ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Special Issue (April)

Online - 2455-3891
Print - 0974-2441

Full Proceeding Paper

A CASE STUDY ON SOFTWARE PROJECT DEVELOPMENT COST, SCHEDULE, AND EFFORT ESTIMATION

PRABHAKAR RAO BVANSS[1,2]*, SEETHA RAMAIAH P[3]

[1]School of Computing Science and Engineering, VIT University, Chennai Campus, Chennai, Tamil Nadu, India. [2]Department of CSE, Research Scholar, JNTUK Kakinada, Andhra Pradesh, India. [3]Department of CS and SE, Andhra University, College of Engineering (A), Visakhapatnam, Andhra Pradesh, India. Email: prabhakarrao@vit.ac.in

Received: 23 January 2017, Revised and Accepted: 20 June 2017

ABSTRACT

This paper theme is to provide a case study of Software Project Development cost, effort, and schedule estimation. From recent past, a remarkable research takes place in developing different techniques on software effort and cost estimation. Making estimation before start of any project is necessary to be able to plan and manage any project. The estimate is an intelligent guess for the project resources. Nowadays, software has become a major contributor to economic growth for any nation. Making an estimate before starting any software project is vital for the project managers and key stakeholders. Major project milestones such as project schedules, budgeting, resource allocation, and project delivery dates are set on the effort and cost estimates. Thus, the reliability of the estimation leads any project success or otherwise fail. In this article, author's idea is to work with function point analysis and include the concept of workforce scheduling in a better way while taking the decision in the contract phase. That leads to strengthening the relations between the developer and the customer. Basically, size is a main measured unit of the software project. Based on the size and other functionalities, the software managers estimate the total effort required to develop the project. From the effort and work schedule, the total cost can be estimated.

Keywords: Workforce scheduling, Software, Project development, Effort, Cost, Estimation, Measurement, FPA, COCOMOII, General system characteristics, Value adjustment factor.

 DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.19538

INTRODUCTION

Nowadays, software industry is very strong and interconnected with other industries. Its role and involvement are high. Without software systems nothing is possible. For the last few decades onward, the embedding software into hardware components is relatively high [1-3]. All most all kinds of problems are being tackled by software industry. Once in a time, software development was very small and small in nature, but the cost of the software project was vast. However, things have changed like anything now the common man can reach to solve their problems [4,5]. In this context, and huge demand in industry and the healthy competition software industry has to quote an exact figure on the product.

NEED OF SOFTWARE ESTIMATION

Gauges estimate the future as well as regularly influence it. As well - low gauges can prompt to lower quality, conceivable revised in later stages, and higher dangers of venture disappointment; as well high gauges can diminish profitability as per Parkinson's Law, which expresses that work extends to fill the time accessible for its consummation.

This is the reason it is essential to think about whether as an exertion gauge is truly required. On the off chance that you do not generally require gauges or simply require them at a later stage, it may be more secure to abandon them or delay the estimation until more data are accessible [6-9]. Agile programming improvement - which includes arranging only the following sprint or discharge by utilizing criticism from past sprints or discharges - may be a decent approach to keep away from the potential mischief from assessing too soon.

REVIEW OF LITERATURE ON ESTIMATION MODELS

Different estimation models were designed by experts to meet estimation requirements. Different models also were developed in different periods because software engineering itself was evolving. Tables 1 and 2 provide a brief comparison enumerating the advantages and disadvantages of a few popular estimation methods [10-13].

Table 3 provides a select list of tools that are popular in the software estimation community. Each tool has been designed with a specific estimation method and a specific output as an objective [14-16].

