Productivity measurement in Software Engineering: a study of the inputs and the outputs

Adrián Hernández-López

Department of Computer Science, Universidad Carlos III de Madrid, Spain

Ricardo Colomo-Palacios *

Faculty of Computer Sciences, Østfold University College, Norway

Pedro Soto-Acosta

Department of Management & Finance, Universidad de Murcia, Spain

Cristina Casado Lumberas

Department of Education, Universidad Internacional de la Rioja, Spain

Abstract

Productivity measurement is constructed by the measure of tree categories of elements: inputs, outputs and factors. This concept, which started being used in the manufacturing industry, has been also a research topic within Software Engineering (SE). In this area, the most used inputs are time and effort and the most used outputs are source code and functionality. Despite of their known limitations, many of the most used productivity measures are still being used due to the information that they provide for management goals. In order to enable the construction of new productivity measures for SE practitioners, the existence of other inputs apart from time and effort, and other outputs, apart from source code and functionality is analyzed in this paper. Moreover, differences in usage of the inputs and production of the outputs among some SE job positions are analyzed and explained.

Keywords: software engineering; job position; knowledge worker; productivity; measurement.

1. Introduction

A wider use of productivity measures started with the industrialization era. The complexity of the measures has increased since those days. And measures have to take into account now more items than just the manufactured objects and the labor effort. These items that compose productivity measures are twofold: inputs (the elements needed in the production process) and outputs (the elements produced in the production process). In addition, factors, which are not considered neither inputs nor outputs, influence the productivity results (Lagerström, Würtemberg, Holm, & Luczak, 2012; Trendowicz & Münch, 2009). A relationship between these elements (inputs and outputs) is commonly established in order to construct a productivity measure. Typically, this relationship is a ratio, i.e. how much output is produced per unit of input.

Software Engineering (SE) is a new industry in comparison with traditional manufacturing sectors where the productivity measurement was born. In this area, productivity measurement has been present in research since the late 70s and beginning of the 80s (Brooks Jr., 1985). The productivity measures used in those years followed a ratio approach between outputs and inputs, and new approaches started to be considered due to the increase on the area research. The most used inputs were time and effort while the most used outputs were source code, and later functionality. Nowadays, the most considered inputs are still the same. Nevertheless, the functionality is more used than the source code, which has been shelved, in the most commonly used measures (Petersen, 2011).

However, SE practitioners can be classified as knowledge workers which are significantly different than the workforce that was employed at the beginning and middle of the last century. These practitioners handle other type of inputs, in addition to the resources given by the manufacturing workers (their skills, strength, and time). They give to the employers their knowledge, experience, and social competences among others. On the other hand, they do not produce a single output, as industrial age workers did. They produce an amalgam of outputs including intangible elements such as quality or added value.

Any worker performs a job position regardless of the industry in which he/she works. Each job position represents a unique definition of a job within an organization. These positions can be generalized in order to make them applicable to other more specific job

positions. For example, a definition of the software engineer job could be obtained from O*Net ("Software Engineer (O*Net Definition)," 2010); nevertheless this position has be characterized to each organization in order to adapt it to their reality, missions and goals. Each job position requires a set of inputs in order to produce a set of outputs within a production process and with the interaction of other job positions. So, if the job positions are different among themselves, then the productivity measures considered, and later used, for different jobs can be different. One example within SE is represented by the use of lines of code or functionality as the solo output in productivity measures. So, do all the jobs in SE produce these outputs? Are there any other possible outputs that could be considered when measuring SE practitioners' productivity?

In addition, the level of measurement has to be taken into account. Within SE, there is a lack of research on the lower levels (e.g., team and individual level) for productivity measures (Petersen, 2011). Also, the same measures are used in various levels (e.g., using a project level measurement to measure a programmer productivity) despite of their differences (Hernández-López, 2012; Hernández-López, Colomo-Palacios, & García-Crespo, 2013). Considering that the granularity of possible measures for inputs and outputs is not the same at different levels of measurement, then using the same measures at different levels do not make sense (e.g., project and worker levels). Thus, using the same measures at job level and in higher levels can be seen as controversial.

In this work the existence of different inputs (other than the time and the effort) and different outputs (other than the source code and the functionality) is analyzed. Moreover, the existence of differences on the usage of the inputs and on the production of the outputs among some SE job positions will be also tackled. For these purposes, the authors developed and executed a method divided into two phases. The first phase explores the goals (and generates the research hypotheses) and the second phase contrasts the stated hypotheses from the previous phase. The remainder of this paper is organized as follows: first the background related to the concepts under study is presented; second, the method used for the research is explained; third, the method results are used to contrast the stated hypotheses; fourth, the results are discussed; and finally, there is a discussion of the results and findings.

2. Background

Traditionally, productivity has been defined as the ratio of output produced per unit of input (Jefferys, Hausberger, & Lindblad, 1954); the inputs are the resources needed to produce the outputs produced. This definition fits perfectly into the manufacturing paradigm, because it is based on standardized quantities of measurement units clearly identified for inputs and outputs. However, it does not fit in the service industries neither in Information and Technology industry, in which there are many non standardized units, and many intangible elements that coexist with tangible ones (e.g., quality or client satisfaction). These issues makes the measurement of workers' productivity in these industries a challenge (Drucker, 1999).

Within SE, productivity is usually measured by a product size ratio between the effort required to produce the product (MacCormack, Kemerer, Cusumano, & Crandall, 2003): e.g., Source Lines of Code per unit time (SLOC/t) or some variant of Function Points per unit time (FP/t). Some of the most commonly used productivity measures are presented in Table 1. The study of productivity in the field began in the late 70's and early 80. In those days, the productivity measures were focused on the programming activity (Chrysler, 1978). And the literature underlined the importance of human factors in software development. In that context, the measures were created based on the engineering philosophy that sees productivity as a synonym of resource efficiency. In that sense, the IEEE Std. 1045-1992 defines productivity as the relationship of an output primitive (source statements, function points or documents) and its corresponding input primitive (effort, e.g. staff-hours) to develop software. So the concept of inputs and outputs appear in the SE productivity definitions like in the traditional definitions.

Regarding the level of measurement, the research done in SE has focused on: the sector (Tsunoda, Monden, Yadohisa, Kikuchi, & Matsumoto, 2009), the organization (Anda, Sjoberg, & Mockus, 2009), and the project level (Kitchenham & Mendes, 2004). But the research at lower levels (e.g., teams, job) has not attracted the same interest. Nevertheless, at

team level more research have been done (e.g., Melo, Cruzes, Kon, & Conradi, 2013) than at job level (Hernández-López, et al., 2013), perhaps because of the difficulty of the measurement at that level (Ramirez & Nembhard, 2004) or due to the importance of team concept in software engineering (Salas, 2005). However, it is noteworthy that the measures employed at the job level are the same as those used to higher levels of measurement (Hernández-López, Colomo-Palacios, García-Crespo, & Cabezas-Isla, 2011). This may lead to a problem because the resources used (inputs), the products and/or services produced (outputs) and the factors that affect productivity at this level are not the same as at higher levels. So the measures employed at the job level are of doubtful validity for this level of measurement (Briand, Morasca, & Basili, 2002). As a consequence of this, further research on new productivity measures for software practitioners makes sense; e.g. using a conceptual approach (Yusoff, Mahmuddin, & Ahmad, 2012).

