

# Features of software testing in the development of geographic information systems

*Ekaterina Galimova*<sup>1,\*</sup>

<sup>1</sup>Saint-Petersburg State University of Industrial Technologies and Design, High School of Printing and Mediatechnologies, 13, Jambul lane, St. Petersburg, 191180, Russian Federation

**Abstract.** In this article, the method of selecting a method of testing geographic information systems is proposed. Nowadays, thanks to advances in information technology new methods of data exchange, positioning technologies, improvements in spatial databases, there has been a significant leap in development of geographic information system. When choosing a testing method, it is recommended to take into account not only the financial component, but also the specific characteristics of the tested software system. A list of questions is developed by which qualitative properties of software systems are formalized. The list of questions is based on the ISO/IEC 25010:2011 model. The relevance of this work follows from the fact that new technologies of software development require new approaches to testing software systems. Properly developed testing technique improves the quality of geographic information systems created. Improving quality increases the competitiveness and demand of software products.

## 1 Introduction

Manual testing is more flexible and better mimics user actions. Automated test sets are easy to reuse. Semi-automated testing helps to break down a large automation task into several small parts. The interaction between these parts is carried out in manual mode [1, 2]. In this article, we offer a method of test method selection using expert evaluations of product managers or programmers and testers. The product manager or developer is offered a list of questions that have been developed taking into account the quality criteria of the geographic information system. In the first group of questions, the answer “yes” means the usability of testing automation usage. Positive answers to questions of the second group increase the priority of manual testing. The “yes” answers to third-group questions increase the advantage of semi-automatic testing.

---

\* Corresponding author: [galim81@mail.ru](mailto:galim81@mail.ru)

## 2 Materials and Methods

### 2.1 Sample questions from the questionnaire with comments

Will fuzzing be carried out? Fuzzing is a software testing technique that involves sending incorrect, unexpected, or random data to an application. The input data is generated by a special random algorithm with mutations, written to a file or transmitted to the application input in a network packet. As a result of data processing, the process may end with failure. This situation means vulnerability. The fuzzing algorithm is described in [3, 4].

Do you plan to use acceptance tests? This type of complex testing is necessary to determine the readiness level of the system for subsequent operation. Testing is done from the client point of view, verifying all actions and scenarios that the client should and should not perform. An important part of acceptance testing is beta testing on real devices by end users. Some beta testing tools interact with many internal and external testers, inviting them to participate in projects through email notifications. Tools that use cloud services are also effective for acceptance testing.

Is user interface convenience important? It is recommended to use the "fault tolerance" principle [5]. If the field requires natural numbers, the programmer can prohibit entering negative values at the code level. It is suggested to use the plan-do-check-act (PDCA) method: collect user feedback on the operation and design of the application; improve the software product taking comments into account. Emotional response evaluation involves studying the user feelings after the interaction with the software product has been completed. The feedback indicator will help to assess how well the user remembers working with the software after a long break. Repeating operations after timeout should be faster for an experienced user. The "Accuracy" parameter calculates the number of errors the user has made when working with the application. The user does not have to remember the information from the first part of the dialog to use it in subsequent parts. "Efficiency" is the number of steps and time it takes a user to complete basic tasks. The software product should be convenient for both experienced and inexperienced users. Error messages should clearly articulate the problem and offer a constructive solution. The software product should provide the ability to customize frequently used operations.

Is installation testing required? During the testing process, make sure that all user data is saved after updates. Updates must be supported by current and earlier versions of the operating system.

Will regression testing be performed? In regression testing, it is recommended to avoid tests redundancy [6] because one of the current issues is the limited resources allocated in the project for testing [7]. In general terms, regression testing consists of two steps. First, a test check for the detected error is developed. Secondly, the created test check is restarted after each change in the code.

