Title	Towards holistic goal centered performance management in software development: lessons from a best practice analysis
Author(s)	Murphy, Thomas; Cormican, Kathryn
Publication Date	2015
Publication Information	Murphy, T,Cormican, K (2015) 'Towards holistic goal centered performance management in software development: lessons from a best practice analysis'. International Journal Of Information Systems And Project Management, 3 :23-36. DOI: 10.12821/ijispm030402
Publisher	SciKA
Link to publisher's version	http:/dx.doi.org/10.12821/ijispm030402
Item record	http://hdl.handle.net/10379/6174
DOI	http://dx.doi.org/10.12821/ijispm030402

review version

International Journal of
Information Systems and Project Management
ISSN (print):2182-7796, ISSN (online):2182-7788, ISSN (cd-rom):2182-780X
For peer review only

Towards holistic goal centered performance management in software development: Lessons from a best practice analysis

Thomas Murphy
NDEC
Dublin City University
Ireland
Thomas.Murphy5@dcu.ie

Kathryn Cormican
College of Engineering & Informatics
National University of Ireland, Galway
Ireland
Kathryn.Cormican@nuigalway.ie

Abstract:
There are strong motivating factors for more effective performance measurement practices in software development. Astute practices in this domain are lauded to improve efficiency and effectiveness. However previous studies have shown that measurement in software is intricate, complex and fraught with challenges. Consequently it is poorly managed in practice. Our research seeks to better understand performance management in a real wold software development setting in order to identify the challenges and generate a roadmap for improvement. This paper presents findings from an inductive analysis of a radical measurement program in a global software organization. Our study investigates the level at which non-compliance with best practice can explain the company's disappointing results. We found that a narrow focus on projects, rather than on organizational goals has seriously hindered its success. We also found that the rate of change in the organization as a whole was impinging on the effective implementation of its measurement program. An analysis of the results demonstrates just how challenging software measurement is. The findings provide an evaluation of best practice relative to the literature that is informed by real industry experience.

Keywords:
software measurement, metrics, best practice, project management, challenges, non-compliance, exploratory research

1. Introduction

Researchers recognize that performance measurement programs are central to the improvement of the software development process and hence an organization's overall productivity levels [1]-[2]. However, these programs are extremely difficult to implement and success rates are low [3]-[5]. The challenges are wide and varied but seem to arise from two main sources: software and people. Software by its very nature is difficult to quantify and measure; and people, again by their nature, are resistant to having their work measured. Therefore it seems that an organization that wishes to improve productivity through measurement must ensure that the correct metrics are implemented and that the system makes sense to its people. Notwithstanding the inherent difficulties in measuring software productivity, there is a wealth of research in the field offering guidance as to best practice. This research investigates if there are underlying reasons as to why, despite the availability of guidance, software measurement programs still encounter difficulties and yield disappointing results. We present a case study of a global software company (referred to as GSC) that has been through a radical measurement program and has undergone significant organizational change in the process. The organization in question has not had the desired level of productivity improvement from the program. This research focuses on the Irish business unit. It was conducted through a series of detailed structured interviews with 21 people from three different levels involved in the measurement program and aims to determine the extent to which non-compliance with best practice can explain GSC's poor results from their measurement programs.

The research found that there was a disparity between how senior managers, middle managers and developers viewed the effectiveness of the measurement process. It extrapolates the reasons for the perceived failure of the measurement program across the roles of senior management, middle management and developers, and synthesizes and collates their experience. The lessons learned in GSC were found to be primarily in terms of attitudes towards measurement and the difficulties inherent with dealing with high levels of change. Furthermore, we identified a pervading culture of focusing on the delivery of individual projects, rather than learning from other projects and sharing information. It also seems that in the rush to deliver projects on-time, quality and productivity issues were being neglected. It is hoped that the findings of this paper will offer a more complete overview of how a large software organization can avoid some of the pitfalls inherent in software measurement programs.

The next section synthesizes the literature and identifies, categorizes and discusses critical success factors found to be effective in software measurement. Section 3 presents the research methodology employed in this study. Section 4 analyses the results from our study and finally section 5 discusses these results and provides some recommendations to academics and practitioners based on our findings.