Knowing exactly what a measurement intend to measure is a key milestone before any attempt to measure something, because without this knowledge it is impossible to establish a measurement (Tangen, 2005). In this direction, a way for obtaining this knowledge is asking about a definition of the measure (Sink, Tuttle, & DeVries, 1984). So, in a previous research the authors asked the SE practitioners about the definitions they give of productivity in three levels of measurement (organization, project and personal) in a previous research (Hernández-López, Colomo-Palacios, & García-Crespo, 2012). The results pointed to a more abstract definition of productivity. Specifically, the obtained definitions added the concept of work/task in place of the specific measures of outputs such as SLOC or FP. This added concept is similar to the "human productivity" as is defined in ISO 9126-4. So, the obtained definitions are less clear than conventionally used definitions such as SLOC/t or FP/t. Anyway, the results present a common point with previous definitions: the use of time as a resource. Once the definition of SE productivity at the job level has been outlined, it is possible to follow the steps to the construction of new productivity measures: knowing what inputs and outputs could be included in that measure.

2.1 Inputs and outputs included in software engineering productivity measures

As before stated, and in order to construct new productivity measures in SE, identifying the inputs and the outputs that should be included in those measures is a needed task. In order to do so, the authors briefly introduce the current situation and finish with the statement of two hypotheses.

On the one hand, the most used input is a unit of time, mainly the worked hours or days (Gómez, Oktaba, Piattini, & García, 2008). The use this input, as the only input, may be due to two important concerns for project success: the project delivery on time and the time to market of the product (Trendowicz & Münch, 2009). Moreover, the personnel cost is the largest cost in a software company, so the time consumed in developing the software is, as a consequence, the main cost (Jørgensen & Shepperd, 2007). The use of time as the only input fits within the *economist* definitions of productivity (Ghobadian & Husband, 1990), but productivity is not just an economic indicator. Furthermore, the use of the time as the only input raises some questions unanswered. What "time" should be measured? The time employed by personnel for carrying out their assignments? Or the time paid (contract) despite of the (real) worked hours? From other point of view, the hardware and equipments costs have been reduced, and it is almost irrelevant in large software projects. Thus, the focus on inputs for measuring productivity will be mainly related, directly or indirectly, with the human resources involved in the projects. Consequently, time is, without any doubt, one of the main inputs, but not the only one (Boehm & Ross, 1989).

On the other hand, the most widely used outputs follow the same criteria than the inputs, i.e. the focus on product delivery. So, the outputs that have been more mainly measured are size of source code or functionality. These measures are quite popular in organizations despite their limitations. The functionality is (usually) measured by FP (in any of its variants), which represent the amount of functionality to develop and, therefore, a measure of what is delivered to the client. The source code is measured by SLOC, which represent the size in lines of code of the developed product, which is a low profile measure and it is more tangible than FP measures. However, these measures are not the only ones that can be used to measure the outputs in SE. For example, a major output, which is outside of

these outputs, is the quality, which has effect into the output produced (effectiveness) and into the production process itself (efficiency). In addition, there are some characteristics of the outputs produced that can affect the productivity results, both for its initial creation and its subsequent use. Some examples of this issue are reuse (Anselmo & Ledgard, 2003) and documentation (Boehm, 1987). Moreover, the correlations between inputs and outputs considered in the software productivity measurement are not always what one would expect or intuit (Rodríguez, Sicilia, García, & Harrison, 2012).

Previously to the present paper, a Systematic Literature Review (SLR) was carried out with the purpose of knowing the inputs and the outputs used in the measurement at job level and also the type of measurements used (Hernández-López, et al., 2013). This work has pointed out that there is certain unanimity with respect to the inputs used in the measurement of productivity, as it was stated before. This circumstance is rooted on the fact that the inputs used in the measures at higher levels (for instance, at organizational level) are normally the time or the effort (e.g., Anselmo & Ledgard, 2003; Kitchenham & Mendes, 2004). The use of this input as the only input at lower levels matches perfectly in manufacturing scenarios. In such setups each task has a definite time for each operation and thus the planning and measurement is based on those measures. SE workers can be considered knowledge workers or white-collar workers. An example input used by these workers is knowledge along with other intangible resources. In any case, time is a resource that is equally consumed by knowledge workers. So, the time as a solo measure does not reflect the characteristics of the SE jobs but is needed to illustrate the SE practitioners' performance.

With regard to the outputs, the SLR results point to two totally different output groups (Hernández-López, et al., 2013). One group is focused on the use of SLOC as a productivity measure. This approach may be of great interest in certain environments such as software factories where the core business is to code the designs and requirements of their clients. The other group is focused on the completion of tasks. This group opens a parallel way for measuring the productivity and fits within the engineering point of view (Ghobadian & Husband, 1990). The measures within this group are general because their usage is universal - any job, usually in its job description has the tasks to be performed defined. However, this

group has the difficulty of assessing the work by using a particular measure that should be different depending on the characteristics of each job position.

Taking into account the previous researches, the authors formulate a first group of hypotheses to be tested in this work:

Hypothesis 1. Apart from the time, there are other inputs be used by software development workers, and are suitable to be included in the productivity measures.

Hypothesis 2. Apart from the SLOC and the functionality, there are other outputs produced by software development workers, and are suitable to be included in the productivity measures.

2.2 Inputs and outputs under job position point of view

The structure of any organization, regardless of its form, requires the definition of the jobs that form its structure. These definitions are the gears that, all together, should meet the goals and mission of the organization. Specifically, a job definition may contain at least the following information: its role and mission, its situation within the organizational chart, the tasks to be performed in it (including information about what is needed, how is done, for what is done, the frequency of each task, the dedicated time, the autonomy, the relationships with other jobs, etc.), a workflow, the required effort (physical and intellectual), the risks, the working conditions, the supervision required and/or carried, the knowledge and skills necessary, the values, and the access to other jobs within the career path of the organization (Hackman & Oldham, 1980). In addition, it should be noted that the same person may hold more than one job position in the same organization, and that several people can perform the same job position.

Within SE, there are several jobs, more or less universally recognized, although its definition varies in each organization and are constantly updated (Colomo-Palacios, Tovar-Caro, García-Crespo, & Gómez-Berbís, 2010). For example, job positions such as project manager, programmer or analyst have been reference positions, both in academia and in industry, for years but their definition is vague (Litecky, Aken, Ahmad, & Nelson, 2010).

Despite of the variety of job positions, the SE jobs can be grouped within the so-called white-collar workers, also called knowledge workers (Drucker, 1999; Ramirez & Nembhard, 2004). These jobs are mainly characterized by the use of human, intellectual and relational capital to perform the tasks and reach their objectives (Colomo-Palacios, Cabezas-Isla, García-Crespo, & Soto-Acosta, 2010). These characteristics differ from the jobs in other industries where labor from a physical point of view, rather than intellectual, is required. As well as, the resources needed for jobs within IS are largely intangibles (e.g., knowledge and experience) versus the tangible assets of the traditional industries (Rus & Lindvall, 2002) .

