



ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2016; 2(11): 439-443
www.allresearchjournal.com
Received: 02-09-2016
Accepted: 03-10-2016

R Manjula

Associate Professor (School of Computing Science and Engineering), VIT University, Tamil Nadu, India

Jainishkumar Anghan

School of Computing Science and Engineering (SCOPE) VIT University, Vellore, Tamil Nadu, India

Neelanjana Bhattacharyya

School of Computing Science and Engineering (SCOPE) VIT University, Vellore, Tamil Nadu, India

Comparison between static testing tool and dynamic testing tool which tools is good to use for find bugs

R Manjula, Jainishkumar Anghan and Neelanjana Bhattacharyya

Abstract

The Reviews of code is conducted in software implementation because the aim of this is we want to improve our code quality. During this, its check coding standard and many other things by different SAT (Static Analysis Tools). Static Analysis is Activity which is Verify given Source Code by using SAT.

Using SAT, we can determine and solve run-times error like array Out of Bound, divide by Zero, Null Reference inters and we also determine the complexity of code by use of metrics produced. On Other side DAT (Dynamic Analysis Tools) is working on running environment. Difference between SAT and DAT that SAT is examine the while Code is not running and DAT is examine the code while running of Code.

Static analysis is automated, which means you can do the analysis without executing the program or Code. DAT is automated but it is Provide Flexibility to what you want for Scan. Most of the STAs can only scan source code, which is problematic. In this paper, we investigate when it is better to use SATs and when we use DATs. In SAT, it is mandatory of having Source Code for Analysis But in Dynamic Analysis tools, not mandatory of having Source Code for Analysis. In SAT, there is need Highly Trained Developer for use of SAT But in DAT there is no need of Highly Trained Developer for to use of Analysis Tools. We Will Conduct the Interview with 15 Developers to know when we use which type of Analysis Tools for Analysis the Code. We prepare the list of Question for Developers and Ask How we improve the Analysis of Code.

Keywords: Static analysis tool, dynamic analysis tool, developers

1. Introduction

Quality and reliability of software is now days becoming important things. Now Days, our faith on Software is incensing day by day because most of the people use software product in their daily life. For assuring software quality there is many various ways available, considering code examination and accurate testing. When Software is lead to failure its can cost companies momentous amount of resources and also cost [2, 3].

In SAT, to examine code it is not mandatory to run the code and it also helping to ensure about that the quality of software is maintain or not during the process of development. There is many several of paths to implement automatic static analyses in our projects, including when creating a project or module and when developer want to analysis code. SATs [1] may Also take input that what type of bugs developers wants to find and also sometimes it's may define new types of bug pattern which is not in system. For Static we can use tools like PMD [4]. To help explain the SATs, we can look into Find Bugs [5] as a consideration of how it work [6]. We include Find Bugs in the Eclipse [7] and Net Beans [8] as Plugins. It's also having its distinct tool or we can execute it via using of cmd. When it will executing in the IDE, Every errors is entrusted an intensity, determining importance of the defect; either it can be high, medium or low, each of them are represented by red, yellow and green background respectively. Sometimes may be some situation occurs and developer may use SATs to solve the error of code during Software implementation.

Mostly, DATs are very useful when we are not able to detect which type of errors occurs in our project or in other words we say that the error which is much hard to find while static analysis of code. DATs are called as dynamic because they require the code to be in a running state. DATs used to know what happened behind the scenes when the code is in running state.

Correspondence

R. Manjula.

Associate Professor (School of Computing Science and Engineering), VIT University, Tamil Nadu, India

They are used for detect memory leak, identify pointer arithmetic error like null pointer, to find the time dependency.

DATs are frequently combine with SATs and CMTs. Take an example, a developer want to analysis of their code to find and locate error by using static analysis and then he can remove that error before the whole code is send for testing purpose. We have many Dynamic Tools ^[9] for analyze of code, like Daikon, Dynamic Load Testing, Program Analysis, Program slice, etc. Daikon ^[10] is usually use for java, C, C++ for Analyze the code. Programming Slice is also most useful tool for dynamic analysis, in this we use to part the program in slice and then we analyze code and remove memory leak, pointer problem, and unused variable. In software analysis, Static analysis is always executing in a non-runtime environment using source code. Typically SATs will examine program code for all possible run-time behaviors and seek out coding flaws, back doors, and potentially malicious code. Dynamic analysis take up the opposite approach and is executed while a program is in operation. A dynamic test is oversees system memory, functional behavior, response time, and overall performance of the system.

Literature Survey

There Is So Many People Studied On how SATs and DATs are used. Our work targets on how the Developer's Perception of Static Analysis and Dynamic Analysis Tools.

