SOFTWARE PROCESS IMPROVEMENT pdf

1: Software Process Improvement | Process Improvement Planning

CMM - Defined (Level 3) â€¢ The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization â€¢ All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

Training Certification The Rational Model is characterized by a set of software best practices and the extensive application of use cases. A use case is a set of specified action sequences, including variant and error sequences, that a system or subsystem can perform interacting with outside actors. The RUP has been developing since and can claim well over 1, user organizations. CMM is an organizational maturity model, not a specific technology model. Maturity involves continuous process improvement based on evaluation of iterative execution, gathering results, and analyzing metrics. As such, it has a very broad universe of application. The CMM is based on four principles: The process view tells us that a process can be incrementally improved until the result of that process becomes adequately reliable. Process maturity has distinguishable stages. The five levels of the CMM are indicators of process maturity and capability and have proven effective for measuring process improvement. Evolution implies that some things must be done before others. Experience with CMM since has shown that organizations grow in maturity and capability in predictable ways. Maturity will erode unless it is sustained. Lasting changes require continued effort. The five levels of the CMM, in order of developing maturity, are as follows: Level 1 Ad Hoc: Characterized by the development of software that works, even though no one really understands why. The team cannot reliably repeat past successes. Characterized by requirements management, project planning, project tracking, quality assurance, configuration management. Organization project focus and project definition, training program, integrated software management, software product engineering, intergroup coordination, peer reviews. Quantitative process management, software quality management. Defect prevention, technology change management, process change management. Note that level 3 already seems to be higher than most software development organizations attain to, and would seem to be a very worthy goal for any development organization. It would seem that embracing a multitude of systems and models has helped software developers in India take a rapid lead in product and process improvement, but still there is no silver bullet! ISO Software Development Guidance Standard This guidance standard is a guideline for the application of standards to the development, supply, and maintenance of computer software. Neither is it a certification process. It is a guidance document that explains how ISO should be interpreted within the software industry see Figure 1. Prudent compliance of ISO may result in the following benefits: It is not intended as a certification document, like other standards in the ISO series. Copies of the guideline can be ordered from the ISO in Switzerland. Also, many consulting firms have Web sites that present the ISO guidelines in a cogent, simplified, and accessible way. It allows software vendors to be certified for upholding the ISO standard after passing the required audits. As with other ISO standards, there is a great deal of emphasis on management, organization, and process that we will not describe in this brief overview. Rather, we will emphasize the ISO development procedures that control software design and development. These include the use of life-cycle models to organize and create a suitable design method by reviewing past designs and considering what is appropriate for each new project. The following three sets of issues are addressed: Preparation of a software development plan to control: Technical activities design, coding, testing Managerial activities supervision, review Design input functional specs, customer needs Design output design specs, procedures Design validation.

2: Software Process Improvement | Software Development Methodology Today | InformIT

Be aware of your organization's current www.enganchecubano.com of the significant forces that affect the success of your process improvement efforts is the culture of your organization.

