Evidence-Based Software Portfolio Management

Hennie Huijgens
Delft University of Technology,
Delft, The Netherlands
h.k.m.huijgens@tudelft.nl

ABSTRACT
In this paper, we describe the research proposal for an approach for Evidence-Based Software Portfolio Management; a new way to help software companies in steering their software portfolio's based on cost, duration, defects found on the one hand and stakeholder satisfaction and perceived value on the other. The research approach is based on instruments such as a Cost / Duration Matrix, the identification of success and failure factors for software projects, and the collection of data on finalized software projects from portfolios of different companies in a research repository.

Categories and Subject Descriptors
D.2.8 [**Software Engineering**]: Metrics – *Process Metrics, Performance Measures.*

General Terms
Management, Measurement, Economics.

Keywords
Evidence-Based Software Engineering, Software Portfolio Management, Software Benchmarking, Software Economics.

1. INTRODUCTION
The goal of data-driven software portfolio management is to use project data collected from the past to predict and monitor the success of other software projects, now and in the future. In such a portfolio management perspective, measuring project size, project costs, project duration and post-release defects is a common practice. Nevertheless, these core metrics only tell a part of the story, and as such companies should be careful in steering their software project portfolios on these data points alone.

It could, after all be possible that a specific project costing twice as much as typical for its size would still be highly valuable to the organization. Performing within time and cost constraints is important, but especially in environments that use agile approaches additional goals enter the arena, such as early delivery of valuable software and an increased focus on stakeholder satisfaction.

Where many other studies use either a quantitative approach (e.g. analyze core metrics) or a qualitative approach (e.g. perform surveys or interviews) to analyze software projects, we combine both ways and look at a company's software project portfolio from a holistic point of view. The goal of our research is to combine a quantitative, data-driven approach on analysis of finalized software project portfolios with a qualitative, survey-based approach in order to identify factors related to project success and failure, in combination with an approach to measure and analyze stakeholder satisfaction and perceived value of software projects.

In this paper, we describe the research proposal for the development of evidence-based software portfolio management as a practical approach on organizing and decision-making with regard to large portfolios of software projects in information-intensive companies. In particular, the main contributions in the current state of the research are:

1. We propose a Cost / Duration Matrix as an instrument for analysis of good practice and bad practice in large, company-wide portfolios of software projects.
2. We identify success and failure factors for software projects, based on analysis of a large subset of data of finalized software projects from three different companies.
3. We analyze series of software releases in order to identify additional factors that contributes to projects being best-in-class.
4. We propose a light-weight value measurement technique based on quantitative analysis and post-project interviews.
5. We provide data of industrial software projects on a standardized set of metrics: project size, cost, duration, and defects. We contrast these core metrics with collected data on stakeholder satisfaction and perceived value, and look for links between them.

The remainder of this paper is organized in the following way: In Section 2 we outline relevant prior work. In Section 3 we describe our research objectives and questions. In Section 4 our research approach is described. Section 5 is about the most important metrics with regard to our research and Section 6 is about the data analysis methods and techniques that we apply. In Section 7 we evaluate validity threats. Finally, Section 8 includes a summary of the current status of our research and planned next steps.

2. BACKGROUND AND RELATED WORK
In this section we describe a brief survey of the background of our research area and related work with regard to our research subject.

A common idea of many research performed in the former millennium is that success and failure of software projects are interconnected with process-based activities: in other words, follow the process and success will come [1] [2] [3] [4] [5] [6] [7]. More recent work emphasizes the success and failure factors of shorter iterations due to an agile way of working [8] [9] [10].

From the millennium onwards concepts such as agile and added value become important factors in software engineering [11]. And one of the effects that can be seen in industry nowadays is that the instruments of the "old" world, such as algorithm based estimation, functional size measurement, and measurement and analysis seem not to go together with “new” tools such as story points, planning poker, and less focus on control and documentation.

Shepperd argues that “the primary goal of more accurate cost prediction systems remains largely unachieved” [12]. Software engineering economics is likely to remain very challenging, as is for example showed in recent research that undermines the long-lasting application of algorithmic cost models [13]. Nevertheless the need for good economic models will grow rather than diminish as software becomes increasingly ubiquitous [12]. Besides that combinations of effort estimation methods in many cases show better results than single effort estimation methods [14].

