

Study of Software Maintenance and Maintenance Estimation Guidelines

P. Ashok Reddy, K. Rajasekhara Rao, M. Babu Reddy

Abstract: Programming maintenance is a costly motion that devours a noteworthy bit of the price of the total project. Special physical games finished amid assist encompass the expansion of new features, erasure of antique code, remedy of errors, and so forth. Programming practicality implies the ease with which those duties may be executed. Inside the event that the viability may be anticipated in early periods of the product development, it helps in better arranging and excellent resource utilization. Estimation of plan houses, for instance, coupling, attachment, and so on in early degrees of development regularly drives us to infer the bearing on viability with the help of prediction models. On this paper, we executed out a planned audit of the present day-day investigations related to software viability. In this exploration work we foresee the product unwavering best as a long way as time taken to preserve up the errors and enhancements. Counterfeit Neural network is applied to interrupt down and count on the product unwavering splendid of the renovation level. In the direction of the give up measurable outcomes and proposed neural gadget consequences are likewise contrasted with ensure that expected outcomes are identical to the yield effects.

I. ADVENT

The formal which means of 'programming help' given with the useful resource of IEEE [1] is alternate of a programming object after conveyance to deal with problems, to beautify the execution or other trends, or to alter to an altered area." software program practicality is defined because the gain with which those changes can be made" [2]. It has been observed that for the complete lifestyles cycle of the object, sincerely 30– forty% is expended in development and equalization 60– 70% is gobbled within the renovation of the object interms of property, time, coins and endeavors [2]. Marco [3] recommended that we can control this fee just via estimating it. Irrespective of the truth that it thoroughly can be estimated amid the maintenance level, however it is probably beyond the point of no return via at that point. Analysts are constantly trying to plot a manner to conjecture the viability of any product in early phases of programming development lifestyles cycle (SDLC) via estimating the capabilities of its

Revised Manuscript Received on December 22, 2018

P. Ashok Reddy Head Of the Department, Dept.of MCA, LakireddyBaliReddy College Of Engineering, Mylavaram, A.P

K. Rajasekhara Rao, Director, Usha Rama College Of Engineering, Telaprolu, Vijayawada, A.P

M. Babu Reddy HOD, Dept. of Computer Science, Krishna University, Machilipatnam, A.P

¹ashokreddimca@gmail.com

structure. In the ebb and waft methodical survey, all exploration papers, audit articles, whitepapers, reviews and techniques of conferences acknowledged to writers due to the fact 1990 to date have been gathered, investigated, accumulated and examined. The goal of the current review is to kind out experimental evidence in far reaching structure on the following aspect:

- . Considered one of a type variables which have an impact on practicality.
- . Wonderful techniques and strategies to beautify practicality.
- . The usage of forecast models for practicality in the early tiers of development.
- . Searching at the execution of various practicality forecast fashions in terms of the exactness.
- . Distinguish the focal points and weaknesses of various forecast fashions over each other.
- . Distinguish the product measurements which can be carried out in forecast reveal making process.
- . Distinguishing portions of evidence of the current holes for future prospect of research in the area of software viability.

II. MOTIVATION

On the identical time as task the triumphing survey, the 2 smooth inquiries turn out to be beneath:

- . For what reason is it the proper trouble for a survey?
- . Is there any ongoing survey conveyed here?

As of late, a pinnacle degree view led by Jones [4] observed out that amid the Nineteen Fifties only 10% of the complete experts conveyed in programming corporation were locked in in maintenance work, and constantly 2025 this parent may want to ascend to seventy seven%. Similarly, it claims that intense lack of programming work force is likewise because of the blasted of maintenance work. Consequently, experts are creating a first rate try and make possible software so that the general mission fee can be managed and the item may be managed optimally. As a top issue of the higher arranging,

frequently designers foresee practicality

Of the product primarily based on its structured attributes. This conjures up us to compile each one of the investigations inside the stated field so that it will distinguish what quantity has been achieved truly because the functionality zones of studies relying on the modern-day holes. Many reviews have been led in the past to set up the investigations recognized with software maintainability with the aid of way of Riaz et al. [5], Tieng et al. [6], Ghosh et al. [7], Saraiva [8] and Saraiva et al. [9], but this exam isn't pretty similar to every considered one of them in 3 views.

