



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SUCCESSFUL PROCESS MANAGEMENT IN ORGANISATIONS BY USING SPI APPROACH

¹Author- Puppala Krupa Sagar, Scholar at CSE department and

²Author- Dr. Pankaj Kawadkar, Professor at CSE department both from Sri Satya Sai University of Technology and Medical Sciences-Sehore, MP

ABSTRACT

SPI exercises in programming affiliations are as frequently as conceivable acted considering all around portrayed reference models distinguishing issues is a crucial part in all product cycle improvement. A lot of variable accepts huge part in programming process. The rule reason for this work is to establish the framework for SPI by discussing the piece of programming, the changes in programming advancement systems and programming improvement process models, and the quality turn of events. The main aim of this study is to discuss the Successful Process Management in Organizations by Using SPI Approach. In addition, SPI experiences are analyzed. This designing exploration technique gives a practical method for managing SPI strategy headway research, as a result of its in-manufactured considered constant turn of events and change. Moreover, it slowly makes and evaluates SPI a strategy, intertwining means to achieve the recently referenced essential accomplishment factors. The investigation relies upon a couple of modern context oriented examinations.

Keywords: Management, SPI, Software, Programs, Organization etc.

1. INTRODUCTION

Software Process Improvement (SPI) focuses on upgrading the time, cost and nature of designing and the board practices in programming affiliations. SPI exercises in programming affiliations are as frequently as conceivable acted considering all around portrayed reference models, for instance, CMMI and ISO 15504. The consequences of collecting projects, gadgets, and procedures to be used to plan and execute improvement projects might be described as the SPI framework. SPI techniques are depicted as a part of or in conjunction with well-known cycle improvement frameworks such as ISO/IEC 15504 and CMMI. SPIRE, SATASPIN, PRISMS, MESOPYME, MoProSoft, and MPS are other SPI structures that illustrate SPI methods. The

related elements of the two widely used reference models, ISO/IEC 15504 and IDEAL, depict the SPI architecture from a broad perspective. They highlight the meaning of organizing SPI works out, name the critical activities and ideas and portray their relations. Notwithstanding unequivocal portrayals of reference models, instruments and methods and kinds of assessment techniques, the current

structures don't give a point by guide SPI system toward plan and administer SPI exercises. In the methods portrayed, the relationship of key ideas and utilization focal points are passed on to the experts.

They prescribe SPI projects to be figured out in course like lifecycle and in a best down way. Thusly affiliations every so often don't use these approaches and present their own unique approach to handling SPI excursions. We developed BG-SPI (Bilgi Group Software Process Improvement Methodology), which provides clear direction on the most effective method for leading SPI, covering the lifecycle, errands, approach, resources, devices, and other supporting resources. Using BG-SPI and the help of free SPI experts, organisations can easily plan and manage the SPI lifecycle by implementing a ready-to-use SPI plot. BG-SPI, in contrast to traditional philosophies, provides a planned technique for controlling lead process improvement in current rises with an appropriate long stretch aide. BG-SPI combines SPI

works out's rapid and dirty inscriptions. moreover process resources and documentation. Affiliations that utilization BG-SPI take out the de-inspiration to lead SPI brought about by dubiousness, achieve at this very moment benefits and overall enable SPI dares to be driven even more suitably. BG-SPI is utilized to lead SPI practices in ten particular programming relationships with different characteristics. SPI experts from Bilgi Group have been incorporated to coordinate the use of the methodology. It is watched that utilizing an organized SPI framework redesigns the SPI application according to various perspectives. In this paper, we explain the BG-SPI and present the benefits found in the SPI activities of the affiliations.

