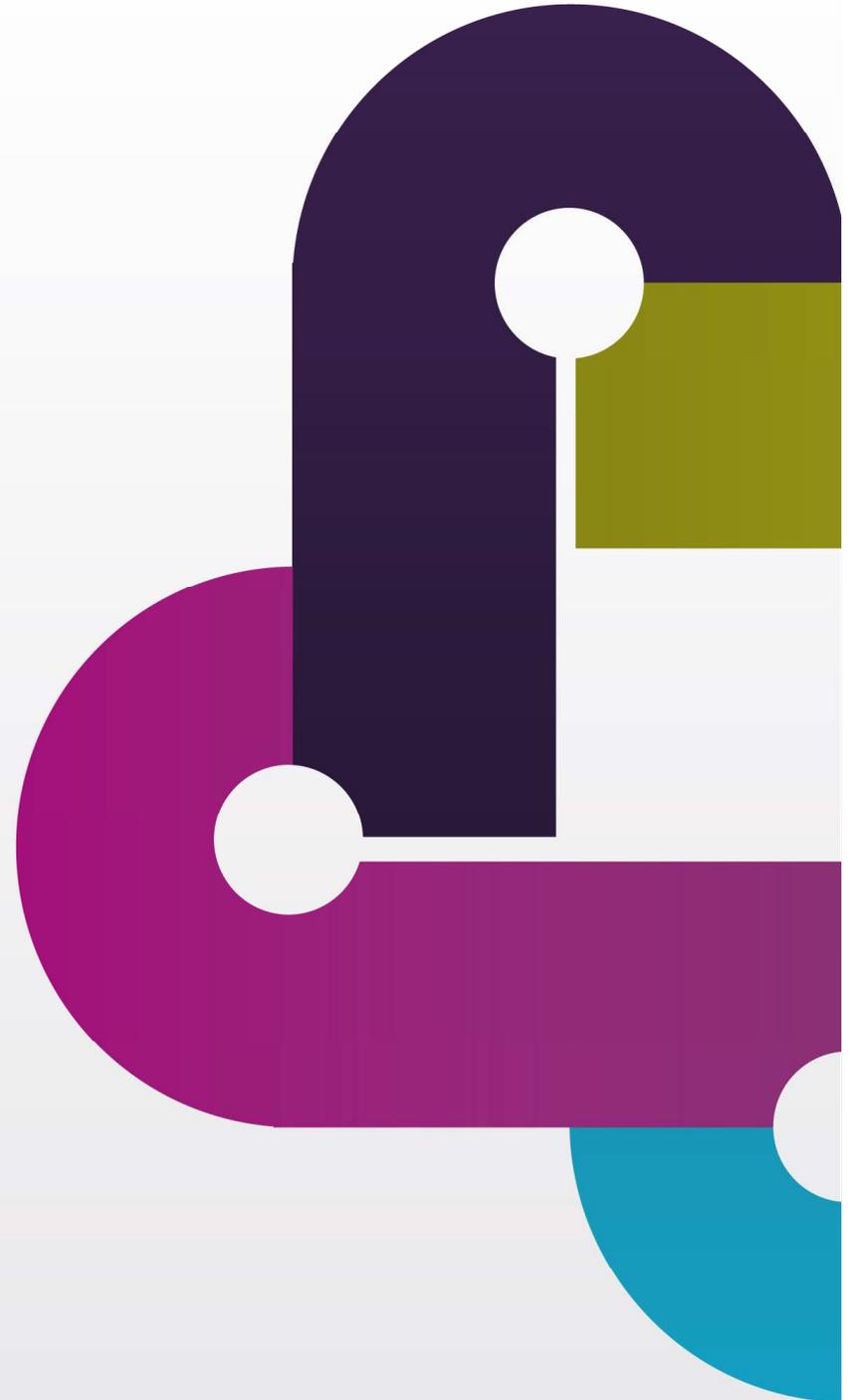


IBM Rational Day

Jeudi 25 octobre 2012 - 8h30 à 17h00,
Pullman Bercy

Ingénierie système guidée par la modélisation

Retours d'expériences
dans la défense et l'industrie



Observations



Observation

- Les entreprises qui développent des « systèmes produits » basent leur travail sur
 - ▶ Des documents *et/ou*
 - ▶ Des modèles
 - Descriptifs *et/ou*
 - Exécutables
 - Non testés *et/ou*
 - Testés

Pourquoi?