Thus, the inputs used and the outputs of each job must be explicit in the definition of each job position (Hackman & Oldham, 1980). For example, let us consider the Software Analyst job definition provided in *Métrica 3* ("Métrica Versión 3," 2000), which defines its goals as *"[...] to develop a detailed catalog of requirements that describe accurately the information system, for which he or she holds interviews and working sessions with managers of the client organization and users, acting as the interlocutor between them and the project team as far as requirements are concerned. These requirements allow analysts to develop different models that provide the basis for the design, obtaining the data models and process, in the case of structured analysis, and the classes model and objects interaction, in object-oriented analysis. Also he or she does the interface specifications between the system and the user."* (*Métrica* 3 is only available in Spanish so the authors of this paper have translated this fragment). According to this definition, it can be said that the outputs produced by an analyst are, but not limited to, the requirements catalogs, the data models, the processes models, the class and interaction models, and the interface specifications. With regard to the inputs that analysts use, it can be said that an analyst employs the knowledge of the users and the client, and also the requirements catalog. In addition, to generate the outputs there is an interaction with other jobs and with people outside the organization so this interaction can be seen also as another input. This example illustrates the difficulty of measuring productivity at the job level because the typically outputs and inputs used in the commonly used measures are not used (or at least they are not visible in the definition) and there are other inputs and outputs uncommonly included. Also, in this example the boundary between the inputs and the outputs

is diffuse. The catalog of requirements is produced ("[…] *to develop a detailed catalog of requirements* […]") and it's also used ("[…] *these requirements allows the analyst* […]"). This is a classical problem within service industries (Gupta, 1995). Thereby, it seems clear that the inputs and outputs used for productivity measurement within SE, at the job level, should be linked to the job positions and not just to the global inputs and outputs of the development process.

Therefore, considering the previously presented issues, the authors formulate a second group of hypotheses to be tested in this work:

Hypothesis 3. The inputs used are different for each job position in software development projects.

Hypothesis 4. The outputs produced are different for each job position in software development projects.

3. Method

The method has been split in two phases. In the first phase, an exploratory qualitative research was carried out in order to know if the research was in the right direction. In the second phase, an empirical research were developed and executed to obtain data that enable the authors to contrast the hypotheses.

Figure 1. Research Method

INSERT FIGURE 1

3.1 Phase I: qualitative research

Given the exploratory nature of this phase, the authors decided to use a qualitative methodology. Qualitative research is primarily used for the investigation of sociological events, cultural and anthropological, i.e., situations where people are involved (Denzin & Lincoln, 2011). Given that SE is highly dependent on human capital, using a qualitative methodology makes sense (Hove & Anda, 2005). Furthermore, qualitative research investigates from the data to create theories. This research approach is highly valuable taking

into account that the object of study - worker productivity - is an area requiring further research (Ramirez & Nembhard, 2004). This type of research has been used in the area of knowledge workers productivity in a satisfactory manner (Erne, 2011).

In particular, a content analysis approach by the use of semi-structured interviews as information gathering method were used (Kvale, 2008). This data collection approach provides information that could not be obtained through a quantitative approach as it allows opinions, thoughts and feelings (Rubin & Rubin, 2011). The process used to obtain participants was as follows. Firstly, one of the authors contacted via email with ex-alumni with experience of at least one year in any of the activities of SE. From these emails, 15 positive responses were obtained. Secondly, the interviews were conducted between April and October 2011, all were conducted in Spanish (although some participants were working abroad).

The final sample consisted in 15 subjects (14 men, one woman). Four of them worked as Project Managers (PM) ("Project Manager (O*Net Definition)," 2013) while the others (11) worked as Software Engineers (SEs) ("Software Engineer (O*Net Definition)," 2010) in nine different organizations. The mean age was 30.47 years (5.18 SD), with an average seniority of 2.56 years in the current job position (2.71 SD) – 5.13 years of seniority in the case of PM and 1,63 for SEs - and an average of 5,93 years in the field of SE (4.85 SD) - 10,75 years in the field for PM and 4.18 for SEs.

The total recorded time of interviews was 9 hours, 43 minutes and 18 seconds with an average of 38 minutes and 53 seconds per interview. Interviews were conducted and recorded by an interviewer, and later transcribed by the same interviewer (a translation of the interview script is available online[i]). The interview script included a question about what the professional uses to produce the job's outputs and another about what he/she produces in the job. It also had other questions related to the same global topic: the definition of productivity (Hernández-López, et al., 2012) or the job satisfaction and motivation along with the factors that influence productivity (Hernández-López & Colomo-Palacios, 2012).

The process used for codification was as follows. Before transcribing the interviews, the authors created an initial code list which included the codes that cover the inputs and

outputs that could be mentioned. These codes were based on author's knowledge in the field, on the state of the art about productivity measures, and thus the possible inputs and outputs, and also based on the research goals (Miles & Huberman, 1994). During transcription and coding process, the authors added extra *in vivo* codes (based on a word or short phrase with a significant meaning by itself) to the list in order to specify as much as possible the items mentioned by the participants. Finally, the authors got the list of codes and sub codes from the previous coding process based on an open coding approach (Strauss & Corbin, 1990). Once all data were coded, the data related to inputs and outputs were isolated to be analyzed and discussed. *Atlas.ti 6* software was used for transcription and coding the interviews.

3.2 Phase II: quantitative research

Once the results of the qualitative research were analyzed, it was possible to continue. From the results, the authors constructed a questionnaire to be able to contrast the stated hypotheses. In SE research, questionnaires are a frequent information gathering tool (Ciolkowski, Laitenberger, Vegas, & Biffl, 2003). The designed questionnaire contained demographic questions and 32 Likert questions of six values (1 - Never, 2 - Very rarely, 3 - Rarely, 4 - Occasionally, 5 - Frequently, 6 - Very Frequently) for the four job positions defined in a Spanish software development methodology called *Métrica 3* ("Métrica Versión 3," 2000). The jobs are programmer, analyst, consultant, and project manager. For each job, each participant may fill 32 questions, 16 concerning the degree of utilization of resources (inputs) and 16 regarding to the level of production of products/services (outputs) for the job positions in which they have experience. These 32 items are drawn from the results obtained in the previous phase. The items were selected from the most mentioned items, and adding the inputs and outputs most used in productivity measurement.

The questionnaire was created with Google Docs platform which enables the construction of electronic online questionnaires. Two media were used to deliver the survey. On the one hand, an email was sent to personal contacts of the authors. And on the other hand, the authors wrote some posts in LinkedIn groups related to SE. Specifically, 300 emails

were sent and it was posted into 36 groups. The questionnaire included a question about how the participant has accessed to it, in order to establish the response rate for each publishing method. Within this question, a third option was added to allow a third case in which the participant accesses the questionnaire through a known person.

The inputs selected for the questionnaire were: *Time, Knowledge, Planning, Estimation, Allocated Goals, Software, Hardware, Working Facilities, Requirements Specification, Functional Knowledge, Client, Motivation, Documentation, Experience, Education,* and *Previous Source Code*. And the outputs were: *Source Code, Product, Documentation, Finished Task, Goal Committed, Estimation, Planning, Quality, Sales, Tests, Experience, Knowledge, Problem Solved, Bug Solved, Client Satisfaction,* and *Functionality*. These lists are not limited, i.e. this is not the entire universe of possible inputs and outputs but represents a significant sample of them according to the results from the previous phase.

The non-parametric Kruskal-Wallis test was selected to check the differences in the degree of utilization of inputs and the degree of production of the outputs for the jobs under study. This test was chosen at the expense of one-way ANOVA, since the data to be analyzed are ordinal (Likert scale). In addition, the Dunn's test was used as post hoc test, which is the non-parametric analog of Holm-Sidak multiple t-test, in order to compare the difference between job positions for each input and output.

The final sample presents the following characteristics: 158 responses to questionnaire for a total of 345 jobs (125 programmers, 95 analysts, 65 consultants and 60 project managers), with a mean of 2.18 jobs per response; 131 men (82.91%) and 27 women (17.09%), with an average age of 33.94 years (8.62 SD). With regard to the media used to access the questionnaire: 89 (56.33%) accessed from the email (24% response rate), 49 (31.01%) from the LinkedIn groups and 20 (12.66%) from known persons.

