cetic
Your Connection to ICT Research

Geek Anonymous

Sélectionner un composant Open Source … QualOSS

Jean-Christophe DEPREZ

Liège, le 21/04/2017

PLUS HAUT ET PLUS PROCHE
LE FONDS EUROPÉEN DE DÉVELOPPEMENT RÉGIONAL ET LA WALLONIE INVESTISSENT DANS VOTRE AVENIR
FEDER
UNION EUROPÉENNE
Wallonie
digital wallonia .be

Centre d'Excellence en **Technologies** de l'**Information** et de la **Communication**
www.cetic.be

Agenda

- Petites parenthèses de départ : Les licences
- Une petite rétrospective: QSOS, OpenBRR, … Ohloh et QualOSS (projet FP6 2006-2009)
- Les contextes d'acquisition d'un composant open source
- Et si d'autres aspects que la partie logicielle comptaient dans une acquisition?
  Sélectionner ~~un projet~~ une initiative open source … c'est quoi une initiative?
- Méthodologie QualOSS … Une vision d'évaluation orientée risque
- Méthode d'évaluation de base de QualOSS
- Quelques exemples: GCC, Asterisk

Petite parenthèse de départ

cfr Geek Anonymous passé pour plus de détail sur les Business modèles liés à l'open source

- → Utilisation Interne ?
  - → Pas de distribution donc pas de véritable problème de licence
  - → Attention un service logiciel accessible via le web = distribution
- → Intégration dans un logiciel qui sera distribué?
  - → Problème de licence
  - → Décision: Le logiciel sera-t-il lui aussi Open Source? Ou sera-t-il sous licence propriétaire?
    - → Si propriétaire, … composant sous licence « commercial friendly »: MIT, BSD, … ou parfois aussi OK: E(clipse)PL, A(pache)PL, LGPL
    - → Si open source: Comment le logiciel sera-t-il utilisé par vos clients/utilisateurs?
      - → Interne?
      - → Intégration dans d'autres solutions
  - → Décision: Compatibilité entre les divers licences de composants open source utilisés

Rétrospective

- QualOSS projet européen FP6 (2007-2009)
- Constat début 2007 des méthodes de sélection émergent
  - Les premiers essais: Open Source Maturity Model (Capgemini 2003 et Navica 2004)
  - Les plus abouties: OpenBRR (2005) et QSOS (2006)
- Mais Elles
  - Focalisent sur le périmètre d'un projet FlOSS
  - Font certaines suppositions quant au modèle open source à adopter sans le dire explicitement
  - Se reposent sur le point de vue de l'évaluateur et donc peuvent manquer d'objectivité
  - Supposent un effort de + ou - 2h pour évaluer un projet open source – donc valeur ajoutée limitée
- Et aussi Ohloh.org (Microsoft managers managers Jason Allen et Scott Collison en 2004)
  - Pas de méthode d'évaluation ou sélection … juste des données avec une belle visualisation
  - Certaines données brutes sont objectives mais d'autres fournies par qui veut !
  - Aussi focalisé au niveau du périmètre d'un projet FlOSS
  - 2009 - Racheté par Geeknet (owner of SourceForge)
  - 2010 – Cédé à BlackDuck → Openhub.net

Contexte d'acquisition … d'implication

Niveau d'implication d'un intégrateur mais aussi

Qu'attend-t-on d'autres contributeurs?

→FlOSS Exploit (modèle d'utilisation en interne)

→Utilisation mais pas de véritable contribution (au-delà de rapporter des bugs)

→C'est la plus part des cas mais … cela peut avoir de lourdes conséquences dans le moyen terme si on veut changer de contexte d'acquisition (par exemple aller de FlOSS Exploit à FlOSS Full Collaboration)

→FlOSS Fork

→Repartir d'un composant existant (encouragé par le modèle cathédral)

→FlOSS Take-over

→Reprise du leadership d'un projet Open Source

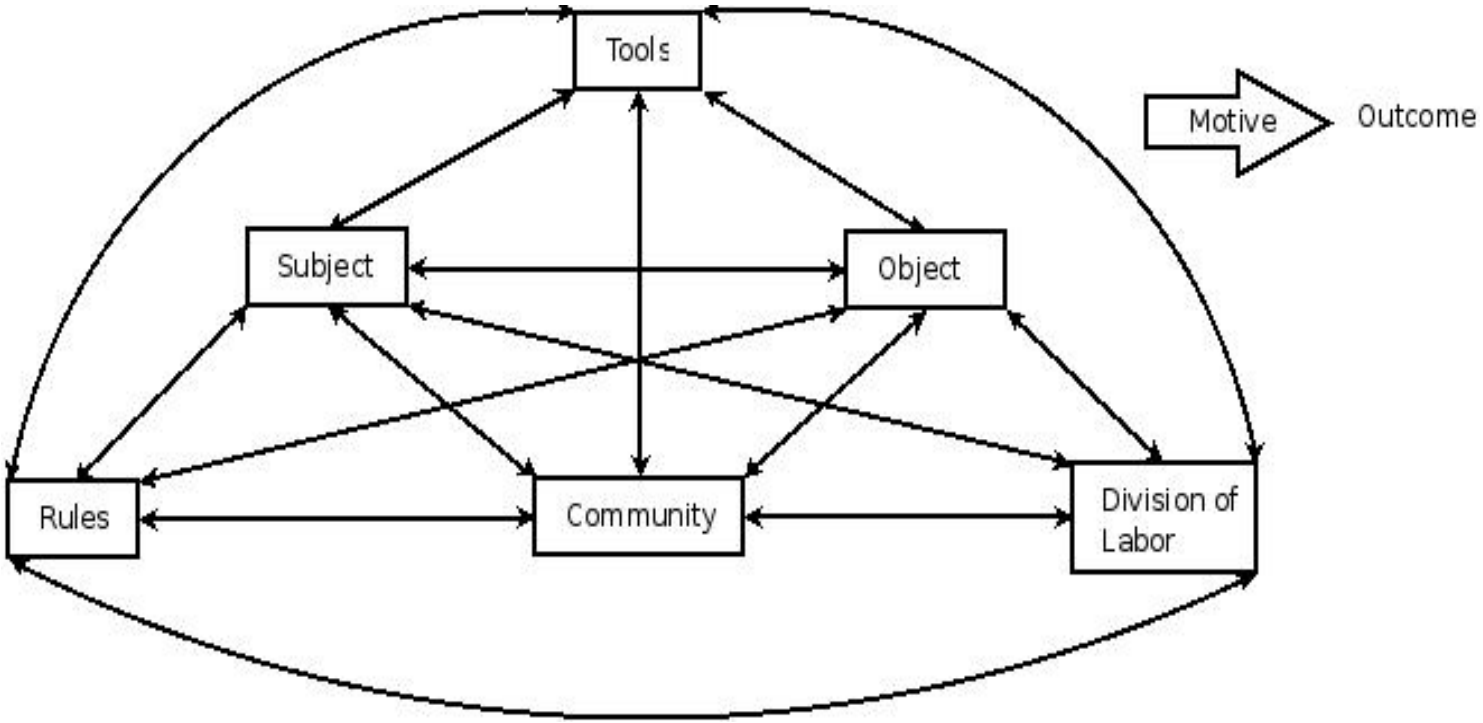
→FlOSS Full Collaboration (correspond plutôt au modèle bazar)

