IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 10, Oct 2018

www.ijasrm.com

ISSN 2455-6378

# Determination of similar and dissimilar use cases for new agile project by applying case based reasoning

**R.Thirumalai Selvi[1] and R.Manjula[2]**

[1]Asst.Prof.,Govt.Arts College,
Nandanam.Chennai,Tamilnadu,India.

[2]Research Scholar, Bharathiyar University,
Coimbatore,Tamilnadu,India

**Abstract**

There is a common misconception among developers who follow Agile development methods, that following formal processes and modeling are unnecessarily and perceived as a waste of effort . The initial intension of Agile was not an anti-methodology movement but rather a balance between processes and production. Adopting more inclusive knowledge management processes and concepts can be used to overcome some of Agile's challenges. It focuses primary on transferring the tacit type of knowledge within production teams. Yet transferring the explicit type of knowledge is important as well. Reusing past project's artifacts will have a positive impact. Use cases are used to estimate software effort. This paper focuses on reusing Historical project use cases to the new project with the help of case based reasoning method and similarity functions. Case Based Reasoning(CBR) method is a knowledge based reasoning process which is used to retrieve the usecases from the past project with the specified criteria. Its also evaluate the number of similar and dissimilar use cases present in the new project by compare with the past project's usecases.

***Keywords:*** *Case Based Reasoning, Reusability,Similarity function, Use Cases.*

## 1. Introduction

The area of new product development is highly complex and uncertain due to demanding environment characterized by increased globalization and segmentation of markets, increased levels of product complexity, changing customer needs and shorter product life cycles. Agile has gained an increasing popularity in software development during last decade. It has increasingly used at various types of projects such as e-commrece, e-services , e-government etc. due to its quickness in responding to unpredictability business changes. Agile has a collection of core value statements and principles. Cost can be reduced by reusing the past project's artifact and also increased the overall learn ability, productivity and efficiency of the organization . Reusability is one of the bottlenecks of agile software development process. Reusability finds limited scope in agile software development and therefore adding reusability to agile software development is a challenge. Essence of agile software development is rapid software development and less cost..During the analysis of a project, identifying potential opportunities for reuse is important one. To implement reusability concept past project data's are required. Not all the data are reused, only the use cases are taken from the past project. Use case are cluster as similar and dissimilar with the help of case based reasoning and similarity function. Detailed description are shown in the following section.

## 2.Literature Survey

A systematic literature review is conducted methodically by following a set of guidelines to collect and analyze all available evidence about a use of Historical project data set, Case Based Reasoning

and similarity function. A prototype system is developed to improve the software quality management by applying case based reasoning is discussed[1]. Mostly used Case Based Reasoning tools are discussed through comparative analysis study to find out the determined factors that affect the CBR is presented[2]. Guidelines are provided through case based reasoning to improve the agile practices in software development[3]. Item Case Based Reasoning is applied on the scrum methodology to maximize the use of cooperative wisdom and experience found within organizational entity[4].Models are developed for user based similarity calculation to enhance the recommendation performance and to estimate the similarities for each user.

## 3.Reusability

Software reuse is the process of implementing or updating software systems using existing software components. A good software reuse process facilitates the increase of productivity, quality, and reliability, and the decrease of costs and implementation time. An initial investment is required to start a software reuse process, but that investment pays for itself in a few reuses. In short, the development of a reuse process and repository produces a base of knowledge that improves in quality after every reuse, minimizing the amount of development work required for future projects, and ultimately reducing the risk of new projects that are based on repository knowledge.. With all the costs and prerequisites, software reuse may seem like more effort than it is worth. However, the number of success stories with increases in productivity, quality, and reliability, and decreases in production time, hint toward a goal worth achieving.