4. Results

4.1 Phase I: qualitative research

It has to be taken into account that during the paper writing, the terms *code* and *sub code* were change to Items and Sub Items. For example, the authors got an Item tagged "Quality" with various Sub Items tagged "Source code quality", "Product quality", "Value added"... The results are presented in two separate subsections to make them more readable: one for the inputs and other for the outputs. Also a preliminary contrast of the hypotheses based on these results is presented in this section.

Inputs

The inputs reported by participants are included in Table 2 (and our research definition of each item is available online[ii]). The most mentioned input was time and work management (n=14). Time include as sub items: date (8), planning (5) and delivery on time (3). This result is in consonance with the importance of time in the most commonly used measures of productivity. Also, work management has some concepts related to time such as goal allocation (6), task allocation (6), and estimation (4). With less mentions, there are other inputs: requirements (11), knowledge (11), client (11), resources (13), documents (8), experience (8), and education and training (3). These results illustrate the existence of outputs used also as inputs within the same job position.

Within requirements, which are an input for many SE tasks, their definition is the most cited sub item (10), along with changes on them (5). In addition, the integration of client for requirement elicitation (7) which is under client item is related to the requirements item. Also, the customer tastes (1) which is under other intangibles item is related to both of these items.

Moreover the following sub items were mentioned more than once: software (8), hardware (7), software documentation (5), self-experience (5), project documentation (4), team experience (3), specific training (3), experience in similar tasks (3), working facilities (2), continuous learning (2), material resources (2), know-how (5), resolution of doubts (3), project knowledge (3), functional knowledge (2), and previous product (2).

Outputs

The outputs reported by the participants are included in Table 3 (and their research definition is available online – see Inputs). The most mentioned output was documentation (n=13), including as sub items: design (6), analysis (6), project (4), process (3), source code (2), and non-specified documentation (7). The participants also mentioned source code (10), work management (9), knowledge (9), quality (8), sales (7), tests (5) and experience (1). And some tangible outputs (12) such as product (9) and requirements specification (2); and some intangible outputs (12) such as bug and problem solved (6), analysis (4), project (3), functionality (3), design (3), client satisfaction (3), and other ones with just one mentioned.

The most used outputs used within the productivity measures are present in the results: source code (10) and functionality (3). In addition, the documentation, which is usually calculated with respect of the SLOC, is also present. It is important to note that some items that are mentioned as outputs were also mentioned as inputs illustrating the transformation processed of some items during the development process. For example, knowledge (output 9, input 11), and experience (output 8, input 1). Therefore, a transformation process for some inputs is present on these jobs and adds more difficulty to the productivity measurement task.

Contrast of Hypotheses

Despite of the exploratory nature of this phase, it is possible to tentatively contrast the hypotheses H1 and H2. Regarding H1, there were other items (e.g. requirements, knowledge, client, resources, documents, experience...) and not just the time as input. And related to H2 there were mentions to source code and functionality but also to other outputs. Thus, H1 and H2 can be considered as (preliminarily) verified despite of not having a statistical support.

H3 and H4 cannot be contrasted in this phase due to the reduced sample used. However, some relevant information can be extracted. For example, sales are mentioned as an output by four software engineers (36.36%) and by three project managers (75%), so it is

possible to start thinking about a difference in the degree of outputs produced by each job position (H4). On the other hand, the previous source code was mentioned by two software engineers (18.18%) but not by project managers, so H3 could be (preliminarily) supported. In any case, these results are just tentative, and its final assessment is made on the second phase of this research.

4.2 Phase II: quantitative research

The descriptive results of this phase are included in Table 4 (inputs) and Table 5 (outputs); the definitions of these items are the same used in the previous phase. From these data the authors can state that nearly all of the inputs selected for the survey are widely used by the selected job positions: the items have a median equal or greater than 4 (occasionally) except Previous Source Code. From the point of view of the outputs, only Sales (for programmers and consultants), Source Code (for analysts, consultants and project managers), Bugs solved (for analysts, consultants and project managers), and Product (for consultants) have a median lower than 4. Thus, the selection of inputs and outputs can be considered as accurate for the selected job positions.

The Kruskal Wallis test results are included in Table 6. Many of the inputs included in the survey (11/16) present a statistical significant difference in their usage by some of the job positions. Therefore H3 is supported. On the other hand, many of the outputs included in the survey (10/16) present statistical significant differences in their production by some of the job positions. So H4 is supported. In addition, and in order to known the differences in the inputs used and outputs produced between the jobs that had not equal median, i.e. those items with $p < 0.05$ in Kruskal Wallis test, the authors executed the Dunn test.

The first step for constructing a Dunn test is to establish the significance level of Dunn test (Equation 1). In this case the value is 0.004166667.

Equation 1. Significance level of Dunn test

$$\alpha = \frac{\alpha'}{K(K-1)}$$

where

α' is the adjusted alpha used in Kruskal Wallis test (0.05),

and K is the number of groups (4).

The next step is to establish the theoretical difference for each pair of compared groups for each item (Equation 2).

Equation 2. Theoretical difference for each pair of compared groups

$$\Delta_{ij} = Z_{1-\alpha}\sqrt{\frac{N(N+1)}{12}\left(\frac{1}{n_i}+\frac{1}{n_j}\right)}$$

where

$Z_{1-\alpha}$ is the inverse of the accumulative normal distribution (2.638257273),

N is the size of the sample (345),

and n_i and n_j are the sizes of the compared groups (125, 95, 65, 60).

The last step is to calculate the observed difference for each item within each group (Equation 3). If the observed difference is higher than the theoretical difference (Equation 2) then there is a significant difference between the groups for that item. The theoretical differences are included in Table 7, and the observed differences along with the mean ranks for each item are included in Table 8 (inputs) and Table 9 (outputs).

Equation 3. Observed difference for each pair of compared groups

$$\Delta'_{ij} = \left|\overline{R}_i - \overline{R}_j\right|$$

where

$\overline{R}_i$ and $\overline{R}_j$ are the mean ranks of the two groups under comparison.

A special case appeared in the case of motivation (input) where no differences were found in the Dunn test results despite the Kruskal Wallis results pointed to a difference between some of the jobs. Then, the authors decided to test the differences using U Mann-Whitney test for this input and found statistical differences between the following pairs of job positions: programmer vs. consultant ($z = -2.335$, $p < .05$), programmer vs. project manager ($z = -2.129$, $p < .05$), and analyst vs. consultant ($z = -2.079$, $p < .05$).

From the inputs point of view, the most different job is programmer with 21 significant differences found, followed by project manager with 16 differences (out of 96 possible differences). In addition, it can be observed that for some inputs there are differences among roles. For example, the degree of usage of Planning and Allocated Goals by project managers is statistically different from the other jobs. The same pattern is reproduced in the case of the programmers. For example, the degree of usage of Client, Software, Hardware, and Previous Source Code is statistical different from the other jobs. In the outputs side, the most different is again the programmer job position with 22 differences found followed by project manager with 9 (out of 96 possible differences). The same pattern that was present in the inputs usage is present in the outputs production. For example, the production of Source Code, Planning, Sales, and Bug Solved are statistically different from each other jobs. These patterns add extra support to H3 and H4.