2. Challenges with software measurement

Software measurement programs are lauded to enable effective control, reliable communication and improved productivity [6]-[7]. However, software measurement is an emerging field and remains a major challenge for many organizations [8]-[9]. Published research has highlighted many problems in this regard [7], [10]-[11]. For example, we learn that despite significant investment the majority of software measurement programs fail in practice. In other words, they do not last more than two years, and they do not have an impact on the organization's management decisions. Many researchers refer to the complex nature of productivity measurement which inhibits attempts to improve the productivity of software development processes in software engineering organizations [6], [10]-[11]. Researchers such as Munson [10] emphasize the enormity of the challenge experienced by software engineering organizations in implementing metrics programs. It seems that the software industry is making slower progress than the hardware industry in measuring, analyzing and implementing improvements in productivity. Kasunic [12] outlines problems in this regard that clearly needs to be addressed. These include:

- There is still a significant gap between the current and the desired state of measurement practice due to a lack of effective implementation and follow-through when it comes to measurement practices.

- There are different perspectives between management (executives, program managers or project managers) and practitioners (engineers, analysts and programmers) indicating a lack of "congruent communication" between the two groups.
- Measurement information is not always used effectively because it is not acted upon.

It seems that a holistic, goal-orientated approach to measurement, combined with effective management practices, is required to address the challenges faced in these areas directly.

3. Holistic goal centered measurement

Izhar, et al [13] stress that measurement must start with the articulation of high-level organizational goals, putting it simply: '*a bottom-up approach will not work*'. In order to achieve a purposeful measurement system specific performance metrics must be developed that operationalize and align with the organizations goals [14]-[15]. Indeed if metrics are not tied to goals and context, there will be no way of interpreting the data in a meaningful way. Many models have been proposed to provide such alignment [7], [16]-[17]. Attention must also be paid to wider organizational factors that impact measurement of software productivity, and many researchers stress the importance of measuring all project and process activities. A growing number of researchers argue for a holistic view that considers both technical and non-technical aspects measures as well as their interconnectedness [5], [18]-[22]. A synthesis and analysis of the literature identified critical success factors (CSFs) known to be effective in software productivity measurement. These are (a) people, (b) technology, (c) organization, (d) information structure and (e) management practice. These critical success factors, when considered together, ensure a balanced and holistic approach for effective software measurement.

3.1 Organization

A goal-centered approach that encompasses business objectives is critical to the success of a measurement initiative [12]-[13], [20]. Metrics should be tied to business and improvement goals, and all data should have a specific purpose. Otherwise the program runs the risk of collecting redundant data. Furthermore, a measurement program needs dedicated resources; if measurement is done on a voluntary basis without key personnel it will become directionless. Therefore a metrics program should have dedicated personnel involved with the necessary skill sets for driving the program and making the data collected meaningful.

3.2 Management practices

Senior management practices and attitudes towards measurement programs will influence their success: if management insists on a radical approach prioritizing their own department's metrics, there will be mixed results. There also needs to be cross-departmental agreement on the measurements use so that data is clear and meaningful to all. Research suggests that an incremental approach should be adopted. Defining and collecting metrics incrementally over time allows for timely feedback and fine-tuning of the program. Incremental implementation is also less risky than a single push. [23]-[25]. In addition measurement programs should be standardized in order to reduce effort and ensure clarity [23]-[26].

3.3 People

Non-technical, 'human' issues are the most critical to the success of an organization's metrics program. Metrics need to be transparent and there needs to be clarity across the key stakeholders as to the rationale for measurement. It is the work of practitioners that will be mostly measured so that there needs to be buy-in from the outset. If the activities being measured are viewed as being to blame rather than improve then the data is open to manipulation. Software developers should be involved from the outset of the design of a metrics program in order to ensure buy-in and active participation during the implementation phase [23], [25], [27]. In addition, internal or external champions for the measurement program can increase awareness, enthusiasm and understanding [23], [25]-[26]. As with any process

improvement program, staff should be trained at the appropriate level, whether in terms of raising awareness of the rationale for the program, or specific training in data collection or analysis.

3.4 Information and communication

Clear, positive communication regarding the rationale behind a metrics program is needed to secure tacit support from practitioners and management; otherwise the program runs the risk of being viewed as being for blame purposes rather than for productivity improvement. Best practice should ensure the following three factors. Consequently transparency is essential. In other words, there needs to be clarity on the nature of the data being collected and also the on purpose to which it is being put [23], [25], [28]. There also need to be clarity on the usefulness of the data so that participants understand the rationale for collecting it [23]. Feedback mechanisms increase visibility of a metrics program and reassure participants that the data is being put to use [25], [27].