In practice many software companies perform benchmarking of their software activities, often based on measurement of the functional size of projects and software applications [15]. Yet, a growing variety of available benchmarks, including large differences in outcomes of analyses on different benchmark sources, does not always help to make life easier for decision makers involved in software development [16].

Varieties of value-based software engineering are examined [17] [18] [19]. However, a clear link with existing approaches that focus on measurements such as project size, project cost, project duration, and number of defects is not to be found. A challenge in industrial practice is that usually several approaches for estimation, monitoring and control, and benchmarking of software projects are in place and that, replacing an organizational process at once is not feasible due to technical and social issues. However the change could be adjusted incrementally, as for example is found in [20].

Recent research on motivation [21] [22] [23] shows that, although it is difficult to quantify, motivation is considered to an important factor in software developer productivity. There are also suggestions that low motivation is an important factor in software development project failure [23].

3. RESEARCH OBJECTIVES

We define as our research objective: “an integrated approach that links existing measurements based on project size, cost, durations, and defects of software projects with a limited set of relatively easy to collect additional measures on stakeholder satisfaction and perceived value, and additional qualitative research on the backgrounds of project success and failure”.

Based on the above we define the following research questions:

RQ1: In what way are project size, project cost, project duration, and process quality (measured in number of defects) in a software project portfolio correlated?

RQ2: What factors can be found that influence a company’s software project portfolio in a positive or negative way?

RQ3: What factors can be found that characterize ‘best-in-class’ software projects?

RQ4: Are function points (FPs) compatible with story points (SPs) on agile projects?

RQ5: Can a statistical, empirical, evidence-based pricing approach for software engineering, be used as a single instrument (without a connection with expert judgment), in distributed environments to create cost transparency and performance management of software project portfolios?

RQ6: In what way are project size, project cost, project duration, and process quality (measured in number of defects) in a software project portfolio correlated with stakeholder satisfaction of finalized software projects?

> RQ6.1: Is process quality, measured in number of defects found during a project, an early indicator for stakeholder satisfaction?

RQ7: In what way are project size, project cost, project duration, and process quality (measured in number of defects) in a software project portfolio correlated with perceived value of finalized software projects?

> RQ7.1: Is project size, measured in function points (FPs), an early indicator for perceived value of finalized software projects?

4. PROPOSED APPROACH

In this section we describe the approach that we use in order to answer the research questions as stated above. Because we perform our research in close cooperation with software companies (to be read as information-intensive companies, such as banks, telecom companies, governmental organizations), we set-up the research in a way that fits with practice. Where appropriate, we perform case studies [24] [25]. The case studies that we perform are mixed studies: we perform both quantitative and qualitative research on the subject projects within a portfolio as a whole of a company or organization. Our focus is not to study single software projects, but instead look at the effects of all software projects performed over a period of time in a portfolio as a whole; by doing so we assume to analyze both good practice projects and bad practice projects.

Where applicable we will use electronic surveys among stakeholders of software projects, supplemented with non-structured interviews as techniques to challenge findings from the quantitative analysis.

A precondition that limits our research approach is the fact that we perform research in real, live organizational environments. Therefore the approach must not interfere with the daily operation of the studied software projects. Surveys should impose limited burden on people, and analysis is usually to be useful for improvement purposes in daily operations.

5. IMPORTANT METRICS

For our research we make use of an existing data set of 352 finalized software projects from three different organizations. This research repository is collected over a period of four years, receding our research. Table 1 gives an overview of the organization of this research repository. During our research we continue the collection of data of finalized software projects; the research repository will mature during the development of the research, both in number of projects and in applied metrics.

Based on the collected metrics as inventoried in Table 1 we calculate three key performance indicators.

1. Cost per FP;
2. Duration per FP;
3. Defects per FP.

For all three indicators we use project size (FPs) as the weighting factor (instead of number of projects). In order to measure stakeholder satisfaction and perceived value we ask all stakeholders of a finalized software project (e.g. project manager, business representative, product owner, business analyst, scrum master, developer, and tester) to rate scores for both metrics on a 5-point scale. Stakeholder satisfaction is measured for both satisfaction with regard to the projects process and the deliverables of the

Table 1. Overview of the initial research repository; the number of projects in the repository will grow over time.