### 2.2 Selecting a software test method using a multicriterion problem

Binary responses received from a product manager or developer form

$$\beta = (\beta_1, \beta_2, \dots, \beta_k) \quad (1)$$

In the testing group, these answers are provided with weights

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k) \quad (2)$$

It is recommended to assign weights as a small discrete set, for example  $[0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1]$ . In the first group of questions, the choice is made on the following principle in terms of this criterion: whether it is convenient to use automatic testing (score 1); rather convenient (estimate  $\frac{3}{4}$ ); there is doubt in its convenience (score  $\frac{1}{2}$ ); there are serious doubts about its convenience (score  $\frac{1}{4}$ ); completely inconvenient (score 0). In the second group, the principle in terms of this criterion is reverse - whether it is convenient to use manual testing (score 0); rather convenient (estimate  $\frac{1}{4}$ ); there is doubt in its convenience (score  $\frac{1}{2}$ ); there are serious doubts about its convenience (score  $\frac{3}{4}$ ); completely inconvenient (score 1). In the third group, the principle is the same as in the first. Estimates of  $\beta_1, \beta_2, \dots, \beta_k$  are inconvenient for further processing, since  $\beta_j = 0$  answer when multiplied by weight always yields zero, and hence weight is not taken into account. It is proposed to separate the answer from zero (and from 1 at the same time) by replacing the numbers 0 and 1 with the numbers  $\frac{1}{4}$  and  $\frac{3}{4}$ . If  $\beta_i = 0, i = 1, \dots, k$ , then it is replaced by  $b_i = \frac{1}{4}$ . If  $\beta_i = 1, i = 1, \dots, k$ , then it is replaced by  $b_i = \frac{3}{4}$ . Weights  $\alpha_1, \alpha_2, \dots, \alpha_k$  for evenness of further processing according to group rules are replaced by  $a_1, a_2, \dots, a_k$ . For the first group of questions focused on choosing automatic testing, weights  $\alpha_1, \alpha_2, \dots, \alpha_{k1}$  only need to be normalized by groups - divided by  $k1$ :

$$a_i = \alpha_i / k1 ; i=1, \dots, k1 \tag{3}$$

For the second group of questions focused on the selection of manual testing, it is necessary to invert weights  $\alpha_{k1+1}, \alpha_{k1+2}, \dots, \alpha_{k2} (\alpha_i \rightarrow (1 - \alpha_i))$  and change the sign in them:

$$a_i = (\alpha_i - 1) / k2 ; i=k1+1, \dots, k2 \tag{4}$$

For the third group it is necessary to center estimates  $\alpha_{k2+1}, \alpha_{k2+2}, \dots, \alpha_k$ . Estimate of  $\frac{1}{2}$  must be neutral (zero). Given the number of criteria in this group ( $k - k1 - k2$ ), we obtain:

$$a_i = (\alpha_i - \frac{1}{2}) / (k - k1 - k2) \text{ для } i= k2+1, \dots, k \tag{5}$$

Next, we calculate the group convolution by the formula:

$$D = a_1 b_1 + a_2 b_2 + \dots + a_k b_k \tag{6}$$

The specified value is always within  $-1 \leq D \leq 1$ . The expected value of such convolution on the set of all possible initial estimates  $\beta_1, \beta_2, \dots, \beta_k$  and weights  $\alpha_1, \alpha_2, \dots, \alpha_k$  equals to 0. Let us assign values  $D_1$  and  $D_2$ , which determine the type of testing. Further we trisect the interval  $[0, 1]$ :  $D_1 \geq 0, D_1 < D_2, D_2 \leq 1$ . This leads to the third case where a formalized solution (manual, automated) is considered as not sufficiently valid. As specific values  $D_1$  and  $D_2$ , it is proposed to take values  $D_1 = 0.33$  and  $D_2 = 0.66$ .

Selecting the  $R1_G, R2_G, \text{ or } R3_G$  test type for a particular module or the entire PC is done by the value of the  $D$  function.