## 4.Historical Data

Collecting data from the pas project is called historical project data, it includes most data generated either manually or automatically within an organisation. Storage capacities have increased significant in recent years and cloud storage has taken some of the storage administration from many organisation, required amount of historical data depends on the type of user that will be using the data. Users are in two types farmers and explorers. Farmers are analysts they know the content of the data not the type of data, they do the same types of analysis repeated. Farmers do not require great deal of history. Explorers are people who think outside the box. They look at very large amounts of data and very unpredictable. Explorers need a lot of historical data and they look for patterns of data. Organisation make sense to periodically monitor the usage of the data. Historical data can be removed from the system where there is very frequent access of the data. There are three reasons to keep the past project data.

### 4.1 Understand the past

Keeping historical data enables the company to understand the past project detail. Time-to-resolution(TTR) will be faster, so it satisfies the client. The client is frustrated and the TTR is long with historical data, the problem would have been easily identified. It helps to improve the quality of the project.

### 4.2 Understand the evolution

Keeping historical data enable to understand the evolution pattern, customer expectation and trace the average variation of marketers.

### 4.3 Enable Forecasting

With the help of historical data future events can be easily forecasted. To predict marker changes in future, historical data provides valuable information to help the project manager to understand trends over time.

### 4.4 Estimating initial costs

Initial financial guidelines for the project can be established using previous project data. It also provides an opportunity to explore the feasibility of reusing components from a similar system. This form of reuse makes cost estimation much easier.

To implement the reusability concept historical project data sets are collected. Not all the details are collected, from the past project only the use case scenario are collected, It includes use cases and its relevant project complexity.

## 5. Complexity

Complexity means intricacy of the project which may cover multiple features or functionality. Business complexity are occur due to the uncertain conditions, varied methods and technological advancements. Project level complexity lies in two different categories.

1) Project complexity
2) Requirement complexity

### 5.1 Project complexity

It can be simple, medium and complex. The level of project complexity can be measured from three different attributes.

- Uniqueness – Every project has unique attributes and requirements. Project uniqueness prevails when the organisation has no prior experience in running similar projects.
- Limiting factors – Some of the project parameters are known as limiting factors. They are schedule budget etc.
- Uncertainty: It can be driven by external or internal factors. The most common external factors government regulation, market change and economic climate. Internal factors are real and contribute to enhanced levels of volatility in process.The rate of success in any particular organisation can be determined from project complexity.

### 5.2 Requirement complexity

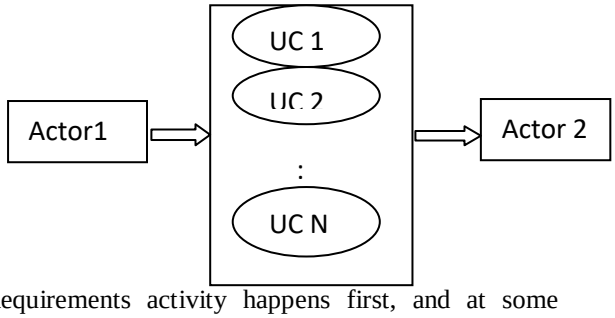
To understand the business problem and needs requirement analysis are performed. The level of unknown are volatility which must be ensured at a very granular level systems, function business rules etc.

- Volatility-Frequent changes of requirements from the starting of the project to the implementation is known as volatility. Requirements volatility leads to important risk and its consequent uncertainty.

## 6.Use Cases

Usecases describe the step-by-step process in which a user goes through to complete the goal. It has a description of an end-user wants to use a system. To describe the functional requirements of the system, use cases are used to capture user point of view. It also capture the wrong things which affect the goal. It describe non-functional requirements from the user perspective to achieve the particular goals. There is a many-to-many to relationships between requirements and goals. Team member has to verify that whether the use cases communicate the most important goal which is addressed by a stakeholder. All specifications of use cases represented through graphical and textual parts which is known as use case diagram. Use case can have three part.

