

Les principes fondamentaux du génie logiciel s'appliquent-ils au logiciel libre?

Normand Séguin, UQAM

Renaud Loiseau-Dupuis, UQAM

Alain Abran, ETS

Robert Dupuis, UQAM

Laboratoire de recherche sur
gestion du logiciel

Plan de la présentation

- Survol de la recherche sur les principes du génie logiciel
- Application des principes au contexte du logiciel libre

Introduction

- Génie logiciel
 - Cycle de vie court des
 - Technologies
 - Techniques
 - Outils
 - A la recherche de ses fondations
 - Les principes, une base plus stable

Principes du GL : constat

- Recensé plus de 300 principes (1970-2005)
- Constat
 - Nombre élevé de principes
 - Faiblesse dans les méthodes d'identification
 - Critères d'identification inexistants
 - Peu d'études définissent le terme principe

Objectif et définition

- Parmi les 308 principes recensés, quels seraient LES principes du génie logiciel ?
- *Proposition fondamentale de la discipline formulée sous forme prescriptive (règle), à la source des actions, pouvant être vérifiée dans ses conséquences et par l'expérience*

Méthode

Phase 1

- Définitions
- Critères d'identification
- SWEBOK (concepts GL)
- ISO/IEC 12207 (activités)

39

Phase 2

- Catégoriser les propositions
- Critères d'ensemble
- ISO/IEC 12207

34 propositions

Phase 3

308 propositions

Exemples de principe

- Build with and for reuse
- Define software artifacts rigorously
- Design for maintenance
- Determine requirements now
- Don't try to retrofit quality
- Don't write your own test plans
- Establish a software process that provides flexibility

Principes et logiciel libre

- Est-ce que les principes candidats s'appliquent au contexte du logiciel libre ?
- Résultats d'une première analyse du projet Mozilla

Caractéristiques du projet

- Projet d'envergure et complexe
- Processus discipliné de développement
- Mise en place d'un contrôle à 3 volets
 - 1 – Gestion du code source
 - 2 – Gestion des versions livrables
 - 3 – Assurance qualité

Align incentives for developer and customer

- En GL : source -> management
- En LL : source -> développeur
 - Au mérite
 - Contribution au projet
 - Motivations implicites
 - Fierté, passion, l'essor du logiciel libre

Give product to customer early

- *En GL* : Approche par prototypage
(but : validation)
- *En LL* : Versions préliminaires
 - la communauté s'organise autour de

Implement a disciplined approach and improve it continuously

- Une approche disciplinée a été mise en place dans le cadre du projet Mozilla
 - Contrôle à 3 volets
 - Incluant les revues techniques à deux niveaux
- Pas un besoin explicite d'amélioration sur une base continue

Involve the customer

- En GL : impliquer le client aux différentes étapes de développement (validation)
- En LL : La communauté s'implique d'elle même dans le projet
 - Pouvoir sur la destinée du produit

Strive to have a peer, rather than a customer find a defect

- S'applique très bien au contexte du libre
- Baker : « *open source projects rely on peer review* »

Principes moins pertinents

- Exigences du logiciel
- Maintenance
- Gestion et estimation des coûts
- Mesures en provenance du processus
- Use better and fewer people

Conclusion

- Certains principes du génie logiciel s'appliquent au contexte du logiciel libre
- D'autres ne s'appliquent pas
- D'autres projets de logiciel libre devront être étudiés pour tirer des conclusions définitives.