ISSN 2278-5612

A Comparative Analysis of Different Clustering Approaches for Software Process Improvement

Rajni Rana[1], Dr. P.K. Suri[2]
[1]HCTM Technical Campus, Kaithal, Haryana, India.
rajnirana207@gmail.com
[2]Dean, Research and Development,
Chairman, CSE/IT/MCA, HCTM Technical Campus, Kaithal, Haryana, India.
pksuritf25@yahoo.com

ABSTRACT

Software development team tries to increase the software quality by decreasing the number of defects as much as possible. Number of defects remaining in a system provides an insight into the quality of the system. Software defects are one of the major factors that can decide the time of software delivery. The proposed system will analyze and categorize the software defects using some clustering approach and then the software defects will be measured in each clustered separately. Clustering is the process to present the data in an effective and organized way. There are number of existing clustering approaches but most of them suffer with problem of data distribution. If the distribution is non linear it gives impurities in clustering process. The proposed work is about to use defect prevention for process improvement with the help of clustering algorithms.

KEYWORDS

Defect Prevention,Clustering, C-Means, K-Means, Hierarchical

Council for Innovative Research
Peer Review Research Publishing System
Journal: International Journal of Management & Information Technology

Vol. 5, No. 1
editor@cirworld.com
www.cirworld.com, member.cirworld.com

1. INTRODUCTION

The process of grouping a set of objects into classes of similar objects is called Clustering. By clustering one can identify dense and sparse region and overall distribution pattern. Clustering technique categorize the defects correction effort is used as the criterion for classifying the defects thereby providing the project managers with general idea about the nature of complexity of defects. This Clustering technique helps the project managers to understand about the nature of defects and allocate resources [4]. Software Defect can be defined as “Imperfections in software development process that would cause software to fail to meet the desired expectations”. In software development, lot of defects would emerge during the development process. It is a fallacy to believe that defects get injected in the beginning of the cycle and are removed through the rest of the development process [8]. Defects occur all the way through the development process. Hence, defect prevention becomes an essential part of software process quality improvement. Defect prevention (DP) is a process of improving quality whose purpose is to identify the common causes of defects, and change the relevant processes to prevent that type of defect from recurring [2]. DP also increases the quality of a software product while reducing overall costs, schedule and resources. This ensures a project can maintain cost – schedule – quality equilibrium. The purpose of defect prevention is to identify those defects in the beginning of the life cycle and prevent them from recurring so that the defect may not surface again. In this study, in order to improve software process quality, defects are first identified from a given set of projects, classified and and then clustering algorithms are used. The scope of this paper is to provide a comprehensive view on the need of defect prevention, process improvement workflow and different clustering algorithms.

1.1 Need for Defect Prevention

Defect prevention is an important activity in any software project. In most software organizations, the project team focuses on defect detection and rework. Thus, defect prevention, often becomes a neglected component. It is therefore advisable to make measures that prevent the defect from being introduced in the product right from early stages of the project. While the cost of such measures are the minimal, the benefits derived due to overall cost saving are significantly higher compared to cost of fixing the defect at later stage. Thus analysis of the defects at early stages reduces the time, cost and the resources required. The knowledge of defect injecting methods and processes enable the defect prevention. Once this knowledge is practiced the quality is improved. It also enhances the total productivity.

