

Adaptation of Six Sigma Method in Software Development

Aditya Pai H^{1*}, Chimaya Dash², Prakash Behera³, Piyush Kumar Pareek⁴

¹Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru and Research Scholar, VTU Belgaum, Bengaluru, India

^{2,3}Dept. of CSE, Claret College, Jalahalli, Bengaluru and Research Scholar, JJTU, Rajasthan, India

⁴Department of Computer Science and Engineering, EWIT, Bengaluru, India

DOI: <https://doi.org/10.26438/ijcse/v7si15.309314> | Available online at: www.ijcseonline.org

Abstract—The six sigma is used in targeting the business process of the organization by reducing and fixing the defects being occurred by driving the six standard deviations among the mean and its specification limit, which is why it is known to be six. The research study of the paper suggests the application aspect in the software development that are proven to be eliminating the final defects from the overall products be delivered thus by reducing the variations.

Keywords—Agile, Lean, Six Sigma, Software Development.

I. INTRODUCTION

Software industry got energy these days. The reality that it has showed up a very long time subsequent to assembling gives it the opportunity to get profit by strategies and practices which are now demonstrated and utilized in assembling of rehashing the wheel.

The diverse idea of software industry presents therequirement for hypothetical examination of assembling strategies before applying them in software advancement.

Assembling process relies upon monotonous execution of errands to create comparable items. Design and assessment of these errands are performed amid beginning times; this makes producing a human driven process by then of time. After effective execution of assignments, the nature of producing will in general be operational.

Center is coordinated towards delivering items with business needs. This business objective is accomplished by persistently checking and controlling activities to wipe out the reasons for deformities and varieties.

Before the finish of the 1970s, Japanese organizations beat their contenders by coordinating learning, abilities and advancements into their industry [1]. This let them rapidly convey great items with lower costs. Motorola was one of the contenders and it chose to period and it was effective in executing it.

The effective usage of the Six Sigma program in Motorola prompted a few celebrated organizations following Motorola in actualizing the Six Sigma [1]. In this manner, it was

additionally effectively actualized by General Electric and Texas Instruments [2].

To accomplish Six Sigma, a process must not create more than 3.4 imperfections per million open doors [2]. A Six Sigma imperfection is characterized as the disappointment of gathering client details. Then again, open door in Six Sigma is the absolute amount of chances for an imperfection [3].

This paper is organized as pursues: Six Sigma philosophy and its methodologies are introduced in area II.

At that point, the joining of Six Sigma in software industry is examined in segment III. From that point onward, the system of this paper is clarified in area IV and the outcomes are exhibited in segment V. The paper is closed in segment VI; this is pursued by impediments and future work in segment VII.

II. SIX SIGMA METHODOLOGY

Six Sigma is an assembling procedure expects to lessen deformities and varieties in items by improving forms that produce these items [3]. There are numerous structures for Six Sigma usage relying upon the circumstance. To improve a current undertaking the Six Sigma critical thinking system: Define, Measure, Analyze, Improve and Control (DMAIC) is utilized. For new tasks, the most famous six-sigma system utilized is: Define, Measure, Dissect, Design and Verify (DMADV) [4].

DMAIC is utilized when a procedure as of now exists yet the item made from this procedure does not meet client determinations. Then again, DMADV approach splendidly coordinates the circumstance when new procedure is to be

created or after the disappointment of DMAIC approach in meeting client particulars [4].

The extent of this paper is to improve presently existing forms this why we will depict the periods of DMAIC approach in more subtleties in the accompanying segments.

A. Characterize

In the characterize stage, the issue to be comprehended is distinguished what's more, issue proclamation is made. Additionally, the group to work in the execution of the improvement activities is chosen. Improvement plans are set up with their estimations; too, the procedures against which the improvement is performed are characterized [5].

B. Measure

Measure stage helps in evaluating the issue by estimating the present condition of the procedure that is destined to be improved. The consequences of this stage will be utilized later to look at the old condition of the procedure with the new state it comes to after the improvement is connected [5].

C. Dissect

This is the stage in which underlying driver examination procedures are connected in order to recognize the reasons for the imperfections. In resentment of the significance of this stage to the successful execution of next stages; it is some of the time skipped. Individuals think of it as a time squandering action however the expense of poor investigation is much higher than the expense of performing appropriate examination. Off-base results lead to the creation and execution of off-base improvement plans [5].

D. Improve

This is the period of arrangement advancement and execution. It begins by meetings to generate new ideas to create however much arrangements and thoughts as could be expected. At that point the results are ordered and assessed to encourage the choice procedure. Basic leadership depends in money saving advantage examination and information accumulation to guarantee the choice of the best also, best arrangement. After the choice is made, the arrangement is created and actualized [5].

