

An Hybrid Approach and Enhanced Analytic Tool for Global Software Development

S. Hariharan¹, A. Renga Rajan² and R. Prem Kumar³

ABSTRACT

Requirements Engineering is, as its name suggests, the engineering discipline of establishing user requirements and specifying software systems. There are many definitions of Requirements Engineering. however, they all share the idea that *requirements* involves finding out what people want from a computer system, and understanding what their needs mean in terms of design. Requirements Engineering is closely related to software engineering, which focuses more on the process of designing the system that users want. Perhaps the most concise summary comes from Barry Boehm: requirements are “designing the right thing” as opposed to software engineering’s “designing the thing right” (Boehm, 1981). Requirement engineering is the important phase in the software development. Here, we gather all requirements of the software which is proposed for development. An agile methodology produces high quality software and takes less time in comparison to traditional methods .Agile was discovered for managing the development process in the environment where requirement could be changed during development process. Requirement engineering is the important phase in the agile development methodology. In this paper, the software development tool which is described, analysis all the life cycle of a software and intended to produce a quality software.

Terms: Requirement engineering, Agile methodology, JAD, Viewpoint,SDLC

1. INTRODUCTION

Agile software development methods have become more famous [1]. A number of methods have been developed for delivering the software on time and fulfill all the requirements of the customer. All these approaches have some shared philosophy: Improved customer satisfaction, implementing the varying requirements, frequently delivering the working software, and tight association of customer and developers. [2]

On the other hand, Requirements engineering (RE) is a software engineering process aimed to identify the requirements, analyze the requirements, documentation of requirements and validate the requirements for the proposed one[3]. Often, requirements engineering and agile development are not matched easily: RE depends upon the documentation while agile development process have customer and developer collaboration to develop the system. In RE, we have JAD technique for gathering the requirements .In JAD session developer, customer and stakeholders discussed about desired product requirement. This kind of session is very productive in terms of requirement engineering. The aim of this paper is to find out a hybrid approach to serve the purpose of agile requirements. For prioritization of the gathered requirements for the incremental delivery of the software is done through the viewpoint approach which is also a requirement approach. Viewpoints are used for getting the viewpoint of system in the terms of customer. So it will be helpful to decide which requirements are necessary to focus first. The next section of this paper briefly

¹ Research Scholar, Saveetha University, Department of CSE, Chennai, Tamilnadu, India

² Professor, Vel Multi Tech, Department of CSE, Chennai, Tamilnadu, India

³ Research Scholar, Saveetha University, Department of CSE, Chennai, Tamilnadu, India

E-mail: hari2418@gmail.com

*Corresponding author: Hariharan S.

gives an overview of the related work which has done in the field of agile requirement [4]. In Section 3 of this paper, we propose a model for RE in agile with help of JAD. In section 4, we examine the result of this approach and discuss the benefits of the model. The last section summarizes the model and result.

Agile principles

The Agile Manifesto is based on 12 principles

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstance

2. OVERVIEW

James E. Tomayko explains that the later stages of software development. An agile method provides a better way to deal with these unstable requirements. He suggested two methods for requirement elicitation, user stories and prototyping. The main difficulty in agile environment is cost estimation. Cost could be estimated for current cycle only while cost involved with unknown requirement is difficult to measure. Agile has its own limitation and dependence on test-first programming. It is a basic principle of the extreme programming. It says that early detection of minor errors and detection of defects at the time of integration. [2] [3]