1.2 Process Improvement Work Flow Stages

There are five types of stages used for process improvement:-

1. Defect Identification
2. Defect Classification
3. Defect Analysis
4. Preventive Actions
5. Process Improvement

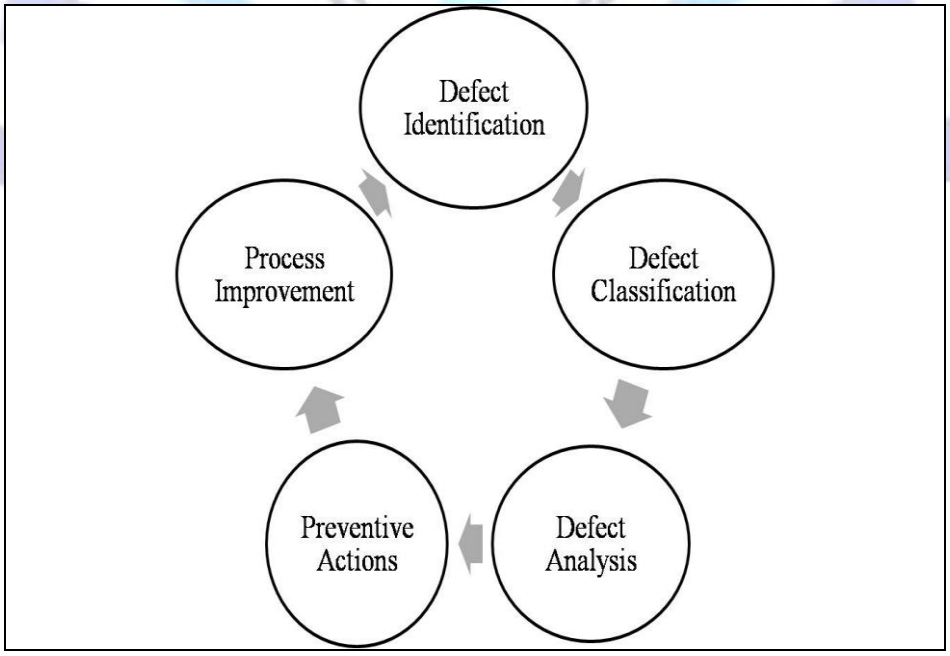

Figure 1: Process Improvement Workflow

a.) Defect Identification

Defects are found by preplanned activities specifically intended to uncover defects. In general, defects are identified at various stages of software life cycle through activities like Design review, Code Inspection, GUI review, function and unit testing. Once defects are identified they are then classified using first level of Orthogonal Defect Classification.

b.) Defect Classification

Orthogonal Defect Classification (ODC) is the most prevailing technique for identifying defects wherein defects are grouped into types rather than considered independently. ODC classifies defect at two different points in time.

Time when the defect was first detected – Opener Section.

Time when the defect got fixed – Closer Section.

ODC methodology classifies each defect into orthogonal (mutually exclusive) attributes some technical and some managerial [6].

These attributes provide all the information to be able to shift through the enormous volume of data and arrive at patterns on which root-cause analysis can be done. This coupled with good action planning and tracking can achieve high degree of defect reduction and cross learning. For small and medium projects, in order to save time and effort, the defects can be classified up to first level of ODC while critical projects typically large projects needs the defects to be classified deeply in order to get analyze and understand defects. In this paper, the project that is selected for analysis being a project coming under the category of small and medium size project, the analysis of defect is done by using first level of ODC defect classification. First level of ODC includes classifying the defects under various defect types like Requirements, Design, Logical (Logical defects are found by testing the code using functional/unit testing), and Documentation. Defects are classified under these types and then analysis of defects is carried out.

c.) Defect Analysis

Defect Analysis is using defects as data for continuous quality improvement. Defect analysis generally seeks to classify defects into categories and identify possible causes in order to direct process improvement efforts. Root Cause Analysis (RCA) has played useful roles in the analysis of software defects. The goal of RCA is to identify the root cause of defects and initiate actions so that the source of defects is eliminated. To do so, defects are analyzed, one at a time. The analysis is qualitative and only limited by the range of human investigative capabilities. The qualitative analysis provides feedback to the developers that eventually improve both the quality and the productivity of the software organization [6]

d.) Defect Prevention

Defect prevention is an important activity in any software project. The purpose of Defect Prevention is to identify the cause of defects and prevent them from recurring. Defect Prevention involves analyzing defects that were encountered in the past and taking specific actions to prevent the occurrence of those types of defects in the future. Defect Prevention can be applied to one or more phases of the software lifecycle to improve software process quality [7].

e.) Process Improvement

The suggested preventive actions are implemented by rewriting the existing quality manuals and tweaking the SDLC processes and come out with an improved SDLC processes and documents. Next set of projects follow the revised quality processes there by effectively all the preventive actions are followed meticulously.

2. LITERATURE SURVEY

P.V. Ingle, Software defects are not only an inherent component of software product but also significant factors of software Quality[1]. It is impossible to completely eliminate software defects. People can attempt to avoid or minimize defects in software development as much as possible. To deliver a defect free software it is imperative to predict and fix the defects as many as possible before the product delivers to the customer. In order to take full advantage of a large number of defects data collected in the software development process, improve the efficiency and quality of software testing .This paper applied the analysis of defects data based on its advantages in data processing.

