

Towards Benchmarking User Stories Estimation with COSMIC Function Points-A Case Example of Participant Observation

Jayasri Angara¹, Srinivas Prasad², Gutta Sridevi³

^{1,2}Department of Computer Science, Koneru Lakshmaiah Education Foundation, India

³Department of Computer Science, Malla Reddy Institute of Technology, India

Article Info

Article history:

Received Oct 21, 2017

Revised Feb 22, 2018

Accepted Jul 19, 2018

Keyword:

COSMIC function points

Agile/DevOps estimation

Software estimation

Story points

User stories

ABSTRACT

Shorter time-to-market and unstable requirements is leading to introduction of Agile and DevOps practices. Story point estimation is becoming handier in Agile/DevOps setting. However, developing user stories and defining sizing unit in terms of story point is subjective process. It lacks benchmarking/standardization in terms of sizing measure and productivity of historical data. COSMIC has been considered as FSM (Functional Size Method). It has flexibility to receive requirements as User stories (popular Agile/Devops method) and derive COSMIC functional units using parametric approach. COSMIC method reduces the subjectivity and populates the productivity parameter for benchmarking. It standardizes the estimation process and can be easily deployed in Agile or DevOps setting. This paper presents the related work on linkages between User stories, COSMIC methods and traditional function point methods. It also presents the outcome of the industry survey conducted on 49 practitioners working in 10 different domains with respective to parametric estimation process adoption and presents 9 real-time case studies developed to demonstrate the usage of COSMIC method in various domains. This paper also attempts to derive mapping process of COSMIC functional process with User Stories with the help of 2 real-time industry case studies.

*Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Jayasri Angara,
Department of Computer Science,
Koneru Lakshmaiah Education Foundation,
Vijayawada, AP, India.
Email: angara.jayasri@gmail.com

1. INTRODUCTION

In the evolving role of software complexity and corresponding development methodologies, estimating accurate software size, cost, effort and schedule has been always thorny problem for the professionals. Software development cycles have been evolving from traditional waterfall model to contemporary Agile (parallel development and test) and DevOps (continuous integration, continuous testing and continuous delivery) practices. This phenomenon is resulting into introduction of new estimation standards and techniques [1]. Tradational estimation techniques are losing importance due to varius reasons like dependence on expert decisions and historical data, lack of early estimation methods, labour-intensive & complex collection process, calculation inaccuracies etc.[2] The purpose of estimation is to make significant decisions in the software delivery. Estimation has much bigger role in the context of continuous planning, continuous testing and delivery. However, the challenge in the Agile or DevOps execution is requirements are not known or not detailed before estimation. They are detected as User Stories, converted to Story Points and evolved during various iterations. Having said that story points are not standardized. Story point

estimation method is based on team's experience and intuitive driven. Two different teams may differ in arriving common story point number for the same user story.

Though Story point estimation is more user-friendly and acceptable in Agile / DevOps context, it is also important to devise an alternative method to predict project delivery metrics and bring perspective of parameterization in the estimation. Early sizing is precursor to risk analysis and accuracy [3]. In this context, this paper proposes the use of emerging COSMIC (Common Software Measurement International Consortium) Functional Size Method for estimation. It bridges Function Points method of traditional functional size method and Story Points of Agile/DevOps methodology [4]. It is measured against aggregation of various data movements within each functional process. Each data movement is considered as one COSMIC function point. If requirements are frozen and parameterized, COSMIC estimation method becomes prime estimation technique for a particular project. If requirements are volatile and parameterized, COSMIC method becomes supplement or complement to existing user story (story points) estimation method.

This paper structured as follows. Section II presents the related work on linkages to COSMIC estimation technique and Agile / DevOps setting. Section III presents parametric estimation adoption survey results in the industry, few industry case studies developed to demonstrate the flexibility of COSMIC function points method in various domains and user story mapping with COSMIC Function Units. Section IV presents the conclusion.