Flemming Nielson, Hanne Riis Nielson, Chris Hankin a study that both are Static and Dynamic analysis Tools are required for testing. Dynamic analysis tool become overhead when finding bug on hundred line of code. So when we use Dynamic Tools after that use Static Analysis tool for that particular part and solve the bug ^[11].

Y. P. Khoo ET. Al's studied and focused on how the SATs work on source code and how can we improve the SATs for analysis of code ^[12]. For Improvement of SATs They developed one UI. Which is known as Path Projection. It was designed for imagination of path based error report and using this toolkit we described error as false positive or false negative. In this, we can trace error one by one using BLAST, Which is Software tool for analysis of code. Using this study we can improve our work on UI for SATs and DATs.

L. Layman ET. Al's are studied about that if during the analysis of code we find an error so what type of factor can be considered by developer to decide that the error is consider or solve or not at that time ^[13]. The longer fault stay in code it will take more time to fix. So for that they are use automated fault detection (AFD), Which is presenting a prompt to developer with recently introduce fault, which is helping developers for fix error quickly and also decrease the time for fixing error. For this they interviewed some developer to find out the factor which are useful to decide that error is important to solve when it notified or it can be ignorable for now.

Methodology

For this, we were interviewed some developer, among them some have 1 year of experience and some have 9 to 10 year of experience. This interviewed was held approximately 30-40 minutes. Aimed of this type of interview is to get required information from software developers ^[16]. We were prepared list of question of this interview but this questions

was not fixed we added or remove the question on the answered got from the developers.

We created and changed the question on depending on the response from the trail interview of our 4 trail participants or developers ^[13].

3.1 Discussant

For getting more understandability about SATs and DATs, We interviewed 10 to 12 developers which was in big as well as small software company. We imitated same methodology which is imitated by Layman ET. Al's for studying 18 developers or Discussant ^[13]. We recruited that Participants by our industrial contact with the company we sent mail to that company and company made its developers interview with us. Most of our participants were professional developers at a large company and some were developers from small company. Some of them had less professional experience i.e. experience with 1-2 year. We conducted interview with our developers using remote method, we made a phone call or video call for interviewed our Participants.

3.3 Research Questions

We want to learn about SATs and DATs, so for this we asking following question to our participants,

- **RQ1:** According to you which is better tools for Analysis? (Static or Dynamic)
- **RQ2:** How both of the tools can fit on your Workflow?

We consider workflow as writing, modifying and inspecting the code which is made by developers.

We asked this type of question to the developer because it is help to improve the user interface of SATs and DATs and also for future work. Research was showed in way that developers were use the tools in right manner so it can be avoid interrupt using tools ^[14, 15]. This Interview is focused on developer's experience with finding bugs or defects with SATs and how to use DATs. We learn about how experienced while using SATs and DATs and to find why the developer don't use static analysis tools. The interviews are organize in three parts: Question and short response (3.2.1 Part I), Interactive Interview (3.2.1 Part II), and Participatory Design (3.2.1 Part III).

3.2.1 Part I (RQ1): Questions and Short Responses

In this Session, we asked question to developer about their general use of SATs and DATs, and in response we got their understanding and its opinion about SATs and DATs. The question which is included in this part is following:

- What is your first experience about Static Analysis tool?
- What is your first experience about Dynamic Analysis tool?
- Do you face any difficulty to use SATs and DATs?
- What kind of Improvements you want in both type of tools?
- What is critical characteristics of a good SATs or DATs in your opinion?

3.2.2 Part II (RQ2): Interactive Interview

In this part we interact with the developer about SATs and DATs so that it's called the Interactive Interview. The aim of this, we want to know how developers are understands the SATs and DATs. This also allowed us to find out more

details about information that how the developers use this type of analysis tools. Our aim during second part is to use the information which are get during this interview. We asked our participants how they use this tools, so we better understand their workflow and how they insert, update their code using analysis tools. We listed following question during this interview:

- After using SATs and DATs, what is your next Move?
- Does SATs and DATs help in assessing of warning, what you do about it?
- Do you think that code suggestion and quick fix is really helpful?
- Do you think use of DATs are very useful in projects?

4. Results

Through this paper expectation there is two type of remark on static and dynamic tool, positive remark and negative remark, which is equally important to define our questions, in many cases positive remark is reason to discontinue of interview and sometimes negative remark reason of discontinues. We have to separate the statement which is given by the developer; it may possible that both tools get negative remark as well as positive remark.

4.1 RQ1: According to you which is better tools for Analysis? (Static or Dynamic)

We asked developers that between static or dynamic analysis tools which one is best relatively. The developer a fore mentioned that Static associate in nursing analysis involves no dynamic execution of the code and might observe do able defects in an early stage, before running the program. Static analysis is completed once writing and before unit testing. Therefore it helps to confirm correct writing standards and conventions area unit accustomed construct the program. On the opposite hand Dynamic program analysis tools might need loading of special libraries or perhaps recompilation of program code. Using DATs we analysis the code during running which is not we able to using SATs. Therefore it depends on our demand that we might like. We are able to not say that which one is better.