5. Discussion

The results can be discussed from different points of view. One of them is the definition of productivity. As it was previously stated, the IEEE Std. 1045-1992 defines the productivity as the relationship of an output primitive and its corresponding input primitive to develop software. So, from the results it is possible to establish many relationships between primitive inputs and outputs (e.g., a relationship between the source code developed and the knowledge used to produce it, or between previous source code and the bugs solved, or between the knowledge used and the knowledge generated). Nevertheless, it seems that this definition does not cover all the possible relationships since it establishes a relationship between the inputs and the outputs, and many are not related in pairs, e.g. the hardware used (input) and the sales (output). In addition, this definition is reduced to just a relationship between one input and one output (1 - 1) and the results point to a multiple relationship between the inputs and the outputs (M - N).

From the point of view of the most commonly used productivity measures in SE, results are controversial. The most commonly used measures use a ratio (relationship) between a measure of product size (SLOC or FP) and the effort required to produce it

(hours/man-hours). However, the results point that there are more than one input and one output; consequently many relationships between them could be considered. This limitation of the productivity definitions and the measures commonly could be explained because they are derived from the used at the project level. Given that the definition of productivity is not the same at all levels (Hernández-López, et al., 2012) then the use of the same productivity measures in all levels of the organization does not make sense. In addition, the authors assert that the most commonly used measures are not reliable productivity measures for measuring the SE practitioners' productivity. These measures have to be considered as specific measures of productivity. For example, FP/t measure has to be considered as a merely relationship between the amount of functionality developed and time, i.e. FP/t is a measure of software product delivery productivity, in which the produced software is measured just by the functionality and the time. These measures leave out other outputs (e.g., the quality) and other inputs despite some of them could be considered taking into account that in a lower level of measurement the granularity should be reduced. Hence, if a measure which uses just an output and an input is used, it must be considered that any other modified input and output, when a productivity improvement is pursued, will be external factors and won't be included in the measure. For example, if an organization gives extra training to project managers to improve their project planning and estimations with the purpose of improving productivity of software delivery (i.e., increase the measure output per unit of effort), there will not be possible to check if the further productivity values are influenced by the given training because it is a factor, but is not the unique factor that will be modified in further measurements. In this example, maybe an extra productivity measure for measuring the project planning and estimation productivity should shed some light in combination with other productivity measures, i.e. productivity as a combination of productivity measures or indicators. For this goal it is possible to use Data Envelopment Analysis (DEA) as reported in the literature (Petersen, 2011).

Within the results there are some inputs that were mentioned also as outputs (e.g., knowledge and experience). This fact adds extra difficulty to the task of measuring the practitioners' productivity. This circumstance introduces a new concept not included into the

most used definitions and measures: the transformation and creation of inputs into outputs of the same kind. In those measures and definitions, the inputs and outputs are insulated compartments. Nevertheless, the results point out the existence of some inputs that are outputs at the same time. These inputs are normally modified and used during the job performance of the practitioners. It is widely accepted that if a worker leaves and/or enters an already started project, then the productivity results will change. But, if the time or the effort are the unique inputs used by the SE practitioners to develop software projects, why productivity results change when the team composition changes? One of the possible answers is that there are other inputs which were not taken into account in the previous productivity measures. If just the time or the effort are used within the productivity measures as inputs, then the productivity will not be affected by the project team changes. In other words, the jobs within SE are not as automated and standardized as the manufacturing jobs.

At this point, a proper definition of the job positions seems important for measuring (and improving) the productivity. The job positions definitions include a complete list of the inputs used and the outputs produced along with the work process. So, if there are differences between the inputs used (H3) and the outputs produced (H4) by each job, and if a productivity measure specific to each job is pursued, then, knowing the inputs used and the outputs produced by each job, should be considered a starting point. This task can be done by a job position analysis. In sum, the new productivity measures should consider job position definition as a guide to develop such measures.

6. Conclusion

According to the results it is possible to assert that other inputs, not just the time and the effort, and other outputs, not just the source code and the functionality, are used and produced in the SE jobs. These findings call into question the utility of some of the most widely used productivity measures in SE which measure the relation between a product size measure and the time or effort used to produce it. Thus, those measures should be considered as specific measures of productivity (e.g. the source code delivery productivity) and its validity for measuring a more global concept such as productivity is compromised. In addition, the authors have found difference in the usage of inputs and in the production of

outputs between some pair of SE jobs. This finding lead the authors to think about a specific measure for each job position and not using the same measure for different jobs.

Also, the authors conclude that the SE practitioners can be grouped within knowledge workers under the light of the results. These practitioners produce other outputs intangible which are not commonly measured neither valued within the productivity measures (e.g., experience and knowledge), they use other inputs that are not workforce resources (e.g., training, education, documentation...), and they interact with other people (e.g. with client for requirement elicitation, and with their coworkers). Also, the quality is an output for these practitioners. Therefore, using the same philosophy of the productivity measures that were developed for the manufacturing sector, which focus on the amount of outputs, produced does not make sense within the in SE.

As further researches the authors propose to validate the results with other inputs and outputs, using the same method or other, in order to contrast the hypotheses, i.e. the replication of the presented research. In addition, a taxonomy of the inputs and the outputs that are used and produced within software development process along with possible measures for them could set another start point in the construction of new measures. Finally, the construction of new productivity measures for the SE practitioners that take into account the existence of other inputs and outputs apart from those commonly used and also the differences between job positions is our next research step.

References

Anda, B. C. D., Sjoberg, D. I. K., & Mockus, A. (2009). Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System. *IEEE Transaction Software Engineering, 35*(3), 407-429.

Anselmo, D., & Ledgard, H. (2003). Measuring productivity in the software industry. *Communications of the ACM, 46*(11), 121-125.

Boehm, B. W. (1987). Improving Software Productivity. *Computer, 20*(9), 43-57.

Boehm, B. W., & Ross, R. (1989). Theory-W Software Project Management Principles and Examples. *IEEE Transactions on Software Engineering, 15*(7), 902-916.

Briand, L. C., Morasca, S., & Basili, V. R. (2002). An Operational Process for Goal-Driven Definition of Measures. *IEEE Transactions on Software Engineering, 28*(12), 1106-1125.

Brooks Jr., F. P. (1985). *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley Professional.

Ciolkowski, M., Laitenberger, O., Vegas, S., & Biffl, S. (2003). Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering. *Lecture Notes in Computer Science, 2765*, 104-128.

Colomo-Palacios, R., Cabezas-Isla, F., García-Crespo, Á., & Soto-Acosta, P. (2010). Generic Competences for the IT Knowledge Workers: A Study from the Field. In M. D. Lytras, P. Ordonez De Pablos, A. Ziderman, A. Roulstone, H. Maurer & J. B. Imber (Eds.), *Knowledge Management, Information Systems, E-Learning, and Sustainability Research* (Vol. 111, pp. 1-7): Springer Berlin Heidelberg.

Colomo-Palacios, R., Tovar-Caro, E., García-Crespo, Á., & Gómez-Berbís, J. M. (2010). Identifying Technical Competences of IT Professionals: The Case of Software Engineers. *International Journal of Human Capital and Information Technology Professionals, 1*(1), 31-43.

Chrysler, E. (1978). Some basic determinants of computer programming productivity. *Communications of the ACM, 21*(6), 472-483.

Denzin, N. K., & Lincoln, Y. S. (2011). *The SAGE Handbook of Qualitative Research*. Thousand Oaks, CA: Sage Publications.

Drucker, P. (1999). Knowledge-Worker Productivity: The Biggest Challenge. *California management review, 41*(2), 79-85.