3.5 Technology

The technology used in a metrics program should maximize data collection efficiency and accuracy – ideally data collection should be automated and a platform that can be understood and utilized across the organization for both reference purposes, comparison and historical analysis. Automated collection of data ensures more efficient use of resources and accuracy of data [23], [25]-[26]. A data repository is also needed to store data for comparative analysis and to evaluate overall trends, allowing a cyclical process whereby metrics are controlled and evolve according to business needs [23]. Finally, the metrics used should be based on robust data that is not open to manipulation [25].

4. Research methodology

4.1 Research method

The goal of our study was to uncover deep insights into a particular context specific phenomenon. More specifically we wanted to examine the attitudes of staff to a software measurement program and to determine why significant improvements are rarely made. Therefore an inductive case study analysis was used as it is best suited to this type of research. The strength of case study research lies in the ability to undertake an investigation into a phenomenon in its context. This case study is used to explain a situation and to provide a basis to apply solutions to situations. The advantages of the case study method are its applicability to real-life, contemporary, social- technical systems [29]. Case study results relate directly to everyday experience and facilitate an understanding of complex real-life situations.

4.2 Case profile

GSC is a leading global financial services organization providing enterprise software development services to the financial sector. There are approximately 35 business units. The Irish operation was established almost 20 years ago and it focuses on software development. Significant organizational transformation has occurred within the software engineering division to improve efficiency and effectiveness. The total technology spend has been significantly more than its competitors for many years. However, management has had difficulty in demonstrating the return on investment of such a significant investment. Consequently, the measurement program was reviewed and amendments to metrics and processes ensued. The major challenge has been to transform the metrics from simply measuring adherence to activities, to metrics that clearly demonstrate the value-add or improvements in those activities. A monthly and quarterly analysis of all key software engineering activities is presented to the top management of the organization and trends and trajectories are examined in a bid to spur key organization improvement changes. Despite these efforts GSC found that they were not making significant productivity improvements and that the overall results were disappointing.

Table 1. Critical success factors used in data collection protocol

Category	CSF	Description
Organization	Goal-oriented approach	Metrics should be tied to business and improvement goals, and all data should have a specific purpose
	Dedicated metrics team	A metrics program should have dedicated personnel involved with the necessary skill sets for driving the program and making the data collected meaningful
Management Practices	Incremental approach	Defining and collecting metrics incrementally over time allows for timely feedback and fine-tuning of the program. Incremental implementation is also less risky than a single push
	Standardised procedures	Standardised processes and procedures in measurement activities within an organization reduce effort and ensure clarity
People	Developer participation	Software developers should be involved from the outset of the design of a metrics program, in order to ensure buy-in during the implementation phase
	Practitioner training	Staff should be trained at the appropriate level, whether in terms of raising awareness of the rationale for the program, or specific training in data collection or analysis
	Program champions	Internal or external champions for the measurement program can increase awareness, enthusiasm and understanding
Information and Communication	Transparency	There needs to be clarity on the nature of the data being collected and also the on purpose to which it is being put
	Usefulness	There also need to be clarity on the usefulness of the data so that participants understand the rationale for collecting it
	Feedback	Feedback mechanisms increase visibility of a metrics program and reassure participants that the data is being put to use
Technology	Automated data collection	Automated collection of data ensures more efficient use of resources and accuracy of data
	Metrics database	A data repository is needed to store data for comparative analysis and to evaluate overall trends, allowing a cyclical process whereby metrics are controlled and evolve according to business needs
	Metrics integrity	The metrics used should be based on robust data that is not open to manipulation

