T U M

I N S T I T U T F Ü R I N F O R M A T I K

A Structured Review of Productivity Factors in Software Development

Stefan Wagner, Melanie Ruhe

TUM-I0832
September 08

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-09-I0832-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

A Structured Review of Productivity Factors in Software Development

Stefan Wagner
Institut für Informatik
Technische Universität München
Garching b. München, Germany
wagnerst@in.tum.de

Melanie Ruhe
Siemens AG
Corporate Technology, CT SE 3
Munich, Germany
melanie.ruhe@siemens.com

Abstract

Analysing and improving productivity has been one of the main goals of software engineering research since its beginnings. A plethora of studies have been conducted on various factors that resulted in several models for analysis and prediction of productivity. However, productivity is still an issue in current software development and not all factors and their relationships are known. This paper reviews the large body of available literature in order to distill a list of the main factors influencing productivity investigated so far. The measure for importance here is the number of articles a factor is used in. Special consideration is given to *soft* or human-related factors in software engineering that are often not analysed with equal detail as more technical factors. The resulting list can be used to guide further analysis and as basis for building productivity models.

1 Introduction

Productivity in software development has been an important research area for several decades. It is the key for a successful software company to control and improve its productivity. However, in contrast to traditional industrial work, it is hard to measure for software development. There are several terms that are used more or less synonymously such as performance or efficiency [15]. There are also various definitions of which output divided by input is the most general one.

In software development lines-of-code (LOC) and function points (FP) are traditional measures for productivity, i.e., the amount of LOC or FP produced per hour by a developer. Based on this, there is a large amount of studies on various aspects of productivity. The two mentioned measures and several more dimensions have been analysed and detailed. Models have been built that should explain, analyse and predict productivity. Finally, several studies analyse the factors that influence productivity in a software project.

1.1 Problem

Frese and Brodbeck [23] claim that the scientific discussion about the work situation in software development and about productivity factors in such projects is done based on an insufficient empirical basis. According to them, it is dominated by shallow surveys and qualitative experience reports.

Moreover, the software engineering literature in that area often has a strong emphasis on mainly technical factors such as the software size or the product complexity. However, Brodbeck [12] shows that more than a third of the time a typical software developer is not concerned with technical work. Meetings and talks constitute 21.1%, presentations and project organisation 9.6%, and independent qualification 6% of the work time. Hence, these efforts are significant.

1.2 Contribution

We provide a structured review of software engineering, management and organisational psychology literature on productivity factors in software development. A list of these factors is distilled from the literature in order to aid model building, productivity improvement and further research.

The importance of the factors and thereby their inclusion in the list is based on their mentioning in the analysed literature. Hence, it is secured that several authors used and/or analysed the factor. Furthermore, we put specific care to the equal consideration of technical and soft factors in order to represent their significance.

1.3 Organisation

We describe the used approach in the literature review in section 2. The studies that are finally considered in the review in section 3. The resulting factors are summarised and discussed in section 4. We close with related work (section 5) and final conclusions (section 6).

2 Review Approach

For the structured review of the productivity literature, we use a combined approach of automated and manual search. Our aim is to include literature from the areas software engineering, management and organisational psychology as these are the main sources of relevant literature. For this we used a query on four portals for scientific literature that contains the typically used terms in papers about what we call *productivity factors*. For the automated search we used the following expression:

```
software AND (productivity OR "development
efficiency" OR "development effectiveness"
OR "development performance")
```

It resulted in the following numbers of results:

- ACM's The Guide: 10,017
- IEEE Xplore: 1,408
- ScienceDirect: 508
- Google Scholar: 680,000

These large numbers show on the one hand the significance of the topic in research but on the other hand prevents the manual analysis of all these papers. We inspected the first 100 results of each portal whether they are suitable for inclusion in our study. In this inspection we omitted very specific analyses of single, detailed factors because of brevity reasons. We also excluded studies that only showed that factors have no influence on productivity. Although these studies are of interest in general, they do not help in building a list of factors that do influence productivity.

In addition to the papers retrieved using that query, we also collected papers manually in a number of important journals in software engineering (e.g. IEEE Transactions on Software Engineering), in management (e.g. Management Science) and organisational psychology (e.g. Journal of Occupational and Organizational Psychology). From these and by following references from the already found papers, the complete body of papers that build the basis of this study was collected. Moreover, the well-known books by Boehm [10] and Jones [35] on software productivity were included as a baseline.

We derived from this body of papers factors about the product, process and people and unified synonymous terms as far as possible. Then the extracted factors were ranked by appearances in literature. We mainly aimed at finding different authors that used the factors. Based on this, the final list was compiled.

3 Considered Studies

We first briefly describe each of the studies that are considered in the final list of productivity factors. We decided to organise the papers in the order of their

publication which has the additional benefit that the developments over time become visible.

3.1 1970–1979

Walston and Felix [62] analysed in 1977 in one of the first larger studies factors that correlate significantly with programming productivity (measured in effort per SLOC). Several of the later publications use the same or a variant of these factors. A number of the described factors obviously decreased in importance over the decades. For example, *chief programmer team usage* is not a common practice today. Also with the more and more standardised hardware, *previous experience with operational computer* does not seem to be a problem anymore. Nevertheless, the majority of factors, such as *user participation*, *overall constraints on program design* or *previous experience with programming language* are still valid.