Erne, R. (2011). What is Productivity in Knowledge Work? - A Cross-industrial View. *Journal of Universal Computer Science, 17*(10), 1367-1389.

Ghobadian, A., & Husband, T. (1990). Measuring total productivity using production functions. *International Journal of Production Research 28*(8), 1435-1446.

Gómez, O., Oktaba, H., Piattini, M., & García, F. (2008). A Systematic Review Measurement in Software Engineering: State-of-the-Art in Measures. In J. Filipe, B. Shishkov & M. Helfert (Eds.), *Software and Data Technologies* (Vol. 10, pp. 165-176): Springer Berlin Heidelberg.

Gupta, A. (1995). Productivity measurement in service operations: a case study from the health-care environment. *Managing Service Quality, 5*(5), 31-31.

Hackman, J. R., & Oldham, G. R. (1980). *Work Redesign*. Reading, MA: Addison-Wesley.

Hernández-López, A. (2012). Satisfaction and motivation: IT practitioners' perspective *International Journal of Human Capital and Information Technology Professionals, 3*(4), 39-56.

Hernández-López, A., & Colomo-Palacios, R. (2012, 13 June). *Job Satisfaction and Motivation of Software Engineering Practitioners.* Paper presented at the First Workshop on Managing the Influence of People and Team Factors in Software Engineering (INTEAMSE 2012), Madrid, Spain.

Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2012, 23/06/2012). *Productivity in software engineering: a study of its meanings for practitioners.* Paper presented at the Second European Workshop on Computing and ICT Professionalism (EWCIP 2012), Madrid, Spain.

Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2013). Software Engineering Job Productivity - A Systematic Review. *International Journal of Software Engineering and Knowledge Engineering, 23*(3), 387-406.

Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., & Cabezas-Isla, F. (2011). Software Engineering Productivity: Concepts, Issues and Challenges. *International Journal of Information Technology Project Management, 2*(1), 37-47.

Hove, S. E., & Anda, B. (2005). *Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research*. Paper presented at the Proceedings of the 11th IEEE International Software Metrics Symposium.

Jefferys, J., Hausberger, S., & Lindblad, G. (1954). *Productivity in the distributive trade in Europe: wholesale and retail aspects*: Organisation for European Economic Co-operation.

Jørgensen, M., & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering, 33*(1), 33-53.

Kitchenham, B. A., & Mendes, E. (2004). Software Productivity Measurement Using Multiple Size Measures. *IEEE Transactions on Software Engineering, 30*(12), 1023-1035.

Kvale, S. (2008). *InterViews: Learning the Craft of Qualitative Research Interviewing*. Thousand Oaks, CA: Sage Publication.

Lagerström, R., Würtemberg, L., Holm, H., & Luczak, O. (2012). Identifying factors affecting software development cost and productivity. *Software Quality Journal, 20*(2), 395-417.

Litecky, A., Aken, C., Ahmad, A., & Nelson, J. (2010). Mining for Computing Jobs. *IEEE Software, 27*(1), 78-85.

MacCormack, A., Kemerer, C. F., Cusumano, M., & Crandall, B. (2003). Trade-offs between Productivity and Quality in Selecting Software Development Practices. *IEEE Software, 20*(5), 78-85.

Melo, C., Cruzes, D. S., Kon, F., & Conradi, R. (2013). Interpretative case studies on agile team productivity and management. *Information and Software Technology, 55*(2), 412-427.

Métrica Versión 3. (2000). Consejo Superior de Informática - Ministerio de Administraciones Públicas.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. Thousand Oaks, CA: Sage Publications, Inc.

Petersen, K. (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology, 53*(4), 317-343.

Project Manager (O*Net Definition). (2013). from http://www.onetonline.org/link/summary/15-1199.09

Ramirez, Y. W., & Nembhard, D. A. (2004). Measuring knowledge worker productivity: A taxonomy. *Journal of Intellectual Capital, 5*(4), 602-628.

Rodríguez, D., Sicilia, M. A., García, E., & Harrison, R. (2012). Empirical findings on team size and productivity in software development. *Journal of Systems and Software, 85*(3), 562-570.

Rubin, H., & Rubin, I. (2011). *Qualitative Interviewing: The Art of Hearing Data* (3 ed.). Thousand Oaks, CA: Sage Publications.

Rus, I., & Lindvall, M. (2002). Knowledge management in software engineering. *IEEE Software, 19*(3), 26-38.

Salas, E. (2005). Is there a big five in teamwork? *Small Group Research, 36*(5), 555-599.

Sink, D. S., Tuttle, T. C., & DeVries, S. J. (1984). Productivity measurement and evaluation: what is available? *National Productivity Review, 3*(3), 265-287.

Software Engineer (O*Net Definition). (2010). from http://www.onetonline.org/link/summary/15-1132.00

Strauss, A., & Corbin, J. M. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Thousand Oaks, CA: Sage Publications, Inc.

Tangen, S. (2005). Demystifying productivity and performance. *International Journal of Productivity and Performance Management, 54*(1), 34-46.

Trendowicz, A., & Münch, J. (2009). Factors Influencing Software Development Productivity - State of the Art and Industrial Experiences. In V. Z. Marvin (Ed.), *Advances in Computers* (Vol. 77, pp. 185-241): Elsevier.

Tsunoda, M., Monden, A., Yadohisa, H., Kikuchi, N., & Matsumoto, K. (2009). Software development productivity of Japanese enterprise applications. *Information Technology and Management, 10*(4), 193-205.

Yusoff, M. Z., Mahmuddin, M., & Ahmad, M. (2012, 4-6 July). *A Conceptual Model of Knowledge Work Productivity for Software Development Process: Quality Issues.* Paper presented at the Knowledge Management International Conference (KMICe), Johor Bahru, Malaysia.

Table 1. Commonly used productivity measures used in SE

Items used	*Method*	References
Output (Lines of Code) Input (Effort)	**Ratio** Productivity = LOC / Effort	(López-Martín, Chavoya-Peña, & Meda-Campaña, 2012; MacCormack, et al., 2003; Maxwell, Wassenhove, & Dutta, 1996; Moazeni, Link, & Boehm, 2013; Sison, 2009; Tan et al., 2009)
Outputs (Lines of Code, Function Points) Input (Total Labor Hours)	**Data Envelopment Analysis** Inputs = Total Labor Hour; Outputs = LOC & FP.	(Asmild, Paradi, & Kulkarni, 2006; Liping, Qiusong, Sun, Tong, & Wang, 2005; Ruan et al., 2007; Stensrud & Myrtveit, 2003; Yang & Paradi, 2004)
Output (Functional Size measured in Function Points, COSMIC...). Input (Effort)	**Ratio** Productivity = FS /Effort	(Bok & Raman, 2000; de Souza Carvalho, Rosa, dos Santos Soares, Teixeira da Cunha Junior, & Buiatte, 2011; Desharnais & April, 2010; Desharnais, Yıldızoğlu, April, & Abran, 2013)
Outputs (Adjusted Size, Total Web Pages, High Effort Features/Functions, New Images) Input (Effort)	**Regression** Multiple Size Measures	(Kitchenham & Mendes, 2004)