Licence This is an open access article distributed under the terms of the Creative Commons Attribution License , which permits unrestricted use, distribution, reproduction and adaptation in any medium and for any purpose provided that it is properly attributed. PeerJ Computer Science 2: Abstract Software process improvement SPI has been around for decades: However, the sheer mass of concepts, approaches, and standards published over the years overwhelms practitioners as well as researchers. What is out there? Are there new trends and emerging approaches? What are open issues? Still, we struggle to answer these questions about the current state of SPI and related research. In this article, we present results from an updated systematic mapping study to shed light on the field of SPI, to develop a big picture of the state of the art, and to draw conclusions for future research directions. An analysis of publications draws a big picture of SPI-related research of the past quarter-century. Our study shows a high number of solution proposals, experience reports, and secondary studies, but only few theories and models on SPI in general. New and specialized frameworks account for the majority of the contributions found approx. Furthermore, we find a growing interest in success factors approx. Beyond these specific topics, the study results also show an increasing interest into secondary studies with the purpose of aggregating and structuring SPI-related knowledge. Finally, the present study helps directing future research by identifying under-researched topics awaiting further investigation. Introduction Software process improvement SPI; according to Humphrey, aims to improve software processes and comprises a variety of tasks, such as scoping, assessment, design and realization, and continuous improvement, e. However, SPI is a diverse field: In response, several tailored standard SPI models or custom SPI approaches are proposed, inter alia, to better address needs of small and very small companies, e. Moreover, since SPI is mainly a human endeavor, much research was spent to study human factors, e. Beyond, we find numerous experience reports, guidelines, and toolsâ€”all together providing a huge body of knowledge on SPI. Are there new trends and emerging approaches, and if yes, what are the new trends? What is the current state of SPI and related research after all? The field of SPI evolved for decades and provides a vast amount of publications addressing a huge variety of topics. Still, we see new method proposals, research on success factors, and plenty of experience reports. Yet, missing is a big picture that illustrates where SPI gained a certain level of saturation, what are the hot topics, and what are unresolved issues calling for more investigation? To better understand the state of the art in SPI, we aim to analyze the whole publication flora to draw a big picture on SPI. In this article, we present findings from an updated comprehensive systematic mapping study. Starting with a curiosity-driven study, in two stages, we conducted a broadband search in six literature databases and one meta-search engine to harvest SPI-related publications from the past 26 years, and we incrementally analyzed the resulting publications for publication frequency, research type facet, contribution type facet, and we categorized the found publications using a set of 40 metadata attributes. Our results show a constant publication of new approaches while evaluation of these proposals is scarcely available. Our data shows rare evidence and, notably, missing long-term and independently conducted replication studies. However, the data also reveals some still emerging topics, e. The present study is a substantial update of our initial study published in Kuhrmann et al. The update adds new papers to the result set, which now contains papes in total. Furthermore, we modified the data classification approach. To achieve higher precision, we defined 40 metadata attributes, and we applied these attributes to the dataset while excluding the focus type facet from the analysis cf. Finally, while our initial study aimed to identify major trends, in this article, we provide a more detailed analysis of the trends found using the new classification. The remainder of this article is organized as follows: Yet, available secondary studies mainly focus on investigating success factors, e. Some studies provide insights into selected SPI topics, as for instance: All these representatively selected studies address specific topics, yet they do not contribute to a more general perspective on SPI. Such general studies are scarcely to find. However, they select few studies of which they assume to be good representatives thus providing a limited picture only. In terms of analyzing the entire domain and providing new generalizable knowledge, Unterkalmsteiner et al. They conduct a systematic review for the purpose of synthesizing a list of evaluation and measurement approaches, which they also analyze for the practical application. The study at hand does not aim at generating generalizable knowledge for one or more SPI-related topics in the first place. The purpose of the present study is to draw a big picture of the current state of the art of SPI in general. That is, as there is no comparable study available, this article closes a gap in literature by providing a comprehensive picture of the development of the field of SPI over time and by summarizing the current state of the art. Research Design In this section, we present the overall study design. After describing the selected research method, we introduce the research questions, and describe the different instruments used for data collection and analysis, and the validity procedures. Research method In this study, we ground the overall research approach in the procedures implemented for our previously published initial study. In Kuhrmann et al. While carrying out the study update, we used and improved the methods applied, which was necessary to develop a strategy that allows for continuous study updates. Figure 1 shows the overall research approach for which we provide details in subsequent sections. Overview of the applied research methods in the initial study left part of the figure as well as in the study update procedure right part of the figure. The initial study was designed as a breadth-first search to cover the SPI domain as complete as possible. In February , we performed the study preparation, conducted a series of test runs, and refined the search queries iteratively. End of April , we conducted the main search, which resulted in about 85, hits. As we expected this large number of results and in order to support the dataset cleaning, we defined filter questions, which we applied to the initial result set. When the initial result set was cleaned, we performed a voting procedure to select the relevant publications from the result set. Based on this selection, we developed the classification schemas by manual sampling as well as tool-supported and harmonized the dataset e. Therefore, after having conducted and analyzed the initial study, we collected lessons learned and developed the update strategy. The outcome is shown in the right part of Fig. The update procedure was defined in August , and the actual update was performed from September to November In subsequent sections, we describe this new strategy, whereas the particular changes are documented in detail in the appendix of this article. Research questions Our objective is to capture the domain of Software Process Improvement SPI , to provide a continuously updated snapshot of the available publication pool, and to investigate research trends. Therefore, we define the following research questions: What is the general publication population on SPI? This research question aims to get an overview of the general publication pool on SPI. We are interested in getting information regarding publication count, frequency and, eventually, an overview of the different research type facets addressed by the found publications. What is the contribution population? Based on the found publications, we are interested in the addressed topics and major contributions e. The third research question aims at investigating the focus points addressed by SPI research so far, and to work out gaps as well as trends. This research question shall pave the way to direct future research on SPI. After the initial result set analysis, the query strings were critically reviewed and updated Fig. However, no new search terms were added, only the structure of the queries required some updates to address the new data source that serves as main input. In a nutshell, due to the change of the search engine, the main search strings S1â€”S8 were integrated with the context and filter queries, which were required in the initial study to help querying the different literature databases. Data sources and data format. In the present study, after reviewing the initial study designs and results, we looked for more efficient ways to fetch papers for the update and eventually opted for Scopus 1 as new search engine. Having executed the different queries, obtained data was merged into one spreadsheet that structures the data and contains the attributes shown in Table 1. The data structure shown in Table 1 follows the structure used in the initial study. Spreadsheet layout to collect, structure, and evaluate data.

3: Software Process Improvement Models - Wikiversity

What is Software Process Improvement? Software process improvement is usually one of the most important areas an organization considers when improving the overall performance of its business processes and practices.

The assessor validates this data to ensure it is accurate and completely covers the assessment scope. Process rating requires some exercising of expert judgment on the part of the assessor and this is the reason that there are requirements on assessor qualifications and competency. The process rating is then presented as a preliminary finding to the sponsor and preferably also to the persons assessed to ensure that they agree that the assessment is accurate. In a few cases, there may be feedback requiring further assessment before a final process rating is made. This is an elaboration of the process reference model PRM provided by the process lifecycle standards. Tools used in the assessment[edit] There exist several assessment tools. The simplest comprise paper-based tools. In general, they are laid out to incorporate the assessment model indicators, including the base practice indicators and generic practice indicators. Assessors write down the assessment results and notes supporting the assessment judgment. There are a limited number of computer based tools that present the indicators and allow users to enter the assessment judgment and notes in formatted screens, as well as automate the collated assessment result i. Assessor qualifications and competency[edit] For a successful assessment, the assessor must have a suitable level of the relevant skills and experience. The competent lead assessor defines when the assessment is successfully performed. There exist schemes for certifying assessors and guiding lead assessors in making this judgement. It specifies requirements for improvement programmes and provides guidance on planning and executing improvements, including a description of an eight step improvement programme. Following this improvement programme is not mandatory and several alternative improvement programmes exist. Target process profiles are particularly important in contexts where the organization for example, a government department is required to accept the cheapest qualifying vendor. This also enables suppliers to identify gaps between their current capability and the level required by a potential customer, and to undertake improvement to achieve the contract requirements i. Work on extending the value of capability determination includes a method called Practical Process Profiles - which uses risk as the determining factor in setting target process profiles. It has the support of the international community. Over 4, assessments have been performed to date. Major sectors are leading the pace such as automotive, space and medical systems with industry relevant variants. There have been many international initiatives to support take-up such as SPiCE for small and very small entities. This has been for several reasons: In addition there are methods available that adapt its use to various contexts.