Proper off the bat, the winning examination is performed in step with the rules given via Kitchenham and Charters [10] and Pickard et al. For guiding an green review in the discipline of programming building. Except, in this audit, specific examinations on software practicality expectation (SMP) are shortlisted and dissected both qualitatively surely as quantitatively in classified structure for clean comprehension. A thorough studies changed into brought on cover one of a kind elements of programming maintainability predictions, for example, the forecast strategies, programming structure metrics, datasets, units, expectation precision of the fashions, and so forth. Thirdly and most importantly, not one of the surveys [5– 9] is as thorough as the prevailing one in which more than ninety six contemplates distributed from the 12 months 1991 until date are explored. Three. Software program preservation look at consequences

Amid the preceding yr, we had been analyzing the paintings that management programming life cycle bolster focuses perform to keep weapons frameworks operational. We have visited numerous workplaces and met greater than thirty weapons frameworks undertakings to accumulate our choices. The aftereffects of those endeavors are condensed because the software renovation paintings Breakdown form (WBS) that shows up as table 1. Each sizable motion completed in this life cycle arrange is separated into its part assignments indoors this desk at some component element workable. The purpose for this breakdown is to understand what maintainers do in every one of the real kinds of physical video games that are being completed.

This desk is essential since it features the manner that product individuals whole significantly some thing aside from programming upkeep as a difficulty in their entrusting in the ones management programming life cycle bolster focuses. This is important in mild of the truth that the charge fashions carried out for assessing these prices don't typically create forecasts that unfold the whole quantity of hard work protected. As a result, we've got got blanketed definitions for key programming preservation phrases in Appendix A. We gave those because of the fact we determined that the ones cutting-edge meanings of help as characterized in ISO/IEC 14764.2006 and IEEE benchmarks [2] simply secured the square discharge and retirement office work

1.0	Operations, Maintenance & Support	This entry collects the total cost associated with maintaining a system after it has been accepted by the customer (DD-250).
1.1	Maintenance	This entry summarizes the costs for updating and repairing elements of system.
1.1.1	Release Planning	This entry records the costs associated with developing plans, budgets and schedules for block releases.
1.1.2	Hardware Defect Repair	This entry records the costs for repairing hardware defects. Such costs include engineering and test.
1.1.3	Software Defect Repair	This entry records the costs for repairing software defects. Such costs include engineering and test.
1.1.4	Hardware Enhancements	This entry records the costs associated with developing hardware enhancements and making perfective changes. Such costs include both engineering and test.
1.1.5	Software Enhancements	This entry records the costs associated with developing software enhancements and making perfective changes. Such costs include both engineering and test.
1.1.6	Release Testing and Delivery	This entry records the costs for acceptance test and delivery of the release including the costs of integration and verification.
1.2	Sustaining Engineering	This entry summarizes the costs associated with sustaining operations in the field. It includes the costs of analysis and studies, emergency repairs and user handholding and support.
1.2.1	Analysis and Studies	This entry records the costs associated with the conduct of analysis and studies stemming from operational issues and problems.
1.2.2	Emergency Repairs	This entry records the costs associated with emergency repairs including those associated with development and delivery of patch releases to the field.
1.2.3	User Training & Support	This entry records the costs associated with providing user support and training.
1.2.3.1	Help Desk	This entry records the costs for manning and operating a help desk.
1.2.3.2	User Group Support	This entry records the costs for coordinating and conducting user group meetings and blogs and for maintaining a web site.
1.2.3.3	User Training	This entry records the costs for providing the training needed to maintain and operate the system as it evolves.
1.3	Independent Test & Verification	This entry summarizes the costs associated with independently verifying and validating the system as releases are prepared and released typically by third parties. Such verification activities can range from independent testing to detailed analysis of both designs and code on a separately maintained test-bench.
1.3.1	Test Planning	This entry records the costs associated with preparing test plans.
1.3.2	Test Preparation	This entry records the costs associated with developing test cases and scenarios and the related test tools needed to run them.
1.3.3	Test Conduct	This entry records the costs associated with conducting the tests, capturing results, verifying release requirements are satisfied and developing regression test baselines for use in revalidating the system when future changes are made.
1.3.4	Independent Analysis & Verification	This entry records the costs associated with performing the detailed analysis of designs and code needed to provide additional confirmation that requirements including those for security and safety have been satisfied.
1.4	Product Support	This entry summarizes the costs associated with maintaining the overall quality of the processes, products and supplier networks used by the system in operations, maintenance and support.
1.4.1	Configuration Management	This entry records the costs associated with configuration management including those associated with CCB operations and tracking configurations, spares, licenses and parts among various operational and support sites.
1.4.2	Quality Assurance	This entry records the Quality Assurance costs aimed at ensuring the quality and

