

Study Of Feature Driven Development Using The Concepts Of Object Oriented Programming System

Raj Kumari

University of Engineering And Technology
Punjab University
Chandigarh, India

Heena

University Of Engineering And Technology
Punjab University
Chandigarh, India

Abstract

Feature Driven Development is one of the finest methodologies which come under the agile development pattern which is getting used in the companies rapidly. Feature Driven Development (FDD) is nothing but just a methodology in which we check out that whether the software is performing its operations according to the client environment or not. The parameters for which we are considering our work are as follows:

a) LOC

b) RFC – response for class

c) Dead codes and functions

We will be implementing the above parameters on a open source software with FDD.

1. Introduction

All **Software Development Models** provide basis for the development of software projects and represents the whole development process of the software. Various development models are available for development of software and different software models are adopted for different software [3]. Wide ranges of methodologies are available such as waterfall model, Rational Unified Process (RUP) model, Extreme Programming (XP), Scrum, Spiral, Feature Driven Development (FDD).[4]

2. Object Oriented Programming System

Object oriented programming is a programming language model that represents concepts as objects and have attributes that describe the object and associated events known as methods. Object oriented

programming is frequently used in software today. One of the reason that OOP's came to existence was to increase reusability and maintainability of code. Object

is an instance of class. Applications are designed on the basis of the interaction between objects. C++ and Java, C#, Visual Basic.NET, Python and are popular OOP languages.

Basic Principles of Object Oriented Programming:

- 1) Encapsulation
- 2) Polymorphism
- 3) Inheritance

1) Encapsulation:

Encapsulation means creating modules which bind data to functions. Its main aim is to prevent unauthorized use of data by hiding the variables inside the class. Internal representation is hidden from outside view.

2) Polymorphism:

Polymorphism means creating variable or function having more than one form. The principle of polymorphism is to program in such a way that objects of different types define a common interface of operations for users.

3) Inheritance:

Inheritance provides relationship between objects. It is a way of reusing the code. It helps to reduce the code and hence reducing lines of code in an software.[4]

3. Feature Driven Development (FDD)

The *A feature*, in software terms is referred to client-valued functions which describes any capability which is valuable for software. Features are generally short so they can be implemented in two weeks or even less [1]. Functionality has basic unit called feature which is developed by feature team and delivered to clients. [2] They are planned and developed incrementally thus producing substantial and firm results. Feature

templates are used to describe feature list. Format of template is as under:

<action> the <result><by | for | of | to>a(n)<object>

E.g. compute the total of a sale, authenticate the password of the user, cite the spellchecker for a text, improve the quality of a service and many more. Some features can be built-in into single class, such as the foremost and the third one. While some features cannot be included such as the third one which requires minimum two classes the spell-checker and the text. [1]

Definition of **feature** in IEEE 829 is "A distinguishing characteristic of a software item (e.g., performance, portability, or functionality)."

Feature-driven development (FDD) is one of the agile methodology that is used to develop a software. Its foremost function is to deliver tangible, running software continually in timely manner [5]. Feature is the essential element in FDD.[6] In FDD industry-recognized best practices all determined from a client-valued functionality (feature) point of view. It provides an iterative and incremental development process of a software.[5]

Feature Driven Development contains five steps. They are as follows:

- 1) Develop an Overall Model.
- 2) Build a Features List.
- 3) Plan by Feature.
- 4) Design by Feature.
- 5) Build by Feature.

Description of these steps:

1) Develop an Overall Model:

This phase is also known as domain modeling. In this phase classes in a domain, their connection and interaction are represented by UML diagrams. The

necessary methods and attributes are positioned in the classes. The model is modernized with new methods, attributes and relations for each user story that extends the present model.

2) Build a feature list:

In this phase the list of features is prepared according to priority. To prepare this feature list, feature team should have basic understanding of the project. Features should be generally short if they are large then they

should be further broke down into small features such that they can be completed in two weeks or less.

3) Plan by feature:

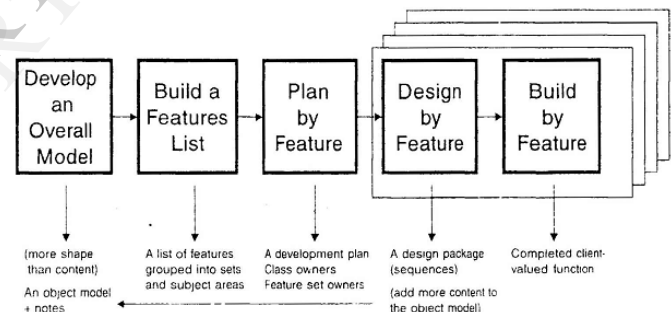
After the feature list is being prepared next step is plan by feature. In this phase implementation order of feature is decided. This order depends upon certain factors such as features dependency, its complexity. In this phase, developers are assigned classes for its development and maintaince.