4: PPT â€“ Software Process Improvement PowerPoint presentation | free to view - id: e0-Y2JiN

Process improvement is an enabler, not an end in itself. You must understand the business goals of your organization in order to implement an effective information technology (IT) process. For example, the goal of always being first to market with new functionality will require a process with a different emphasis than that of minimizing defects.

The point of the SE-CMM is not to indicate "who" does the kinds of things described in a particular process area, but to indicate that the work needs to be performed by someone regardless of their role. The goal of this research was to acquire an understanding of the benefits gained and problems confronted with in relation to IPD-CMM implementations. The result of these findings were collected in a database and published by Cusick. The Cusick study showed that the components of IPD-CMM Table 1 that comprise the model are relevant to the success factors of implementing the integrated product development model. IPD-CMM Components Description Focus on people and personal commitment Organisations committed to training and personal development, development tools that promote individual accountability and move away from forced control over employees, gain a high personal commitment to product success. An emphasis on planning Planning takes on increased importance due to individual involvement and interest A focus on measurement and processes Project performance measures are tracked to establish metrics for future projects and to fix current inefficiencies. Careful monitoring of the decision making processes Sometimes teams get stuck due to lack of decision making. Successful organisation closely monitors how decisions are made and use this data to monitor the health of the team. Leadership dedicated to IPD Leaders need to set and communicate clear, measurable goals and put effort to remove barriers that teams identify to achieving goals. The truth is somewhere in between. Over time this aspect has been de-emphasised however. The authors suggest that TSP can in fact be a catalyst for process improvement. Motivations for introducing TSP must be understood and buy-in from all parties must still be achieved. Criticisms[edit] The repositioning of TSP suggests that to some extent there were difficulties with how it was presented initially. Some are specific to the individual standards, others are shared concerns. Standards, in a sense, guarantee only that the standard has been followed. All standards have a reputation for being process and document heavy, and for getting "in the way" of the development process. For any of the standards, figuring out how they can be mapped to Agile methods is a significant challenge. Some work has been done on this already, but it is yet to achieve widespread adoption. In fact, it may never be widely adopted as it is not necessary. Standards need to remain relevant. The fact that these standards reflect what should be best practice for software engineering is frequently overlooked because the negative aspects of standards overshadow the positive. Reaching out beyond the often captive audience for standards that currently exist and addressing the ways that they can be realized using Agile methods is important for the future of standards. CMMI and associated standards are constantly under revision and there are continuing efforts to measure and improve the effectiveness of CMMI as well as exploring ways of introducing and achieving CMMI certification. However, there is a question over the relevance of CMMI. The process is difficult to Also, the rise of other apparently more effective, less expensive software processes and methodologies from the Agile world have to some degree left CMMI behind. Poppendieck and others are positively dismissive of CMMI. And certainly CMMI will always have an audience in the military and safety critical industries where standards are a requirement for doing business, and rightly so.

5: Process Impact -- Software Process Improvement Consulting and Education

â€ Software Process: The set of all tasks involved in the production and evolution of a software product Software Process Improvement: An Introduction Slide 8.