Albrecht then proposed his famous *function points* [4]. In this study, he analysed factors like the used programming language and the project size.

3.2 1980–1989

Brooks [14] uses factors from Walston and Felix [62] as basis in his study at IBM. He found especially the effects of program complexity and structured programming to be important.

Vosburgh et al. [61] describe one of the early investigations of a larger amount of projects at ITT. They especially analysed the importance of productivity factors in these projects. In most cases, they found similar results as Walston and Felix [62]. Only for client participation and experience they identified the opposite influence on productivity.

Hanson and Rosinski [28] describe a study of the perceived productivity improvement by programmers using different software tools. Tools like an *interactive debugger* and a *screen editor* ranked highly.

Jones started with [34] a series of books about programming productivity. He was one of first that analysed various productivity factors over various domains and could provide industry averages. He focused his measures strongly on LOC and FP.

DeMarco and Lister [19] then aimed in a completely different direction from the LOC- and FP-centred research. They point out that "The major problems of our work are not so much *technological* as *sociological* in nature." They consider turnover as one of the central factors influencing productivity. "People under time pressure don't work better; they just work faster." They also mention the importance of a proper work place with windows, natural light, quietness, etc. They substantiate this by showing that a noisy workplace with a high probability leads to more defects. Moreover, there are huge differences in the performances of individuals. The used language, years of experience, number of defects and salary do not have an significant effect on productivity in their opinion.

They further claim that "Quality, far beyond that required by the end user, is a means to higher productivity." They then discuss work interruption as important issue and introduce the *Environmental Factor* or *E-Factor* as measure for this:

$$\text{E-Factor} = \frac{\text{Uninterrupted Hours}}{\text{Body-Present Hours}} \tag{1}$$

Typical E-Factors range between 0.10 and 0.38.

They discuss that methodologies or development processes have advantages of converge of methods. For example, "Maintenance personnel will be able to relate more quickly to new products, developers will be able to move onto new projects and get up to speed more quickly". However, they state that these benefits can also be generated by training, tools, and peer review. They claim that by forcing methodology on people, it guarantees

- a morass of paperwork,
- a paucity of methods,
- an absence of responsibility, and
- a general loss of motivation.

DeMarco and Lister further introduce the concept of a "jelled team" which is a team that has aligned goals and strong interactions. They give five indicators for that:

- low turnover
- strong sense of identity
- sense of eliteness
- joint ownership of the product
- obvious enjoyment

Finally, they list six factors that they called "teamicide", i.e., measures that are the main obstacles in building (or growing) teams that partially repeat earlier mentioned factors:

- defensive management
- bureaucracy
- physical separation
- fragmentation of people's time
- quality reduction of the product
- phony deadlines

- clique control

In summary, DeMarco and Lister provided in [19] the first and still most comprehensive work on the soft factors influencing productivity in software development.

Grady and Caswell [26] propose a simple model for productivity that contains *value* and *cost* as main factors on productivity. These factors have in turn a set of 16 further influencing factors. Mainly technical factors are considered.

Guzzo reviews in [27] the psychological research on productivity. Unfortunately, he concentrates on productivity in production industries. However, several of the factors should be applicable to software development as well. He argues that "Several reviews exist of the effects of realistic job previews on subsequent productivity." Especially, it is a way to reduce turnover. In terms of quantity of the output, "it appears that selection practices can contribute significantly to increased productivity". In a series of studies training was found to be the most powerful means of increasing productivity. He also discusses that feedback on their work increases the productivity of employees. *Goal Setting* is a rather general managerial technique that showed to improve productivity. Financial incentives have a positive effect but with a high degree of variance. A strong participation of employees in management raises employee satisfaction which in turn should be related to productivity. The degree to which the employees can control their own processes have an influence as well.

Cerveny and Joseph [16] looked at several projects implementing the same requirements but with different development methods. Surprisingly, they found that the development effort was twice as high in projects that used *structured design and programming* in comparison with projects using non-structured approaches. Yet, they did not consider later test and defect costs which could be significantly lower.

The most famous model that involves productivity is COCOMO by Boehm [10]. It is a cost-estimation model in which the productivity of the developers obviously plays a decisive role. These factors have been derived empirically from a large project database. The factors are discussed in more detail in section 3.4 with COCOMO II.

Abdel-Hamid [1] analysed specifically the effects of changing the staffing during a project using simulation. He found that it can have significant effects on the productivity in the projects.

3.3 1990–1999

Jones mainly updated his earlier findings in [35]. Again, on the basis of function point measurements various productivity analyses are described and a large amount of data is provided.

Scudder and Kucic [53] propose a multi-dimensional measure for information system productivity. The single parts of this measure are similar to productivity factors used in other studies such as *staff satisfaction* or quality factors such as *system availability*.

Banker, Datar and Kemerer [6] analysed banking applications w.r.t. their maintenance productivity and corresponding influencing factors. The found mostly quality, staff ability, response time and application experience to be significantly important.