3.1 Software maintenance analysis

Understand that product maintenance offers extraordinary specialised and the board troubles for programming engineers. Attempting to find out an imperfection in a 500K line of code framework that the maintainer did no longer create is a test for the maintainer. Additionally, contending with programming designers for belongings is a regular fight. Getting prepared for a destiny discharge, even as coding the subsequent discharge, and conveying crisis patches for the present discharge, is furthermore a check. The accompanying talks about a part of the specialised and the board issues figuring out with programming development and assist.

3.1.1 Technical problems

3.1.1.1 limited comprehension [Pfl98:c10s10.3; TG97:c3; DT97: c8s11.4] Practitioners and scientists display that some forty% to 60% of the upkeep exertion is dedicated to know-how the product to be adjusted. In this manner, the factor of software perception is one essential to maintainers. Interest is increasingly hard for content material cloth primarily based portrayal. Usually tough to conform with the improvement of the product through its adaptations, modifications aren't recorded, and the designers are commonly not round to clarify the code. On this way, maintainers have a confined comprehension of the product and need to get acquainted with the product all on my own.

Three.1.1.2 finding out [Pfl98:c10s10.3; Art88:c9] The price of rehashing complete checking out on a noteworthy little bit of programming may be crucial as a long way as time and coins. Relapse attempting out, the particular retesting of a framework or phase to affirm the changes have not triggered unintentional impacts, is essential to help. Studies endeavors into territories, as an instance, "decreasing" take a gander at this hassle. Discovering time to check is regularly tough [Pfl98]. Section five of the guide to the SWEBOK offers subtleties of testing.

3.1.1.3 Three effect exam [DT97:c8s10.1-3; Pfl98: c10s10.5; Art88:c3]The programming and the association must each enjoy sway research. Number one aptitudes, documentation, and processes are required for this territory. Effect research is critical for threat reduction. Programming supposed for viability encourages sway examination. Three.Three.1.4 Maintainability [ISO14764:s6.8s6.8.1;Pfl98: c8s8.4;Pig97:c16] The IEEE laptop Society [IEEE610.12] characterizes practicality due to the fact the simplicity with which programming may be saved up, upgraded, adjusted, or redressed to satisfy indicated necessities. ISO/IEC characterizes viability as one of the satisfactory attributes. Practicality highlights need to be consolidated into the product development exertion to lower life cycle expenses.

On the off threat that that is completed, the man or woman of development and aid of the code can enhance. Viability is regularly an trouble in aid considering that practicality isn't joined into the product improvement way, documentation is missing, and software program consciousness is tough. Practicality can be finished through incorporating it in conditions, shape, and development.

3.1.2 manipulate

Three.1.2.1 Alignment with authoritative troubles [DT97: c8s6; Pfl98:c10s10.3] Dorfman and Thayer [DT97] relate that arrival on project isn't always easy with assist. Subsequently,

there is a regular war to gather assets. 3.3.2.2 Staffing [Pfl98:c10s10.3; Dek92:pp10-17; Par86: c4s8-s11; DT97:c8s6] renovation artwork stress regularly are seen as peasants [Pfl98] and clear up endures [DT97]. Help isn't always visible as exquisite artwork. Deklava gives a rundown of staffing associated troubles dependent on have a study records [Dek92].

