

Improving Software Quality Attributes of PS using Stylecop

Rajesh Kulkarni¹, P.Padmanabham² and M.S.Namose³

¹ BSCOER Engg. College affiliated to Pune University, India

Received: 10 December 2012 Accepted: 4 January 2013 Published: 15 January 2013

Abstract

Software product quality improvement is a desired attribute and a strenuous effort is required to achieve that. Static Code Analysis (SCA) is used to find coding bugs. Stylecop is an SCA tool from Microsoft. In this paper SCA and stylecop usages are discussed. Also a comparison of Software Testing, software Metrics Source Code Analysis (SCA) is done. Product PS designed by author1 and author2 is introduced. PS application is analyzed using static SCA tool Stylecop. After analysis recommendations were applied to PS. The results showed improved software quality attributes.

Index terms— SCA, PS, stylecop, software quality.

1 Introduction

Software quality improvement is of foremost importance for product success. Techniques such as software testing, Usability testing and source code analysis are prevalent. Also secure programming to tackle code-based vulnerabilities is in vogue nowadays [2]. Our work focuses on evaluation of PS product designed by author 1 and author 2 using static source code Analysis (SCA) tool Stylecop. In this paper overview of SCA, comparison of SCA with software testing and software metrics [1] is done. Also introduction to PS product [8], [9] and Stylecop [1], [5], [6], [7] is given. The organization of the paper is as followed: section I which is the current section introduces SCA, section II distinguishes SCA with testing and metrics along with that evaluation criteria of SCA tool selection is also done, section III gives the introduction to PS, section IV details stylecop tool, section V discusses evaluation of PS product using stylecop and section VI lists results of analysis of PS using stylecop. The static analysis approach is meant to review the source code, checking the compliance of specific rules, usage of arguments and so forth. Static code analysis is the analysis of computer software which is performed without the actual execution of the programs built from that software, as opposite of dynamic analysis (testing software by executing programs) [6]. The output of static analysis tools still requires human evaluation [6].

The BCS SIGIST defines static analysis of source code as the "analysis of a program carried out without executing the program" [2]. This definition emphasizes the contrast to dynamic analysis, where the behavior of program is observed while it is executed. Static analysis can be used for varied purposes such as software quality improvement, code optimizations and identifying vulnerabilities [2]. Manual reviewing of code is a form of static code analysis, it is a time consuming process and it should be done at the very early stage of design [6]. Manual review kind of static code analysis can be either a self-review or a third party review [6].II.

2 SCA vs Testing and Metric

Software Testing is the process of executing a program or system with the intent of finding errors [10]. Software testing tools are programs that are trying to find errors, defects, bugs, failures, etc in the software [1]. Test cases are generated and results are compared with expected results. Nonconformance leads to debugging the program. Correctness is the minimum requirement of software, the essential purpose of testing. Correctness testing will need some type of oracle, to tell the right behavior from the wrong one [10]. While according to [11] software metrics is defined as the continuous application of measurement-based techniques to the software development process and its products to supply meaningful and timely management information, together with

the use of those techniques to improve that process and its products. Software metrics are programs or tools that collect information about the software regarding its characteristics for example metrics such as Lines Of Code (LOC), size, complexity, Number of functions per class, etc [1]. While the high level goal of software metrics is similar to software testing in improving the overall quality of the software, there are several differences between software testing and metric tools [1]. Software metrics are usually collected after testing and after making sure that program is free from errors (at least syntactic errors) [1].

3 III.

4 PS Product

The PS (personal Secretary) web application is targeted for online users who has many email ids, other site ids, does online banking and online transactions. Presently he stores his email ids and passwords, ATM pins as emails or in written form on a paper which he keeps in his wallet. His constant fear is if he loses his wallet there may be misuse of his ids and pins. He also faces the problem of recollecting the ids, passwords Year complete solutions for the above vulnerabilities through a single get way using the internet as the sole medium. It will enable the online user to retrieve ids, passwords and ATM pin numbers at will 24/7. The administration module will enable a user to allow addition, deletion and updating of ids, passwords and documents.

The target user also requires documents frequently such as Copies of mark memos, certificates and updated resume for applying for jobs. Copies of documents such as form 16s, salary slips, ids for loan application. Insurance policy numbers PS was designed using TEIM(The Evolved Integration Model) of software engineering and human computer Engineering proposed by us [8].This paper is a continuation of efforts for validating the utility of TEIM and acceptance from peers [9].