Simmons [54] also studied the effects of communication in and between teams and called it a *dominator*. This means that effective communication can have a ten-fold impact on productivity.

Rasch studies in [50] the effect of factors such as team member rotation, role ambiguity and role conflict on job satisfaction and actually quantifies them based on a survey.

Ford and McLauglin [22] interviewed managers about the success of project teams. The size of the department was the only significant difference between teams that met the expectations of the managers and those who did not.

Tsuda et al. [57] measured productivity by *program generation rate*, *fault density* and *learning curve of inexperienced programmers*. They showed by this that a CASE tool improved productivity.

The book [41] edited by Luedecke points out the importance of rooms and architecture on productivity.

Lakhanpal [38] concentrated on characteristics of groups and their influence on productivity. The cohesiveness and capability had the strongest influence in 31 development groups, experience had the weakest influence.

Finnie, Wittig and Petokov [21] describe an analysis of productivity factors. They mainly use the factors described in COCOMO [10] and added *user computer literacy*, *management commitment*, *third-party involvement*, *degree of decentralisation*, *user involvement* and *team size*.

Agrell and Gustafson [3] test the Team Climate Inventory (TCI) in Swedish work groups to measure the degree to which the team climate supports innovation. We assume a culture of innovation is helpful for productivity in software and systems development. They found that the four main factors of the TCI are indeed reliable predictors: (1) participation, (2) support for innovation, (3) vision and group goals, and (4) task orientation and climate for excellence.

Brodbeck describes in [11] that in a survey, the projects with a higher communication effort also were more successful. Even the intensity of internal communication is positively correlated with project success. This is in contrast to common software engineering belief that high communication effort hampers productivity.

The importance of interruptions is also investigated by Sonnentag [55]. She found that interruptions are considered as the number one stress factor by software developers.

Wohlin and Ahlgren describe factors and their impact on time to market in [63]. They use 10 different factors in their study, mostly factors that are covered by the publications discussed so far. They also include product complexity, methods and tools and requirements stability that could be considered technical factors. Obviously, it is possible to argue for them to be sociological as well. The factor "priority" is not clearly described and is hence excluded.

Cole [18] says that schedule was most frequently the prime runaway problem. He gives the further following causes (in order of importance): project objectives not fully specified, bad planning and estimating, technology new to the organisation, inadequate or no management methodology, insufficient senior staff on the team, poor performance by suppliers (hardware and/or software).

Maxwell, Van Wassenhove and Dutta [43] analysed the ESA productivity database in order to find the relevant productivity factors in that area. They found that "productivity decreases with increasing storage constraints, timing constraints, reliability requirements, team size and project duration." The also state that productivity tends to increase with system size.

Blackburn, Scudder, and Van Wassenhove [8] studied the factors and methods that improved productivity in Western European companies. They found *project duration* and *team size* to be significant.

McLean, Smits, and Tanner [45] analysed the importance of salary. They conclude that it is important in the early career but other factors tend to crowd out salary in importance later.

Gorla and Ramakrishnan [25] investigated the effects of the structure of a software on maintenance productivity. This is one of the few factors developers have under their own control. They found that there is a significant correlation.

Chatzoglou and Macaulay [17] interviewed participants of over a hundred software projects about several factors and their influence on productivity. They found that experience, knowledge and persistence of the team members is considered important. Also the motivation of the users and their communication with the rest of the team plays a role. Finally, the available resources, tools and techniques used and the management style are important factors.

Glass summarised in [24] his findings on project "runaways". He states that common causes for such failing projects are that they are huge, that there are usually a multiplicity of causes and that they were aimed to be "breakthroughs" in comparison to older systems. However, he also suggests that technology is increasingly often the cause for project failure.

Hill et al. [31] investigated the influence of virtual offices on aspects of work. Most interestingly in the productivity context is that the perception that "teamwork has been diminished".

Port and McArthur [47] analysed the introduction of object-oriented methods at Hughes Space and Communications. They found that an object-oriented development approach coupled with object-oriented implementation improves overall project productivity.

Linberg [40] discusses factors such as job satisfaction and the temperament of the developers. He analysed project failures using structured interviews, reviews, and surveys. He also derives relationships between factors.

3.4 2000–2007

The most thorough work in the area of productivity and its influencing factors is COCOMO II by Boehm et al. [9]. They have a long experience in that area [10]

and derive their factors from a large empirical body. The technical factors they identified are:

- Precedentedness (how similar are the projects)
- Development flexibility (how strong are the constraints on the system)
- Architecture / risk resolution (how are the risks mitigated by architecture)
- Process maturity
- Required software reliability
- Database size
- Product complexity
- Developed for reusability
- Documentation match to life-cycle needs
- Execution time constraints
- Main storage constraint
- Platform volatility
- Use of software tools

Boehm et al. also analysed various soft factors and found that those factors combined are more important than all the others. The factors are:

- Team cohesion (how well are the stakeholders working together)
- Analyst capability
- Programmer capability
- Personnel continuity (turnover)
- Applications experience
- Platform experience
- Language and tool experience
- Multisite development
- Required development schedule

Jones states in [36] that software projects are influenced by about 250 factors. Individual projects "are usually affected by ten to 20 major issues." Of course, he also investigated a series of soft factors. He lists and discusses several factors based on case studies partially with quantitative results. 36 of these factors are considered the major factors. In comparison to other studies, he adds explicitly the support for modern telecommunication facilities such as video conferencing.