Three.1.2.Three method issues [DT97:c8s3] protection requires a few bodily activities that are not observed in programming development, (e.G., help paintings area help). Those present troubles to the board [DT97].

3.1.2.4 Organizational factors of upkeep The organization that builds up the product isn't constantly used to hold up the framework as soon as it is operational. A maintainer ought to be identified and there are a few alternatives as examined beneath.

3.2.4.1 The Maintainer [Pfl98:c10s10.2; Pig97:c2s2.5; Par86: c4s7; TG97:c8] often, a completely unique organization (or maintainer) is applied to guarantee that the framework runs legitimately and develops to satisfy changing requirements of the clients. There are various upsides and disadvantages to having the number one fashion designer or a first rate institution maintain up the product [Pfl98] [Pig97] [Par86]. That desire should be located forth on a protection by means of-case premise. 3

.Three.2.4.2 Outsourcing [DT97:c8s7;Pig97: c9s9.1,s9.2] Outsourcing of help is becoming a noteworthy employer. Large groups are re-appropriating whole responsibilities, together with programming assist. All the greater often redistributing is finished for fringe programming, as corporations are reluctant to discharge the product implemented in its middle company. One of the substantial difficulties is for the redistribute help organization to determine the extent of the exertion. Redistributing businesses through and massive spend numerous months surveying the product earlier than it'll renowned an settlement [DT97]. A few other test is the trade of the product to the redistributed business enterprise [Pig97]. 3.3.2.4.Three Organizational form [Pig97:c12s12.1-s12.3] based totally at the fact there are nearly the same variety of authoritative structures as there are programming protection institutions, a hierarchical structure for guide is first-rate created on a case-with the aid of the use of-case premise. What is important is the undertaking or mission of safety responsibility to a assembly [Pig97], paying little heed to the authoritative shape. Further as with exquisite endeavors, protection may be fruitful with entire management manual.

Three.Three.3 maintenance fee and maintenance price Estimation software engineers have to understand the numerous classifications of manual, these days pointed out, as a way to cope with the rate of upkeep. For arranging functions, comparing prices is a vital part of programming renovation.

4. Proposed protection ESTIMATING guidelines

To deal with the assessing deficiencies confirmed in table 2 (i.e., fee version Mappings to software program upkeep WBS), we've got built up an entire lot of version-self maintaining evaluating guidelines for use in assessing programming renovation and bolster charges. The ones rules, which may be abridged in desk three, were created for the four varieties of obligations we located that management programming life cycle bolster focuses typically finished, which consist of the accompanying:

Continuing aid – As tons as 80% of the work in government programming existence cycle bolster fixates is essentially performed on a stage-Of-strive (LOE) premise with the aid of manner of little companies of ten or a whole lot much less. These organizations are entrusted with doing all things needed to hold the framework operational in the situation. They set up their updates relying on criticality of the restoration and play out whatever undertakings are required together with machine maintenance and place backing to keep the framework working. They may consolidate improvements, restore problems and can make required pass to deal with execution issues. Inside the occasion that essential, they may even head out to restoration and fasten frameworks inside the subject. Though, most probably, their critical center is tending to programming inconvenience reviews, a large wide variety of that are taken from their gift accumulation.

Obtaining help – We found that as heaps as 30% of the paintings finished in government programming existence cycle bolster focuses become gone for giving procurement the board backing to application offices overseeing contractual employees that have been entrusted to do the preservation. Staff doled out to those companies gave oversight and path to the temporary humans. From time to time they carried out unusual examinations and one-of-a-kind events they done fluctuating stages of autonomous sorting out. This attempting out became in all instances long beyond for giving this system administrative center more confirmation that the framework being stored up could satisfy its conditions and execute now not enormously in its operational condition. Such attempting out went from independent checking out to unbiased Verification and Validation (IV&V). IV&V contrasts from self sustaining checking out in that it offers a gradually sizable entire help cycle survey of items as they're constructed (i.e., checks the plan meets necessities, guarantees the code fulfills the form, confirms the code is without imperfection and approves the release closer to stipulations set up for it). Furthermore, there is probably specific attempting directed for the motives for security or atomic affirmation additionally.