4.3 Sample

Software measurement programme spans many functions, disciplines and hierarchies. Therefore it is essential to set boundaries in order to maintain a clear scope. Consequently, the unit of analysis in this study centered on; (a) senior managers, (b) middle managers and (c) developers. To do this a purposive, stratified non-probability sample was employed. Tansey [30] indicates that a non-probability sample is effective when, as in this study, the research is exploring what is occurring. Patton [31] suggests that a purposive sample '*has a logic and power - and provides rich information'*. Our sample selection was dictated by replication logic instead of a statistical one in accordance with best practice in case study research [32]. Consequently the researcher's judgment was used to select the sample elements in

order to ensure that rich and meaningful data was collected. The sample was stratified to include opinions from three key areas represented in the unit of analysis. The number of subjects chosen from each of the strata was not based on the size of the stratum relative to the target population size. Instead it was specifically chosen to equally represent senior managers, middle managers and developers. Therefore detailed structured interviews were conducted with 21 people from three different levels involved in the measurement program in Ireland. 7 representatives of senior management; 7 representatives of middle management and 7 software developers participated in our study. All of the participants are involved in measurement activities and productivity improvement. Every attempt was made to ensure internal validity (i.e. that any causal conclusion is warranted and free from bias).

4.4 Constructs and measures

In order to analyse the challenges associated with software measurement programs a number of best practice models and frameworks were studied. Five constructs were identified which significantly impact on software measurement programs. These include (a) organisation; (b) management practices, (c) people, (d) information & communication and (e) technology. Upon further analysis we identified 13 factors from the literature that aligned to the categories. Specific variables and measures were developed to operationalize these constructs. Multiple indicators (i.e. independent variables) for each best practice construct (i.e. dependant variable) were developed to increase the validity of the data collection instrument. Table 1 summarises these five categories and 13 success factors.

4.5 Data collection

The method of generalisation for case studies in general and our study in particular is analytical generalisation (as opposed to statistical generalisation) in which a previously developed theory is used as a template to compare the empirical results of the case study [33]. This however is a complex endeavour and so great care and attention was paid to maintaining scientific rigour in our data collection process. Therefore best practice techniques were used to ensure reliability and validity [32]. In order to ensure construct validity a chain of evidence was established and a sample of key informants (n= 6) reviewed the draft case study report. Internal validity was strengthened by the use of respondent maps and pattern matching. In addition a comprehensive research protocol helped to improve external validity.

5. Analysis of results

The results showed that across the board, staff did not view the implementation of the metrics program as effective. While they agreed that the success factors were important, they did not view their company's implementation of them as effective. The views of senior management on how successful their implementation was, is seriously at odds with how middle management and practitioners view them. Middle managers have the highest opinion of the importance of the success factors, yet have the lowest opinion on how the company is implementing them. The results clearly demonstrate that while the organization understood that best practice was important, they were unable to implement it. The underlying reasons for this non-compliance are teased out in the following analysis of the interviews carried out with participants.

Table 2. Attitudes towards critical success factors and implementation

	Senior Management	Middle Management	Software Developer
CSF perceived importance %	86	100	100
CSF implementation effectiveness %	43	7	14
CSF implementation ineffectiveness %	28	71	57

5.1 Organization

Organization structure and culture accounted for the highest percentage of reasons offered for non-observance of best practice. Strong efforts have been made to move focus away from projects and on to products by creating the product-portfolio organizational structure and relevant changes in information systems and measurement reports, so that application data is incorporated. However, there has been a significant lag in getting working practice behaviors in tune with the new structure. Local issues still resonate more with staff than do enterprise-wide ones. Senior managers also felt that challenges of collecting data across the global organization were not fully appreciated by the executive team, and not sufficiently resourced.

A highly significant finding was the extent to which the company culture was project-driven. Focus on delivering projects on-time means that schedule-orientated success metrics take precedence over measurements that show success at delivering in terms of quality and organizational goals. Project managers tend to operate independently - viewing each project as unique - without regard to projects running in parallel. As a result, lessons learned through project reviews tend to get shelved and are not cross-referenced again. A project-centric culture can inhibit the incremental introduction of a metrics program. It runs the risk of the metrics program being just another project with a definable start and finish date and key deliverables.

5.2 Management practices

Interviewees felt that since the metrics program went global, that data was collected in order to blame individuals for poor performance with a view to forcing behavioral change. Initially, the metrics were viewed as being genuinely geared towards improvement rather than towards enforcing change. We found that when a metrics program is viewed in this negative light, it is more likely that data will be manipulated to show untrue results. There was a widespread reluctance to highlight failures or lack of knowledge to management. It was felt that 'bad' news should be buried as it could hinder career advancement. However when poor results are not admitted to, lessons are not learned.

