



**HAL**  
open science

## La démarche Agile, Science ou Magie ?

Camille Salinesi

► **To cite this version:**

Camille Salinesi. La démarche Agile, Science ou Magie ?. Conférence Agile Secteur Public, Oct 2023, Paris, France. hal-04243789

**HAL Id: hal-04243789**

**<https://paris1.hal.science/hal-04243789v1>**

Submitted on 16 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

*Conférence Agile Secteur Public*

*10 Octobre 2023*

# La démarche agile, science ou magie ?

Camille Salinesi

*Université Paris 1*

*Panthéon – Sorbonne*





2 questions

# Qui suis-je ?

- Professeur des Universités = enseignant - chercheur
  - Génie logiciel, Systèmes d'Information, IA, *Ingénierie des Exigences*
  - 300 publications, 20 docteurs
  - Ancien membre du Comité Exécutif de l'IREB
  - VP Université Paris 1
- Camille.Salinesi@univ-paris1.fr





THALES

*DS* DASSAULT  
SYSTEMES



  
Stago

edf



Alcatel-Lucent



**KTM**   
ADVANCE



france telecom



 Nomia



# Mes centres d'intérêt

Sensibilité et  
adaptation au  
contexte

Variabilité

Exigences  
spécifiques au  
domaine

Sécurité et  
compliance

Configuration et  
re-configuration

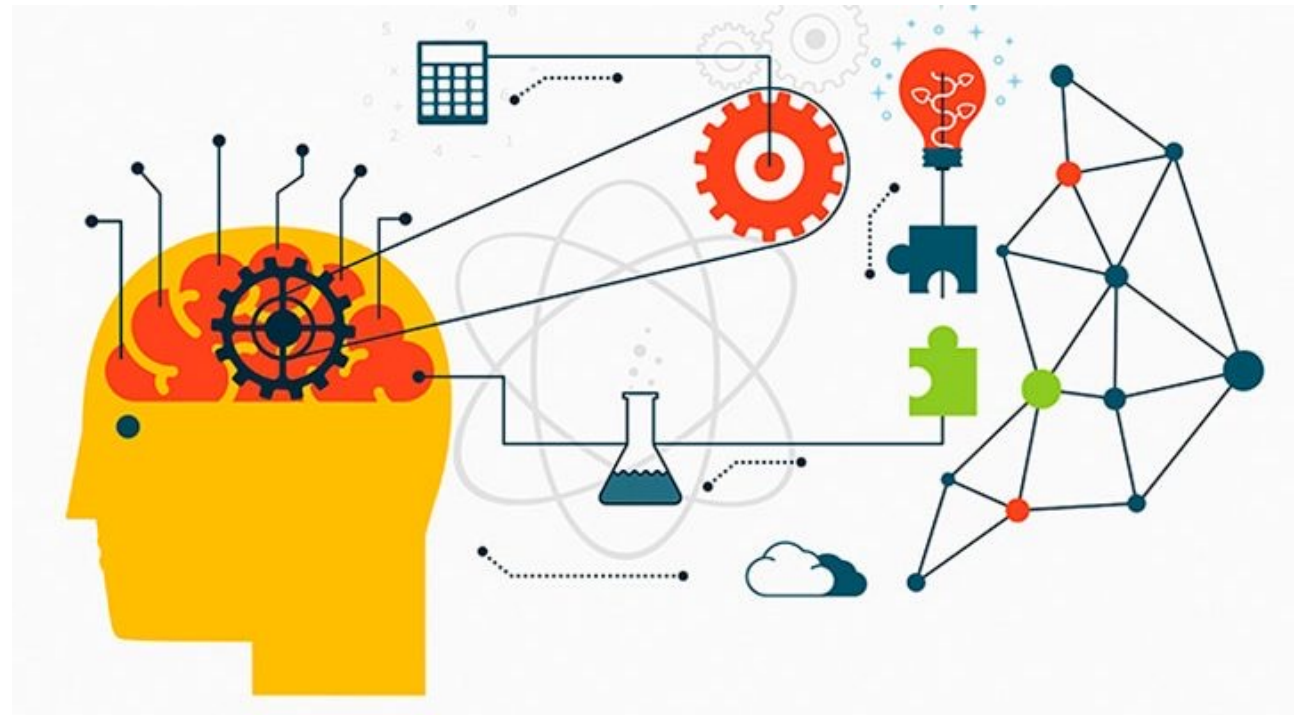
Innovation

Réutilisation

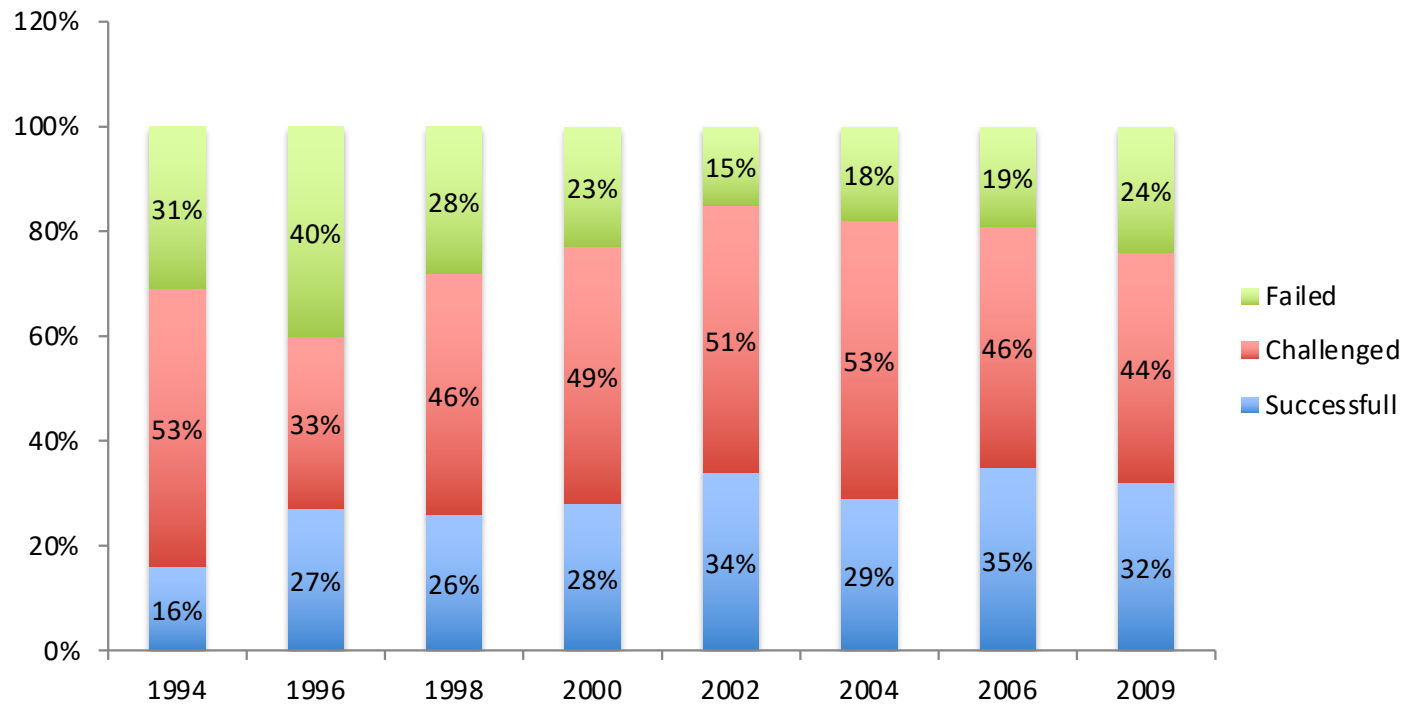
Recom-  
mendation

IA

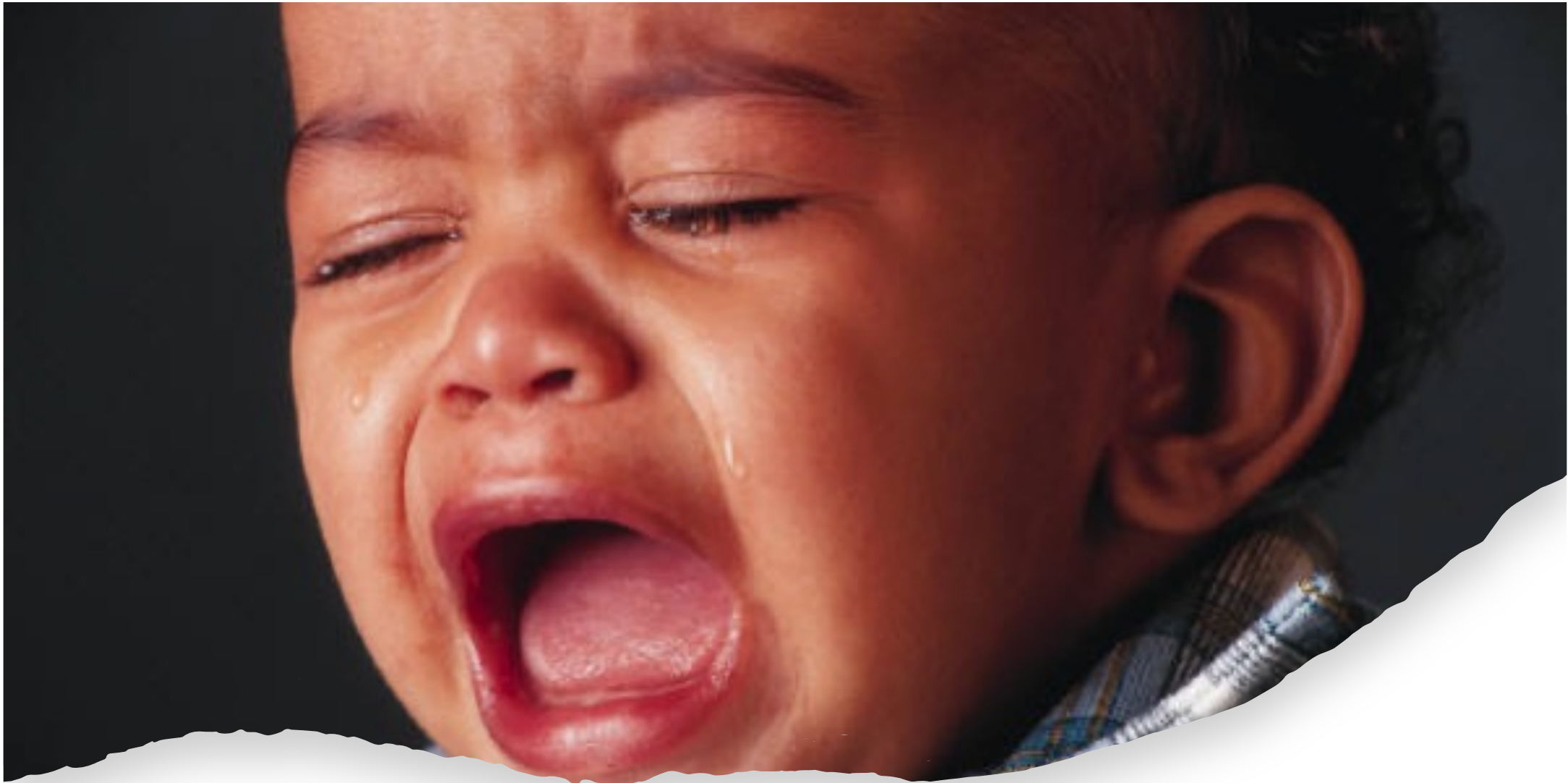
Que dit la recherche au sujet de notre manière de conduire les projets IT?