Observation

- Certaines entreprises qui développent des « systèmes produits » font de la traçabilité et d'autres non

Pourquoi?



Observation

- Certaines entreprises qui développent des « systèmes produits » intègrent l'ingénierie système et d'autres non

Pourquoi?



Observation

- Certaines entreprises qui développent des « systèmes produits » considèrent l'informatique comme un mal nécessaire et d'autres non

Pourquoi?



Observations

- Certaines entreprises qui développent des « systèmes produits » peuvent évoluer très rapidement (1 an) et d'autres très lentement (20 ans)

Pourquoi?



La formalisation des informations descriptive du système produit

Une évolution progressive



Chronologie

- D'abord avec des idées dans la tête des gens
- Ensuite avec des textes dans des documents
 - ▶ Facilite le travaille en équipe
 - ▶ Nécessaire pour
 - Les contrats commerciaux
 - La maintenance
 - L'historisation...
 - ▶ Exemples
 - « Cahier des Charges »
 - « Spécification »
 - Comment ça marche?
 - Comment faudrait-il que ça marche?



Chronologie

- Ensuite avec des diagrammes et des modèles dans des outils
 - ▶ Quand c'est devenu « trop compliqué » de décrire les choses sous forme textuelle
 - Ex: Cas d'utilisation, Scénario de UC, Vortex, Loi de commande, Electronique, déformation physique
 - ▶ Exemple d'outil de modélisation (diagrammes)
 - Visio
 - ▶ Exemple de simulateur (« Exécuteur de modèle exécutable »)
 - Matlab & Simulink, PSPICE, Catia, Adams, ProEngineer, Rhapsody



Cap: Idées => Document

- Difficile...
- Difficulté de « rester important »
 - ▶ Sentiment d'inutilité potentielle
 - ▶ => Rétention d'information



Cap: Document => Modèle

- Difficile...
- Difficulté de structuration des données
 - ▶ Document
 - Word => Texte linéaire (facile)
 - Excel => Texte cellulaire (un peu moins facile)
 - DOORS => Texte cellulaire organisé (plus compliqué)
 - ▶ Modèle
 - UML & C° => Items reliés (« impossible »)



Cap: Modèle descriptif => Modèle exécutable

- Observation: Opposition:
 - ▶ Certaines entreprises ne feraient plus autrement
 - ▶ D'autres entreprises n'aiment pas, ne le croient pas possible, trouvent cela trop compliqué et inabordable



Cap: Modèle descriptif => Modèle exécutable

- Difficile...
- Question de culture
 - ▶ Evident pour un ingénieur logiciel
 - ▶ Inaccessible pour un littéraire
- Tous les personnes qui ont travaillé en boucle ouverte ont affronté cette difficulté
 - ▶ Pas moyen d'évaluer la qualité de leur travail
 - ▶ Quand cela devient possible, c'est difficile d'accepter cette évaluation objective!
 - ▶ ...Et le surcroît de travail qu'il implique (doubler!)



L'évolutivité des informations

- Les documents papier sont « mort »: leur modification est impossible
 - ▶ => Version des documents
 - Complicé, lourd (ex: circuit d'approbation... pour presque rien)
- => Les rendre « vivant »
 - ▶ => Informatisation (volatilité infinie)
 - Cela n'empêche pas de devoir gérer les version...
 - Nécessité de « sauvegarde »
 - Ex: DOORS
 - Ex: Tous les outils de modélisation qui gèrent les versions!



La disponibilité des informations

- Les documents papier sont uniques
 - ▶ => Un seul lecteur à la fois.
- => Les mettre à disposition
 - ▶ => Informatisation (ubiquité infinie)
 - => Fichiers
 - => Duplication facile => Perte du contrôle d'accès
 - => Via un outil
 - => Pas de duplication
 - => Contrôle d'accès possible
 - => Information toujours à jour
 - => Ex: DOORS
 - => Ex: Tous outils de modélisation qui gèrent l'accès aux informations!



Modèle?