Sakthi Kumaresh, To gain a deeper understanding of the effectiveness of the software process, it is essential to examine the details of defects detected in the past projects and to study how the same can be eliminated due to process improvements and newer methodologies[2]. This paper will focus on finding the total number of defects that has occurred in the software development process. The paper also showcases on how the preventive ideas are implemented in a new set of projects resulting in the reduction of the number of similar defects.

Suma. V., and T. R. Gopalakrishnan Nair, Defect prevention [DP] is a process of identifying defects,their root causes and corrective and preventive measures taken to prevent them from recurring in future, thus leading to the production of a quality software product [7]. Hence, organizations should opt for defect detection and prevention strategies for long-term Return on Investment (ROI). Among several approaches, inspection has proven to be the most valuable and competent technique in defect detection and prevention.

Mukesh Soni, "Prevention is better than cure" applies to defects in the software development life cycle as well as illnesses in medical science. Defects, as defined by software developers, are variances from a desired attribute. These attributes include complete and correct requirements and specifications as drawn from the desires of potential customers[8]. Thus, defects cause software to fail to meet requirements and make customers unhappy. And when a defect gets through during the development process, the earlier it is diagnosed, the easier and cheaper is the rectification of the defect. The end result in prevention or early detection is a product with zero or minimal defects.

3. CLUSTERING APPROACHES

Clustering is defined as a process to divide the available dataset in small groups, where each group contains the similar kind of data elements. Clustering is defined as unsupervised learning technique. Clustering basically deals with finding a structure in a collection of unlabeled data. There are different clustering algorithms . Clustering is used because of following reasons:

- Simplifications
- Pattern detection
- Useful in data concept construction
- Unsupervised learning process

Distance Matrices

If we have two cases, C1 and C2 say, which have continuous variables x and y, taking values (x_1, y_1) and (x_2, y_2) respectively, then we can plot the two cases in x-y space[10].

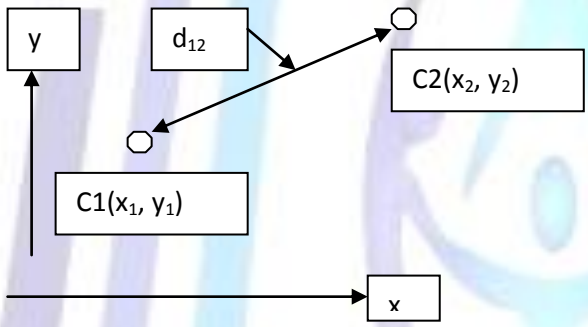

Fig1: Euclidean Distance

Using Pythagorus's Theorem, we may write $d_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ which is the Euclidean distance between the two cases, "in the x-y state-space". We often say that we have defined a "distance metric" when we have defined the formula for the distance between two cases. If the two cases are characterized by p continuous variables ($x_1, x_2, \ldots x_i, \ldots x_p$ say) rather than two (i.e. x, y), [Note: $x \rightarrow x_1$ so x(case 2) $\rightarrow$ x_1(case 2) $\equiv$ $x_1(2)$; similarly $y_2 \rightarrow x_2(2)$], then we may generalize the Euclidean distance to:

$$d_{12} = \sqrt{\sum_{i=1}^{p} (x_i(2) - x_i(1))^2}$$

This can be generalized further to the Minkowski metric:

$$d_{12} = \sqrt[m]{\sum_{i=1}^{p} | (x_i(2) - x_i(1)) |^m}$$

where |x| denotes the absolute value of x, (i.e. the size, without the sign).

A) K- Means Clustering

K-means clustering is an algorithm to classify or to group your objects based on attributes/features into K number of group is positive integer number. The grouping is done by minimizing the sum of squares of centroid[10]. Thus the purpose of k-mean clustering is to classify the data. The basic step of k-means clustering is simple. In the beginning we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroid or the first K objects in sequence can also serve as the initial centroids [3].

Step1: [Begin with a decision on the value of K=number of clusters.]

Step2: [Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

1. Take the first k training sample as single-element clusters.

2. Assign each of the remaining (N-K) training samples to the clusters with the nearest centroid .After each assignment, recomputed the centroid of the gaining cluster.]