CASE STUDY - ESTIMATION PROCESS

Step 1: Identify stakeholders
Steps 2: Identify category of information and data items
Steps 3: Identify category of data items
Steps 4: Identify data functions (Data Files: ILF)
Steps 5: Identify data functions (Data Files: EIF)
Steps 6: Identify transaction functions (EI/EQ/EO)
Step 7: Aggregate data and transaction function (unadjusted).
FP counts - Aggregate the data function FP count and the transaction function FP count to obtain the total unadjusted FP count.
Step 8: Obtain general system characteristic (GSC) values for invoice application
The next step is to obtain the degree of influence rating for each of the 14 general system characteristics (GSCs). Convert the total degree of influence value to value adjustment factor (VAF). Multiply the unadjusted FP count with VAF to obtain the adjusted FP count.
Step 9: Transform the adjusted FP count into the total effort.
Transform the adjusted FP count into the total effort required to execute the software project. This can be achieved by obtaining the delivery rate (productivity) of the project team and then multiplying the adjusted FP count with productivity. The effort thus obtained will encompass all the project execution lifecycle activities that include requirements, design, build (construction) and unit test, and system, and integration tests.

The following six steps take you through the sequence of converting the unadjusted function points count into adjusted function points and then to effort using assumed productivity. Then, finally, it provides you with an option for adding the project management overhead efforts.

1. The VAF = 1.12.
2. Final (adjusted) function points = 287×1.12=322.

Table 1: Pros and cons of various estimation models

Estimation method	Applicable stages	Advantages	Disadvantages
Function point	Requirements to testing	Assurance level is high UP view Language independent, methodologies, or tools Non-technical users have a better understanding	Depends on the subjective weight Needs trained person Manual operation. Measuring rules and business logic are more complex
COCOMOII	High-level requirements	Not only can use source lines of code but also can use object points, unadjusted function points as metrics for sizing a project Estimates of software costs	Local calibration needed for accuracy In early phase of system Size estimated with uncertainty
COSMIC-FFP	High-level requirements	Working for both MIS applications as well as real-time applications Simple to use	Benchmark data is not currently available International acceptance is limited but growing
Wideband Delphi Technique	Pre-requirements, proposal	Useful in the absence of qualified, empirical data	Expert's opinion Requires multiple experts Hard to document
SMC	End of design	Easy to understand and use program perspective	May not be consistent across users Only experts on technology can do

Table 2: Desired parameters in estimation tools

Feature	Description
Project type selection	Development, maintenance, enhancement, migration, porting, etc.
Calibration/standard	IFPUG, COCOMOII, NESMA, Parametric, Monte Carlo
Sizing method	SLOC, function points, COCOMOII, UML, use case, object points, etc.,
Lifecycle selection	Waterfall, iterative, etc.,
Programming language selection	20+popular languages
Maintaining historic data	Build repository
What-if scenarios	Analyze different scenarios
Track scope creep	Ability to track and flag changes
Constraints and priorities	Provision to identify constraints and priorities
Selection of relevant projects	Efficiency, staffing, reliability, phase customization
Storage of metrics and data analysis	Effort, schedule variance, defects, and other overheads
Integrated reporting	Flexible reporting features
Interface to other tools and applications	Interface Microsoft Office and web

Table 3: Popular estimation methods/tools

Popular tools	Available source through net
Construx estimate	http://www.construx.com/estimate
Costar	http://www.softstarsystems.com
CostXpert	http://www.costxpert.com
Function point WORKBENCH	http://www.charismatek.com.au
Knowledge plan	http://www.spr.com
PRICE-S	http://www.pricesystems.com
SEER-SEM	http://www.galorath.com
SLIM-estimate	http://www.qsm.com
COCOMO family	http://csse.usc.edu/tools/COCOMOSuite.php
Comparative estimating tool	www.isbsg.org
Early estimate checker	www.isbsg.org
FP outline	www.totalmetrics.com
PQM plus	www.qpmg.com
SMRe	www.qmg.com

3. Assume the technology is K and productivity for J2EE = 10 FP per person month.
4. Engineering effort required to develop the invoice application = 322/10 = 32 PM.
5. The additional effort for project management and configuration management = 15%.
6. Final effort = 32 + 5 (15%) = 37 person months.

Step 10: Transform the total effort into delivery schedule.

The total effort now needs to be adjusted to a project delivery schedule. There are methods are.