2. RELATED WORK

Any estimation technique used for software size determination should qualify five point criteria: Customer view of requirements, Functional Size as a prerequisite, Mathematical Validity, Verifiable and Bench Marking. If not all five parameters, they should be qualified maximum extent. There are various types of estimation techniques like judgment and rules of thumb based, analogy and work breakdown based, factors and weights based, size based, fuzzy and other models. There was a survey conducted on all these models against the five criteria parameters. It was found that Size based estimation qualifies completely on 5 point scale. Rest of the estimation models got rejected on more than two important criteria parameters. COSMIC Function Points technique overcomes the limitations posed by Function Point and satisfies the size based estimation criteria [5]. In addition, There exists various automation tools to convert story points or COSMIC function points to standard metrics. COSMIC fares second position in terms of usage preference in twenty software bench mark organizations. It is quite possible to extrapolate size from partial requirements using mathematical and statistical tools which is critical requirement of agile or DevOps [6].

Estimating large projects is always daunting task. For projects above 10,000 function points in size (equivalent to 1,250,000 C statements), both defect removal and documentation is more expensive than code itself. Hence, it is important to estimate the size [7]. Even though requirements are not clear in the beginning, estimation can be prepared with certain accuracy. At concept level, the estimation may be accurate ± 50 percent level, at function orient level, the estimate can go up to ± 25 percent accuracy level and at implementation orientation level, the estimation can be up to ± 10 percent accuracy level. These percentages may vary but fundemantely it makes sense to fix the deviation [8].

Agile method is preferred over rigid traditional water fall model because scope is not clear. Requirements are keep changing. Resources are new. In this scenario, agile recommends relative sizing or story points. Though agile recommends lean estimation process, it is important to complement results with parametric estimation technique like COSMIC which resembles user story and brings the rigor of traditional function points. Certain overheads are cut like meetings, documentation etc in agile which is positive aspect. However, this poses another challenge. Less documentation and quick estimation may have long term impact on the system [9]. In this context, creating linkages between agile and COSMIC is an important step in software development lifecycle. Generally in agile, requirement is defined as User Story. It is expressed in common usage language and understandable between all stakeholders. All user stories are estimated in the form of story points.

A typical User story looks like: As a ... [type of user] ...I want... [To perform some goal or task]. So that I can.. [Achieve some goal or value] [10]. After defining user stories, velocity has to be established (number of User Story Points delivered per iteration). Team calculates number of hours spent for each task. This way productivity gets established in agile. However, these USPs (user story points) cannot match with external benchmarking data. Hence, it is important to benchmark using ISO standard functional size measurement like COSMIC which is measured in CFP (Cosmic Function Points). It is technology independent [11]. To create mapping or linkage between user story and COSMIC method, all user stories need to be re-written to suit to COSMIC definition. In the case of COSMIC, it is read as: As a ... [functional user] ...I want to ... [respond to an event] ...[With some frequency and/or quality characteristic] ...So that

... [useful output, or outcome, produced. After defining functional processes, determine functional size in terms of CFP. Calculate effort per CFP. Alternatively, using a supervised text mining approach, COSMIC functional size can be automatically approximated from informal written textual requirements (in the form of User stories) which is demonstrating the COSMIC applicability in the agile process [12]. A study on 23 projects with 700 functional processes resulted into an estimated size of 475 CFP. Using regression equation it was found that

$$\text{CFP} = -16.5 + 6.698 \times \text{FPs} \quad [13]. \quad (1)$$

There was a study made for mapping of traditional function points of IFPUG to Full Function Points (FFP) and COSMIC method. There is a clear correlation established between IFPUG Function Points, FFP and COSMIC methods [14]. This process may be useful if requirements have to be mapped against function points, COSMIC function points and related story points. In another study the equation looks as follows

$$Y(\text{Cfu}) = 1.1 * (\text{UFP}) - 7.6, R^2 = 0.97, N = 4 \quad [15] \quad (2)$$