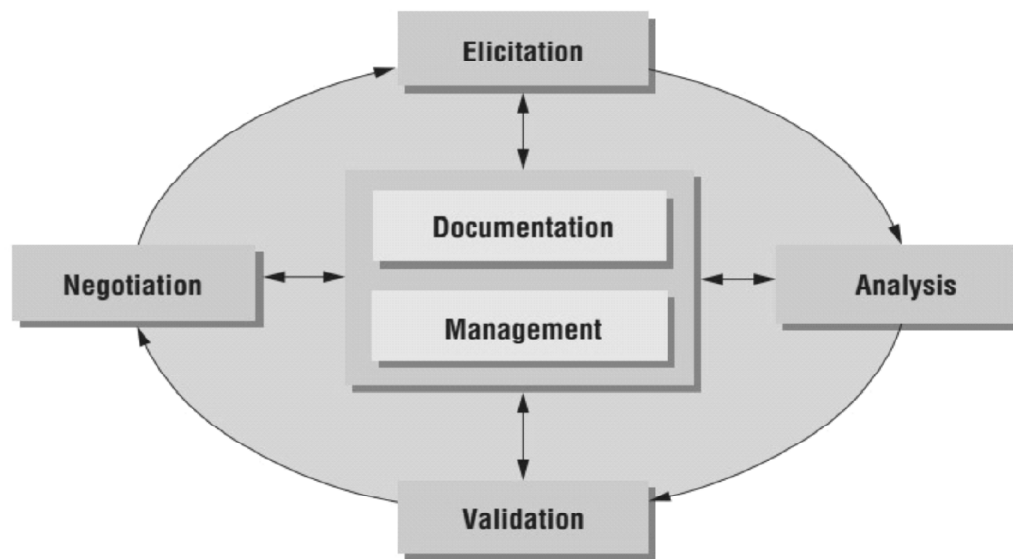
Agile techniques rely on minimum documentation but documentation is very important part in maintenance phase so it is responsibility of development team to ensure proper documentation. [4]

Lan Cao and Balasubramaniam Ramesh collected data from 16 organization to perform a study that how agile methodology differ from traditional methodology. Most of the organization that work on agile methodology reported that the main challenge in agile is to maintain a common consensus between various customer groups. They explore test driven development as least used RE approach between organizations. The study identify the most important requirement engineering practices in agile is thorough communication between developer and customer. Instead of using formal specification procedure like in traditional approach, agile requirement engineering is adaptive and dynamic in nature. [5]

Zornitza Racheva, Maya Daneva has focused a most important mechanism of agile that is known as Re- prioritization which helps to accommodate unstable requirements. They derive a conceptual model on the basis of multiple case studies. This model helps for understanding the iteration prioritization process. This model gives answer to the question that, what are the main concepts when consider requirement prioritization process. [6]

Process Map of Requirements Engineering

Requirements engineering [1, 38] is the most important activity in software project development as the other phases in the life cycle of software development depends on this important activity. As the name implies requirements engineering is a dig field responsible to cover all the activities involved in discovering, documenting, and maintaining a set of requirements for a computer based systems. A numbers of consequences may arise due to wrong requirements such as the system may be delivered late, more costly than the original estimation, customer and end-user will not be satisfied, the system may be unreliable and there may be regular system defects. According to the survey conducted by ESPI in 1995 that about 40-60% of all defects found in a software project can be traced back to errors made during the requirements stage.



3. PROPOSED WORK

Agile methodology produces high quality software and takes less time in comparison to traditional methods. It was discovered for managing the development process in the environment where requirement could be changed during development process. In traditional methods it is compulsory to fix all requirements before starting development phase while agile removes these barriers. It is not compulsory to lock all the requirement and design before going to development phase.

The agile methodologies follow the following methods:

Table
Methods and there description

<i>Methods</i>	<i>Description</i>
<i>Iterative Development</i>	Agile method is lightweight process and it relies on iterative cycles that are short in nature.
<i>Increments of work</i>	Incremental approach provides a way for getting feedback from stakeholder about the actual requirement product.
<i>Collaboration</i>	It believes in assemble customer and development team at one platform. So that they work together
<i>Adaptation</i>	The customer and development team review the requirement elicited and trace the progress of requirement and find the ambiguous requirement to fix it.
<i>Emergence</i>	The team structure and requirement are not fixed at the early phase. It emerges with the progress of the development. Team starts his work with preliminary requirements.

We are here giving the idea for a variant of agile which is SCRUM.

The Scrum methodology is a part of agile methodology [8] . It is designed to handle with environment where requirement could change with time. Scrum methodology project progress forward on the basis of interaction among team members and further divide work in to small modules that is known as “sprints”, with each sprints has limitation of 2 to 4 weeks. Scrum methodology mainly concentrates on management of progress of projects in comparison to how to improve coding techniques. This methodology mainly applied on small projects. Small project can be handled by individual teams while large projects could be divided into subproject and each subproject handled by a scrum team. Communication and priorities between subproject team are managed with the help of scrum technique.