Category	Type	Occurrence	N	Definition of Project Factors
Company ID (ORG)	Nominal	3	352	Identification code of the company where a project was performed; three companies were applicable (nr. of occurrence between brackets): B1 (206), B2 (125), T (23).
Project ID	Nominal	352	352	Identification code of a project.
Year of Go Life	Ordinal	6	352	Year when a project was finalized; the following years Go Live were applicable: 2008 (32), 2009 (59), 2010 (81), 2011 (131), 2012 (41), 2013 (10).
Business Domain (BD)	Nominal	10	352	Customers business sector; the following BD were applicable: Finance & Risk (54), Internet & Mobile (54), Payments (50), Client & Account Management (incl. CRM systems) (46), Savings & Loans (40), Organization (incl. HRM) (31), Call Centre Solutions (21), Mortgages (21), Data warehouse & BI (18), Front Office Solutions (17).
Primary Programming Language (PPL)	Nominal	21	352	Primary used programming language; the following PPL were applicable: JAVA (154), .NET (59), COBOL (55), ORACLE (29), SQL (9), 3GL (8, unknown was what specific languages were applicable here), Visual Basic (6), RPG (6), FOCUS (5), PowerBuilder (5), PRISMA (4), MAESTRO (3). In the analysis 4th Generation (1), PL1 (1), JSP (1), C++ (1), Clipper (1), Document (1), PL/SQL (1), Siebel (1) and Package (1, unknown what specific language was applicable) were referred at as Other.
Delivery Model (DM)	Nominal	2	352	Classification of the used delivery model; two DM were applicable: Structured (e.g. Waterfall) (307), and Agile (Scrum) (45). One project reported as DM RUP is included in the analysis of Structured.
Development Class (DC)	Nominal	4	352	Classification of the development; the following DC were applicable: New development (173), Major enhancement (25-75% new) (124), Minor enhancement (5-25% new) (27), Conversion (28).
Project Keyword (KW)	Nominal	20	351	Characteristics on a specific project (multiple keywords could be mapped on one project, on one project no keyword was mapped); the following keywords were applicable: Single-application (270), Business driven (150), Release-based (one application) (144), Once-only project (122), Phased project (part of program) (65), Fixed, experienced team (62),), Technology driven (58), Steady heartbeat (49), Dependencies with other systems (41), Migration (35), Rules & Regulations driven (33), Multi-application release (21), Many team changes, inexperienced team (17), Package with customization (16), Legacy (15), Security (14), Pilot; Proof of Concept (10), Bad relation with external supplier (9), New technology, framework solution (3), Package off-the-shelf (1).
Measure	**Type**	**Occurrence***	**N**	**For every project in the repository the measurements indicated in the table below are inventoried.**
Size (FP)	Ratio	-	352	Size of a project in Function Points (FPs).
Duration	Ratio	-	352	Duration of a project in Months; measured from the start of Project Initiation to (technical) Go Live.
Cost	Ratio	-	352	Cost of a project in Euros; measured from the start of Project Initiation to (technical) Go Live.
Effort	Ratio	-	352	Effort spent in a project in Person Hours (PHRs); measured from the start of Project Initiation to (technical) Go Live.
Defects	Ratio	-	172	The number of errors or faults found in a project from System Integration Test to (technical) Go Live. Not for all projects defects were administrated; for 172 projects defects info was recorded in the repository.

*No occurrences are indicated for the 5 Measures, due to the fact that these are different for every measured project.

project (the product). Perceived value is measured for four aspects: a company's customer, financial, internal process, and innovation.

6. DATA ANALYSIS TECHNIQUES

In our research we make use of different data analysis techniques, as described in the following paragraphs.

6.1 The Cost / Duration Matrix

The most important data analysis instrument that we use is a so-called Cost / Duration Matrix (see Figure 1). This matrix is a model based on power regression of project cost (Euros) versus project size (FPs) and project duration (months) versus project size (FPs). For both regressions the percentage deviation from the mean is calculated for each software project. This percentage deviation is for both cost and duration plotted in a plotter chart; the Cost / Duration Matrix.