Another study arrives at

$$Y(\text{Cfu}) = 1.2 * (\text{UFP}) - 87, R^2 = 0.99, N = 11 \quad [16]. \quad (3)$$

Another study arrives at

$$Y(\text{Cfu}) = 0.84 * (\text{UFP}) + 18. \quad (4)$$

However, it was found that 30 to 40% of functional size comes from the data files. If they are excluded i.e.

$$(\text{UFP-TX}), \text{ equation can be arrived at } Y(\text{Cfu}) = 1.35 * (\text{UFP-TX}) + 5.5 \quad [17]. \quad (5)$$

The more accurate study made by Desharnais 2006 arrived at $Y(\text{Cfu}) = 1.36 * (\text{UFP-TX}) + 0, R^2 = 0.98,$

$$N = 14 \quad [18]. \quad (6)$$

Another study proposed $\text{CFP} = 1.22 (\text{NESMA FP}) - 64, R^2 = 0.97, N = 26 \quad [19]. \quad (7)$

These studies promote the usage of standard measurement methods like COSMIC Function Point with Statistical Process Control such as Six Sigma within the Agile Development framework. This kind of adoption from Six Sigma perspective provides a tool for early estimation and adaptation to measurable requirements change over sprints/iterations [20].

3. THE PARTICIPANT OBSERVATION WITH CASE STUDIES

In this section, it is explained the results of various case studies and observation details. The discussion is presented in several sub-chapters.

3.1. Practitioner's Survey on Adoption of Parametric Estimation

Though function points and COSMIC function points bring the rigor of standardization, they are easily fit in the context of agile. Functional user requirements of COSMIC are closely resemble to user stories, but adoption of these processes in the industry is low. We conducted industry practitioner's survey to understand the adoption levels and reasons for low usage of parametric estimation techniques when compared to conventional estimation techniques.

Total number of practitioners surveyed is 49 who are working in 10 different domains. As depicted in Figure 1, the survey states that 56% of participants are still comfortable with WBS, 19% are comfortable with KLOC method, 16% are comfortable with use case method and only 9% are comfortable with Function Points method. 94% of participants with less than 3 years experience are not aware of the estimation process and 6% of them having reasonable awareness. As depicted in Figure 2, 92% of sample with greater than 8 years of experience are using conventional estimation process and 8% of them using parametric.

This survey brought out the importance of orientation towards parametric estimation techniques like COSMIC in software development process especially in the context of Agile/DevOps. COSMIC brings perspective of logical and predictable correlation between various characteristics of project. User Story Points can be still primary method of estimation in Agile and DevOps, but complementing with COSMIC

FFP is going to be useful for sustainable project management. It brings benchmarking. The Function Point method did not meet the expectations because various issues like user's lack of expertise, non compliance of FP Technique for Real time projects, upper limit issues while calculating effort using Function Points, practitioners comfort zone with WBS (Work Breakdown Structure) etc. COSMIC estimation technique has potential to fill these gaps.