Medium-Scale enhancements – improvements are preferred protection ventures in which any other discharge is created thru an control programming lifestyles cycle bolster interest organization dependent on situations. The in-house programming life burn bolster employer keeps through a ordinary development cycle to refresh the product basically to fuse adjustments that cope with those conditions, repair appeared issues and make required execution enhancements (perfective modifications). However, these organizations whole drastically greater paintings than meets the eye. They furthermore help and maintain up the workplaces essential to create and take a look at the brand new discharges. The ones workplaces regularly consolidate greater gear than the ones

dedicated to advancement seeing that operational tool can and is regularly placed on top of it, at some thing factor ability. Medium-sized upgrades are the ones updates that may be cultivated in about a year through groups getting into length from ten to fifty human beings. The entice in assessing for such medium-sized upgrades is to ensure that the monetary plans handy are sufficient to play out the continuing designing and bolster assignments. Thinking about that spending plans for those activities are a first-rate deal of the time given as discrete facts, they're regularly insufficient to carry out the duty.

Huge Scale enhancements – massive-scaled enhancements variety from what we depicted for medium-sized ventures in-house lifestyles cycle bolster company length and term. Considering the fact that those multi-three hundred and sixty five days ventures employ greater than fifty individuals, they have got the economies of scale to reallocate staff to play out the whole-scope of assignments expected to hold up and manual the product (and the system). They have extra noteworthy independence than different types of duties. This permits them extensively extra possibility of interest. The trap in assessing for such large predicted improvements is to keep the adaptability to reallocate personnel beginning with one challenge then onto the subsequent, whilst the want emerges. As an example, they may need to get rid of workforce from working each other discharge and reassign them to persevering with designing undertakings whilst important new operating framework discharges had been being fused to their level.

Five. Modern-day traits AND future possibilities

At a few factor factor a mistake happens in any product, a specific diploma of time is expected to properly understand, disengage and expel the blame. The greater it takes to get better from the event of a blunder, the higher might be the costs related to programming maintenance. From as in advance of agenda as 1969 thus far, the arena of programming help has superior over some stretch of time. Each yr many research productions are introduced to the efficiently handy sizeable measure of facts. We've were given masterminded all the certified examinations sequentially and watched the sample. Next to major the appraisal of the outcomes acquired in each one of the shortlisted investigations, we have were given assessed them and outstanding imperatives present that allows you to recognize future bearings in this `eld. In view of the aftereffects of essential investigations, few growing sub-fields in the location of programming maintenance are featured as underneath:

Five.1. Enlargement of dynamic measurements along static measurements

In the beyond area, while finding the reaction for RQ2, we've got mentioned unique forms of measurements suites proposed, assessed and accredited observationally by way of using numerous specialists. As apparent from desk 6, C&ok metric suite and Li and Henry metric suite are the most widely diagnosed ones and used by numerous scientists in their individual investigations. We've were given visible that

tragically every the metric suites are surely static in nature. Considering the reality that certain strains of supply code possibly may not execute amid execution of the product counting on the records sources supplied and exceptional situations, as a consequence referring to those static plan measures to practicality won't be right. As Bieman and Otthas expected the ability attachment of their research, it might be of amazing enthusiasm to assess and examine the dynamic situations among one-of-a-type programming antiquities. One of the promising fields is to quantify those form measurements regularly in place of statically utilising Dynamic loss of brotherly love (DLCOM), Dynamic response For a class (DRFC), and so forth.