Step 11: Map the resource loading to meet the delivery schedule.

Transform the total effort obtained in Step 9 above into a resource loading chart. This step is significant in the sense that assigning the appropriate person with the right skills for the appropriate lifecycle phase of the project is critical for the project to be a success [17].

Step 12: Project managers Assignment.

- One project manager for every eight to ten technical staff members.
- One full-time project manager for every 1500 function points.
- One project manager for roughly every 1,50,000 source code statement.
- Project management starts before requirements and runs after the project ends.
- Project management work = 35% of the available management time.
- Personnel work = 30% of the available management time.
- Meetings with other managers or clients = 22% of the available management time.
- Department work = 8% of available management time.
- Miscellaneous work = 5% of available management time.

WORKFORCE SCHEDULING

Workforce scheduling includes putting the right individuals on the right employments at the right times to take care of client demand [18]. The objective of workforce scheduling is to coordinate the quantity of laborers accessible with the client request that exists in any given day and age.

Table 4: Project benefit analysis - make or buy decision

PInsF	Worth	Acceptance	Reusability	Solution	Teachability	Risk	Result
1	High	Positive	None	None	Frequent	Low	Good
2	Negative	*	*	*	*	*	Poor
3	Low	*	*	*	*	High	Poor
4	Moderate	Neutral	Adequate	Complete	Difficult	High	Poor
5	Low	Negative	None	Partial	Frequent	Low	Poor
6	High	Negative	Partial	None	Difficult	Moderate	Good
7	High	Positive	Partial	Complete	Frequent	High	Poor
8	High	Positive	Partial	Partial	Possible	Low	Poor
9	Low	Positive	Adequate	None	Frequent	Low	Good
10	High	Negative	Partial	None	Frequent	High	Good
11	Low	Positive	None	Complete	Difficult	Moderate	Poor
12	Low	Neutral	Adequate	Complete	Frequent	Low	Good
13	Low	Neutral	None	None	Difficult	Low	Good
14	Moderate	Positive	Adequate	None	Difficult	High	Poor
15	High	Negative	Adequate	Partial	Frequent	High	Poor
16	High	Negative	Partial	Complete	Possible	Low	Good
17	Moderate	Negative	None	Partial	Difficult	High	Good
18	Moderate	Neutral	Adequate	Partial	Difficult	Low	Poor

Four major steps are:

i. Forecast customer demand.
ii. Translate demand estimates of requirements.
iii. Schedule.
iv. Refine the schedule as per the need.

Algorithm 1: Workforce scheduling

Input: Customer demands and development industry requirements for a particular domain.
Output: Scheduling and timeline for each task of the development life cycle;

Repeat

i. Predict Customer interest for your administration by predicting potentials of the administration exchange that change after some time, for example, client landing rates.
ii. Calculate the quantity of representative hours required to fulfill the request anticipated in step one. This requires setting the quantity of workers with fitting expertise levels that are expected to serve clients enough amid the era being referred to.
iii. Develop the genuine work routine by considering representatives' aptitudes, yearnings, and solicitations, and afterward choosing who will do what work at what time.
iv. Change the work game plan as required by genuine request. This last stride guarantees compelling client benefit.

Until fine tune schedule

There are four cases can be experienced within planning:

- A planning skyline is a period for which calendars are produced at 1 time.
- Planning periods, which are likewise called arranging interims, are subsets of the booking skyline and are the nitty gritty interims utilized for staff arranging.
- Overstaffing, or surplus staffing, is a circumstance where more staff is booked in an arranging period than is in a perfect world required.
- Understaffing, or short staffing, is a circumstance where less staff is booked in an arranging period than is in a perfect world required.

