Communication par pattern : synthèse à jour pour chaque domaine du SI

## ■ Culture risque/valeur

- Passer du contrat tout ou rien et des coûts partiels en J/h, à la construction incrémentale par *dé-excelisation*, *valorisation*, *tests*

## ■ Culture commerciale

- Technique : passer de la norme à la solution
- Application : passer du cahier des charges à la traversée de l'abîme



# Changer les gens ...

## ■ Culture du savoir

- Passer du repli identitaire/refuge à l'échange de savoirs
- Communication par pattern : synthèse à jour pour chaque domaine du SI

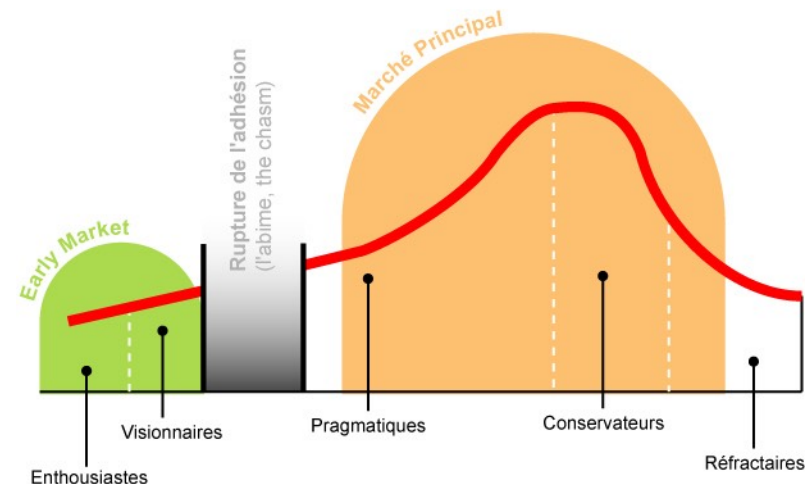


## ■ Culture risque/valeur

- Passer du contrat tout ou rien et des coûts partiels en J/h, à la construction incrémentale par *dé-excelisation*, *valorisation*, *tests*

## ■ Culture commerciale

- Technique : passer de la norme à la solution
- Application : passer du cahier des charges à la traversée de l'abîme



# Changer les gens ...

## ■ Culture du savoir

- Passer du repli identitaire/refuge à l'échange de savoirs
- Communication par pattern : synthèse à jour pour chaque domaine du SI

Rupture de confort

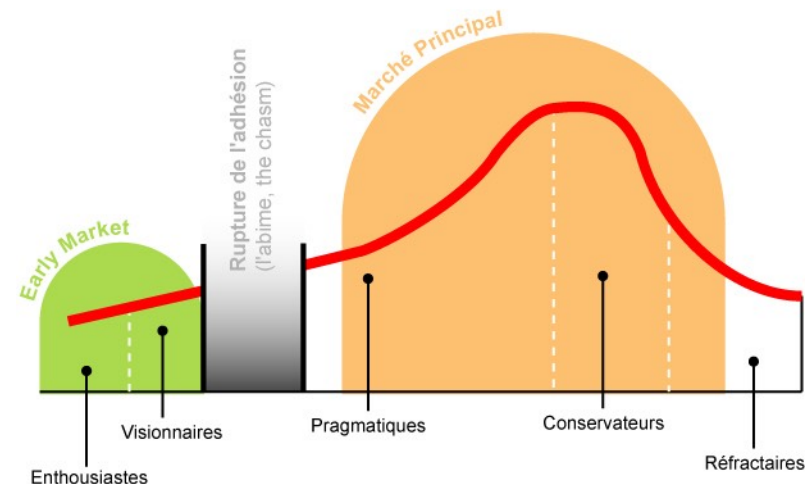
## ■ Culture risque/valeur

- Passer du contrat tout ou rien à des coûts partiels en J/h, à la construction incrémentale
- *Excelisation, valorisation, tests*

