



Libraries and Learning Services

University of Auckland Research Repository, ResearchSpace

Version

This is the Accepted Manuscript version. This version is defined in the NISO recommended practice RP-8-2008 <http://www.niso.org/publications/rp/>

Suggested Reference

Ralph, P., & Kelly, P. (2014). The Dimensions of Software Engineering Success. In *ICSE 2014 Proceedings of the 36th International Conference on Software Engineering* (pp. 24-35). Hyderabad, India: ACM/IEEE.

doi: [10.1145/2568225.2568261](https://doi.org/10.1145/2568225.2568261)

Copyright

Items in ResearchSpace are protected by copyright, with all rights reserved, unless otherwise indicated. Previously published items are made available in accordance with the copyright policy of the publisher.

For more information, see [General copyright](#), [Publisher copyright](#).

The Dimensions of Software Engineering Success

Paul Ralph and Paul Kelly

Lancaster University

Lancaster, UK

paul@paulralph.name; paulrichardkelly@gmail.com

ABSTRACT

Software engineering research and practice are hampered by the lack of a well-understood, top-level dependent variable. Recent initiatives on General Theory of Software Engineering suggest a multifaceted variable – Software Engineering Success. However, its exact dimensions are unknown. This paper investigates the dimensions (not causes) of software engineering success. An interdisciplinary sample of 191 design professionals (68 in the software industry) were interviewed concerning their perceptions of success. Non-software designers (e.g. architects) were included to increase the breadth of ideas and facilitate comparative analysis. Transcripts were subjected to supervised, semi-automated semantic content analysis, including a software developer vs. other professionals comparison. Findings suggest that participants view their work as time-constrained projects with explicit clients and other stakeholders. Success depends on stakeholder impacts – financial, social, physical and emotional – and is understood through feedback. Concern with meeting explicit requirements is peculiar to software engineering and design is not equated with aesthetics in many other fields. Software engineering success is a complex multifaceted variable, which cannot sufficiently be explained by traditional dimensions including user satisfaction, profitability or meeting requirements, budgets and schedules. A proto-theory of success is proposed, which models success as the net impact on a particular stakeholder at a particular time. Stakeholder impacts are driven by project efficiency, artifact quality and market performance. Success is not additive, e.g., ‘low’ success for clients does not average with ‘high’ success for developers to make ‘moderate’ success overall; rather, a project may be simultaneously successful and unsuccessful from different perspectives.

Categories and Subject Descriptors

D.2.0 [Software Engineering]: General, D.2.8 [Software Engineering]: Metrics, D.2.9 [Software Engineering]: Management.

General Terms

Management, Measurement, Human Factors, Theory.

Keywords

Success, General Theory of Software Engineering, Interview, Semantic Analysis, Design, Interdisciplinary.

1. INTRODUCTION

What does it mean for a new software engineering (SE) technology or practice to be good? At one level, quality dimensions vary across artifact types. At a higher level, however, artifacts are good insofar as they positively affect success. But what does success mean in a software engineering context? What are its dimensions? How can we measure it?

While much research has addressed related topics including software quality, project success and information systems success, little research has focused on either the broad dimensions or the unique aspects of success in software engineering. Many SE academics and practitioners continue to conceptualize success in overly simplistic or contractual terms, e.g. ‘software is successful if it meets its requirements.’ Consequently, this paper investigated the following research question.

Research Question: What are the primary dimensions of software engineering success?

Here, software engineering includes everything from the initial conceptualization of the problem or project to the maintenance of existing software artifacts. Below, the paper proceeds by reviewing related work on project success, software quality and information systems success (§2). The methodology (§3) produces 11 themes (§4) but comparative analysis between SE and non-SE participants indicates only one element unique to software (§5). These insights contribute to an initial theory of SE success (§6), which is followed by a discussion of key implications (§7) and summary of the paper’s contribution (§8).

2. RELATED WORK

Recent workshops have revealed an emerging consensus that SE needs one, top-level dependent variable, which might simply be called *Software Engineering Success* (SES) [22, 26, 42]. A single variable is needed to facilitate both high-level theorizing about software engineering and meta-analyses of the effectiveness of different tools and practices. Some questions (e.g. *which is more important for software engineering, good architectural modeling or good unit testing?*) depend on a unifying top-level dependent variable to answer.

While the exact nature of this variable remains unclear, it is obviously multifaceted [42]. SE produces software artifacts with myriad effects and quality levels. Insofar as SE is organized into projects, these projects are subject to established dimensions of project success (e.g. completing on time). Like information systems, software artifacts must be adopted and used to produce many of their intended benefits. This section therefore takes an interdisciplinary perspective on success measures and constructs.

Measuring success is a major topic of interest in the project management literature. The *Project Triangle* or *Iron Triangle* posits that the quality of project outcomes is constrained by scope (how much we want to accomplish), budget (how much money we have to spend) and schedule (how much time we have) [4]. To an extent, reducing scope takes less time and less money, more money can make up for less time or larger scope, and more time can make up for less money or larger scope. While it has been used to understand project success since the 1950s or earlier [4], it is now clear that the project triangle is oversimplified and delivers insufficient accounts of project success [4, 5, 42-44, 52, 53]. Shenhar et al. [44] propose a more comprehensive taxonomy of project management success (Table 1).

Table 1: Project Success Dimensions (adapted from [44])

Success Dimension	Definition
project efficiency	meeting schedule and budget goals
impact on customer	meeting functional performance requirements, fulfilling customer needs, solving customer problems, customer using the product, customer satisfied
business success	commercial success, market share
preparing for the future	creating a new market or product line, developing a new technology

While a comprehensive review of the project management literature is beyond the scope of this paper, several immediately relevant findings may be noted. Project management studies (e.g. [3]) increasingly treat projects as vehicles for organizational learning, concordant with Shenhar et al.’s “preparing for the future” dimension. When a project results in a product, project success may be distinct from product success [6]. Where Shenhar et al. emphasize financial and tangible effects, others emphasize emotional impacts (e.g. satisfaction [29]) which may constitute a deficiency in Shenhar et al.’s model. Specific success criteria vary across industries and project types [33]. More generally, the vast literature on traditional project management practices is often disconnected from the realities of real projects – specifically, real-world performance criteria are often more complex and ambiguous than the conventional discourse recognizes [15].

Although “there is no common consensus as to what the concept of a stakeholder means” [31], Stakeholder Theory fundamentally posits that entities (including projects) are ethically responsible not only to an explicit client but also to other impacted individuals and groups [24]. This highlights a deficiency in Shenhar et al.’s taxonomy, which only includes impacts on “the customer”. Success perceptions may vary between stakeholders – in client-contractor relationships, “contractors put more emphasis on minimizing project cost and duration, whilst clients put more emphasis on satisfying the needs of other stakeholders” [12]. Moreover, objectives may vary between stakeholders and over time [18].

Moving to the software engineering literature, developers often view success differently from project managers. For example, developers see projects as successful if they get to do interesting work, are treated well and build high-quality software that meets user needs, regardless of meeting schedule and budget goals [30, 36]. However, “different stakeholders involved in the software development may attribute success to different indicators” [21].

More generally, the success of SE projects is complicated by the production of artifacts that often outlive the projects that create them. These artifacts may have quality and success dimensions that are largely independent from classic project success dimensions. For example, much SE research is concerned with software quality. In some studies (e.g. [10]), software quality is used as a dependent variable for evaluating a tool, technology or practice. Others (e.g. [8, 9, 14]) propose taxonomies of characteristics constituting software quality. The ISO/IEC 9126 standard organizes quality attributes into six dimensions – *functionality, reliability, usability, efficiency, maintainability and portability*. However, an empirical evaluation of the standard concludes “that ISO/IEC 9126 is not suitable for measuring design quality of software products” as “the standard was ambiguous in meaning, incomplete with respect to quality characteristics and overlapping with respect to measured properties” [1]. In summary, while academics and practitioners

appear to agree that software quality is important, no widespread agreement concerning its dimensions is evident.