As Figure 1 visualizes the matrix shows larger or smaller dots, depending on the size in FPs of a specific project. A color ranging from blue to red indicates the process quality (number of defects per FP) of each project, where blue stand for a good process quality and red for a bad quality (meaning more than average defects per FP for a specific project). The Cost / Duration Matrix (see Figure 1) is used as a model to visualize and assess the performance in terms of cost, duration and quality of software projects based on four quadrants that describe specific characterizations [26] [27]:

- Good Practice (upper right): This quadrant shows software projects that scored better than average of the total repository (or a specific subset of the repository) for both cost and duration.
- Cost over Time (bottom right): In this quadrant software projects are reported that scored better than the average of the total repository (or a specific subset of the repository) for cost, yet worse than average for duration.
- Bad Practice (bottom left): This quadrant holds software projects that scored worse than average of the total repository

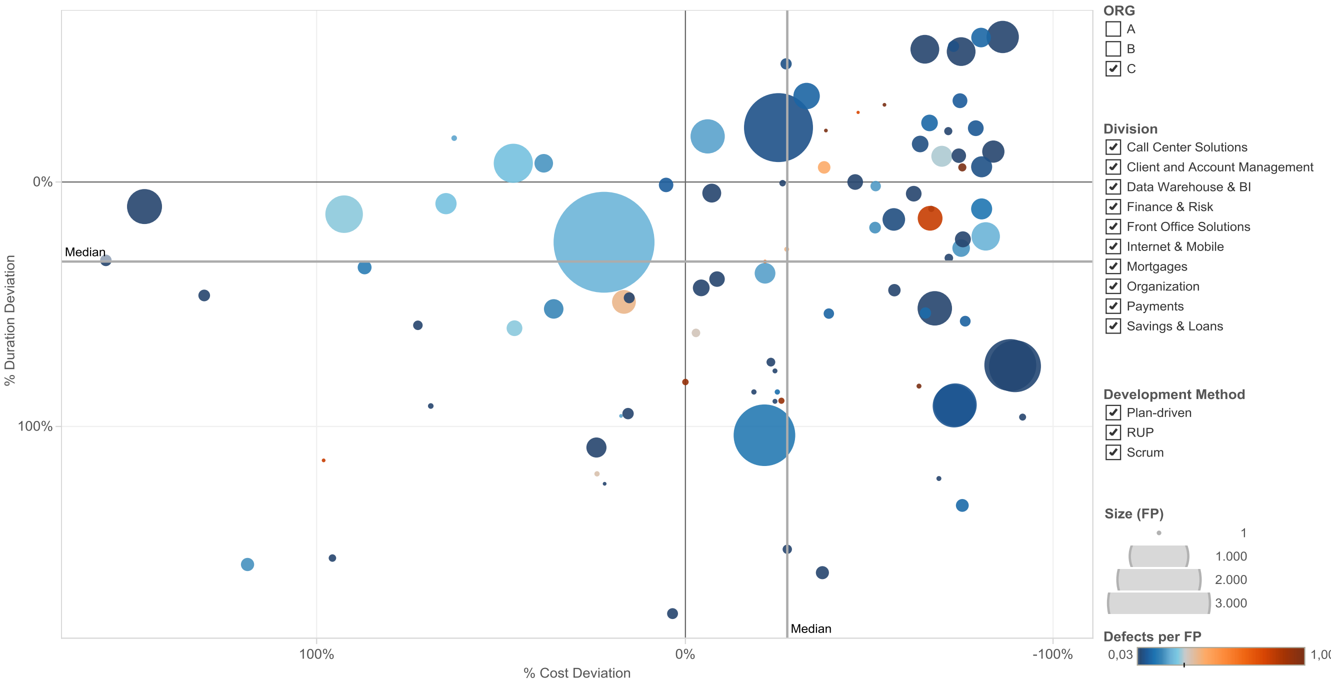

Figure 1. The Cost/Duration Matrix, representing a subset of finalized software projects.

(or a specific subset of the repository) for bot cost and duration.