Rupture de confort

## ■ Culture commerciale

- Technique : passer de la norme à la solution
- Application : passer du cahier des charges à la traversée de l'abîme



# Changer les gens ...

## ■ Culture du savoir

- Passer du repli identitaire/refuge à l'échange de savoirs
- Communication par pattern : synthèse à jour pour chaque domaine du SI

Rupture de confort

## ■ Culture risque/valeur

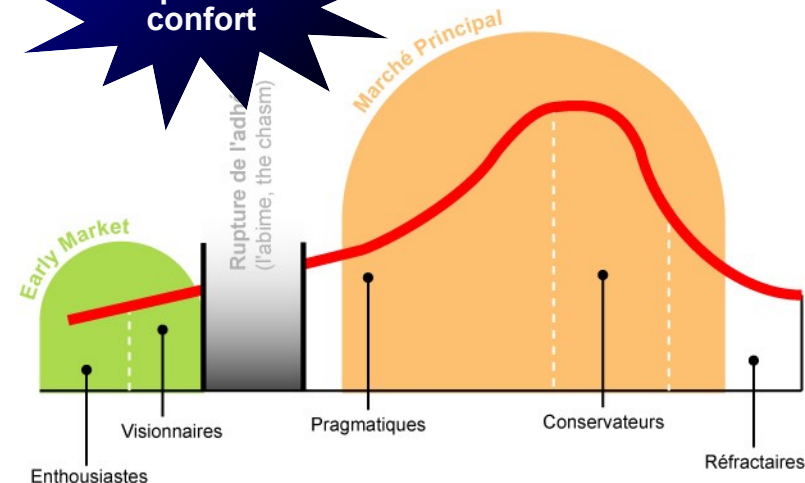
- Passer du contrat tout ou rien à des coûts partiels en J/h, à la construction incrémentale
- *Excelisation, valorisation, tests*

Rupture de confort

## ■ Culture commerciale

- Technique : passer de la norme à la solution
- Application : passer du cahier des charges à la traversée de l'abîme

Rupture de confort



# Changer les gens ...

## ■ Culture du savoir

- Passer du repli identitaire/refuge à l'échange de savoirs
- Communication par pattern : synthèse à jour pour chaque domaine du SI



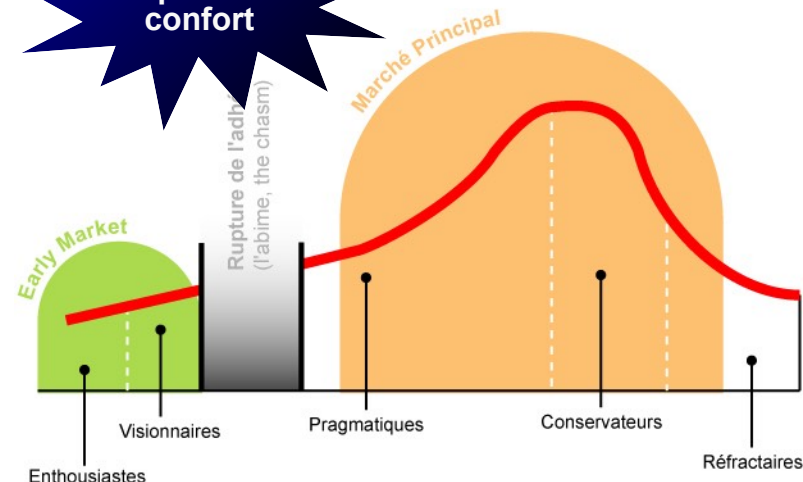
## ■ Culture risque/valeur

- Passer du contrat tout ou rien à des coûts partiels en J/h, à la construction incrémentale
- *Excelisation, valorisation, tests*



## ■ Culture commerciale

- Technique : passer de la norme à la solution
- Application : passer du cahier des charges à la traversée de l'abîme