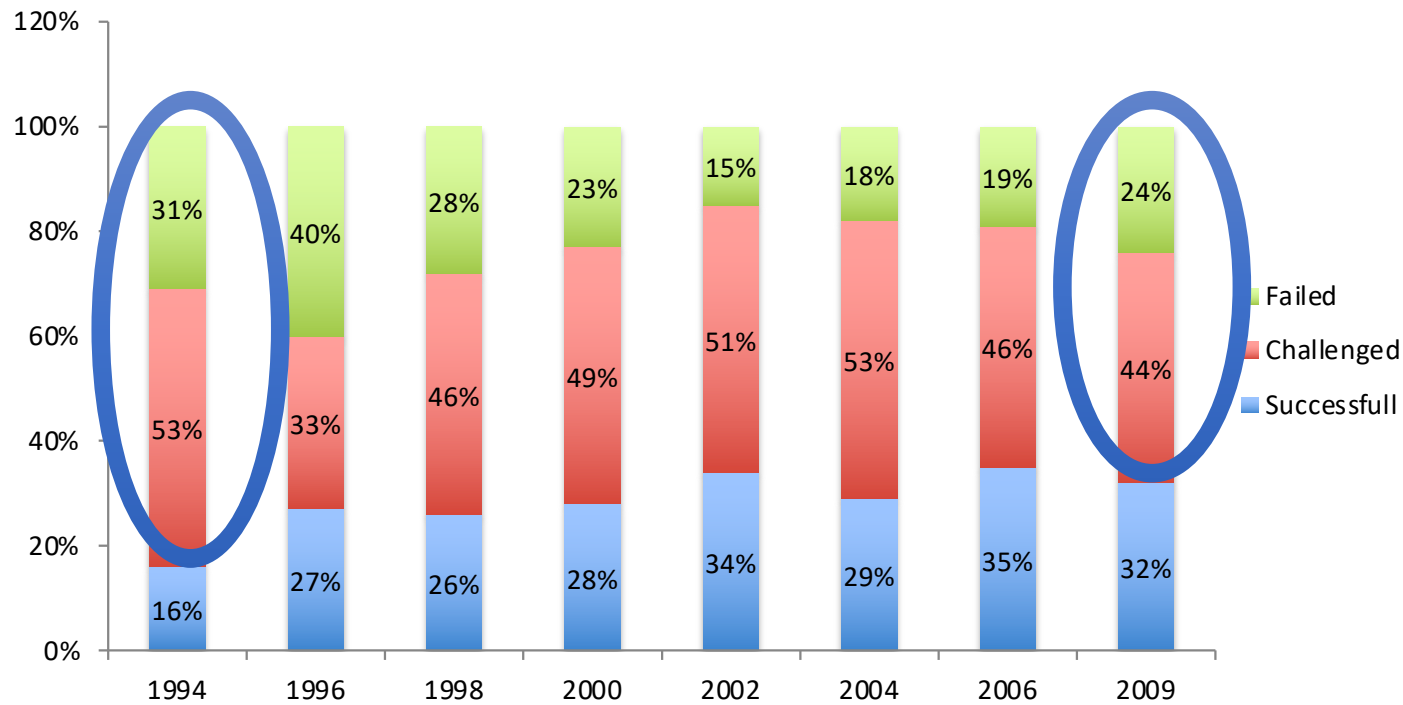
En 15 ans *2 fois plus* de projets sont *des succès!*



Source: rapport Chaos v3, Standish Group



**68%** des projets ne sont toujours  
*pas des succès!*



Source: rapport Chaos v3, Standish Group



## Pourquoi ?

“Tout le monde sait de quoi on a besoin”

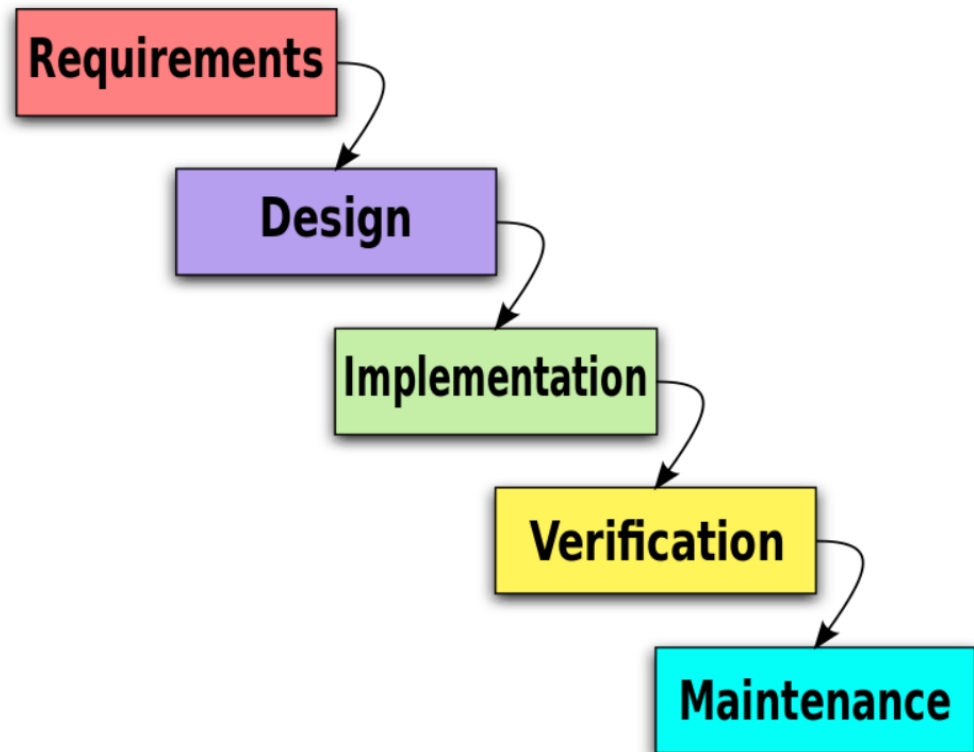
“Les besoins doivent absolument rester stables”

“Il suffit de prendre la solution X”

“Il faut vite passer au développement”

“L’analyse doit être conduite par les informaticiens car c’est eux qui savent faire”

# Le coupable est . . .

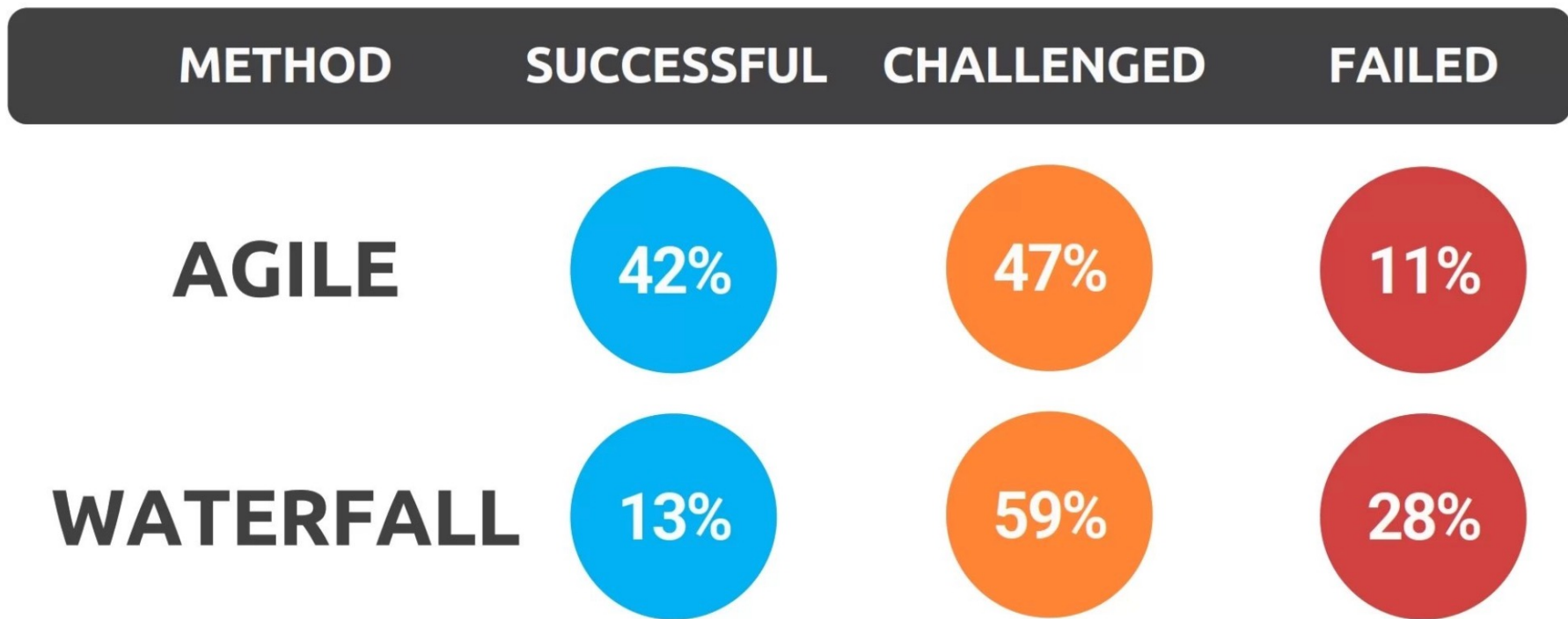






« ça ne se passera  
pas chez nous! »

**58%** des projets *menés en agile* ne sont *pas des succès!*



[WWW.VITALITYCHICAGO.COM](http://WWW.VITALITYCHICAGO.COM)

Source: Standish Group Report 2020

L'impact des **compétences en gestion de projets** sur le succès dans les projets menés **en agile** est faible, voire **négatif** !

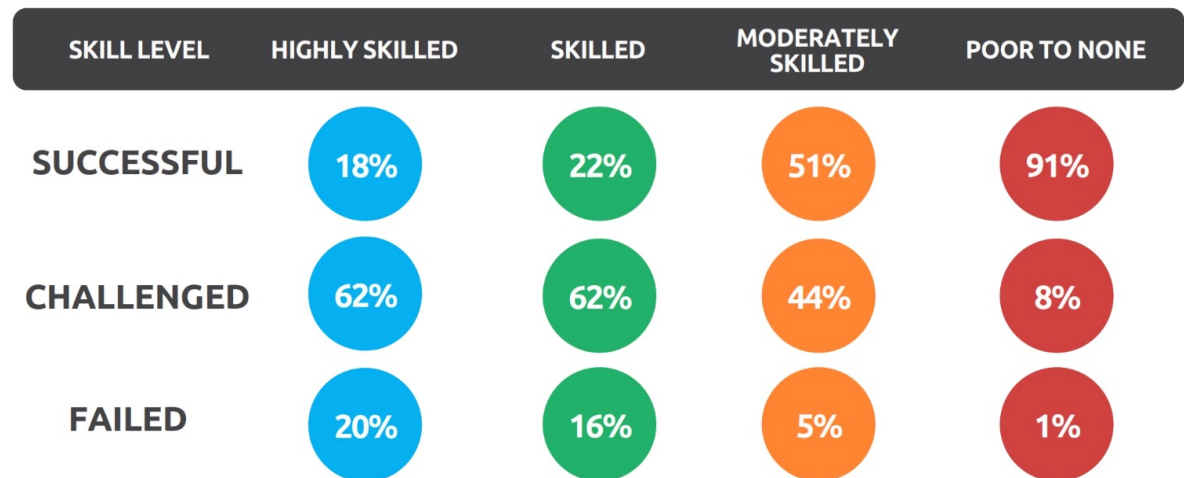


Chart 10.2: The numbers show that the quality of a project manager's skills has little effect on a positive outcome. The results are based on 10,000 agile projects in the CHAOS database.