Developers were frequently not included in the metrics definition process and consequently the measures were often seen as impractical or unsuitable. Developers also seemed to resist measures because they did not trust the motives of management. There was also a strong resistance to the use of outside experts, and a prevailing view that the company had all the expertise they needed in-house.

5.3 People

There was no specific training provided to staff in measurement activities and it is expected that people engaged in this activity as part of their everyday roles. Similarly there was a lack of appropriate training for middle management.

Middle management development programs incorporate two narrow strands: a people management strand and project management certification. There is no training on wider strategic operations management skills, or industry best practices such as ISO 20000, Capability Maturity Model Integration (CMMI) or Information Technology Infrastructure Library (ITIL). Voluntary roles being filled by people unsuited to the role was also a problem that was noted.

In terms of involving metrics champions in the program, there was no skills database in operation. Interviewees opined that individuals selected to champion the program were those who had good relationships with management, and that the skills of individuals further down the organization were ignored.

There are no random audits done on data. This leads the data open to manipulation: if individuals feel they are overworked, they will supply whatever they can; if they are afraid of being negatively impacted by "bad" data they will omit the "bad" data.

Practitioners felt they have no meaningful input into the measurement process because they have no control over or input into what data is sensible to collect. This leads to an attitude of "it is what it is" towards the quality of data gathering. Practitioners also were found to hold the view that that GSC as an organization will not reward good measurement behavior but will punish for bad behavior.

5.4 Information and communication

Most developers were not aware of the existence of the GSC measurement program. There are no communication processes which enable developers to view this information. Thus developers do not see how the organization has benefited from their measurement efforts, and momentum is lost.

We learned that project managers cannot interpret the relevance of the metrics to their particular function, and therefore are unable to deduce what components of an activity needs targeted improvement. For example, when a metric report highlights below-target project budget and schedule results, the reason why cannot be clearly seen. Whilst there are categories of reasons for failure, no in-depth analysis occurs as to what contributed to the failure. It seems that this loop should be closed.

Much of the information on the metrics used in the GSC measurement program is used to generate statistics on adherence to particular processes. For example, the percentage of project managers that are certified, the percentage of projects that went through a development audit, and percentages of staff have received 'Secure Code' training, and so forth. However, the value of these activities is not clearly demonstrated or reported on. This has caused employees to be resentful of carrying out these activities, resulting in compliance being lower than it should or could be.

We found that senior managers implicitly trust the data that is passed upwards. They believe that the threat of deliberate misinformation carrying disciplinary penalties is adequate to ensure the "real" story is being told. Interviewees viewed this trust as being misplaced in some instances.

Control mechanisms surrounding project development audits, technical reviews and secure code reviews are relationship oriented or subjectively (as opposed to objectively). The level of scrutiny an audit panel can bring to the measurement program depends on who they are reviewing, and there were instances noted of passing projects that should fail an audit. This leads to a perception that key audit processes are "toothless" in reality and foster a culture of "process for process sake". In time, the audit reports are not taken seriously, but are produced to provide a number for the measurement report.

Practitioners feel that if senior management knew the level of effort that goes into manual data gathering, they would be more committed to automation. Senior management, on the other hand understood the difficulties inherent in automation – they noted that only when the measurement processes, data accuracy, reporting mechanisms, verified and visible use of the data are all in place can automation be attempted. Better communication of one another's viewpoint would help achieve a more positive commitment to automation.

Key measurement reports do not get communicated across or down the organization. Many practitioners and middle managers do not see the end-result of their data gathering endeavors and therefore do not see the value of what they are doing. There was also no voluntary cascading of data as personnel do not see the benefit for themselves in doing it.

5.5 Technology

Significantly for a technology organization, this was the lowest scoring category, emphasizing the importance of the interpersonal and behavioral aspects to measurement. A large degree of autonomy was given to the site during the initial phase meaning that choices of technology were based on local preference, rather than on compatibility with the technology at other sites. This is being addressed now, but there is a long timeline needed to address the current challenge of consolidation. This has also meant that where it is reasonable to deploy a minor tool in a location as a test-bed for wider organizational use, this gets little support from senior management. For example, an automated peer code review application was developed by a graduate intake program group. This would have simplified peer code review report and data gathering and aided timely submission. It was not accepted, even as an interim solution, on the basis that a larger enterprise-wide suite of tools would have the necessary functionality in time.