# Peut-on changer tout le monde ?



# Peut-on changer tout le monde ?

- **OUI, les vraies questions sont *en combien de temps* ?**
  - Tri sélectif : 10 ans,  $\frac{3}{4}$  de la population française aujourd'hui





# Peut-on changer tout le monde ?

- **OUI, les vraies questions sont en combien de temps ?**
  - Tri sélectif : 10 ans,  $\frac{3}{4}$  de la population française aujourd'hui
- **Et avec ou sans guerre anciens/modernes ?**



# Changer le système

- **Commençons par diminuer le passif - les dettes techniques**
  - logiciels sans tests, logiciels stratifiés à alléger, logiciels redondants à fusionner ...
- **L'investissement dans ces chantiers de refactoring est un mal nécessaire au maintien de la productivité**
  - S'alléger et industrialiser l'architecture technique
- **Une enveloppe annuelle de 10% des études semble un minimum**
  - Plutôt sous la forme d'une ligne budgétaire en j/h dans laquelle chaque projet peut piocher

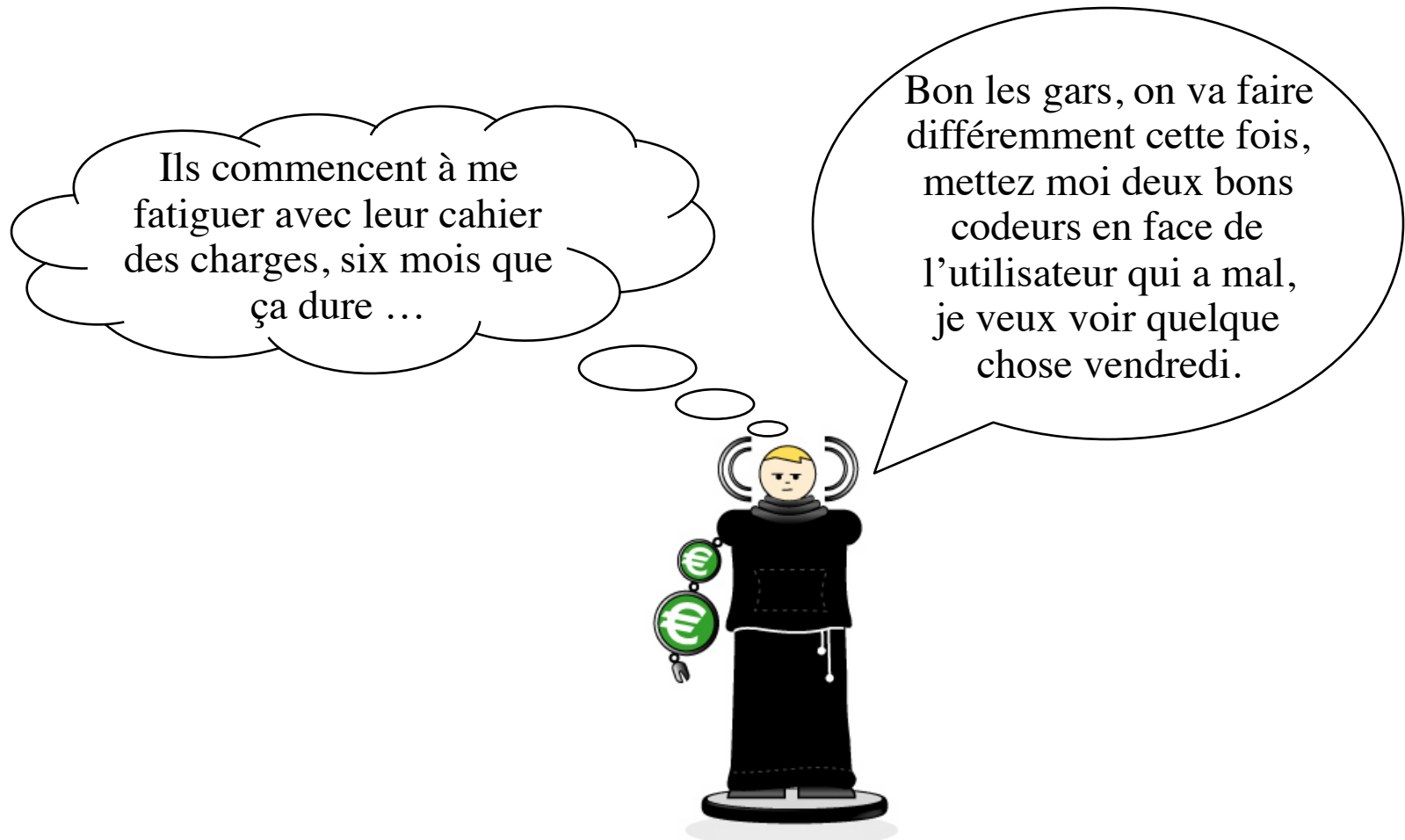


Et demain matin ?



