Re-Evaluation of Visual Studio 2010 Add-ins for Coding Guidance

Pallavi S. Bangare, Pooja More, Sunil L. Bangare, Ashish Upadhye, Pooja Zambad

Abstract— ***The add-in tool offers developers new means to check code quality and view suggestions based on standards used in coding domain today, thus adding flexibility to the current Visual Studio IDE. Code review is an important phase in implementation which we plan/propose to automate. Visual Studio 2010 add-in is assurance for coding guidelines and to run the review tool so as to find out the guideline violations at the time of coding itself***

Keywords— ***Regular expressions, Visual Studio 2010 IDE, Add-in, Coding Guidelines.***

I. INTRODUCTION

Add-in is the supplemental programs that users can install to extend the capabilities of the software program. Add-ins are third party or community developed. Add-ins can increase memory, graphics or communications capabilities to a computer. Idea in Visual Studio 2010 is to list some common recurring review comments and to write a tool to automate review process which will detect the violation of any coding guidelines at the time of coding itself. So that the manual review process can concentrate on the high level design or optimization review. Managed add-ins are typically found in the following location on windows: C:\Users\{username}\Documents\Visual Studio {version}\Add-ins. COM-based add-ins can be installed anywhere as their directories are specified via the registry.

II. VISUAL STUDIO 2010

A. History of Visual Studio 2010

Microsoft released Visual Studio 2010, codenamed *Dev10*, and .NET Framework 4 on 4thApril 12, 2010. Visual Studio 2010 supports developing applications in Windows 7. It Supports IBM DB2 and Oracle databases, in addition to Server. It includes tools for debugging parallel applications. Microsoft Visual Studio 2010 is an integrated development environment (IDE). It is used to develop console and interface applications along with Windows Forms applications, web sites, and applications. It provides form designer, server explorer and so on for rapid designing and development of .net applications. The IDE has been written using the Windows Presentation Foundation (WPF), whereas the internals have been redesigned using Managed Extensibility Framework (MEF) that offers more extensibility points than previous versions of the IDE that enabled add-ins to modify the behaviour of the IDE[1].

Table1. Comparisons in VS 2010

Visual Studio 2010 product	Extensions	Test Impact Analysis
Express	No	No
Professional	Yes	No
Premium	Yes	Yes
Ultimate	Yes	Yes
Test Professional	No	Yes

B. History of Visual Studio Ultimate 2010

Visual Studio Ultimate 2010 is codenamed *Rosario*. Visual Studio Ultimate 2010 also includes *Test Impact Analysis* which provides hints on which test cases are impacted by modifications to the source code, without actually running the test cases. Visual Studio Ultimate 2010 architecture is as shown below [1].

System architecture diagram

Figure 1 is the high level System architecture diagram. There are three inputs like

1. Code C#/C++
2. Visual Studio Extensibility Component
3. Set of Guidelines.

And one output is GUI displaying warning/suggestions.
Code can be written in C#/C++ in code model. Visual studio Extensibility model allows us to navigate through the code model. Using this code model we can iterate code items such as namespaces, classes, functions etc. Set of coding guidelines which want to add will store in XML file (XML database). Set of coding guidelines can review code while user writing code as well as after finishing implementation part. After compilation of code it will display warnings/suggestions. This information would be easier to improve code quality and code standards.

Manuscript published on 30 March 2013.
*****Correspondence Author(s)
Pallavi S. Bangare, Assistant Professor, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, University of Pune, India.
Sunil L. Bangare, Assistant Professor, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, University of Pune, India.
Pooja More, B.E.Student, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, University of Pune, India.
Ashish Upadhye, B.E.Student, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, University of Pune, India.
Pooja Zambad, B.E.Student, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, University of Pune, India.

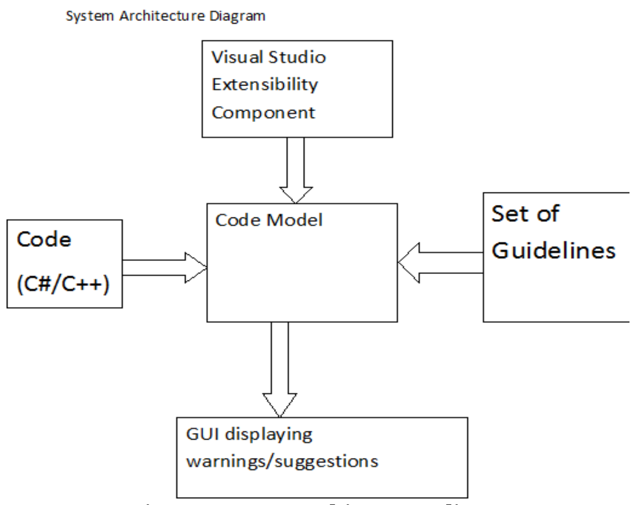

System Architecture Diagram
Visual Studio Extensibility Component
Code Model
Code (C#/C++)
Set of Guidelines
GUI displaying warnings/suggestions

Fig. 1 System Architecture diagram

C. Some available useful Add-ins

1) Productivity Power Tools

- Allows you to control+click on an item to navigate to its definition.
- Adds Solution Navigator (similar to Solution Explorer) to navigate through files, methods, variables, etc.
- HTML Copy – Allows you to copy text from the editor and paste while maintaining the style of the code.
- Add column guides by right clicking on any line in the code and using the appropriate context menu.
- Move entire lines of code up and down by pressing alt + up arrow/down arrow [6].