Decision tree for benefit analysis(Table 4)

Decision trees helps to evaluate the given options. As per the sample data presented in Table 4 main project leading team can choose the projecting "Expected Outcomes". In this analysis, a decision tree structures a question/answer pattern. Apart from the data given, Table 5 demonstrations a broad question where each encounter has several significant possible outcomes and the closely related states = ('Willing', 'Avail'), Observations = ('Proc', 'Outs', 'Inso'), start_probability = {'Willing': 0.6, 'Avail': 0.4}, transition_probability = {'Willing': {'Willing': 0.7, 'Avail': 0.3}, 'Avail': {'Willing': 0.4, 'Avail': 0.6}}, emission_probability = {'Willing': {'Proc': 0.1, 'Outs': 0.4, 'Inso': 0.5}, 'Avail': {'Proc': 0.6, 'Outs': 0.3, 'Inso': 0.1}}. This message advocates the use of decision trees in combination with cost benefit analysis for decisions supporting enterprise technology projects. The point of this message is to encourage architects, project leads and engineers to bring structured decision tools to implementation decisions, rather than relying on mind-share, word-of-mouth, marketing, ease of access, or other rote decision aid. An example of the sort of thing you will end up with is shown in Fig. 1 with respect to the delivery status and Fig. 2 with respect to required type of specialists in large corporates. As an example (from the project management perspective) a root question might be: How do we handle a change in requirements? Graph 1 shows a possible workforce with actvitiy planning, supporting

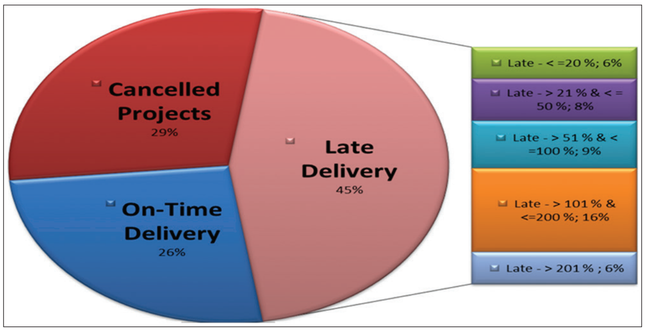


Fig. 1: Software project delivery status

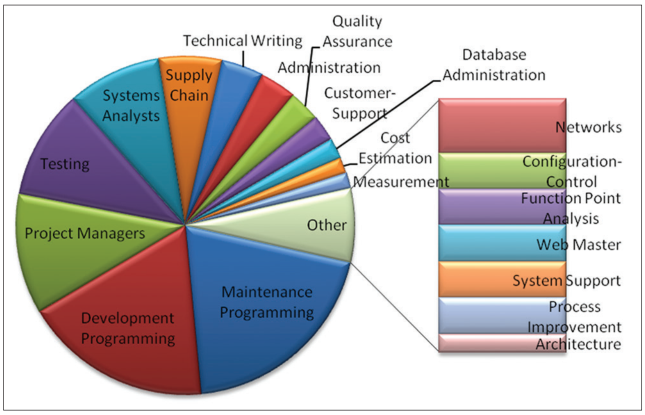
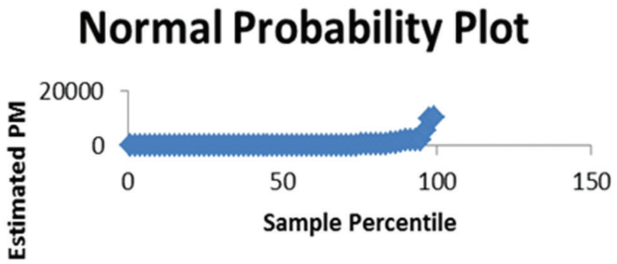


Fig. 2: Required type of specialists in large corporations

Table 5: Estimated person - months using regression model

SUMMARY OUTPUT



Regression statistics	
Multiple R	0.834987755
R^2	0.697204551
Adjusted R^2	0.670444703
Standard error	1055.418357
Observations	63

ANOVA

	df	SS	MS	F	Significance F
Regression	3	153890357.2	51296785.75	69.07678632	2.72566E-19
Residual	60	66834474.55	1113907.909		
Total	63	220724831.8			