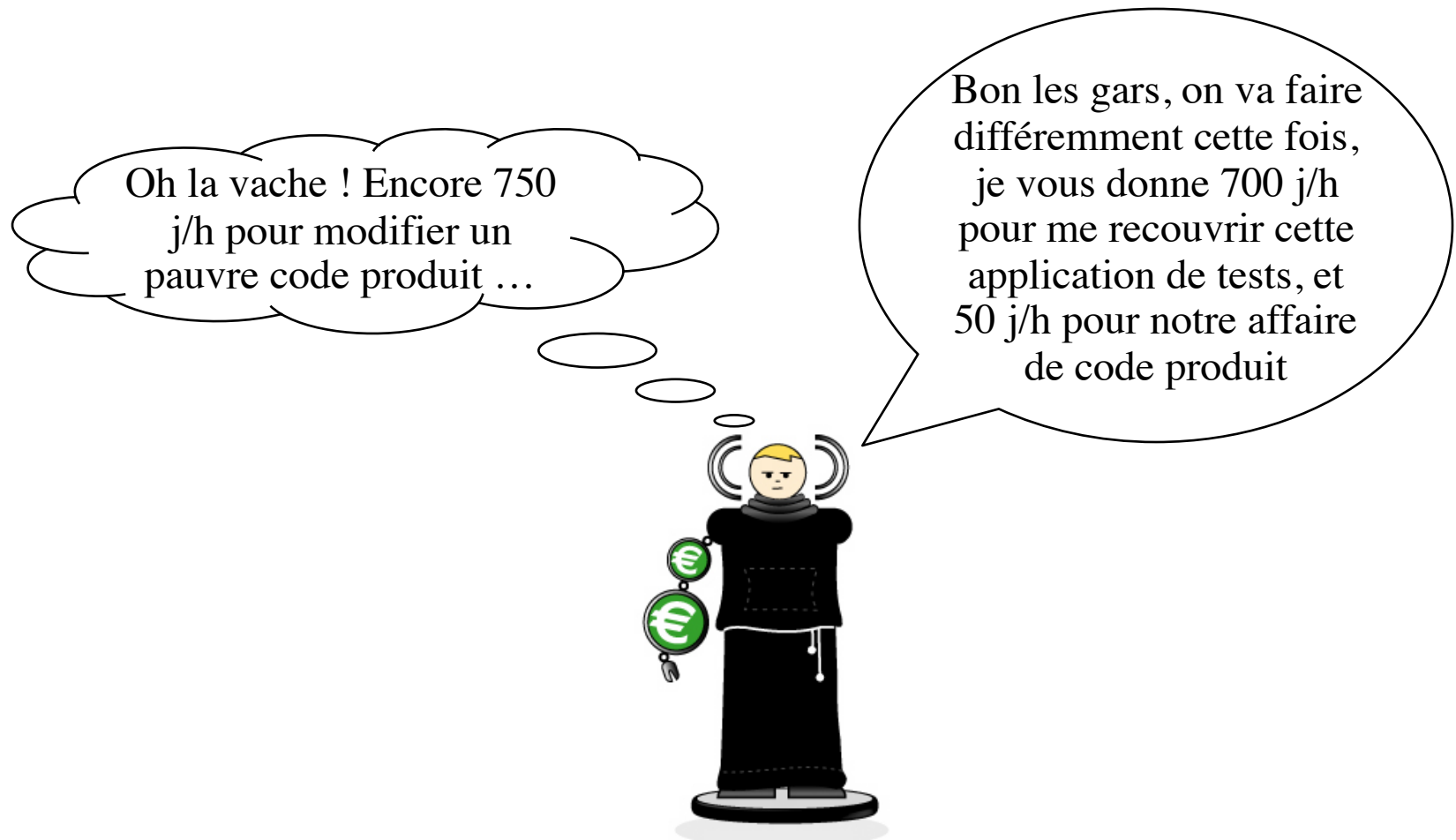
# Demain matin, réveil du manager

- Lancer un premier projet *incrémental* (un deuxième...) et piloté par les tests



# Demain matin, réveil du manager

- Rembourser une *dette technique* identifiée et douloureuse



# Demain matin, réveil du manager

- **Demander aux gestionnaires de domaines métier des *analyses par pattern* de leur patrimoine**

Imbitable la présentation de dédé ... si moi je déchiffre pas, je vois pas comment mes troupes ou Gerboulade Partner vont y comprendre quelque chose

...



Bon les boss, faut faire plus synthétique, je mets dans vos objectifs la publication d'une présentation de synthèse recensant les principaux traits structurants de vos SI. Organisez des cycles de conférences et de formation, faites évaluer les sessions.

# Demain matin, réveil du manager

- A partir d'individus de la production et des études, constituer un embryon de **CCI** sur des fondations demandées par les projets

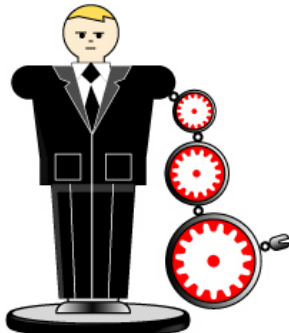
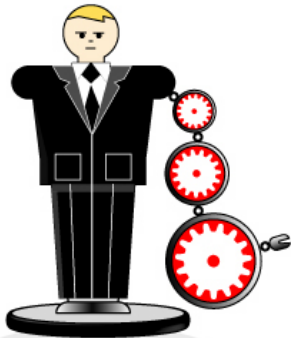
L'administrateur et l'architecte technique ... il y en a toujours un pour m'expliquer que l'autre a tort, forcément c'est pas la même paroisse ...