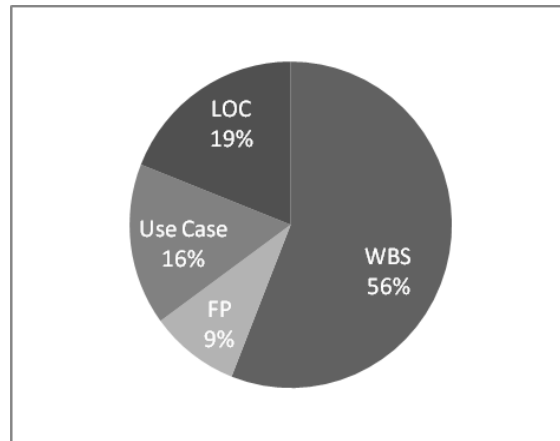


Figure 1. Practitioner's Preference in Estimation

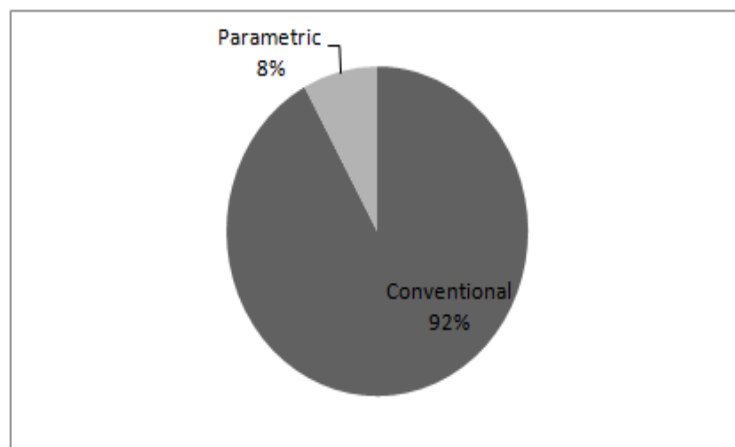


Figure 2. Usage of Conventional vs. Parametric Estimation techniques by sample with > 8 years experience

3.2. COSMIC FFP Case Studies: COSMIC FFP Estimated Effort vs. WBS (Actual) Effort

In order to understand COSMIC estimation process and deviations between COSMIC FFP estimation effort and Work Break Down based effort, we conducted a study on 9 case studies (real-time industry projects).The following Table 1 provides details of the projects (Project Domain, estimated efforts, deviation %) which are considered for our study.

The total COSMIC estimated effort for all above 9 projects is 764 man days. The total Work break down (WBS) for all 9 projects is 798 (actual effort incurred). The total deviation for all combined 9 projects combined is -4% man days which is in acceptable range. However, this sample size can to be increased for further benchmarking. In this case, the productivity (Hours/CFU) considered for various type of requests for OLAP requests is 2.5, Crystal Reports 1.5, ETL 3.5, Database 5, Platform 1, Enterprise requests 6.5. We conducted telephonic interviews to 5 COSMIC practisioners and collected these productivity numbers for various domains.

Table 1. Projects vs. Estimation Method vs. Size in CFU/FP and Effort

Sr.No	Project Domain	Estimation Method	Size	Effort (in days)	Deviation %
1	Telecom	COSMIC	50cfu	31	
		WBS (Actual)	-	29	6%
2	Embedded	COSMIC	390cfu	292	
		WBS (Actual)	-	300	-3%
3	Enterprise	COSMIC	124 cfu	72	
		FP	84 FP	69	-12%
4	Embedded	WBS (Actual)	-	77	-7%
		COSMIC	80cfu	64	
5	Datawarehousing (OLAP)	WBS (Actual)	-	67	-5%
		COSMIC	365 cfu	137	
6	Datawarehousing (Static Reporting)	WBS (Actual)	-	150	-9%
		COSMIC	165 cfu	31	
7	Data warehousing (ETL Mapping)	WBS (Actual)	-	28	10%
		COSMIC	232 cfu	101	
8	Datawarehousing (Database Requests Oracle based)	WBS (Actual)	-	110	-9%
		COSMIC	46 cfu	29	
9	Datawarehousing (Platform Maintenance Requests)	WBS (Actual)	-	31	-7%
		COSMIC	61 cfu	7	
		WBS (Actual)	-	6	14%

3.3. COSMIC FFP Case Studies: COSMIC FFP Units vs. User Stories Mapping

The most difficult step in agile process is benchmarking story points estimation and mapping of User stories to COSMIC Functional Process. We attempted to map User Stories to COSMIC Functional Process. For this, we have developed 2 realtime industry case studies for large market wide used software products.

3.3.1. Install Shield Wizard Case Study 1 (Embedded Domain)

The Install shield wizard enables end user to install various games, DVDs, Audio files from remote location. It is used to provide entertainment to the passengers. Stewards install this software in the pilot's cabin and make goodies such as games, DVDs and Audio files. These are available automatically to the passengers. The following Table 2 details Functional Sub Processes (COSMIC Functional Units) vs. User Story Mapping.