Five.2. Diploma with importance to outside brilliant homes

However the reality that C&ok metric suite may be very widely diagnosed among specialists, one constraint visible with the aid people is that it considers simply the internal structure measurements on the identical time as dismissing the importance of outdoor quality developments, for instance, nature with the code, flair dimension of software application engineer, development competencies, and so on which can be semantic in nature. As a matter of reality, indoors plan measurements suite is consistent with the information of ISO 9126 programming fine model, in this manner, concentrating the impact of out of doors satisfactory inclinations on practicality is each different promising situation which want to be researched.

5.3. Greater measurements to quantify current applications, for instance, transportable packages, giant databases, and so forth

In previous activities, the data it's positioned away on the backend may additionally have been gotten to a few instances each week, anyway with the boom inside the utilization of portable and mobile based applications, currently it's far gotten to sever a instances each hour. Because the product frameworks vigorously use databases, in the end we noticed that C &okay metric suite should not be enough as it'd now not capture the database taking care of components of the packages. Every different promising sub-area for studies is to analyze the important association of measurements underneath the trendy situations in which the programs are profoundly information excessive along their precise experimental approvals. One such observational research has been executed through the creators Malhotra and Chug [114] in advance than. In this research, damage inspite of regard for the database gets to changed into given and the brand new metric suite became experimentally proposed and showed to be unmatched. Studies of a exceptional deal of common measurements to gauge the framework's viability and abusing the conceivable combination of those measurements into a document for the framework's practicality are as yet a take a look at, but to be unraveled.

6. PROPOSED model

The created neural device relies upon at the again unfold calculation. BP calculation is honestly adaptable in its working limit. There are three layers in created demonstrate. The primary layer is inputs which have one neuron. The facts esteems are taken thru this layer. Second layer is shrouded layer which have 10 neurons to method the given records. Yield layer is the 1/three layer it is consists of one neuron to

talk to the consequences. The trade capacity, loads and dispositions of the sources of data are limited with the aid of way of the combination of included up and yield layers. The trade capacities are the arithmetical capacity which is probably direct and nonlinear. The affiliation among the neurons is balanced through the hundreds. On the factor whilst the records is displayed as the contribution to the neural tool at that point to build up the instance of statistics, masses and predispositions are balanced likewise. It is finished as such that the precise yield can be produced. Masses and inclinations are balanced by the BPN learning rule. After this adjustments, organize is ready to a best authorised measurement with the purpose that version can carry out to create desired effects.

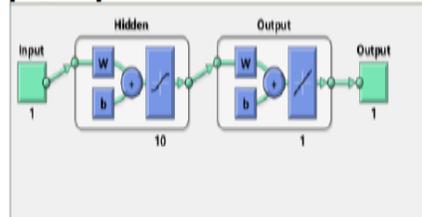


FIGURE 1: Basic structure of proposed NN

A. MODEL RESULTS

While improvement record is as enter and days taken to spare the overhauls are in light of the fact that the yield then the eventual outcomes of ANN are showed up in fig 2 and 3. The dashed line demonstrates the most ideal outcomes nearby yield = targets. There are such an extensive number of assortments inside the information units. Exactly when R cost is very nearly 1 then it systems the association among enter and target is gigantic. The R cost is 0.Sixty six which demonstrates the association between's the substances regards is one way or the other huge. As various components aren't considered in the facts, consequently this backslide cost is goliath in evaluation. The road addresses the relationship among the characteristics. The centers that are a long way from the dashed line are called irregularities. In endorsement plot the mentoring chart demonstrates that the sound isn't always right as a rule certainties esteems are exceptions.