**Direction de la Production** (Run)



**Direction des Etudes** (build)

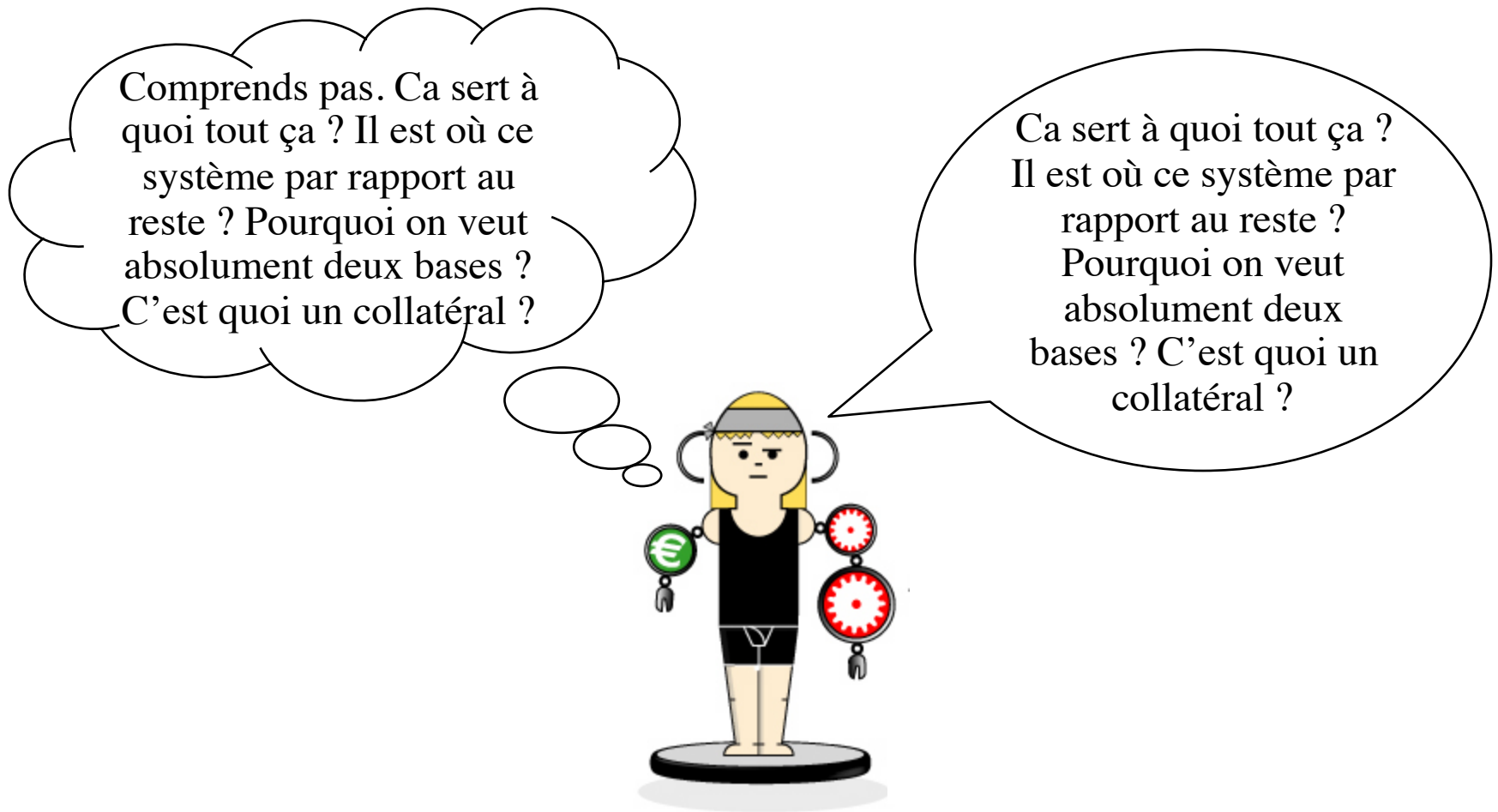


Bon les gars, j'ai décidé que vous bosseriez ensemble pour offrir une prestation de soutien au développement et à l'exploitation sur la plate-forme Solaris/C/PowerBuilder/Sybase



# Demain matin, réveil de l'opérationnel

- **Ne plus accepter de coder sans design général compréhensible et synthétique**