- Cost over Time (upper left): In this quadrant software projects are plotted that scored better than average of the total repository (or a specific subset of the repository) for duration, and worse than average for project cost.

Keep in mind that the underlying nominator for all software projects in the Cost / Duration Quadrant is functional size (measured in FPs). Due to this we can compare the performance in terms of cost, duration, defects found, satisfaction, and value of projects with different sizes with each other.

6.2 A Software Project Benchmark Tool

We develop a Software Project Benchmark Tool that enables the functionality for practitioners in industry to benchmark a subset of finalized software projects against our research repository. The tool is based on the Cost / Duration Matrix and makes it possible for measurement practitioners in industry to upload a subset of finalized software projects from their own organization and to benchmark the performance in terms of cost, duration, and defects found with that of comparable projects in our research repository.

6.3 Stakeholder Satisfaction and Value

In order to add new metrics such as Stakeholder Satisfaction and Perceived Value of finalized software projects to our research repository, we build a survey questionnaire that is send to stakeholders once software projects are finalized. We add the metrics resulting from this questionnaire to our research repository and relate the outcomes with software projects in the four quadrants of the Cost / Duration Matrix.

6.4 Prioritize projects in a portfolio

Finally we develop and describe Evidence-Based Software Portfolio Management, as an approach for software companies to prioritize software activities within their software project portfolio, based on quantification of project size, cost, duration, defects found, stakeholder satisfaction, and perceived value of finalized software projects. We test this approach as a whole in a different information-intensive organization to examine whether the outcomes correlate with those companies that are already available in our research repository, and to analyze whether the approach can be used in practice as a valuable addition to tools already in place for a software project portfolio management capability in organizations.

7. VALIDITY THREATS

With regard to validity constraints we assess the following threats.

We use function point analysis (FPA) as a way to normalize software projects and to make it possible to compare performances of projects with different sizes. We use functional documentation as a source for FPA. A consequence is that low quality documentation can lead to low quality FPAs. However, we thoroughly review all sets of documentation on completeness and correctness and have FPAs performed by experienced, and in many cases certified, experts. FPAs are reviewed by different experts than the ones that performed the count itself to prevent from bias. With regard to data quality we argue that all project data is reviewed by the applicable project managers. All data is discussed with the applicable company management and the financial controller.

By normalizing all project data with the functional size in FPs we warrant internal validity, the extent to which a causal conclusion is based on our study. Due to this we can objectively compare performances of all software projects, in order to minimize systematic error. The effect of outliers is limited and the risk on bias is mitigated responsibly based on the diversity of projects and business domains within each subject company, the number of software projects, and the fact that we measure and analyze software project portfolios as a whole in an empirical way.

8. CURRENT STATUS AND NEXT STEPS

8.1 Finalized research

At this moment the following research results are in place:

8.1.1 Good Practice versus Bad Practice (RQ1/RQ2)

We analyzed a dataset containing 352 finalized software projects, with the goal to discover what factors affect software project performance, and what actions can be taken to increase project performance when building a software project portfolio. The software projects are classified in four quadrants of a cost/duration matrix: analysis is performed on factors that are strongly related to two of those quadrants, Good Practices and Bad Practices. A ranking is performed on the factors based on statistical significance, resulting in an inventory of 'what factors should be embraced when building a project portfolio?' (Success Factors), and 'what factors should be avoided when doing so?' (Failure Factors). This research result is documented in a paper that was accepted at ICSE 2014, SEIP-track [26].

8.1.2 Best-in-class software projects (RQ3)

We aimed to identify distinguishing factors in software releases. For this purpose we analyzed the metrics of 26 software projects. These projects were release-based deliveries from two stable, experienced development teams in a Banking company. During the measurement period both teams transformed from a plan-driven delivery model (waterfall) to an agile approach (Scrum). Overall, we observed that these small release-based projects differ largely from non-release-based projects. Our research indicates that a combination of release-based working, a fixed and experienced development team, and a steady heartbeat contribute to performances that can be characterized as best practice. This research result is documented in a paper that was accepted at IWSM-Mensura 2013 [27].