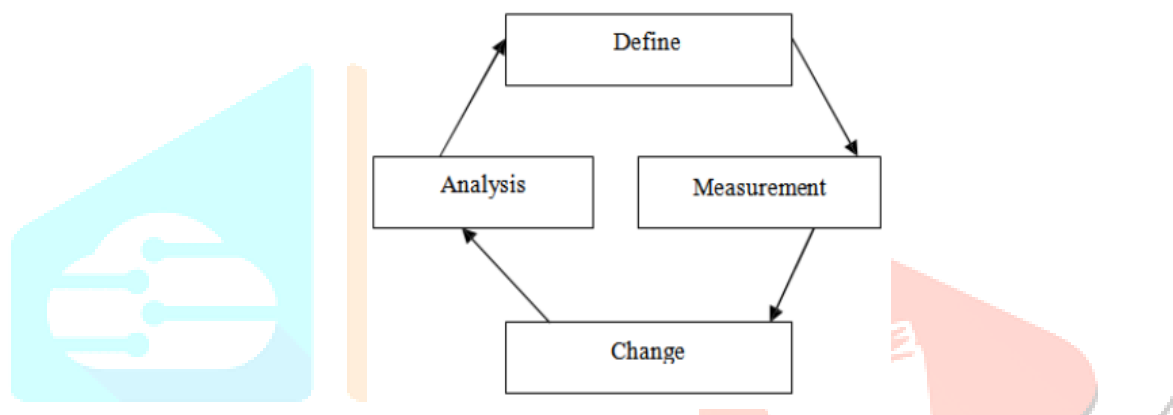


Figure 1: A Software Life Cycle

Programming process improvement term is used for working on the association. There are a couple of activities for this;

- Perceive the current cycle
- Perceive the current status of the association
- Perceive the strength and inadequacy of the association
- Separate where the movements are required
- What will be the effect of these changes?
- Comprehend these movements that the sum they are impacted

A couple of stages for the interaction improvement are following

- Careful for association culture
- Consistence should be there
- Make target and recall

- Put an option simultaneously
- Set up a well correspondence partner and senior administration
- Deal with process improvement like a task
- Make changes according to the need
- Make a profile of the association
- Present a product interaction planning gathering (SEPG) to your association
- Have constancy
- Reuse the previous outcome

2. SOFTWARE PROCESS IMPROVEMENT MODELS

To realize the product cycle improvement, investigators proposed a couple of models which help in understanding the methods which joins into the improvement of programming. The models which help in the improvement of programming process are

known as programming process improvement models.

- (1) **Six Sigma:** Six sigma procedures were planned by Quality Engineer, Bill Smith at Motorola in 1986. This thought came when the CEO of Motorola, Bob Galvin was attempting to improve the collecting system. They observed that their quality was adequately awful and hence, accepted the six sigma method to improve the quality. Further, Jack Welch made it major to his business approach at General Electric in 1995.
- (2) **IDEAL Model:** The IDEAL model is delivered by Carnegie Mellon University's Software Engineering Institute (SEI) was founded in 1996 to improve programming processes. It is a method for managing continuous improvement that combines a few stages to create a successful improvement framework. This model contains five phases for example beginning, diagnosing, and developing, acting and using stage.
- (3) **Spice Model:** ISO/IEC supported an assignment under the functioning title SPICE (Software Process Improvement and capacity assurance) in January 1993. It fuses the participation of overpowering world specialists in the field of programming. The guideline reason behind this endeavor was to conclude a norm for concluding the limit of its relentless improvement and programming creation process. Standard ISO/IEC 15504 or SPICE gives a framework to the assurance of programming processes. This design can be used by affiliations drew in with contriving, executing, checking, controlling and upgrading the satisfaction, supply, advancement, activity, estimation and backing of programming.
- (4) **CMM Model:** Ability Maturity Model (CMM) for Software Engineering Institute (SEI) of Carnegie Melon University created programming in 1991. As it assists the affiliation and gives a compact portrayal of how they with canning upgrade the cycle to make programming. It ends up being an important advancement in the product interaction the executives. There are five advancement levels of the product CMM-

Initial, Repeatable, Defined, Managed and Optimized.

- (5) **ISO Model:** ISO represents International Organization for Standardization. The term was singled out the grounds that it implies "equivalent" in Greek, so the connection expected to communicate the chance of uniformity. It is a norm for quality administration system. Its affirmation is imperative to upgrade the affiliation. Standard assurance a few fits properties of items and administrations, for instance, quality.