One of the significant forces that affect the success of your process improvement efforts is the culture of your organization. Process, organizational structure, and corporate culture all go hand-in-hand â€" change one and you will affect the others. A common mistake that organizations make is to over specify the processes that they intend to follow. Never forget that your goal is to produce working software that meets the needs of your user community and that your staff likely has a pretty good idea of how to do this although they could help with a little guidance from time to time. Give them just enough guidance for their needs. Align your software process with business goals and objectives. Do you intend to build a portfolio of applications that integrate with, and build upon, one another? If so then portfolio management is important. Do you intend to sell shrink-wrapped software to be used by millions of users? If so then architecture may be less important to you in favor of getting a product to market quickly. A retrospective is a process improvement meeting where you ask four fundamental questions: What did we learn? What should we do differently next time? What still puzzles us? Retrospectives can be simple 15 minute discussions or formal meetings over a day or two. The goal is to learn from your experiences. Always one to cause trouble, I then asked if the client had any similar documents from four or five years ago. Keep the real goal in mind. My experience has been that software processes, when applied intelligently, increase the productivity of developers. My experience has also been that when processes are applied less than intelligently, or when the paper pushers have too much influence within an organization, processes can also decrease your productivity. Organizations that keep the end goal in mind â€" that of developing, maintaining, and supporting software that fulfills the needs of their user community â€" will be successful with implementing software processes. Those that follow processes simply for the sake of doing so are likely to fail. Recognize that the fundamentals remain the same, the details vary. Contrary to popular belief, the fundamentals of software development have been known for many years. You need to perform requirements engineering. You need to model. You need to write code. You need to test. You need to perform change control. You get the picture. Every successful software organization will have a similar set of processes but the way that your organization brings them in and how they implement them will differ. Your requirements process may be slightly different than your competitors, but you will both have one that will generally do the same sort of thing. You need more than one process. You need different processes, or at least different flavors of your process, for different situations. Use the right process for the job. Run a trailblazer project to validate your new processes. Regardless of how well you define a process, no process is perfect. Treat process improvement like a project. Have an experienced project manager, ideally someone with experience in both process-oriented and object-oriented development. Define the requirements for your processes, model them, implement them, test them with a trailblazer project, and then improve the processes. Improve your processes in priority order. The reality of process improvement is that you cannot make all of the changes that you want to immediately; it is simply too great a change for your organization to absorb at once. Both of these organizations suggest that you prioritize the process improvements that your organization needs to make, expect that it will take several years to make the needed changes, and expect that you will experience difficulties while doing so. There are five maturity levels in the CMMI for a reason: The implication is that by knowing which aspects of a software process map to which CMM maturity levels you have a rough idea of the order in which you should introduce those processes to your staff. Experience shows that organizations that try to make immediate, large-scale process changes are likely to fail doing so. The reality is that it takes time, often several years, to permanently improve the productivity of your software development efforts. There is not a simple, quick fix to your problems. It is essential that you let others know what are you doing, why are you doing it, the business case for it, success stories, etc. Keeping them informed on how things are going, even if you are encountering difficulties, will help get them and keep them on board. There are various options for this: Trumpet your successes and share your lessons learned with the appropriate people. Accept that the big picture is overwhelming. Because of the complex nature of software development most developers specialize in one aspect of it and focus solely on that. This becomes a problem for organizations that wish to tailor a software process for their exact needs because when they put the individual pieces together the overall process becomes very large. Democracies do not always work, nor do dictatorships. Organizations that wish to reach consensus regarding their software process tend to flounder. Effective process improvement efforts seek consensus at some points and dictate things at other points. Identify the consumers and suppliers for each process. Every process has inputs and outputs, and you need to ensure that there is a supplier for each input and a consumer for each output. Fundamentally, if nobody is going to use an artifact that is produced by a given process then why bother producing it? You also need to look at collections of processes to see if the artifacts that they produce add value in combination. Defining a process is the easy part. Many organizations are very successful at defining a software process, often producing binders or web pages of documentation. Getting people to accept your new process, and making the changes that go along with it, will take significant time and effort to accomplish. Writing a process is the easy part, following it is the hard part. Staff your SEPG with actual practitioners. The people who best understand how to develop software are the people who are very good at developing software. Reuse existing process materials. A common mistake is to produce volumes of documentation describing your processes. Your goal is simply to describe your process materials to such a level that they can be given to a professional skilled in the techniques of that process so they can work the processes appropriately. Adopt processes because they make sense. If a process makes sense to you, and you believe it will add value to your effort, then adopt it. Hold everyone responsible for process improvement. Senior management must be willing to actively support and sustain process improvement, project managers must be held responsible for ensuring that their teams follow the defined processes, and developers must be held responsible for learning and then following the processes. This is often a difficult task because senior management often demands immediate results, whereas process improvement often takes years. Project managers resent diverting scarce resources from their projects, and developers often resent being told how to do their jobs. Bring in an expert to advise you. Process improvement is a complex and difficult endeavor, one for which you are likely to need help to accomplish. You can increase your chance of success by bringing in a consultant who has both a process background and an OO development background â€” someone who has been actively involved in a process improvement program and who has worked on large-scale, mission-critical software development projects using OO technology. Do not think that everyone is on board. There is likely to be a small core of people within your organization who do not want to use object technology for large, mission-critical projects, and these people will actively undermine your efforts. You need to identify these dissenters and work together with them to help them see the advantages of working with object technology and of following a set of defined process patterns to help in the development OO software. A fool with a process is still a fool. For your organization to be successful with a software process your software professionals will need to understand the processes, the concepts, the techniques, and the problem domain. Implementing a new process in your organization involves more than going out and purchasing a couple of new books and development tools. Develop a user guide for your process. You can make it easy for your staff to learn your chosen processes by providing a well-written overview of your process as it is to be implemented in your organization. In fact, this may be all the process material that you need. Progress will be slow at first, slower than you hoped or expected. Introducing a software process into an organization takes time â€” the required culture shift often takes years to complete. Too many process efforts run aground because of preconceptions forced on them by senior management, an overly burdensome documentation and review process, or unrealistic requirements to achieve consensus. Define your process early. The longer you leave process definition the bigger the mess you will have to clean up. The DAD framework is a people-first, learning-oriented hybrid agile approach to IT solution delivery. It has a risk-value delivery lifecycle, is goal-driven, is enterprise aware, and provides the foundation for scaling agile. This book is particularly important for anyone who wants to understand how agile works from end-to-end within an enterprise setting. Data professionals will find it interesting because it shows how agile modeling and agile database techniques fit into the overall solution delivery process. Enterprise professionals will find it interesting beause it explicitly promotes the idea that disciplined agile teams should be enterprise aware and therefore work closely with enterprise teams.

6: The Software Process Improvement (SPI) â€“ Reward or Risk - Mohamed Sami

Indeed, it is found at the core of most improvement projects, and software process improvement projects are not exceptions. The fact is that software development today is a complex activity and organizations where it is developed are also, often a complex mix of people, competing priorities and numerous constraints.