Von Mayrhauser et al. [60] describe a method for assessing efficiency and include soft factors. A case study from NASA is also described. They use similar factors as in other papers discussed.

Alper, Tjosvold and Law [5] show in a study that cooperative conflict management is significant for team effectiveness.

Maxwell and Forselius argue in [44] that the influencing factors on productivity depend on the business domain the software is produced for. For example, in the insurance domain *requirements volatility*, *software's logical complexity* and *tools use* are significant while in the public administration domain *number of inquiries* and *customer participation* are of importance.

Wohlin and Andrews analyse factors for project success [65]. They especially analyse subjective measures in [64]. Many of the used factors are the same as in [63]. The assessment and priorisation of such factors is described in [66].

Haslam, Eggins and Reynolds [29] describe a detailed model of personal and team identity and stress their importance for productivity.

MacCormack et al. [42] studied projects of HP w.r.t. productivity (measured in new LOC) and quality (measured in defect rates). They found a significant correlation between the completeness of the design specification and productivity. Also early prototyping increases productivity.

Turcotte and Rennison [58] show empirically based on a survey of Canadian companies that using computers, education and training, and profit-sharing compensation scheme are associated with higher productivity.

Kitchenham and Mendes [37] found that reuse is taking place has a significant effect on productivity. The amount of reuse is not that important. They also suggest that the productivity is not significantly different in different countries.

Roth and Moser [51] discuss various factors including *goal orientation*, *task accomplishment*, *cohesion* and *responsibilities*. These factors cover largely areas similar to the ones found in the other sources.

Verner and Evanco [59] conducted a survey about success factors. They mainly analysed factors w.r.t. the project manager, requirements, and post-mortem reviews.

Premraj, Shepperd, Kitchenham, and Forselius [48] analysed a large amount of data of Finnish companies. They found that there is no significant difference in the productivity of new development and maintenance projects. The most significant factors are *company*, *business sector*, *year*, and *hardware*. Only the latter is suitable for our factors because the others cannot be influenced in a company.

Berntsson-Svensson and Aurum [7] analysed in a survey factors influencing project and product success. They found that different industries define success

in different terms. However, the identified influencing factors are similar to other studies: well defined project scope, complete and accurate requirements, good schedule estimations, customer/user involvement, and adding extra personnel.

Agrawal and Chari [2] studied CMM level 5 projects among others for their effort and found that from the influencing factors they used, e.g., product complexity, schedule pressure, or personnel capability, only the software size had a significant influence.

Jiang et al. [33] analysed the large ISBSG data repository with 4106 projects in order to find factors that significantly affect productivity. The identified factors are:

- Average team size
- Language (2GL, 3GL, 4GL, or application generator)
- Platform (mainframe, mid range, mulit-platform, or PC)
- Object-oriented analysis and design
- Event modelling
- Regression testing
- Business area modelling

Lehigh [39] describes how to use Six Sigma goals to improve productivity in software development. He concentrates largely on various aspects of defect detection, i.e., its effectiveness, efficiency, and predictability.

Mohagheghi and Conradi [46] analysed especially the connection between software reuse and productivity among other factors. They show that there can be a strong positive influence.

Edmans investigates in [20] the relationship between job satisfaction and stock market value of companies in the U.S. He finds a positive correlation and shows that those kinds of "intangibles" are not fully valued on the stock market. Even though the study cannot prove a causal relationship between job satisfaction and market value, it demonstrates the importance of those soft factors. The soft factors used are taken from a survey by A Great Place to Work that uses five dimensions:

- Credibility
- Respect
- Fairness
- Pride
- Camaraderie

Spiegl reports in [56] on a survey on project management issues in software development conducted with project managers. In terms of soft factors he found that support of the top-management, business culture, promotions, team building, relationship management and communication, freedom and responsibility, and motivation and appreciation are important.

4 Results

As mentioned above, we roughly divide the productivity factors into *technical* and *soft* factors. We see soft factors as all non-technical factors influencing productivity. These factors mainly stem from the team and its work environment. Obviously, the borderline between these two groups is sometimes blurry and is only intended to aid easier comprehension.

The technical factors are summarised in table 1. In this group we structured the factors in three categories. The *product* category contains all factors that have a direct relation to the product, i.e., the system itself. The category *process* is comprised of the technical aspects of the process. Finally, the category *development environment* contains factors about the tools the developer uses in the project. We also added possible ways of measurement of these factors together with the scale type the measures will be in.