2.1 Software Process Management

Two focuses of programming process the board are to understand the usefulness and ampleness that outcome from an efficient way to deal with achieving programming processes and creating work items be it at the individual, project, or hierarchical level and to present new or further developed cycles. Processes are changed with the assumption that another or adjusted cycle will work on the capability as well as feasibility of the interaction and the nature of the subsequent work items. Changing to another interaction, working on a current cycle, hierarchical change, and foundation change (advancement addition or changes in mechanical assemblies) are immovably related, as all are normally started to work on the expense, improvement schedule, or nature of the product items. Process change has impacts not only for the product item; they frequently lead to hierarchical change. Changing a cycle or presenting another interaction can have steadily extending impacts generally through an association. For instance, changes in IT foundation contraptions and development frequently require process changes. Existing cycles might be changed when other new cycles are passed for the principal run on through (for instance, presenting an investigation movement inside a product advancement undertaking will likely effect the product testing process-see Reviews and Audits in the Software Quality KA and in the Software Testing KA). These circumstances can likewise be named "process headway." If the adjustments are expansive, then, at that point, changes in the hierarchical culture and plan of action will presumably be important to oblige the interaction changes.

2.2 Software Process Infrastructure

Laying out, carrying out, and overseeing programming cycles and programming life cycle models regularly occurs at the degree of individual programming projects. Regardless, orderly use of programming cycles and programming life cycle models across an association can give benefits to all product work inside the association, despite the fact that it requires responsibility at the authoritative level. A product interaction foundation can give process definitions, approaches for deciphering and applying the cycles, and portrayals of the technique to be used to do the cycles. Furthermore, a product cycle foundation might give subsidizing, contraptions, preparing, and staff people who have been allotted obligations in regards to laying out and keeping up with the product interaction framework.

The programming process framework changes depending on the organization's size and unpredictability, as well as the activities it undertakes. The foundation necessities of small, direct organizations and undertakings are minimal. Large, complicated organisations and projects, by necessity, have larger and more perplexing programming process frameworks. In the last option case, different hierarchical units might be laid out, (for instance, a programming interaction bundle or a guiding leading body of legal administrators) to direct execution and improvement of the product processes.

A common misperception is that laying out a product interaction framework and carrying out repeatable programming cycles will add time and cost to programming improvement and upkeep. There is an expense related with presenting or further developing a product cycle; in any case, experience has shown that carrying out methodical improvement of programming cycles will overall outcome in lower cost through superior adequacy, evasion of patch up, and more solid and reasonable programming. Process execution appropriately impacts programming item quality.

2.3 Software process assessment and improvement:

Models for evaluating programming processes, strategies for evaluating programming processes, models for improving programming processes, and

continual and arranged interaction appraisals Programming process evaluations are used to evaluate the structure and content of a product cycle, which may be demonstrated using a standardized collection of metrics. Instead of process evaluation, the terms "process examination" and "capability assessment" are used on occasion. Ability assessments are ordinarily performed by an acquirer (or possible acquirer) or by an outside specialist for the benefit of an acquirer (or expected acquirer). The outcomes are used as a sign of whether the product processes used by a supplier (or expected supplier) are OK to the acquirer. Execution evaluations are regularly performed inside an association to recognize programming processes needing improvement or to decide if a cycle (or cycles) fulfills the measures at a given degree of interaction capacity or development.

2.4 Steps to Continuous Improvement

Most quality professionals understand the implied Deming's "PLAN-DO-CHECK-ACT" persistent improvement cycle. To be sure, it is found at the focal point of most improvement tasks, and programming process improvement projects are not unique cases. The truth is that product development today is a sporadic process, and the organisations that produce it are frequently a wonderful combination of people, competing requirements, and other imperatives. Product interaction improvement projects emerge in this environment, and some succeed while others fail. Thusly, typically, programming process improvement drives should pass a more than intellectual, business case based channel that will just let through worth full opportunities.

At times, it isn't simply the high monetary costs the will make a central really reconsider proceeding with a product cycle improvement drive. Person's issues can likewise prompt numerous cerebral pains in term of hierarchical change and opposition. Remembering this, it most likely won't be an exercise in futility to underline the significance of selling the interaction improvement drive at every single hierarchical level, in solicitation to acquire the fundamental administration responsibility. The 7 phases presented in figure, when fittingly followed, will prompt the effective execution of CMMI based improvement drives.