- Usecases
- Actors
- Functions to interact between them




Requirements activity happens first, and at some point, there is *enough* complete to begin design. After a point, development can begin, etc. It does not mean that the previous activity has stopped. The use case process should be started after the goals and scope are defined for the project. There are four main reasons to write use cases:

- ❖ Discover requirements that would otherwise go undiscovered.
- ❖ Capture the correct requirements.
- ❖ Communicate the requirements effectively.
- ❖ Prioritize the requirements.

### 6.1Agile User Stories Vs Use Cases

Any agile development approach to anything better be aligned with the agile manifesto. There are significant benefits to using an incremental delivery process. Agile software development processes vary in the details, but universally include incremental delivery as a key structural component of the process. Incremental delivery has two primary benefits over waterfall process:

1. Value is delivered faster by releasing valuable subsets of the product as you go.
2. More value is delivered in total because feedback during the process improves ongoing decisions.

User stories are used in agile to help breakdown features into sizable chunks which can fit into a sprint. The purpose of user story is to describe the users or user personal, what they need and why they need it. User story will follow the format

**"As a [persona], I [want to], [so that]."**

The User story will also have Aceeptance criteria to confirm the user story. Use cases are often used in software and system engineering as a part of requirement definition. The purpose of use case is to describe user or system steps of process. Use case mainly consists of a title, Description, user or user personal, Pre-conditions, Basic flow, Alternative flow, Negative flows and post conditions. Process flow can be used to visualize the steps of the use case. In Agile project, use case is developed first to understand the process between the users and the system. Use cases help to discover the beginning and end state of the process and different sceneries that may occur in that process. Use case can create a process flow in order to visually represent the steps and flows. On Agile projects, use cases and process flows can be used to derive epics and user stories. On Non-Agile projects use cases and process flows are used in requirement document to define process steps between user system.Table 1.shows the difference between User story and Usecases

| Sl. No. | User story | Use cases |
|---|---|---|
| 1 | User stories are centered on the result and benefit of the thing | Use cases are more granular and describe how the system will act. |
| 2. | It is a short description of customer work | It is a description of a set of interaction between a system and one or more actors. |
| 3. | These are written from the point of view of a person | It include one or more actor or people or other systems |
| 4. | User story is written using the format[canonized by mike cohn] | They are created as documents. |
| 5. | User story include Acceptance criteria which define the boundaries of a user story. | Use case include title,descritption, user, Pre-condition, Basic flow etc. |
| 6. | User story need a decision of stand-up meeting | Use case need up-front research. |
| 7. | User story is a great way to describe requirement | Use case is a great way to detailed user stories. |

Table 1.User story Vs Usecases

## 7.Case-Based Reasoning

CBR is one of the newer artificial intelligence approaches that enables problems to be solved by reference to past problem situation or cases held in a case library. Traditional software approach model consist of heavy weight planning, coding, severe reusing, complex documentation and design interface. So, these are called Heavy-weight development methods. Each case typically contains a description of the problem, plus a solution and/or the outcome. The knowledge and reasoning process used by an expert to solve the problem is not recorded, but is implicit in the solution.To solve a current problem: the problem is matched against the cases in the case base, and similar cases are retrieved. The retrieved cases are used to suggest a solution which is reused and tested for success. If necessary, the solution is then revised. Finally the current problem and the final solution are retained as part of a new case. Case-based reasoning is liked by many people because they feel happier with examples rather than conclusions separated from their context. A case library can also be a powerful corporate resource, allowing everyone in an organisation to tap into the corporate case library when handling a new problem. All case-based reasoning methods have in common the following process:

- retrieve the most similar case (or cases) comparing the case to the library of past cases;
- reuse the retrieved case to try to solve the current problem;
- revise and adapt the proposed solution if necessary;
- retain the final solution as part of a new case.