This pattern is typical of *Essentially Contested Concepts* [25]. A concept is “essentially contested” if several parties agree it is important but have irreconcilable disagreements regarding its optimal instantiation. Essentially contested concepts assess an achievement, which is internally complex (such that many rival descriptions of the relationships between the achievements’ parts and overall nature are reasonable) and subject to unpredictable revisions following changing circumstances [25]. While software quality and indeed software engineering success seem to manifest these qualities, new innovations may lead to previously unforeseeable reconciliation, perhaps by redefining the term to unify the parties. (In our case, the proto-theory proposed below is an attempt at a reconciling innovation for SES.)

In contrast, the field of information systems manifests widespread agreement on the analogous concept of *Information Systems Success*, centered on DeLone and McLean’s model thereof [19, 20, 35] (Figure 1). This model emphasizes that benefits to stakeholders (including users, clients, developers and the public) are primarily determined by the quality of the system, the information it contains and the service it provides. However, it further posits that quality only translates into benefits insofar as the system is used and the users are satisfied. That is, if usage is low or users dissatisfied, the client reaps fewer rewards and the developer loses profits and reputation. While this focus on use likely derives from the intense interest in use within the IS community (cf. [50]), artifact use seems equally relevant in the SE context. Indeed, use/popularity is an important success metric for open source software [17]. Furthermore, the DeLone and McLean’s Model of IS Success illustrates how a multifaceted construct may actually include causal relationships – system quality, for example, is modeled as both part of success and causally related to use and user satisfaction.

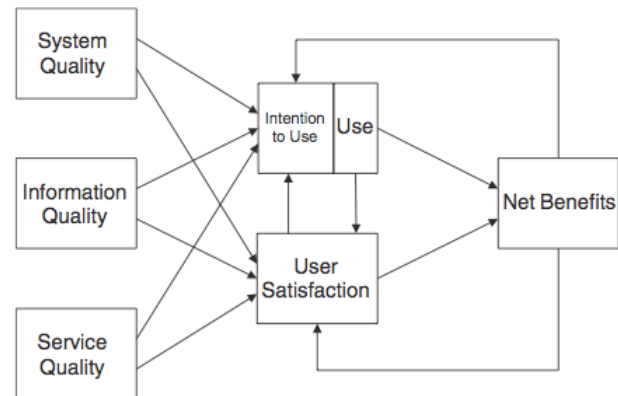


Figure 1: The DeLone and McLean Model of IS Success (adapted from [20])

In summary, existing research from several disciplines suggest myriad possible dimensions of Software Engineering Success including project efficiency, artifact quality, impact or benefits, satisfaction, stakeholders (internal and external), use, market success and learning.

3. RESEARCH METHODOLOGY

Briefly, we analyzed transcripts of interviews with diverse design professionals including software engineers. We used a semi-automated content analysis technique to derive 11 themes from the corpus. As this is an exploratory study, which aimed to

generate rather than to test theory, no upfront hypotheses are stated.

3.1. Sampling and Data Collection

At first we considered defining the population of interest as software developers. However, we expanded the population to design professionals in general for four reasons:

1. Software developers' success perceptions may be systematically biased in ways only obvious through contrast with other disciplines
2. Success perceptions vary across stakeholders [21]
3. Success perceptions vary across industries [33]
4. If existing success perceptions are oversimplified as suggested by the above literature review, more diverse perspectives appear more likely to generate a more nuanced picture.

Here, *design professionals* refers to individuals whose work includes a substantial amount of conceptualizing, analyzing, constructing or modifying virtual, tangible or socially constructed artifacts. We also included professionals who manage design work. Moreover, we did not restrict participants to any particular country, as different cultures and countries should add even more nuance [27].

Unfortunately, as no accurate population list of either software developers or interdisciplinary design professional is available, random sampling is not possible. We therefore used convenience sampling, aiming for high sample diversity. As this is an exploratory, theory-building study, an interdisciplinary convenience sample was deemed appropriate. Only currently employed professionals (not students) were interviewed.

More specifically, sampling was implemented through a series of course projects. Undergraduate and postgraduate management students were trained in interviewing and then recruited participants through their professional networks. Armed with an interview guide provided by the authors (§9), students conducted one or two interviews and transcribed the results. (The students later pooled their transcripts and analyzed them for course projects.) The diversity of students, many of whom were international, contributed to the diversity of the sample. Although student interviewers may not have been as effective as professional researchers would have been, they produced far more – and more diverse – interviews than would otherwise have been possible. Moreover, the authors view the interviews as quite good on average, and many professional SE researchers have no experience or formal training in interviewing before their first empirical projects.

A total of 191 professionals were interviewed between October 2011 and March 2013. Interviews were conducted face-to-face (19%) where possible, otherwise via Skype (62%) or telephone (19%). All interviews were conducted in English and transcribed. Interviewers adapted the interview guide (§9) based on the participant's industry and used follow-ups and probes to elicit more detailed responses.

Participants had an average (standard deviation) of 10.67 (9.45) years of experience and 4.45 (5.30) years in their present positions. Participants' highest educational qualification varied from none (i.e. leaving secondary school at age 16) to PhD, although most left school with a bachelors (48%) or masters (28%) degree. Participants came from a variety of industries (Table 2) although we intentionally included a disproportionate number of software developers. Categories having five or fewer participants were grouped under *other*. Participants represented diverse companies including Accenture, Airbus, Asus, BMW,

eBay, Dell, Ericsson, IBM, Infosys, KPMG, Novartis, Rolls Royce, Tata Consultancy Services, Thomson Reuters, Toyota and the UK National Health Service.

3.2. Analysis

The interview transcripts were analyzed using semi-automated content analysis in Leximancer. This analysis begins with unsupervised semantic mapping, which works as follows.

A unified body of text is examined to select a ranked list of important lexical terms on the basis of word frequency and co-occurrence usage. These terms then seed a bootstrapping thesaurus builder, which learns a set of classifiers from the text by iteratively extending the seed word definitions. The resulting weighted term classifiers are then referred to as concepts. Next, the text is classified using these concepts at a high resolution, which is normally every three sentences. This produces a concept index for the text and a concept co-occurrence matrix. By calculating the relative co-occurrence frequencies of the concepts, an asymmetric co-occurrence matrix is obtained. This matrix is used to produce a two-dimensional concept map via a novel emergent clustering algorithm. The connectedness of each concept in this semantic network is employed to generate a third hierarchical dimension, which displays the more general parent concepts at the higher levels. [45]

Table 2: Participant Industry

Category	Categories	
	Frequency	Percent
Software Development	68	36%
Management	32	17%
Other	22	12%
Art and Graphics	19	10%
Architecture	13	7%
Engineering (non-software)	13	7%
Financial Analysis and Accounting	9	5%
Product Design	8	4%
Marketing and Business	7	4%

Leximancer provides several features for analyzing the concept map, including *query* (examine interview excerpts containing a concept), *pathway* (examine chains of excerpts that link two concepts), *themes* and *concepts* (lists of themes and concepts ranked by relevance). The analyst can drill into a concept by retrieving all of its associated text snippets. Where the context is not clear from the snippet alone, the analyst can quickly view the full transcript to examine more context.