E. Control

This stage is performed to ensure the maintainability of the improvement that was made to the procedure. One of the exercises performed in charge stage is looking at the new condition of the procedure with the past state to gauge the improvement. Moreover, checking and controlling plans are created to distinguish any adjustment all the while in order to promptly take restorative activities [5]. The periods of DMAIC approach are appeared in Fig.1 [1]

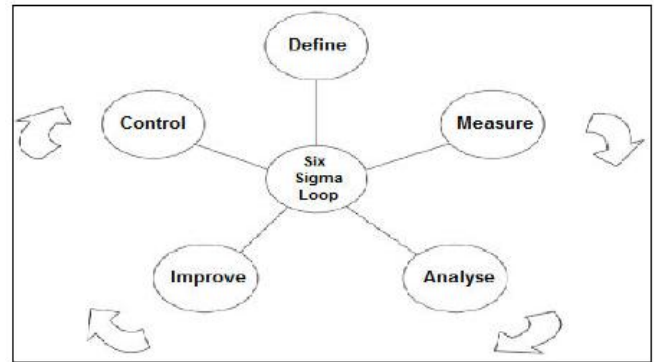


Fig 1. Implementation of DMAIC

III. SIX SIGMA IN SOFTWARE INDUSTRY

Programming improvement life cycle comprises of four stages: Prerequisites, Design, Development and Testing. Numerous programming approaches and practices developed to meet a few explicit requirements or to upgrade the productivity yet the previously mentioned stages are played out; the thing that matters is in their method for execution, dimension of covering and requesting [6].

The utilization of Six Sigma into programming advancement requires a great deal of research and examination to think about the idea of programming advancement and Six Sigma independently at that point finding whether it is gainful for programming and how it can be connected.

A. Programming Development as Human Centric Process
Programming improvement is the way toward structure programming frameworks beginning by gathering and examining their prerequisites at that point planning and actualizing them and at long last perform testing to guarantee they fill in according to their determinations [6]. This procedure is upheld by documentation and support exercises. All these forms unequivocally subject to the human factor; this why abilities like activity, innovativeness and development are exceedingly acknowledged in the field of programming advancement. This is not at all like the assembling that comprises of tedious procedures which depend on machines and talented labourers.

B. Programming Development as Business Process
The reason for programming designing is to create programming based frameworks to enable clients to accomplish their business objectives [6]. Consequently, programming advancement isn't a disengaged specialized procedure yet rather it is one of the procedures by which we convey a result that fits clients need and accomplish our business objective.

C. Six Sigma Compared to Agile and Lean
There is perplexity when we talk around Six Sigma, Deft and Lean approaches. They have something in regular yet at the same

time every single one of them has an alternate set of attributes that particularly recognizes it. Deft programming improvement strategy centres in compelling change the executives and client contribution. It plans to constantly creating and conveying working programming [6].

It is a human driven procedure since it removes the expense of executing overwhelming procedures and replaces it with compelling correspondence and cooperation between the colleagues [6]. The iterative idea of lithe is like the methodology taken by Six Sigma in taking care of value issues.

On the opposite side, lean advancement plans to dispose of squander and coordinates the exertion toward the execution of the forms that effect and enhance the last item [7]. Lean was first presented in assembling by Toyota at that point received by programming industry. Six Sigma likewise indicates squander end however it considers imperfections and varieties as the waste [7].

IV. METHODOLOGY

The reconciliation of Six Sigma in programming improvement requires escalated examination and examination for key forms in programming improvement life cycle to approve their reasonableness as focus for Six Sigma Implementation. The point of receiving Six Sigma in programming improvement is to lessen the bugs and to lessen the varieties in the nature of the last programming items. In assembling, the centre is in decreasing the varieties in the items themselves however in programming industry variety isn't viewed as a restriction yet rather an advantage since it brings an esteem required by the client to fulfil a certain need. Accordingly, the decrease of variety in the nature of the programming items is focused on paying little heed to the kind of that item.

In this exploration, key procedures in programming improvement life cycle are right off the bat chosen; at that point for each chosen procedure, profound examination is performed to address two inquiries. The first inquiry is: What are the periods of Six Sigma that can be coordinated into this chosen procedure? Remembering that we limit our examination to DMAIC approach of Six Sigma.

It isn't down to earth to delineate period of DMAIC to as it were one procedure inside programming improvement life cycle in light of the fact that there is some covering of those stages when they are converted into procedures; this covering is acquired from the idea of programming improvement.