Table 2. Functional Processes vs. User Story Mapping

Functional Process	COSMIC Sub Processes	User Story
Developing 14 .dll files	55	As a power user, I can develop 14 dll files (can be divided into 14 User stories) (55 CFUs mapped);
Linking between dll files	4	As a power user, I can link between given dll files (4 CFUs mapped)
Creating as setup.exe	3	As a power user, I can create setup exe (3 CFUs mapped)
Installing Acrobat	5	As a power user, I want to install latest acrobat version (5 CFUs mapped)
Installing files in hard disk	3	As a power user, I want to install files in hard disk (3 CFUs mapped)
Installing Database Server	4	As a power user, I want to install database server (4 CFUs mapped)
Archiving un-install folder	2	As a power user, I want to un install folder (2 CFUs)
Updating Registry entries	2	As a power user, I want to update registry entries (2 CFUs)

Each of these main functional processes is mapped to a single declarative user story which again mapped on to average 2-6 CFUs. In this case, 21 User Stories are mapped to 81 CFU. Total effort was 514 Hours. Each User story mapped to 3.8 Cfu.

3.3.2. Repair Management System Case Study 2 (Enterprise Domain)

Business Partner sends an excel sheet to telecom vendor with few details like shipping address, quantity, code of equipment, serial number etc. Telecom vendor receives, update into system, checks warranty, communication on warranty update, final updated message back to business vendor. The following Table 3 details Functional Sub Processes (COSMIC Functional Units) vs. User Story Mapping.

Total Number of CFUs against 34 Functional Process or User Stories is 124. Total effort consumed with productivity of 6.5 hours= $124 * 6.5=806$ hrs. The following Table 4 attempts to cross verify both case studies. It is found that there is a relationship between 2 case studies in terms of number of COSMIC functional units per user story and effort per user story. However, this benchmarking process has to be vetted against larger sample so that it brings wide acceptance in terms of productivity for user stories as well as COSMIC functional units.

Table 3. Functional Processes vs. User Story Mapping

Functional Process	Sub Processes	Data Movement	User Story
Add Admin	4	4	As a super user, I want to add admin (4 CFUs mapped)
Modify Admin	6	6	As a super user, I want to modify admin (6 CFUs)
Delete Admin	4	4	As a super user, I want to delete admin (4 CFUs mapped)
Add User	4	4	As a Admin, I want to adduser (4 CFUs mapped)
Modify user	6	6	As a Admin, I want to modify user (6 CFUs mapped)
Delete user	4	4	As a Admin, I want to delete user (4CFUs mapped)
Add Region	4	4	As a Admin, I want to add region (4 CFUs mapped)
Modify Region	6	4	As a Admin, I want to modify region (6 CFUs mapped)
Delete Region	4	4	As a Admin, I want to delete region (4 CFUs mapped)
Add Country	4	4	As a Admin, I want to add country (4 CFUs mapped)
Modify Country	4	4	As a Admin, I want to modify country (6 CFUs mapped)
Delete Country	4	4	As a Admin, I want to delete country (4 CFUs mapped)
Add BP	4	4	As a Admin, I want to add BP (4 CFUs mapped)
Modify BP	6	6	As a Admin, I want to modify BP (6 CFUs mapped)
Delete BP	4	6	As a Admin, I want to delete BP (4 CFUs mapped)
Add COMCODER	4	4	As a Admin, I want to add COMCODER (4 CFUs)
Modify COMCODER	4	6	As a Admin, I want to modify COMCODER (6 CFUs)
Delete COMCODER	4	4	As a Admin, I want to delete COMCODER (4 CFUs)
Super User Logs in	4	4	As a Super User, I want to login (4 CFUs mapped)
Admin User Logs in	4	4	As a Admin User, I want to login (4 CFUs mapped)
Add RMA	6	5	As a BP, I want to add RMA (5 CFUs mapped)
Clarify RMA	5	5	As a BP, I want to Clarify RMA (5 CFUs mapped)
Modify RMA	6	6	As a BP, I want to Modify RMA (5 CFUs mapped)
Delete RMA	4	4	As a BP, I want to Delete RMA (4 CFUs mapped)
View Requests submit	3	3	As a BP, I want to View Submitted Requests (3 CFUs)
View Requests Clarified	4	4	As a BP, I want to View Clarified Requests (4 CFUs)
View Requests approve	4	4	As a BP, I want to View Approved Requests (4 CFUs)
View Requests Rejected	4	4	As a BP, I want to View Rejected Requests (4 CFUs)
BP User Logs In	4	4	As a BP, I Log in (4 CFUs mapped)
BP-Date from to display	4	4	As a BP, I want to view BP-Date (4 CFUs mapped)
Approve RMA	5	5	As a Client, I want to approve RMA (5 CFUs mapped)
Reject RMA	5	5	As a Client, I want to reject RMA (5 CFUs mapped)
Client Logs In	4	4	As a Client, I want to log in (4 CFUs mapped)