However, the concept map produced by unsupervised semantic mapping is often noisy. For example, our initial concept map contained the concept “of course” due to the frequency of participants saying “of course.” To refine the map, the analyst iterates between analyzing the concept map, editing the concept list and re-generating the map. The map cannot be edited directly (similar to inversion of control in graphical user interface design). During the refinement process, we made three types of changes to the concept list:

1. deleting irrelevant concepts (e.g. *course*)
2. merging synonymous concepts (e.g. *deadline* and *deadlines*)

- adding concepts that were identified by Leximancer but not recognized as relevant (e.g. *learn* as in learning from project failures)

In the course of this analysis, the authors read extensively from the interview transcripts to better understand and interpret the concept map. The discussion of the eleven themes in the following section is grounded in our theme-centric reading of transcript excerpts and subsequent interpretation based on existing literature, including the frameworks described in Section 2.

4. ELEVEN THEMES

The above analysis produced a rich map of the themes (Figure 2, Table 3) and concepts (Figure 3 and 4) evident in the dataset. In the theme and concept maps, hotter colors (red, orange) indicate greater importance while cooler colors (blue, purple) indicate lower importance. Theme size does not reflect importance or prevalence. To oversimplify slightly, the grey lines show the most direct concept interconnections while spatial proximity reflects the combination of direct and indirect relationship. As Figures 2 and 3 are simply different views of the same spanning tree, visual comparison shows which concepts are in which themes. This section discusses each theme's composition, interpretation and implications.

Below, quotation marks indicate direct quotations from participants. As English was some participants' second or even third language, some quotations may appear awkward; however, participants' original phrasing is preserved.

Table 3: Theme Connectivity

Theme	Connectivity	Informal Meaning
Project	100%	work is organized into projects
Client	95%	work is associated with specific customers
Time	73%	delivering work within a reasonable time and for a reasonable price is an important success dimension
Design	61%	a well designed artifact is an important success dimension
Work	58%	good working relationships with colleagues and management is an important success dimension
Emotion	14%	the emotional reactions of stakeholders constitute an important success dimension
Feedback	12%	feedback from stakeholders is a primary indicator of success
Designer	10%	the designer's work is critical for success
Analysis	6%	analysis is a core aspect of design work
Parts	3%	artifacts are made from components and component quality is an important success dimension
Winning	2%	market performance and market share are important success dimensions

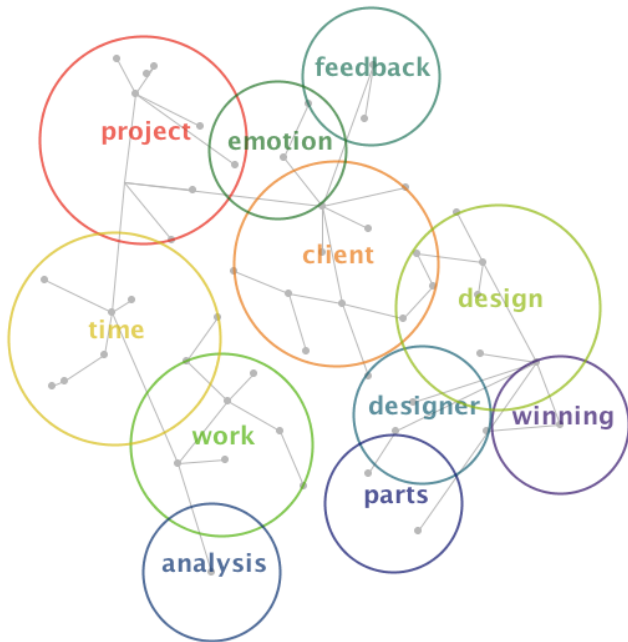


Figure 2: Thematic Map

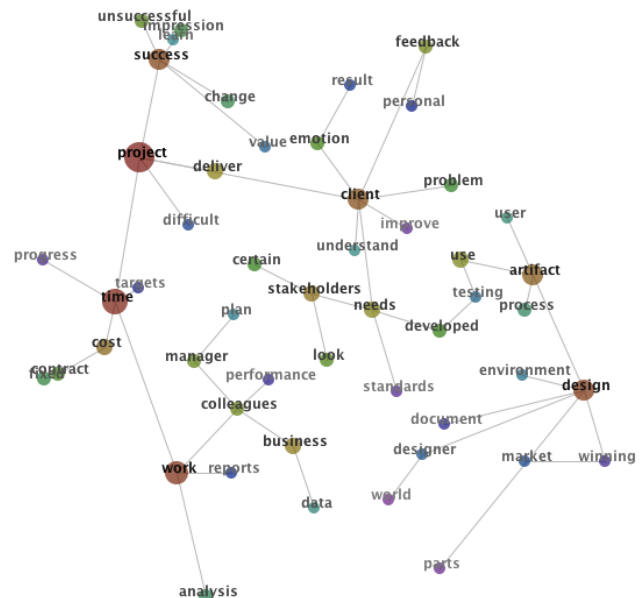


Figure 3: Concept Map¹

¹Blurry areas around cost and success are clarified in Figure 4.

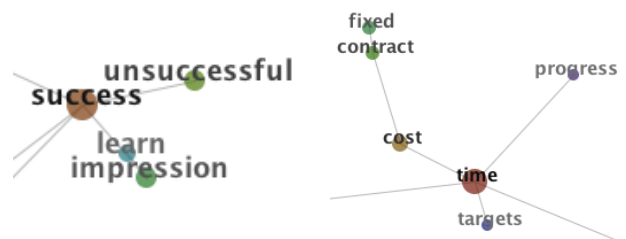


Figure 4: Success and Cost Subtrees Rotated for Clarity

4.1. Project

Although software engineering is not inherently project-based, participants largely perceive their work as being organized into projects; i.e., temporary, goal-oriented work systems [2, 41]. One participant explained, “the jobs basically come in project form”. Participants associate success and failure more closely with the project than with themselves, their organizations or the artifacts produced. Projects vary substantially in intrinsic difficulty, which affects success perception in that an improvement of a particular magnitude represents greater success in a more difficult project than a less difficult one. Learning, especially from mistakes, is closely related to success (Figure 4).

Project is the central (most connected) concept in the map. This suggests that theoretical distinctions between software *engineering* success and software *project* success are not practically important.

4.2. Client

While participants discuss numerous stakeholders, an explicit client or customer is often central to their conception of success. Clients are perceived as having problems to solve or needs to fulfill – “when the clients knew what to expect, based on their own expertise, they could be very clear about their own goals”. Understanding and addressing the client’s needs and problems is seen as a powerful determinant of success. Participants also recognize that success is related to the extent to which clients use provided artifacts; e.g., “we have 3 different consumers who all use [our product] in a different way”. The centrality of clients and client needs in participants’ conceptualizations of success is analogous to the traditional view of the firm, where the firm has a fiduciary duty to put the needs of its shareholders first. Stakeholder theory (above) holds that fixation on shareholder interests leads to unethical behavior; analogously, participants’ focus on clients above other stakeholders may lead to unethical behavior.