The second inquiry is: How to productively install DMAIC stages into the focused on programming process? In responding to the second inquiry, current practices and

techniques of each chosen procedure are examined and assessed against DMAIC application. The assessment criteria vary starting with one procedure then onto the next dependent on the Six Sigma esteem that the procedure conveys. In addition, the ones that best fit the utilization of DMAIC are featured.

A. Programming Architecture

Programming engineering is a lot of abnormal state choices made amid the advancement and development of a product framework [6]. In DAMIC usage, both investigate stage furthermore, improve stage targets programming engineering. The examination performed in investigate stage to distinguish the root reason for the bugs is material to the dimension of programming engineering itself. This since certain bugs are early infused in the product amid the product engineering structure process because of awful decisions and choices.

Then again, in the improve period of DAMIC, if the main driver of the bugs is in the product engineering at that point changes are to be made against that design to fix the issue. In this manner we will talk about normal programming models and look at their appropriateness for the application of Six Sigma (break down and improve stages). For each building design considered in this paper, we centre on addressing two inquiries:

- How simple to research and investigate the bugs?
- How simple to adjust the design to dispose of the found issues that caused the bugs?

1) Micro-administrations Architecture Pattern: This example comprises of independently conveyed units that are inexactly coupled. Every unit can be called administration part also, it epitomizes at least one modules; those modules together convey the business rationale or administration required [8].

The separation highlight of this design makes it simple to investigate the main driver of bugs; knowing the capacity conveyed by a specific segment courses our assessment to that part on account of an imperfection in the capacity related with that part. The upkeep and redeployment of the product in this design is additionally simple in light of the autonomy of the segments and the well-defined interfaces and correspondence style between them [8].

Moreover, testing will be effective since no full relapse for the entire framework is required in the wake of altering just single part an issue that diminishes the expense of change.

2) Event Driven Pattern: Occasion driven is a famous circulated non concurrent design used to create very versatile applications [8]. It is comprised of profoundly decoupled, single purpose occasion handling parts that get and process

occasions [8]. It is hard to explore and investigate the underlying driver of bugs utilizing this example on the grounds that what's more, outer rationale might be expected to create explicit arrangement of occasions to recover the issues; a similar trouble is looked in testing stage. Then again, it is anything but difficult to convey in light of the fact that of the way that occasion preparing units are detached and self-contained.

3) Layered Architecture Pattern: It is a flat structure of layers in which each layer conveys a particular usefulness to the entire engineering. Investigating of bugs is simple since the layers of this engineering are consistently isolated and each layer as it were performs explicit pre-characterized task [8]. This sensible division helps in rapidly partner a certain bug with the layer it has a place with by applying consistent mapping. In addition, the adjustment is additionally simple in view of the idea layer of the engineering don't affect segments in other layers [8]. Shockingly, the arrangement isn't simple since we can't send just the piece of the application in which the change happened yet rather we have to re-convey the entire application.

4) Plug-ins Architecture Pattern: This example is made out of centre framework with the capacity of adding extra highlights to that centre in the type of modules. This gives extensibility quality while keeping the seclusion between the modules and the centre framework and between the modules themselves [8]. The autonomy between the modules helps in deformities examination since you can without much of a stretch expel the modules one after another until you discover the abandoned region. This aides in both investigating and testing. Hot organization of the recently created modules can be accomplished utilizing this example an issue that encourages the change. Then again, if the centre module is absconded then it is a significant issue since the primary operational rationale of the framework is brought together in this module and the modules just adds highlights deeply [8].

B. Programming Construction

Dissect and improve periods of Six Sigma require programming development (coding). In the dissect stage, in profundity examination is done to address the main driver of the issues. This examination can begin by code survey and white box testing yet it might require composing a bit of code or adjusting condition in the purpose of re-creating the issue to distinguish it.

The utilization of legitimate logging rationale helps in rapidly following what's more, understanding the reason for the issue while the utilization of standard and generally utilized instruments and outsider items ensures the accessibility of help and online web journals. On the other hand some programming ideas like reflections for instance

make the examination procedure somewhat troublesome since some portion of the code is made at run time and can't be followed through code audit. In the improve stage, coding is required for executing the arrangement that fixes the found issues. To support this process the selection of utilization improvement systems is exceptionally prescribed. Notwithstanding, coding rehearses acquired from lithe like pair programming and outrageous programming chops down the time required for investigating and fixing the deformities while delivering top notch programming. Moreover, test driven technique is a decent counterpart for Six Sigma application since those two offer a similar objective of dispensing with the imperfections.