A case study that replicates the research above, with additional qualitative research on a series of best-in-class releases from another Telecom company is described in a paper that is to be submitted.

8.1.3 Story Points versus Function Points (RQ4)

In order to find differences and similarities between two many used size metrics we replicated a study on the relation between story points and function points performed in 2011 by a group of Brazilian researchers. We used data collected in a Banking organization. Based on a statistical correlation test we conclude that it appears too early to make generic claims on the relation between function points and story points; in fact FSM-theory seems to underpin that such a relationship is a spurious one. The results of this research were published in a paper that was accepted at WETSoM 2014 [28].

8.1.4 Pricing via functional size (RQ5)

We analyzed how a medium-sized west-European telecom company experienced a worsening trend in performance, indicating that the organization did not learn from history, in combination with much time and energy spent on preparation and review of project proposals. In order to create more transparency in the supplier proposal process a pilot was started on Functional Size Measurement pricing (FSM-pricing). In our research we evaluated the implementation of FSM-pricing in the software engineering domain of the company, as an instrument useful in the context of software management and supplier proposal pricing. We found that a statistical, empirical, evidence-based pricing approach for software engineering, as a single instrument (without a connection with expert judgment), can be used in distributed environments to create cost transparency and performance management of software project portfolios. A research paper on our research results is accepted at ESEM 2015 [29].

8.2 Studies in preparation

The following research topics are to be studied in the remaining part of the research period:

8.2.1 Software project benchmark tool

We developed a tool based on the cost/duration matrix as used in [26] and [27] with the purpose to support software companies to benchmark the performance of their own software delivery against 400 finalized software projects in our research repository. We validate the tool by analyzing the performance of a subset of finalized software projects from the ISBSG repository [30].

8.2.2 Stakeholder Satisfaction and Perceived Value (RQ6 and RQ7)

As a point on the horizon we will focus on research on the quantification of Stakeholder Satisfaction and Perceived Value of software projects, related to the cost/duration matrix that we defined in earlier research (see Figure 1). We enrich this model by mapping Stakeholder Satisfaction and Perceived Value with regard to a company's customers, financial, internal process and innovation aspects to cost, duration and quality of finalized software projects. A paper on this subject, including a survey on five finalized projects in a Telecom company is in preparation.

ACKNOWLEDGMENT

I thank Arie van Deursen and Rini van Solingen for their great support and work as advisors for my PhD activities. Furthermore I thank all companies that support our research for their generosity to allow us to use company data for research purposes.

REFERENCES

[1] T. Hall, A. Rainer and N. Baddoo, "Implementing Software Process Improvement: An Empirical Study," *Software Process Improvement and Practice,* vol. 7, pp. 3-15, 2002.

[2] J. Reel, "Critical Success Factors in Software Projects," *IEEE Software,* Vols. May-June, pp. 18-23, 1999.

[3] T. Dyba, "An Empirical Investigation of the Key Factors for Success in Software Process Improvement," *IEEE Transactions on Software Engineering,* vol. 31, no. 5, pp. 410-424, 2005.

[4] T. Dybå, "Factors of Software Process Improvement Success in Small and Large Organizations: an Emperical Study in the Scandinavian Context," in *ESEC/FSE,* Helsinki, Finland, 2003.

[5] M. Niazi, D. Wilson and D. Zowgh, "Critical Success Factors for Software Process Improvement Implementation: An Empirical Study," *Software Process Improvement and Practice,* vol. 11, pp. 193-211, 2006.

[6] A. Rainer and T. Hall, "Key success factors for implementing software process improvement: a maturity-based analysis," *Journal of Systems and Software,* vol. 62, no. 2, pp. 71-84, 2002.

[7] D. Stelzer and W. Mellis, "Success Factors of Organizational Change in Software Process Improvement," *Software Process Improvement and Practice,* vol. 4, pp. 227-250, 1998.

[8] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *The Journal of Systems and Software*, vol. 81, pp. 961-971, 2008.

[9] S. C. Misra, V. Kumar and U. Kumar, "Identifying some important success factors in adopting agile software development practices," *The Journal of Systems and Software,* vol. 82, pp. 1869-1890, 2009.