Exemples

- Un modèle mathématique d'un tourbillon
 - ▶ C'est une représentation d'un tourbillon sous forme d'équations mathématiques
- Un modèle d'une loi de commande
 - ▶ C'est une représentation d'une loi de commande sous formes d'un réseau de fonctions de transfert



Le modèle d'un bidule, c'est...

- Un ensemble d'éléments de nature connue qui représentent (totalement ou partiellement) ce bidule

- NB: Larousse: Chose dont on ne sait pas exactement le nom ou qu'on ne tient pas à préciser davantage.



Le modèle d'un moteur DC, c'est...



Sa description 3D

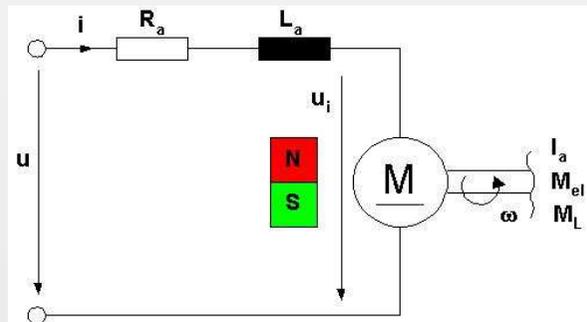
$$u = iR_a + L_a \frac{di}{dt} + u_i$$

$$u_i = c\Phi \omega$$

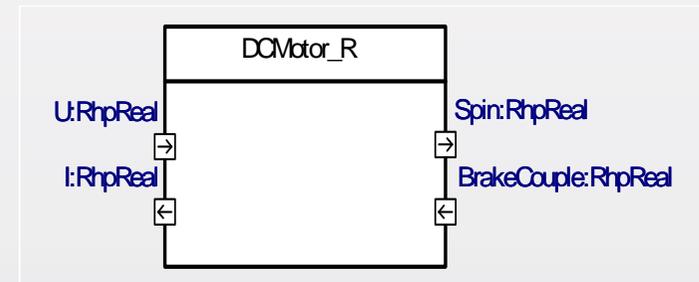
$$I_a \frac{d\omega}{dt} = \sum_n M_n = M_{el} - M_L$$