4) Design By Feature:

In this phase feature set is prepared. Chief programmer selects the features to be implemented. Activities are performed for each feature. Design inspection is performed in this phase.

5) Build by feature list:

After the features are selected and being inspected next step is to implement the features being selected. Their respective code are developed and tested. To check whether it meets all requirements or not code inspection is performed. [5]



FDD Process[5]

4. Feature Driven Development with OOPS

Feature Driven Development is a way to design software. It basically emphasis on features which is client value functionality. Feature can be calculate total sale etc.

LOC in software terns represents lines of code which is used to design program thus provide functionality to any application. Classes , functions in software are basically code. More lines of code (LOC) code can make things worse. More code implies more problems and effort. Development groups make every effort to reduce their LOC while creating new features rather than increasing it.[2]

Each and every process takes a memory when it is executed in the system. When so ever another process is attached within the same system , it is required that the memory cycle should rotate so that we can keep the memory cycle active and no deadlock occurs . But dead locks occur often when we implement the software in real time . A live example for this is google chrome browser .We often see that when a number of tabs are open in the browser, it gets hanged .The reason behind this is that it sometimes fails to rotate the process and the memory is stucked.

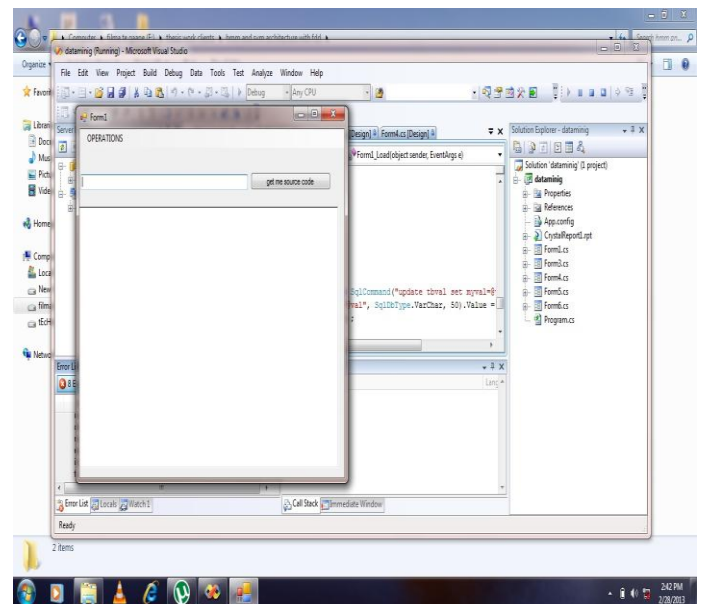
Dead Code and Functions are functions which are not currently used by application but still occupying memory.

Releasing memory at runtime helps to save memory. As memory is released at runtime and each time dead code and functions varies and memory get refreshed so memory released varies.

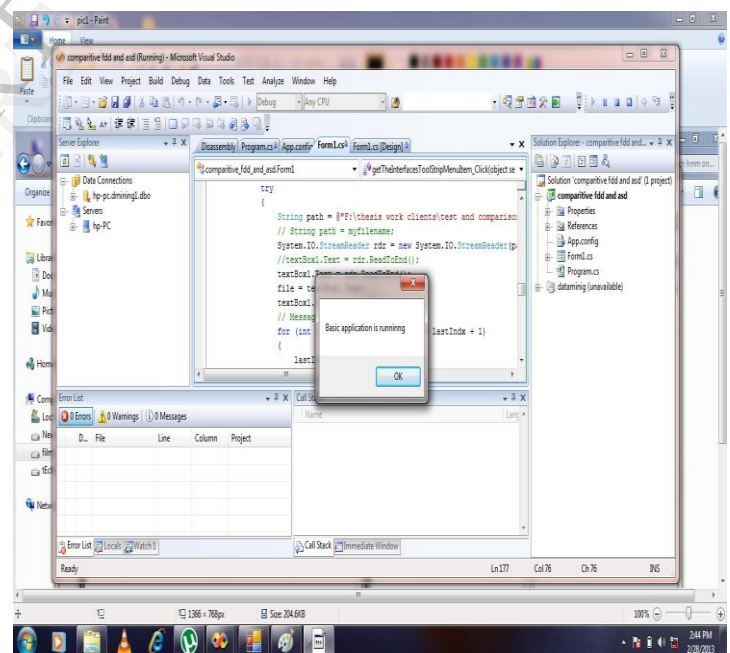
Open source software using FDD was designed in .Net. Basic software(designed in.NET) is designed whose parameters were calculated using the main software. Basic software takes the link of any news website and extract its code. After refinement of code ,processing of news is done to check to which category this news belong. Forms i.e cs files of this basic software are used by main software.