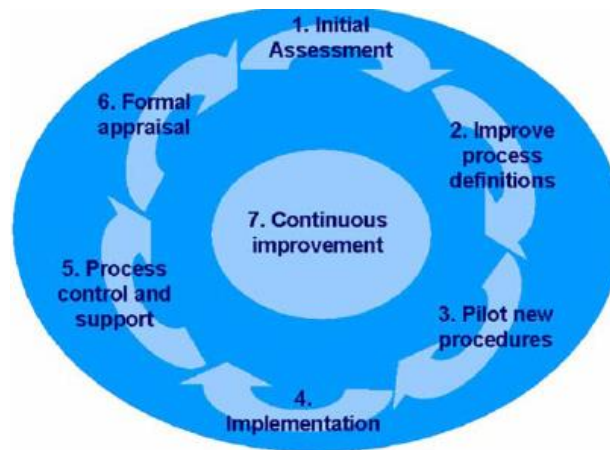


Figure 2: 7 steps to continuous improvement

3. SOFTWARE PROCESS IMPROVEMENT APPROACH

Perpetual interaction and thing improvement are the goals of any NASA programming process improvement effort. To get this impartial result, the programme employs a three-stage interaction approach: Understanding, Assessing, and Packaging. These stages are tirelessly executed in any advancement condition inside the affiliation.

- **Phases 1—Understanding** In the Understanding Phase, the alliance's cycle and things are portrayed and weird state objections for improvement are perceived. The motivation driving this stage is to ceaselessly get the properties of the item cycle and things inside the task connection and make models, affiliations, and general portrayals of the communication and things. Understanding is the important beginning stage of the general association improvement course of action, and it never closes, since changes should dependably be perceived and portrayed. Without this standard of the collaboration, things, and condition, not a great reason for change or improvement exists. An affirmation for change and improvement can be made and quantitative targets set precisely when the characteristics of the propelling framework and things are gotten and comprehended.
- **Phase 2—Looking over** In the Assessing Phase1, explicit objections for improvement are set, something like one changes are brought into the current collaboration, and the developments are then poverty stricken down to evaluate their effect on both thing and cycle. Change may unite presenting a procedure, gadget, or the chief's approach. This stage for the most part is considered as the test experience in which some

portrayed change to the cycle is thought about in contrast to the action. The developments are inspected through explicit assessments drove in picked experiences.

- **Phase 3—Packaging** In the Packaging Phase, changes that have made tasteful outcomes and shown quantifiable improvement are standardized and joined into the standard of the connection. Amidst this stage, the experts foster new models, rules, and arranging materials thinking about what has been recognized amidst the Assessing Phase. The things made by the reviewers are dealt with by the thought staff into a vault (i.e., a trouble base) and are given to the organizers upon ask. Packaging ordinarily intertwines rules, approaches, and handbooks; preparing; and instruments. For approaches or frameworks that don't show any incredible effect amidst the Assessing Phase, comes about must notwithstanding be gotten and archived (i.e., bundled) so the corporate memory is dependably moved along. This packaging may join reports or papers that are kept up in the corporate vault. The possible results of the packaging stage are taken care of back to those people expected with base-covering before the going with related undertaking or test. As such a specific improvement can be evaluated through different appraisals, every one fostering the bundled consequences of the past experiment(s).

3.1 Overview of the SEI CMMI-SW/SE:

The CMMI-SW/SE can be thought of as a collection of "best-practices" compiled over a number of cycle regions. Each interaction region has one nonexclusive and explicit goal to achieve, as well as a set of linked regular and clear practises. An orderly portrayal and a persistent portrayal are offered as two structures in the

model. Process areas are assembled in five development levels in the ordered presentation. Level 1 - Initial-doesn't contain any interaction region. Level 2 - Managed - revolves around fundamental venture the board. Level 3 - Defined - revolves around the normalization of authoritative cycles and acquaints different interaction regions related with designing exercises, board process, and a more awaited task for the executives Level 4 - Quantitatively overseen - is made up of process areas that deal with quantitative executive exercises. Last but not least, Level 5 - Optimizing - is obsessed with continuous progress.