Step 3: [Take each sample in sequence and compute its distance from the centroid of each of the clusters, if a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing sample.]

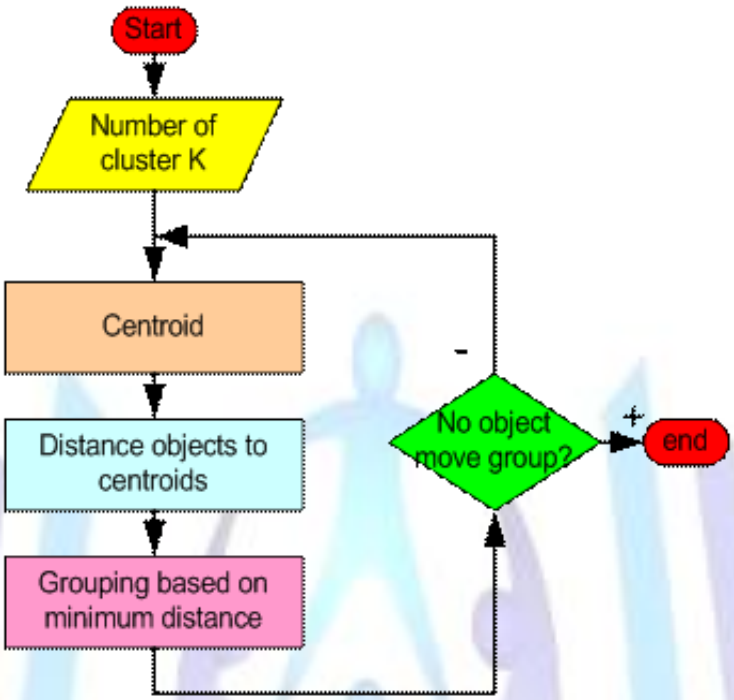

Fig2: Flowchart For K-Means Clustering

B) C-Means Clustering

Partitional clustering is an important part of cluster analysis. Based on various theories, numerous clustering algorithms have been developed, and new clustering algorithms continue to appear in the literature[10]. It is known that Occam's razor plays a pivotal role in data-based models, and partitioned clustering is categorized as a data-based model.

Algorithm:

Step1: [Fix the number of cluster.]

Step2: [Randomly assign all training input vector to a cluster .this creates partition.]

Step3: [Calculate the cluster center as the mean of each vector component of all vectors assigned to that cluster. Repeat for all cluster]

Step4: [Compute all Euclidean distances between each cluster center and each input vector.]

Step5: [Update partition by assigning each input vector to its nearest cluster with minimum Euclidean distance.]

Step6: [Stop if the center do not move any more otherwise loop to step3, where is the calculation of a cluster center.]

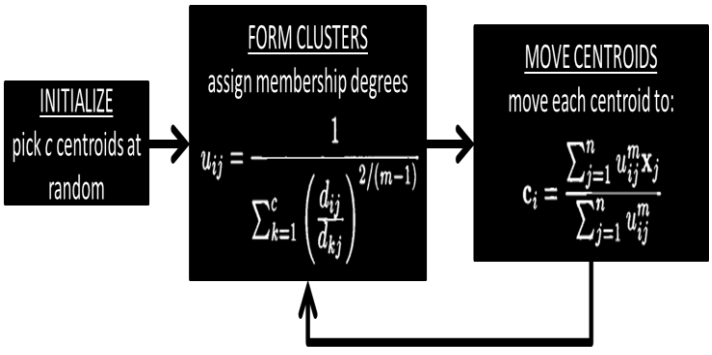

Fig3: Flowchart For C-Means Clustering

C) Hierarchical Clustering

A hierarchical algorithm yields a dendrogram representing the nested group of pattern and similarity levels at which grouping change dandles at which corresponding can be broken at different levels to yield different clustering of the data[5]. Most hierarchical clustering algorithm is variants of the single-link Complete-link and minimum-variance algorithm. Of these the single-link and complete-link algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distance between all pairs of pattern drawn from the two clusters. In the complete-link algorithms, the distance between two clusters is the maximum of all pair wise distance between patterns the two clusters.

Algorithm:

Step1: [Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.]

Step2: [Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to d[(r),(s)] = min d[(i),(j)] where the minimum is over all pairs of clusters in the current clustering.]