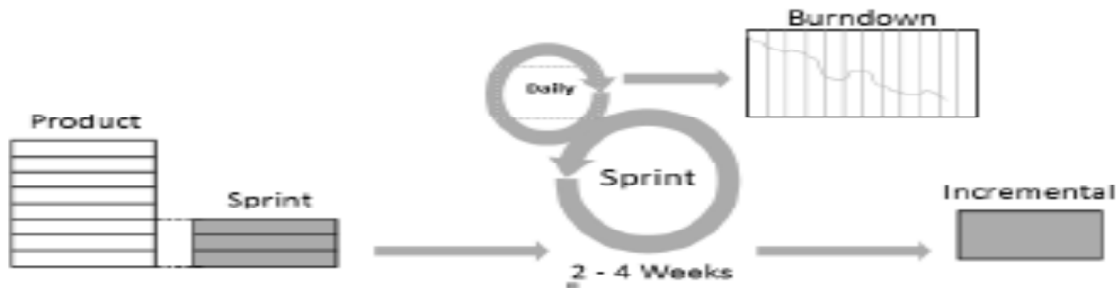


Figure 1: Scrum model

Now, we look how JAD and viewpoint may be implemented in agile software development.

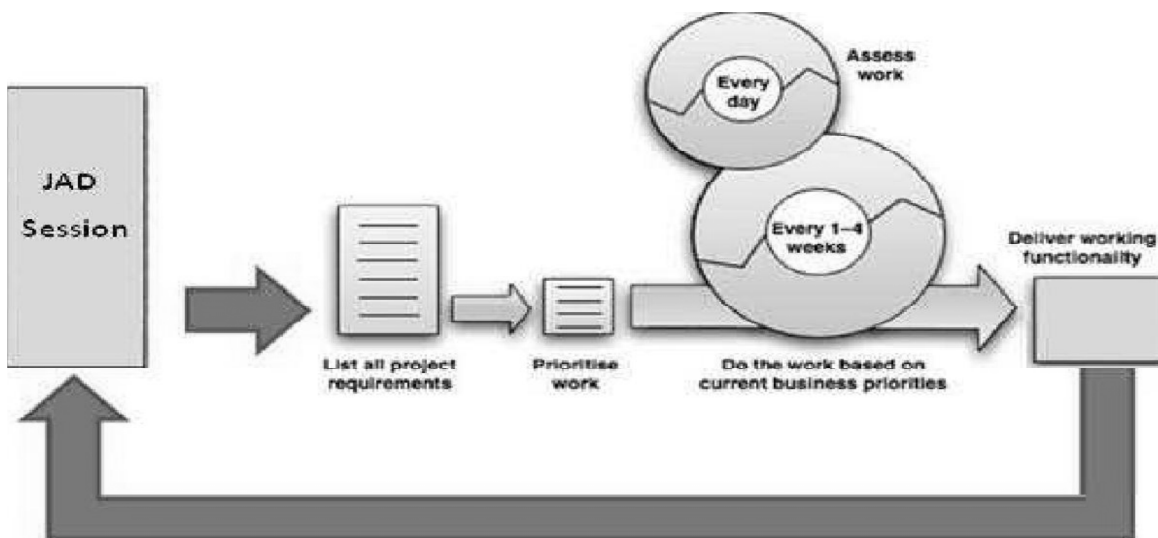
Joint Application Development for requirement engineering:

The JAD facilitator and project manager conduct a workshop for all executive sponsor and other stakeholders where they will their own expectations about the system.

Gathering Requirements through sessions

Extraction of requirement from customers is an iterative process. The fundamental idea behind JAD session is to produce detailed requirements on which JAD group members agree completely As a result of JAD session, all documented requirements are describing the requirement thoroughly.

This document contains use cases as well as business models which describe how the customer could interact with the system.



Session for Requirements Analysis

The gathering process assembles the requirements from various sources while requirement analysis process examines the requirements whether they are complete or not.