### 7.1Suitability of CBR

Some of the characteristics of a domain that indicate that a CBR approach might be suitable include:

1. records of previously solved problems exist;
2. historical cases are viewed as an asset which ought to be preserved;
3. remembering previous experiences is useful;

IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 10, Oct 2018

www.ijasrm.com

ISSN 2455-6378

4. specialists talk about their domain by giving examples;
5. experience is at least as valuable as textbook knowledge.

Case-based reasoning is often used where experts find it hard to articulate their thought processes when solving problems. This is because knowledge acquisition for a classical KBS would be extremely difficult in such domains, and is likely to produce incomplete or inaccurate results. When using case-based reasoning, the need for knowledge acquisition can be limited to establishing how to characterise cases.Case-based reasoning allows the case-base to be developed incrementally, while maintenance of the case library is relatively easy and can be carried out by domain experts.

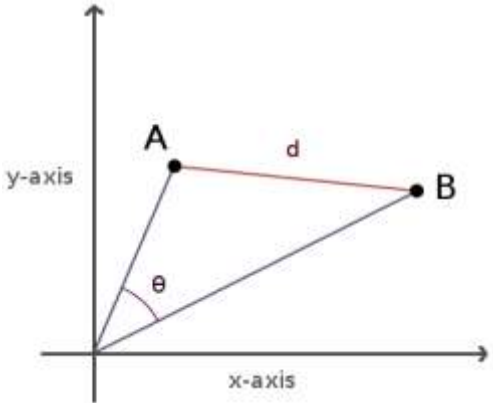
## 8.Similarity Function:

In statistics and related fields, a similarity measure or similarity function is a real-valued function that quantifies the similarity between two objects. Similarity or distance measures are core components used by distance based clustering algorithms to cluster similar data points into the same clusters, while dissimilar or distant data points are placed into different clusters. The similarity function measures the level of similarity between projects. The performance of similarity measures is mostly addressed in two or three-dimensional spaces. By selecting the a good distance function over input data set the performance of the algorithm is calculated

### 8.1Euclidean distance

It measures dissimilarity. The coordinates that are the same are less important than the coordinates that are different.

$$distance(A,B)=\sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\ldots+(a_n-}$$




### 8.2 Cosine similarity

This method is very similar to the one above, but does tend to give slightly different results, because this one actually measures similarity instead of dissimilarity. Here's the formula:

$$similarity=\cos(\theta)=\frac{A\cdot B}{\|A\|\|B\|}=\frac{\sum_{i=1}^{n}A_iB_i}{\sqrt{\sum_{i=1}^{n}A_i^2}\sqrt{\sum_{i=1}^{n}B_i^2}}$$

The values will range between -1 and 1. -1 means that 2 items are total opposites, 0 means that the 2 items are independent of each other and 1 means that the 2 items are very similar.

## 9. Proposed Methodology

The use of analogy is also called case-based reasoning. The estimator seeks out projects that have been completed (source cases) and that have similar characteristics to the new project (the target case). The effort that has been recorded for the matching source case can then be used as a base estimate for the target. The estimator should then try to identify any differences between the target and the source and make adjustments to the base estimate for the new project. A problem here is how you actually identify the similarities and differences between the different systems. In this paper, we propose an approach to derive use case metrics from the Historical Project use cases with the help of case based reasoning and similarity function to count the number of simple, medium and complex use cases.

Algorithm:

If NP(Ts,Tech,D,WE,TM) = HP(Ts,Tech,D,WE,TM) then

-Apply Case –Based Reasoning Method to retrieve use case

-Euclidean distance function used to find dissimilar use cases between new and old project

-Cosine Similarity function is used to find similar use cases between new and old project

-Count the number of similar and dissimilar use cases in simple,medium and complex

End if

[Note : Ts-Tools,Tech-Technology, D-Domain, WE-Working Environment, TM- Team Member]

Tools, Technology, Domain and Team Member informations are collected from Historical Project and New Project . If the collected details are same for both the project, then use case libraries are retrieved using case based reasoning method. . To find out the dissimilar use case Euclidean function is used and count the number of dissimilar use cases. Similar use cases are find out using cosine similarity function . From this number of similar use cases are counted between the new and old project. The similar use cases and it related functions are used for new project implementation. Identified dissimilar use cases are needs to be developed for the new project.