Table 1: The derived technical factors

Factor	Description	Source	Measurement
Precedentedness	How similar are the projects?	[9, 24]	Interview (ordinal)
Required Software Reliability	The level of reliability needed.	[9, 21, 43]	MTTF (ratio)
Database Size	How large is the data compared to the code?	[9, 62]	Bytes of data/LOC (ratio)
Product Complexity	The complexity of the function and structure of the software.	[9, 14, 21, 25, 44, 61]	Interview (ordinal)
Developed for Reusability	To what extent the components should be reusable.	[9, 21, 37]	Interview (ordinal)
Execution Time Constraints	How much of the available execution time is consumed.	[9, 14, 21, 43, 44, 61, 62]	used time/available time (ratio)
Main Storage Constraint	How much of the available storage is consumed.	[9, 14, 61]	used storage/available storage (ratio)
Software Size	The amount of code of the system.	[2, 4, 14, 43]	LOC (interval)
Product Quality	The quality of the product influences motivation and hence productivity.	[6, 19]	Quality metrics (ordinal – interval)

Table 1 – continued

Factor	Description	Source	Measurement
User Interface	Degree of complexity of the user interface.	[14, 44, 62]	Interview (ordinal)
Development Flexibility	How strong are the constraints on the system?	[9, 62]	Interview (ordinal)
Reuse	The extent of reuse.	[37, 46]	Interview (ordinal)
	Process		
Architecture Risk Resolution	How are the risks mitigated by architecture?	[9]	Interview (ordinal)
Process Maturity	The well-definedness of the process.	[9]	CMMI appraisal (ordinal)
Platform Volatility	Time span between major changes.	[9, 43, 48]	Time (interval)
Early Prototyping	Early in the process prototypes are built	[42]	Interview (nominal)
Completeness of Design	The amount of the design that is completed when starting coding	[14, 42]	Design size measure (ratio)
Effective and Efficient V&V	The degree to which defects are found and the needed effort.	[39]	found defects/total defects, person-hours/defect (ratio)
Project Duration	Length of the project.	[8, 43]	Time (interval)
Hardware Concurrent Development	Is the hardware developed concurrently?	[14, 61, 62]	Interview (ordinal)
	Development Environment		
Use of Software Tools	The degree of tool use.	[4, 9, 17, 21, 28, 44, 57]	Interview (ordinal)
Programming Language	The level of the used programming language.	[4, 33, 44]	2GL, 3GL, ... (ordinal)
Use of Modern Development Practices	Are modern methods applied?	[4, 14, 16, 33, 47, 61, 62]	Appraisal (nominal)
Documentation match to life-cycle needs	How well the documentation fits to the needs.	[9, 62]	Interview (ordinal)

The soft factors are summarised in table 2. Overlapping factors are combined as far as possible. Possible ways of measuring the factors are given. We

employed a simple, non-unique categorisation to aid a quick comprehension. *Corporate Culture* contains the factors that are on a more company-wide level whereas *Team Culture* denotes similar factors on a team level. In *Capabilities and Experiences* are factors summarised that are related to individuals. *Environment* stands for properties of the working environment. Finally, project-specific factors are in the *Project* category.

Table 2: The derived soft factors

Factor	Description	Source	Measurement
	Corporate Culture		
Credibility	Open communication and competent organisation.	[20, 36, 53, 56]	Interview (ordinal)
Respect	Opportunities and responsibilities.	[3, 20, 27, 51, 53, 56]	Interview (ordinal)
Fairness	Fairness in compensation and diversity.	[20, 27, 45, 56, 58]	Interview (ordinal)
	Team Culture		
Camaraderie	Social and friendly atmosphere in the team.	[20]	Interview (ordinal)
Team Identity	The common identity of the team members.	[19, 29]	Interview (ordinal)
Sense of Eliteness	The feeling in the team that they are "superior" and to take pride in the product, team, company etc.	[3, 19, 20]	Interview (ordinal)
Clear Goals	How clearly defined are the group goals?	[3, 17, 27]	Questionnaire (Likert)
Turnover	The amount of change in the personnel.	[9, 17, 19, 27, 53, 63, 64]	% / year (ratio)
Team Cohesion	The cooperativeness of the stakeholders.	[5, 9, 17, 38, 50, 51, 53, 61, 62]	Interview (ordinal)
Communication	The degree and efficiency of which information flows in the team.	[11, 17, 54, 63]	Interview (ordinal)
Support for Innovation	To what degree assistance for new ideas is available.	[3]	Questionnaire (Likert)
	Capabilities and Experiences		
Developer Temperaments	The mix of different temperaments on the team.	[40]	Interview (Keirsey temperament)
Analyst Capability	The skills of the system analyst.	[9, 14, 21, 36, 58, 62–64]	Interview (ordinal)