In later phases it became mandatory to use a particular software suite for many data-related processes on project billing, time, compliance, and utilization. However many interviewees find the current tool unwieldy and non-user-friendly. The same is true of a problem and incident management tool. But a strong cultural bias of "Made in GSC" inhibits decision-makers as to what makes sense from a functional and value perspective with regard to using third-party tools.

Many expressed the urgent need to have information systems and technologies that enable data to be aggregated within a single repository, so that various analyses can be performed on the same raw data. Currently in GSC there are separate systems for processing customer-billable hours of work, timesheet entry, email, expense reporting and vacation management. The desire by executive management to have information quickly, especially in times of transformation and change, was cited as the reasons for a reluctance to consolidate data onto one system as it could potentially involve delays.

Developers strongly believe that it is not possible to compare 'like with like' in any meaningful way in terms of productivity measurement. They cited the range of diverse technologies coupled with the varying complexities of each environment as a major reason. The GSC development environment involves a mixture of pure development projects, enhancement projects, technical support projects, a wide variety of hardware platforms, a wide variety of development tools, and many versions of the tools.

6. Discussion and recommendations

This research uncovered new information in terms of the underlying reasons for non-observance of measurement program implementation best practice. The key contributory issues are summarized as follows:

- A project-based view of measurement prevents customer-centric measurement.
- A silo mentality to work processes rather than a holistic and enterprise-wide approach restricts measurement in a global organization.
- Managing by fear inhibits a lessons learned culture developing in measurement.
- Executive management not being aware of the true levels of change, and the ad hoc nature of change initiatives.
- Over-reliance on relationships-oriented work practices leading to subjective rather than objective measurement decision-making.
- Metrics based on measuring adherence to activities rather than measuring the true value-add of the activities.

- Personal motives and politics, and lack of data audits which lead to data manipulation and measurement role issues.

A number of the characteristics of the measurement program in GSC concurred with the problems identified in the literature; amongst these were the difficulties inherent in measuring software activities and a lack of congruent communication between the different layers in the organization. Moreover, a project focus, rather than a goal orientated approach meant that many of the employees in GSC could not see that value-add of the measurement program. Furthermore, the need for an effective framework to link organization improvement goals with measurement goals ensuring appropriate data collection was borne out by the research. Our study also revealed a disparity in how senior management and practitioners engage with the process.

In light of this analysis we propose the following recommendations to practitioners.

6.1 Adopt a holistic approach

In many organizations there are a number of transformation programs running in parallel. These initiatives exert pressure on personnel, processes and activities in a measurement program. Often there is no process in place to streamline the resources used on the various programs. The result is that there are multiple levels of data gathering, analysis and reporting, leading to over-burdened personnel and inefficient resource use. Executive and senior managers must try to view the organization holistically and see the overlaps within information technology, software engineering, product measurement and program/project management.

6.2 Customer centric best practice

Management must be more conscious of industry best practice. In large scale organizations, there can be a view that the organization's own practices are unique, and are the only ones that fit the organization. Especially in technology companies, there can be an over-reliance on in-house technology and tools. Rigid adherence to established company process can lead to a belief that other practices are not translatable to the organization. Executive management should consider an industry best practice think-space among executive, senior and middle manager by ensuring regular attendance and contribution to appropriate industry fora. In relation to productivity measurement itself, the end-product or service, not the delivery mechanism – the project – ought to be the focal point of productivity measurements. These measurements can therefore include true external performance indicators based on quality, reliability, cost, functionality, speed and durability rather than on the more inward-looking measures of delivery cost, delivery speed and functionality, which typical projects measure.

6.3 Goal centered decision making

Management decisions frequently do not lead to productivity improvement and do not have a clear link to organization goals. To address this, a management-decision audit system should be put in place, where all key decisions must demonstrate the business justification behind them, linking them to the organization goals. One of the many goal centered models could be used as a core for this audit system. Furthermore a clearly understood productivity model should be put in place that contains outputs that can be quantified using a standard productivity equation thereby demonstrating value-add and costs.