4.3. Time

The *Time* theme includes not only plans, schedules and deadlines but also cost and contracts. Participants felt that delivering a product within a reasonable time and for a reasonable cost is an important aspect of success, although time is mentioned more often than cost. Participants also felt that sticking to schedules, budgets, plans and other targets impact success. One participant explained: “generally we have a one-, two-, sometimes even three-year contract with review points and we follow the review points”; another said “you have to follow the budget otherwise you cannot meet your objectives”. Clearly, however, being efficient is not equivalent to meeting time and cost goals as the latter are not always reasonable or possible; e.g., “If they understand us, they will give us reasonable schedule. But if they don’t know our profession...”. Although not all development involves a contract, many participants emphasized the importance of meeting contract terms; e.g. “it’s always important to stick to the contract”. Despite the proliferation of fixed-price/-schedule contracts, however, some participants emphasized that contracts are not set in stone; e.g., “It is important to stick to the contract since that shows that you are doing the work as you committed. However, contracts could be negotiable if there are any unavoidable circumstances.”

4.4. Design

Steve Jobs famously said, “Design is a funny word. Some people think design means how it looks. But of course, if you dig deeper, it’s really how it works.” Indeed, the denotation and connotations of *design* vary substantially across fields [41]. For example, while

software engineers often contrast design with analysis, others maintain that analysis and design are the same cognitive process [16, 37, 40]. In some fields (e.g. fashion design, interior design) design is strongly associated with aesthetics; e.g., “I would firstly check ... if it looks smudgy, if the design is properly spaced”. However, other participants speak of designing objects that have few aesthetic dimensions (e.g. engine components). Yet others, including an industrial designer who designs trams and trains, suggest that aesthetic and functional design dimensions are closely related and determined simultaneously. Moreover, for some participants (e.g. architects, urban planners) design is presented as a kind of planning – separate from construction. Other participants meanwhile see design and construction as fundamentally the same process; e.g. “We had this *design* competition on boats we had to *make*, which could take on a lot of weight” (italics added).

A subtle linguistic difference is evident here. Where participants from other fields say *design*, participants from software projects use two different words – *design* and *develop*. Develop is usually used broadly to refer to the whole process of conceptualizing, specifying, creating and maintaining artifacts, while design is often restricted to either aesthetic dimensions or architectural planning. SE’s uniquely restricted conceptualization of *design* may indicate either that software design is fundamentally different from other kinds of designing or (more likely) that conceptual distinctions between design and development are misleading. The latter ties into existing critiques (e.g. [38]) of technical rationality as the dominant paradigm in SE.

Interestingly, no participants mentioned a relationship between designing and refactoring as is common in the Agile literature [7].

The *Design* theme includes the concepts of the artifact (or product) and its design, testing and use. Participants felt that good design and effective artifacts (products) are important aspects of success, somewhat independent of project variables; e.g. “Is the product very safe or not?”. Participants also recognized that artifacts are used by users; therefore both use and impacts on users are relevant to success. Moreover, many participants discussed the importance of testing and passing tests for understanding success. Finally, participants acknowledged that an artifact is designed for a particular environment and its success may therefore vary by context.

It should be noted here that at lower levels of abstraction, these concepts would be divided into two themes, roughly, the artifact theme (the product and its use) and the design theme (the process of designing). The main points are that success is related to good design and that the success of an artifact is somewhat distinct from the success of the project that creates it.

4.5. Work

Unsurprisingly, participants spoke at length about their work, including the business they work for, their colleagues and their managers. Many participants indicated that their work is data-intensive and often involves reading or writing reports. Participants not only see their work as causally connected to success but also conceptualize the nature of their work as an aspect of success; e.g. “when I first started, success to me was working a reasonable houred day”. Having interesting, engaging or motivating work was seen as an important aspect of personal success. Similarly, working with a high-performing team of colleagues under reasonable management was considered highly desirable. Following a major success, one participant recounted: “We were treated by our lead and manager; we went out for a dinner together, had a great time and our manager spoke few lines about each one of us and really complimented us as a team.”

However, this having-good-work aspect of success is semantically remote from the project, suggesting that project success and having-good-work are perceived as relatively independent.

4.6. Emotion

The emotion concept, which includes satisfaction, is equally connected to the concepts *client* and *personal*, as in client satisfaction and personal satisfaction.

Participant descriptions of client reactions suggest that emotion mediates the relationship between clients and project results. In other words, success is not only about artifacts meeting specifications or projects meeting contractual obligations but also about how the client feels about the results. Many participants made statements similar to “it’s important that the client is happy with what you’ve done”. One explained: “the client ... has to feel completely, completely at ease in that space.” Participants’ descriptions of clients feeling happy or satisfied suggest that some aspects of success are fundamentally based on aesthetics, sentiment and instinct rather than rationality and utility – consistent with existing literature on emotional design [34]. Participants’ language also suggests that many decisions are based on what feels right rather than logically defensible reasoning.

Furthermore, participants discussed their own emotional relationships to their work, projects, colleagues and artifacts. Their language suggests that personal success is in some ways a feeling – I am successful if I feel successful – e.g. “It gives me an immense pleasure being associated with an organization like [this one] ... I feel that sense of pride”.

Moreover, participants may be deeply frustrated when their feelings about a project conflict with the client’s feelings. For example, one participant explained, “For me personally, it is first important if I am happy with my work.” Another told a story “where the client had got his caterer to do a design ... and I said you don’t want that, you want something else. It wasn’t a good design. So we did another design but the client said he would rather have what the caterer designed. And that was a horrible feeling because ... that was a disaster but the client was happy.”

4.7. Feedback

Participants indicated that stakeholder feedback is a primary mechanism for understanding success (e.g. “Obviously, the lead and manager’s feedback matters to me a lot but if I get feedback straight from the client it really holds importance”), specifically how the project has solved client problems, fulfilled client needs and generally impacted the client. Feedback is associated with clients more often than other stakeholders. Here, feedback may be formal or informal, direct or indirect.

4.8. Designer/Developer

The structure of the semantic map suggests a close relationship between the design and the designer or developer. That is, the designer is important by virtue of designing the artifact, which is one of the core success dimensions. Unsurprisingly, participants (who are mostly designers) described designers as being central to success; e.g. “If a particular client asks me to do a particular project and I think that adding other features as a developer would make the product excellent, then I go ahead with it and it usually turns out to be more than what the client was expecting.” This theme may be inflated relative to the other themes by myside bias [47, 48] – i.e., favoring one’s own perspective, role and context.

4.9. Analysis

Participants indicated that much of their work involves analysis. Analysis is closely related to the concepts of data and reports, i.e.,

participants analyze data and produce reports from their analysis. In some cases, analysis and design are combined in a person’s title, e.g., *programmer analyst*. Participants clearly felt that good analysis was critical to success; e.g., “Second part is the analysis and writing the test cases. That’s another very very important factor because a right test case is the one that would really help you in designing a successful project”.

4.10. Parts

Participants indicated that artifacts are composed of parts or components (e.g., “the good that we sell is made up of many parts”) and that designs are similarly specified in terms of these parts (e.g., “I have one subordinate who works for designing some parts which I have to evaluate”). Moreover, properties of individual parts could be crucial for success, e.g., one participant explained, “The parts also had to have high ductility in the casting material for crash resistance.”

4.11. Winning

In some cases participants described successes as a relative phenomenon (e.g. “we have to win in a competitive market place”, “fortunately we won in this competitive market”). This supports Shenhar et al.’s [44] third dimension. It also suggests that success depends on both financial success and market share, which do not necessarily covary.

5. WHAT’S SPECIAL ABOUT SE?

To determine whether participants in SE roles conceptualize success differently from participants in non-SE roles, we divided transcripts into two groups – *software development* (n=68) and *everyone else* (n=123). We independently repeated the same analysis used above on each group. While the resulting spanning trees have many small differences, overall concept usage is remarkably similar (Table 4). Small differences are evident in vocabulary (e.g. the SE group more often says “software” or “system” where the non-SE group says “product”) and ordering (e.g. the SE group mentions “use” more often than the non-SE group). However the top four concepts – *project*, *design*, *work* and *success* – are identical. Moreover, some concepts that appear for one group but not the other (e.g., *team*, *analysis*) are included in the other group’s concept map – just not in the top 20.