The consistent portrayal is aimed at organisations that don't need or want to operate with the preset set of development levels that the organised portrayal requires. Regions are divided into four categories in the nonstop depiction process: project the board, process the executives, design, and support. The project board is an important part of the CMMI-SW/SE framework. The point of convergence of CMMI-SW/SE (arranged portrayal) level 2 is on fundamental undertaking the executives. This implies that the beginning stage on the development improvement adventure is laying out an essential venture the executives' ability in the association that will later permit more discipline to be wide into the designing cycles. CMMI-SW/SE contains 6 cycle regions collected in the task the executives' class.

4. ISSUES IN THE IMPLEMENTATION OF SOFTWARE PROCESS IMPROVEMENT PROJECT

Programming neighborhood progressively careful that first class programming improvement cycles will convey an incredible item. The initiatives are propelling this understanding. 'The nature of an item is, for the most part, addressed by the nature of the interaction employed to gather it,' to put it another way. A similar report produced by Software Task would ultimately fail due to a lack of clarity that can be effectively addressed with a convincing administration. Various programming subject matter experts and experts have organised their endeavours on the interaction assessment of programming as a result of this agreement. Software Process Improvement and Capability Assurance (SPICE), BOOTSTRAP, and other programming process improvement (SPI) models ISO 9000 and the Capability Maturity Model (CMM) have been proposed to assist associations with achieving additional obvious outcomes by fusing showed norms and frameworks into their product cycle. Associations

that have used these principles maintained in ISO 9000 and CMM have, generally speaking, shown astonishing upgrades. For example, by "further developing its progression interaction as per CMM 'development', Hughes Aircraft worked on its usefulness by 4 to 1 and saved enormous number of dollars." This reported achievement has fundamentally prodded further the application and reinforcement of SPI drives in programming associations.

Software Process Improvements: The cycles for enormous extension programming advancement might be quite large and complex, and they could involve a variety of computer programmers, designers, and creators. As a result, these cycles are frequently difficult to identify, interpret, and sometimes even set up or standardise. To address these interaction-related issues, programming process improvement (SPI) emerges. The core misunderstanding behind SPI and the various SPI structures is that the quality cycle will have an impact on the item's nature. What is required for a quality interaction or a trained cycle is the place where it demonstrates the plan of concludes that would bring about lead consistent with those principles. For an interaction to be effective, people should consider it and should be prepared in it. In another words, to convey a quality or restrained interaction, it should be characterized, prepared and executed until it become totally mature and is persistently improving sporadically.

4.1 Resistance Factors in Software Process Improvements Projects

Of course, there are various difficulties and issues which have been represented by associations going through SPI projects. There are various elements which are impacting the execution of SPI projects. Then, at that point, essentially recognized principle issues examined most SPI execution which incorporate fluctuating goals, no imagined status of SPI project, incapably managed data, cloudy work transport and barely moved development.

Similarly, effective communication in the programming project development environment among planners, clients, and project managers is a critical factor in reducing project unhappiness. Using these perceived concerns and Beecham's assessment, as well as Brietzke and Rabello's work, as a foundation, this analysis has identified resistance elements that can be classified into two categories:

- Hierarchical elements and
- Project factors

The hierarchical variables related to issues inside the degree of the association and are usually under positioning chief's obligation as uncovered, meanwhile project factors related to issues with respect to the product project the board, for instance, arranging activities and asset scattering among others.

5. THE LIFE CYCLE OF OUTSOURCING PROJECTS TO IDENTIFY MAIN PROBLEMS IN PROCESS IMPROVEMENT

Programming Development Life Cycle is a framework that makes programming in the briefest time with the best and most minimal expense. SDLC includes an isolated arrangement for how to make, keep up with, change, and supersede a product framework. SDLC incorporates various stages, including - arranging, investigation, plan, building, testing, and sending.

➤ Phase of the Software development life cycle:

The product improvement life cycle characterizes the business standard inside the undertaking. In IT works out, SDLC enormously influences delivering an inside and out directed a lot of regulated development of an advancement experience while managing all of the essential specific methods required gathering an authoritative fulfillment. It includes stages by which everyone is subject to its past one; those stages are the preparation, the investigation, the arrangement, improvement, testing, and support stage. Those stages are analyzed immediately underneath.