C. Programming Quality Metrics

Programming quality measurements are a subset of programming measurements that attention on the quality parts of the item, process, and venture; however they are all the more intently connected with procedure also, item than with venture [9]. They are partitioned into three classes: item quality measurements, in procedure quality measurements also, support quality measurements.

The measure stage, improve stage and control period of DMAIC depend on programming quality measurements in their application. In this area we address which sort of programming quality measurements is a decent counterpart for each period of DMAIC usage.

In the measure stage, the present condition of the procedure is assessed to be utilized as a benchmark when performing control stage later. The nature of the product item that is produced from a specific arrangement of procedures mirrors the nature of these procedures; this why programming quality measurements will be utilized as a pointer for the procedure quality. Imperfection thickness is the most valuable quantitative measurement at this stage. Also, unwavering quality measurements like interim to disappointment and mean time to recuperate can help in the assessment. In the improve stage upkeep measurements are utilized since they demonstrate how simple to adjust the code and how simple to expel the deformities. A similar programming quality measurements that have been utilized in measure stage can likewise be utilized later in the control period of Six Sigma to envision how the alterations connected to the procedures all through programming advancement life cycle had improved the nature of the last items. Deformity thickness when utilized here will deceive since it treats every one of the deformities along these lines. It is smarter to give weight to each desert preceding computing the deformity thickness be that as it may, this can't be connected without performing examine stage.

D. Programming Testing Methodology

Testing is the way toward discovering deficiencies in programming and estimating whether it produces right yield or not [6]. At the point when Six Sigma is coordinated in programming advancement, testing turns into a pivotal piece of measure, break down and control stages. In performing measure stage; testing is executed to portray the present condition of the procedures. Since testing is performed against the items that are created utilizing the process under investigation, we just need to apply discovery testing at that point utilize the outcomes later for benchmarking. Dissect period of Six Sigma additionally includes programming testing to grow better comprehension of the reason for the emission; dark box testing is required in this stage to recognize the product region that contains the deformity before conveying the outcomes to advancement for additional inside and out examination. In the control stage we apply a similar dimension of testing that has been connected in measure stage; this other than relapse tests and non-practical tests to ensure that the constant changes does not have sway on the in general non-useful highlights of the framework.

V. RESULTS

In the past areas of this paper the reconciliation of Six Sigma in programming engineering plan, programming development, programming quality measurements and programming testing were analysed. Fig.2 graphically demonstrates the relationship between programming improvement exercises and the periods of Six Sigma DMAIC usage. Likewise, Table.1 condenses the outcomes acquired from the investigation.

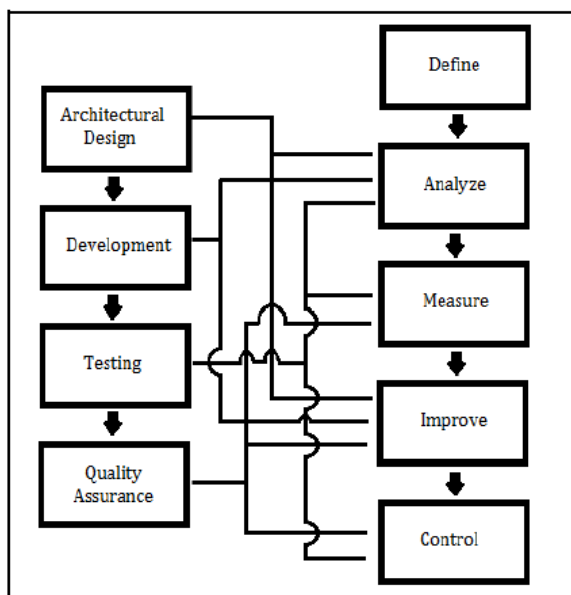


Fig 1. Software Development Using DMAIC

Table 1. Suggestions to eliminate the necessary failures

Software Process Stage	Phase of the Six Sigma	Tool Used and Practices Approached
Software Architecture Stage	Analyze and Improve	The Plug-ins architecture
Development Stage of the Software	Analyze and Improve	The Application framework for development is used for adopting the standards
Quality Metrics of the software	Analyze, Measuring and Controlling	New set of metrics is used for the six sigma
Testing of the software	Measuring, Improving and Control	Black box testing

VI. CONCLUSION

This paper reviewed the use of Six Sigma in programming improvement forms. It has been presumed that the determination of a suitable arrangement of programming forms helps in the smooth incorporation of Six Sigma system. Disregarding the distinction between programming advancement procedure and assembling process in certain viewpoints; they share a similar business objective of gathering client details.