Two software are connected with each other. If our basic application runs then main software will run.

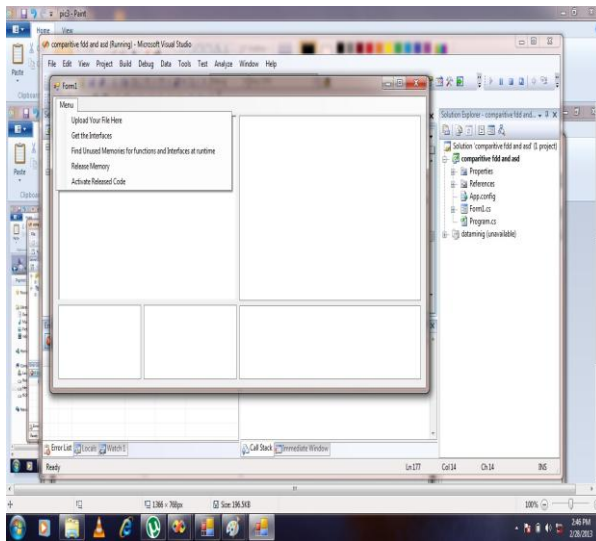
5. Screenshots



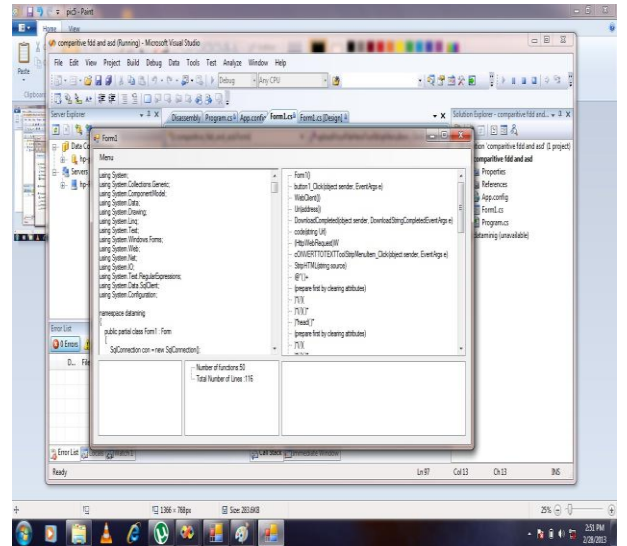
1) Basic software



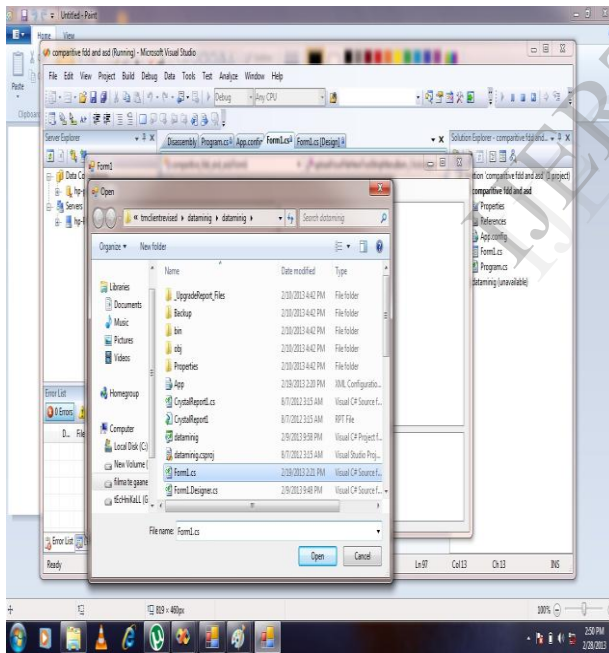
2) Main software



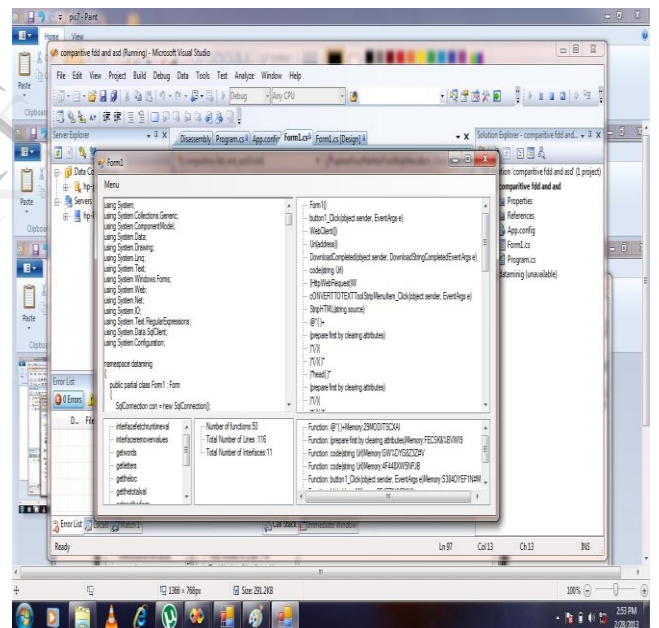
3) Menu with Features.