[WWW.VITALITYCHICAGO.COM](http://WWW.VITALITYCHICAGO.COM)



# Fred Brooks

*"The hardest single part of building a software system is **deciding** precisely what to build.*

No other part of the conceptual work is as difficult as establishing the detailed technical requirements . . . No other part of the work so cripples the resulting system if done wrong.

No other part is as difficult to rectify later"

1999



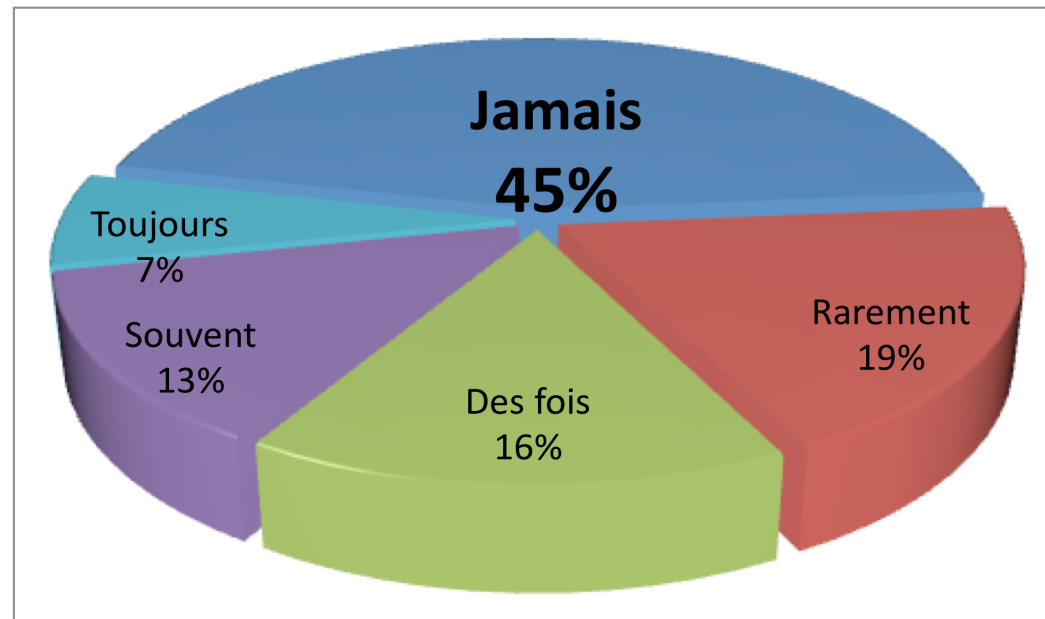
# Standish Group Chaos Report (2020) Beyond Infinity

16/10/2023

« Our current data does not support this conclusion [that a good project manager is essential to any project]. Highly skilled project managers have a success rate of only 23% and a failure rate of 19%; but projects with no project manager at all show a success rate of 58% and a failure rate of only 9%. Why is this? The **real key to project manager success is a project manager who can make quick decisions**, take risks, and not get bogged down in the project management bureaucracy. »

16

45 % des fonctionnalités *jamais* utilisées!



Source: rapport Chaos v3, Standish Group

L'Ingénierie des Exigences  
est  
une **discipline critique** pour la  
pratique de l'agile!

# Coûts de réparation



Source [Boem & Papaccio, 1988]



# La compétence en IE a une très forte influence sur le résultat du projet

## Cocomo II Project Effort Adjustment Factors

Factor	Very Low	Low	Norm	High	Very High
Requirements Analyst Capability	1,42	1,19	1,00	0,85	0,71
Programmer Capability	1,34	1,15	1,00	0,88	0,76
Personal Continuity (Turnover)	1,29	1,12	1,00	0,90	0,81
Application (Business Area) Expertise	1,22	1,10	1,00	0,88	0,81
Required Software Reliability	0,82	0,92	1,00	1,10	1,26
Product Complexity	0,73	0,87	1,00	1,17	1,34

Source: McConnell, S.: Software Estimation: Demystifying the Black Art, Microsoft Press, 2006



Rassurons nous!



Picture: pixabay.com

Vous considérez vous comme  
une **simple**  
**chambre d'enregistrement?**



A person wearing a white long-sleeved shirt is holding a large, thick stack of colorful folders and papers. The stack is held in front of their chest with both hands. The folders are in various colors including blue, red, yellow, and green. The papers are white and appear to be part of a large document or report. The background is a plain, light-colored wall.

Vous contentez vous de **relayer les décisions** prises par d'autres ?



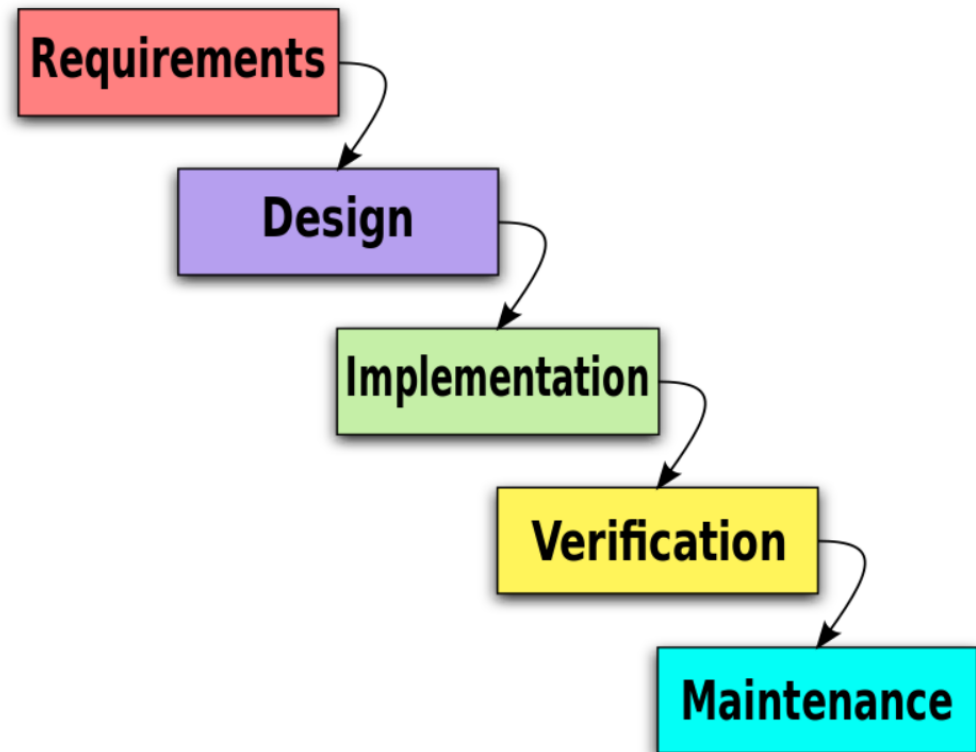


Considérez vous que role soit de **produire**  
**de la documentation** ?

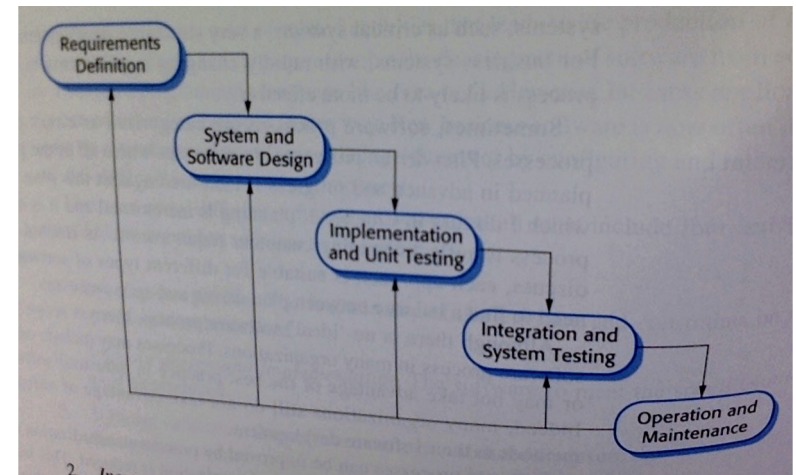
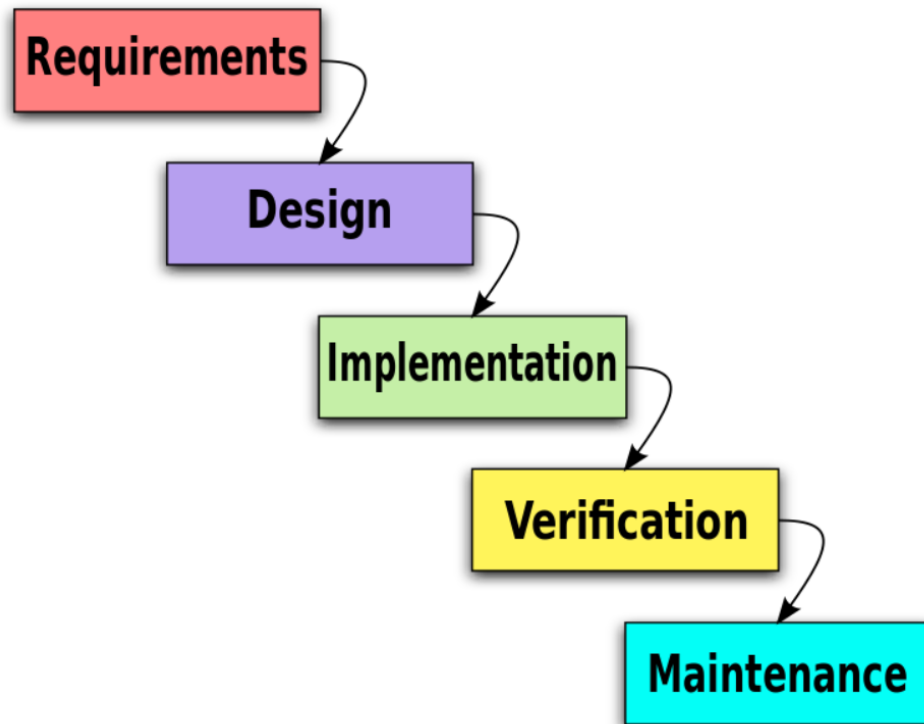
N'y aurait-il pas quelques  
incompréhensions ?



# Quels sont les faits?



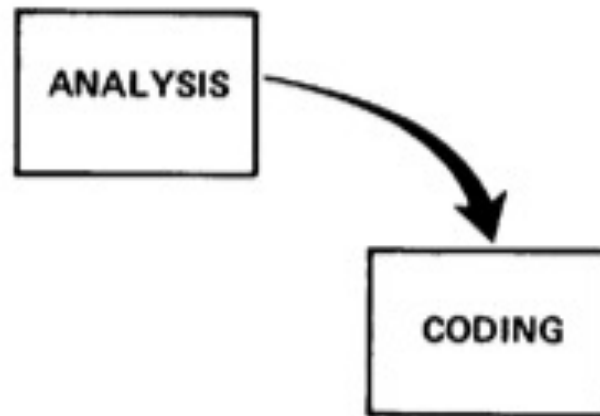
Connaissons nous *vraiment* le coupable ?  
Faites entrer l'accusé !