Table 2 – continued

Factor	Description	Source	Measurement
Programmer Capability	The skills of the programmer	[6, 9, 21, 36, 38, 53, 58, 61–64]	Interview (ordinal)
Applications Experience	The familiarity with the application domain.	[6, 9, 14, 21, 34, 53, 62]	Interview (ordinal)
Platform Experience	The familiarity with the hard- and software platform.	[9, 14, 18, 21, 34, 62, 64]	Interview (ordinal)
Language and Tool Experience	The familiarity with the programming language and tools.	[9, 14, 17, 21, 34, 44, 62, 64]	Interview (ordinal)
Manager Capability	The control of the manager over the project.	[17, 18, 36, 38, 56, 62, 63]	Interview (ordinal)
Manager Application Experience	The familiarity of the manager with the application.	[62, 64]	Interview (ordinal)
	Environment		
Proper Workplace	The suitability of the workplace to do creative work, e.g., windows, natural light, size of room and desk	[19, 36, 41]	Interview (ordinal)
E-Factor	This environmental factor describes the ratio of uninterrupted hours and body-present hours.	[19, 36]	Time documentation (ratio)
Time Fragmentation	The amount of necessary "context switches" of an employee.	[19]	No. of projects per person (interval)
Physical Separation	The team members are distributed over the building or multiple sites.	[9, 19, 21, 63]	Distance between team members (interval)
Telecommunication Facilities	Support for work at home, virtual teams, video conferencing with clients.	[31, 36]	Interview (ordinal)
	Project		
Schedule	The appropriateness of the schedule for the development task.	[7, 9, 18, 19, 63]	Interview (ordinal)
Requirements Stability	The number of requirements changes.	[7, 21, 44, 61–63]	No. of changes per month (ratio)
Average Team Size	Number of people in the team.	[1, 8, 21, 22, 24, 33, 43, 53, 61, 62]	Average number of team members (ratio)

In general, what is surprising in the studies is that communication effort is positive for productivity. It is often discussed that communication should be reduced to decrease "unnecessary" work. However, it seems the problem is only that with increasing people the communication effort increases strongly. Yet, a high fraction of effort on communication seems like a good investment.

Then there is some agreement in the few studies that analysed this factors that the business domain plays a role. Either the domain itself has an influence on productivity or at least it determines which of the other factors have the strongest influence. This contradicts general and generic productivity models but suggests that individual models are needed.

It is also notable that although experience is often brought up and is in interviews considered important, in empirical studies it is rather insignificant. By far more interesting is the capability of the developers. Hence, this suggests that only being in a profession for a long time does not make one productive but working on high capabilities.

5 Related Work

An early review of the state of the art in software development productivity was done by Jeffery and Lawrence [32]. They concentrated on the conflicting results w.r.t. some factors such as experience or size that in some studies were found to have a positive in others a negative effect. We do not consider that in our paper but only analyse the relevance of a factor in general.

Maxwell, Van Wassenhove and Dutta [43] relate their research on productivity factors in military software projects to earlier studies in other areas and the factors found there. however, this work is already 12 years old and a large number of studies have contributed to the knowledge about productivity factors since then.

Ramírez and Nembhard [49] analysed the more general category of *knowledge worker* productivity. Software developers are in their work part of these knowledge workers (KW) as opposed to manual workers. They state that "it seems to be of common agreement that to date there are no effective and practical methods to measure KW productivity." Hence, they concentrate on a review of the dimensions used in the literature whereas our review considers the factors influencing productivity.

6 Conclusions

The productivity of the development team is decisive for successful software projects. Hatton [30] shows that there are large difference, especially in the abilities of the developers. "[. . .] in most experiments, analysts regularly record variations of a factor of 10 or more in the individuals' performance." This illustrates the large potential for improvements in development projects.

However, controlling productivity is only possible if the influencing factors

are known. "You cannot control what you cannot measure." [19] Hence, a clear list of influences on productivity in software development is needed in order to organise corresponding analysis and control activities. Existing productivity models and methods already make use of lists of productivity factors.

Yet, there is a large body of literature on productivity and productivity factors acummulated over the last decades. This paper provides a structured review of this literature and a derived list of important factors based on their use in the studies. Soft and technical factors are investigated in equal detail and a list of factors is provided for each.

This list can now be used for modelling productivity and for productivity improvement methods. For example, the ProdFLOW^{TM}[1] method described in [52] uses interview techniques for determining the most influential factors in productivity for a specific organisation. These interviews can be supported by the comprehensive knowledge about existing factors from the compiled lists.

For further research, we need to add further detail to the lists of factors by determining whether the factors influence productivity positively or negatively which is important for productivity models. Furthermore, there are influences between factors that can also have significant effects that need to be considered in this list and corresponding models.

References

[1] T. K. Abdel-Hamid. The dynamics of software project staffing: A system dynamics based simulation approach. *IEEE Transactions on Software Engineering*, 15(2):109–119, 1989.

[2] M. Agrawal and K. Chari. Software effort, quality, and cycle time: A study of CMM level 5 projects. *IEEE Transactions on Engineering Management*, 33(3):145–156, 2007.

[3] A. Agrell and R. Gustafson. The Team Climate Inventory (TCI) and group innovation: A psychometric test on a Swedish sample of work groups. *Journal of Occupational and Organizational Psychology*, 67:143–151, 1994.

[4] A. J. Albrecht. Measuring application development productivity. In *Proc. Joint SHARE/GUIDE/IBM Application Development Symposium*, pages 83–92, 1979.

[5] S. Alper, D. Tjosvold, and K. S. Law. Conflict management, efficacy, and performance in organizational teams. *Personnel Psychology*, 53:625–642, 2000.

[6] R. D. Banker, S. M. Datar, and C. F. Kemerer. A model to evaluate variables impacting the productivity of software maintenance projects. *Management Science*, 37(1):1–18, 1991.