## 10. Results and Discussion

Historical project use cases are collected from the product backlog of old project. New Project use case are mapped to the Historical project use cases. Use cases contain complexity values. Complexity metric values are assigned based on the number of functions involved in the use cases. Complexity value can be differ from organization to organization. To assign the complexity values as simple, medium and complex guideline should be prepared by the team member and project manager. For example

Let $A_i = \{a_1, a_2, \ldots\ldots a_n\}$ be the number of use cases in Historical project

$B_i = \{b_1, b_2. \ldots. B_n\}$ be the number of use cases in New project

To determine dissimilar use cases Euclidean distance is used. It identifies the source case that is nearest the target by measuring the Euclidean distance between cases. The source case that is at the shortest Euclidean distance from the target is deemed to be the closest match. The Euclidean distance is calculated:

```
 [1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 1]
[1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1,
0, 1]
```

So 4 coordinates are different., ie., four usecases are dissimilar.

To determine similar use cases between the new project and past project cosine similarity functions are used. Result are discussed in this section.

| | Use cases | Simple | Medium | Complex |
|---|---|---|---|---|
| Configuration (Custom fields, Email templates, custom object, page layout, Validation, work flow, Report, Dashboards etc.. ) | Custom Objects (1 object) | 1 | 1 | 1 |
| | Custom Fields (10 fields) | 1 | 2 | 3 |
| | Custom Button | 1 | 1 | 1 |
| | Page Layout (1 Layout) | 1 | 2 | 4 |
| | Validation (5 Validation) | 8 | 16 | 24 |
| | Email Templates (1 Template) | 1 | 3 | 12 |
| | Work flow (5) | 2 | 5 | 8 |
| | Report (1) | 2 | 4 | 8 |
| | Dashboard (1) | 1 | 2 | 4 |
| | Record Type (2) | 1 | 1 | 1 |
| | Report Type (1) | 1 | 2 | 2 |
| | Profile(1) | 4 | 4 | 4 |
| | Role | 8 | 8 | 8 |
| | Web to Lead(1) | 1 | 1 | 1 |
| | Security | 8 | 8 | 8 |
| | Assignment Rules (5) | 1 | 3 | 8 |
| | Auto Response Rules (Lead & Case) | 1 | 3 | 8 |
| | Email to Case(1) | 2 | 2 | 2 |
| | Product Price Book Setup | 5 | 5 | 5 |
| | Quote | 3 | 5 | 8 |
| | Knowledge base | 16 | 16 | 16 |
| | Customer Portal(1) | 24 | 24 | 24 |
| | Partner Portal(1) | 24 | 24 | 24 |
| | Content | 8 | 8 | 8 |
| | Manage Territories | 8 | 8 | 8 |
| | User Management | 8 | 8 | 8 |
| | Sharing Rules | 1 | 1 | 1 |
| | Public Groups | 2 | 2 | 2 |
| | Salesforce Mobile | 8 | 8 | 8 |
| | Salesforce for Outlook | 8 | 8 | 8 |
| | Connect Offline | 4 | 4 | 4 |
| | Home Page Layout | 2 | 4 | 6 |
| | Chatter | 8 | 8 | 8 |
| | Custom Apps | 1 | 1 | 1 |
| | Salesforce to Salesforce | 6 | 6 | 6 |
| | Console | 4 | 4 | 4 |
| | Search Layout | 1 | 1 | 1 |
| | Approval Process | 2 | 4 | 6 |
| | Tab | 1 | 1 | 1 |
| | Queue | 1 | 1 | 1 |
| Customization (VF page, Trigger, Apex class, Batch Class, Schedulable Class, VF Email Templates, Custom JS button) | VF Page | 8 | 12 | 24 |
| | Triggers | 4 | 8 | 12 |
| | Apex Class | 5 | 10 | 16 |
| | Test Class | 2 | 4 | 6 |
| | Sites | 2 | 2 | 2 |
| Integration(Web Service, Call outs, Single Sign On) | Web Service | 16 | 24 | 32 |
| | Single Sign On | 16 | 16 | 16 |
| | CTI | 16 | 16 | 16 |
| Migration | Data Migration | 6 | 10 | 14 |