The Waterfall model.  
From: Ian Sommerville « Software Engineering ». Pearson. 2011



Royce, W. "The software lifecycle model (waterfall model)." *Proc. Westcon*. Vol. 314. 1970.



# La cascade comme métaphore

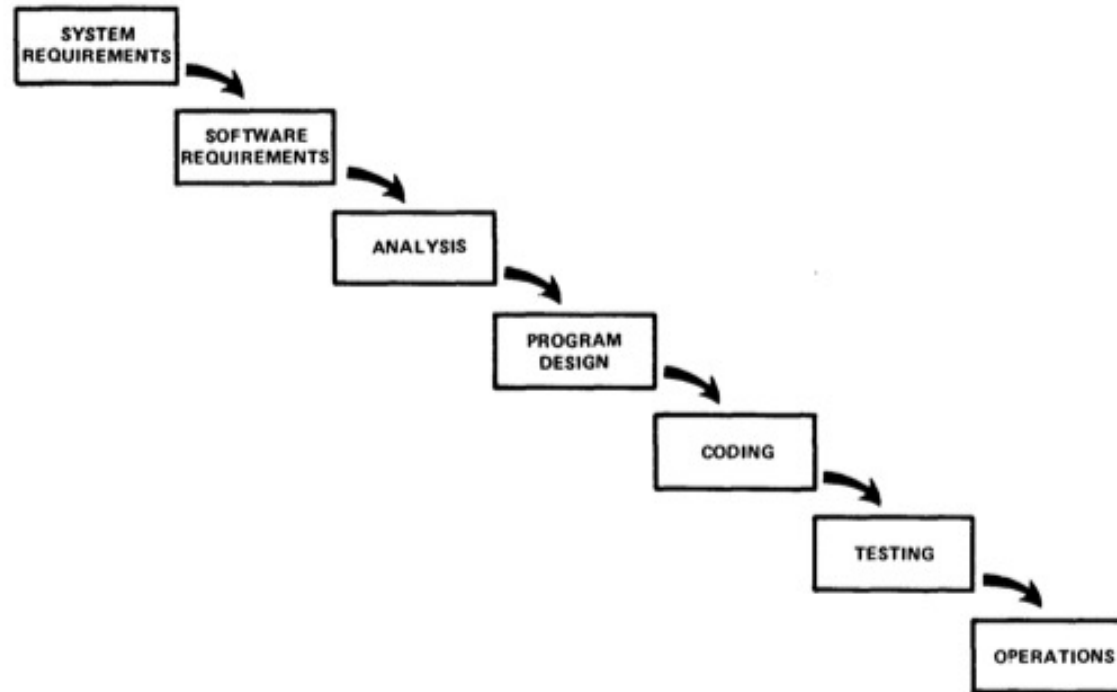
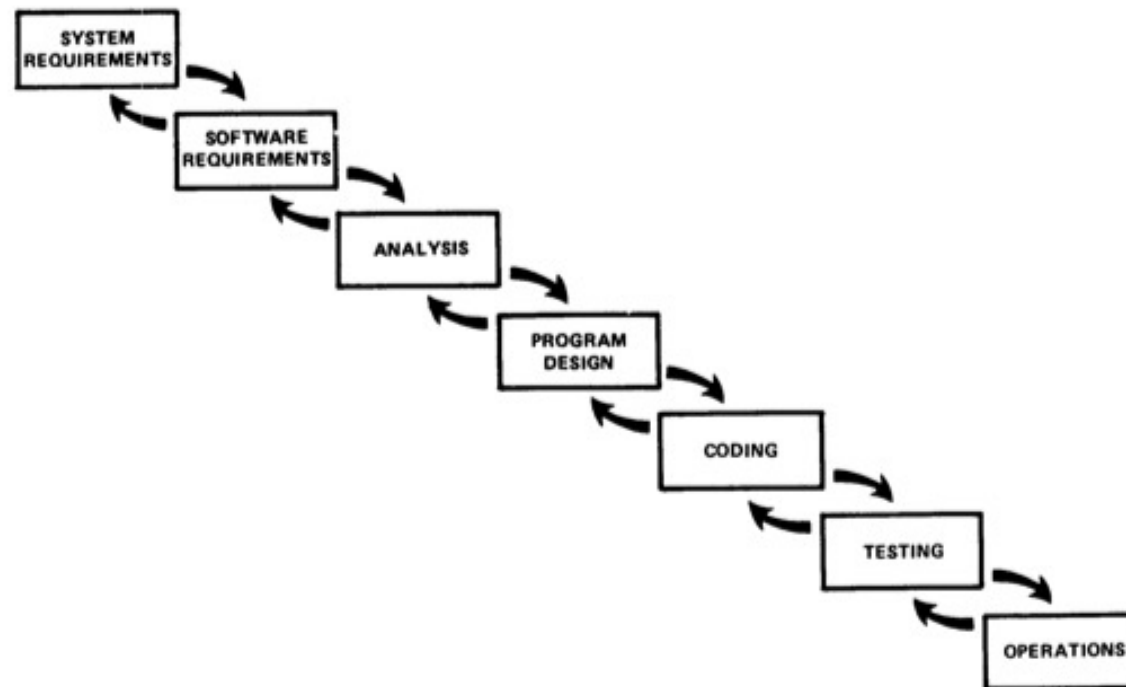


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

# L'itération comme outil flexible de gestion des évolutions



# Le besoin d'adaptabilité

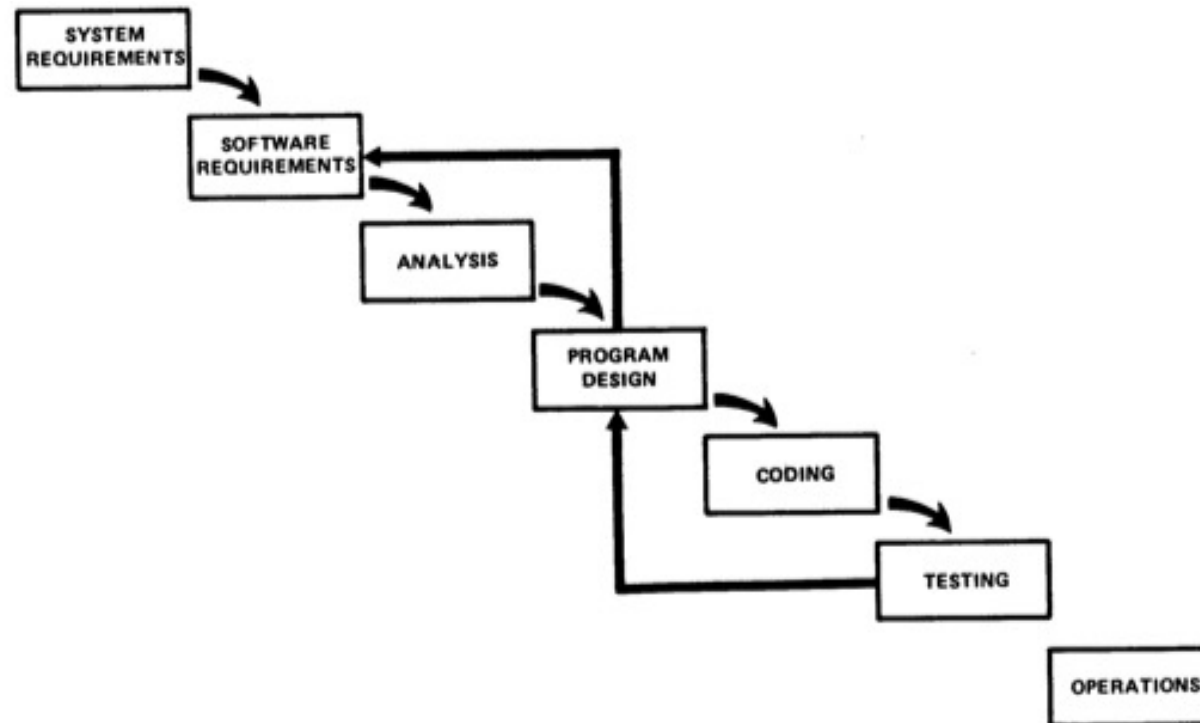


Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.

# La participation des clients / utilisateurs

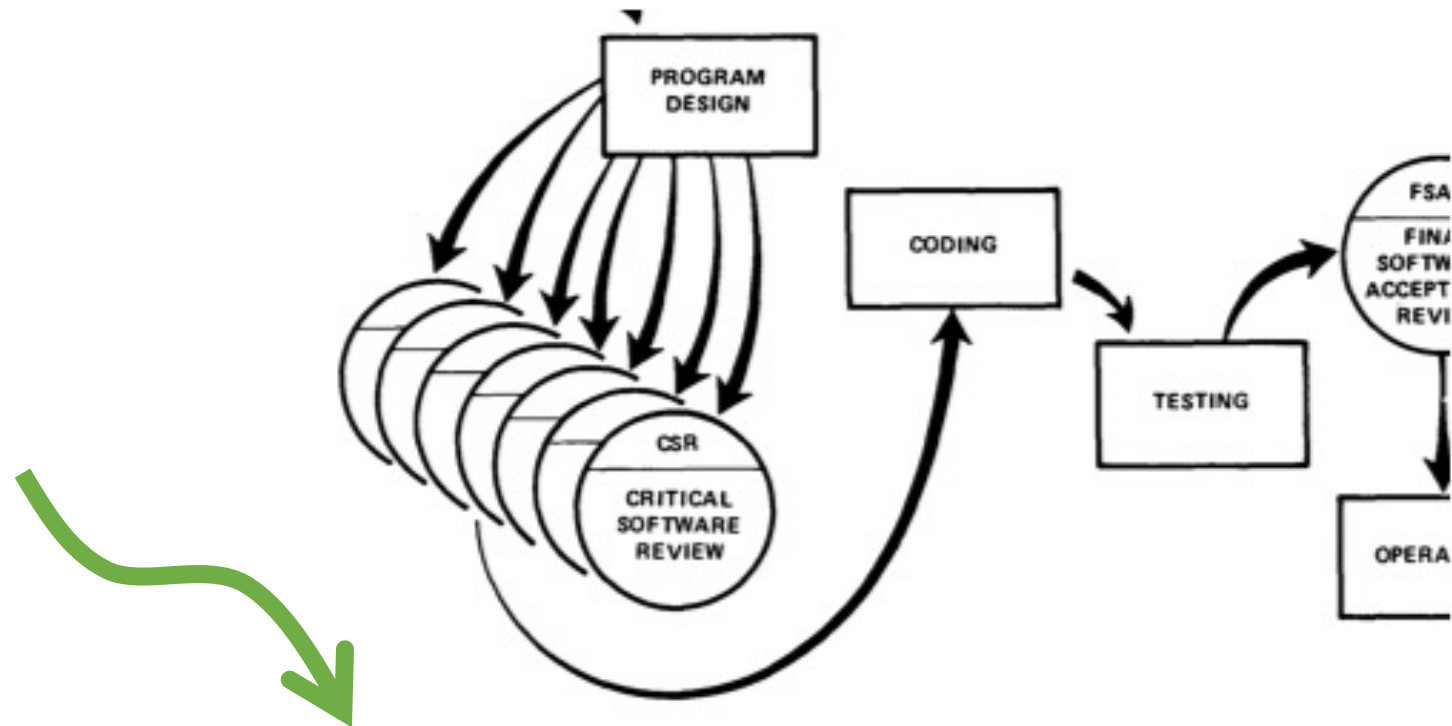


Figure 9. Step 5: Involve the customer – the involvement should be formal, in-depth, and continuing.