→Véritable effort partagé (organisé ou moins)

Définir un cadre plus formel d'une évaluation

- La ***chose à évaluer*** mais aussi le ***processus d'évaluation*** lui-même
- La chose à évaluer:
  - Evaluation d'un composant … oupppssss mais quid de la communauté et autres aspects liés au travail de la communauté pour développer ce composant
  - Evaluation d'un projet Open source … oupppsss mais si on veut pas exactement tout le composant produit projet: juste une partie de projet ou contrairement, un groupe de composants à intégrer?
  - Evaluation d'une initiative (endeavour) - … il faut définir le périmètre de la chose à évaluer
- OK mais c'est quoi les fondations pour définir « la chose à évaluer »
  - Un projet ou une initiative open source = **activité** d'une communauté pour développer un composant logiciel
- OK mais c'est quoi les fondations pour définir « une évaluation »
  - Une évaluation = **activité** d'un ou plusieurs personnes pour sélectionner une initiative open source
- Et si on regarder ce qui est proposé dans la « Théorie de l'Activité »
  - Date du début 20eme
  - Utilisé en psyco/sociologie pour étudier ce qui influence l'action de faire une activité

Cadre de base de la Théorie de l'Activité



Une initiative FlOSS selon la Théorie de l'Activité

- Objet = Un ensemble de fichiers
  - dont certains sous licences open source (ou creative commons)
  - Pas juste du code source mais aussi d'autres fichiers
- Sujet et Communauté = les personnes qui ont contribué des données en lien à l'ensemble de fichiers considérer
  - Si on prend en compte une communauté d'utilisateurs plus large cela peut influencer la liste de fichiers à considérer … par exemple, tutos sur Youtube, des Q/A sur Stack Overflow, …
- Tools = outils de gestion de source, de test, de gestion et suivi (forge, bug tracker, forum, mailing list etc.) ainsi que dépendances à d'autres composants (open source) utilisées
- Rules and Division of Labor = méthode de travail utilisé par les personnes contributrices pour produire l'ensemble de fichiers à considérer
  - = Processus de développement d'un logiciel

Pour être considérer par une évaluation les règles et la division du travail doivent se trouver dans certains fichiers (ou pages web)

Une évaluation d'une initiative FlOSS selon la Théorie de l'Activité

- Objet = Liste de fichiers à considérer lors d'une évaluation (inclus les données sur les personnes de la communauté et sur ses règles)
- Sujet et Communauté = Personnes qui se partageront le travail d'une évaluation des fichiers considérés
- Règle = Documentation sur les étapes d'une évaluation (= méthodologie QualOSS)
- Outil = Documentation et outils sur la méthode d'évaluation QualOSS spécifiquement utilisée (par exemple, méthode générique de QualOSS pour une Full FlOSS Collaboration)
- Partage du Travail = Documentation qui explicite qui fait quoi lors d'une évaluation