2) GhostDoc

- Allows you to document classes, methods, variables by right clicking on the item and selecting "Document this". Base documentation is created for you based off of the name of the method, variables, class, etc. GhostDoc is very handy for creating documentation quickly when you are already naming items descriptively [6].

3) PowerCommands for Visual Studio 2010

PowerCommands provides some functionality in VS 2010 that is much needed and surprisingly not present in the IDE by default.

- Copy and paste project references
- Open Command Prompt at directory of items folder
- Open Containing Folder
- Right Click -> Close ALL from the document tab [6].

C. Visual Studio 2010Add-ins

1. Create Add-in

Visual Studio 2010 Add-in follows below steps to create Visual Studio extensibility project and overall details of implementation.

From Visual Studio click on New project -> Other Project Type -> Extensibility From this select Visual Studio Add-in.

Once we select this project type, follow steps as mentioned in GUI. Select "Create Add-in using C#. De-select Macros. (Macros functionality would not be implemented in this project). Once we create this project type, we get already implemented class which is inherited from IDTExtensibility2.

This class has application object of type DTE2 which helps to retrieve all necessary information. All retrieved information will be stored in internal data structure which will be used to check for guidelines violations. The rules which will be input from user will be stored in separate xml file so that user doesn't need to input this information every time. Entire coding will be performed using C# language.

2. Overview of Visual Studio 2010 Add-in

1. IDE Add-ins

a) Source control support

- AnkhSVN – Provides a free Subversion client for Visual Studio
- VisualSVN - Subversion integration for Visual Studio 2003/2005/2008/2010.
- VsTortoise - A free TortoiseSVN add-in for Microsoft Visual Studio 2005/2008/2010

b) Refactoring and productivity

- CodeRush - Refactoring and productivity plugin
- ReSharper - Refactoring support for .NET languages[4]
- Visual Assist X - Productivity plugin, like Resharper. Notable for C++ support
- JustCode - Refactoring and code analysis productivity plugin

c) Other

- CodeIt.Right combines static code analysis and automatic refactoring for Microsoft Visual Studio 2012, 2010, 2008, 2005 and 2003.
- PVS-Studio - Static Code Analyzer for C/C++/C++11. Supports Visual Studio 2005/2008/2010.
- Designbox - Adds a toolbox that lets you associate initial property values with controls
- Gemini - Gemini issue tracking plugin for Visual Studio 2008/2010
- Help Generator – Generates HTML Help topic files and screenshots of your application.
- Koders – Adds a search plug-in to search the Koders database
- Reflector - a code browsing utility
- Dotfuscator – Provides tools to help prevent reverse engineering
- VSdocman - Visual Studio code commenter (XML doc comments) and API documentation generator. For VS 2010/2008/2005/2003/2002.
- XMLSpy – Integrates the XMLSpy IDE into Visual Studio
- Liquid XML Studio – Integrates Liquid XML Tools: XML Schema Editor, WSDL Editor, XPath Expression Builder, and Web Services Test Client into Visual Studio 2005/2008/2010.
- MPCL - Enable/Disable parallel compilation for Microsoft Visual C++ 2005 and Microsoft Visual C++ 2008 [1].

III. AVAILABLE CODING GUIDELINES

CodeIt.Right is a new generation code analysis tool that combines static code analysis and automatic refactoring to best practices in one application. But there is more CodeIt.Right will *automatically* correct code errors and violations.[5]

List of coding guidelines as Follow:

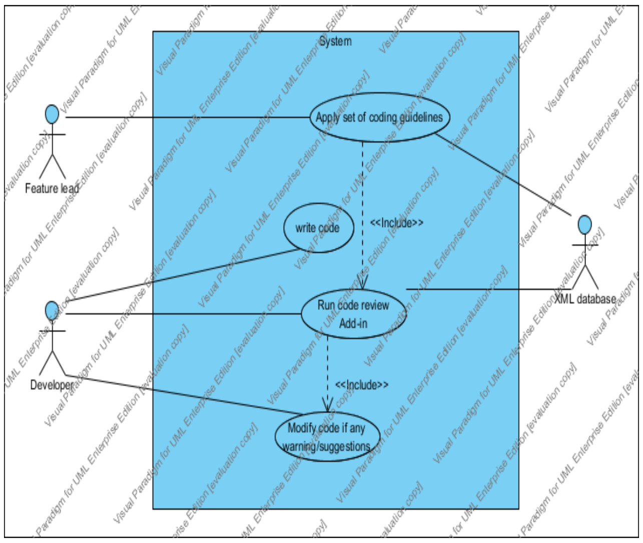
1. Capitalisation Styles
   1.1. Pascal Case
   1.2. Camel Case
   1.3. Uppercase
2. Case Sensitivity (not applicable to VB)
3. Abbreviations
4. Word Choice
5. Avoid Type Name Confusion
6. Namespace Naming Guidelines
7. Class Naming Guidelines
8. Interface Naming Guidelines
9. Attribute Naming Guidelines
10. Enumeration Type Naming Guidelines
11. Static Field Naming Guidelines
12. Parameter Naming Guidelines
13. Method Naming Guidelines
14. Property Naming Guidelines
15. Event Naming Guidelines
16. Control Naming Guidelines
17. Specifying Particular Control Variants
18. Table of Standard Control Prefixes
19. Menu Controls
20. Data Naming Guidelines[5]

IV. NEW PROPOSED GUIDELINES

As mentioned above, Code review is the most important phase in implementation. This helps to find out the code bugs and helps to improve the code quality. Developers tend to do mistakes while writing code and sometimes those mistakes are common and repetitive.In Visual Studio Ultimate 2010, we are going to insert following coding guidelines in new Add-in.

1. Detect deep nesting and warn to refractor the code.
2. Detect exception catch body which does nothing.
3. Detect the use of generic exceptions say catch (Exception) and warn to use specific exception. In case of C^{++} detect catch (…).
4. Suggest refactoring of the code if number of lines in function exceeds a given number say 100.
5. Don’t use magic numbers.
6. Try to limit the number of parameters send to a function.
7. Initialize static class members with static constructors.
8. Prefer foreach loop instead of for loop.
9. Always use properties instead of accessible data members.
10. Try to layout the class structure. Like ordering of Private Members, Public, Functions etc.
11. Use comments on all Public functions or public properties.
12. Accepts IEnumerable<T> or ICollection<T> instead of a concrete collection class.
13. Define private assignment operator if not already defined.
14. Detect empty string literals "" and suggest using String.Empty.
15. Prefer Variable Initializers to Assignment Statements[3][4]..

V. USE CASE DIAGRAM FOR PROPOSED SYSTEM



A use case defines a set of use-case instances. An actor communicates with a use-case instance of the system. The functionality of a system is defined by different use cases, each of which represents a specific flow of events. The description of a use case defines what happens in the system when the use case is performed.

VI. CONCLUSION

This paper gives details of various add-ins and some new coding guidelines. The proposed Add-in automates Code review hence helps to find out the code bugs and helps to improve the code quality. This will provide developers to follow a new set of rules which will help them in becoming better in Visual Studio with respect to standard coding conventions today.

VII. ACKNOWLEDGMENT

This work is supported by S.T.E.S.’s Sinhgad Academy of Engineering. We are thankful to Dr. A. G. Kharat, Principal, Col. Prof. O. P. Misra & Dr. C. B. Bangal, Vice Principal, Prof. A. N. Adapanawar, HOD-IT & Prof. Manoj L. Bangare, SKNCOE, Pune for their support and guidance.

REFERENCES

1. www.microsoft.com/visualstudio.
2. Static Analysis Tools For .NET.
3. Framework Design Guidelines: Conventions, Idioms, and Patterns For Reusable .Net Libraries - Krzysztof Cwalina, Brad Abrams.
4. Effective C#: 50 Specific Ways To Improve Your C# by Bill Wagner.
5. www.submain.com.
6. http://visualstudiogallery.msdn.microsoft.com/d0d33361-18e2-46c0-8ff2-4adea1e34fef

AUTHOR PROFILE

Pallavi S. Bangare, Assistant Professor, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, India, ME [IT], BE [E&TC], PGDC-DAC, Microsoft Certified Technology Specialist (MCTS). Associate member of CSI. Published 11 papers in various International Journals & Conferences. Also published 3 books for Information Technology & Computer Engineering.

Sunil L. Bangare, Assistant Professor, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, India, M. Tech [IT], BE [E&TC], PGDC-DAC, Life member ISTE, Associate member of CSI & AMIE. Published 11 papers in various International Journals & Conferences. Also published 3 books for Information Technology, Computer Engineering & Mechanical Engineering.

Pooja More, B. E. Information Technology student, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, India

Ashish Upadhye, B. E. Information Technology student, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, India

Pooja Zambad, B. E. Information Technology student, Department of Information Technology, S.T.E.S.'s Sinhgad Academy of Engineering, Pune, India.