[1] ProdFLOW is a registred trademark of the Siemens AG.

[7] R. Berntsson-Svensson and A. Aurum. Successful software project and products: An empirical investigation. In *Proc. 2006 ACM/IEEE International Symposium on Empirical Software Engineering (ISESE '06)*, pages 144–153. ACM Press, 2006.

[8] J. D. Blackburn, G. D. Scudder, and L. N. Van Wassenhove. Improving speed and productivity of software development: A global survey of software developers. *IEEE Transactions on Software Engineering*, 22(12), 1996.

[9] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. Reifer, and B. Steece. *Software Cost Estimation with COCOMO II*. Prentice-Hall, 2000.

[10] B. W. Boehm and P. N. Papaccio. Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, 14(10):1462–1477, 1988.

[11] F. C. Brodbeck. Intensive Kommunikation lohnt sich für SE-Projekte. In Brodbeck and Frese [13], pages 51–67.

[12] F. C. Brodbeck. Software-Entwicklung: Ein Tätigkeitsspektrum mit vielfältigen Kommunikations- und Lernanforderungen. In Brodbeck and Frese [13], pages 13–34.

[13] F. C. Brodbeck and M. Frese, editors. *Produktivität und Qualität in Software-Projekten*. R. Oldenbourg Verlag, 1994.

[14] W. D. Brooks. Software technology payoff: Some statistical evidence. *The Journal of Systems and Software*, 2(1):3–9, 1981.

[15] M. Broy, F. Deissenboeck, and S. Wagner. The chimera of software productivity. Manuscript for publication in Communications of the ACM, 2008.

[16] R. P. Cerveny and D. A. Joseph. A study of the effects of three commonly used software engineering strategies on software enhancement productivity. *Information & Management*, 14(5):243–251, 1988.

[17] P. D. Chatzoglou and L. A. Macaulay. The importance of human factors in planning the requirements capture stage of a project. *International Journal of Project Management*, 15(1):39–53, 1997.

[18] A. Cole. Runaway projects – cause and effects. *Software World*, 26(3), 1995.

[19] T. DeMarco and T. Lister. *Peopleware. Productive Projects and Teams*. Dorset House Publishing, 1987.

[20] A. Edmans. Does the stock market fully value intangibles? employee satisfaction and equity prices. `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=985735`, 2007. Accessed 2007-09-11.

[21] G. R. Finnie, G. E. Wittig, and D. I. Petkov. Prioritizing software development productivity factors using the analytic hierarchy process. *Journal of Systems and Software*, 22(2):129–139, 1993.

[22] R. C. Ford and F. S. McLaughlin. Sucessful project teams: a study of MIS managers. *IEEE Transactions on Engineering Management*, 39(4):312–317, 1992.

[23] M. Frese and F. C. Brodbeck. Einleitung. In Brodbeck and Frese [13], pages 7–10.

[24] R. L. Glass. Software runaways–some surprising findings. *Journal of Systems and Software*, 41(2):75–77, 1998.

[25] N. Gorla and R. Ramakrishnan. Effect of software structure attributes on software development productivity. *Journal of Systems and Software*, 36(2):191–199, 1997.

[26] R. B. Grady and D. L. Caswell. *Software Metrics: Establishing a Company-Wide Program*. Prentice-Hall, 1987.

[27] R. A. Guzzo. Productivity research: Reviewing psychological and economic perspectives. pages 63–81. Jossey-Bass Publishers, 1988.

[28] S. J. Hanson and R. R. Rosinski. Programmer perceptions of productivity and programming tools. *Communications of the ACM*, 28(2):180–189, 1985.

[29] S. A. Haslam, R. A. Eggins, and K. J. Reynolds. The ASPIRe model: Actualizing social and personal identity resources to enhance organizational outcomes. *Journal of Occupational and Organizational Psychology*, 76:83–113, 2003.

[30] L. Hatton. The chimera of software quality. *Computer*, 40(8):102–104, 2007.

[31] E. J. Hill, B. C. Miller, S. P. Weiner, and J. Colihan. Influences of the virtual office on aspects of work and work/life balance. *Personnel Psychology*, 51:667–683, 1998.

[32] D. R. Jeffery and M. J. Lawrence. Some issues in the measurement and control of programming productivity. *Information & Management*, 4(4):169–176, 1981.

[33] Z. Jiang, P. Naudé, and C. Comstock. An investigation on the variation of software development productivity. *International Journal of Computer and Information Science and Engineering*, 1(2):72–81, 2007.

[34] C. Jones. *Programming Productivity: Steps Toward a Science*. McGraw-Hill Series in Software Engineering and Technology. McGraw-Hill, 1986.

[35] C. Jones. *Applied Software Measurement: Assuring Productivity and Quality*. McGraw-Hill, 1991.

[36] C. Jones. *Software Assessments, Benchmarks, and Best Practices*. Addison-Wesley Information Technology Series. Addison-Wesley, 2000.

[37] B. Kitchenham and E. Mendes. Software productivity measurement using multiple size measures. *IEEE Transactions on Software Engineering*, 30(12):1023–1035, 2004.