• Planning Phase

Without a good course of action set up, it is hard to change everything to your task objectives and judge the aggregate of its expansions, characteristics, and difficulties included. The arranging is to guarantee the improvement goes smooth, and accomplish its ideal progression inside the limitation of time given.

• Analysis Phase

The necessities and execution of the product through its creation stages is essential to gathering process practicality. This stage dependably assists with knowing where you most certainly stay all the while and what makes the stuff go through the subsequent stage down the way.

• Configuration Phase

The arrangement stage trails the examination. The basic point in this stage is to make a solid, feasible arrangement of the product strategy. This assists with eliminating any flaws or mistakes that might frustrate the activity.

• Development Phase

Right when the arrangement is prepared, the improvement assumes control over nearby capable recording and data. This is a staggering stage where focus and clearness are inconceivably critical.

• Testing Phase

The testing stage is unavoidable as it looks at and peruses the product for any slips up and bugs that might raise an issue.

• Maintenance Phase

Here in this stage, the item is appropriately kept up with and upgraded as and when expected to make it progressively versatile to the objective market.

6. THE KEY PROCESS AREAS AND KEY ELEMENTS FOR PROCESS IMPROVEMENT IN SOFTWARE OUTSOURCING BASED ON SPI

Mechanical kinds of progress sway our life in different propensities, and control our technique for living in all areas. In the item improvement field, we can see the high spread of SSDFs any place on the world. These affiliations expect a critical part in the economy of different nations, where they foster a monster section of the fundamental programming applications, offer many open positions, and experience new types of progress. Incredibly, these affiliations are encountering issues connected with encouraging their item things as they are unaware of the essential programming progression best practices. The major motivation to this is that the greater part of them are including uncommonly assigned way for the

item improvement. Similarly, hypotheses firms have nonattendance of cognizance of the achievement factors of SPI and need more individuals to play out all the SPI works out. Therefore, they end up to be extremely far away from executing formal SPI standard models and standards, for example, ISO 9001 Series, ISO/IEC 15504 (SPICE), ISO/IEC 12207 and BOOTSTRAP, where these models and standards were made for tremendous and huge firms, extraordinarily disappointed and exorbitantly costly to possibly be finished by SSDFs.

Notwithstanding the way that the SPI standard models and guidelines are difficult to be executed directly by SSDFs, SPI in these associations is at this point possible through modifications of these models and norms. There are a few neighborhood drives of SPI which were created for SSDFs, for instance, OWPL in Belgium; ASPE-MSD in Brazil; PRISMS in Britain; iFLAP in Sweden; MESOPYME in Spain, MoProSoft in Mexico; and MPS in Brazil. Regardless, these drives are not sensible for SSDFs wherever on the world, as they were made ward on the characteristics, conditions, and foundations of firms in these specific countries where the models began. Likewise, the advancement of these drives relied upon working on the SPI traditional models or principles by choosing the suitable Key Process Areas (KPA) of SPI standard models or norms which are sensible for SSDFs in the specific country, without distinguishing the fitting programming improvement practices that would achieve overall quality level the necessity for a proper SDPIF for SSDFs. This improvement structure should decide how to do the tasks of further developing the product processes.

To the degree that "connection improvement projects," exercises can be portrayed as those momentary or transient undertakings expected to chip away at a cycle and achieving unrivaled execution in a key presentation sign of the business. Considering everything, what business pioneers care about is further creating results. That is reality: further created execution. This essential objective is created through projects, and expressly, through process improvement projects. "Further creating results" assembles that results are really surveyed; appropriately, process improvement projects are centered around dealing with key estimations of the business. In any case, regularly improvement projects are not useful. There are assortments of inspirations driving why these endeavors bite the dust: nonappearance of sponsorship, insufficiently picked estimations, packs not coordinating, and thoughts dependent upon

hunches rather than on information and genuine elements. Stating that there has been improvement when, to be sure, no confirmed improvement has happened is additionally amazingly standard. Such issues make it a lot harder for project bosses to procure realness and backing for future undertaking endeavors. There are, in any case, five key activities that, when finished really, can extraordinarily grow the conceivable outcomes of an actually run process improvement project that closes with progress results.