$$M_{el} = c\Phi i$$

Ses équations mathématiques



Son schéma électromécanique



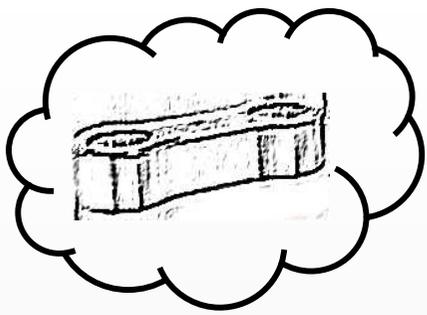
Son modèle UML



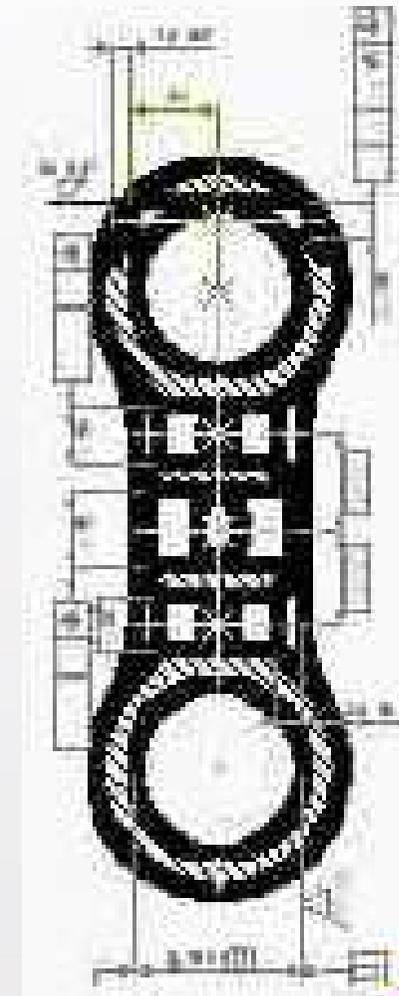
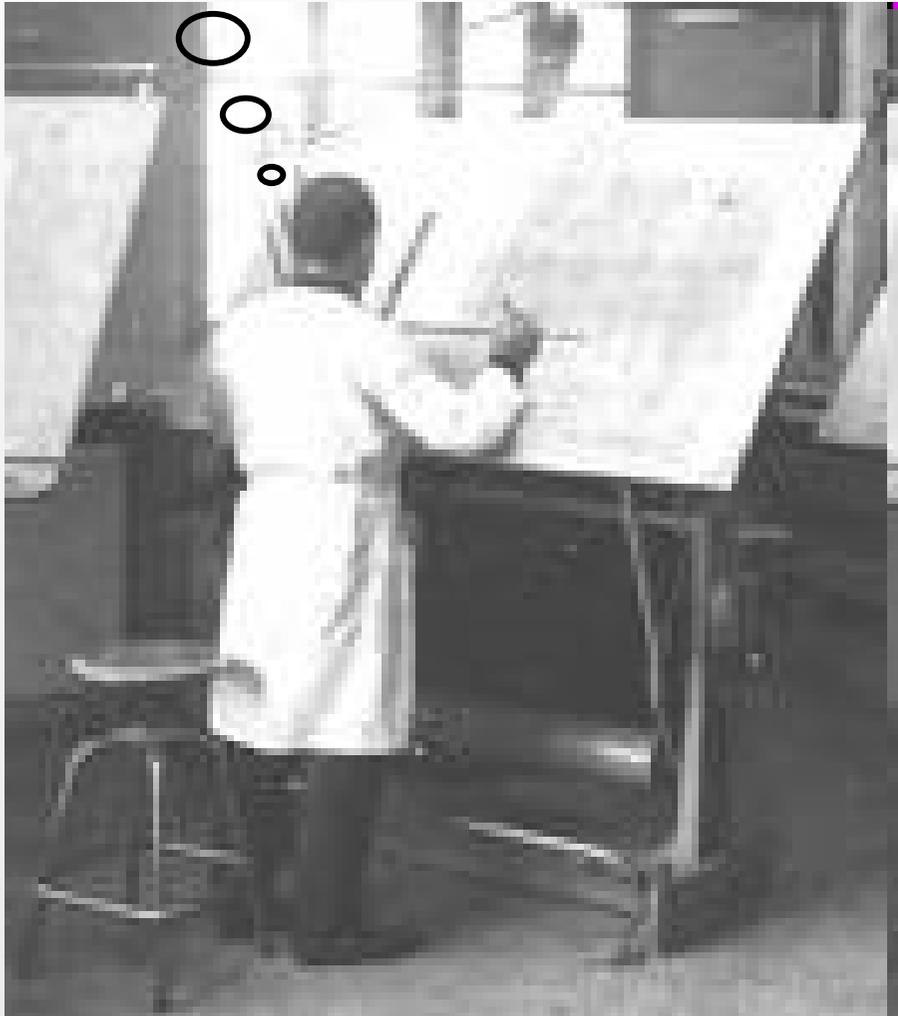
Evolution de la modélisation