JAD requirement analysis performs by various steps:

1. Design a Higher order Activity plan
2. Modeling of requirement on basis of use cases by recognizing the following phases:
 - a) Construct documentation for the software requirements by identifying the main flows, business rules, exception flows, alternate flows as well as all other design constraints.
 - b) Set priority sequencing for the use cases and requirement lists. These items will be used in iterative release plan later.
3. Generate the State Diagrams and requirements Storyboard for Graphical User Interface design.
4. Establish the business objects and discover their relationships among the use cases. The process of building the Domain Object Model is described below:
 - a) Recognize the “real world” objects first.
 - b) Identify the associations of those objects from other objects
 - c) Identify the multiplicity factor between them.
 - d) Finally publish the List of attributes and couple the operation with each object.
5. Develop a dictionary of all definitions like use cases, actors and business objects etc.

JAD group performs the requirement analysis step based on collected requirement from user interaction. These groups transform the requirements into high level solution which will assist in development phase. Groups include the business subject, matter expertise as well as a complete technical team [9].

The levels of work taken under Focus Group Analysis Session is described as follows:

1. Each use cases with detailed information generated in the requirement phase will be provided. Every use cases must describe the interaction process between user and system. These use cases should contain proper paper prototypes and associated screenshot
2. All objects, used in the use cases, must be represented with use case scenario.
3. All workflows must be shown by the sequence diagram.

These types of focus groups allow shared requirements. Output of these sessions produces highly refined requirements for development process in a short and certain span of time.

All requirements would be documented for every aspect including management perspective and also tried to discover the uncovered requirements by changing the participant in the JAD sessions. Customer will be directly involved in the requirements gathering process since it is also calculated with the team score.

2. Product backlog by JAD: Product backlog is created by gather all requirements using JAD output. So all known requirement is now collected in productbacklog[10].

3. Prioritization of requirement: Now all requirements are priorities using VIEWPOINTS with the concern of stakeholder’s representative who is associated with development process in scrum. [11] [12]. Viewpoint Requirement engineering is a process where we elicit, model and validate requirement [13]. In this process we consider different viewpoint, which helps us to identify requirements from various

perspective. A view point approach is based on the assumption that the entire system requirement cannot be covered by taking single perspective of the system.

A viewpoint extracts information from stakeholder which is to the interest of stakeholder. Many view point—oriented approaches predefine the configuration of viewpoint before their process.

[14] On the basis of viewpoint based methods that have been developed, there are two main types of viewpoints:

Viewpoints related to system stakeholders. A stakeholder could be anyone who is affected directly or indirectly by the presence of system. System stakeholder can include end users, manager of organization, all other external entities that are somehow affected by system and the other entire technical person who are responsible for development and maintenance of the system.

Viewpoints related to organizational and domain knowledge. It is the knowledge about constraints associated with the system. The constraints may be organizational, physical, human related. Legal constraints, regulation and standard related constraints are also associated with system.

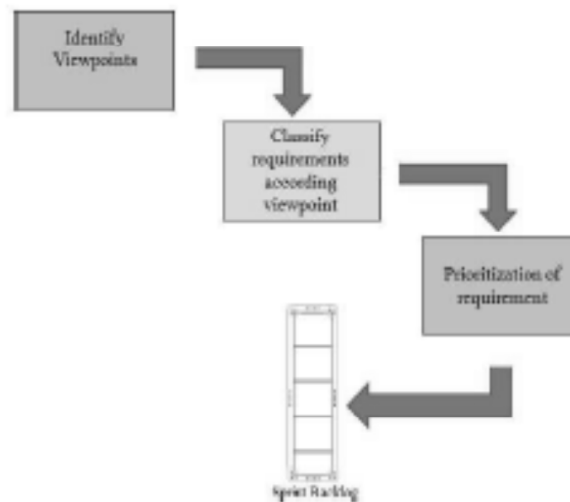


Figure 3: Steps to making print Backlog

Viewpoint gives the ideas that which requirements from product backlog are of higher priority and required to deliver with the current increment.

[15] The steps which are associated with the prioritization of product backlog for creating the sprint backlog [16][17]. These steps are shown in fig. 4

- a. Identify the possible viewpoints for the system. There could be stakeholder's viewpoint, manager's viewpoint and all other possible view point of that system.
- b. Next step is to classify among the all requirements specified by JAD. Categorize the requirement according to various viewpoint identified in first step and put it in the category where it is most important.
- c. Now identify that which viewpoint is most important and prioritize viewpoint. And next is to prioritize the requirement in every viewpoint.
- d. In sprint backlog we will take requirements from the viewpoint which is of higher importance in system. We will work on the requirements that belong to most critical viewpoint.