When  $D \geq D_2$ , automated testing method is recommended. With  $D \leq D_1$ , manual testing method is recommended. With  $D_1 < D < D_2$ , mixed testing is recommended.

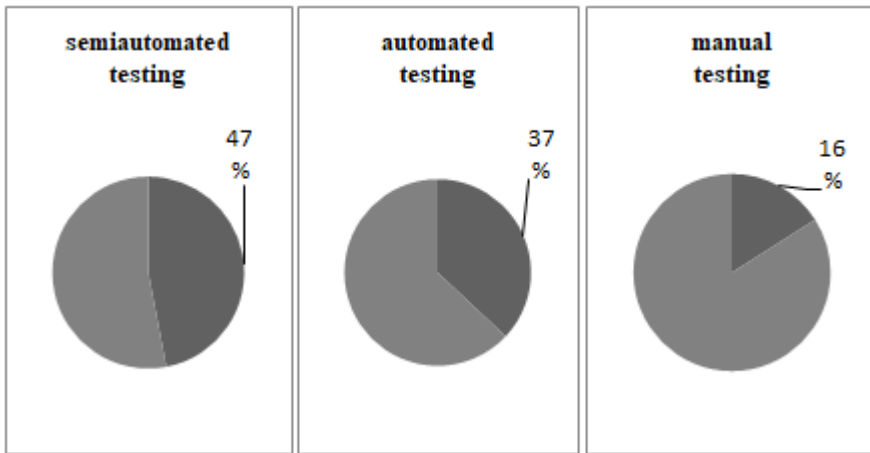
### 3 Results and Discussion

This technique was applied to three classes of geographic information systems (GIS): mobile GIS, server GIS, desktop GIS [8]. Technical characteristics of these GIS classes are given in Table 1:

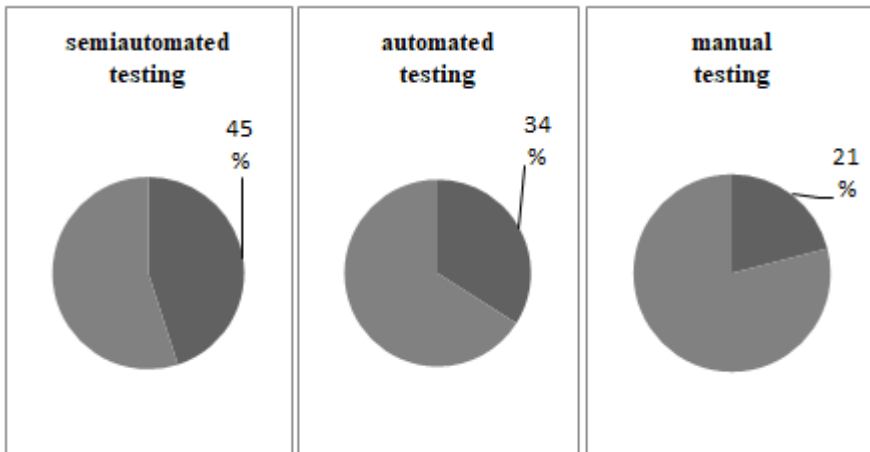
**Table 1.** Technical characteristics of GIS classes studied

Technical characteristics	Desktop GIS	Mobile GIS	Server GIS
Installation process	Standardized, does not require special skills	Installation with specialized services or manually.	Requires specific knowledge
Installer availability	Yes	Built into the OS.	No
Operating modes	Works with data on the current device	The native applications work with the hardware part. Can work autonomously. Hybrid applications use both query and internal smartphone resources.	Query mode
Internet connection required	Partially autonomous operation	Partially autonomous operation for native and hybrid applications	Compulsory connection
User Interaction Tools	Mouse, keyboard, graphic tablet, VR-headset	Keyboard, sensing elements. Voice (Google Now). Gesture (Hand wave).	Mouse, keyboard
Limiting memory resources	Generally no significant restrictions	Significant restrictions	No significant restrictions
Number of potential users	Limited number	Very large target audience	Large target audience
Stationality	Stationary	Performed “on the go”	May have mobile versions
Uninstaller	Standardized, process performed semi-automatically or automatically	Built into the OS	Requires administrator® participation