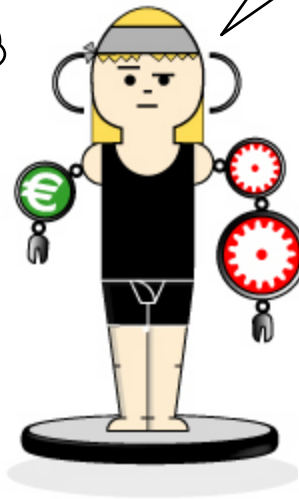


# Demain matin, réveil de l'opérationnel

- **Ne plus accepter de coder dans des lieux à forte dette technique**

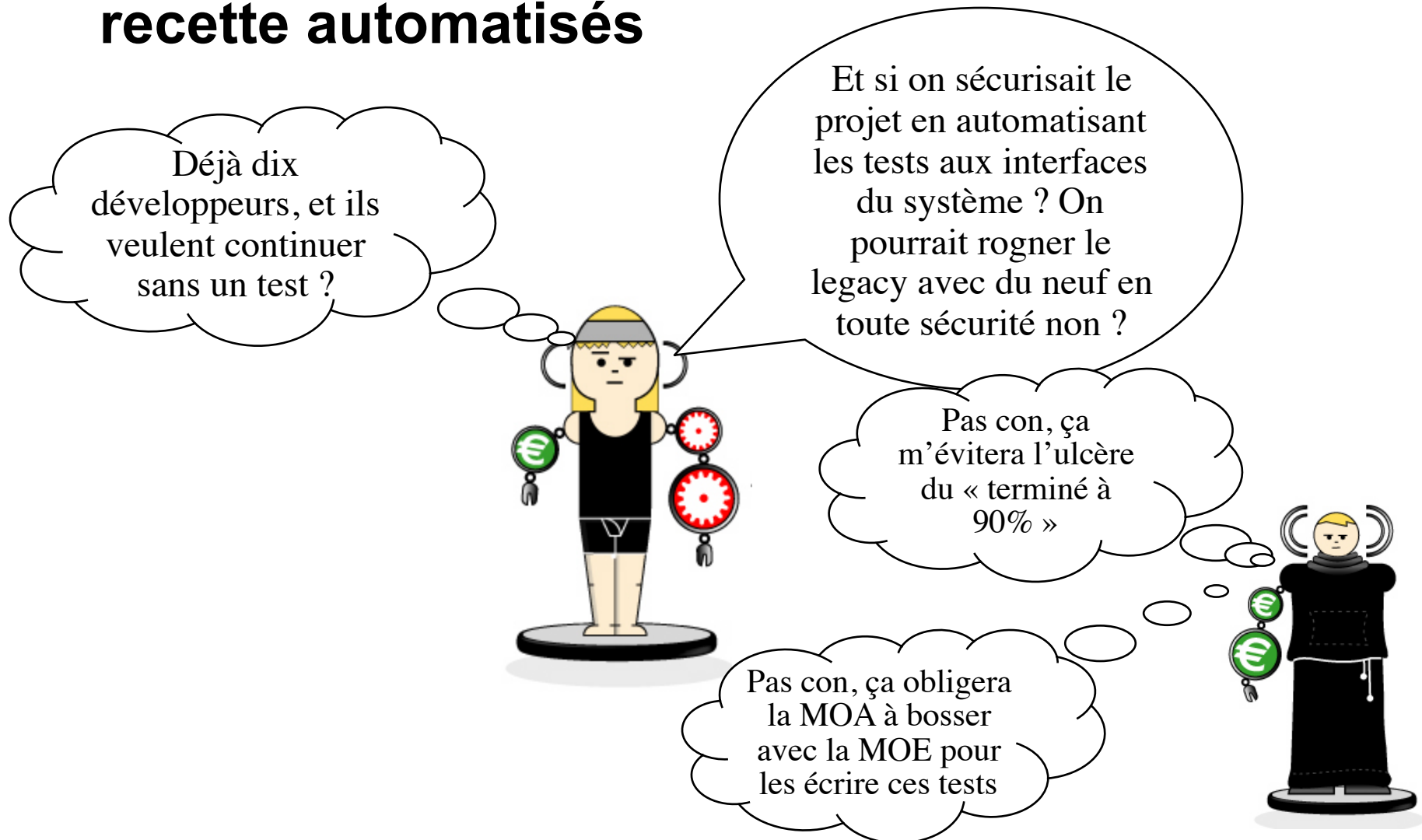
Waw le chantier. 20 ans que ça fermente là-dedans, au moins. Et ils veulent continuer sans un test ? Avec cette application ultra-redondante ? Avec cet outillage pourri ?