So the decision of increment is associated with viewpoint. Now, we get the sprint backlog which serves the back for incremental approach of the further development.

4. Delivery in increments: increments are verified using viewpoint which was selected by the SCRUM manager. These delivered increments are verified by viewpoint after the release with the help of stakeholder's representative.

5. Repetition for further increments: Step 1 to steps 4 are repeated up to full software development is not done. Next increment is targeted through the rest unfulfilled requirements available in the product backlog .Here, if all the requirements are not fulfilled then again arrange a JAD session. These requirements are priorities through the viewpoints, means we will find out the next important aspect of the system.

4. Result and Discursion: Requirements are most sensitive phase in agile development process. All requirements are gathered through arranging the JAD session. Joint Application Development (JAD) focuses on group session (or workshop) that follows structured analysis approach. So after this, we get all the requirements of the software .so this provide a systematic way to get all the requirements using interaction with the stakeholders. This is suitable model to use the requirement gathering approach in the agile development. Now the another main task is develop the prioritized sprint backlog which is handled through the viewpoint .Viewpoint gives the idea to focus on the requirements which are important for the increments. Therefore, this hybrid approach gives an effective way to handle the issues of the agile development 's requirements gathering.

5. Conclusion: JAD and view point serve as the key tool for agile software requirement gathering. JAD gives the total customer satisfied requirements which are decided in the JAD session. So it supports the feature of agile. Customer is also in the requirement elicitation process. Increments selection is very much tedious job which is very easily done by using VIEWPOINT. So this hybrid approach reduces the overhead of the scrum manager as well as we can strongly focus on the perfect incremental delivery as well as good quality software.

4. EXPERIMENTS

Our Tool sample screenshots are given in figure 4.1, 4.2 and 4.3 ,which produces quality software.