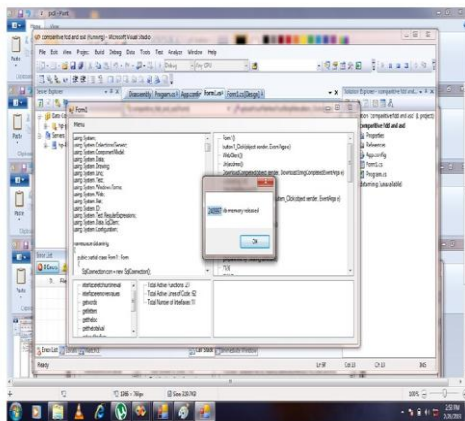
5) Source code of form1 is displayed. Lines of code and number of functions of this form are calculated and displayed in this window.



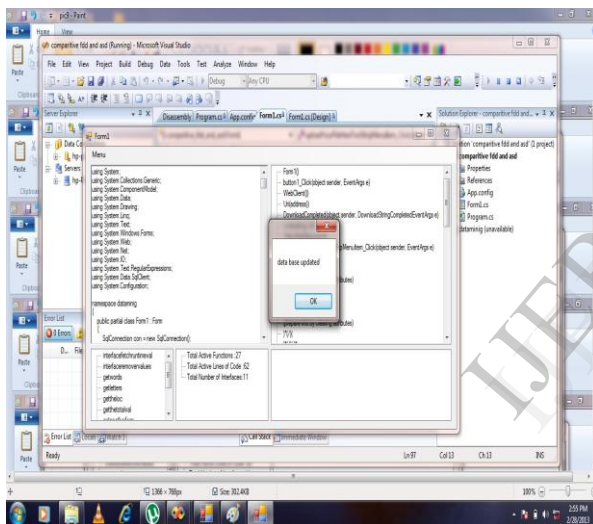
4) cs files of basic software



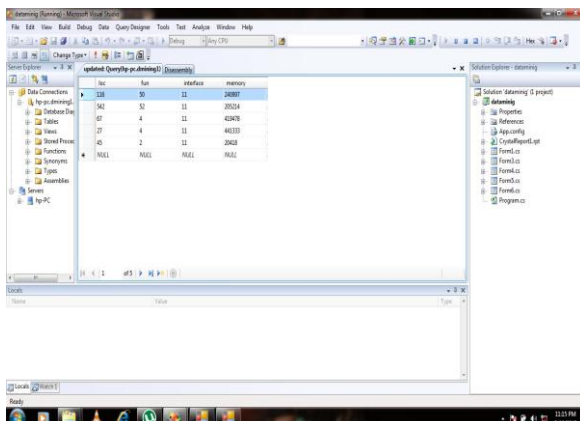
6) Total number of interfaces in basic software are calculated.



7) Active functions and respected lines of code are calculated and unused memory of dead functions is released at runtime and displayed.



8) Information is updated in database.



9) Database updated with the information of parameters using cs files of basic software.

6. Conclusion and Future work

Feature driven development is an agile methodology which is used to design software. It mainly focuses on features. We have shown LOC, dead code functions and memory concept with software using environment of FDD. We can also show similar using other agile methodology such as Adaptive Software Development(ASD).

7. References

- [1] Jianxiong Pang Lynne Blair, "Refining Feature Driven Development - A methodology for early aspects" Computing Department, Lancaster University, Bailrigg, Lancaster, LA1 4YR, U.K.
- [2] Larman & Vodde, Addison-Wesley, *Excerpt from Scaling Lean & Agile Development: Successful Large, Multisite & Offshore Products with Large-Scale Scrum*, 2008
- [3] Lei Wang, "Research of Software Development Model Based on the Theory of Software Engineering", Department of Computing, The Chinese People's Armed Police Forces Academy, Langfang, 065000, China, 2nd International Conference on Computer and Information Application (ICCIA 2012)
- [4] www.wikipedia.com
- [5] Sertac Ozercan, "Adapting Feature-Driven Software Development Methodology to Design and Develop Educational Games in 3-D Virtual worlds".