Table 2. Inputs mentioned by the participants in the qualitative phase

Item	N (SE+PM)	Sub Item	N
Time	14 (11+3)	Time	14
		Date	8
		Planning	5
		Delivery on time	3
Work management	14 (10+4)	Planning	9
		Allocation of objectives	6
		Task allocation	6
		Estimation	4
		Follow-up meeting	4
		Communication with the team members	2
		Management models	2
		Work processes	1
Resources	13 (10+3)	Software	8
		Computer (and Hardware)	7
		Working facilities	2
		Material resources	2
		Workplace	1
		Telephone	1
Requirement	11 (9+2)	Requirements specification	10
		Requirements changes	5
		Requirements accomplished	1
Knowledge	11 (8+3)	Knowledge	8
		Know-how	5
		Resolution of doubts	3
		Project knowledge	3
		Functional knowledge	2
Client	11 (8+3)	Integration for the requirements elicitation	7
		Frequent interaction with the project client	4
		Interaction (with client) for quality assurance tests	2
		Constant interaction with the client	1
Other intangibles	9 (7+2)	Team management competencies	2
		Previous product	2
		Collaboration with the team members	1
		Myself	1
		Human resources	1
		Information	1
		Task's difficulty	1
		Design	1
		Customer tastes	1

Document	8 (7+1)	Software documentation	5
		Project documentation	4
		Reports	1
		Email	1
		Standard	1
Experience	8 (4+4)	Self-experience	5
		Team experience	3
		Experience in similar tasks	3
Other tangibles	5 (5+0)	Previous source code	2
Education and Training	3 (2+1)	Specific training	3
		Continuous training	2
		University education	1

Table 3. Outputs mentioned by the participants in the qualitative phase

Item	**N (SE+PM)**	**Sub Item**	**N**
Documentation	13 (10+3)	Documentation	7
		Design	6
		Analysis	6
		Project	4
		Process	3
		Source code documentation	2
Other tangibles	12 (9+3)	Product	9
		Quality	2
		Requirements specification	2
Other intangibles	12 (8+4)	Bugs and problem solving	6
		Analysis	4
		Project	3
		Functionality	3
		Design	3
		Client satisfaction	3
		Architecture	1
		Experimentation	1
		User story points	1
		Team management	1
		Project enhancement and maintenance	1
		Accomplished requirement	1
Source Code	10 (7+3)	Source code correction	2
Work management	9 (7+2)	Finished tasks	7
		Estimation	4
		Planning	3
		Goal commitment	2
		Task allocation	2
		Coordination	2
		Traceability	1
Knowledge	9 (8+1)	Learning-by-doing	3
		Founded solutions	2
		Resolution of doubts	2
		Information	1
		Knowledge included in the documentation	1
		Project knowledge	1
Quality	8 (6+2)	Quality (in general)	6
		Source code quality	3
		Product quality	2
		People development quality	1
		Developed software performance	1
		Added value	1
Sales	7 (4+3)	Sales	7
Tests	5 (4+1)	Software tests	5
		Special cases	1
Experience	1 (1+0)	Experience	1

Table 4. Descriptive statistics of the inputs

	Programmer (n=125)				*Analyst* (n=95)				*Consultant* (n=65)				*Project Manager* (n=60)			
	Mode	*Percentiles*			*Mode*	*Percentiles*			*Mode*	*Percentiles*			*Mode*	*Percentiles*		
		25	*50*	*75*		*25*	*50*	*75*		*25*	*50*	*75*		*25*	*50*	*75*
Time	6	4	5	6	6	4	5	6	6	5	6	6	6	5	6	6
Knowledge	6	5	6	6	6	5	5	6	6	5	6	6	6	5	6	6
Planning	4[a]	4	5	5	5	4	5	6	5	4	5	6	6	5	6	6
Estimation	4	3	4	5	5	4	5	6	5	4	5	6	6	5	5.5	6
Allocated Goals	5	4	5	5	5	4		5	5	4	5	6	6	5	6	6
Software	6	5	6	6	5	4	5	5	5[a]	3	4	5.5	5[a]	3	5	6
Hardware	6	3	5	6	5	2	4	5	2	2	4	5	4	2	3	4
Working Facilities	5	4	5	5.5	4	3	4	5	5	3	4		5	3	4	5
Requirements Specification	5	3	5	5	6	4	5	6	5	3	4	5	6	4	5	6
Functional Knowledge	5	4	5	5	6	5	5	6	6	4	5	6	5	4	5	6
Client	4	3	4	5	6	4	5	6	6	4.5	5	6	5[a]	4	5	6
Motivation	5	4	5	5	4	4	4	5	5	4	5	6	6	4	5	6
Documentation	5	4	5	5	5	4	5	6	5	3.5	5	5	5	4	5	6
Experience	5	4	5	6	6	4	5	6	6	5	5	6	6	4	5	6
Education	5	4	5	5	5	4	4	5	5		5	6	4[a]	3	4	5
Previous Source Code	4	3	4	5	3	2	3	4	2	2	2	4	2	2	2	4

a. Multiple modes exist. The smallest value is shown

Table 5. Descriptive statistics of the outputs

	Programmer (n=125)				*Analyst* (n=95)				*Consultant* (n=65)				*Project Manager* (n=60)			
	Mode	*Percentiles*			*Mode*	*Percentiles*			*Mode*	*Percentiles*			*Mode*	*Percentiles*		
		25	*50*	*75*		*25*	*50*	*75*		*25*	*50*	*75*		*25*	*50*	*75*
Source Code	6	5	5	6	2	2	3	4	1	1	2	4	1	1	2	4.75
Product	5	4	5	5	5	3	4	5	2[a]	2	3	5	5	2	4	5
Documentation	5	3.5	4	5	6	5	5	6	6	4	5	6	5	4	5	6
Finished Task	5	4	5	5	5	4	5	6	5	4	5	6	6	5	5	6
Goal Committed	5	4	5	5	5	4	5	6	5	4	5	6	5	5	5	6
Estimation	4	3	4	5	5	4	5	5	4[a]	4	5	5	5	4.25	5	6
Planning	4[a]	3	4	5	5	4	5	6	5	4	5	5	6	5	5	6
Quality	5	4	5	5	5	4	5	6	5[a]	4	5	6	5	4	5	6

Sales	2	1	2	3	2	2	3	4	5	2	4	5	4	2.25	4	4
Tests	5	4	5	5	5	3	4	5	4	2	4	5	5	3	4	5
Experience	5	4	5	5.5	5	4	5	6	6	4	5	6	6	4.25	5	6
Knowledge	5	4	5	6	5	4	5	6	6	5	5	6	6	4	5	6
Problem Solved	5	4	5	6	5	4	5	6	5	4	5	6	5	4	5	6
Bug Solved	5	4	5	5	3	2	3	4	2	2	3	4	3	2	3	4
Client Satisfaction	5	3	4	5	5	4	5	5	5	4	5	6	5	5	5	6
Functionality	5	4	5	6	5	4	5	6	5[a]	4	5	6	6	4	5	6

a. Multiple modes exist. The smallest value is shown

Table 6. Kruskal Wallis test results (grouping variable: job)

		Chi-Square	*Asymp. Sig.*
Inputs	*Time*	14.752	0.002
	Knowledge	6.29	0.098
	Planning	29.05	0.000
	Estimation	34.986	0.000
	Allocated Goals	24.825	0.000
	Software	44.532	0.000
	Hardware	28.951	0.000
	Working Facilities	7.075	0.070
	Requirements Specification	19.427	0.000
	Functional Knowledge	11.019	0.012
	Client	41.051	0.000
	Motivation	9.121	0.028
	Documentation	4.8	0.187
	Experience	3.906	0.272
	Education	4.43	0.219
	Previous Source Code	62.451	0.000
		Chi-Square	*Asymp. Sig.*
Outputs	*Source Code*	106.906	0.000
	Product	22.771	0.000
	Documentation	25.846	0.000
	Finished Task	5.968	0.113
	Goal Committed	8.617	0.035
	Estimation	37.112	0.000
	Planning	55.848	0.000
	Quality	4.587	0.205
	Sales	35.913	0.000
	Tests	23.435	0.000
	Experience	6.256	0.100
	Knowledge	5.2	0.158
	Problem Solved	5.572	0.134
	Bug Solved	58.192	0.000
	Client Satisfaction	28.105	0.000
	Functionality	3.522	0.318

