ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

AGILE METHODOLOGY AND SOFTWARE REUSE A COMMON APPROCH TO SOFTWARE DEVELOPMENT

Manjot Singh Ahuja, Neha Sadana
CSE Department
Shivalik Institute of Engineering and Technology, Aliyaspur, Haryana, India

Abstract

Years before software were not so complex, but with changing requirements size of software has increased and complexity too. Fast development, adaptability to constantly changing requirements, low cost and of high quality is also demand of time. In earlier software development processes there were no interaction between development team and client but new development processes these interactions have become common. New development processes, that is, Agile Methodology, came in light because in previous ones some deficiencies appeared, that is, time to delivery problems. As with time, deficiencies appear in every approach, and agile development is not an exception to the same. Deficiency found in Agile Methodology is of reusability, which is due to lack of time to delivery. Reusability can be acceleration to agile methodology. To adapt reusability in Agile, we need to build a database which will contain reusable components, that is, component based development. To build that database, it will take time and effort, but once it is adapted, it will be useful in improving quality and reducing time of development.

Keywords: Agile Methodology, Software Reuse, Factors affecting reusability.

Introduction

Different software development models failed to satisfy needs of present software industry. The aim of all the process models is to deliver quality product with reduced time of development, and

ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

reduced cost. Still, no single process model is complete in itself. Software industry is moving towards new methodologies, such as, Agile Methodology. Agile processes provide a room for rapid changing requirements throughout the development cycle. It also helps in providing effective conversation between software developers and customers and also helps in early product delivery too. Agile methodology has adopted some different principles to show its effective presence. Following are the principles followed in Agile Development [1]:

1) Highest priority is to please the customer through early and continuous delivery of valuable software.
2) Welcome changing requirements, even at later stages in development. Agile processes harness change for the customer's competitive advantage.
3) Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4) Business people and developers must work together daily throughout the project.
5) Build projects around motivated individuals. Give them the environment and support they need.
6) The most efficient and effective method of conveying information to and within a development team, that is, face-to-face conversation.
7) Working software is the primary measure of progress.
8) Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
9) Continuous attention to technical superiority and good design enhances agility.
10) Simplicity.
11) The best architectures, requirements, and designs emerge from self-organizing teams.

ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

12) At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Methodology

Agile is collection of various methodologies, these methodologies are consistent with the above principles. The most popular ones are:

Dynamic System Development Method (DSDM) - DSDM [2] was in light before the term 'Agile' was even invented, but is absolutely based on all the principles that are known for Agile. It is an organized process focused on delivering business solutions quickly and efficiently. It is similar in many ways to SCRUM and XP, but it has its best uses where the time requirement and recourses are fixed. DSDM makes heavy use of prototyping to make sure interested parties have a clear picture of all aspects of the system. DSDM differentiates on the following four types of prototypes [3].

1) Business Prototype: Allow assessment of the evolving system

2) Usability Prototype: Check the user interface

3) Performance Prototype: Ensure solution will deliver good performance while handling big volumes

4) Capability Prototype: Evaluate possible options

In Traditional Method, Recourses and Time are Variable	In Traditional Method, Functionality are Fixed

ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

Recourses	**Time**	**Functionality**
In Dynamic System Development Method, Recourses and Time are Fixed		In Dynamic System Development Method, Functionality are Variable

Scrum- It is also an agile development method, which concentrates particularly on how to manage tasks within a team-based development environment. It allows the process to progress by iterative and incremental development called agile sprints. Each sprint typically goes between 2-4 weeks. It is ideally suited for projects which are prone to rapid changes. Scrum [4] is relatively simple to implement and addresses many issues that have plagued IT development teams for decades.

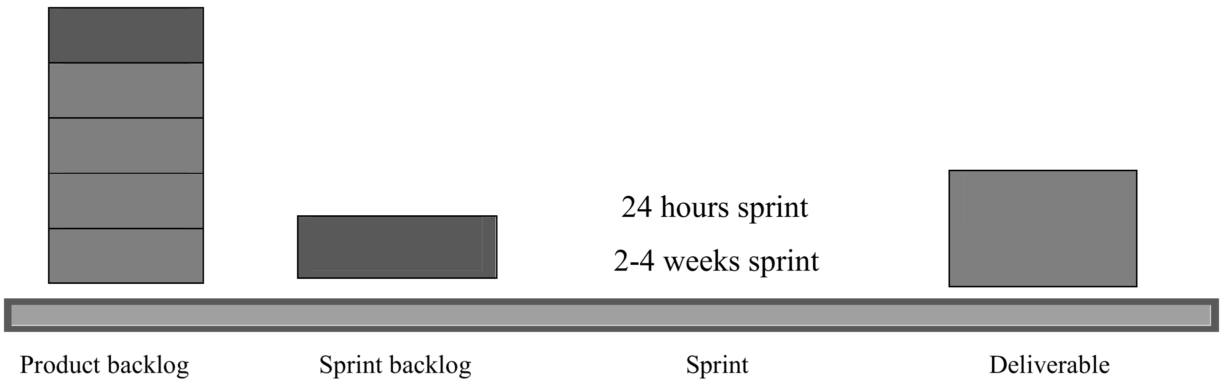

Fig 1: Working of Scrum (proceeds from left to right direction)

Sprint - meeting in which discussion is there regarding the progress of project.

ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

Backlog – unfinished parts of the process.

Scrum Team and Roles

1) **Product Owner**- Ensures that team delivers value to the business. Product owner write user stories (customer-centric), prioritize them, and adds them to product backlog if needed.
2) **Development Team**- Responsible for delivering product increments at the end of each sprint. Its size is small (3-9 members) and having cross-functional skills (analyze, design, develop, test, technical communication, document, etc.).
3) **Scrum Master** - Is responsible for removing impediments to deliver sprint goals or deliverables. Scrum master ensures that scrum process is used as intended, in other words we can say is enforcer of rules. This helps team to focus on task at hand.
4) **Stakeholders** - Are customers or vendors that are directly involved in sprint reviews.
5) **Managers** – People who control work environment.

Daily sprint answers the following-

1) Last 24 hour tasks
2) Plan for next 24 hours tasks
3) What are all obstructions of the task

XP (Extreme Programming) [5]- A light weight agile methodology focusing more on the software engineering process and addressing the analysis, development and test phases with novel approaches that make a substantial difference to the quality of the end product. In this testing take place in parallel. Business requirements are gathered in terms of stories and these stories are stored in place called parking lot. Releases are based on shorter cycles with span of 14 days.

ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

Phases of Extreme Programming [6]-

1) Planning
2) Analysis
3) Design
4) Execution
5) Wrapping
6) Closure

Software Reuse

Software engineering deals with the development of software systems. As the size of the software system is increasing because of complexity and demand time is decreasing, new approaches of software development coming in the environment. These approaches include Object-oriented programming [8], Component-based programming [9], and Aspect-based programming [10]. Above mentioned approaches are effective for software development but there is need to reduce the effort, time, and cost to build the software so that productivity and quality of software programs can be increased. Software reuse [7] can be a means to reduced development cost and can improve quality. Software reuse is the use of existing software to build new software. Reusable software can be code, templates, functions, procedures, objects, routines or framework.

International Manuscript ID : ISSN23194618-V2I1M13-012013

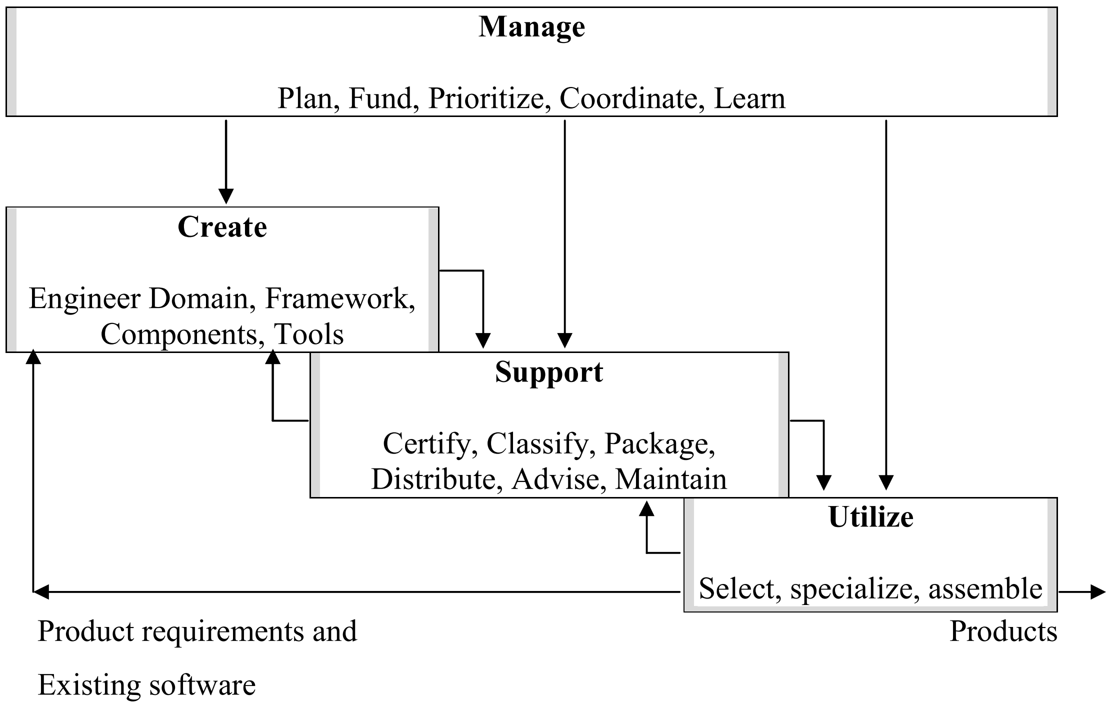
Manage
Plan, Fund, Prioritize, Coordinate, Learn
Create
Engineer Domain, Framework, Components, Tools
Support
Certify, Classify, Package, Distribute, Advise, Maintain
Utilize
Select, specialize, assemble
Product requirements and Existing software
Products