In fact, the only substantive, meaningful difference revealed by this analysis is that the SE group is concerned with *requirements* while the non-SE group has no equivalent concept. This could be a non-representative artifact of the data. Alternatively, software engineering may be systematically different from other design disciplines (including other fields of engineering) insofar as projects involve many identifiable requirements. In contrast, this may support the theory that software requirements are a kind of mass delusion within the SE community [11, 39]. Given the many problems with requirements (e.g. instability [13], undermining creativity [32]), simply abandoning the notion of requirements may be preferable. If engineers, architects, product designers and artists can get by without requirements, maybe software developers can too.

6. DIMENSIONS OF SE SUCCESS

The eleven themes identified in this study suggest that Software Engineering Success is a multidimensional variable; however, the themes do not map directly into dimensions. For example, *feedback* is an indicator rather than a dimension of success. This section therefore synthesizes a proto-theory of SES by combining participants views expressed with existing literature.

Table 4: SE vs. Non-SE Concept Ranking

Software Engineering (n=68)		All Others (n=123)	
Concept	Relevance	Concept	Relevance
project	100%	project	100%
design	58%	work	79%
work	54%	success	66%
success	40%	design	62%
use	36%	time	36%
client	29%	client	36%
time	29%	people	35%
<i>requirements</i>	23%	job	24%
team	19%	doing	24%
people	18%	look	23%
feedback	16%	use	22%
job	15%	different	22%
software	14%	company	22%
doing	14%	feedback	19%
customer	14%	example	19%
important	14%	customer	17%
manager	13%	product	16%
company	13%	analysis	15%
different	12%	process	12%
system	12%	change	11%

6.1. Impact on Stakeholders

Many participants mentioned different kinds of stakeholders. Two of the themes (*client* and *designer*) and several of the concepts (including *manager*, *colleagues*, *business* and *users*) are types of stakeholders. Moreover, the top-level dependent variable in the DeLone and McLean model (above) is *Net Benefits*, which includes benefits for all stakeholders, not just the client [20].

Clearly, a project may simultaneously benefit some stakeholders and harm others. For example, developers working overtime on a project may experience a decrease in emotional wellbeing, yet build a system that generates large profits for the business. The same system may benefit users by easing their workload but fail to produce performance improvements for the client organization. Consequently the same project may be perceived as successful by some stakeholders and unsuccessful by others. It therefore appears that success is stakeholder-specific.

Furthermore, *stakeholder satisfaction* is obviously insufficient to capture the breadth of impacts. Here we are concerned with financial, emotional, social and physical impacts, all of which vary across time. Moreover, while satisfaction is always perceptual, impacts may be analyzed as perceived, objective or both, depending on the context.

6.2. Project Efficiency

The *Time* theme includes not only time and schedule (targets) but also cost and contracts. This gels with both the project triangle

(above) and the project efficiency dimension of Shenhar et al.'s [44] taxonomy.

However, participants discussed at least two parallel senses of project efficiency. In one sense, successful projects meet explicit cost, schedule and scope targets. However, targets are often capricious and unrealistic. In another sense, therefore, a good project is one having a high scope-to-resources (including time and money) ratio, regardless of targets. This latter sense is consistent with views expressed by software developers in previous case studies (e.g. [30]).

This suggests that project efficiency combines both the project's scope-to-resources ratio and the relationship of that ratio to the plans, schedules, budgets, goals and contracts governing the project. This conflict over whether to conceptualize success in terms of a plan or *a priori* goal relates to a deeper theoretical and philosophical dispute over whether human action is plan-centered or improvisational (cf. [49]).

6.3. Artifact Quality

The *Design* theme evidences the perceived importance of well-designed artifacts for SES. This concurs with extensive research on code quality (see above) and the *system quality* construct of the DeLone and McLean model. Unsurprisingly, Shenhar et al.'s model does include a similar construct – stakeholders and efficiency are common to projects in general but not all projects create artifacts. Here, *artifact quality* refers to an artifact's intrinsic characteristics rather than its impacts in particular contexts.

While we would expect artifact quality to be related to stakeholder reactions and project efficiency, exceptions are obvious in at least three senses. First, late and over-budget projects may produce good designs; indeed, the pressure to produce a better design may be what drives a project to exceed its budget and schedule. For example, the Hubble Space Telescope, an evidently well-designed artifact, substantially exceeded its schedule and budget. Second, various stakeholders may like a poorly-designed system or dislike a well-designed system. For example, the United States Navy appeared quite satisfied with their radar interface until the USS Vincennes shot down a passenger airline by mistake because the radar's user interface was so unnecessarily confusing [51]. Third, artifacts may evolve unpredictably and be used by stakeholders in ways unforeseen by designers. The Internet is an obvious example – the inventors of packet switching could not have foreseen World of Warcraft and Instagram.

6.4. Market Performance

To a lesser extent, participants also mentioned financial success, use and winning in a competitive market. These appear as *use*, *market* and *winning* in the concept map. Moreover, in the DeLone and McLean model, financial success is part of *net benefits* and use is included explicitly. Financial success and winning are closely related to the *business success* dimension of Shenhar et al.'s taxonomy.

However, the *Market Performance* dimension combines three distinct things. First, for commercial products, making large profits is obviously an aspect of success. Second, however, products may engage in a more direct competition. For example, not long ago Sony and Toshiba engaged in a format war over the next optical disc format standard. Sony's Blu-ray unambiguously won the format war in February 2008 when Toshiba conceded and stopped developing HD-DVD. Regardless of Blu-ray's ensuing profitability, in this strictly competitive sense Blu-ray succeeded where HD-DVD failed. Third, some products are judged more by the extent of their adoption and use. For example, the Apache

server project is successful in the sense that it is widely used, despite not winning an outright war or being highly profitable.

Breaking market performance into profitability, use and winning outright conflicts is particularly useful in SE where free systems often compete against for-profit systems in the same competitive space. In such spaces, a free system may be successful in that it is well-used while a for-profit may simultaneously be successful by generating reasonable profits.

6.5. (The Other) Time

Time, as a theme in both the existing literature on success and the above data analysis concerns delivering a product on time. However, time appears crucial to understanding SES in another, apparently less salient sense: the way we understand a project’s successfulness varies over time.

It seems intuitively obvious that impacts on stakeholders will manifest at different times. For example, when a project begins, the developers will be impacted early by their new work, while the client may not be impacted until the first version is ready. Similarly, market performance may not exist early in a project, while blowing the budget may cease to matter years later if the product is highly profitable. Moreover, participants may initially understand artifact quality from test results but later understand artifact quality from user feedback. While not all projects follow these specific patterns, it appears inescapable that success-understanding varies over time.

Participants in this study largely did not express the view that success varies across time. However, some participants gave examples of serious problems with a design discovered years after project completion, e.g., “a pressure type part and we had been making it for like 4 years and all of a sudden they had some parts that started leaking in Mexico”. Likewise, the existing studies discussed above do not explicitly model success-understanding over time although it is strongly implied; e.g., Shenhar et al.’s inclusion of *learning for the future* implies a temporal dimension while causation in DeLone and McLean’s model implies a temporal sequence. Time is nevertheless included here explicitly as it appears to be an unstated assumption underlying existing models and known phenomena. Additionally, the alternative – that success-understanding is invariant across time – seems incredulous.

