



Phương pháp phân tích sự ảnh hưởng khi thay đổi chương trình

Tác giả: Lê Bá Cường – Học viện Kỹ thuật Mật Mã



Giới thiệu

Bảo trì phần mềm hiện nay được đánh giá là một trong những giai đoạn khó khăn nhất, tốn kém nhất và đòi hỏi kỹ thuật cao trong suốt quá trình phát triển phần mềm. Các sản phẩm phần mềm thường xuyên có những thay đổi như thêm mới yêu cầu, sửa lỗi phần mềm trong khi lập trình viên thường chỉ có mã nguồn, không có hoặc thiếu tài liệu phát triển phần mềm. Mỗi sự thay đổi có thể tác động đến rất nhiều phần khác nhau của chương trình. Phân tích sự ảnh hưởng khi thay đổi phần mềm (*Software change impact analysis - CIA*) là các kỹ thuật nhằm xác định những tác động này, đóng vai trò quan trọng trong việc hiểu về chương trình, dự đoán những tác động, đánh giá chi phí, lựa chọn các ca kiểm thử để kiểm thử lại cũng như thực hiện chiến lược sửa đổi.

Các kỹ thuật CIA có 2 hướng tiếp cận chính:

- Dynamic CIA
- Static CIA:
 - Structural Static Analysis
 - Historical Analysis

Mục tiêu

- Đề xuất kỹ thuật CIA mới cải tiến độ chính xác của tập IS dựa trên sự kết hợp một số kỹ thuật truyền thống với các kỹ thuật dựa trên lịch sử phiên bản phần mềm
- Nghiên cứu khả năng ứng dụng trong việc lựa chọn ca kiểm thử để kiểm thử lại và lựa chọn kế hoạch sửa đổi phần mềm khi có yêu cầu nâng cấp, dựa trên việc phân hạng các phần tử trong tập IS
- Phát triển các công cụ hỗ trợ các phương pháp đề xuất và thực nghiệm với những phần mềm kích thước lớn, độ phức tạp cao

Kết quả dự kiến

- Đề xuất phương pháp mới với độ chính xác cao hơn các phương pháp trước đây:
 - Các phương pháp hiện tại thường có độ chính xác chưa cao, tập Impact Set thường có quá lớn (khi áp dụng Static CIA) hoặc thiếu sót nhiều phần tử.
 - Các bài toán thực tế thường có độ phức tạp cao, nhiều loại ngôn ngữ lập trình.
 => Chúng tôi sẽ cải tiến các phương pháp hiện có nhằm nâng cao độ chính xác của tập IS thu được, đồng thời nghiên cứu, ứng dụng kỹ thuật CIA phù hợp cho các bài toán thực tế
- Xây dựng công cụ hỗ trợ có thể ứng dụng trong thực tế
 - Một công cụ phát triển cho các ứng dụng J2EE
 - Một công cụ phát triển cho ứng dụng được viết trên C++
- Đề xuất định hướng ứng dụng các kết quả của kỹ thuật CIA trong bảo trì, kiểm thử lại phần mềm
- Dự kiến:
 - Đăng 01 bài tạp chí quốc tế
 - Đăng 02 bài tạp chí trong nước
 - Đăng 02 báo cáo hội nghị quốc tế

Kế hoạch dự kiến

- 2015: Khảo cứu các nghiên cứu trong nước cũng như quốc tế, bổ sung các kiến thức chuyên sâu về CIA
- 2016: Đề xuất cải tiến phương pháp CIA, áp dụng cho các ứng dụng thực tế (J2EE)
- 2017: Phát triển cải tiến công cụ đánh giá kỹ thuật đã đề xuất
- 2018: Nghiên cứu khả năng ứng dụng CIA vào kiểm thử lại phần mềm, bảo trì phần mềm..
- 2018: Viết báo cáo tổng hợp, chuẩn bị bảo vệ luận án

Tài liệu tham khảo

- [1]. B. Ryder and F. Tip, "Change impact analysis for object-oriented programs," in *Proceedings of the 2001 ACM SIGPLAN-SIGSOFT (PASTE '01)*, USA, 2001, pp. 46–53.
- [2]. X. Ren, "Change impact analysis for Java programs and applications," *Ph.D. dissertation, New Brunswick Graduate School, Rutgers University, New Brunswick, New Jersey, USA, October 2007.*
- [3]. L. Badri, M. Badri, and D. St-Yves, "Supporting predictive change impact analysis: a control call graph based technique," in *Proceedings of APSEC '05*, December 2005, p. 9.
- [4]. M. Lee, A. J. Offutt, and R. T. Alexander, "Algorithmic analysis of the impacts of changes to object-oriented software," in *Proceedings of the TOOLS 34*, Santa Barbara, CA, USA, July 2000, pp. 61–70.
- [5]. X. Sun, B. Li, C. Tao, W. Wen, and S. Zhang, "Change impact analysis based on a taxonomy of change types," in *Proceedings of the IEEE 34th Annual Computer Software and Applications Conference*, Seoul, Korea (South), July 2010, pp. 373–382.
- [6]. X. Sun, B. Li, S. Zhang, C. Tao, X. Chen, and W. Wen, "Using lattice of class and method dependence for change impact analysis of object oriented programs," in *Proceedings of the SAC '11*, ACM, New York, USA, 2011, pp. 1439-1444.
- [7]. B. Li, Q. Zhang, X. Sun, and H. Leung, "WAVE-CIA: a novel CIA approach based on call graph mining," in *Proceedings of the 28th SAC '13*, ACM, New York, NY, USA, 2013, pp. 1000-1005.
- [8]. T. Zimmermann, P. Weissgerber, S. Diehl, and A. Zeller, "Mining version histories to guide software changes," *IEEE Transactions on Software Engineering*, vol. 31, no. 6, pp. 429–445, June 2005.
- [9]. B. Fluri, H. C. Gall, and M. Pinzger, "Fine-grained analysis of change couplings," in *Proceeding of the Fifth IEEE International Workshop on Source Code Analysis and Manipulation 2005*, November 2005, pp. 66–74.
- [10]. B. Fluri and H. C. Gall, "Classifying change types for qualifying change couplings," in *Proceeding of the 14th ICPC 2006*, Athens, 2006, pp. 35–45.
- [11]. L. Hattori, G. d. Santos Jr., F. Cardoso, and M. Sampaio, "Mining software repositories for software change impact analysis: A case study," in *Proceedings of the 23rd Brazilian symposium on Databases*, Brazil, October 2008, pp. 210– 223.
- [12]. S. Lehnert, "A Review of Software Change Impact Analysis", *Ilmenau University of Technology, Technical report* (2011).
- [13]. B. Li, X. Sun, H. Leung, and S. Zhang, "A survey of code-based change impact analysis techniques", *Softw. Test., Verif. Reliab.* 23 (8) , 2013, p. 613-646.
- [14]. X. Sun, Bin Li, B. Li, and W. Wen, "A comparative study of static CIA techniques", In *Proceedings of the Fourth Asia-Pacific Symposium on Internetware*, ACM, New York, NY, USA, 2012, Article 23 , 8 pages.
- [15]. Hattori L, Guerrero D, Figueiredo J, Brunet J, Damsio J. On the precision and accuracy of impact analysis techniques. *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science*, Portland, Oregon, 2008; 513–518.