

ABSTRACT

Software Testing is an important phase in the Software Development Life Cycle, to ensure that the verification and validation of the software meet the requirements. Also it is important to have a lot of testing in this phase, before the software is released, to ensure that the software is bug free. However, shipping a software with tons of bugs is something unintended behavior, and unacceptable from the user point of view. Therefore, predicting the software bugs at the early stage of the software development will reduce the time and cost required for testing the software, which saves a lot of money for the companies, moreover it increases the software quality. Predicting software bugs had some challenges, such as generating test data that had been used into the test, and exploring the method paths. This research addresses these issues, by introducing a hybrid approach proposed to identify the potential bugs in the source code for the method under test by constructing an Abstract Syntax Tree model for the method, then traversing the tree and exploring all paths to find the bugs. Hence, Smart Unit Tests are generated accurately to cover all possible execution paths for the tested method. At the end, the proposed approach using static analysis is able to predict all kinds of static bugs and generate the minimal suite of unit tests which are able to cover all the possible execution paths for the tested code. This indicates that the proposed approach achieves good results compared with other techniques in terms of type of bugs that can be predicted as well as the number of generated unit tests that are required to test the code.

الخلاصة

يعتبر إختبار البرمجيات من أهم المراحل في دورة حياة تطوير البرمجيات، لضمان التحقق وصحة البرمجيات، وتوافقها مع المتطلبات. كذلك من المهم في هذه المرحلة من عمل عدة إختبارات للبرمجيات قبل تصديرها، لضمان خلوها من الأخطاء البرمجية. مع ذلك، يعتبر تصدير البرمجيات التي تحتوي على عدة أخطاء سلوك غير مرغوب فيه، وكذلك غير مقبول من وجهة نظر المستخدم. لهذا فإن التنبؤ بمثل هذه الأخطاء في المراحل الأولى سوف يقوم بتقليل الوقت والتكلفة المطلوبة لإختبار البرمجيات، الأمر الذي سيحفظ الكثير من الأموال للشركات. علاوة على ذلك سوف يزيد من جودة تلك البرمجيات. فعملية التنبؤ بأخطاء البرمجيات يحتوي على بعض التحديات مثل توليد بيانات الإختبار والتي تستخدم في عملية الإختبار، و عملية إستكشاف مسارات الإجراء المراد إختباره. هذا البحث سيتطرق لكل هذه المشاكل، بواسطة تقديم منهجية هجينة تم إقتراحها للتنبؤ بالعلل والأخطاء الموجودة في الشفرة المصدرية للإجراء المراد إختباره وذلك بواسطة تشييد نموذج أشجار الصياغة المجردة للإجراء، ثم التنقل وإستكشاف جميع المسارات لإيجاد هذه الأخطاء. فبالتالي يتم عملية توليد مجموعة من وحدات الإختبار الذكية بدقة، والكافية لفحص وتغطية كافة مسارات التنفيذ الممكنة للشفرة المراد إختبارها. أن المنهجية المقترحة حققت نتائج جيدة بالمقارنة مع التقنيات الأخرى من ناحية أنواع الأخطاء المراد التنبؤ بها وكذلك عدد وحدات الإختبار اللازمة لإختبار الشفرة.