	Coefficients	Standard error	t stat	p value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	−3213. 814452	2270.850627	−1.415246962	0.162165666	−7756.19198	1328.563071	-7756.191976	1328.563071
A	0	0	65535		0	0	0	0
SF	2853.96917	2016.652366	1.41520136	0.162178976	−1179.93613	6887.874469	-1179.936129	6887.874469
SIZE	9.048737351	0.812250955	11.14032221	3.06063E-16	7.423993549	10.67348115	7.423993549	10.67348115

Residual output | Probability output

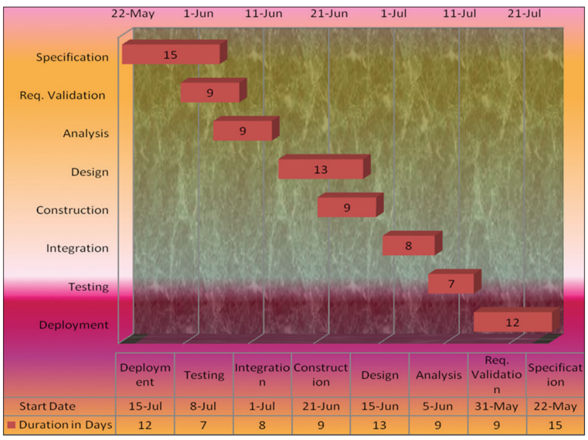
Observation	Predicted estimated PM	Residuals	Standard residuals	Percentile	Estimated PM
1	1.995315971	5.544524749	0.005383123	0.793650794	4.309697782
2	−161.0446523	168.7875518	0.163874138	2.380952381	6.283058065
3	0.547517995	5.73554007	0.005568578	3.968253968	6.906514749
4	−154.7105361	163.6907679	0.158925721	5.555555556	7.539840721
5	−190.0006118	198.9915695	0.193198915	7.142857143	7.742899481
6	1177.064348	−935.6440353	−0.908407391	8.73015873	8.980231733
7	−169.1885159	173.4982137	0.168447672	10.31746032	8.990957752
8	64.97452793	−24.61058797	−0.023894173	11.9047619	10.82916542
9	−72.36702623	107.8229938	0.104684261	13.49206349	17.99188977
10	301.4359251	−283.4440354	−0.275192965	15.07936508	18.11759714
11	−142.9471776	179.8004337	0.174566434	16.66666667	21.97078955
12	428.118248	−257.5350612	−0.250038203	18.25396825	23.50701941
13	99.55898343	0.775569755	0.000752993	19.84126984	26.63052188
14	−9.025864777	39.13548385	0.037996248	21.42857143	30.10961908
15	−9.025864777	85.08644388	0.082609573	23.01587302	35.45596759
16	−81.41576358	99.53336072	0.09663594	24.6031746	36.85325614
17	36.21782198	7.211267767	0.007001347	26.19047619	38.95280218
18	30.589326	−8.618536453	−0.00836765	27.77777778	40.36393996
:	:	:	:	:	:
:	:	:	:	:	:

this question. Depending on the project and organization change in deadline is not considered explicitly. Graph 2 shows that resource loading decision-making can be complex, and incorporate evidence of uncertainty. Graph 3 shows that work hours with respect countries decisions are made implicitly by other decision-makers on a daily basis. Decisions based largely on personal experience are subject to many biases. Graph 4 shows that decision analysis and cost-effectiveness analysis are systematic approaches used to support decision-making under conditions of uncertainty that involve important adjustments.

