Researching of the problem of solution automation of software systems compatibility

Polikashin D., Enikeev A., Georgiev V.
Kazan Federal University, 420008, Kremlevskaya 18, Kazan, Russia

Abstract

© Springer International Publishing AG 2017.During software development is often a lack of backward compatibility between different versions of software components. It will lead to: violation of the ecosystem that is built around this software component, uncoupling community of programmers who use this software in their projects, which entails the general slowdown in the implementation of projects related to this ecosystem and the development of the ecosystem itself; the use in the development of software components that are absolute, which in turn entails problems with safety; the economic costs, during the implementation of a software system migration to a new version of the software component.

References

[1] Fluri, B., Wursch, M., Pinzger, M., Gall, H.C.: Change distilling: tree differencing for fine grained source code change extraction. IEEE Transactions on Software Engineering №11, pp. 725–743 (2007)

[2] Neamtiu, I., Jeffrey, S. Foster, Hicks M.: Understanding source code evolution using abstract syntax tree matching. In: MSR '05 Proceedings of the 2005 International Workshop on Mining Software Repositories, New York, USA (2005)

[3] Ponomarenko, A., Rubanov, V.: Automated verification of shared libraries for backward binary compatibility. In: VALID '10: Proceedings of the 2010 Second International Conference on Advances in System Testing and Validation Lifecycle. IEEE Computer Society (2010)

[4] Ponomarenko, A., Rubanov, V.: A combined technique for automatic detection of backward binary compatibility problems. In: PSI'11: Proceedings of the 8th International Conference on Perspectives of System Informatics, Springer. (2011)

[5] Ponomarenko, A., Rubanov, V.: Automated analysis of feedback Linux binary compatibility libraries. In: Seventh-Conference of Developers of Free Software. Pereslavl-Zalessk, 2010 (2012)

[6] Richards, M.: Software architecture patterns, O'Reilly (2015)

[7] Saukh, A.M.: Semantic analysis of some aspects of the source code of programs on the basis of formal syntax and semantics of the specifications. Appl. Discrete Math. № 5 (adj.), 110–111 (2012) Irkutsk

[8] Turbin, A.: Complementary Hashing Subsets. Seventh-conference of developers of free software, Pereslavl-Zalessk (2010)

[9] Welsch, Y., Poetzsch-Heffter, A.: Verifying backwards compatibility of object-oriented libraries using boogie. In: Proceedings of the 14th Workshop on Formal Techniques for Java-like Programs, pp. 35–41. NY:ACM New York, USA (2012)

[10] Wursch, M.: Improving Abstract Syntax Tree based Source Code Change Detection, Software Evolution and Architecture Lab (Harald Gall), Zurich: University of Zurich (2006). http://www.merlin.uzh.ch/publication/show/2913

[11] AngularJS Developer Guide.Migrating from 1.3 to 1.4. https://docs.angularjs.org/guide/migration

[12] GitHub. https://github.com/

[13] Google Code Project. https://developers.google.com/

[14] IBM DeveloperWorks. http://www.ibm.com/developerworks/

[15] Intel Developer Zone. https://software.intel.com

[16] Microsoft Developer Technologies. https://dev.windows.com

[17] Migrating to Direct3D 11. https://msdn.microsoft.com/en-us/library/windows/desktop/ff476190(v=vs.85).aspx#direct3d_9_to_direct3d_11

[18] OpenCV 3.0 API Changes. http://docs.opencv.org/master/db/dfa/tutorial_transition_guide.html#gsc.tab=0

[19] Qt API Changes. http://doc.qt.io/qt-5/sourcebreaks.html

[20] Ruby version policy changes starting with Ruby 2.1.0. https://www.ruby-lang.org/en/news/2013/12/21/rub-version-policy-changes-with-2-1-0/

[21] Sourceforge. http://sourceforge.net/

[22] StackOverflow. http://stackoverflow.com/

[23] Yandex Technologies Catalog. https://tech.yandex.ru/