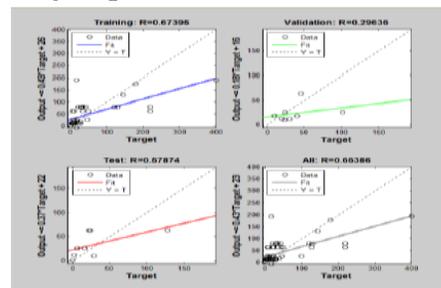


FIGURE 2a: Regression Plot of Enhancement record

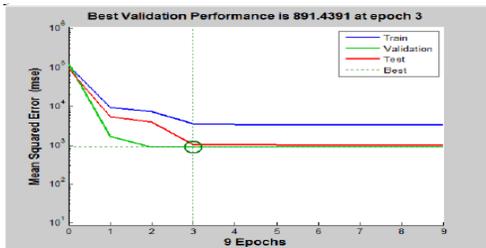


FIGURE 2b: Performance Plot of ER

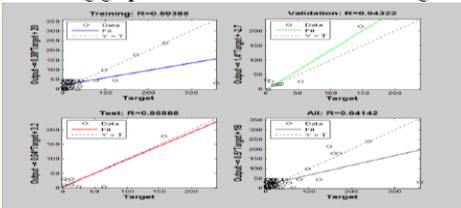


FIGURE 3a: Regression Plot of Correction Record

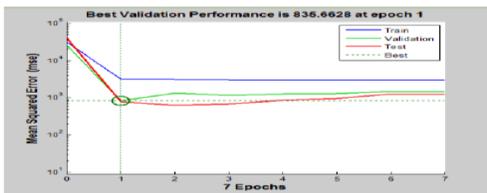


FIGURE 3b: Performance plot of Correction Record.

III. CONCLUSIONS

In this work, we played out a prohibitive report and exact review of the examinations dispersed in the field of programming practicality. New models and imaginative frameworks have been given the objective that item bolster gauge could be surveyed even more definitely. In the present paper, an effort has been made to review all of these models, factors, programming practices, etc and recognize distinctive basic perspectives which could uncommonly affect the help effort.

REFERENCES

1. R. Moser, W. Pedrycz, A. Sillitti, and G. Succi, "A model to identify refactoring effort during maintenance by mining source code repositories", 9th International Conference on Product Focused Software Process Improvement (PROFES 2008), Frascati (Rome), Italy, 23 - 25 June 2008.
2. E. Di Bella, I. Fronza, N. Phaphoom, A. Sillitti, G. Succi, and J. Vlasenko, "Pair Programming and Software Defects – a large industrial case study", Transaction on Software Engineering, IEEE, Vol. 39, No. 7, pp. 930 - 953, July 2013.
3. P. Abrahamsson, R. Moser, W. Pedrycz, A. Sillitti, and G. Succi, "Effort Prediction in Iterative Software Development Processes – Incremental Versus Global Prediction Models", 1st International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), Madrid, Spain, 20 - 21 September 2007.
4. R. Moser, P. Abrahamsson, W. Pedrycz, A. Sillitti, and G. Succi, "A case study on the impact of refactoring on quality and productivity in an agile team", 2nd IFIP Central and East European Conference on Software Engineering Techniques (CEE-SET 2007), Poznań, Poland, 10 - 12 October 2007.

5. R. Moser, A. Sillitti, P. Abrahamsson, and G. Succi, "Does refactoring improve reusability?", 9th International Conference on Software Reuse (ICSR-9), Turin, Italy, 11 - 15 June 2006.
6. P. Ciancarini, and A. Sillitti, "A Model for Predicting Bug Fixes in Open Source Operating Systems: an Empirical Study", 28th International Conference on Software Engineering and Knowledge Engineering (SEKE 2016), Redwood City, San Francisco Bay, CA, USA, 1 - 3 July 2016.
7. N.F. Schneidewind, "The State of Software Maintenance," IEEE Transactions on Software Engineering, vol. SE-13(3), March 1987, pp. 303-310 doi: 10.1109/TSE.1987.233161.
8. D.P. Hale and D.A. Haworth, "Software maintenance: a profile of past empirical research," Proceedings of the Conference on Software Maintenance, Oct. 1988, pp. 236-240, doi: 10.1109/ICSM.1988.10168.
9. D.P. Hale, D.A. Haworth and S. Sharpe, "Empirical software maintenance studies during the 1980s," Proceedings of the Conference on Software Maintenance, Nov. 1990, pp. 118-123, doi: 10.1109/ICSM.1990.131340.
10. S. Dekleva, "Delphi study of software maintenance problems," Proceedings of the Conference on Software Maintenance, Nov. 1992, pp. 10-17, doi: 10.1109/ICSM.1992.242564.