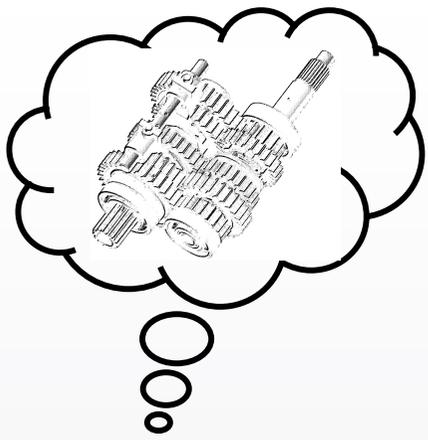
Exemple: la mécanique



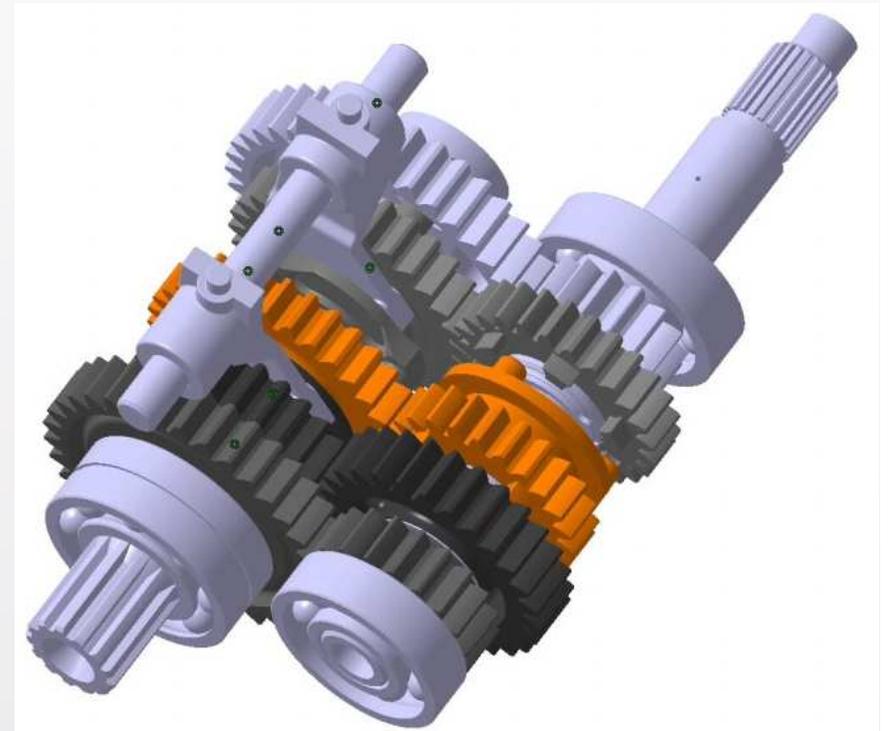
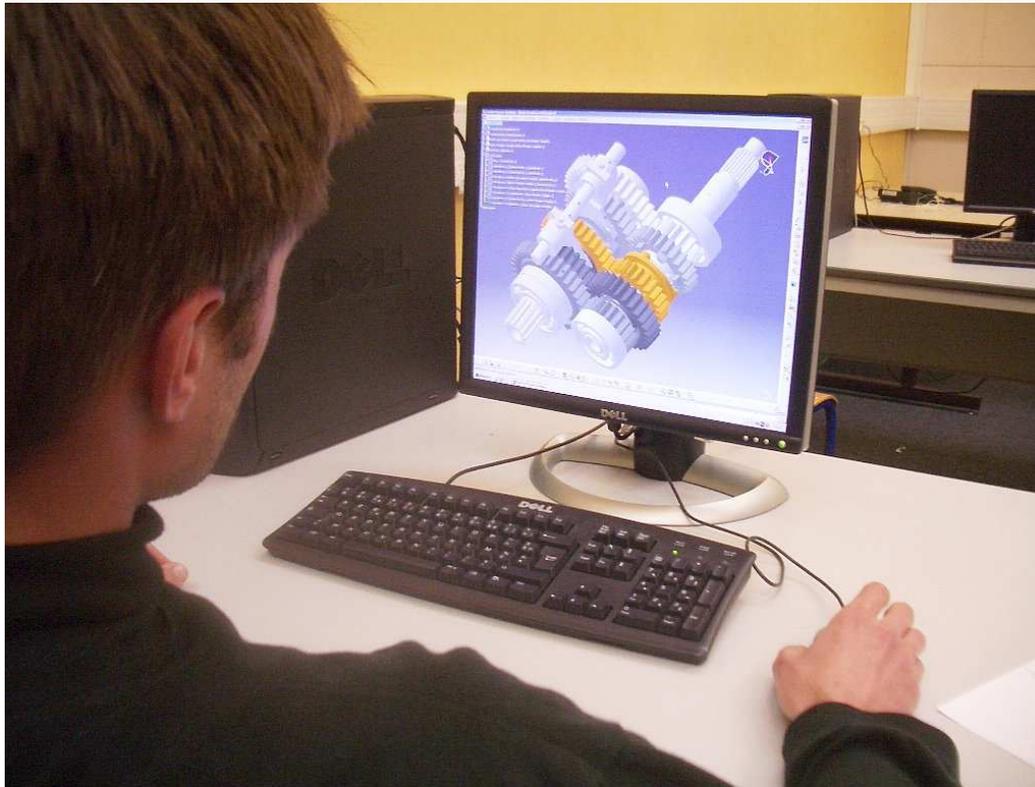


Il y a longtemps... (mais pas tant...)





Depuis quelques temps... (déjà...)



Ce changement...