[38] B. Lakhanpal. Understanding the factors influencing the performance of software development groups: An exploratory group-level analysis. *Information and Software Technology*, 35(8):468–473, 1993.

[39] K. Lehigh. Improving software productivity using lean six sigma methods and tools. In *Proc. 2007 Crystal Ball User Conference*, 2007. `http://www.crystalball.com/cbuc/2007/papers.html`. Accessed 2007-09-11.

[40] K. R. Linberg. Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*, 49(2-3):177–192, 1999.

[41] G. A. Luedecke, editor. *Mehr Produktivität durch gute Räume*. ECON Verlag, 2. edition, 1992.

[42] A. MacCormack, C. Kemerer, M. Cusumano, and B. Crandall. Trade-offs between productivity and quality in selecting software development practices. *IEEE Software*, 6:78–79, 2003.

[43] K. Maxwell, L. Van Wassenhove, and S. Dutta. Software development productivity of european space, military and industrial applications. *IEEE Transactions on Software Engineering*, 22(10):706–718, 1996.

[44] K. D. Maxwell and P. Forselius. Benchmarking software development productivity. *IEEE Software*, 17(1):80–88, 2000.

[45] E. R. McLean, S. J. Smits, and J. R. Tanner. The importance of salary on job and career attitudes of information systems professionals. *Information & Management*, 30:291–299, 1996.

[46] P. Mohagheghi and R. Conradi. Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empirical Software Engineering*, 12(5):471–516, 2007.

[47] D. Port and M. McArthur. A study of productivity and efficiency for object-oriented methods and languages. In *Proc. Sixth Asia-Pacific Software Engineering Conference (APSEC '99)*, pages 128–135. IEEE Computer Society, 1999.

[48] R. Premraj, M. Shepperd, B. Kitchenham, and P. Forselius. An empirical analysis of software productivity over time. In *Proc. 11th IEEE International Software Metrics Symposium (METRICS '05)*. IEEE Computer Society, 2005.

[49] Y. W. Ramírez and D. A. Nembhard. Measuring knowledge worker productivity: A taxonomy. *Journal of Intellectual Capital*, 5(4):602–628, 2004.

[50] R. H. Rasch. An investigation of factors that impact behavioral outcomes of software engineers. In *Proc 1991 conference on SIGCPR*, pages 38–53. ACM Press, 1991.

[51] C. Roth and K. Moser. Partizipatives Produktivitätsmanagement (PPM) bei komplexen Dienstleistungen. *Zeitschrift für Personalpsychologie*, 4(2):66–74, 2005.

[52] M. Ruhe and S. Wagner. Using the ProdFLOW approach to address the myth of productivity in R&D organizations. Submitted to ESEM '08, 2008.

[53] R. A. Scudder and A. R. Kucic. Productivity measures for information systems. *Information & Management*, 20(5):343–354, 1991.

[54] D. Simmons. Communications: a software group productivity dominator. *Software Engineering Journal*, 6(6):454–462, 1991.

[55] S. Sonnentag. Streß in SE-Projekten. In Brodbeck and Frese [13], pages 71–85.

[56] K. Spiegl. Projektmanagement Life – Best Practices und Significant Events im Software-Projektmanagement. Masterarbeit, Universität Wien, 2007.

[57] M. Tsuda, Y. Morioka, M. Takadachi, and M. Takahashi. Productivity analysis of software development with an integrated CASE tool. In *Proc. 14th International Conference on Software Engineering (ICSE '92)*, pages 49–58. ACM Press, 1992.

[58] J. Turcotte and L. W. Rennison. The link between technology use, human capital, productivity and wages: Firm-level evidence. *International Productivity Monitor*, 9:25–36, 2004.

[59] J. M. Verner and W. M. Evanco. In-house software development: What project management practices lead to success? *IEEE Software*, 22(1):86–93, 2005.

[60] A. von Mayrhauser, C. Wohlin, and M. C. Ohlsson. Assessing and understanding efficiency and success of software production. *Empirical Software Engineering*, 5(2):125–154, 2000.

[61] J. Vosburgh, B. Curtis, R. Wolverton, B. Albert, H. Malec, S. Hoben, and Y. Liu. Productivity factors and programming environments. In *Proc. 7th International Conference on Software Engineering (ICSE '84)*, pages 143–152. IEEE Press, 1984.

[62] C. E. Walston and C. P. Felix. A method of programming measurement and estimation. *IBM Systems Journal*, 16(1):54–73, 1977.

[63] C. Wohlin and M. Ahlgren. Soft factors and their impact on time to market. *Software Quality Journal*, 4(3):189–205, 1995.

[64] C. Wohlin and A. A. Andrews. Assessing project success using subjective evaluation factors. *Software Quality Journal*, 9(1):43–70, 2001.

[65] C. Wohlin and A. A. Andrews. Analysing primary and lower order project success drivers. In *Proc. 14th International Conference on Software Engineering and Knowledge Engineering (SEKE '02)*, pages 393–400, New York, NY, USA, 2002. ACM Press.

[66] C. Wohlin and A. A. Andrews. Prioritizing and assessing software project success factors and project characteristics using subjective data. *Empirical Software Engineering*, 8(3):285–308, 2003.