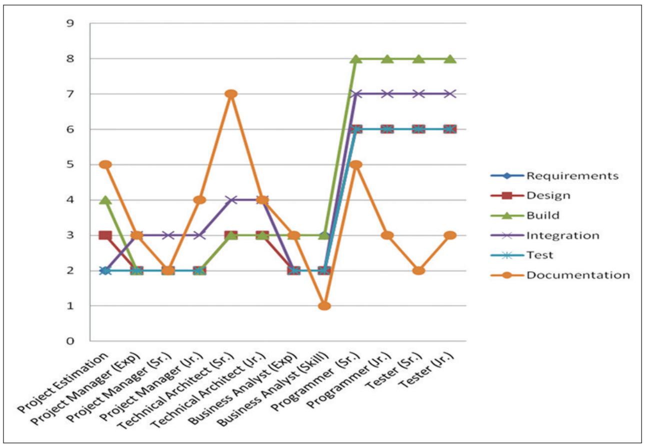
Regression model(Table 5)

Effort = −433.25*TeamExp + 408.8057*ManagerExp + 201.2701*Length + 4.2361*Transactions + 7.9056*Entities + 4.4594*Points Adjust + 92.1389*Envergure + −1777.4579*Langage + −278.0786.

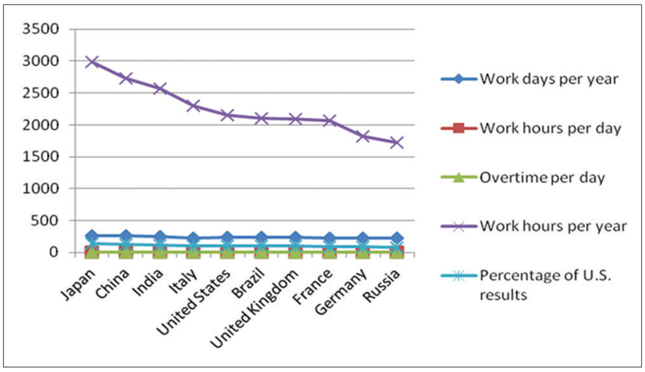
No two representatives have the very same abilities or longing to work a similar number of hours and in light of the fact that the administrator should likewise notice government controls, organization approaches,



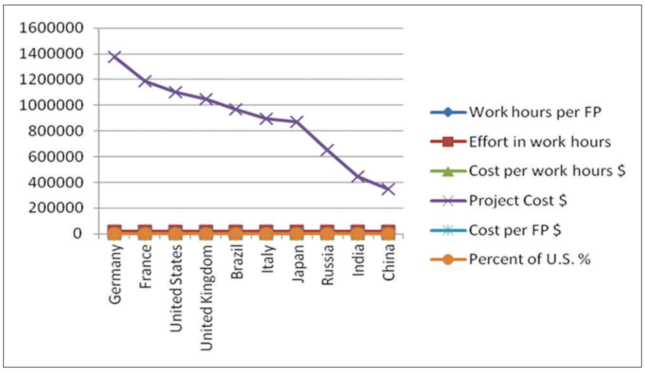
Graph 1: Workforce scheuling outcome for SDLC activity planning



Graph 2: Resource loading chart



Graph 3: Approximate number of work hours per year in ten countries



Graph 4: International cost comparison for 1000 function points

and authoritative commitments. At any given minute, having an excessively couple of workers - or enough representatives, yet not the individuals who have the fundamental aptitudes - can bring about poor client benefit; disappointed, exhausted representatives; and lost deals. Then again, having excessively numerous representatives either lessens working edges, if additional hours are booked or brings about low worker assurance if representatives work less hours than they want in light of the fact that the accessible work is spread daintily among numerous workers. All most all cases plan to build up a "discovery" scheduler for a conspicuous organization that still uses it every week for planning a huge number of representatives.

CONCLUSION

In this case study, effort, schedule, and cost estimation for software development project discussed. During the software estimation, we analyze and evaluate different existing techniques, and tools. A better comprehension of the unpredictability of the assignment, additionally that leaves away with a few thoughts regarding how to better deal with the booking procedure in any association. In light of what have seen, associations may have numerous chances to enhance their workforce-booking process. In this paper, the proposed estimation framework is well suited for application and embedded software development and maintenance projects. Nowadays, the open source software development impact is more on the development. Off-the-shelf products and shirk-wrapped projects are not considered for this purpose. In the future, we need to focus on that area.