- A permis de résoudre des nombreux problèmes de délai & de qualité
 - ▶ Plus grande précision
 - ▶ Plus de pièces plus complexes
- A pris du temps
 - ▶ 20..30 ans



Ce changement...

- A nécessité
 - ▶ De penser autrement
 - Du Rotring à la souris
 - De la planche à dessin à l'écran de l'ordinateur
 - ▶ Des investissements
 - Des ordinateurs, des bases de données, des réseaux
 - ▶ De la formation puis de l'accompagnement
 - ▶ De changer les méthodes et outils de travail



Le ROI est là!

- Pas en €
- En capacité à savoir faire de la mécanique beaucoup plus
 - ▶ Précise
 - ▶ Complexe
 - ▶ Performante
 - ▶ ... Dans le même délai!
- => Décorrélacion du cout et de la complexité



Validation du système p/r au besoin



Valider que le système fait ce qu'il devrait faire

- Garantie complète
- Mais:
 - ▶ Coûte cher
 - ▶ Possible que très tard dans le développement
 - ▶ Nécessite un prototype
 - ▶ Tout valider n'est pas possible



Valider que le système devrait faire ce qu'il devrait faire

- Garantie = toutes les exigences ont été prises en compte
 - ▶ => Traçabilité
- Ne garantit pas que le système fait ce qu'il devrait faire
- Mais:
 - ▶ C'est « simple »



Valider que le modèle du système fait ce qu'il devrait faire

- Garantie sur le modèle du système, pas sur le système
- Mais
 - ▶ Coûte pas cher
 - ▶ Possible dès le début du développement
 - ▶ Nécessite le modèle du système
 - ▶ Tout valider est possible



Et l'ingénierie système?



L'enjeu

- Comment faire travailler ensemble des experts techniques qui ne communiquent pas pour réaliser un « système produit »?
- Entrée =
 - ▶ Le besoin du client / les règlements & lois / le besoin du fournisseur
- Sortie =
 - ▶ Un cahier des charges par technique
 - i.e. informatique, mécanique, électrique, électronique...



Une longue histoire...

- L'ingénierie système existe depuis plus d'un siècle
- Elle est née avec la genèse de grands systèmes
 - ▶ Ex: Bâtiments terrestres et marins
 - ▶ Ex: Le transport par rail et par avion
 - ▶ Ex: Apollo puis la navette spatiale puis la station spatiale internationale
- L'arrivée de l'informatique a bouleversé l'ingénierie système
- La nouvelle ingénierie système n'est encore stabilisée



Dernière évolutions

- Tendance
 - ▶ Passage de « Travail multidisciplinaire » à « Nouvelle discipline » à part entière
- Admettre puis accepter que les experts de techniques différentes ne parviendront « jamais » à communiquer parce qu'ils ne peuvent pas se comprendre



Sans ingénierie système

- Les « donneurs d'information » au développement logiciel sont (trop) à l'aise
- Les ingénieurs des autres techniques sont « discrets »
- Les informaticiens sont:
 - ▶ Débordés
 - ▶ Stressés
 - ▶ Mis sous pressions
 - ▶ Non reconnus voire dénigrés
 - « C'est toujours de notre faute »



Pour les managers qui ne la connaissent pas...

- L'ingénierie système fait peur:
 - ▶ Une étape supplémentaire dans le processus
 - ▶ Esotérique / incompréhensible
 - « Déjà qu'on comprenait rien en informatique! »
 - ▶ Encore plus d'interlocuteurs
 - ▶ Encore plus de livrables
 - ▶ Encore plus cher



Pour les informaticiens qui ne la connaissent pas...

- L'ingénierie système fait peur:
 - ▶ Consciemment
 - « Elle va nous prendre notre travail »
 - « Qu'allons-nous encore recevoir comme spec? »
 - « Déjà qu'aujourd'hui, c'est compliqué mais alors avec l'IS! »
 - ▶ Inconsciemment
 - Sentiment de perdre le contrôle de son travail
 - Sentiment de cadrage (trop) fort
 - Sentiment de perte de liberté