7. Conclusion

The findings of our analysis support two key claims of this research. Firstly despite the availability of best practice factors for program measurement implementers, there are many underlying factors that impede effective implementation. Secondly, an analysis of the real reasons for the non-compliance of best practice, rather than simply highlighting non-observed factors is critical to solving the measurement program implementation issues that are pervasive in the software engineering industry. Research on measurement program implementation in software

engineering primarily focus on validating new, and re-validating existing, best practice for measurement program success. Yet, the majority of measurement program implementations continue to fail in the longer term. This research has highlighted a number of underlying reasons that need to be addressed holistically if measurement implementation success rates are to be improved. Program managers can check if staff is trained or untrained, involved in design or not involved, whether suitable data repositories and information processing technologies are deployed or not deployed, and so forth. However, unless the intangible aspects: the psychological motivators and de-motivators of personnel, their true capabilities, abilities to cope with and manage change, to learn new technologies, to lead people effectively, to articulate a vision of improvement, and create a holistic learning organization culture – are attended to, the gap between measurement program implementation knowledge and actual measurement program implementation effectiveness will remain. The findings presented in this study are the first step towards addressing that gap.

These findings deepen and expand our current understanding of the area in a real world setting and provide a unique insight into the area of performance management in software development. In sum the contributions of our study are three fold. The study has answered calls for research, provided new insights on variables and uncovered novel relationships between the constructs studied. However it is important to note that this study is not without its limitations. This study focuses on a single site case study and the findings may be seen as context specific and difficult to generalise. In addition, the data collection method employed in the study relied on self-reported data which can be difficult to verify independently. In other words, we were obliged to take what respondents say during interviews at face value. However certain biases could be apparent such as; selective memory (i.e. only recalling or not remembering experiences or events); positive attribution (i.e. attributing positive events and outcomes to themselves but attributing negative events and outcomes to external forces) and exaggeration (i.e. representing outcomes as more significant than is actually the case). Future research could be conducted in multiple case settings to calibrate the internal and external validity of our findings.

References

[1] G. A Rummler and A. P Brache, *Improving performance: How to manage the white space on the organization chart*. John Wiley & Sons, 2012.

[2] R. Colomo-Palacios, C. Casado-Lumbreras, P. Soto-Acosta, F. J. García-Peñalvo and E. Tovar, "Project managers in global software development teams: a study of the effects on productivity and performance," *Software Quality Journal*, vol. *22*, no. *1*, pp. *3-19*, 2014

[3] K. El Emam, and A. Gunes Koru, "A replicated survey of IT software project failures," *Software, IEEE*, vol. *25*, no. *5*, pp. *84-90*, 2008.

[4] S. Y. T. Lee, H. W. Kim, and S. Gupta, "Measuring open source software success," *Omega*, vol. *37*, no. *2*, pp. *426-438*, 2009.

[5] N. Fenton, P. Krause, and M. Neil. "Software measurement: Uncertainty and causal modelling," *IEEE software*, vol. *9*, no. *4*, pp. *116-122*, 2002.

[6] F. García, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini, and M. Genero, "Towards a consistent terminology for software measurement," *Information and Software Technology*, vol. *48*, no. *8*, pp. *631-644*, 2006.

[7] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Munch, D. Rombach, and A. Trendowicz, "Linking software development and business strategy through measurement," *IEEE Computer*, vol. *43*, pp. *57–65*, 2010.

[8] H. Edison, N. B. Ali, and R. Torkar. "Towards innovation measurement in the software industry," *Journal of Systems and Software*, vol. *86*, no.*5*, pp. *1390-1407*, 2013.

[9] N. Sekitoleko, F., Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H. H. Olsson, "Technical Dependency Challenges in Large-Scale Agile Software Development," in *Agile Processes in Software Engineering and Extreme Programming*. Springer International Publishing, 2014, pp. *46-61*.

[10]. J. C. Munson, *Software engineering measurement*, CRC Press, 2013.

[11] M. Unterkalmsteiner, T. Gorschek, A. M. Islam, C. K. Cheng, R.B. Permadi, and R. Feldt, "Evaluation and measurement of software process improvement—A systematic literature review," *Software Engineering, IEEE Transactions on*, vol. *38*, no. 2, pp. *398-424*, 2012.

[12] M. Kasunic, "The state of software measurement practice: Results of 2006 Survey," Software Engineering Institute Technical Report CMU/SEI-2006-TR-009, 2006.