Mohamed Sami Most of the Software companies large, medium, small, or startup usually face issues in their software development projects and its delivery. The issues can vary from lack of documentation, lack of following the process, lack of process governance, lack of the integration and collaboration between the teams, lack of requirements traceability, lack of technology management, â€¦etc. Therefore, some methods and techniques started to exist to tackle the software process issues to suggest different improvements and identify issues and inefficiencies in the process. These methods became a standard which the companies can follow to improve their software process. Moreover, each method established its ecosystem, from providing the training and certificates for the method to provide consultancy to help companies to improve based on actual practices. In this article, It will be good to ask yourself if the software process improvement is a peril to have or a promise for a better change for the organization and to have a superior advantage in the market. For answering this, we will discuss in this article what is SPI? Software Process Improvement SPI methodology is defined as a sequence of tasks, tools, and techniques to plan and implement improvement activities to achieve specific goals such as increasing development speed, achieving higher product quality or reducing costs. This definition is combined from [1][2]. SPI can be considered as process re-engineering or change management project to detect the software development lifecycle inefficiencies and resolve them to have a better process. This process should be mapped and aligned with organizational goals and change drivers to have real value to the organization. SPI mainly consists of 4 cyclic steps as shown in the figure below, while these steps can be broken down into more steps according to the method and techniques used. While in most cases the process will contain these steps. Current Situation Evaluation This step is the initial phase of the process and it is mainly to assess the current situation of the software process by eliciting the requirements from the stakeholders, analyzing the current artifacts and deliverables, and identifying the inefficiencies from the software process. The elicitation can be conducted through different techniques. For example, individual interviews, group interview, use-case scenarios, and observations. The key considerations in this step to identify organization goals and ask the solution-oriented questions. Moreover, identifying the measurement using the GQM Goal â€” Question â€” Metric technique that will help in measuring the current status and measuring the effectiveness of the improvement process. Improvement Planning After analyzing the current situation and the improvement goals, the findings should be categorized and prioritized according to which one is the most important or have the most severity. We should observe what is the new target level of improvements should look like. Moreover, in this step, the gap between the current level and the target level should be planned in terms of a set of activities to reach that target. These activities should be prioritized with the alignment of the involved stakeholders and the organization goals, for example, if the project is using the CMMI model, the target could be reaching maturity level 4 and the company at level 3, in that case, the plan should be focused on the process areas and their activities which is related to that level of improvement with the alignment of the organization goal. Improvement Implementation In this step, the planned activities are executed and it puts the improvements into practice and spreads it across the organization, what can be effective at the 2nd, 3rd, and 4th step that planning and implementation could be an iterative way, for example, implementing improvement for improving requirements first, then implementing the reduction for testing process time, and so forth. This iterative way of implementation will help the organization to realize the early benefits from the SPI program early or even adopt the plan if there is no real impact measured from the improvement. The before improvement measures, after the improvement measures, and the target improvement measure. Measurement, in general, permits an organization to compare the rate of actual change against its planned change and allocate resources based on the gaps between actual and expected progress. There are a lot of motivators from different perspectives for companies, management perspectives, sales perspectives, employee perspectives, and others. I will mention here the most common motivators for SPI: Standardization and Process consistency To have a standard and practical process for software development mapped to organization goals and strategy. Cost Reduction To improve projects cost by enhancing the process and eliminate issues, redundancies, and deficiencies. Competitive Edge Being certified in CMMI for example, can put the company in higher competitive edge and make it gain more sales due to the evidence of existing mature software process based on standard method. Meeting targets and reduce time to market Meeting organization goals, projects delivery, quality standards, valuable products, professional documentation are outputs from SPI. Improve customers satisfaction Project delivery on time and based on the specification with high quality will improve customers satisfaction and improve the sales process. Job satisfaction, Responsibilities, and Resource Management Employees get job satisfaction from producing a good quality product and knowing what to do without workload and the time consumed to resolve conflicts or to eliminate issue due to an immature process. Automation and Autonomy Introducing tools to automate things and improve quality and ensure consistency. Moreover, enabling different employees to play different roles in the project. Proven outcome There is a lot of evidence for the value of SPI projects which are successfully implemented. Despite these motivators, companies may be afraid to go through SPI project because a lot of factors and some companies may already have sufficient process and have a good revenue and it is successful, is it a necessity for that kind of companies to pursue SPI project! I think not, if you already do not have a pain you do not have too, so, what about the other companies? What are the Demotivators for SPI? Similar to the motivators, demotivators can be taken from different perspectives, and here are the common demotivators for SPI and they are very correlated: Time pressure Due to the nature of the companies to deliver the projects on time, they faced a lot of time pressure which make it harder for them to dedicate time to the SPI project. While I see this as weakness and actually a driver for SPI. SPI took a long time and it is a costly process while It is necessary if you have issues as discussed before. Budget Constraints As we just mentioned SPI is a costly process, because it needs time and dedicated resources, and not only that but also skilled resources especially in SPI. Inadequate metrics Most of the small companies do not have metrics to measure and compare their progress or improvement which make it sometimes impossible to identify measure the improvements of the SPI. Lack of Management Commitment It is mainly because the management cannot understand the benefit from SPI and they do not fully support doing this change as well as the other factors like lack of resources, budget, time, â€¦etc. Staff turnover Sometimes, the company has a high staff turnover which can be an issue to impose the SPI culture change and this can lead to endless SPI. Micro Organization Some organization are very small and have very few resources. The SPI will be too big for that kind of companies. Bad Experience and lack of evidence for direct benefits Some organizations may face a bad implementation for SPI that made them do not want to be involved in the same issue once again or maybe from another organization who tried to go through the SPI project. Another demotivator that they did not have the proper orientation of SPI and the direct benefits in practice. What are the different SPI methods? Similar to the SDLC, SPI has a lot of methods and you can as well define your own method if it is effective or combine between more than one if you do not have any preferences or organization need to adopt a specific method. SPI methods mainly categorized into two categories: The table below contains examples of the methods in each category.