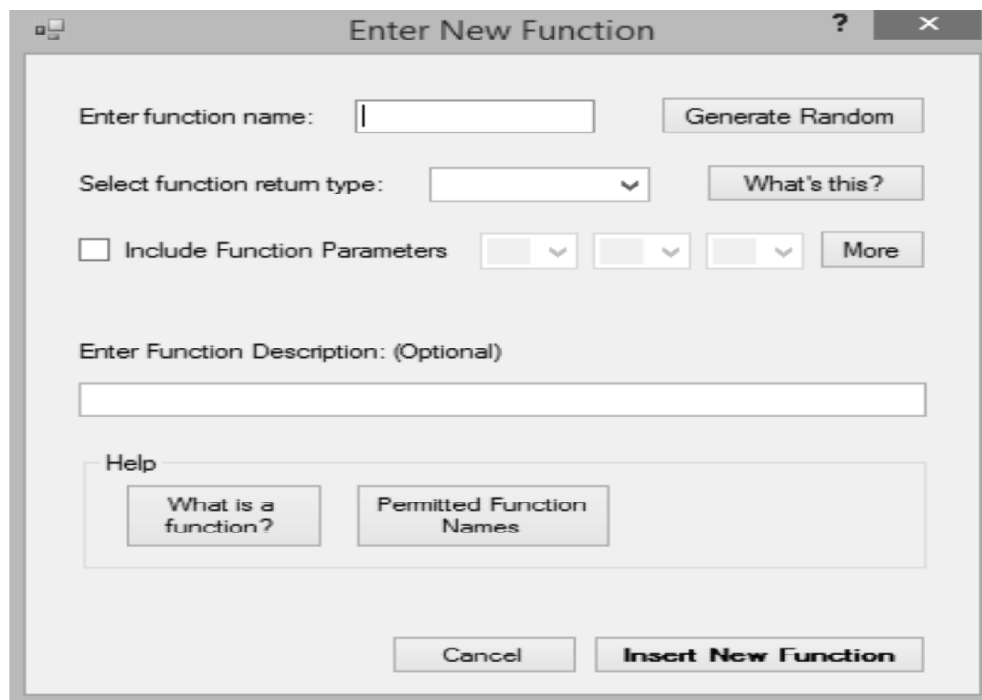


Figure 4.1

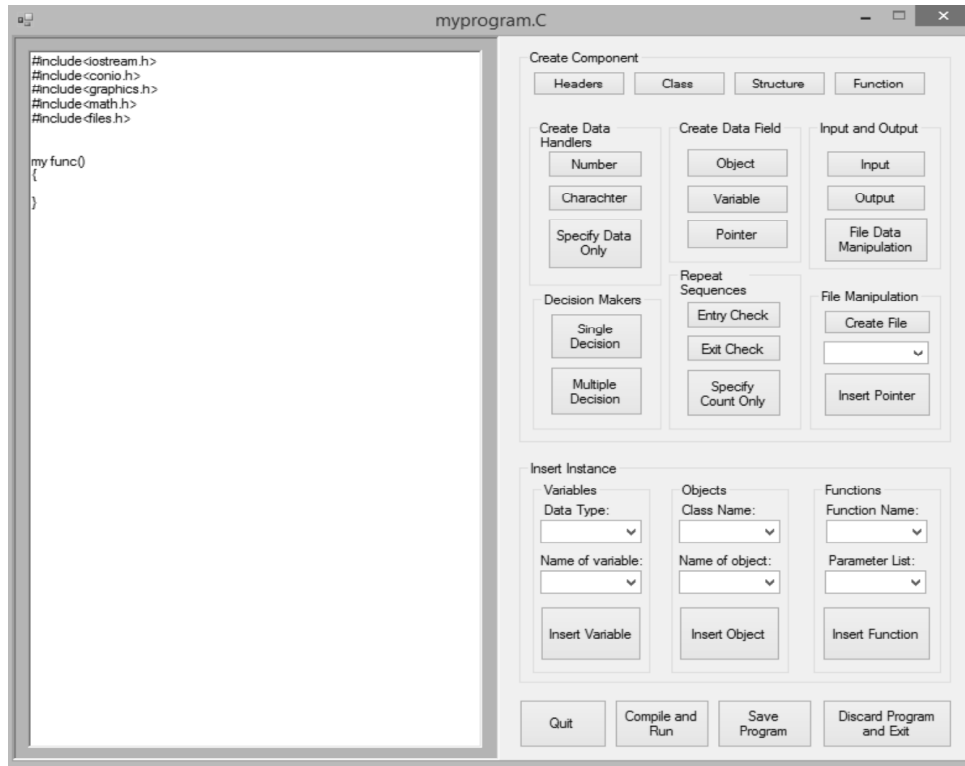


Figure 4.2

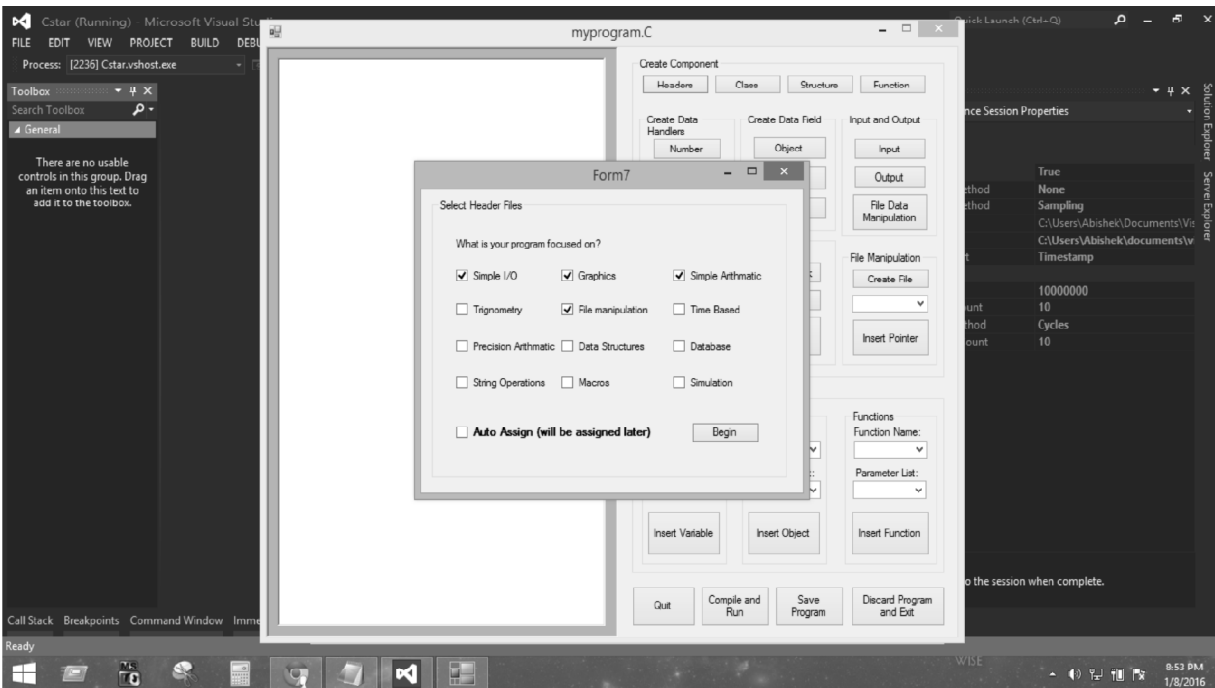


Figure 4.3

REFERENCES

- [1] Cao, Lan, and Balasubramaniam Ramesh. "Agile Software Development: Ad Hoc Practices or Sound Principles?." *IT Professional* 9.2 (2007): 41-47.
- [2] Tomayko, James E. "Engineering of unstable requirements using agile methods." *International Workshop on Time-Constrained Requirements Engineering (7'CRE '02)*. Essen. 2002.

- [3] Zhang, Zheyang. "Effective Requirements Development-A Comparison of Requirements Elicitation techniques." *Tampere, Finland, INSPIRE* (2007).
- [4] Paetsch, Frauke, Armin Eberlein, and Frank Maurer. "Requirements engineering and agile software development." *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WETICE 2003. Proceedings. Twelfth IEEE International Workshops on*. IEEE, 2003.
- [5] Cao, Lan, and Balasubramaniam Ramesh. "Agile requirements engineering practices: An empirical study." *Software, IEEE* 25.1 (2008): 60-67.
- [6] Racheva, Zornitza, *et al.* "A conceptual model and process for client-driven agile requirements prioritization." *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*. IEEE, 2010.
- [7] Bjarnason, Elizabeth, Krzysztof Wnuk, and Bjorn Regnell. "A case study on benefits and side-effects of agile practices in large- scale requirements engineering." *Proceedings of the 1st Workshop on Agile Requirements Engineering*. ACM, 2011.
