Demigha Anis
16703666

Résumé de l'article " Characterizing and Predicting Blocking Bugs in Open Source Project"

Ivaylo Ganchev
22/ Février/2015

1-Introduction:

Dans ce travail nous avons choisi de résumé l'article ' Characterizing and Predicting Blocking Bugs in Open Source Project'. Dans cet article les auteurs ont étudié les bugs de blocage qui empêchent d'autres bugs d'être corrigé dans six projets open source (**Chromium, Eclipse, FreeDesktop, Mozilla, NetBeans et OpenOffice**) et propose un modèle pour les prédire. Leurs objectifs est d'aider les développeurs à identifier ces bugs de blocage dès le début.

2-Caractérisation des bugs:

Avant de présenter l'étude de cas, les auteurs ont discuté trois aspects de bugs qu'ils considèrent importants pour mieux comprendre leur impact.

2.1-Fixation du temps :

Les bugs bloquants sont nocifs pour le processus de maintenance, ils retardent la réparation d'autres bugs (c.-à-d**. Les bogues bloqués**) non parce qu'il n'y a pas assez de ressources pour les corriger, mais parce que les composants affectés par **ces bugs bloqués** dépendent d'autres composants qui doivent être Fixé en premier. D'autre part, des bugs sans dépendances (**c'est-à-dire non bloquants**) sur d'autres bugs peuvent être corrigés immédiatement et, par conséquent, leur temps de fixation peut être plus court.

2.2- Temps pour identifier un bug de blocage :

Certains bugs de blocage sont plus difficiles à identifier que d'autres bugs de blocage. nous pouvons voir cette période de temps comme une mesure du degré de difficulté à identifier les bugs de blocage. Par exemple pour **calculer le temps d'identification d'un bug de blocage** pour les projets utilisant *Bugzilla*, nous cherchons la date de la première apparition de la balise "blocks" dans l'historique des bugs. Lorsqu'aucune information ne peut être obtenue à partir de l'historique du bugs , nous recherchons les historiques des bogues bloqués associés et sélectionnons la date de la première dépendance signalée.

2.3- Degré de blocage :

Le degré de blocage se réfère au nombre de bugs qui dépend Sur le même bug et il peut être vu comme une mesure de la gravité D'un bogue de blocage. Par exemple, le bogue 309165 dans Eclipse est Bloquant quatre autres bugs, cela signifie que son degré de blocage Est égal à quatre. La présence de bugs à forte Devient des goulets d'étranglement pour le maintien et l'évolution du code. Ces bugs de blocage pourraient être le résultat d'une conception de Des composants critiques (par exemple, fortement couplés), des Les exigences, etc.

3-étude de cas :

Dans cette partie les auteurs rendent compte des résultats de leur étude sur six Projets et répondent à deux questions de recherche :

Demigha Anis
16703666

3 .1-Peut-on construire des modèles très précis pour prédire si un nouveau bug sera un bug de blocage?

3.1.1-Motivation :

Nous voyons que les bugs de blocage prennent beaucoup plus de temps à être Fixes que les bugs non bloquants. Nous voulons aider les développeurs à signaler Bloquer les bugs dès le début, afin qu'ils puissent éviter ce gaspillage de temps Et l'effort. Par conséquent, nous voulons construire des modèles de prévision Pour identifier ces bugs de blocage et nous voulons savoir si nous pouvons prévoir avec précision ces bugs de blocage en utilisant les facteurs que nous avons proposé.

3.1.2-Approche :

Nos modèles de prévision sont basés sur les algorithmes d'arbre de décision, car il s'agit d'un modèle explicatif qui peut facilement être compris par les praticiens.

3.1.3-Résultats :

Les résultats qui ont été obtenu à partir de l'expérience est que nous pouvons construire des modèles de prévision qui Peuvent atteindre des valeurs de mesure F de 15% - 42% lorsque les bugs de blocage sont détecter.

3.2- Quels facteurs sont les meilleurs indicateurs de blocage des bugs?

3.2.1 Motivation :

En plus d'alerter sur les bugs potentiels de blocage, nous Souhaiteront identifier le facteur ou le groupe de facteurs qui ont un Impact sur la détermination de ces bugs de blocage. Avec ces Informations, nous pouvons informer les développeurs des facteurs à utiliser Afin de détecter les bugs de blocage dès le début.

3.2.2 Approche :

Nous effectuons une analyse Top Node afin de déterminer Quels sont les meilleurs indicateurs pour déterminer si un bogue sera Bloquant ou non. Dans l'analyse de Top Node, nous examinons les Arbres de décision créés par la validation 10 fois croisée et nous Les occurrences des facteurs à chaque niveau des arbres. Les facteurs pertinents sont toujours proches du nœud racine (niveau 0, 1 et 2). Pls nous traversons vers le bas de l'arbre, les facteurs deviennent moins pertinents.

3.2.1 Résultat :

Dans cette partie les auteurs ont démontré que **le texte de commentaire** est le facteur le plus important dans la détermination des bugs de blocage pour tous les projets sauf pour le projet Mozilla dans lequel le facteur le plus important est **le numéro dans la liste CC.**

4-Menace à la validité :

4.1-Validité Interne :

Pour les annonces, nous avons regroupé les annonces avec moins de cinq contributions dans une catégorie. Cette approche a considérablement réduit le nombre d'annonces différents. Étant donné que toutes les annonce peuvent introduire du bruit et donc avoir un impact sur nos résultats. De plus, nous avons utilisé le nombre de bugs rapportés précédemment comme une expérience d'une annonce. Dans certains cas, l'utilisation du nombre de

bugs précédemment signalés peut ne pas être révélatrice de l'expérience réelle des développeurs, mais des mesures similaires ont été utilisées dans des études antérieures [1]. Notre ensemble de données souffre du problème de déséquilibre de classe. Dans la plupart des projets, le pourcentage de bugs de blocage représente moins de 5% du total des données. Cela provoque est ce que classificateur n'apprend pas à identifier les bugs de blocage.

4.2-Validité externe :

Dans ce travail, les auteurs ont étudié 402 962 rapports de bugs à partir de six projets open source, donc leurs résultats peuvent ne pas généraliser tout les projets de logiciels commerciaux et donc les résultats Peuvent ne pas être généralisés pour tous les projets existant**.**

5-Conclusion :

Dans cet article, les auteurs construisent des modèles de prédiction basés sur des arbres de décision pour prédire si un bug sera un bug de blocage ou non. Comme leur jeu de données, ils ont utilisé 14 facteurs extraits des dépôts de bugs de six grands projets open source. Les résultats de leurs enquête montrent que leurs modèles atteignent 9-29% de *précision* , 47-76% *de rappel* et 15-42% *F-mesure* lors de la prédiction des bugs de blocage. D'autre part, leurs analyse du Top Node montre que les facteurs les plus importants sont Les commentaires, la taille des commentaires, le nombre de Les développeurs dans la liste CC et l'expérience des annoces.

Référence :

Valdivia Garcia, H., & Shihab, E. (2014, May). Characterizing and predicting blocking bugs in open source projects. In Proceedings of the 11th working conference on mining software repositories (pp. 72-81). ACM.

[1] E. Shihab, A. Ihara, Y. Kamei, W. Ibrahim, M. Ohira, B. Adams, A. Hassan, and K.-i. Matsumoto, "Studying re-opened bugs in open source software," Empirical Software Engineering, vol. 18, no. 5, pp. 1005–1042, 2013.