Table 7. Theoretical differences (Dunn test)

	Analyst	*Consultant*	*Project Manager*
Programmer	35.81530462	40.2382638	41.326569
Analyst		42.3560672	43.3912878
Consultant			52.3508412

Table 8. Input mean ranks and observed differences (with p < 0.05 in Kruskal Wallis)

	Mean Rank		**Observed differences**		
Item	**Job**	Mean Rank	Analyst	Consultant	Project Manager
Time	Programmer	159.67	0.76	28.40	**47.06***
	Analyst	158.92		29.16	**47.82***
	Consultant	188.08			18.66
	Project Manager	206.73			
Planning	Programmer	147.69	31.32	15.13	**79.57***

	Analyst	179.01		16.19	**48.25***
	Consultant	162.82			**64.44***
	Project Manager	227.26			
Estimation	Programmer	140.85	**40.45***	29.89	**88.41***
	Analyst	181.31		10.56	**47.96***
	Consultant	170.75			**58.52***
	Project Manager	229.27			
Allocated Goals	Programmer	150.73	20.42	19.72	**74.34***
	Analyst	171.15		0.70	**53.92***
	Consultant	170.45			**54.62***
	Project Manager	225.08			
Software	Programmer	218.53	**69.03***	**76.27***	**69.85***
	Analyst	149.49		7.24	0.82
	Consultant	142.25			6.42
	Project Manager	148.68			
Hardware	Programmer	209.18	**50.11***	**51.75***	**72.66***
	Analyst	159.08		1.65	22.55
	Consultant	157.43			20.91
	Project Manager	136.53			
Requirements Specification	Programmer	157.02	**44.70***	11.88	34.00
	Analyst	201.72		**56.58***	10.70
	Consultant	145.14			45.88
	Project Manager	191.02			
Functional Knowledge	Programmer	153.57	**42.66***	23.09	19.17
	Analyst	196.23		19.57	23.50
	Consultant	176.66			3.93
	Project Manager	172.73			
Client	Programmer	132.28	**46.73***	**86.15***	**66.80***
	Analyst	179.02		39.41	20.07
	Consultant	218.43			19.35
	Project Manager	199.08			
Motivation	Programmer	160.33	1.96	34.43	32.46
	Analyst	162.29		32.47	30.49
	Consultant	194.76			1.98
	Project Manager	192.78			
Previous Source Code	Programmer	224.73	**60.71***	**95.67***	**97.68***
	Analyst	164.02		34.96	36.97
	Consultant	129.06			2.01
	Project Manager	127.05			

*** There is a significance difference between these groups**

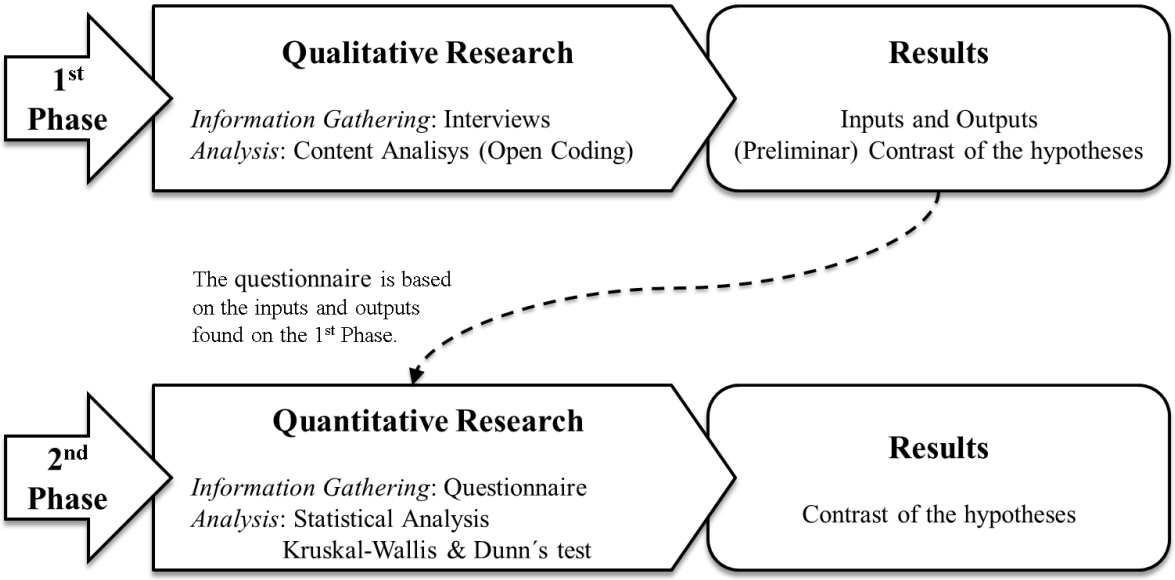
Table 9. Outputs mean ranks and observed differences (with p < 0.05 in Kruskal Wallis)

	Ranks		**Observed differences**		
Item	**Job**	Mean Rank	Analyst	Consultant	Project Manager
Source Code	Programmer	244.85	**102.68***	**120.22***	**120.31***
	Analyst	142.16		17.53	17.62

	Consultant	124.63			0.09
	Project Manager	124.54			
Product	Programmer	203.67	**41.89***	**66.80***	37.68
	Analyst	161.78		24.91	4.21
	Consultant	136.88			29.11
	Project Manager	165.99			
Documentation	Programmer	139.54	**62.42***	**48.92***	40.57
	Analyst	201.96		13.50	21.85
	Consultant	188.46			8.35
	Project Manager	180.11			
Goal Committed	Programmer	157.94	15.46	17.21	**43.47***
	Analyst	173.40		1.75	28.01
	Consultant	175.15			26.25
	Project Manager	201.41			
Estimation	Programmer	136.19	**51.99***	38.80	**87.31***
	Analyst	188.18		13.19	35.32
	Consultant	174.99			48.51
	Project Manager	223.50			
Planning	Programmer	129.21	**63.99***	**40.86***	**106.20***
	Analyst	193.20		23.12	42.21
	Consultant	170.08			**65.33***
	Project Manager	235.41			
Sales	Programmer	135.16	**40.00***	**73.10***	**75.06***
	Analyst	175.16		33.10	35.06
	Consultant	208.26			1.96
	Project Manager	210.22			
Tests	Programmer	202.54	35.40	**69.74***	38.29
	Analyst	167.15		34.34	2.89
	Consultant	132.81			31.45
	Project Manager	164.26			
Bug Solved	Programmer	225.38	**73.11***	**96.29***	**81.14***
	Analyst	152.28		23.19	8.04
	Consultant	129.09			15.15
	Project Manager	144.24			
Client Satisfaction	Programmer	143.14	25.10	**58.18***	**68.92***
	Analyst	168.24		33.08	**43.82***
	Consultant	201.32			10.74
	Project Manager	212.06			

*** There is a significance difference between these groups**

Figure 1. fff



[i] http://goo.gl/WV58vg
[ii] http://goo.gl/5tt9lB