# Software Testing

#### Gordon Fraser and José Miguel Rojas

**Abstract** Setiap program nontrival berisi ny nontrivial program contains some errors in the source code. These “bugs” are annoying for users if they lead to application crashes and data loss, and they are worrisome if they lead to privacy leaks and security exploits. The economic damage caused by software bugs can be huge, and when software controls safety critical systems such as automotive software, then bugs can kill people. The primary tool to reveal and eliminate bugs is software testing: Testing a program means executing it with a selected set of inputs and checking whether the program behaves in the expected way; if it does not, then a bug has been detected. The aim of testing is to find as many bugs as possible, but it is a difficult task as it is impossible to run *all* possible tests on a program. The challenge of being a good tester is thus to identify which are the best tests that help us find bugs, and to execute them as efficiently as possible. In this chapter, we explore different ways to measure how “good” a set of tests is, as well as techniques to generate good sets of tests.

All authors have contributed equally to this chapter.

G. Fraser

University of Passau, Passau, Germany e-mail: [gordon.fraser@uni-passau.de](mailto:gordon.fraser@uni-passau.de)

J. M. Rojas

University of Leicester, Leicester, UK e-mail: [j.rojas@leicester.ac.uk](mailto:j.rojas@leicester.ac.uk)

© Springer Nature Switzerland AG 2019

S. Cha et al. (eds.), *Handbook of Software Engineering*, <https://doi.org/10.1007/978-3-030-00262-6_4>

123

# Software Testing

#### Gordon Fraser and José Miguel Rojas

**Abstract** Any nontrivial program contains some errors in the source code. These “bugs” are annoying for users if they lead to application crashes and data loss, and they are worrisome if they lead to privacy leaks and security exploits. The economic damage caused by software bugs can be huge, and when software controls safety critical systems such as automotive software, then bugs can kill people. The primary tool to reveal and eliminate bugs is software testing: Testing a program means executing it with a selected set of inputs and checking whether the program behaves in the expected way; if it does not, then a bug has been detected. The aim of testing is to find as many bugs as possible, but it is a difficult task as it is impossible to run *all* possible tests on a program. The challenge of being a good tester is thus to identify which are the best tests that help us find bugs, and to execute them as efficiently as possible. In this chapter, we explore different ways to measure how “good” a set of tests is, as well as techniques to generate good sets of tests.

All authors have contributed equally to this chapter.

G. Fraser

University of Passau, Passau, Germany e-mail: [gordon.fraser@uni-passau.de](mailto:gordon.fraser@uni-passau.de)

J. M. Rojas

University of Leicester, Leicester, UK e-mail: [j.rojas@leicester.ac.uk](mailto:j.rojas@leicester.ac.uk)

© Springer Nature Switzerland AG 2019

S. Cha et al. (eds.), *Handbook of Software Engineering*, <https://doi.org/10.1007/978-3-030-00262-6_4>

123