Step3: [Increment the sequence number : m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)]

Step4: [Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old Cluster (k) is defined in this way: d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]

Step5: [If all objects are in one cluster, stop. Else, go to step 2.]

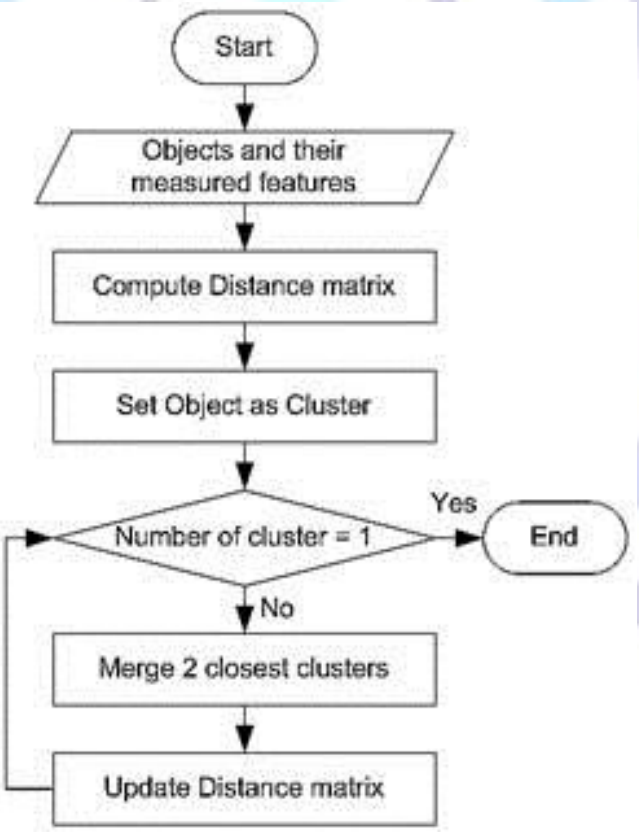

Fig4: Flowchart For Hierarchical Clustering

4. CONCLUSION

In order to improve quality of software development, we can make use of different Clustering techniques. This paper reviewed the software defect management based on different types of defects by using clustering algorithms.

5. ACKNOWLEDGEMENT

Sincere Thanks to HCTM Technical Campus Management Kaithal-136027, Haryana, India, for their constant encouragement.

6. REFERENCES

[1] P.V.Ingle, M.M.Deshpande (2013) " Software Quality Analysis with Clustering Method" *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA.*

[2] Sakthi Kumaresh, "Defect Analysis and Prevention for Software Process Quality Improvement" *International Journal of Computer Applications (0975 – 8887) Volume 8– No.7, October 2010*.

[3] PuneetDhiman, Manish, RakeshChawla" A Clustered Approach to Analyze the Software Quality using Software Defects"2012

[4] R.Karthik, N.Manikandan,"Defect Association and complexity prediction by mining Association and Clustering Rules" Volume 7, 2010.

[5] Deepak Gupta, VinayKr.Goyal, Harish Mittal"Analysis of clustering Techniques for software quality prediction".

[6] Megan Graham, 2005, Software Defect Prevention using Orthogonal Defect Prevention. http://twinspin.cs.umn.edufiles/ODC_TwinSPIN

[7] Suma V and T R Gopalakrishnan Nair , 2008, " Effective Defect Prevention Approach in Software Process for Achieving Better Quality Levels" Proceedings of World Academy of Science, Engineering and Technology, Volume 32 August 2008.

[8] Soni 2006, "Defect Prevention: Reducing Cost and Enhancing"Quality,*isixsigma.compublishers,*http://software.isixsigma.com/library/content/c060719b.asp

[9] Suma, V., Gopalakrishnan, T.R., (2010). "Impact Analysis of Inspection Process for Effective Defect Management in Software Development", *American Society for Quality (ASQ) Journal, Software Quality Professional, USA, Vol. 12, No. 2.*

[10] Mrs. Bharati,(2009) "A Comparative Analysis of Fuzzy C-Means Clustering and K Means Clustering Algorithms", *International Journal Of Computational Engineering Research / ISSN: 2250–3005*

Authors' biography with Photo

Ms. Rajni Rana, M-Tech (Computer Science and Engineering) from Haryana College of Technology and Management, Kaithal, Haryana,India.