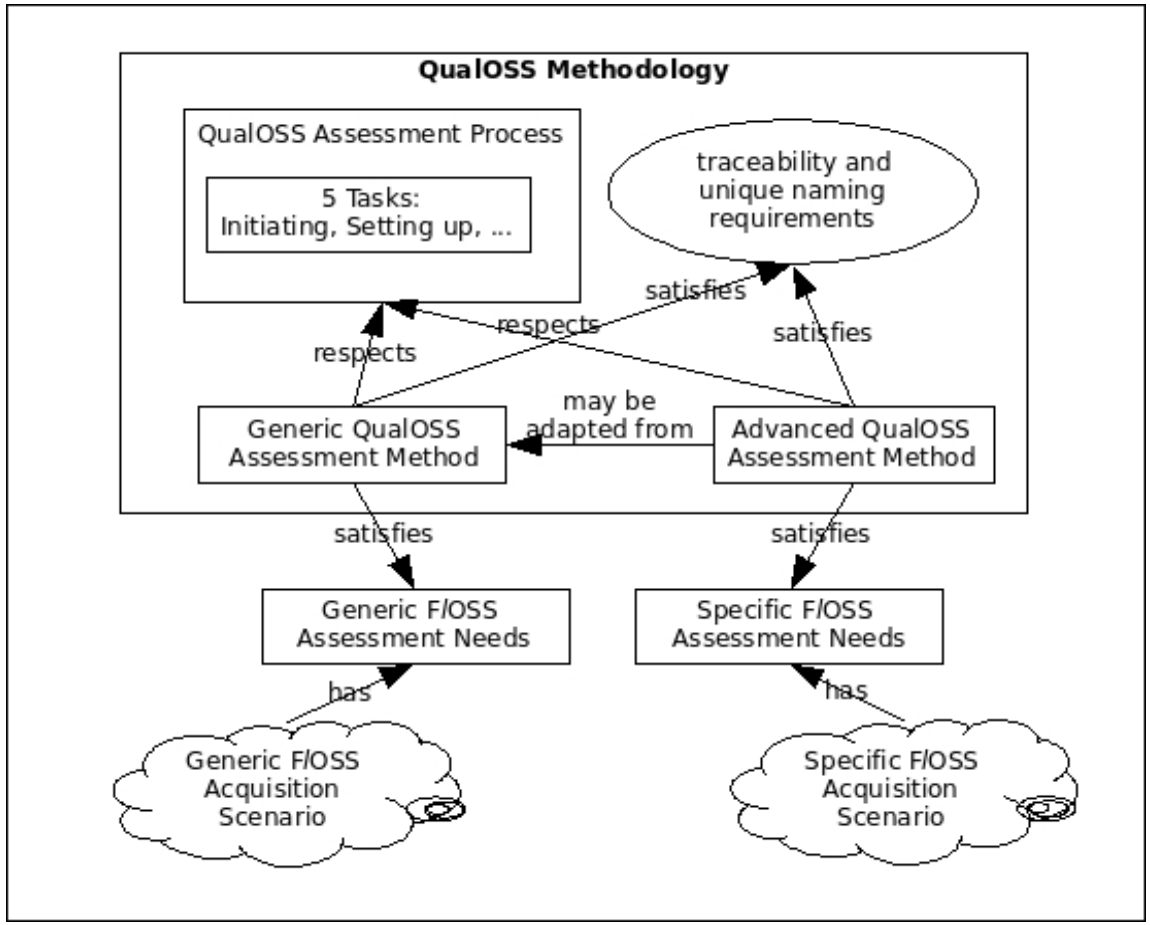
- Besoin d'identifier clairement le « Motive » et « Outcome » → Rendre le contexte de l'évaluation explicit

Par exemple: la méthode QualOSS standard d'évaluation est pour le contexte:

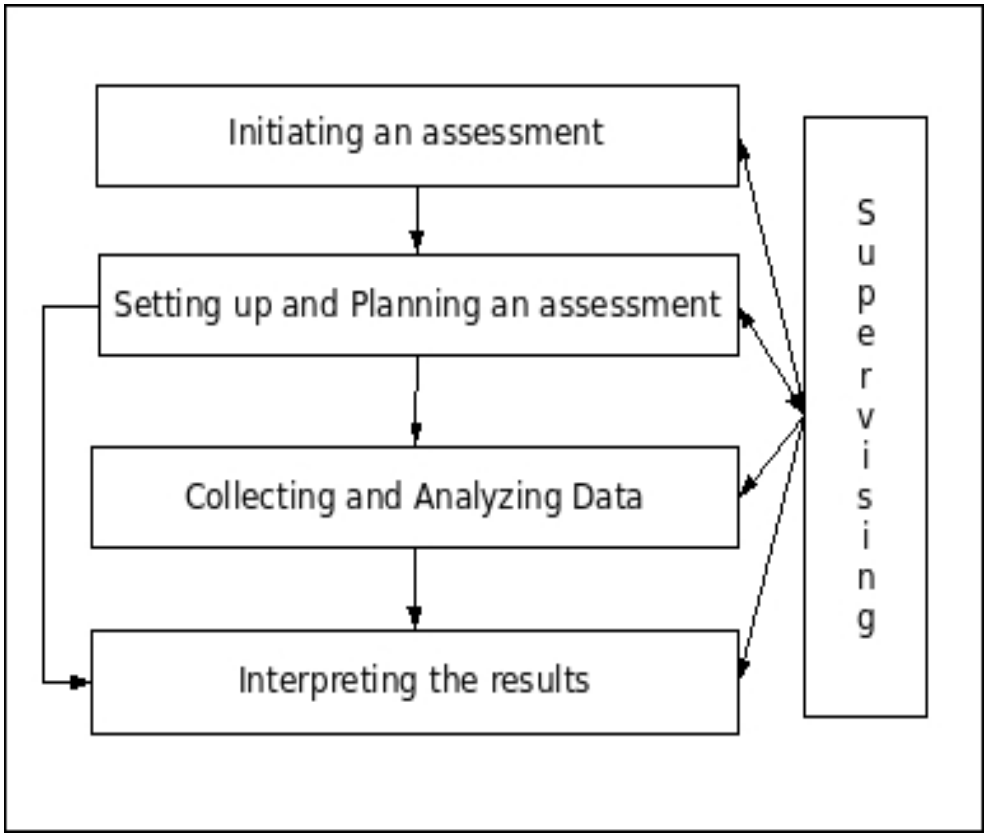
Full FlOSS Collaboration - Sélection d'un composant à intégrer dans un produit logiciel

Mais peut aussi être utilisée à l'inverse par une FlOSS endeavour pour s'auto évaluer et voir si elle fait tout ce qu'il faut pour se rendre attrayante à d'autre

Méthodologie QualOSS et ses méthodes d'évaluation



Tâches du Processus d'évaluation de la Méthodologie QualOSS



Le Méthodologie QualOSS : GQM pour identifier les risques

La Méthodologie QualOSS impose une utilisation normalisée du Goal-Question-Metric à ses méthodes d'évaluation