5 Stylecop

StyleCop is an open source SCA tool from Microsoft that checks. NET code for conformance of several design guidelines defined based on Microsoft's .NET Framework [1][7]. StyleCop evaluates the style of C# source code in order to enforce both a set of style and consistency rules. Style guidelines are rules that specify how source code should be formatted and dictate whether spaces or tabs should be used for indentation and the format of for loops, if statements and other constructs [5]. The goal is to define guidelines to enforce consistent style and formatting and help developers avoid common pitfalls and mistakes. StyleCop contributes to this maintainability by encouraging consistency of style, which in turn makes it easier for developers to pick up existing code and work with it productively, and by encouraging plenty of documentation for future developers to read thereby improving the long term maintainability of the source. [6] Historically, different development groups have used drastically different coding styles. Many teams have used inconsistent coding styles within a single product or even a single source file. StyleCop was originally written to provide a simple and efficient way to enforce a common coding style for C# code throughout Microsoft. Over time, StyleCop evolved to include new rules that go beyond style checks. [6] V.

Evaluation of PS using Stylecop PS application was developed using .NET. Stylecop was configured with .NET during installation. Each of the file of PS was evaluated using Stylecop. Stylecop issued warnings and errors in the following categories: Naming, maintainability, documentation, ordering, readability, spacing, and layout [7]. Except maintainability all the categories belong to refactoring. Refactoring indicates cosmetic improvement in software without changing its functionalities [1].Priority numbers assigned are based on the importance and impact of its program [1].Some sample warnings are shown in TABLE ?. PS Screenshots showing Evaluation by Stylecop are shown in in Figure 2 and Figure 3. After evaluating PS with stylecop manual corrections were made in the code of PS. These corrections led to improvement in software quality attributes of PS as shown in TABLE ?.

6 Conclusion

In this paper we analyzed product PS designed by author1 and author2 using stylecop-a static source code analysis (SCA) tool. Manual corrections in code of PS as per recommendations of stylecop were done. The results showed Manual review was found to be a tedious process in terms of time and planning. As our product is being targeted as a client-server program security was not an issue and so our focus was less on security and more on styling and usability.

7 Global

VII. ¹



Figure 1: Figure 1 :

Email Details: On successful login, user can see email details by default.