7: Enterprise Agile: Software Process Improvement (SPI)

The software process improvement methods described have been selected from ap- plication within the Siemens case study sites. Because of the diversity of application.

Behavior-driven development and business process management [13] Chaos model - The main rule is always resolve the most important issue first. Incremental funding methodology - an iterative approach Lightweight methodology - a general term for methods that only have a few rules and practices Structured systems analysis and design method - a specific version of waterfall Slow programming, as part of the larger Slow Movement , emphasizes careful and gradual work without or minimal time pressures. Slow programming aims to avoid bugs and overly quick release schedules. UP organizes the development of software into four phases, each consisting of one or more executable iterations of the software at that stage of development: Many tools and products exist to facilitate UP implementation. Process meta-models[edit] Some " process models " are abstract descriptions for evaluating, comparing, and improving the specific process adopted by an organization. Independent assessments grade organizations on how well they follow their defined processes, not on the quality of those processes or the software produced. ISO describes standards for a formally organized process to manufacture a product and the methods of managing and monitoring progress. Although the standard was originally created for the manufacturing sector, ISO standards have been applied to software development as well. Like CMMI, certification with ISO does not guarantee the quality of the end result, only that formalized business processes have been followed. This standard is aimed at setting out a clear model for process comparison. It models processes to manage, control, guide and monitor software development. This model is then used to measure what a development organization or project team actually does during software development. This information is analyzed to identify weaknesses and drive improvement. It also identifies strengths that can be continued or integrated into common practice for that organization or team. A variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One software development methodology framework is not necessarily suitable for use by all projects. Each of the available methodology frameworks are best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. Sometimes, contractors may require methodologies employed, an example is the U. A decades-long goal has been to find repeatable, predictable processes that improve productivity and quality. Some try to systematize or formalize the seemingly unruly task of designing software. Others apply project management techniques to designing software. Large numbers of software projects do not meet their expectations in terms of functionality, cost, or delivery schedule - see List of failed and overbudget custom software projects for some notable examples. Composed of line practitioners who have varied skills, the group is at the center of the collaborative effort of everyone in the organization who is involved with software engineering process improvement. A particular development team may also agree to programming environment details, such as which integrated development environment is used, and one or more dominant programming paradigms , programming style rules, or choice of specific software libraries or software frameworks. These details are generally not dictated by the choice of model or general methodology. Software development life cycle SDLC.

8: Software development process - Wikipedia

Software process improvement (SPI) has been around for decades: frameworks are proposed, success factors are studied, and experiences have been reported. However, the sheer mass of concepts, approaches, and standards published over the years overwhelms practitioners as well as researchers.

Acceptance of an evolutionary iterative and incremental approach. Expertise in the new technologies. If you examine the wide array of processes in the industry, you will most likely determine that you can tailor one or more to meet your assessed needs. They advise that uncertain requirements, complex technologies, and low employee turnover favor agile techniques, whereas larger teams and a diverse group of stakeholders favor more prescriptive methods. There are many processes in the industry today, and they are at varying degrees of maturity and acceptance. Pick one or more that will help you achieve your goals. Remember that you will still need to tailor your adopted process. Rarely will a process, even a widely accepted one like the RUP, be a perfect fit for an organization. If a process makes sense to you, and you believe it will add value to your organization, then adopt it. Do not implement processes just because someone else does. Process tailoring is best done in an iterative manner: Process Creation It is possible that no existing process which meets your needs is available; this is very unlikely, given the wide array of processes available today. However, it is possible that you will discover that you need to create specific, detailed procedures and guidance that reflect your exact needs. If groups are unsure about some aspect of a standard process, such as iterative development, then they should try it out. People often get so set in our ways that it becomes difficult to contemplate other ways of doing our work. If you try a standard process, you may be pleased with the results of the test and decide that you can use it after all. If this fails and you need to create your own process, by: Identify the issue s you need to address. Look for existing process material similar to what you need. Engage the appropriate people. Identify people within your organization who are currently fulfilling this need and work with them to identify what they actually do. This will result in processes that people accept and that reflect your environment. Write less process documentation than you think you need. The hard truth is that very few people will actually read the process in its entirety. After that, they may only read the appropriate guidance documentation and use your standard artifact templates, ignoring the rest of your documentation altogether. Take a just-in-time JIT approach. Process Deployment After you have a defined process by either tailoring an existing one or by creating a new one, you need to implement it. For instance, you may find it beneficial to define different tailorings of the process for green field new development and commercial off the shelf COTS implementations. The article Process Tailoring and Deployment Strategies covers deployment issues in detail. Support ing Project Teams After a software process has been defined and implemented, it must be maintained, and teams must be assisted in order to apply it. Immediately after deploying a new process, you should spend considerable time adjusting it, as you adapt based on lessons learned. Running a pilot project or two will definitely help you to identify major issues, but there will always be a need for improvement. This is normal; expect it. One way to identify "lessons learned" is to hold retrospectives. How often you hold them will depend on your comfort level with your process. I prefer to hold them at the end of each iteration so that the project team can tweak their process and learn from their experiences continuously. Some people will hold them at the end of a project, often referring to them as a project post-mortem, but I find this to not work very well in practice because the identified "lessons learned" often prove to be "lessons indicated" which are never acted on. The end is simply too late. Kerth recommends that retrospectives focus on four simple questions: What did we learn? What should we do differently next time? What still puzzles us? Remember that feedback is crucial to a process improvement effort. If something is not working, you need to know about it and adjust things as soon as possible. Do not assume that no news is good news. Motivate people to provide feedback if you must; consider an approach where people can provide feedback anonymously if necessary. The teams might not know enough yet to provide you meaningful feedback: In this article, you learned that: SPI is an ongoing effort; your process will evolve over time. You need to invest in, and support, SPI efforts for them to be successful. The goal of SPI is to ensure that your organization can define, implement, and evolve one or more appropriate processes to help you meet your IT goals. There are many proven software processes; one or more of them will likely help you to meet your needs. Ambler and Michael Vizdos. The extensions include two new phases, Production and Retirement , and several new disciplines: This book, Disciplined Agile Delivery: The DAD framework is a people-first, learning-oriented hybrid agile approach to IT solution delivery. It has a risk-value delivery lifecycle, is goal-driven, is enterprise aware, and provides the foundation for scaling agile. This book is particularly important for anyone who wants to understand how agile works from end-to-end within an enterprise setting. Data professionals will find it interesting because it shows how agile modeling and agile database techniques fit into the overall solution delivery process. Enterprise professionals will find it interesting beause it explicitly promotes the idea that disciplined agile teams should be enterprise aware and therefore work closely with enterprise teams. Existing agile developers will find it interesting because it shows how to extend Scrum-based and Kanban-based strategies to provide a coherent, end-to-end streamlined delivery process.