7. CONCLUSION

Programming movement is in reliable change. New programming improvement structures, techniques, cycles, and contraptions are reliably presented and taken being used. At the same time, the new development and importance of programming has sped up, and programming has transformed into a key piece of an entire degree of various things. Programming improvement frameworks are advancing also: all around the planet dispersed programming movement, utilization of business ready to move (COTS), and Open Source improvement are a few events of the most recent propensities. Reliably fixing contest has incited dense lead-time fundamentals and blend of re-took a stab at programming translations focused to unique business regions. Programming movement should be improved to address these difficulties - without giving up quality. To keep alert to date with change programming process improvement (SPI) ought to grow, as well, after some time.

All around controlled programming progress processes has become fundamental spot limit in various affiliations, enabling lavish programming improvement, quality appraisal, control, and assumption. In light of everything, further making programming improvement processes is referencing and complex endeavor. Different programming process improvement (SPI) techniques in the market offer help and heading, yet shockingly they just to some degree address factors observed key for gaining SPI headway.

This theory makes, presents and fights for the SPI strategies typifying credits organizing towards down to earth association improvement. As the outcomes, the theory takes out crucial accomplishment factors for SPI works on using SPI practices learnt. Also, it reliably makes and surveys SPI methods, entwining desires to achieve the actually alluded to major

accomplishment factors. The assessment relies on a few present day setting centered appraisals.

REFERENCES

1. Dyba, T.: Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context. In: Proceedings of the 9th European Software Engineering Conference, Helsinki, Finland, pp. 148–157 (2003)
2. Gerry Coleman and Rory O'Connor. 2008. Investigating software process in practice: A grounded theory perspective. *Journal of Systems and Software* 81, 5 (may 2008), 772–784.
<https://doi.org/10.1016/j.jss.2007.07.027>
3. Mariano Angel Montoni and Ana Regina Cavalcanti Da Rocha. 2011. Using grounded theory to acquire knowledge about critical success factors for conducting software process improvement implementation initiatives. *International Journal of Knowledge Management* 7, 3 (jul 2011), 43–60.
<https://doi.org/10.4018/jkm.2011070104>
4. Mishra, D., Mishra, A.: Software Process Improvement Methodologies for Small and Medium Enterprises. In: Jedlitschka, A., Salo, O. (eds.) PROFES 2008. LNCS, vol. 5089, pp. 273–288. Springer, Heidelberg (2008)
5. Oktaba, H., García, F., Piattini, M., Ruiz, F., Pino, F.J., Alquicira, C.: Software Process Improvement: The Competisoft Project. *IEEE Computer*, 21–28 (2007)
6. Oktaba, H.: MoProSoft: A Software Process Model for small enterprises. In: Proceedings of 1st International Research Workshop for Process Improvement in Small Settings, pp. 93–101. SEI (2006)
7. Rafaela Mantovani Fontana, Victor Meyer, Sheila Reinehr, and Andreia Malucelli. 2015. Progressive Outcomes: A framework for maturing in agile software development. *Journal of Systems and Software* 102 (apr 2015), 88–108.
<https://doi.org/10.1016/j.jss.2014.12.032>
8. Regina Albuquerque, Andreia Malucelli, and Sheila Reinehr. 2018. Software process improvement programs: What happens after official appraisal?. In Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, Vol. 2018-July. Knowledge Systems Institute Graduate School, San Francisco, USA, 560–565.
<https://doi.org/10.18293/SEKE2018-186>
9. Wangenheim, C.G.v., Weber, S., Rossa Hauck, J.C., Trentin, G.: Experiences on establishing software processes in small companies. *Information and Software Technology* 48(9), 890–900 (2006)
10. Yuki Alqadri, Eko K. Budiardjo, Alex Ferdinansyah, and Mokhammad F. Rokhman. 2020. The CMMI-Dev Implementation Factors for Software Quality Improvement: A Case of XYZ Corporation. In ACM International Conference Proceeding Series. Association for Computing Machinery, Asia, 34–40.
<https://doi.org/10.1145/3379310.3379327>