REFERENCES

1. Dejaeger K, Verbeke W, Martens D, Baesens B. Data mining techniques for software effort estimation: A comparative study. IEEE Trans Softw Eng 2012;38(2):375-97.
2. Rao BV, Ramaiah PS. Software effort estimation framework to improve organizational productivity using emotion recognition of software engineers in spontaneous speech. J Soft Comput 2015;6(1):1076-82.
3. Cunha JC, Cruz S, Costa M, Rodrigues AR, Vieira M. Implementing Software Effort Estimation in a Medium-sized Company. 2011 34th IEEE Software Engineering Workshop,SEW.2011.19, IEEE Computer Society, 1550-6215/11. p. 92-6.
4. Rao BV, Seetharamaiah P. Organizational strategies and social interaction influence in software development effort estimation. J Comput Eng 2014;16(2):29-40.
5. Le-Do TK, Yoon KA, Seo YS, Bae DH. Filtering of Inconsistent Software Project Data for Analogy-based Effort Estimation. 2010 34th Annual IEEE Computer Software and Applications Conference, COMPSAC.2010.56, IEEE Computer Society, 0730-3157/10; 2010. p. 503-8.
6. Nadgeri SM, Hulsure VP, Gawande AD. Comparative Study of Various Regression Methods for Software Effort Estimation. Third International Conference on Emerging Trends in Engineering and Technology, ICETER.2010.22, IEEE Computer Society, 978-0-7695-4246-1/10 2010; 2010. p. 642-5.
7. Hsu CJ, Rodas NU, Huang CY, Peng KL. A Study of Improving the Accuracy of Software Effort Estimation Using Linearly Weighted Combinations. 2010 34th Annual IEEE Computer Software and Applications Conference Workshops, COMPSACW.2010.27, IEEE Computer Society, 978-0-7695-4105-1/10; 2010. p. 98-103.
8. Attarzadeh I, Ow SH. A Novel Soft Computing Model to Increase the Accuracy of Software Development Cost Estimation. IEEE, 978-1-4244-5586-7/10. Vol. 3; 2010. p. 603-7.
9. Rao BV, Ramaiah PS. Adaptive system for estimating development effort. J CNS 2011;1(1):52-6.
10. Keung J. Software Development Cost Estimation using Analogy: A Review. 2009 Australian Software Engineering Conference, ASWEC.2009.32, IEEE Computer Society, 1530-0803/09; 2009. p. 327-36.
11. Zhang B, Zhang R. Evaluation Model of Software Cost Estimation Methods Based on Fuzzy-Grey Theory, 2009 Fourth International Conference on Internet Computing for Science and Engineering, ICICSE.2009.63, IEEE Computer Society, 978-0-7695-4027-6/10; 2009. p. 52-5.
12. Rao BV, Ramaiah PS. A novel approach to design neuro-fuzzy expert system for software estimation. Int J Eng Res Technol 2013;2(13):3012-7.
13. Keung J, Jeffery R. Automated Support for Software Cost Estimation using Web-CoBRA, IEEE Computer Society, APSEC.2008.44, 1530-1362/08; 2008. p. 519-26.
14. Rao BV, Ramaiah PS. Software size estimation using fuzzy back-propagation network method. Int J Comput Sci 2012;9(1):339-48.
15. Idri A, Elyassami S. Applying fuzzy ID3 decision tree for software effort estimation. Int J Comput Sci 2011;8(4)131-8.
16. Nassif BA, Ho D, Capretz LF. Towards an early software estimation using log-linear regression and multilayer perceptron model. J Syst Softw 2013;86:144-60.
17. Parthasarathy MA. Practical Software Estimation - Function Point Methods for Insourced and Outsourced Projects. Saddle River, NJ: Infosys Press, Pearson, First Impression; 2007.
18. Thompson GM. Workforce Scheduling: A Guide for the Hospitality Industry by The Center for Hospitality Research at Cornell University. Cornell CHR Reports. Vol. 4. April; 2004.