9: Software process improvement: a systematic mapping study on the state of the art [PeerJ]

In software engineering, a software development process is the process of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle.

Introduction Software process improvement SPI seems to be a commonly agreed silver bullet to solve the software crisis. The underlying assumption is that a good software process results in a high quality product, delivered in time and within budget. All SPI actions have one or several of the goals: An organisation initiating SPI work can choose among publicly available models for the purpose, e. Success stories and implementations of SPI are published, e. More and more organisations initiate some kind of SPI actions. An obvious value of the ESSI programme for the European software market is the sum of the values for the participating organisations. However, an additional value should be obtained through so called dissemination actions, spreading the experiences and lessons learnt, and preventing other organisations to invest in reinventing the wheel, or repeating the mistakes done by others. The objectives of EUREX are to collect, systematise, and disseminate the experiences and lessons learnt in the process improvement experiments. Naturally, there is a wide variety of PIEs along the dimensions of the subject domain, organisation type, working methodology, objectives of the experiment, to mention a few. The data collection in EUREX is by workshops, discussions in the context of the workshops and otherwise, and studies of the final reports. In a way, all the PIEs should be classified into the subject domain software metrics, as every PIE should include measurement to determine the magnitude of the improvement. However, although software metrics is an established field of software engineering in theory and research, its industrial applications are behind. Although the study was very limited, there was a clear indication of a major potential for the metrics maturity improvement in European software industry. Measurement is the vehicle for control. Without that control, software engineering will not reach the status of sound engineering discipline, but remains a craft. There is a need to point out the importance of measurement, to discuss how it should be conducted and how it should not. The Objectives of the Study The objectives of this study are to increase the awareness of sound software metrics in general, and in particular to point out how a number of PIEs have handled the metrics related issues. The framework for the study is software metrics theory and ESSI guidelines [4], described in Section 4 more in detail. Section 3 presents the basic issues of software metrics, both from theory and practical point of view. Section 4 presents the framework for the study and the results are presented in Section 5. Section 6 contains concluding remarks. Software Process Improvement Software process improvement is commonly agreed being the silver bullet to solve the software crisis of the organisations. It is argued that technology and people in an organisation change, whether we want it or not, but the processes, i. The stability should not imply stagnation, but controlled change. An organisation with stable processes should regard software process improvement as a continuous, never ending activity. Software process improvement can be studied from different aspects, e. In this paper, the context is process improvement experiments within ESSI programme, i. We are not aware of any corresponding publications from large European companies concerning organisation wide improvement or metrics programs. The PIEs with approved final reports are intended to be available on the world wide web [4]. The dissemination actions publishing aggregations and summing up of the experiences and lessons learnt are also publicly available as EC project results. The PIE projects are normally short term, months in duration, and intended to improve the software process of an organisation in some respect. If a project is successful, and welcomed by the staff, it may very well generate further improvement actions. If it is a failure, or if the staff does not see the gain for their working situation or business as a whole, further improvement actions may be hard to propose and to get accepted. Model for Improvement Software process improvement may be undertaken based on a general, publicly available model, e. The general models usually require a long-term investment and duration, longer than 18 months, which is the normal duration for a PIE. Therefore, the PIEs usually have a locally designed model, described in the work packages. The local model should ideally be extended by measurement, along the guidelines of ESSI. CMM or any other model can be used as a meta model, to show the way what should be in place. Every process on an adequate level of detail, should include the following: In addition to the processes, standards for the different work products may be defined, e. Within the PIE projects, there is naturally a management commitment for the activities, as ESSI programme only supports the half of the costs. The abilities in terms of resources is relevant within PIEs as well as in other software process improvement efforts. Experts can be used to cover a lack of competence in the organisation, for defining the process and in training the employees in the new process or in measurement related issues. Measurement and analysis intends to determine the status of the new process, whether it is followed and is functioning as intended. Verification is the same type of control from the management perspective. Software Metrics Software metrics, presented in various textbooks, e. Software metrics relies on the underlying theory, called representational measurement theory, posing some requirements on a correct definition, validation, and use of software metrics. From practical point of view, there are several further questions of importance, e. Software measurement is an activity assigning a number or a symbol to an entity in order to characterise a property of the entity according to given rules. The informal definition, even though giving an idea, must be more precisely defined. The message of the definition is that there should be an entity, a property, a measurement mapping and rules for the mapping. The measurement mapping and the rules is usually called metric. An example of an entity is code. An attribute characterizing the code is size, and one possible metric for measuring size of code is the number of lines of code LOC. Initially, there must be an intuitive understanding of the property of the entity of interest, otherwise there is no way to define an adequate metric. For example, for the entity person, we can intuitively understand the property length, which can be measured in inches or centimetres. If observing two persons, we usually get an understanding who is taller, i. The intuitive understanding can be represented in an empirical relation system, a pair consisting of the set of entities, and a set of relations, e. For the measurement, there must be a corresponding numerical relation system, a pair, with symbols representing the entities and numerical relations corresponding to the empirical relations. There is also a so called representation condition requiring that a measurement mapping must map the entities into numbers and empirical relations into numerical relations in such a way that the empirical relations preserve and are preserved by the numerical relations. The measurement mapping, the empirical and numerical relations are usually called the scale of the measurement. There are five different scales: It is important to establish the scale of the measurement in that different scales allow different manipulations with the measurement data. What is the thing being measured? There are three main classes of entities of interest for measurement in software engineering, namely product, process, and resource. Product is an output from a process, e. A process is one or several activities. Resource is an input to a process, e. Sometimes, we need to measure attributes for a global entity, namely the entire organisation, e. Unless there is a clear statement of the entity, attribute, and metric, it does not make much sense to talk about measurement. For example, the statement "the size is 20 measured in LOC" does not make sense unless we know the entity in question. Unless the attribute is defined, we do not know what property of the entity is supposed to be characterized by the metric. Unless the metric is defined, we do not know even the scale of the measurement, nor can we get an understanding of the relative value of the measurement. For example, the statement "The code size is " does not make sense unless we know if size has been measured in LOC or bytes, or something else. The Study The framework of the study is software metrics theory and ESSI guidelines for mid-term and final reports [4]. Below, those parts of the guidelines in some sense indicating measurement are given: Objectives Explain the method and specific metrics used to measure the impact on the business goals and to verify to what extent the problem has been solved. Phases of experiment Explain any specific training undertaken as part of the experiment as well as other internal dissemination activities. Consultancy during the experiment Explain the role of internal and external consultants, if any, on the project and the reasons why they were needed as well as the effort expended. Resulting scenario This section should detail the actual results obtained from the experiment and your analysis of them compared to the original objectives of the experiment. Include any qualitative and quantitative results and how they were measured. Provide details of any final assessment and, as far as possible, give real figures. Key lessons learnt This section should summarise the key lessons, positive and negative that you have leant to date from undertaking the experiment. It should identify clearly your key lessons leant, from the technological and business point of view. The interpretation of the ESSI intentions we have made is that the expected impact on the business goals should be indicated in quantitative terms and the obtained impact should be measured and compared to the expected, and the metrics used should be reported. Further, specific training if applicable, should be indicated. The specific training may be training is software metrics. Consultants are sometimes needed in a PIE, they may be partners with the required skills and competence, or their services can be bought from an external organisation. In that case, it is interesting to investigate to what extent the universities have participated in the role of metrics expert. Key lessons learnt are important messages to other organisations planning process improvement experiments. The following questions will be studied: Have the objectives been described in quantitative terms? Have any quantitative results been reported? Has metrics specific training been conducted? Has measurement related consultancy been needed? What are the metrics related lessons learnt? In addition to the above questions, we also investigate the distribution concerning the entity measured, i. Further, we present a set of measurements, which are well defined, and another set, which is unclear.