# Le prototypage

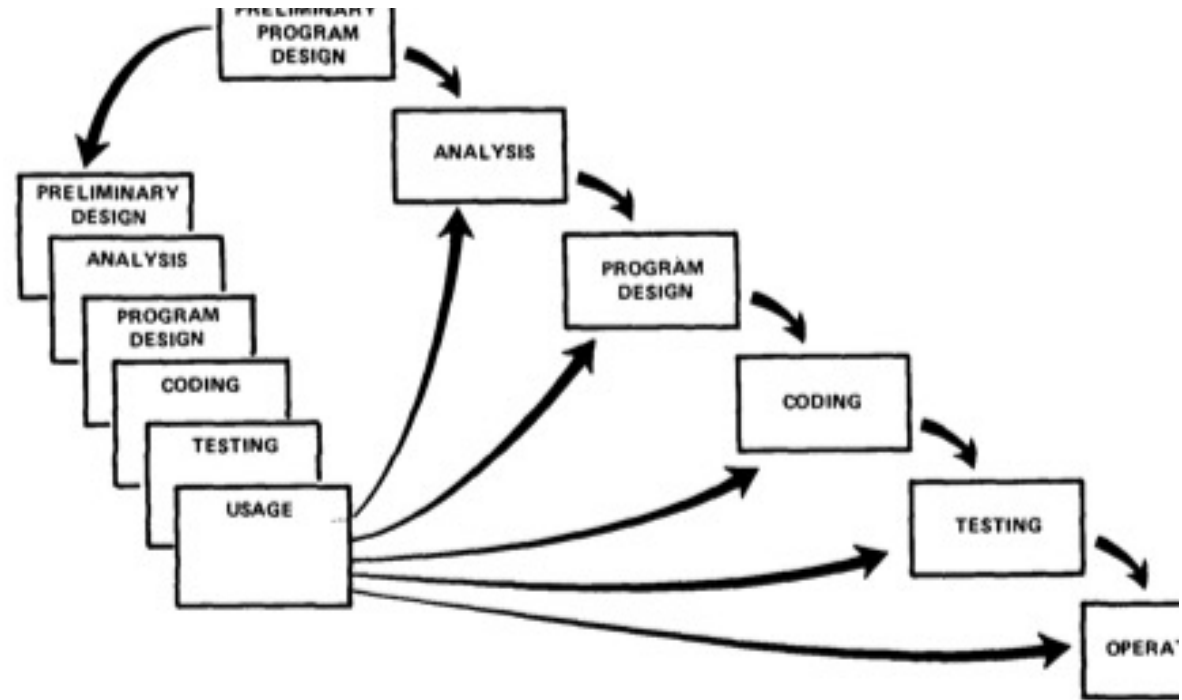


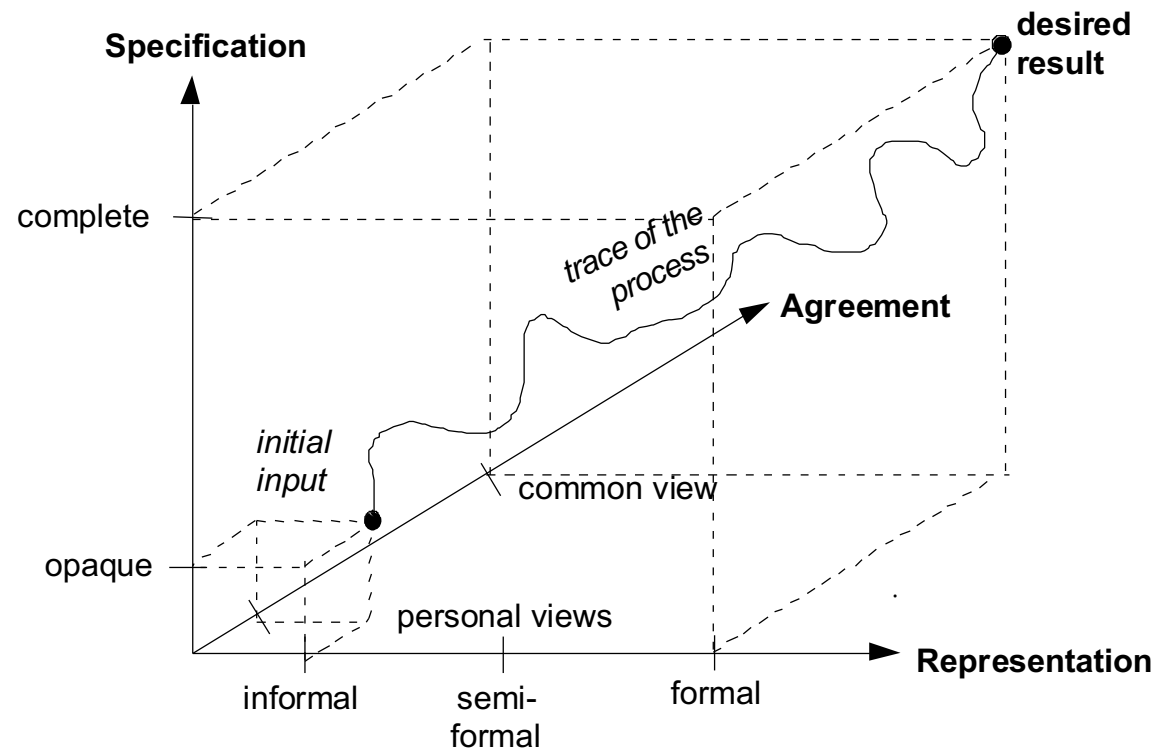
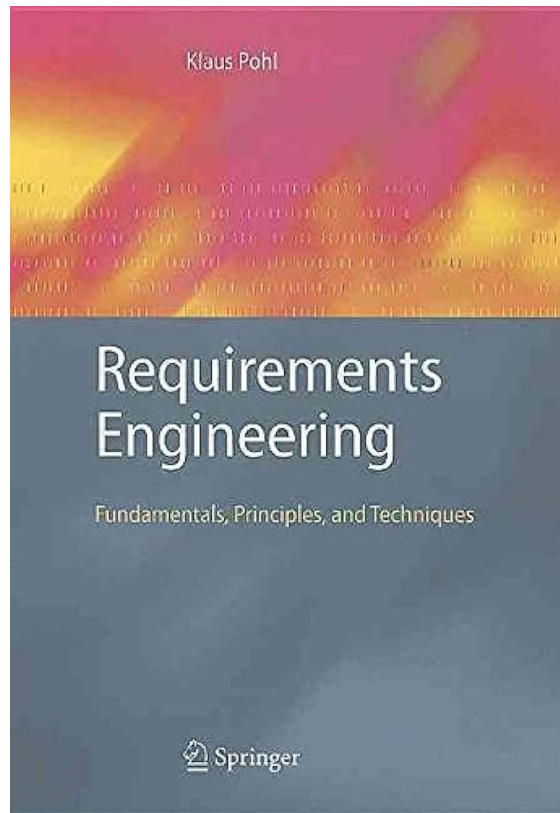
Figure 7. Step 3: Attempt to do the job twice – the first result provides an early simulation of the final product.

A blurred office scene showing a person sitting at a desk with a laptop. The person is wearing a dark blue shirt and dark pants. The desk is white, and there is a blue desk lamp. The background is a bright, out-of-focus office environment.

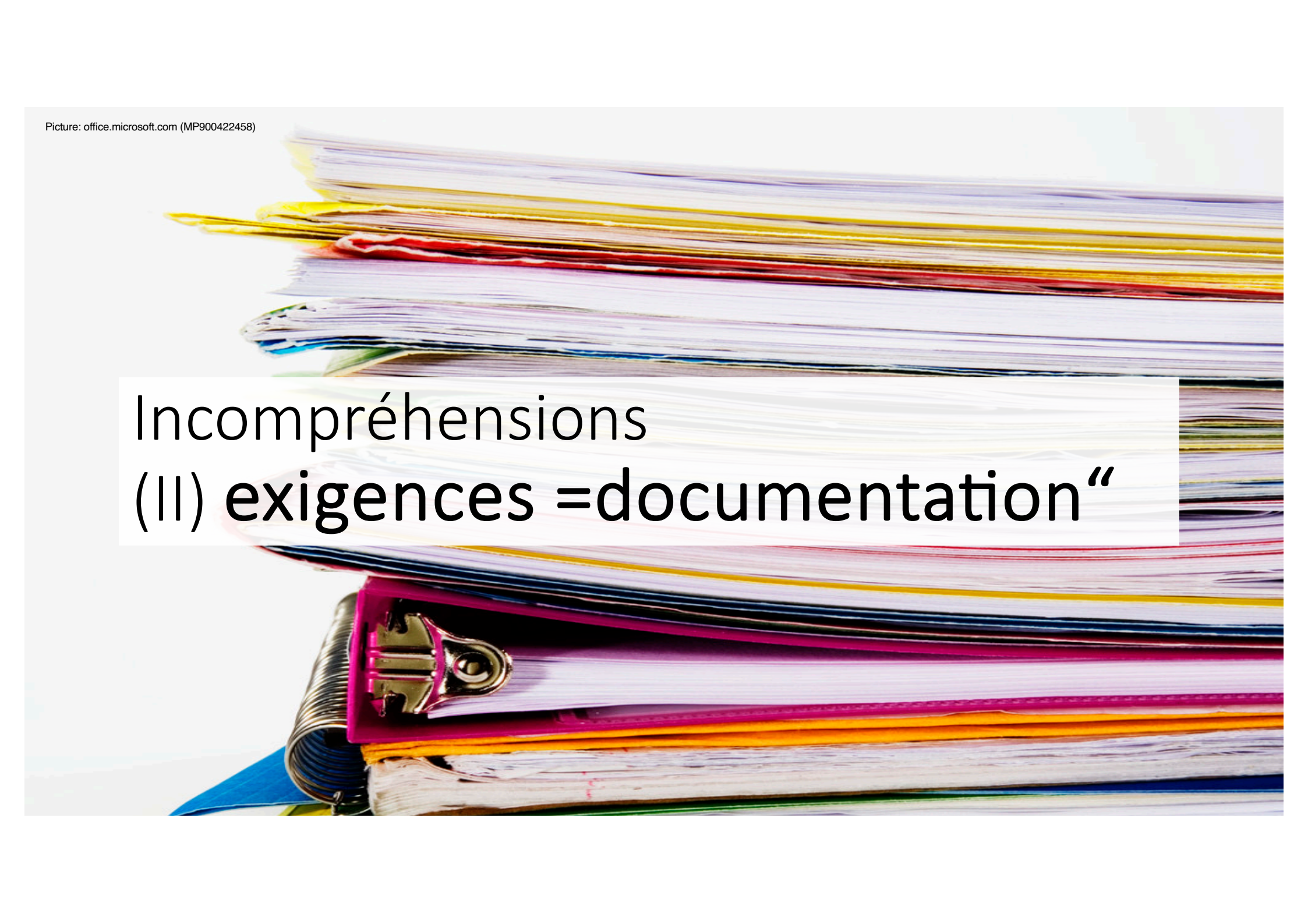
Incompréhensions

(I) exigences = „phase amont“

# 3 dimensions (Klaus Pohl)







Incompréhensions  
(II) exigences =documentation“

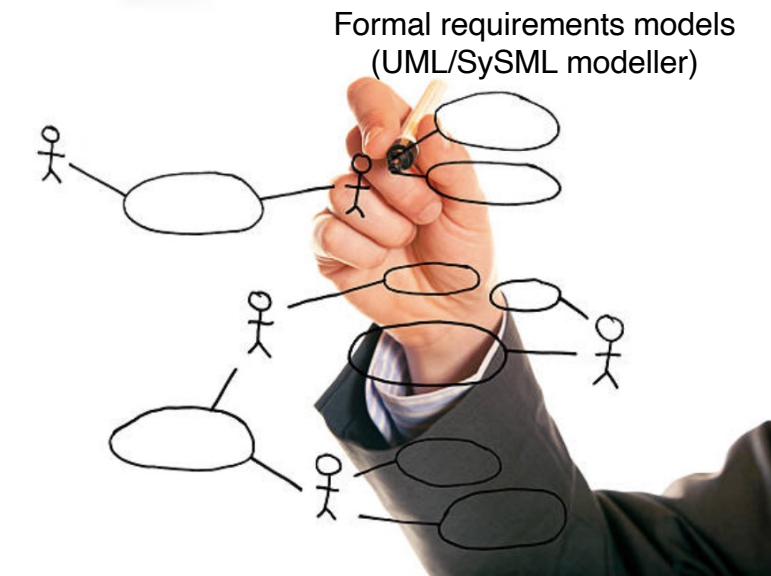
# Différents modèles d'échanges




Oral communication



Written requirements  
(Excel, Word)