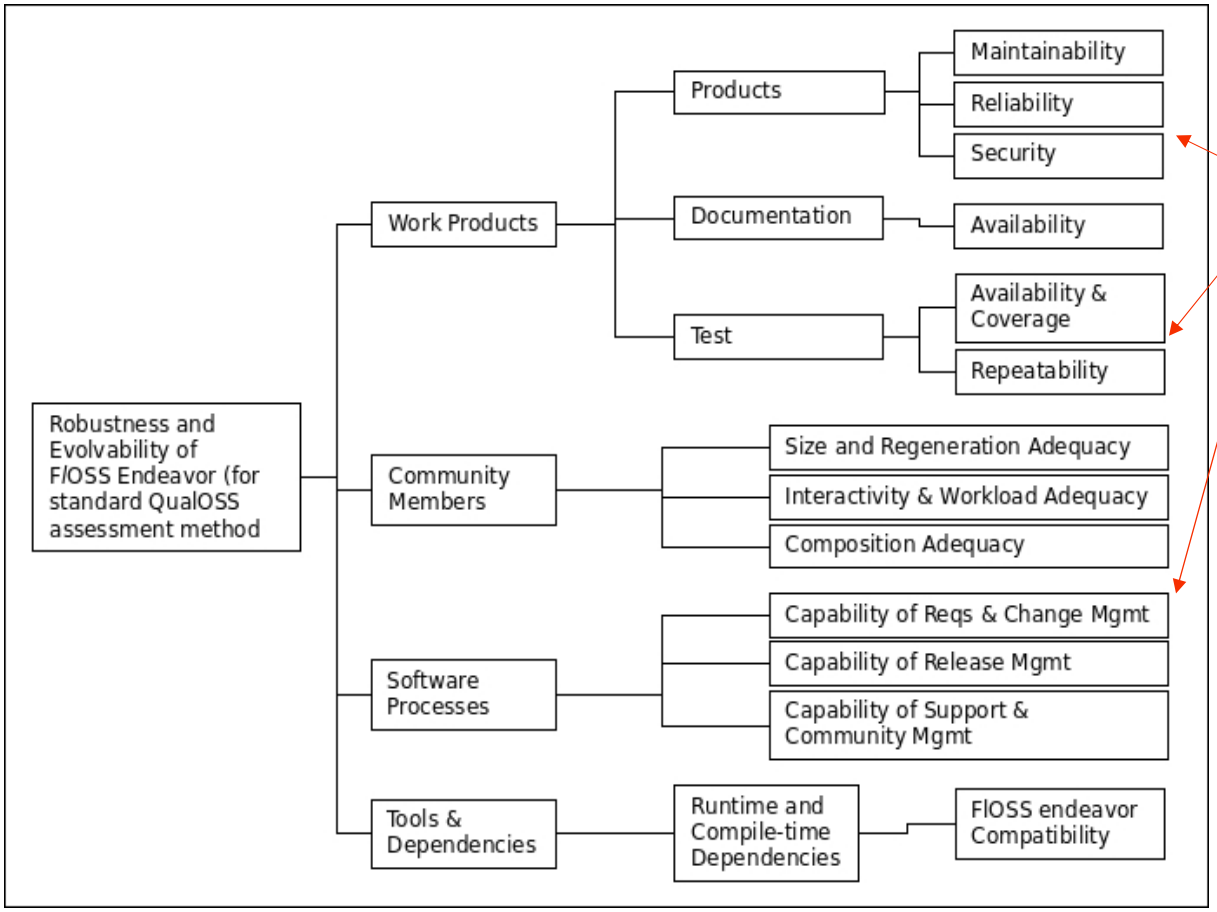
Goals = feuilles de l'arbre.
Pondération à 4 niveau (vert, orange, rouge, noir – cfr Indicateur ci-dessous)

Questions = identifier plus concrètement les points pertinents auxquels une évaluation (eg Full FlOSS collaboration) doit répondre par rapport aux demandes des divers vues des rôles intéressés (Product Manager, Project Manager, Analyste/Développeur, Testeur, Admin IT, rédacteur de documentation, …)

Indicateurs = réponses aux questions d'un point de vue orientée risque
- Formules de métriques
- Normalisées/Echelonnés [0-4]
  ( ]3,4] = risque négligeable,]2,3] = acceptable,]1,2] = conséquent, [0,1 = trop risqué])

Métriques = informations atomiques objectives pour calculer des indicateurs

Eléments évalués par les méthodes QualOSS génériques pour déterminer la *robustesse* et l'*évolutivité* d'une initiative FlOSS



Goals = feuilles de l'arbre.

Robustesse = capacité d'une initiative FlOSS à répondre aux attentes actuelle

Evolutivité = capacité d'une initiative FlOSS à répondre aux attentes futures

Les divers points de vue pour les questions

A partir du contexte « Full FlOSS collaboration » - sélection pour intégration dans un produit logiciel, identifier les **questions posées par les personnes avec différents rôles** dans l'entreprise intégratrice.

- **Product Manager:** qualité d'une initiative FlOSS sur le long terme (plusieurs versions – majeurs et mineurs)
- **Project Manager:** qualité de la version choisie (et version mineur liées) pour intégration + activité de la communauté (développeurs, testeurs, utilisateurs, etc.) sur la version choisie
- **Analyste/Développeur:** Qualité du code et autre fichiers (annexes – build, test, doc)? Réactivité de la communauté de développeurs pour la version choisie?
- **Admin IT, rédacteur de documentation:** même type de questionnement que l'Analyste/Développeur mais sur d'autre types de fichiers

Quelques stats sur la méthode générique QualOSS pour full FlOSS Collaboration

- 13 Leaf Goals (11 évalués – pas d'évaluation des outils et dépendances, ni de la composition de la communauté)
- 83 Questions (11 maint; 4 reli; 4 secu; 26 docu; 5 test; 16 community; 17 soft proc)

Area	Metrics		Indicators		Effort (hours)
	Manual	Automatic	Manual	Automatic	
Work Products / Product / Reliability		9		4	6
Work Products / Product / Maintainability	22	11		15	4
Work Products / Product / Security		38		9	1
Work Products / Test	40			9	1
Work Products / Documentation	41[1]			6	8
Community Members	16	7			8
Software Process	69			17	6
Total	147	65	0	60	34

Tar gz avec les spreadsheets pour évaluer tous les buts, toutes les questions et tous les indicateurs et métriques
https://www.researchgate.net/publication/277957128_Quality_of_Open_Source_Software_QualOSS_-_Standard_Assessment_Method_and_Tool

Exemple d'application du GQM
Product – Maintainability

- Definition: (From ISO Std - 9126, SquaRE 25000)

The degree to which the software product can be modified.

Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.

- Evaluation Goal:

Evaluate the degree of risk related to the maintainability of a FlOSS component based on the current state of the FlOSS component and risk related to the evolution of the FlOSS component based past experiences of the FlOSS endeavor with maintainability issues on the FlOSS component.

Questions d'un Product Manager sur la maintenabilité

- How much time does it take to implement enhancements?

INDICATOR Rapidity of implementation of enhancement proposals	Green: less than 1 month is needed on average to implement a feature request Yellow: between 1 month and 6 months is needed to implement a new feature Red: between 6 months and 1 year is needed to implement a new feature Black: more than 1 year is needed to implement a new feature

- How much code changes are there between major releases, in particular regarding the public interfaces of the FlOSS components?
- Did the FlOSS endeavor undertake significant large refactoring or re-writing efforts on past releases? Were milestones announced and respected?
- Is the evolution of code volume and complexity controlled overtime?

INDICATOR Evolution of number of lines of code between successive releases	Green: there are a few changes in the number of lines (NoL) and cyclomatic complexity (CC) of code in successive releases Yellow: the changes in the NoL and CC of code is average or there is a decrease in the number of lines of code in successive releases Red: there are a lot of changes in the NoL and CC of code in successive releases Black: the changes in the NoL and CC of code between the releases is very high in successive releases

Quelque Résultats d'Indicateurs sur la Maintenabilité

cetic
Your Connection to ICT Research

Results for Findbugs & K3b:

Findbugs Time: a few min

type	name	value	color
metric	its__issue__number_of_accepted_enhancement_proposals	57	
metric	its__issue__number_of_enhancement_proposals	148	
metric	its__issue__number_of_opened_days_of_enhancement_proposals	13747	
indicator	rapidity_of_implementation_of_accepted_enhancement_proposals	241,18	red

K3b Time: (1 day for the script, then) a few min to run it and get the results

type	name	value	color
metric	its__issue__number_of_accepted_enhancement_proposals	283	
metric	its__issue__number_of_enhancement_proposals	527	
metric	its__issue__number_of_opened_days_of_enhancement_proposals	63706	
indicator	rapidity_of_implementation_of_accepted_enhancement_proposals	225,11	red

Tools used: Bicho for findbugs, a bicho like script for k3b (bugs.kde.org)

Quelque Résultats d'Indicateurs sur la Maintenabilité

Results for Findbugs & K3b

Findbugs Time: a few min to combine the results

type	name	value	color
metric	pd__code__number_of_major_releases	2	
metric	pd__code__number_of_minor_releases	4	
metric	pd__code__number_of_major_and_minor_releases	6	
metric	pd__code__cyclomatic_complexity_of_defined_routines_evolution_between_successive_releases	18,60%	
indicator	average_evolution_of_cyclomatic_complexity_of_defined_routines_between_successive_releases	3,72%	green

K3b Time: a few min to combine the results

type	name	value	color
metric	pd__code__number_of_major_releases	3	
metric	pd__code__number_of_minor_releases	6	
metric	pd__code__number_of_major_and_minor_releases	9	
metric	pd__code__cyclomatic_complexity_of_defined_routines_evolution_between_successive_releases	1,30%	
indicator	average_evolution_of_cyclomatic_complexity_of_defined_routines_between_successive_releases	0,16%	green

Tools used: Sissy

Exemple d'application du GQM
Product – Security

- Definition – ISO 25010

Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization

- Evaluation Goal (Viewpoints covered Product Manager and Project Manager)

Evaluate the degree of risk to integrate a F/OSS component based on its security track record. (Based on CVE/NVD entries)

(Code Analysis on Security issues not covered in Standard QualOSS Assessment Method)

Common Vulnerability and Exposure (CVE) - Initiative by MITRE started in 1999

National Vulnerability Database (NVD)

NVD is a product of the NIST Computer Security Division, Information Technology Laboratory and is sponsored by the Department of Homeland Security's National Cyber Security Division.

- 40371 CVE entries – from 2001 until Jan 15 2010
- Avg of 13 CVE entries per day
- All Software; not just F/OSS

Questions d'un Product Manager sur la sécurité

- How risky is the security situation of the F/OSS component based on its whole history? (Eventually include information for the severity of the Security problem history)

Indicator:Global_track_record_of_NVD_Entries_over_time	GREEN: Avg_NVD_entries_per_year <= 12 YELLOW: (12 < Avg_NVD_entries_per_year <= 18) AND (number_of_NVD_entries_for_all_releases_in_last_complete_year + trend_for_NVD_entries_over_the_years * 2 <= 6) RED: !(GREEN OR YELLOW OR BLACK) BLACK: Avg_NVD_entries_per_year > 18
Indicator:Global_track_record_of_High_severity_NVD_Entries_over_time	GREEN: Avg_high_severity_NVD_entries_per_year <= 1 AND trend_for_high_severity_NVD_entries_over_the_years < 0.25 YELLOW: (1 < Avg_high_severity_NVD_entries_per_year <= 4) AND (number_of_High_severity_NVD_entries_for_all_releases_in_last_complete_year + trend_for_high_severity_NVD_entries_over_the_years * 2 <= 0) RED: !(GREEN OR YELLOW OR BLACK) BLACK: Avg_high_severity_NVD_entries_per_year > 6

- Is there a repetitive trend in the security history between the various major Releases?
  - Compare linear regression of NVD datasets of several major releases
  - Problem: many FlOSS do not have enough NVD entries (… Business see this as a risk!)