[13] T. A. T. Izhar, T. Torabi, M. I. Bhatti, and F. Liu, "Recent developments in the organization goals conformance using ontology, " *Expert Systems with Applications*, vol. *40*, no. *10*, pp. *4252-4267*, 2013.

[14] B. Kitchenham, "What's up with software metrics?–A preliminary mapping study," *Journal of systems and software*, vol. *83*, no. *1*, pp. *37-51*, 2010.

[15] C. Ingram and S. Riddle, *Cost-benefits of Traceability*, Springer: London, 2012, pp. *23-4*.

[16] V. R. Basili, G. Caldiera, and H. D. Rombach, "Goal, question, metric paradigm," in *Encyclopaedia of Software Engineering*, vol. *1*, J. J. Marciniak, Ed., John Wiley & Sons, 1994, pp. *528–532*.

[17] R. Kaplan and D. Norton, "The balanced scorecard—measures that drive performance," *Harvard Business Review, January/February*, pp. *71*, 1992.

[18] B. List, R. M. Bruckner and J. Kapaun, "Holistic software process performance measurement from the stakeholders' perspective," In *Proceedings of 16th International Workshop on Database and Expert Systems Applications, IEEE,* pp. *941-947*, 2005.

[19] W. Hasselbring and R. Reussner, "Toward trustworthy software systems," *Computer*, vol. *39*, no. *4*, pp. *91-92*, 2006.

[20] C. Chittister and Y. Y. Haimes, "Risk associated with software development: a holistic framework for assessment and management," *IEEE Transactions on Systems, Man and Cybernetics*, vol. *23*, no. *3*, pp. *710-723*, 1993.

[21] J. H.Iversen, L. Mathiassen and P. A. Nielsen, "Managing risk in software process improvement: An action research approach," *MIS Quarterly*, vol. *28*, no. *3*, pp. *395-433*, 2004.

[22] M. Völter, T. Stahl, J. Bettin, A. Haase, and S. Helsen, *Model-driven software development: technology, engineering, management*. John Wiley & Sons, 2013.

[23] F. Niessink and D. van Vliet, "Measurement program success factors revisited," *Information and Software Technology*, vol. *43*, pp. *617-628*, 2001.

[24] C. Larman and V. R. Basili, "Iterative and incremental development: A brief history," *Computer*, vol. *36*, no. *6*, pp. *47-56*, 2003.

[25] T. Hall and N. Fenton, "Implementing effective software metrics programs," *IEEE software*, vol. *14*, no. *2*, pp. *55-65*, 1997.

[26] A. Rainer and T. Hall, "A quantitative and qualitative analysis of factors affecting software processes," *The Journal of Systems and Software*, vol. *66*, pp. *7-21*, 2003.

[27] D. D. Phan, "Software quality and management – How the world's most powerful software makers do it" *Information Systems Management, Winter*, pp.*56-67*, 2001.

[28] C. R. Pandian, *Software metrics: A guide to planning, analysis, and application*, CRC Press, 2003.

[29] M. B. Miles and A. M. Huberman, *Qualitative data analysis: An expanded sourcebook*, Sage, 1994.

[30] O. Tansey, "Process tracing and elite interviewing: a case for non-probability sampling." *PS: Political Science & Politics*, vol. *40*, no. *4*, pp. *765-772*, 2007.

[31] M. Q. Patton, *Qualitative evaluation and research methods*, SAGE Publications, Inc., 1990.

[32] R. K. Yin, *Case study research: Design and methods*, Sage publications, 2014.

[33] J. Rowley, "Using Case Studies in Research," *Management Research News*, vol. *25*, no. *1*, pp. *16-27*, 2002.

Biographical notes

Thomas Murphy

Thomas Murphy is an experienced Operations Manager, IT Project Manager and business PMO analyst with strong people management, business and leadership skills developed over many years in the IT and Performance Improvement sectors. He has worked in many leading organizations such as in software development, project management and problem solving roles. He is a creative, flexible and dedicated professional who is results driven and committed to high standards of performance and operational excellence.

Kathryn Cormican

Kathryn Cormican (Ph.D.) lectures in the College of Engineering & Informatics at the National University of Ireland Galway. Her research interests lies in the areas of new product development and enterprise systems. Kathryn leads a number of research projects in this area. She has published widely at international conferences and peer reviewed journals. Kathryn works closely with many leading organizations and SMEs helping them to design, develop and deploy new processes and systems.