Fig 2: Systematic reuse process and organization

By systematic reuse [11], it is meant that an institutionalized organizational approach to product development in which reusable assets are purposely created or acquired, and then consistently stored, used, and maintained to obtain high levels of reuse. Thereby optimize the organization's overall ability to produce quality software products rapidly and effectively. This requires a significant effort to change working culture of organization, and a multitude of other factors. Changes associated with CPI (continuous process improvement). It is essential to change the way of viewing software at a fundamental level. It must be viewed as an organizational asset, to be invested in, improved effectively and consistently. Initial investment is required to start the reuse process, in order to find reusable components, so that those components can be reused in future programs, and when this reuse process is used in the new software system it will dwindle time to many folds. To find reusable components [12] we have to find reusability of components. Reusability is the extent to which a component can be reused. To find reusability we need a metric which can find the reusability of components. There exist lot of such reusability metrics; out of those metrics we can adopt any efficient metric to find reusability. To choose effective metric, we should know what the factors which affect reusability, so, following are the factors which we extracted from previous studies [13], [14] and after analyzing them, which affecting reusability:

1) Adaptability
2) Availability
3) Complexity
4) Completeness
5) Correctness
6) Cohesion
7) Coupling
8) Documentation
9) Efficiency
10) Expandability
11) Generality
12) Maintainability
13) Modularity
14) Portability
15) Price
16) Quality
17) Reuse
18) Reliability
19) Testability

So, these are some of the factors which one should keep in mind to decide the reusability of components.

Proposed work

In early studies done on reusability in Agile, it has been observed that most of the emphases were made on the classification of components in database, so that extracting reusable components become easier. No one can deny the usefulness of these researches, but prior to this step, we have to find the components which can be reused. It is important to find the components with higher reusability factor. There are number of factors which affect the reusability of components, and if value of these factors is high, reusability of component decreases. We have to find those components which are having low value of the factors having negative effect on reusability, and then to add those components to database. If this approach is adopted, time of finding reusable component from repository will be decreased and further efforts for modifying those components will be reduced, because we are all ready having those components in repository which are having high reusability value. To find reusability of component we have to test those components with some metric that can find the reusability. And we already have discussed factors which affect reusability. So, in Fig3 there is proposed model which can help us improving work flow in agile atmosphere. In this model we have shown how reusability can be applied in agile atmosphere.

ISSN (Print) : 2321 - 3280 ISSN (Online) : 2319 - 4618

International Journal of Reviews, Surveys and Research (IJRSR)

International Refereed and Indexed Journal for Research Scholars and Practitioners