6.6. Toward a Theoretical Framework

The above five proto-theoretical dimensions are not all equivalent in kind. Rather, they suggest that software engineering success is defined by the following triple.

$$SES = \{Net\ Impact, Stakeholder, Time\}$$

That is, software engineering success is the cumulative effect of a software engineering initiative on a particular stakeholder as of a particular time. Meanwhile, Project Efficiency, Artifact Quality and Market Success are theorized as the primary contributors to Net Impact (Figure 5; Table 5). The same initiative may therefore be highly successful for the business in June but relatively unsuccessful for the employees in July. Obviously, the effect of project, artifact and market vary across time, e.g., poor project efficiency may initially pose serious problems, which are later overwhelmed by strong market performance.

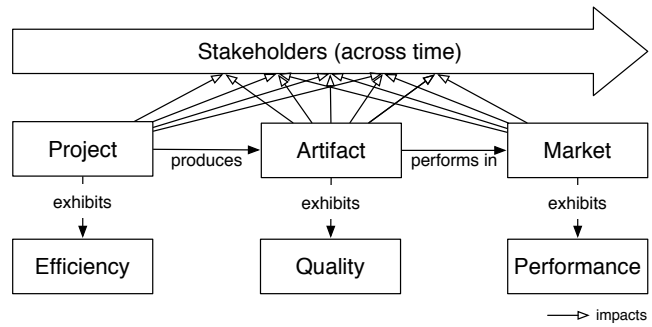


Figure 5: Software Engineering Success Framework

Table 5: Dimensions of Software Engineering Success

Dimension	Meaning	S? ¹
Stakeholders	Everyone impacted by a project, including developers, management, clients, customers and the public	Yes
Project Efficiency	The ratio of scope delivered to resources consumed and its relationship to plans, schedules, budgets, goals and contracts	Yes
Artifact Quality	Properties of the design and artifact independent of performance in specific contexts, e.g., cohesion, coupling, stability, ease of use	Yes
Market Performance	The extent to which an artifact is adopted, used, profitable and defeats competing artifacts	Yes
Time	Success-understanding varies across time, i.e., a project may appear successful before a catastrophic failure occurs.	No

¹Supported by the data collected in this study

In this view, SES is inherently relative to a particular stakeholder. One cannot necessarily average together “high” success for the client and “low” success for the developer to get “medium” success. Such averages are meaningless – the project is simultaneously successful and unsuccessful depending on whose perspective is taken. This relative conception of success supports more nuanced reasoning about success and hinders oversimplification and overgeneralization.

Moreover, SE artifacts and projects simultaneously have intrinsic success dimensions (quality and efficiency) and extrinsic success dimensions (effects on stakeholders). While we theorize that intrinsic success dimensions are causally related to extrinsic success dimensions, this obviously requires empirical investigation.

7. DISCUSSION

The proposed framework extends Shenhar et al.’s taxonomy by adding the *artifact* construct, extends DeLone and McLean’s model by adding the *project* construct, and includes the myriad dimensions of code quality within *artifact quality*. Concerning Shenhar et al.’s other constructs, *project efficiency* is included directly, *business success* is replaced by *market performance* and *impact on customer* and *preparing for the future* are subsumed by the multiple *impacts on stakeholders*. (As the business is a stakeholder, being more prepared for the future is a stakeholder impact.) Concerning DeLone and McLean’s other constructs, the three qualities are compressed into *artifact quality*; *use* is included in *market performance*; *user satisfaction* and *net benefits* are included in *stakeholder impacts*. (Users are stakeholders.)

Furthermore, this view suggests a truly multifarious success construct. It is tempting to conceptualize projects as either successful or unsuccessful. Rejecting this obvious false dichotomy, we are then tempted to conceptualize projects as lying on a spectrum from catastrophic failure to overwhelming success, with most somewhere in the middle. We are tempted to understand project success as a sum or average, e.g., if it goes a little over budget but produces a good artifact we think of it as “challenged” [46]. However, the proposed framework supports a more sophisticated view of success where *a project may be simultaneously successful and unsuccessful*. The Sydney Opera House, for example, overran its budget by 1400% and prevented architect Jørn Utzon from subsequent projects in Australia; yet it is considered an architectural masterpiece and a national symbol [23]. The Sydney Opera House is therefore better understood as simultaneously a raving success and a dismal failure than a ‘moderately successful’ or ‘challenged project’. This unprocessed view may resist oversimplification and over-rationalization to support deeper insight into project outcomes.

Practically then, to better understand success, any one project may need a diverse portfolio of metrics. Moreover, as the nature of success changes during and after a project, project actors may fluidly reprioritize and replace success metrics. For example, budget adherence may be less important early in a project, very important at delivery and later replaced by profitability.

Moreover, the proposed dimensions may manifest causal relationships within the success construct, as in the DeLone and McLean model. This raises an interesting concern in theorizing about success. Suppose we have a causal chain of the form (where $x \rightarrow y$ indicates that x causes y): *Modeling Technique* \rightarrow *Model Quality* \rightarrow *Artifact Quality* \rightarrow *Impacts on Stakeholders*. In the present framework, we draw a metaphorical line between *Model Quality* (e.g., the quality of UML diagrams) and *Artifact Quality* – the constructs to the right of the line are part of success while the constructs on the left are antecedents of success (Figure 6). The precise demarcation between antecedents and success dimensions is not capricious but neither is it objective. *Impacts on Stakeholders* is clearly part of success while *Modeling Technique* is clearly an antecedent. However, in certain contexts, e.g., where better understanding a problem domain is a major goal, some people may feel that high *Model Quality* is an end in itself and would therefore shift the line leftward. Meanwhile, an unadulterated cynic, who perhaps believes that practically all software is poorly designed, may feel that *Impacts on Stakeholders* is the only real success dimension and would therefore shift the line rightward. In summary the demarcation between success dimensions and antecedents is inherently fuzzy and the meaning of success may therefore expand and contract depending on the context.

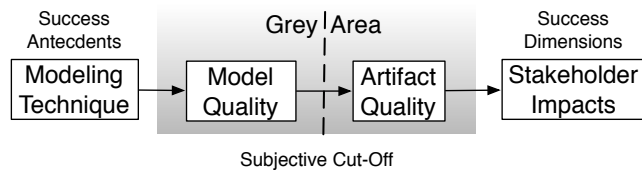


Figure 6: Subjective Demarcation between Antecedents and Dimensions of Success

Finally, the proposed framework may generalize beyond software engineering projects to a broader class of design projects. While much of the above discussion is software-centric, the core concepts of stakeholders, projects and artifacts appear equally applicable to industrial design, product design, architectural and

engineering projects. Clearly, however, more research is needed to investigate success in these domains.

8. CONCLUSION

The primary contribution of this study is the synthesis of a framework for understanding software engineering success. SES is modeled as a multidimensional variable comprising project efficiency, artifact quality, market performance and stakeholder impacts over time. This framework integrates previous research on code quality, information systems success, project success and software engineering in practice. The framework is supported by an extensive, interdisciplinary interview study of design professionals.

Additionally, the study produced several findings that are relevant to but not encompassed by the framework:

- Design work is largely organized into projects
- Designers are often fixated on an explicit client
- Designers appear more concerned with being on time than adhering to budgets or contracts
- The aesthetic connotation of design is not universal
- Designers desire interesting work and capable colleagues
- Designers are more concerned with clients’ emotional reactions to their products than with satisfying explicit requirements
- Designers perceive analysis and design as closely related
- Designers recognize that contracts, plans, schedules and budgets are often unreasonable or misguided.