Formal requirements models  
(UML/SySML modeller)

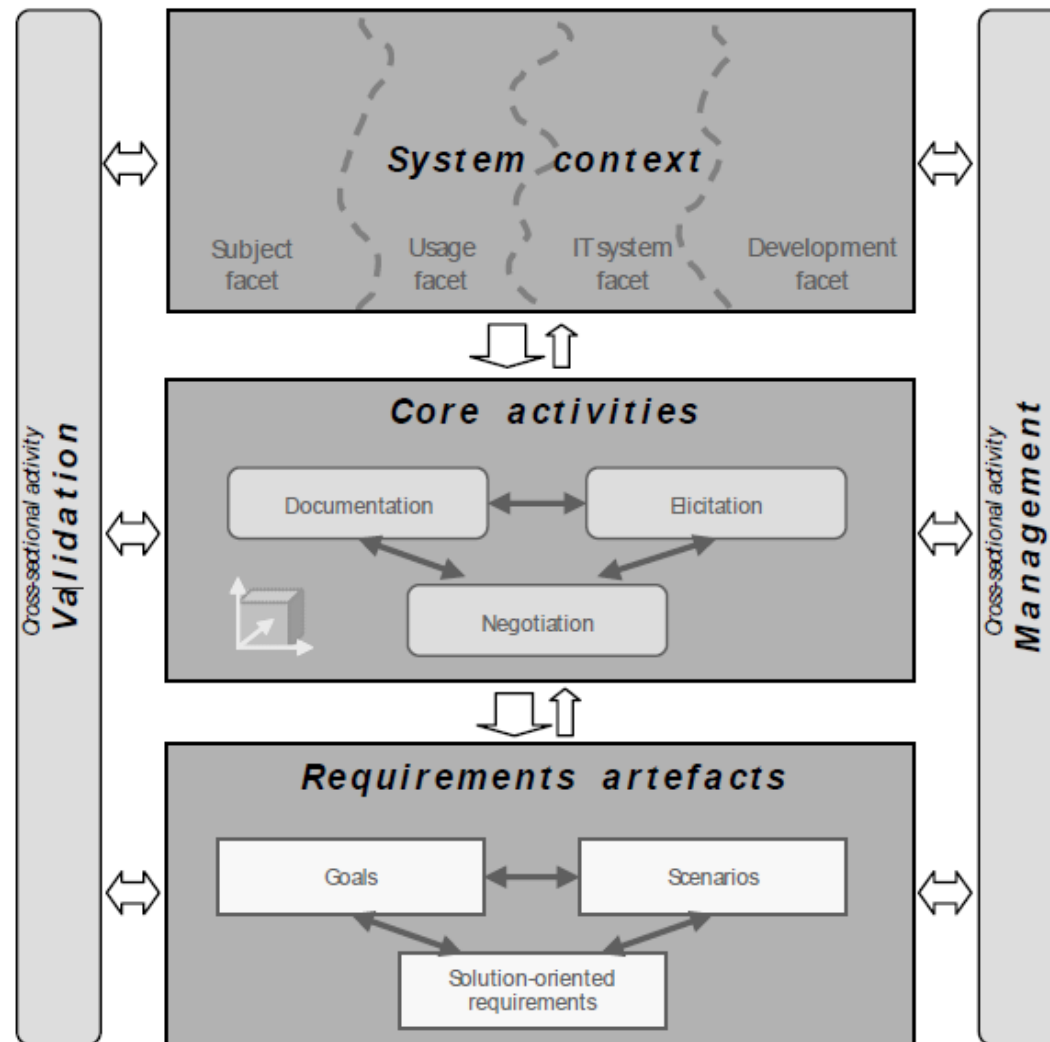



Incompréhensions  
(III) L'exécution du logiciel  
comme unique stratégie de  
validation



# 4 activités principales de l'IE

[Pohl 2010]  
« Requirements Engineering »  
Fig.4.1



The image features three penguins of increasing size, likely King penguins, sitting on a white surface against a grey background. The penguins are arranged from left to right, with the smallest on the left, a medium-sized one in the center, and the largest on the right. A semi-transparent white rectangular box is overlaid on the middle of the image, containing text.

Incompréhensions  
(IV) D'abord le problème,  
ensuite la solution

# Twin peak model (Bashar Nuseibeh)

## 3 principes clef

- Analyse de contexte
- Focus sur les problèmes et les buts
- Bottom-up/top-down

### Weaving Together Requirements and Architectures

Bashar Nuseibeh, The Open University

Compelling economic arguments justify why an early understanding of stakeholders' requirements leads to systems that satisfy their expectations. Equally compelling arguments justify an early understanding and construction of a software-system architecture to provide a basis for discovering further requirements and constraints, evaluating a system's technical feasibility, and determining alternative design solutions. Software-development organizations often choose between alternative starting points—requirements or architectures. This invariably results in a waterfall development process that produces artificially frozen requirements documents for use in the next step in the development life cycle. Alternatively, this

requirements and architectural specifications concurrently, it continues to separate problem structure and specification from solution structure and specification, in an iterative process that produces progressively more detailed requirements and design specifications, as Figure 1 suggests. The Twin Peaks model addresses the three management concerns identified by Barry Boehm ("Requirements that Handle IKIWISI, COTS, and Rapid Change," *Computer*, July 2000, pp. 99-102).

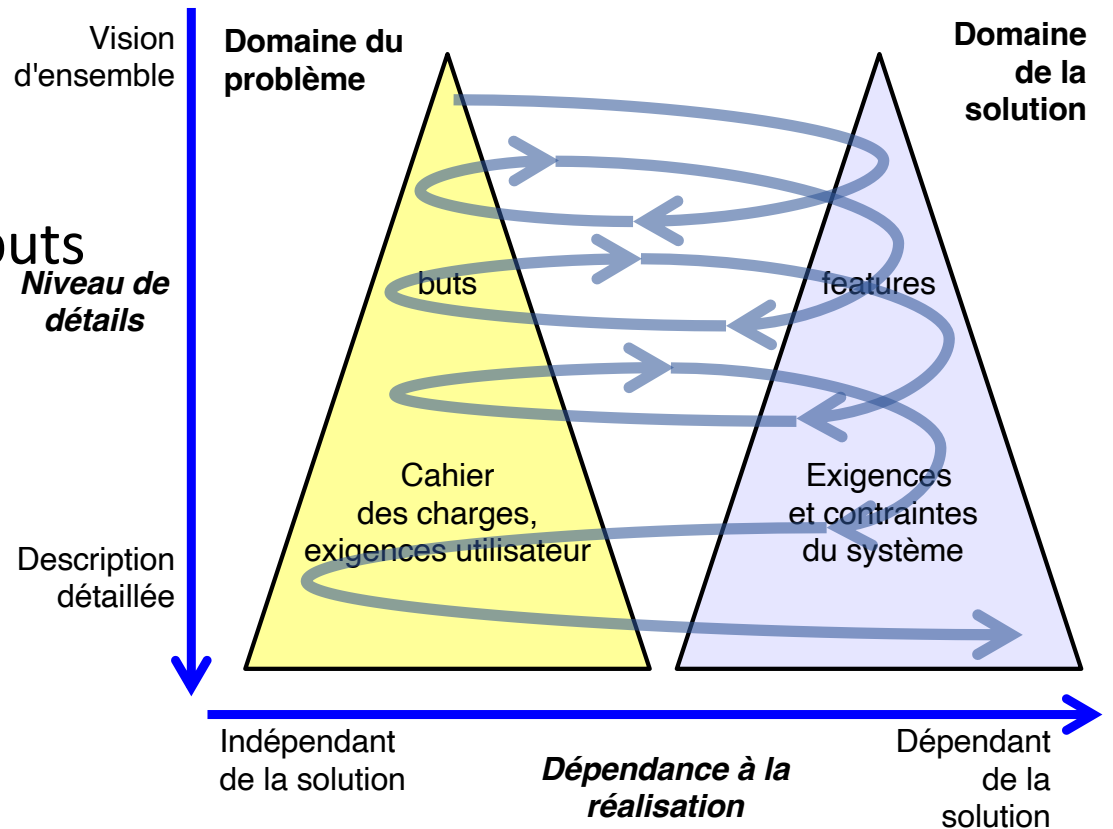
• *I'll Know It When I See It (IKIWISI)*. Requirements often emerge only after users have had an opportunity to view and provide feedback on models or prototypes. Twin Peaks explicitly allows the user to explore the solution space early, permitting incremental development



**Twin Peaks intertwines software requirements and architectures to achieve incremental development and speedy delivery.**

**THE TWIN PEAKS MODEL**  
Except for well-defined problem domains and strict contractual procedures, most software-development pro-

and consequent risk management.  
• *Commercial off-the-shelf software (COTS)*. Increasingly, software development is actually a process of







Et vous ?



# (re)découvrons l'IE

## Amateurisme

- Rigidité
- Effet tunnel
- Faible valeur
- Dette technique
- Non alignement
- ...



## IE professionnelle

- Communication
- Créativité
- Conciliation
- Spécification
- Agilité
- Raisonnement
- Décision
- Durabilité

# Les « 7 bonnes habitudes » de l'IE



Communication



Compréhension



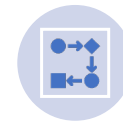
Intégration



Ingénierie



Validation



Itération



Communication



# **Manifesto for Agile Software Development**

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.



# Manifesto for Agile Software Development

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.