La nouvelle ingénierie système

Mon avis



Code de la route

- Passer des documents au modèle
 - ▶ Et générer automatiquement les documents
 - ▶ Mais...
- Structurer les informations avec les concepts adéquats
 - ▶ Exemples
 - Notion de système
 - Héritage / Décomposition
 - Boite fermée / Boite ouverte (Black/White box)
 - Changement de paradigme
 - Interfaces: Command & Feedback
 - Comportement
 - Dissociation événement / réaction
 - Asynchronisation des réactions
 - « Code » / Machine à état / Activité
 - Continu & Discret



Code de la route

- Sortir du mélange conceptuel / technique
- Concevoir = Élaborer quelque chose dans son esprit
 - ▶ => 2 étapes
 1. Elaboration conceptuelle: Que faut-il faire?
 2. Elaboration technique: Comment faut-il faire ce qu'il faut faire?
- => Faire évoluer le processus de développement



Qu'espérer de cette nouvelle ingénierie système?

Retour du terrain... Pas « hypothèse »!



Pour les ingénieurs systèmes...

- L'ingénierie système demande « plus » de travail qu'avant mais...
- L'ingénieur système est sûr de lui et de son travail, il est plus humble
- L'ingénieur système est rassuré:
 - ▶ « Je ne suis pas responsable de ce qui manque dans la spec que j'ai reçue... » parce que « je lui ai dit ce que j'ai vu qui manquait »
 - ▶ « Je me concentre sur mon métier: concevoir des solutions et choisir les techniques pour les réaliser mais plus de technique »



Pour les informaticiens...

- L'ingénierie système rassure:
 - ▶ « Enfin de spec claires, précises et complètes »
 - ▶ « Je ne suis pas responsable de ce que l'ingénieur système a oublié... »
 - ▶ « Je me concentre sur mon vrai métier: l'intégration avec le matériel, le calculateur et toutes les optimisations possibles »
 - ▶ « Je ne dois plus me casser la tête à essayer de comprendre ce que je dois faire »



En réalité...

- L'ingénierie système réduit le coût du développement logiciel tel que:
 - ▶ $\text{Coût}_{\text{Développement Logiciel Avec IS}} + \text{Coût}_{\text{Ingénierie Système}} < \text{Coût}_{\text{Développement Logiciel Sans IS}}$
- A condition de se donner le bon code de la route!



L'informatique, un mal nécessaire?



Constat

- Il n'est plus possible de se passer de l'informatique
 - ▶ Il faudra l'accepter (même les mécaniciens ;-)
- Peur / non confiance des managers dans le développement logiciel
 - ▶ Le plus souvent avec raison...
- => Sécuriser le développement logiciel
- => Le développement informatique doit pouvoir s'engager et tenir ses engagements
 - ▶ Cout / Qualité / Délai
 - ▶ Par partie / global



Une solution

- Là où il n'y a pas d'ingénierie système formelle, le « Développement Logiciel » contient le travail de l'ingénieur système
 - ▶ => Perte de compétences des ingénieurs logiciels
 - Ex: un développeur Java ne connaît pas le concept de JVM!
- Mettre en place l'ingénierie système
 - ▶ => Conversion des ingénieurs logiciels en ingénieur système
- Mettre en place les règles « carrée » de traduction d'un modèle « système » en un modèle « logiciel »



Résistances aux changements

- Observation: Plus l'entreprise est grosse, plus la résistance au changement est importante
- Sentiment de perte de pouvoir
- Confusion
 - ▶ Sentiment de perte de son travail / Evolution de son travail
- « Chez nous c'est (beaucoup) plus compliqué »
 - ▶ Et pourtant 1..3 ans après, le changement est effectué
- Pour convaincre, il faut montrer sur un exemple du client
- Osez essayer!



Conclusion



Conclusion

- La (nouvelle) ingénierie système promet de
 - ▶ Réduire les délais / Augmenter la qualité
 - ▶ Maîtriser la complexité / Faire (beaucoup) plus compliqué plus simplement à iso-coût
 - ▶ Simplifier et acquérir la confiance en l'informatique
 - ▶ Satisfaire les normes
- La modélisation du système permet de
 - ▶ Formaliser toutes les informations descriptives du système
 - ▶ Tracer toutes celles qui doivent l'être
 - ▶ Mettre toutes les informations souhaitées à disposition de « tous »
 - ▶ Valider le modèle du système au plus tôt
- Osez essayer! Devenez un « unreasonable man »