International Manuscript ID : ISSN23194618-V2I1M13-012013

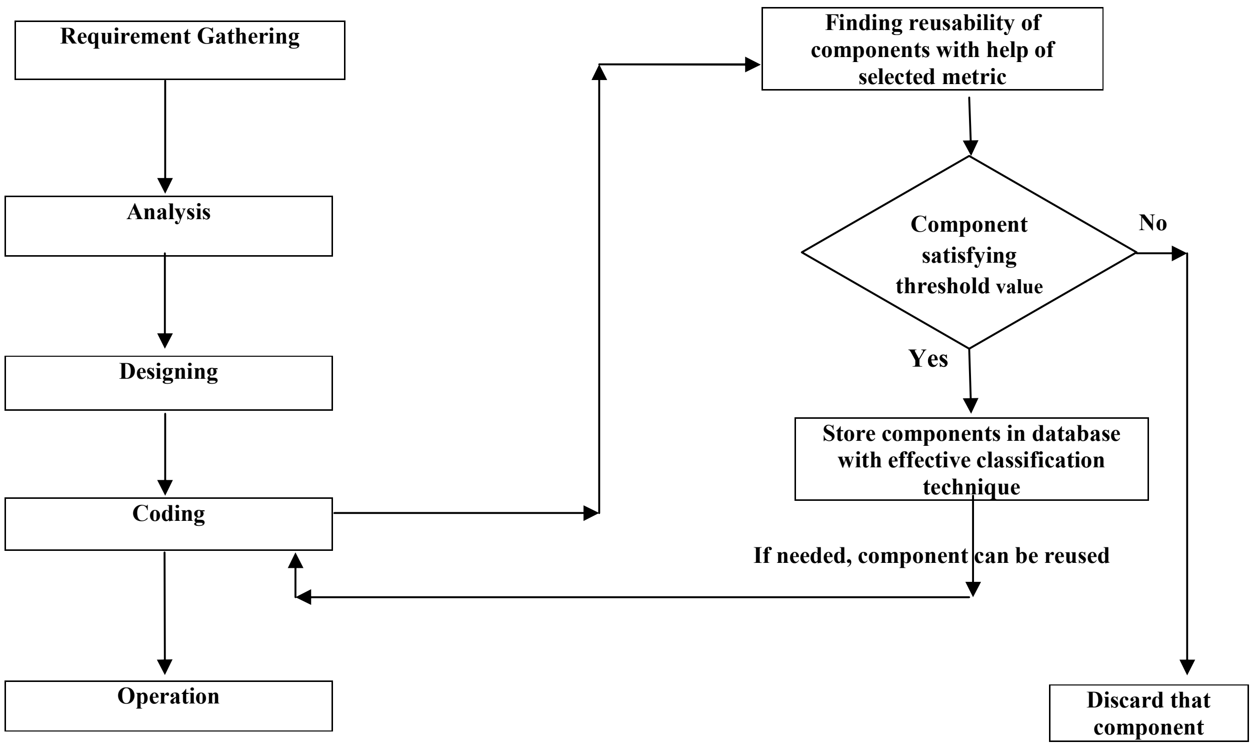
Requirement Gathering
Analysis
Designing
Coding
Operation
Finding reusability of components with help of selected metric
Component satisfying threshold value
No
Yes
Store components in database with effective classification technique
If needed, component can be reused
Discard that component

Fig3: Flowchart for Proposed Model

Explanation of model: while we develop code in agile atmosphere, we can test code components with reusability metric. There will be a threshold value that a code component should satisfy, if that code component satisfies threshold value, component will be added to the database according to classification criteria for further reuse, but if threshold is not achieved, we can discard that component and that will not be added to the reuse component database.

Conclusion and Future Scope

From above defined principles and methodologies it can be clearly observed that Agile Methodologies are inclined only towards that client's need. But work is not done keeping reusability in mind. As in agile methodology, time we have to spend on development is limited; hence compromise with quality is made, as quality of software also depends on quality of code and documentation. So there is need of reusable artifacts (analysis, design document, patterns, etc.). Lack of documentation and design in development make it difficult to extract reusable functionalities. Because of this, difficulty level and cost of modification also increases. So by adopting reusability in agile development, quality of system can be maintained and time can further be reduced to many folds. In this proposed work we have added concept of reusability at coding stage, which can help in reducing coding efforts and saving lot of time.

In future we will present reusability metric for measuring reusability of code and will try to implement scope of reusability at other stages of software life cycle model, helping reducing time and effort.

References

[1] http://agilemanifesto.org/principles.html

[2] http://dsdmofagilemethodology.wikidot.com/

[3] http://www.codeproject.com/Articles/5097/What-Is-DSDM

[4] http://en.wikipedia.org/wiki/Scrum_(development)

[5] http://www.umsl.edu/~sauterv/analysis/f06Papers/Hutagalung/

[6] http://www.brighthubpm.com/methods-strategies/88996-the-extreme-programming-life-cycle/

[7] http://www.cs.toronto.edu/~yijun/ece450h/handouts/lecture8x4.pdf

[8] http://www.codeproject.com/Articles/22769/Introduction-to-Object-Oriented-Programming-Concep

[9] http://acmantwerp.acm.org/wp-content/uploads/2010/10/componentbasedprogramming.pdf

[10] http://en.wikipedia.org/wiki/Aspect-oriented_programming

[11] http://martin.griss.com/pubs/fusion1.htm

[12] G. Caldiera and V.R. Basili, Identifying and qualifying reusable software components, IEEE Computer, vol.24, Feb.1991.

[13] Joakim Fröberg , " Software Components and COTS in Software System Development".

[14] W.J. Salamon , D.R Wallace ," Quality Characteristics and Metrics for reusable software "May 1994

[15] Pressman R. S., "Software Engineering", 7th edition, McGraw Hill Education, 2009