“Working software over comprehensive documentation”

- working  $\neq$  running
- [www.oxforddictionaries.com](http://www.oxforddictionaries.com)
  - Working := “able to function”
  - Function := “Fulfil the purpose or task of (a specified thing)”



# Determiner le degré de “Working”

“Standing”  
Concepts



“Running”  
Software

- Description en prose (tq. ‘shall-statements’, user stories)
- Diagrammes (pex. contexte, parties prenantes)
- Modèles (pex. UML)
- Maquettes UI statiques
- Scenarios
- Story boards
- Cas d’utilisation
- Maquettes interactives
- Prototypes implémentés
- Logiciel en production



A photograph of a sandy beach with footprints and a white text box. The text box contains the following text:

« La maison plutôt que le plan »  
« Le plat plutôt que la recette »  
« Le chemin plutôt que la carte »

L'ingénierie des Exigences  
est parfaitement en phase  
avec ces préceptes





# Exigences et agilité

- **1er Principe:**
- « Notre priorité première est de satisfaire le client en livrant au plus tôt et de manière constante un logiciel de qualité. »
- L'IE apporte une **compréhension des désirs et besoins** des parties prenantes qui est adaptée au développement de logiciels porteurs de valeur



## Exigences et agilité

- **2ème Principe:**
- « Tout changement, même tardif, des exigences pendant le développement est bienvenu. Les méthodes Agiles transforment le changement en avantage compétitif pour le client. »
- L'IE fournit les outils adaptés pour reconnaître les **changements de marchés** et pour défendre **l'avantage compétitif** des parties prenantes



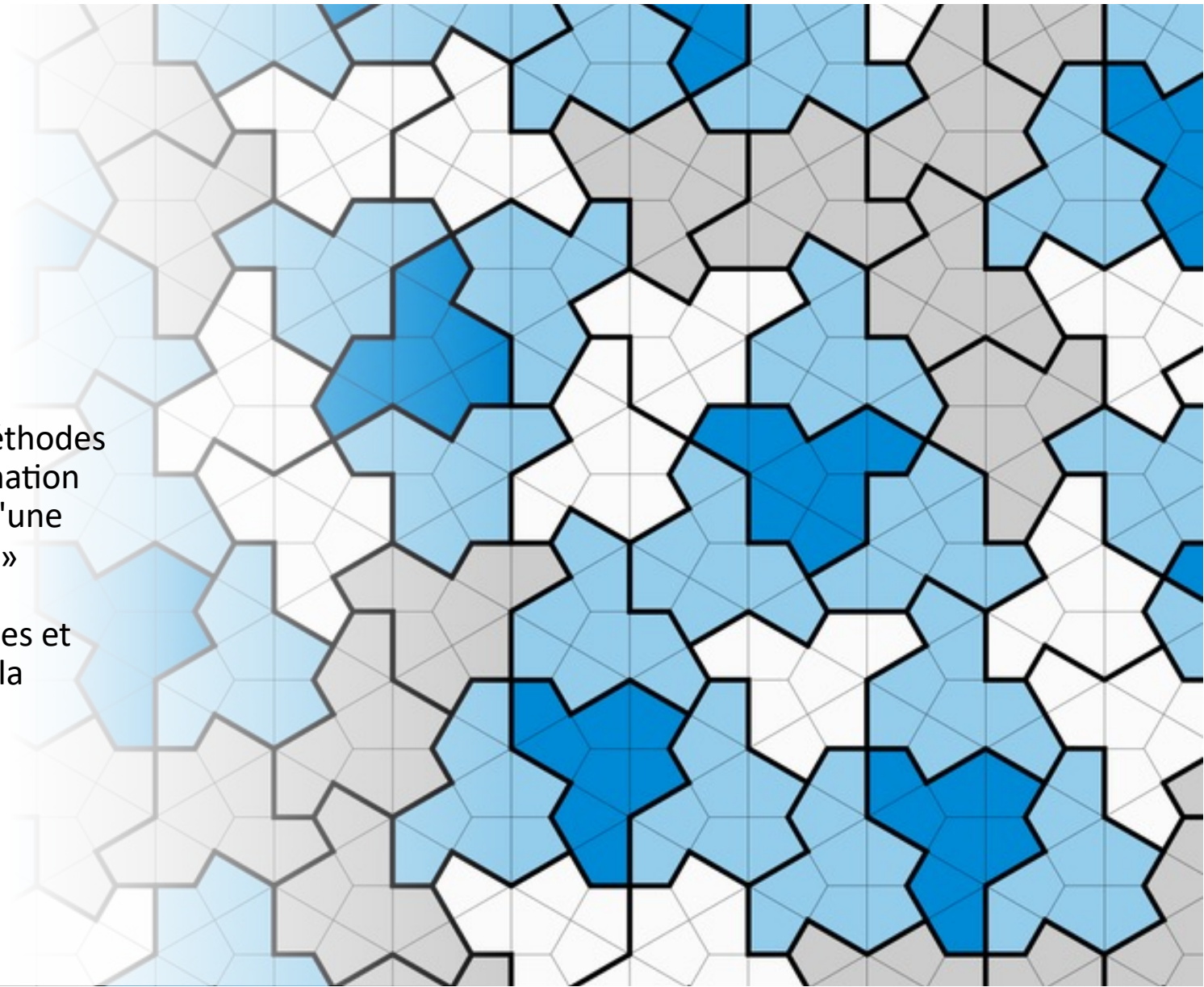


## Exigences et agilité

- **4ème Principe:**
- « Maitrise d'ouvrage et développeurs doivent collaborer quotidiennement tout au long du projet. »
- L'IE apporte les techniques et outils adaptés pour **soutenir une collaboration effective** entre les parties prenantes et les développeurs

# Exigences et agilité

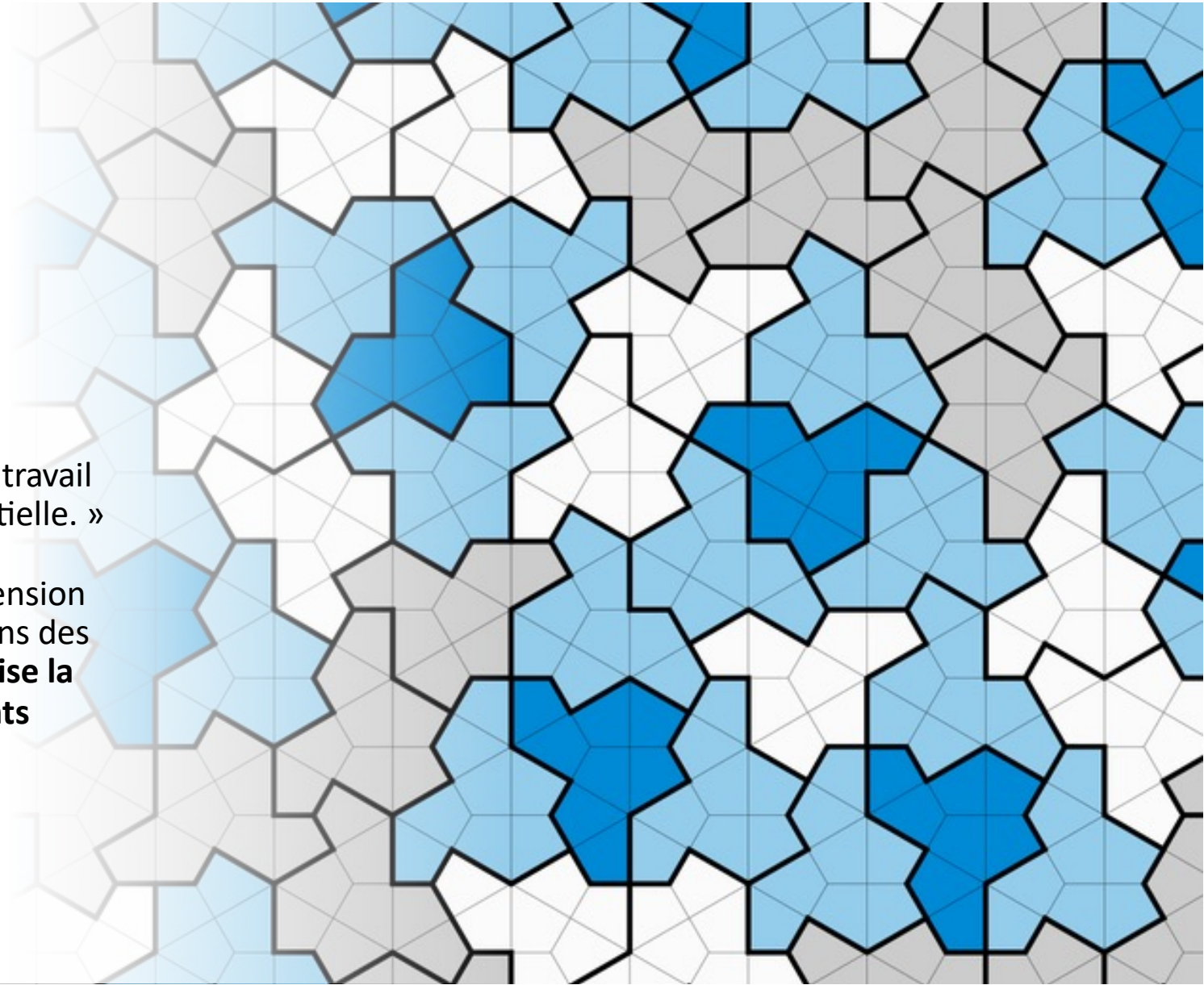
- **6ème Principe:**
- « La plus efficace des méthodes pour transmettre l'information au sein et à destination d'une équipe est le face à face. »
- L'IE apporte les techniques et outils adaptés pour aider à la **communication verbale**