- [8] Hossain, Emam, Muhammad Ali Babar, and Hye-young Paik. "Using scrum in global software development: a systematic literature review." *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*. IEEE, 2009.
- [9] Bragge, J.; Merisalo-Rantanen, H.; Hallikainen, P. "Gathering innovative end-user feedback for continuous development of information systems: a repeatable and transferable e-collaboration process", *Professional Communication, IEEE Transactions on*, On page(s): 55 - 67 Volume: 48, Issue: 1, March 2005.
- [10] Becker, Shirley A., Erran Carmel, and Alan R. O Prioritization of Software Requirements." *International Journal of Recent Trends in Engineering*, Finland (2009).
- [11] Leung, Hareton K.N. "Viewpoint-Based Risk Assessment and Prioritization", *Software Engineering Workshop (SEW), 2012 35th Annual IEEE*, On page(s): 193 —197.
- [12] Pu, Yu-Ning, and Qiang Liu. "A Viewpoint- Oriented Requirements Elicitation Integrated with Aspects." *Computer Science and Information Engineering, 2009 WRI World Congress on*. Vol. 7. IEEE, 2009.
- [13] Li, Jinjin. "Agile Software Development."
- [14] Hoda, Rashina, James Noble, and Stuart Marshall. "Self-Organizing Roles on Agile Software Development Teams." (2013).
- [15] Highsmith, Jim. "Agile project management: Principles and Tools." *Feedback* 4 (2013).
- [16] El Emam, Khaled. "Finding success in small software projects." *Feedback* 4 (2013).
- [17] On, Ken, Ian S. Hayes, Suzanne Robertson, Thomas Marzolf, Michael Guttman, Tom Bragg, Khaled El Emam *et al.* "Agile Requirements." *Feedback* 4 (2013).