This why performing top to bottom investigation of assembling strategies then re-forming them to fit programming advancement will emphatically affect the product business. Programming measurements are of incredible advantage for Six Sigma application in programming advancement; this on the grounds that there is a rich suit of measurements that spread every one of the procedures inside programming advancement life cycle and can give knowledge search for each procedure to follow how Six Sigma improves the nature of that specific procedure, at that point how that influences the nature of the last programming item.

REFERENCES

- [1] Tauseef Aized, "Total Quality Management and Six Sigma", First Edition, Croatia, InTech, 2012.
- [2] George Eckes, "The Six Sigma Revolution", John Wiley & Sons, USA, 2001.
- [3] Dr. Amit Gupta and Yogesh Bhardwaj, "Total Quality Management & Six Sigma Instrument", IJMBS, Vol. 3, Iss. 4, Dec 2013.
- [4] Rafa E. Al-Qutaish and Khalid T. Al-Sarayreh, "Applying Six Sigma Concepts to the Software Engineering: Myths and Facts", 7th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems (SEPADS '08), University of Cambridge, UK, 2008.
- [5] Antonio Carlos Tonini, Mauro de Mesquita Spinola, Fernando Jose Barbin Laurindo, "Six Sigma and software development process DMAIC improvements", Presented at: Technology Management for the Global Future [online], available at: <https://www.researchgate.net>.

- [6] Ivan Marsic, “*Software Engineering*”, Rutgers University, USA, **2012**.
- [7] Jiju Antony and Maneesh Kumar, “*Lean Six Sigma: Research and Practice*”, BookBoon.com, UK, **2011**.
- [8] Mark Richards, “*Software Architecture Patterns*”, First Edition, O’Reilly Media, USA, **2015**.
- [9] Informit.com, “*Software Quality Metrics Overview*”, [online] Available: <http://www.informit.com/articles/article.aspx?p=30306>, **2002**.

Authors Profile

Mr. Aditya Pai H received the BE degree in Computer Science and Engineering in VTU, Belgaum in 2009 and pursued his masters MS degree in Information Technology and Management Manipal University, Manipal in 2012. He is currently pursuing PhD degree in Computer Science and Engineering under VTU, Belgaum at East West Institute of Technology Research Centre, Bengaluru. He has 7 years of teaching experience. Currently working as Assistant Professor in Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru. He has two books being published in Lambert Publishing, Germany. He is also the member of ISTE. His research area is Software Engineering.



Mr. Chinmaya Dashis currently working as assistant professor in St. Claret College. Also pursuing his PhD degree in Computer Science and Engineering in JJTU University in Rajasthan. His Area of Specialization is Software Engineering.



Mr. Prakash Beherais currently working as assistant professor in St. Claret College. Also pursuing his PhD degree in Computer Science and Engineering in JJTU University in Rajasthan. His Area of Specialization is Software Engineering.



Dr. Piyush Kumar Pareek is a Ph.D (2016) in Computer Science Engineering from Jain University, Bengaluru, M. Tech (2012) from Dayanand Sagar College of Engineering, Bengaluru & B.E (2010) from Basaveshwar Engineering College, Bagalkote, Author has over 8+ Years of Academic Experience, He is author or co-author of over 53 international journal papers or conference proceedings, He has Four Books published in Lambert Publications, Germany. He is BOS Member – Alumni – ISE - of Basaveshwar Engineering College, Bagalkote, He is PhD Alumni Convener of Jain University, Bengaluru. He is Recognized Research Supervisor in Computer Science & Engineering (VTU0817401), Visvesvaraya Technological University, Balagavi. He is currently serving in department



of Computer Science & Engineering at East West Group of Institutions, Bengaluru. His Interested areas of Research are Software Engineering, Computer Networks & Machine Learning. He is Editorial Board Member & Reviewer of International Journal of Advanced Research in Computer Science & Technology (IJARCST), Brittany, FRANCE. Reviewer of International journal of Engineering Research and Technology ESRSA Publications (IJERT), INDIA, International journal of Computer Science Research and Technology (IJCSRT), INDIA, Inter Science Research Network (IRNET), Bhubaneswar, INDIA, International Journal of Computer Science and Informatics (IJCSI), Bhubaneswar, INDIA, International Journal of Engineering, Sciences and Management (IJESM), U.P, INDIA., International Journal of Scientific & Technology Research (IJSTR), Paris, FRANCE, Journal of Theoretical and Applied Computer Science (JTACS), Szczecin, Poland, International Journal of Advanced Research in Computer Science & Technology (IJARCST), INDIA, Journal of Harmonized Research in Engineering, INDIA., IASTER’s International Research Journals, INDIA, IJMAS Journals, Bhopal, INDIA, RR Journals, INDIA, Reviewer of IRD INDIA & STEM Journals, INDIA