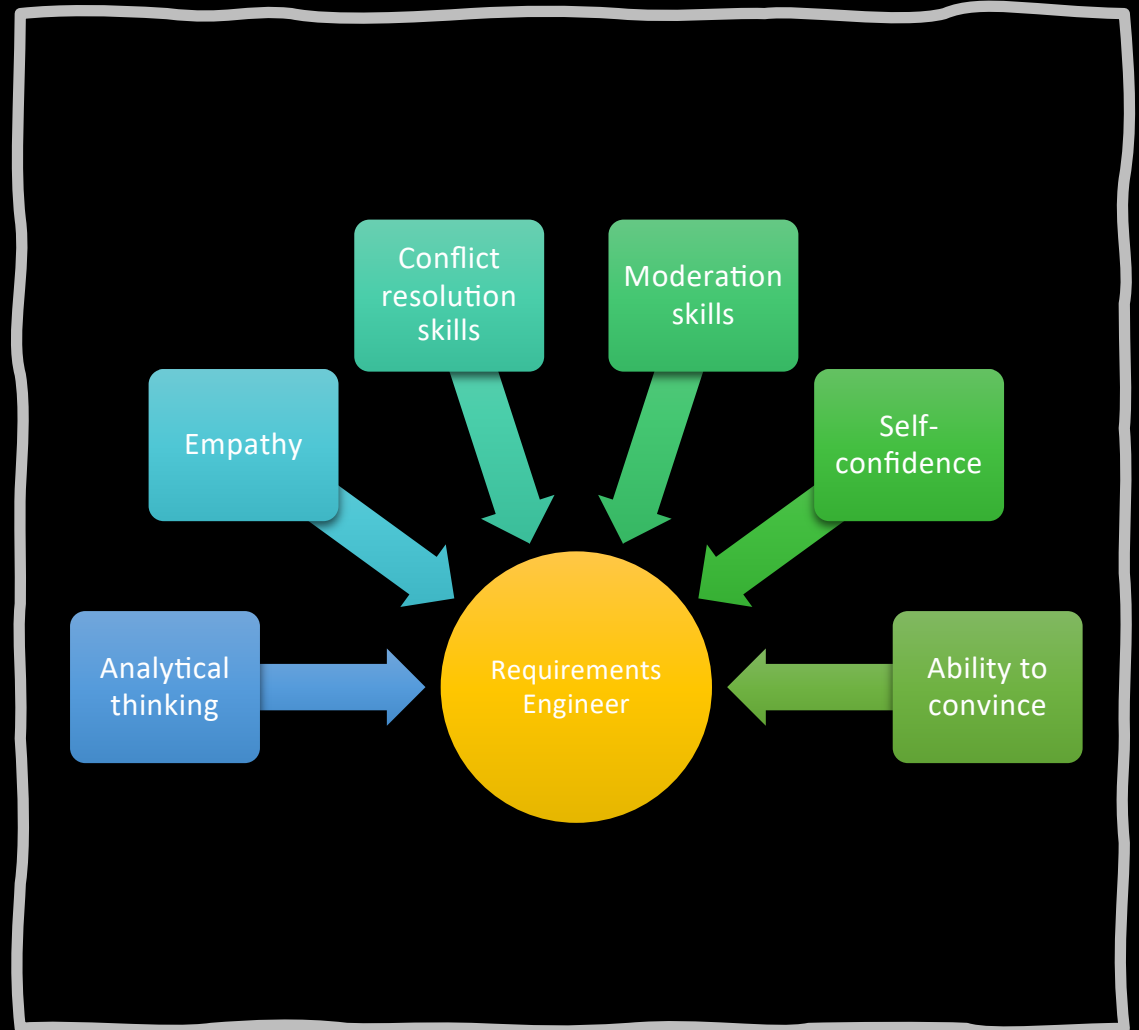


# Exigences et agilité

- **10ème Principe:**
- « La simplicité - l'art de maximiser la quantité de travail à ne pas faire – est essentielle. »
- L'IE apporte la compréhension adaptée des désirs et besoins des parties prenantes et **minimise la quantité de développements logiciels non nécessaires**



# Compétences



# Conclusion: Agile **et** compétent en IE

## **Partagent le même but de livrer des systèmes de qualité**

- Les approches Agile peuvent livrer des logiciels de manière efficace et rapide
- L'IE apporte des techniques pour comprendre les désirs et besoins des parties prenantes pour développer le bon logiciel.

**L'agilité** rend le développement **efficace**

**L'Ingénierie des Exigences** rend le développement **effectif**.



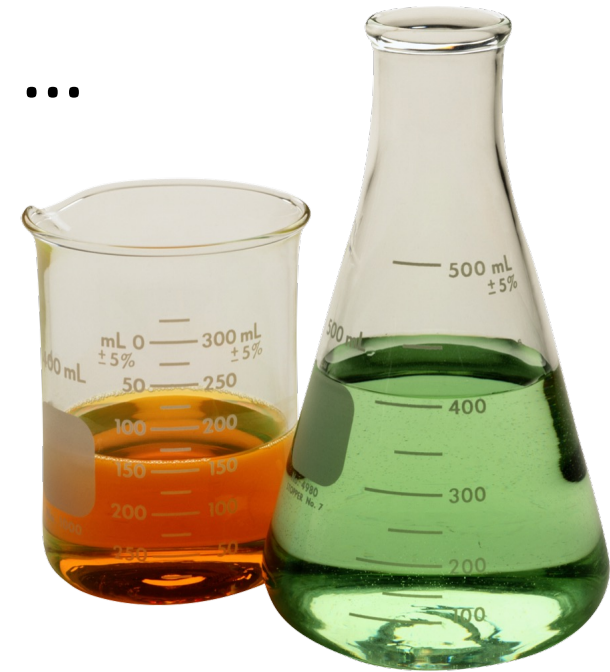
# Merci



Encore un dernier mot...



L'agilité, l'ingénierie des exigences,  
le design, la BA c'est bien, mais ...



... ce ne sont pas des sciences exactes !



# Champs disciplinaires

Mathématiques

Ingénierie

Informatique

Marketing

Management

Linguistique

Psychologie

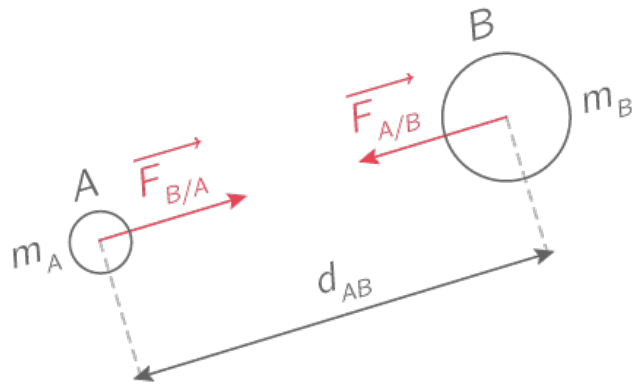
Arts

Philosophie

Sociologie

# Lois

## Théorie de la gravitation universelle « de Newton »



Newton's law (1687) of universal gravitation states that particles in the universe attracts each other with a force that is « *directly proportional to the product of their masses and inversely proportional to the square of the distance between them* ».

# Lois et modèles de références en ingénierie

## Disciplines traditionnelles

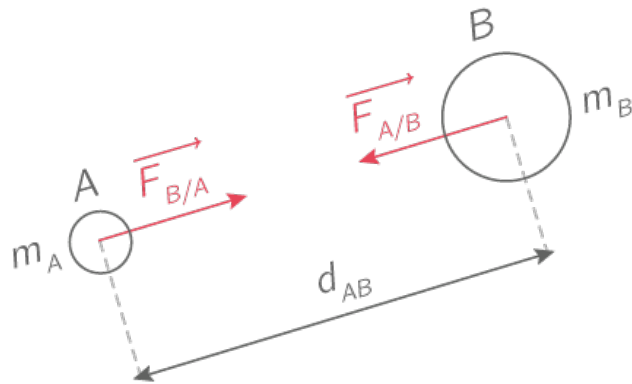
Guidées par des lois physiques qui peuvent directement être expérimentées en laboratoire

## En genie logiciel

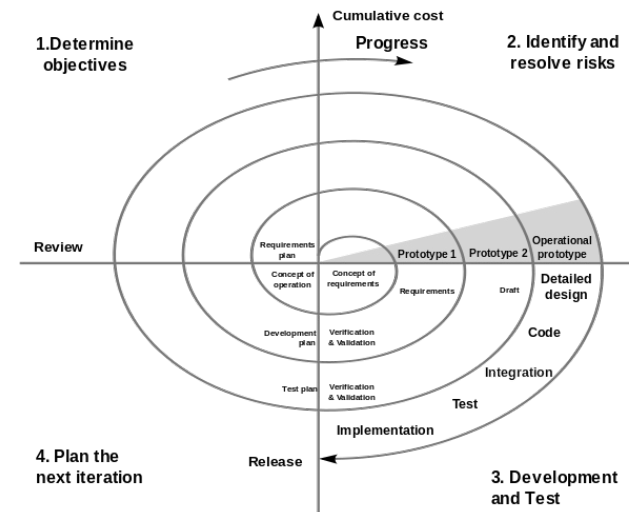
Peu de lois connues. En général les logiciels “fonctionnent”, et on n’apprend que lorsque certaines “lois cognitives” sont violées pendant leur implementation, maintenance ou utilisation

# Lois, VS modèles de référence

## Newton's theory of universal gravitation



## Boehm's SW development model



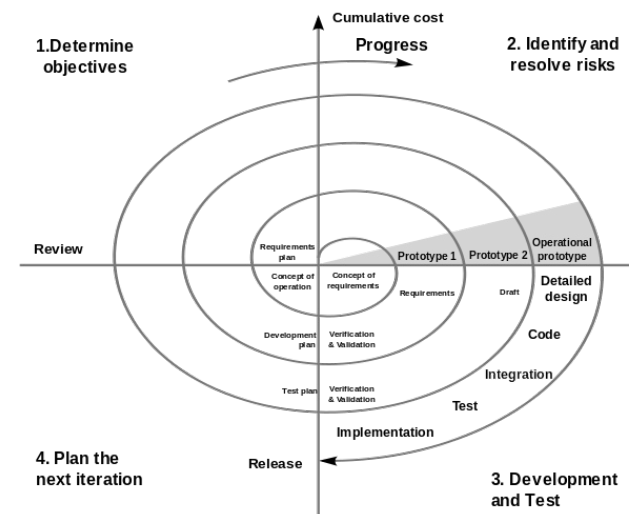


# Modèle de référence



The spiral model (Boehm, 2000) is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, *the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.*

## Boehm's SW development model



Et alors ?

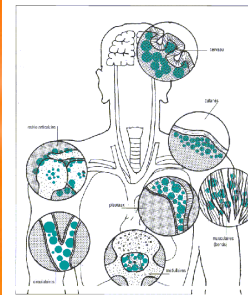


Empirical research in systems and software engineering.  
R. Mazo

# Alors, ce sont des sciences empiriques

**Tables MN90 - FFESSM (plongée à l'air)**

6m					12m					18m					25m					
Prof.	Durée	3m	DTR	GPS	Prof.	Durée	3m	DTR	GPS	Prof.	Durée	3m	DTR	GPS	Prof.	Durée	3m	DTR	GPS	
	15 min		1	A		1h20		1	H		35 min		2	F		5 min			2	B
	30 min		1	B		1h25		1	I		40 min		2	G		10 min			2	C
	45 min		1	C		1h30		1	I		45 min		2	H		15 min			2	D
	1h15		1	D		1h35		1	J		50 min		2	H		20 min			2	E
	1h45		1	E		1h40		1	J		55 min	1	3	I		25 min	1	3	F	
	2h15		1	F		1h45		1	J		60 min	5	7	J		30 min	2	4	H	
	3h00		1	G		1h50		1	K		1h05	8	10	J		35 min	5	7	I	
	4h00		1	H		1h55		1	K		1h10	11	13	K		40 min	10	12	J	
	5h15		1	I		2h00		1	K		1h15	14	16	K		45 min	16	18	J	
	6h00		1	J		2h10		1	L		1h20	17	19	L		50 min	21	23	K	
						2h15		1	L		1h25	21	23	L		55 min	27	29	L	
	15 min		1	B		2h20	2	4	L		1h30	23	25	M		60 min	32	34	L	
	30 min		1	C		2h30	4	6	M		1h35	26	28	M		1h05	37	39	M	
	45 min		1	D		2h40	6	8	M		1h40	28	30	M		1h10	41	45	M	
	60 min		1	E		2h50	7	9	N		1h45	31	33	N		1h15	44	50	N	
	1h30		1	F		3h00	9	11	N		1h50	34	36	N		1h20	47	55	N	
	1h45		1	G		3h10	11	13	N		1h55	36	38	N		1h25	49	58	O	
	2h15		1	H		3h20	13	15	O		2h00	38	40	O		1h30	51	64	O	
	2h45		1	I		3h30	14	16	O											
	3h15		1	J		3h40	15	17	O											
	4h15		1	K		3h50	16	18	O											
	5h00		1	L		4h00	17	19	O											





*Par construction, le chemin est pénible*





Ce qui est vraiment  
intéressant dans les REX  
=> les irritants,  
problèmes, echecs,  
difficultés, obstacles,  
goulets, strategies de  
contournement, etc

A marble statue of a bearded man, likely a philosopher, in a thinking pose. He is seated, with his right hand resting on his chin and his left hand resting on his lap. The statue is set against a solid blue background.

Manipuler l'inférence avec extrême prudence !

Merci