Table 4. Functional Processes vs. User Stories Vs CFU vs Per User Story Effort

Case Study #	No of Functional Processes	No of User Stories	Total CFU	Total Effort (hrs)	Per User Story CFU	Per User Story effort (hr)
Case Study 1	21	21	81	514 hrs	3.8 cfu	24.4 hrs
Case Study 2	34	34	124	806 hrs	3.6 cfu	23.7 hrs

4. CONCLUSION

One of the significant weaknesses of the traditional function points estimation method is its size limit. In contrast, there is no upper limit restriction to the size COSMIC functional process. COSMIC method

is also applicable for enhancement requests by adding up or reducing the number of sub-processes which are changed or deleted. COSMIC-FFP method can be easily applied to real-time, infrastructure software, business software and to different hybrids types. It is easy to learn and apply. However, one of the biggest challenges of software development is collecting requirements and expecting these requirements unchanged. User stories in conjunction to agile method proved most acceptable, flexible technique for collecting requirements and estimation. However, it lacks benchmarking. This paper contributes towards establishing linkages between User stories, COSMIC method and traditional function point method and developed mapping process between them with the help of real-time industry case studies. This paper provides a direction to benchmark estimation process in COSMIC FFP setting while leveraging the flexibility of Agile/DevOps.

REFERENCES

- [1] Prakash B, Viswanathan V A Survey on Software Estimation Techniques in Traditional and Agile Development Model, *Indonesian Journal of Electrical Engineering and Computer Science*, 2017; 7(3): 867 - 876.
- [2] Sahoo P, Mohanty J.R, Early Test Effort Prediction using UML Diagrams, *Indonesian Journal of Electrical Engineering and Computer Science*, 2017; 5(1): 220 - 228
- [3] Jones C. *Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies McGraw-Hill*, New York, 2010.
- [4] Fehlmann T, Santillo L. *From Story Points to COSMIC Function Points in Agile Software Development – A Six Sigma perspective*. in Metrikon - Software MetrikKongress, Stuttgart. 2010.
- [5] Jayakumar K. R, Abran A. A Survey of Software Test Estimation Techniques. *Journal of Software Engineering and Applications*, 2013; 6: 47-52.
- [6] Jones C. *Estimating Software Costs: Brining Realism to Estimating*. 2nd Edition, *Tata McGraw-Hill*, New York, 2007;229-302.
- [7] Jones C. Software Cost Estimating Methods for Large Projects. *The Journal of Defense Software Engineering*, 2005;18(4); 8-12.
- [8] Roetzheim W. Estimating and Managing Project Scope for New Development. *The Journal of Defense Software Engineering*, 2005;18(4); 4-7.
- [9] Rice R. W. Dealing With the Time Crunch in Software Testing. *The Journal of Defense Software Engineering*, 2015;28(2);14-18.
- [10] Schwaber K, Beedle M. *Agile Software Development with Scrum*, *Upper Saddle River, Prentice Hall*, 2002; 230-240.
- [11] Berardi E. et al. Guideline for the use of COSMIC FSM to manage Agile projects: The COSMIC Functional Size Measurement Method, Version 3.0.1 Retrieved July 1, 2016; http://www.cosmicon.com/portal/public/COSMIC_Agile_Projects_Guideline_v10.pdf