Security Indicator Results for v1.0

LIST OF BENCHMARK & TEST ASSESSMENTS	Evolution	Evince	EclipsePlatform	Jmeter	JetSpeed	CVSAnaly	Nautilus	Http 1.3
Security - (Work) Product – Code								
Indicator:Global_track_record_of_NVD_Entries_over_time	Green	Green	Green	Green	Green	Green	Green	Green
Indicator:Global_track_record_of_High_severity_NVD_Entrie	Green	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Yellow
Indicator:Global_track_record_of_Medium_severity_NVD_Er	Green	Green	Green	Green	Green	Green	Green	Green
Indicator:Predictability_of_yearly_trend_of_NVD_Entries_ov	Yellow	Black	Black	Black	Black	Black	Green	Red
Indicator:Predictability_of_yearly_trend_of_High_severity_N	Green	Black	Black	Black	Black	Black	Red	Red
Indicator:NVD_Entry_Status_of_selected_release	Yellow	Green	Green	Green	Green	Green	Yellow	Green
Indicator:High_severity_NVD_Entry_Status_of_selected_rele	Yellow	Green	Green	Green	Green	Green	Yellow	Green
Indicator:Track_record_of_NVD_Entries_for_selected_minor	Black	Green	Green	Green	Green	Green	Black	Green
Indicator:Track_record_of_High_severity_NVD_Entries_for_s	Black	Green	Green	Green	Green	Green	Black	Green

No NVD entries

Exemple d'application du GQM

Community Members – Size and Regeneration

- Definition:

The degree to which the size evolution and regeneration of a FlOSS community happens at an adequate rate to maintain a sustainable community size.

- Evaluation Goal:

Evaluate the degree of risk related to the size and regeneration adequacy of a F/OSS community based on the historical evolution of size and regeneration through the life of an FlOSS endeavor compared to statistically validated dataset from communities of many FlOSS endeavours (**calibration in 2009 on 1467 projects on SourceForge**)

Questions d'un Product Manager sur la taille et la régénération

- Has the evolution of new community members reporting bugs remained stable or grown over the history of the FlOSS endeavor?
- Has the evolution of new code-contributing members remained stable or grown over the history of the FlOSS endeavor?
- Has the evolution of new core contributing members remained stable or grown over the history of the FlOSS endeavor?
- What is the evolution of core members who stopped contributing for a significant period?
- Has the evolution of core members who stopped contributing for a significant period been compensated by the joining of new core members around the same time frame?
- What is the average longevity (in month) of committers to the FlOSS endeavor?

Average committers longevity (month)	Black: [1, 5.53555] Red:]5.53555, 8.5789] Yellow:]8.5789, 12.25] Green:]12.25, +∞]

Asterisk – average_committers_longevity = 14,89 months

Exemple d'application du GQM
Process - Capability of requirements and change management

- Definition:

Degree to which the community behind a FlOSS endeavour adequately manages change process to its software product

- Evaluation Goal:

Evaluate the capability of the community to manage change in the product or products they develop

Evaluation built from expert opinions of developers, contributors, users of FlOSS also with extensive knowledge in software development process assessment (eg CMMI and light version adapted to SME such as ISO 29110). Gather core aspects from good practices from renown FlOSS endeavours.

Questions d’un Product Manager sur le processus de gestion du changement

A few examples from the 11 questions:

- What is the general capability of the community to accept external patches?
- What is the general capability of the community to promote external contributors into established commiters?
- What is the general capability of the community to produce and manage enhancement proposals?
- Does the procedure require an impact/viability analysis to be performed? Does the procedure include justification of decisions made? (whether accepted or rejected)
- What is the general capability of the community to deal with software issues reported by external users and contributors?