Figure 2: Improving

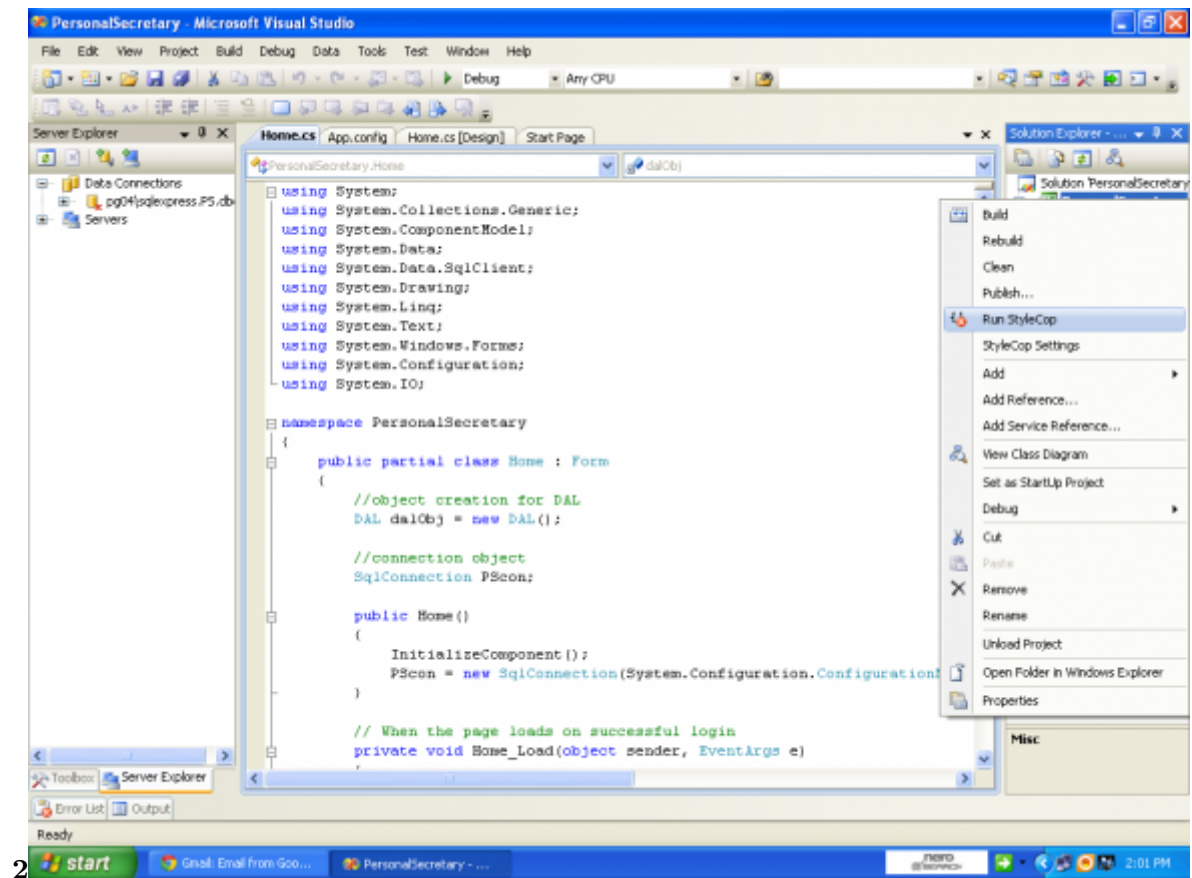


Figure 3: Figure 2 :

Warnings	Category	Priority	File name	Line	Col
The field must have an access modifier	Maintainability	1	DAL.cs	16	1
names must start with a lower-case letter	Naming	2	DAL.cs	16	1
The call to InitializeComponent must begin with the 'this.' prefix to indicate that the item is a member of the class.	Readability	3	Login.cs	19	1
The file has no header, the header Xml is invalid, or the header is not located at the top of the file.	Spacing	4	Login.cs	1	1
The field must have a documentation header.	Ordering	5	Login.designer.cs	37	1
The body of the if statement must be wrapped in opening and closing curly brackets.	Layout	6	DAL.cs	32	1
The class must have a documentation header.	Documentation	7	DAL.cs	13	1

Figure 5: C

1

Figure 6: Table 1 :

2

Tool	Increase complexity	Increase usability
------	---------------------	--------------------

Figure 7: Table 2 :

.1 Acknowledgements

We sincerely thank staff and students of ME program of BSCOER College, Narhe, Pune, India for helping us in designing PS, installation of stylecop and analyzing PS using stylecop.

[Gomes et al.] *An overview on the Static Code Analysis approach in Software Development*, Ivo Gomes , Pedro Morgado , Tiago Gomes , Rodrigo Moreira .

[Microsoft StyleCop ()] <http://code.msdn.microsoft.com/sourceanalysis> *Microsoft StyleCop*, 2008.

[Black (2007)] 'SAMATE and Evaluating Static Analysis Tools'. Paul E Black . *ADA User Journal* September 2007. 28 (3) p. .

[Zoubi et al.] *Scanstud: A Methodology for Systematic, Finegrained Evaluation of Static Analysis Tools*, Q Zoubi , I Alsmadi , B Abul-Huda . Identifier:10.1109/CITS.2012.6220379PublicationYear:2012. Identifier:10.1109/ICSTW.2011.32. p. . (Digital Object) (Study The Impact of Improving Source Code on Software Metrics. Publication Year: 2011, Page(s)

[Pressman ()] *Software Engineering: A Practitioner's Approach*, R Pressman . 2005. McGraw-Hill.

[Kulkarni and Padmanabham ()] 'TEIM-The Evolved Integrated Model of SE and HCI'. R Kulkarni , P Padmanabham . *Validating Utility of TEIM: A Comparative Analysis*, Moritz Martin Johns, . R Jodeit 7, P Kulkarni, Padmanabham (ed.) 2012. 2 p. . (IJACSA, To be Page(s)

[Stylecop and Codeplex (2011)] *WebsiteAvailable at:http://style cop*, Stylecop , Codeplex . October 8, 2011.