- [12] Hussain I, Kosseim L, Ormandjieva, O. Towards Approximating COSMIC Functional Size from User Requirements in Agile Development Processes Using Text Mining. *Natural Language Processing and Information Systems*, The series Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010; 6177; 80-91.
- [13] delBianco V. et.al. Model-based early and rapid estimation of COSMIC functional size – An experimental evaluation. *Information and Software Technology*, 2014;56; 1253–1267.
- [14] VinhT.ho, Abran A, Thomas, F. *A Comparative Study Case of COSMIC-FFP, Full Function Point and IFPUG Methods*. in Proceeding ITON-10;2010;7-11.
- [15] Fetcke T, *The warehouse software portfolio, a case study in functional size measurement*, Technical report no. 1999-20, Département d'informatique, Université du Québec à Montréal, Canada, 1999
- [16] Vogelegang F, Lesterhuis A., *Applicability of COSMIC Full Function Points in an administrative environment: Experiences of an early dopter*, 13th International Workshop on Software Measurement – IWSM2003, Shaker Verlag, Montréal, 2003.
- [17] Abran A, Desharnais J.-M, Azziz F, *Measurement Convertibility: From Function Points to COSMIC-FFP*. 15th International Workshop on Software Measurement – IWSM 2005, Montréal (Canada), Shaker Verlag, Sept. 12-14, 2005; 227-240.
- [18] Desharnais J, Abran, A, Cuadrado J. *Convertibility of Function Points to COSMIC-FFP: Identification and Analysis of Functional Outliers*. Int. Conference on Software Process and Product Measurement -- MENSURA 2006, Madrid.
- [19] Sageti H.H. Software Measurement European Forum 2007, Sogeti, Rome, Italy, May 9-11, Retrieved on July 1 2016 from <http://www.dpo.it/smef2007/presentazioni/day2/203.pdf>
- [20] Cohn M. *Succeeding with Agile.: Software Development Using Scrum*, Pearson Education, Boston, MA, 2009.

BIOGRAPHIES OF AUTHORS

Jayasri Angara is working as Associate Professor in MVGR College of Engineering (Autonomous) and Research Scholar at Koneru Lakshmaiah Education Foundation, Vijayawada, AP, India. Completed BTech (CSE) from Andhra University (GITAM) in the year 2000 and M.S. in Software Engineering from BITS Pilani. Prior to this, Jayasri worked for large software company as Project Manager for 9 years. Overall, Jayasri is having total 17 years of experience in Industry, Academics and Research.



Dr.Srinivas Prasad is working as Professor, Department of Computer Science in Koneru Lakshmaiah Education Foundation, Vijayawada, AP. Dr Srinivas has done his M.Tech from ISM.Dhanbad, M.S.from BITS, Pilani, Ph.D from Utkal University. Prior to this stint, Dr Prasad worked as Professor/Head of the Department in various esteemed engineering colleges and software consultant in USA. Dr Prasad is Fellow Member, Institution of Engineers, India (IE, India) and Fellow Member, Institution of Electronics and Telecommunication Engineers.



Dr. Gutta Sridevi is working as Professor & Head in the Department of Computer Science and Engineering in Malla Reddy Institute of Technology, Hyderabad, Telangana. Dr.Sridevi has done her Ph.D. in Computer Science & Engineering from Acharya Nagarjuna University, A.P., India. Dr Sridevi is having 14 years of teaching and research experience in the areas of software engineering and data mining. Dr Sridevi has published more than 40 publications in various refereed journals in the areas of software reliability, DevOps, Cloud platforms etc.