This contribution should be interpreted within several limitations. First, the sample of interviewees may be systematically biased in some way. Although we can say that the sample is *diverse*, we cannot say it is *representative* as random sampling was not feasible. Second, as the theme/concept map reduces an n -dimensional matrix to a 2-dimensional projection, many visualizations are possible and small changes in seed concepts may therefore appear to significantly alter the organization of the map. Consequently, the thematic map (Figure 2) is somewhat unstable; however, the concept map (Figure 3) appears quite stable. For example, the concept map contains the concepts *design*, *designer*, *market* and *winning*. The thematic map organizes these concepts into three themes: *design*, *designer* and *winning*. As we refined the theme concepts, the *design* and *designer* themes would sometimes merge, other times *designer* merged with *work*. Sometimes *winning* was part of the *design* theme. However, the concepts and their relationships remained stable despite the finicky thematic organization. The concept map is therefore the more reliable expression of the empirical data. Third, the empirical data do not directly support (or refute) the *time* dimension in the proposed framework. Fourth, the particular interview questions may have biased data collection toward some topics and away from others.

These limitations suggest several areas needing future research. More observational field research is needed to examine differences between developers’ explicit and implicit understanding of success; i.e., differences between what they say and how they act. The proposed framework may be tested through case studies, action research or (ideally longitudinal) questionnaire research. Additionally, similar interview-based studies with different participants and questions would strengthen the results. Moreover, studying more developers in different areas of software engineering (e.g. real-time critical systems, embedded systems, model-driven engineering, web development, video game development) and contrasting across sub disciplines may provide a richer account of success. The role of time, especially needs further investigation. Finally, while this paper makes

significant progress in illuminating the top-level dependent variable needed for high-level theorizing in SE, clear definitions and reliable measures of the dimensions of SES are still needed.

Notwithstanding the above limitations, our findings have two main implications. First, the proposed framework highlights the oversimplified, over-rationalized nature of common success measures, metrics and interpretations. Finishing on time or making hefty profits, for example, do not override other aspects of success including employee well-being and impacts on the public. Practitioners should take a broad view of software engineering success to avoid such misleading and ethically dubious simplifications. Managers, in particular, should keep in mind that individual designers rarely possess the kind of rich, multidimensional view of success proposed here; rather, project participants may become fixated on particular stakeholders (e.g., the client), targets and metrics to the detriment of the project.

Second, this paper was motivated by considering what it means for a new tool, language, technology, practice or pattern to be good. We suggested that at some level, many products of SE research are good if using them positively impacts SES. Empirical evaluation of SE research outputs is therefore hindered by a lack of good measures of SES, leading many to call for development of better measures (e.g. [28, 35, 42]). However, devising good measures of SES requires a sound understanding of the dimensions thereof. This paper therefore contributes to SE research by providing a foundation for developing better measures of software engineering success.

9. INTERVIEW GUIDE

9.1. Primary Questions

1. Would you summarize your current position (job)?
 - 1.1. How long have you been doing this job?
2. What sort of design work do you do? / Which parts of your job involve designing?
 - 2.1. What sorts of artifacts do you design or create?
 - 2.2. (If interviewee does not do design work) Who does the design work? How do you interact with him/her/them?
3. Are you involved in any project(s) involving design at the moment?
4. Have you ever had a project that was really successful?
 - 4.1. How could you tell?
 5. Have you ever had a project that was unsuccessful?
 - 5.1. How could you tell?
 6. Have you ever had a project that was ambiguous, as in, it was neither clearly successful nor clearly unsuccessful?
 - 6.1. What made it ambiguous?
 - 6.2. What sort of mixed signals did you get?
 7. Did any of the projects we've talked about have any formal success measure (quantitative or qualitative)
 - 7.1. Metrics? Key Performance Indicators? Rubrics? Performance Indicators?
 - 7.2. Are there any measures or indicators that you feel should be used, but aren't?
 8. What criteria do you use to determine success in your role?
 - 8.1. Does anyone else evaluate your work? (e.g., your boss)
 - 8.2. If so, are their criteria the same or different from yours? If they're different, how are they different?
 - 8.3. Do you evaluate the design work of others?
 - 8.4. If so, what dimensions or criteria do you use?
 9. Is your work driven by contracts with fixed schedules or budgets?
 - 9.1. How important is it to stick to the contracts?
 10. When you've designed an X (building, program, strategy, etc.), are there any characteristics that make a good X

regardless of the context? For example, regardless of what a building is for, it shouldn't fall down. Are there any such criteria for your work?

11. During a project, are there any signals you watch for to see if the project is going well or not?
 - 11.1. What about after a project? Are there any signals or feedback that come in shortly after completion/delivery?
 - 11.2. What about a while after completion/delivery? Does your impression of success/failure ever change because of something you learn years later?
12. Who does your design work affect?
 - 12.1. Whose feedback / opinions matter for you?
 - 12.2. Do the effects on anyone else matter for judging your success? If so, who?
13. Have you ever designed something that had to win in a competitive marketplace?
 - 13.1. How about something that would succeed or fail on its own merits, i.e., not in relationship to something else?
14. Is there anything else you'd like to add? Anything I should know?

9.2. Optional Questions

15. Who evaluates your design work?
 - 15.1. Who do you deliver your work to?
16. How do you know when you're done a (project / design task)?
 - 16.1. How do you measure progress?
 - 16.2. Are there any metrics or quotas or anything like that you have to meet?
 - 16.3. What kinds of signals did you get that let you know how you're doing?
 - 16.4. Do you get feedback from anyone regularly?
17. Hypothetically, if you were your (manager / client), how would you evaluate your work?
 - 17.1. What criteria would you apply?
 - 17.2. Is there anything you would measure quantitatively?
18. Do you follow any kind of formal design process, method or set of guidelines?
 - 18.1. Does this method prescribe specific success measures?
19. What's the first sign of trouble in your design work?
 - 19.1. What's the first sign that things are going well?
20. Do you create any intermediate artifacts (such as blueprints or prototypes)?
21. Has your understanding of success changed over time? how?
22. To what extent do you think your coworkers share your views on design and project success?

10. ACKNOWLEDGMENT

Thanks are due to all the participants who so generously gave up their time to speak with us, and to all the interviewers for their hard work.

11. REFERENCES

- [1] Al-Kilidar, H., Cox, K. and Kitchenham, B. 2005. The use and usefulness of the ISO/IEC 9126 quality standard. *Proceedings of the 2005 International Symposium on Empirical Software Engineering*. 126–132.
- [2] Alter, S. 2006. *The Work system method: Connecting people, processes, and IT for business results*. Work System Press.
- [3] Andrew J, S. 2011. The project workplace for organizational learning development. *International Journal of Project Management*. 29, 8, 986–993.
- [4] Atkinson, R. 1999. Project management: cost, time and quality, two best guesses and a phenomenon, its time to