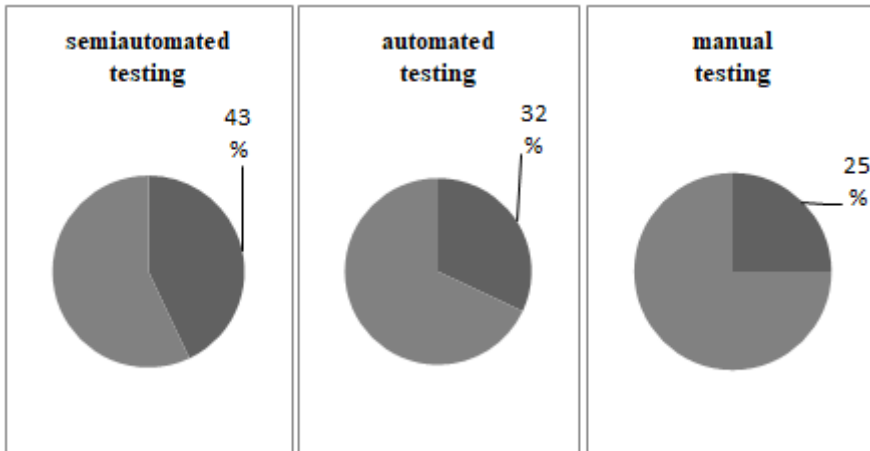
Further, figures 1 — 3 show the results of the technique application to the above GIS classes:



**Fig. 1.** The result of the technique application to the mobile GIS class



**Fig. 2.** The result of the technique application to the server GIS class



**Fig. 3.** The result of the technique application to the desktop GIS class

## 4 Conclusion

Data analysis obtained as a result of the technique application developed by the author makes it possible to conclude that it is preferable to apply mixed testing to all GIS classes considered in the article. The percentage of manual testing varies depending on the GIS class.

## References

1. N. Chen and S. Kim, Puzzle-based automatic testing: Bringing humans into the loop by solving puzzles, *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE, 140–149 (2012)
2. Xiujiang Li, Yanyan Jiang, Yepang Liu, Chang Xu, Xiaoxing Ma, Jian Lu., User Guided Automation for Testing Mobile Apps, *Proceedings of the 21st Asia-Pacific Software Engineering Conference (APSEC)*, 27-34 (2014)
3. Sam Malek, Naeem Esfahani, Thabet Kacem, Riyadh Mahmood, Nariman Mirzaei, Angelos Stavrou, A framework for automated security testing of android applications on the cloud, *2012 IEEE Sixth International Conference on Software Security and Reliability Companion*, 35-36 (2012)
4. Riyadh Mahmood, Naeem Esfahani, Thabet Kacem, Nariman Mirzaei, Sam Malek, Angelos Stavrou, A Whitebox Approach for Automated Security Testing of Android Applications on the Cloud, *Proceedings of the 7th International Workshop on Automation of Software Test*, 2012. 22-28 (2012)
5. Ansar-UI-Haque Yasar, Davy Preuveneers, Yolande Berbers, Ghasan Bhatti, Best practices for software security: An overview, *Proceedings of the 12<sup>th</sup> IEEE International Multitopic Conference*, 169-173 (2008)
6. Sacha Lity, Manuel Nieke, Thomas Thüm, Ina Schaefer, *Journal of Systems and Software*, **147**, 46-63 (2019)
7. Peng Xiao, Bin Liu, Shihai Wang, *Journal of Systems and Software*, **143**, 159-171 (2018)

8. A. G. Karmanov, A. I. Knyshev, V. V. Eliseeva, Geographic information systems of territorial administration, 121 (St. Petersburg: ITMO University, 2015)