Transformative learning : a teaching/learning model for using global literature to positively impact our Iliad book 22 desolation before troy Influence of age and caffeine on resting metabolic rate, blood pressure, and mood state in younger and ol Approximation with rational functions Looking for Thelma Gillian Slovo Roy Blakeley, his story North dakota travel guide Red berets and red crosses Secret of Mountain Top Inn. Risking enchantment Determination of infiltration and percolation rates along a reach of the Santa Fe River near La Bajada, N Close er collections grade 9 answers Military science and technology Drug localization in tissues and cells Psychodynamic therapy for adults Harold S. Kudler . [et al.] Physics hans c ohanian Opposing Viewpoints in American History: From Colonial Times to Reconstruction (Opposing Viewpoints: Amer The case of the venomous voice Where Do I Go From Here? A Simone Dickerson Novel Wade Hampton and the Negro National formulary of Unani medicine. Small claims court guidebook Structuralist theory by ferdinand de saussure Ending, and beginning Introduction Ã la sociologie gÃ©nÃ©rale Grand Prix Motor Cycle Championships of the World, 1949-75 1995 International Conference on Acoustics, Speech and Signal Processing Pt. I. 1705 to 1741 III. Life prefixed to the commentary of Servius. Barker (1968 and Lewin (1936), ecological psychology also relies on Lehninger principles of biochemistry indian edition Groundwater 2000 W Complications in ophthalmic surgery V. 1. Indonesia, Andaman Islands, and Madagascar. My fact in the ATM screen ; The grapes of wrath analysis Crit Assess Jung V 3 (Critical assessments of leading psychologists) Metamorphoses of the body The taming of the C.A.N.D.Y. (continuously advertised, nutritionally deficient yummies! monster Discourses of social transformation