RQ1.1: What is your first experience about Static Analysis tool?

Another our friend and very good software developer is asked what type of experience had during using of SATs. He replies that “Actually for using static analysis tool good skill is required. And most importantly static analysis tool is not fast enough”.

RQ1.2: What is your first experience about Dynamic Analysis tool?

We asked the same developer his first time experience with dynamic analysis tool. According to them it provides flexibility and it can be conducted against any application. Some of them told that it was very difficult to find the error in DATs, because of it is compile whole project instead of some files of project.

RQ1.3: Do you face any difficulty to use SATs and DATs?

We asked our developers’ friend that which type of difficulty you had during using of SATs and DATs at very first time. Most of said that we cannot tell which one is easy

or difficult. Some portion you feel it’s easy and some portion you feel it’s difficult for both the cases.

RQ1.4: What kind of Improvements you want in both type of tools?

We asked to developer that what kind of development you want in both the cases. One of our Participants replied that the false Positive i.e. it means when SATs shows error there is no error at that place and very large number of warnings are the main disadvantage of using SATs to find bugs, however one of our developer believes that ways of the configuration of your tools is play large part in flinging error. On the other hand for DSTs provide false sense of security i.e. developers think that everything in their code is fine and in actually it is not.

RQ1.5: What is critical characteristics of a good SATs or DATs in your opinion?

The last question was to share their view on critical characteristic of good static as well as dynamic analysis tool. He replied the critical characteristic of good SATs is its ability to analysis after coding and before executing unit tests. So it helps to ensure proper coding standards and conventions are used to construct the program. On the other hand for Dynamic analysis it might identifies vulnerability that might be false negative in static code analysis.

4.2 RQ2: How both of the tools can fit on your Workflow?

We asked other developers how both this current tool fit into the workflow. Some of them said that sometimes a developers running a SATs during the implementation, but for this developer not stop their work and run the tools for doing the specific task; we conducted interview with female developer called Reena. She used stopping point for run her code. During the interview, we find that there are many ways to fit SATs and DATs into their development workflow. Some developer prefer SATs and DATs run in background so it is very easy to find error at the development time. Using tools in background, the tools is always in running mode so we do not stop our work for call SATs and DATs. On other hand, some developer does not prefer to use IDEs, so that they must do compiler integration for using of SATs and DATs. Most of the developer which is use integration method are very serious about using integration of tools to the project. On the other hand software quality using dynamic analysis complements static analysis. The methodology include identification of run-time bugs by running the application under review on a selected workflow using dynamic analyzers.

After performing short question and answer season we have conducted an interactive season with our developer. Here we asked them few question about both of the tools in details. On the basis of their reaction we get the clear view about both of the tool and can reach to our conclusion.

RQ2.1: After using SATs and DATs, what is your next Move?

Now through this interactive season we asked one of our developer his next move after getting the feedback from the required using tool. The answer was quite obvious. After getting the error every developer want to fix the problem as soon as possible. That’s why every developer want a proper and clear list of error. This is the reason some developer

prefer Find Bugs static analysis tool. And don't like dynamic tool as it's unable to provide exact location of error.

RQ2.2: Does SATs and DATs help in assessing of warning, what you do about it?

After that we ask our developers that is it helpful in assessing of warning in SATs and DATs. According to the answer we came to know that static tool only can give you the list of error or warning before code execution but it will not indicate how to fix that warning. That is the part of the developer where developer have to find the way to fix the error or warning. But they also mention that the way static tool list down the error it quite helpful for them to fix after getting this. And it's very important to get the error at very stage as it reduces the cost and risk of correcting and detecting of bugs. If this bugs is not fixed and detected by the customer then it would harm the reputation of the organization.

RQ2.3: Do you think that code suggestion and quick fix is really helpful?

Then simply we ask them that do you think that code suggestion and quick fix is really helpful. They readily accept that everybody want quick fixes and if suggestion readily available then they would love to have it. But they also mention that every developer should have some sense of responsibility on their code and they should keep track of their code and error. So that they should not repeat the same mistake. And they also agree that getting suggestion with error or warning will reduce the potential of a developer. So it's a good idea but on the other hand it's not an efficient and fruitful idea.

RQ2.4: Do you think use of DATs are very useful in projects?

During this interactive session last question that we ask that do you think that dynamic analysis tool really helpful in project? They said obviously dynamic tool is very essential for big project. It gives you different functional error like memory leakage, pointer arithmetic error and other functional error which need to find out and fixing it is required before product is launched and the problem is discovered by the customer. But they also agree most of the time only dynamic tool is not sufficient. So they use both static and dynamic tool together.

