

Sommaire :

1. Introduction au logiciel

- Ordinateurs et systèmes logiciels
- Introduction au génie logiciel
- Comparaisons (ponts/bâtiments vs. systèmes logiciels)
- La "crise logicielle" et la demande croissante
- Réactivité des logiciels aux utilisateurs et aux environnements
- Capacité d'auto-maintenance des systèmes
- Un besoin d'IA dans le développement logiciel

2. Problèmes d'IA vs problèmes classiques du génie logiciel

- Définition d'un "problème d'IA"
- Spécifications mal définies
- Solutions "correctes" vs. "assez bonnes"
- L'importance du **comment**, pas seulement du **quoi**
- Problèmes de dynamique, modularité, dépendance au contexte

3. Méthodologie en génie logiciel

- La méthode SAV (Spécifier & Vérifier)
- Limites de la spécification complète
- Qu'est-ce qui est vérifiable ?
- La méthode SAT (Spécifier & Tester) : tests de fiabilité, forces et faiblesses
- Conditions nécessaires pour les tests
- Rôle du prototypage comme lien

4. Méthodologie incrémentale et exploratoire

- Limites des méthodes classiques pour les problèmes d'IA
- Le cycle RUDE (Rapid, User-driven, DEvelopment)
- Début du projet et logiciels "malléables"
- Intégration de l'IA dans un squelette conventionnel
- Approche étape par étape et fin de projet
- Question de la "bidouille" (hacking) et paradigmes classiques

5. Nouveaux paradigmes pour l'ingénierie des systèmes

- Programmation automatique
- Implémentation transformationnelle
- "Nouveaux paradigmes" selon Balzer, Cheatham & Green
- Approche opérationnelle de Kowalski
- Méthodologie POLITE

6. Vers une discipline de programmation exploratoire

- Ingénierie inverse (reverse engineering)

- Logiciels réutilisables
- Connaissance de conception et abstraction progressive
- Décompilation, modification contrôlée, croissance structurée 7. Apprentissage automatique : grandes promesses, nombreux défis
- Logiciels auto-adaptatifs
- Promesses (puissance) vs menaces (complexité)
- État de l'art du machine learning
- Exemples pratiques
- Programmation inductive multiversion

8. Systèmes experts : une success-story

- Les systèmes experts comme logiciels d'IA
- Ingénierie de ces systèmes
- Enseignements pour le génie logiciel

9. IA dans les logiciels pratiques

- Environnements de support
- Réduction de la complexité concrète
- Outils d'ingénierie à base d'IA
- Logiciels auto-réflexifs
- Risque de sur-ingénierie

10. Synthèse et perspective d'avenir

- Récapitulatif des apports
- Réflexion sur les tendances futures du domaine