Et si on dépensait un tout petit peu d'argent pour nettoyer le lieu de travail ? Il y a ça en particulier, ...



# Demain matin, réveil de l'opérationnel

## ■ Ne plus accepter de coder sans tests de recette automatisés





MERCI



# Pour en savoir plus ...

## Les livres Blancs

- Architecture orientée services
- Gouvernance de la données
- Architecture d'applications - La solution .NET
- Le Livre Blanc de Architectures des Systèmes d'Information
- Le Livre Blanc de la Sécurité
- Le Livre Blanc des Serveurs d'applications
- Le Livre Blanc de l'EAI - Intégration des applications
- Le Livre Blanc de l'IRM - Gestion de la relation sur Internet

<http://www.octo.com/>



Laurent Brisse - Julien Cabot - Gilles Laborderie - Pierre Pezziardi - Christophe Thibaut

## Une Politique pour le Système d'Information

Descartes - Wittgenstein - (XML)



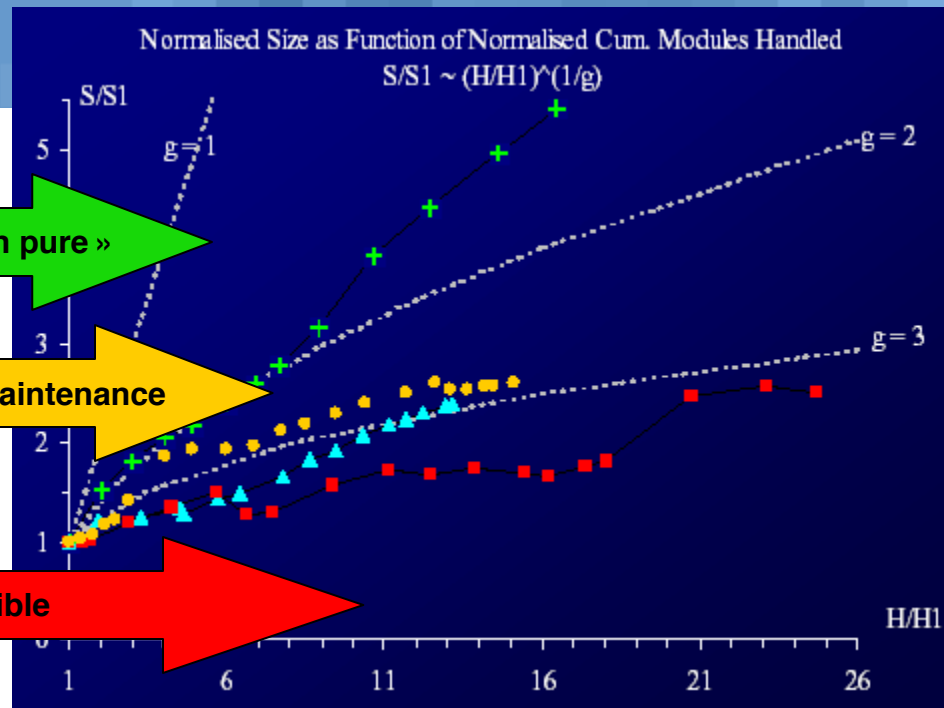
## Les ouvrages



Productivité « comme en création pure »

Productivité moyenne en maintenance

Productivité faible



Source : JF Ramil (2002)  
<http://mcs.open.ac.uk/jfr46>

- Certaines recherches ont permis de modéliser ce vieillissement des applications et de confronter ces modèles à la réalité
  - Travaux de Lehman, Ramil, Cocomo II
  - Travaux empiriques, fondés sur des analyses a posteriori (données 1980-2000 – voir Annexe)