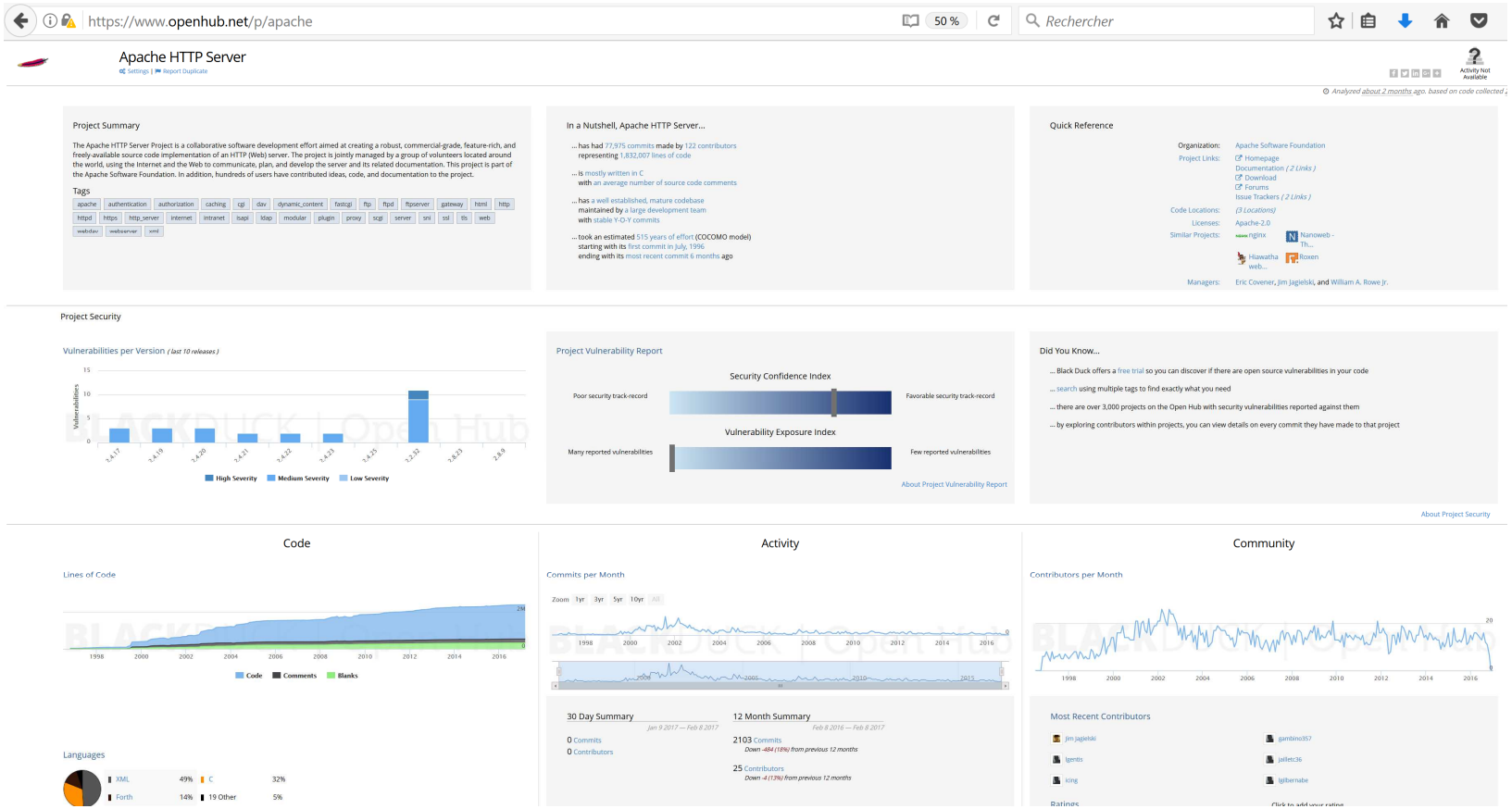
Quelque Résultats d'Indicateurs sur le processus de gestion du changement pour 4 Projets

Characteristics of the Std QualOSS Assessment Method v1.0	LIST OF BENCHMARK & TEST ASSESSMENTS	Evolution	Evince	EclipsePlatform	Jmeter
Software Processes					
Change and Requirement Management					
	Change_Submission_Maturity	4	4	4	4
	Change_Review_Maturity	3	3	4	3
	Change_Review_Adequacy	1	2	3	1
	Committer_Promotion_Maturity	1	1	2	1
	Commit_Review_Maturity	1	1	1	1
	Commit_Review_Adequacy	1	2	1	1
	Enhancement_Proposal_Maturity	1	1	4	2
	Enhancement_Proposal_Adequacy	1	1	3	3
	Issue_Management_Maturity	4	4	4	3
	Issue_Management_Adequacy	3	3	3	1