Result Understand ability

Actually it totally depend upon which tool we are using. Like in static analysis tool if we use Find Bugs tool then it will give you the list of error in a much organized way. So it is very helpful for any developer to fix its error. On the other hand if you use Program Slice Dynamic analysis tool then it will slice our program and then check and find out error in each slice and remove the error like memory leakage and functional error etc. But it also depend upon which programming language and IDE we are using. Like Find Bugs static tool works better for Eclipse and Net Beans IDE. On the other hand Program Slice use for C, C++, Java. So requirement of tool depends upon requirement and circumstances.

5. Impact

Tools like Find Bugs assess each warning based on priority level i.e. which error have how much of priority. If one user

is believe in taking risk, risk which causes the failure of code and whole project are getting high priority. Risk Based assessments will be very from organization to organization and project to project. From Survey we find that using Find Bugs user are find the bugs with priority but user are not interested because it take so much time to find. As we know Find Bugs are used high priority, medium priority and low priority for assign to error. From the interview we find that most of the user get high priority error in code. So users prefer the solving high priority problem first other than low priority based error. The main goal of every developer is to find and solve the error in the minimum time. It is also based on time giving by the organization to developer. DATs are used for finding the complexity of program and find the error on run time. Some error are not showing on at the implementation time like array out of bound, null exception and divided by zero exception. So for finding this type of error in whole code is very difficult so that we also use SATs after using DATs

Comparison

Now after researching on different aspect and asking different question to different developer it is quite easy for us to compare both of this analysis tools. SATs are used to find error on exact location on code. It is managed by a developer which is fully experienced of using this tools and quickly fix the error in minor time. Whereas for dynamic tool it's more flexible compare to static and applicable for any programming language. On other hand Dynamic analysis, use the code on run time environment and it is need to be execute the code for analysis. DATs take care and monitoring of compile time, allocation of memory on system, response time of system and also monitor overall performance of the system. This method is same as third party are take care of both code and developers. SATs and DATs are prefer opposite way for analysis of code. Both of SATs and DATs have advantages as well as Disadvantages.

6. Conclusion

From the Survey we can say that SATs are useful to find important error in the code. It is particularly important in Security error in which very costly when we deploying such type of code with error because when we upload this type of code it is create security issues. Using Find Bugs. We can find the run time error like Null Pointer Exception, which is very costly to solve this because in this type of error is having million dollar cost. So it is important to run the SATs and DATs in a cost effectiveness way for analysis. On the other hand for dynamic it also has few advantages for that developer generally prefer dynamic tool over static tool but the same time it also has some drawback which is generally future concern. So we cannot conclude by saying which one is good or which one is bad. Both is required and necessary but depends upon the condition and requirement of the developer and it's vary upon organization to organization and project to project.

7. Future Scope

In future we can build static tool that will consume less time. It need to be more flexible so that it can support all programming languages. Major drawback that need to improve is to produce false positive and false negative. On the other hand for dynamic it should be improve so that it provide correct sense of security i.e. show security error if

there is any security error is present and address the correct location of error. Another disadvantage of DATs is that it is very difficult to trace the location of error in the project. This are all the things that can be improved future to get the better performance in near future.

8. References

1. List of tools for static code analysis, https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis
2. Briand LC, Thomas WM, Hetmanski CJ. Modeling and managing risk early in software development, in Proc. ICSE, 1993, 55-65.
3. Nagappan N, Ball T. Static analysis tools as early indicators of pre-release defect density, in Proc. ICSE, 2005, 580-586.
4. PMD, <https://pmd.github.io/>
5. Find Bugs, <http://findbugs.sourceforge.net>.
6. Nathaniel Ayewah, William Pugh, David Hovemeyer J, David Morgenthaler, John Penix. Using Static Analysis to Find Bugs, IEEE Softw., 2008; 25(5):22-29.
7. Eclipse. <http://www.eclipse.org/>.
8. Net Beans. <http://www.netbeans.org/>.
9. List of Dynamic Tools, https://en.wikipedia.org/wiki/Dynamic_program_analysis
10. Diakon, <http://plse.cs.washington.edu/daikon/>
11. Flemming Nielson, Hanne Riis Nielson, Chris Hankin. Principles of Program Analysis.
12. Khoo YP, Foster JS, Hicks M, Sazawal V. Path projection for user-centered static analysis tools, in Proc. PASTE, 2008, 57-63.
13. Layman L, Williams L, St. Amant R. Toward reducing fault fix time: Understanding developer behavior for the design of automated fault detection tools, in Proc. ESEM, 2007, 176-185.
14. Gschwind T, Oberleitner J. Improving dynamic data analysis with aspect-oriented programming.
15. Johnson B. A Study on Improving Static Analysis Tools: Why are we not using them? in Proc. ICSE, Student Research Competition, 2012.
16. Hove S, Anda B. Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research, in Proc. METRICS 2005, 1-10.