- accept other success criteria. *International Journal of Project Management*. 17, 6, 337–342.
- [5] Atkinson, R., Crawford, L. and Ward, S. 2006. Fundamental uncertainties in projects and the scope of project management. *International Journal of Project Management*. 24, 8, 687–698.
- [6] Baccarini, D. 1999. The logical framework method for defining project success. *Project Management Journal*. 30, 4, 25–32.
- [7] Beck, K. 2005. *Extreme programming eXplained: Embrace change*. Addison Wesley.
- [8] Boehm, B.W. 1978. *Characteristics of Software Quality*. North-Holland.
- [9] Boehm, B.W., Brown, J.R. and Lipow, M. 1976. Quantitative evaluation of software quality. *Proceedings of the 2nd international conference on Software engineering*. IEEE. 592–605.
- [10] Briand, L.C., Wüst, J., Daly, J.W. and Victor Porter, D. 2000. Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of Systems and Software*. 51, 3, 245–273.
- [11] Brooks, F.P. 2010. *The Design of Design: Essays from a Computer Scientist*. Addison-Wesley Professional.
- [12] Bryde, D.J. and Robinson, L. 2005. Client versus contractor perspectives on project success criteria. *International Journal of Project Management*. 23, 8, 622–629.
- [13] Bush, D. and Finkelstein, A. 2003. Requirements stability assessment using scenarios. *Proceedings of the 11th International Requirements Engineering Conference*. IEEE. 23–32.
- [14] Cavano, J.P. and McCall, J.A. 1978. A framework for the measurement of software quality. *ACM SIGMETRICS Performance Evaluation Review*. 7, 3-4, 133–139.
- [15] Cicmil, S., Williams, T., Thomas, J. and Hodgson, D. 2006. Rethinking Project Management: Researching the actuality of projects. *International Journal of Project Management*. 24, 8, 675–686.
- [16] Cross, N. 1992. Research in Design Thinking. *Research in design thinking*. N. Cross, K. Dorst, and N. Roozenburg, eds. Delft University Press.
- [17] Crowston, K., Annabi, H. and Howison, J. 2003. Defining open source software project success. *Proceedings of the International Conference on Information Systems*. AIS.
- [18] de Wit, A. 1988. Measurement of project success. *International Journal of Project Management*. 6, 3, 164–170.
- [19] DeLone, W.H. and McLean, E.R. 1992. Information systems success: The quest for the dependent variable. *Information Management*. 3, 60–95.
- [20] DeLone, W.H. and McLean, E.R. 2003. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems*. 19, 4, 9–30.
- [21] Egorova, E., Torchiano, M., Morisio, M., Wohlin, C., Aurum, A. and Svensson, R.B. 2009. Stakeholders' Perception of Success: An Empirical Investigation. *Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications*. IEEE. 210–216.
- [22] Ekstedt, M. 2013. An Empirical Approach to a General Theory of Software (Engineering). *Proceedings of the 2nd Workshop on a General Theory of Software Engineering*. IEEE. 23–26.
- [23] Flyvbjerg, B. 2005. Design by deception: The Politics of major project approval. *Harvard Design Magazine*. 22, 50–59.
- [24] Freeman, R.E. 1984. *Strategic Management: A stakeholder approach*. Pitman.
- [25] Gallie, W.B. 1955. Essentially contested concepts. *Proceedings of the Aristotelian Society*. 56, 3, 167–198.
- [26] Johnson, P., Ralph, P., Goedicke, M., Ng, P.-W., Stol, K.-J., Smolander, K., Exman, I. and Perry, D.E. 2013. Report on the Second SEMAT Workshop on General Theory of Software Engineering (GTSE 2013). *SIGSOFT Software Engineering Notes*. 38, 5 (2013).
- [27] Kendra, K. and Taplin, L.J. 2004. Project success: A cultural framework. *Project Management Journal*. 35, 1, 30–45.
- [28] Kitchenham, B. 2010. What's up with software metrics?—A preliminary mapping study. *Journal of Systems and Software*. 83, 1, 37–51.
- [29] Lim, C. and Mohamed, M.Z. 1999. Criteria of project success: an exploratory re-examination. *International Journal of Project Management*. 17, 4, 243–248.
- [30] Linberg, K.R. 1999. Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*. 49, 2, 177–192.
- [31] Miles, S. 2012. Stakeholder: Essentially Contested or Just Confused? *Journal of Business Ethics*. 108, 3, 285–298.
- [32] Mohanani, R., Ralph, P. and Shreeve, B. 2014. Requirements Fixation. *Proceedings of the 2014 International Conference on Software Engineering*. ACM.
- [33] Müller, R. and Turner, R. 2007. The Influence of Project Managers on Project Success Criteria and Project Success by Type of Project. *European Management Journal*. 25, 4, 298–309.
- [34] Norman, D. 2005. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Books.
- [35] Petter, S., DeLone, W. and McLean, E. 2008. Measuring information systems success: models, dimensions, measures, and interrelationships. *European Journal of Information Systems*. 17, 3, 236–263.
- [36] Procaccino, J.D., Verner, J.M., Shelfer, K.M. and Gefen, D. 2005. What do software practitioners really think about project success: an exploratory study. *Journal of Systems and Software*. 78, 2, 194–203.
- [37] Ralph, P. 2010. Comparing Two Software Design Process Theories. *Global Perspectives on Design Science Research: Proceedings of the 5th International Conference, DESRIST 2010*. R. Winter, J.L. Zhao, and S. Aier, eds. Springer. 139–153.
- [38] Ralph, P. 2011. Introducing an Empirical Model of Design. *Proceedings of The 6th Mediterranean Conference on Information Systems*. AIS.
- [39] Ralph, P. 2013. The Illusion of Requirements in Software Development. *Requirements Engineering*. 18, 3, 293–296.
- [40] Ralph, P. 2013. The Sensemaking-Coevolution-Implementation Theory of Software Design. *arXiv: 1302.4061 [cs.SE]*.
- [41] Ralph, P. and Wand, Y. 2009. A Proposal for a Formal Definition of the Design Concept. *Design Requirements*

- Engineering: A Ten-Year Perspective*. K. Lyytinen, P. Loucopoulos, J. Mylopoulos, and W. Robinson, eds. Springer-Verlag. 103–136.
- [42] Ralph, P., Johnson, P. and Jordan, H. 2013. Report on the First SEMAT Workshop on a General Theory of Software Engineering (GTSE 2012). *SIGSOFT Software Engineering Notes*. 38, 2, 26–28.
- [43] Shenhar, A.J. and Dvir, D. 1997. Mapping the dimensions of project success. *Project Management Journal*. 28, 2, 5–13.
- [44] Shenhar, A.J., Dvir, D., Levy, O. and Maltz, A.C. 2001. Project Success: A Multidimensional Strategic Concept. *Long Range Planning*. 34, 6, 699–725.
- [45] Smith, A.E. and Humphreys, M.S. 2006. Evaluation of unsupervised semantic mapping of natural language with Leximancer concept mapping. *Behavior Research Methods*. 38, 2, 262–279.
- [46] Standish Group 2009. *CHAOS Summary 2009*. The Standish Group International.
- [47] Stanovich, K. 2009. *What Intelligence Tests Miss: The Psychology of Rational Thought*. Yale University Press.
- [48] Stanovich, K.E. and West, R.F. 2007. Natural myside bias is independent of cognitive ability. *Thinking & Reasoning*. 13, 3, 225–247.
- [49] Suchman, L. 1987. *Plans and Situated Actions: The problem of human-machine communication*. Cambridge University Press.
- [50] Venkatesh, V., Morris, M.G., Davis, G.B. and Davis, F.D. 2003. User acceptance of information technology: Toward a unified view. *MIS Quarterly*. 27, 3, 425–478.
- [51] Venturi, G. and Troost, J. 2005. An agile, user-centric approach to combat system concept design. *10th International Command and Control Research and Technology Symposium*.
- [52] Wateridge, J. 1998. How can IS/IT projects be measured for success? *International Journal of Project Management*. 16, 1, 59–63.
- [53] Winter, M., Andersen, E.S., Elvin, R. and Levene, R. 2006. Focusing on business projects as an area for future research: An exploratory discussion of four different perspectives. *International Journal of Project Management*. 24, 8, 699–709.