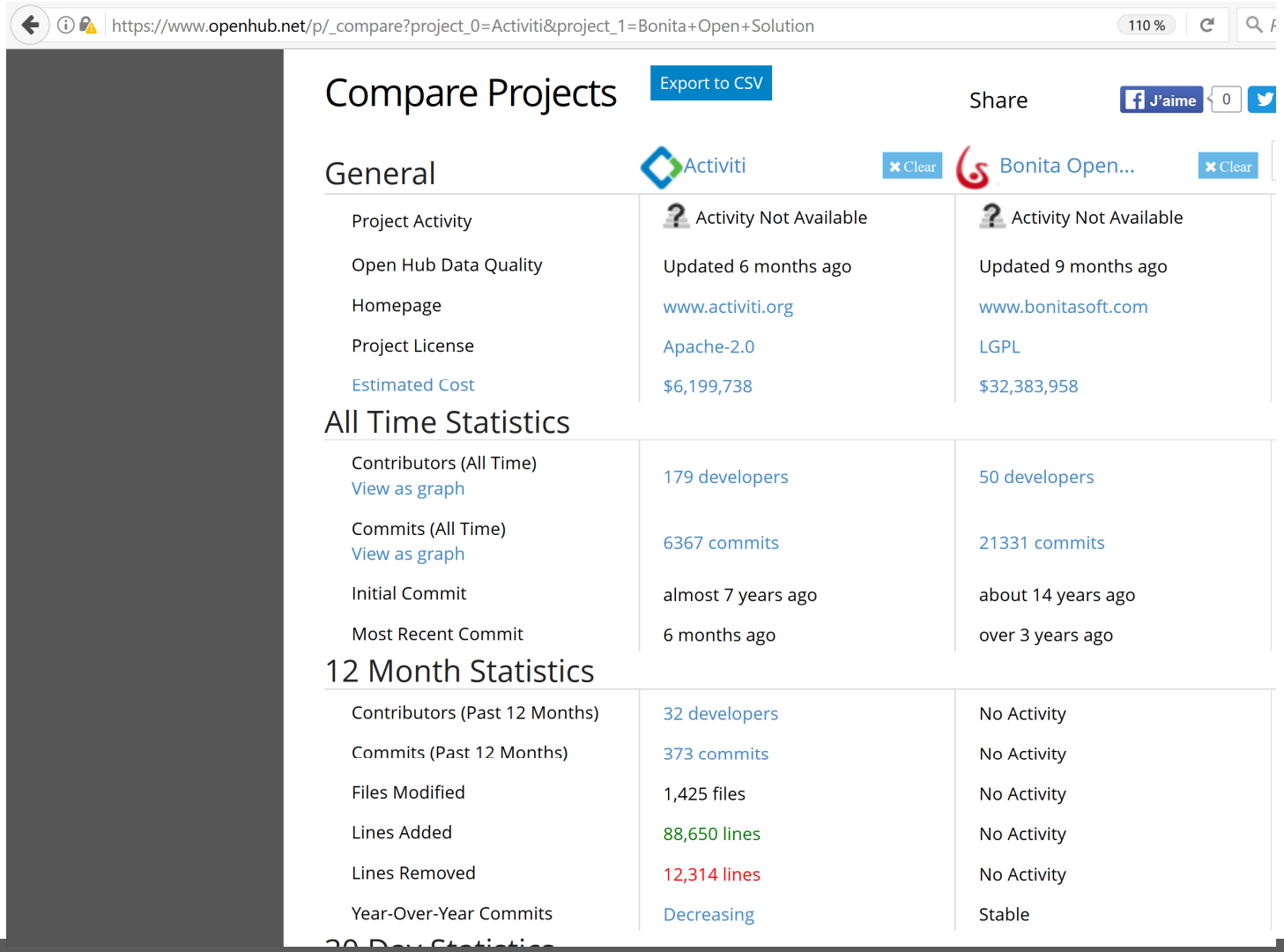
Sans oublier le « Measurement Status

- 0 = Complete Success
- 1 = dataset inexistant
- 2 = dataset incomplete or noisy but measurement could be taken and measure value can be used by indicators
- -1 = Measurement not applicatble (or meaningless in the context of the given assessment)
- -2 = Measurement procedure or tools did not work and could not be adapted
- -3 = No measurement procedure or tools available (i.e. In construction). This status makes it possible to have a measure not yet finalized in a spreadsheet

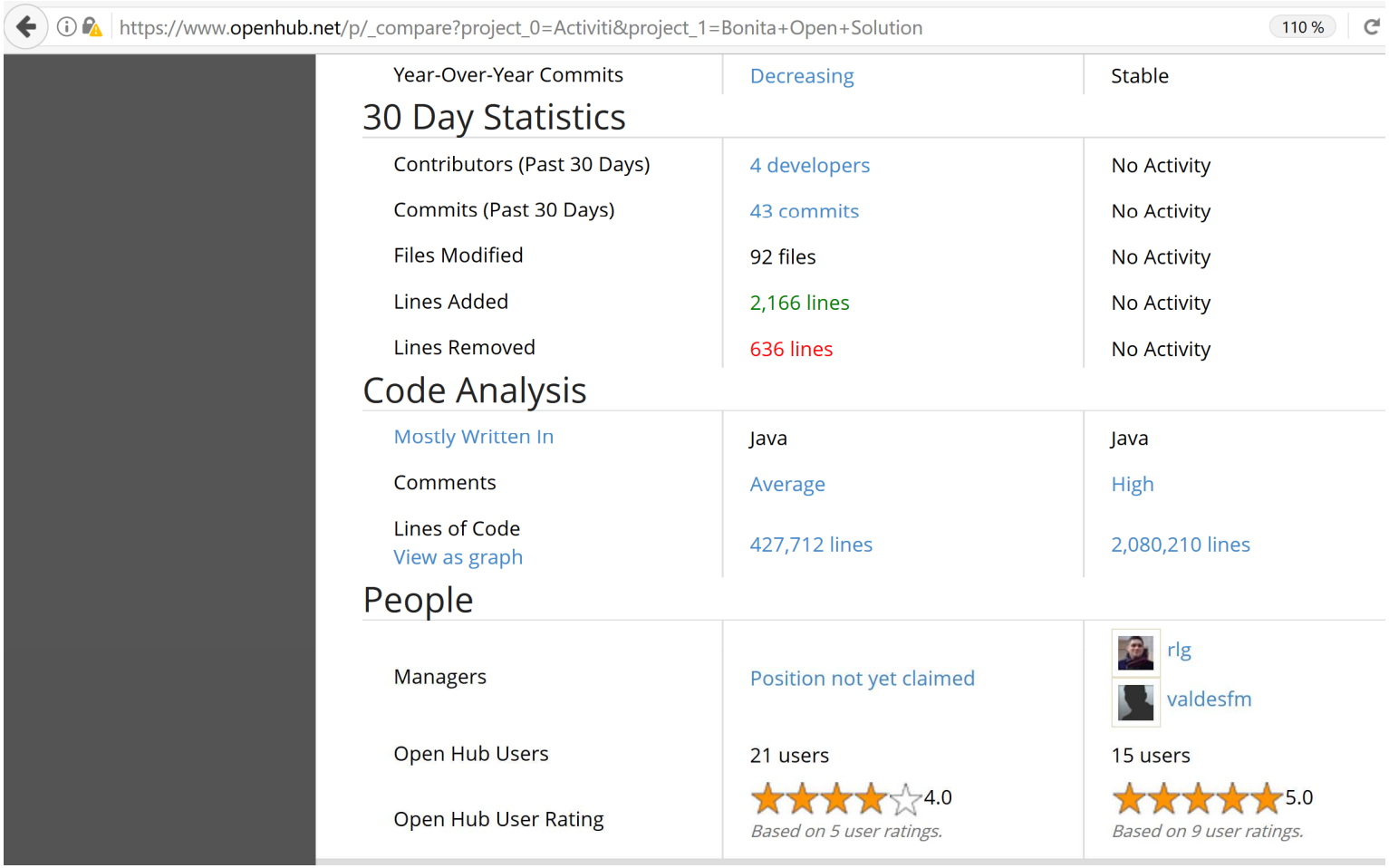
La version turbo d'une évaluation via OpenHub.net – Apache Httpd



Possibilité de Comparaison – ex. Activiti et Bonita



Possibilité de Comparaison – ex. Activiti et Bonita



Your Connection to ICT Research

Aéropole de Charleroi-Gosselies
Avenue Jean Mermoz 28
6041 Charleroi - Belgique


twitter.com/@CETIC
twitter.com/@CETIC_be


linkedin.com/company/cetic


info@cetic.be


+32 71 159 362

Merci

Jean-Christophe DEPREZ

Coordinateur Scientifique

+32 496 316 730

Jean-Christophe.DEPREZ@CETIC.be