[10] J. Sutherland, A. Viktorov, J. Blount and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in *40th International Conference on System Sciences*, Hawaii, 2007.

[11] B. Boehm, "A View of 20th and 21st Century Software Engineering," in *IEEE International Conference on Software Engineering (ICSE)*, Shanghai, Ghina, 2006.

[12] M. Shepperd, "Software project economics: a roadmap," in *Future of Software Engineering (FOSE)*, Minneapolis, USA, 2007.

[13] H. Suelmann, "Putnam's Effort-Duration Trade-Off Law: Is the Software Estimation Problem Really Solved?," in *IEEE IWSM-Mensura*, Rotterdam, The Netherlands, 2014.

[14] E. Kocaguneli, T. Menzies and J. W. Keung, "On the value of ensemble effort estimation," *IEEE Transactions on Software Engineering,* vol. 38, no. 6, pp. 1403-1416, 2012.

[15] A. F. Minkiewicz, "The Evolution of Software Size: A Search for Value," *Software Engineering Technology,* vol. March/April, pp. 23-26, 2009.

[16] C. Jones, "Sources of Software Benchmarks," Capers Jones & Associates, 2011.

[17] B. Boehm, "Value-Based Software Engineering," *ACM SIGSOFT Software Engineering Notes,* vol. 28, no. 2, pp. 1-12, 2003.

[18] S. Biffl, A. Aurum, B. Boehm, H. Erdogmus and P. Grünbacher, Value-Based Software Engineering, Berlin Heidelberg: Springer, 2006.

[19] S. Faulk, D. Harmon and D. Raffo, "Value-Based Software Engineering (VBSE): A Value-Driven Approach to Product-Line Engineering," in *First International Conference on Software Product-Line Engineering*, 2000.

[20] J. Keung, R. Jeffery and B. Kitchenham, "The challenge of introducing a new software cost estimation technology into a small software organisation," in *IEEE Proceedings of the Australian Software Engineering Conference*, 2004.

[21] S. Beecham, N. Baddoo, T. Hall, H. Robinson and H. Sharp, "Motivation in Software Engineering: A Systematic Literature Review," *Elsevier - Information and Software Technology,* vol. 51, no. 1, pp. 219-233, 2009.

[22] H. Sharp, N. Baddoo, S. Beecham, T. Hall and H. Robinson, "Models of motivation in software engineering," *Elsevier - Information and Software Technology,* vol. 51, no. 1, p. 219–233, 2009.

[23] J. Verner, M. Babar, N. Cerpa, T. Hall and S. Beecham, "Factors that motivate software engineering teams: A four country empirical study," *Elsevier - The Journal of Systems and Software*, vol. 92, pp. 115-127, 2014.

[24] P. Runeson, M. Host, A. Rainer and B. Regnell, Case Study Research in Software Engineering; Guidelines and Examples, Hoboken, New Jersey. USA: John Wily & Sons, 2012.

[25] R. Yin, Case Study Research - Design and Methods, Los Angelos, USA: Sage Publications, 2008.

[26] H. Huijgens, R. v. Solingen and A. v. Deursen, "How To Build a Good Practice Software Project Portfolio?," *ICSE Companion 2014 Companion Proceedings of the 36th International Conference on Software Engineering (SEIP),* vol. 2014, no. IEEE, pp. 64-73, 2014.

[27] H. Huijgens and R. v. Solingen, "Measuring Best-in-Class Software Releases," *IWSM-MENSURA 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement,* no. IEEE, pp. 137-146, 2013.

[28] H. Huijgens and R. v. Solingen, "A replicated study on correlating agile team velocity measured in function and story points," *WETSoM 2014 Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics,* vol. 2014, no. ACM, pp. 30-36, 2014.

[29] H. Huijgens, G. Gousios and A. v. Deursen, "Pricing via Functional Size: A Case Study of 77 Outsourced Projects," in *IEEE 9th International Symposium on Empirical Software Engineering and Measurement (ESEM) (in press)*, Beijing, China, 2015.

[30] ISBSG, "International Software Benchmarking Standards Group," 1997. [Online]. Available: http://www.isbsg.org/isbsgnew.nsf/webpages/~GBL~Home. [Accessed 2014].