Table 2. Use cases in Historical project

The above Table 2. contain an example of use cases for an Insurance project. This use case is compared to the new project use cases to determine number of similar and dissimilar use cases. Table 3. contain the number of similar and dissimilar usecases in new project, by compare with past project usecases.

| Use case complexity | Total use cases | similar use cases | dissimilar use cases |
|---|---|---|---|
| Simple | 12 | 5 | 7 |
| Medium | 15 | 6 | 9 |
| Complex | 23 | 9 | 14 |

Table : 3. New project usecases

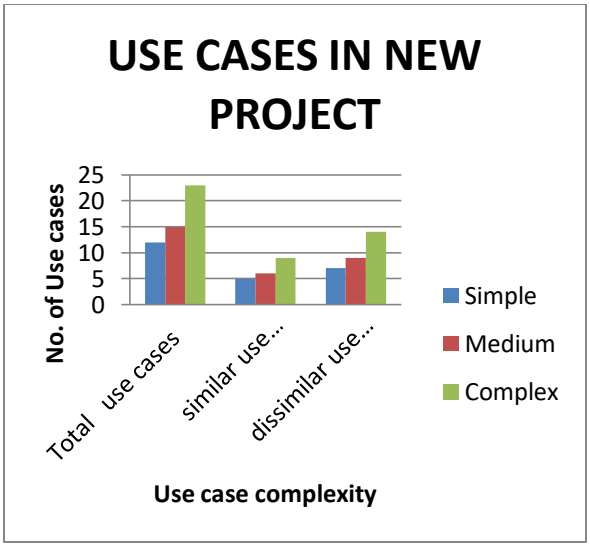



Fig 1.New Project UseCase Representation

Fig 1 represents, number of similar and dissimilar use cases in different complexity metrics such as simple, medium and complex usecases , among the 50 use cases of a new project. Similar usecases are retained to the new project and dissimilar use cases are needs development.

## 11.Conclusion

Agile process are valuable and Use cases are important. There is a lot can do with use cases, especially when applying agile principles to improve the quality of use cases. Historical project's use cases are reused to the new project to improve the productivity, quality and reduce the cost & time. Case Based reasoning techniques are used to retrieve the use cases. The similarity functions Euclidean distance and cosine similarity functions are applied to determine and count the number of similar and dissimilar use cases present in new project by comparing the Historical project use cases. Estimated use cases are help to predict the effort, cost and duration of a new project easily. In future, effort,cost and duration are predicted for the new project using the previous project use cases which is very helpful to the project manager to manage the project easily.

## References:

[1] B Lees1, M Hamza2 and C Irgens2, "Applying Case Based Reasoning to Software Quality Management", University of Paisley, Paisley PA1 2BE, Scotland UK,1996.

[2] Passent ElKafrawy*, Rania. A. Mohamed,"Comparative Study of Case Based Reasoning Software", IJSRMS,ISSN: 2349-3771,2011

[3] Mehwish Mukhtara et.al., "A Hybrid Model for Agile Practices Using Case Based Reasoning",IEEE,2013

[4] Aiman TuranI,"Applying Case Based Reasoning in Agile Software Development",JATIT,2015

[5] Mr. K. G. Saranya*, G. Sudha Sadasivam and M. Chandralekha,"Performance comparison of similarity measures for collaborative filtering techniques