Università degli Studi dell'Insubria

Dipartimento di Scienze Teoriche ed Applicate

PhD dissertation

Data analysis in software engineering: an approach to incremental data-driven effort estimation

Student:

Valentina Lenarduzzi

Advisor:

Prof. Sandro Morasca

Dr. Davide Taibi

Reviewer :

Prof. Alberto Sillitti

2014

EXECUTIVE SUMMARY

Cost and effort estimation in software projects have been investigated for several years. Nonetheless, compared to other engineering fields, there is still a large number of projects that fail into different phases due to prediction errors. On average, large IT projects run 45 percent over budget and seven percent over time, while delivering 56 percent less value than predicted.

Several effort estimation models have been defined in the past, mainly based on user experience or on data collected in previous projects, but no studies support an incremental effort estimation and tracking. Iterative development techniques, and in particular Agile techniques, partially support the incremental effort estimation, but due to the complexity of the estimation, the total effort always tend to be higher than expected.

Therefore, this work focuses on defining an adequate incremental and data driven estimation model so as to support developers and project managers to keep track of the remaining effort incrementally. The result of this work is a set of estimation models for effort estimation, based on a set of context factors, such as the domain of application developed, size of the project team and other characteristics. Moreover, in this work we do not aim at defining a model with generic parameters to be applied in similar context, but we define a mathematical approach so as to customize the model for each development team.

The first step of this work focused on analysis of the existing estimation models and collection of evidence on the accuracy of each model. We then defined our approach based on Ordinary Least Squares regression analysis (OLS)so as to investigate the existence of a correlation between the actual effort and other characteristics. While building the OLS models we analyzed the data set and removed the outliers to prevent them from unduly influencing the

OLS regression lines obtained. In order to validate the result we apply a 10-fold cross-validation assessing the accuracy of the results in terms of R^2, MRE and MdMRE. The model has been applied to two different case studies. First, we analyzed a large number of projects developed by means of the waterfall process. Then, we analyzed an Agile process, so as to understand if the developed model is also applicable to agile methodologies.

In the first case study we want to understand if we can define an effort estimation model to predict the effort of the next development phase based on the effort already spent. For this reason, we investigated if it is possible to use:

- the effort of one phase for estimating the effort of the next development phase
- the effort of one phase for estimating the remaining project effort
- the effort spent up to a development phase to estimate its effort
- the effort spent up to a development phase to estimate the remaining project effort

Then, we investigated if the prediction accuracy can be improved considering other common context factors such as project domain, development language, development platform, development process, programming language and number of Function Points.

We analyzed projects collected in the ISBSG dataset and, considering the different context factors available, we run a total of 4500 analysis, to understand which are the more suitable factors to be applied in a specific context. The results of this first case study show a set of statistically significant correlations between: (1) the effort spent in one phase and the effort spent in the following one; (2) the effort spent in a phase and the remaining effort; (3) the cumulative effort up to the current phase and the remaining effort. However, the results also show that these estimation models come with different degrees of goodness of fit. Finally, including further information, such as the functional size, does not significantly improve estimation quality.

In the second case study, a project developed with an agile methodology (SCRUM) has been analyzed. In this case, we want to understand if is possible to use our estimation approach, so as to help developers to increase the accuracy of the expert based estimation.

SCRUM, effort estimations are carried out at the beginning of each sprint, usually based on story points. The usage of functional size measures, specifically selected for the type of application and development conditions, is expected to allow for more accurate effort estimates. The goal of the work presented here is to verify this hypothesis, based on

experimental data. The association of story measures to actual effort and the accuracy of the resulting effort model is evaluated.
The study shows that developers' estimation is more accurate than those based on functional measurement. In conclusion, our study shows that, easy to collect functional measures do not help developers in improving the accuracy of the effort estimation in Moonlight SCRUM.

These models derived in our work can be used by project managers and developers that need to estimate or control the project effort in a development process.
These models can also be used by the developers to track their performances and understand the reasons of effort estimation errors.
Finally the model help project managers to react as soon as possible and reduce project failures due to estimation errors.
The detailed results are reported in the next sections as follows:

- Chapter 1 reports the introduction to this work
- Chapter 2 reports the related literature review on effort estimation techniques
- Chapter 3 reports the proposed effort estimation approach
- Chapter 4 describe the application of our approach to Waterfall process
- Chapter 5 describe the application of our approach to SCRUM
- Chapter 6 reports the conclusion and the future works

TABLE OF CONTENT

CHAPTER 1 INTRODUCTION

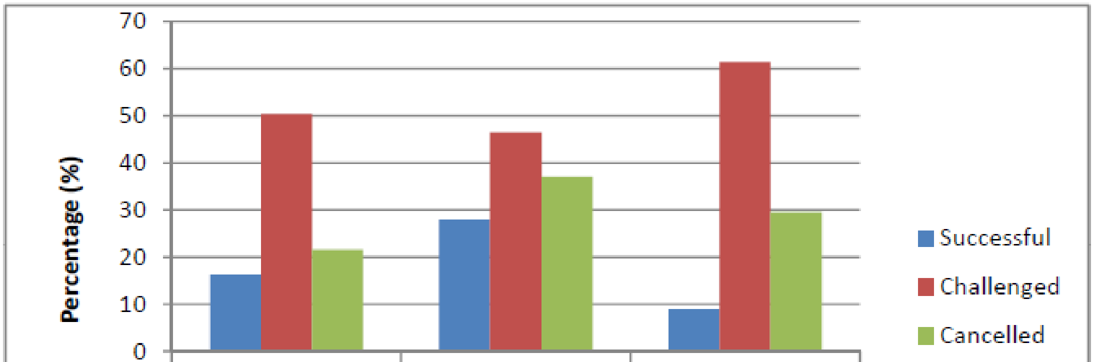
Effort estimation is one of the most important activities in any engineering domain but, unlike in most engineering domains, such as building and mechanical engineering, effort estimation in software engineering is still largely an open issue.

Inaccurate estimations can contribute to the failure of a project. While an overestimation could drive customers to accept bids from other companies, an underestimation can lead to several issues such as project failure due to lack of budget to complete the project or, in some case, to the failure of the company itself.

Since software engineering is a relatively new discipline, several techniques have been developed for estimating effort, but none has yet been deemed satisfactory enough to be widely used in industry. Development technologies and paradigms change rapidly and software engineers must keep updating their technological knowledge and also need to understand how to estimate the costs and effort for new technologies

Software projects fail because of several reasons, such as costs, scheduling and quality issues. These failures cause huge losses in time and money and can establish negative effects to company's growth and development. The causes are typically discovered very late when it is no longer possible to change direction. [36]

To give an idea of the impact of project failures, in Figure 1 we report the results of the CHAOS report [35], which analyzed the failure causes of projects in 2014. This result is also confirmed by a Gartner report[14] that shows that runaway budget costs are the reason of one quarter of project (150 projects analyzed) failures and 45% of those failed for errors in effort estimation as shown in Figure 2. According to Gartner [14] one-quarter of small project in term of size fail for runaway budget costs. In fact the failure rate of big projects is almost 50% higher than for projects with low budgets as reported in Figure 3.



	Small IT projects	Medium IT projects	Large IT projects
Successful	16,2	28	9
Challenged	50,4	46,5	61,5
Cancelled	21,6	37,1	29,5

Figure 1: project failure causes [35]

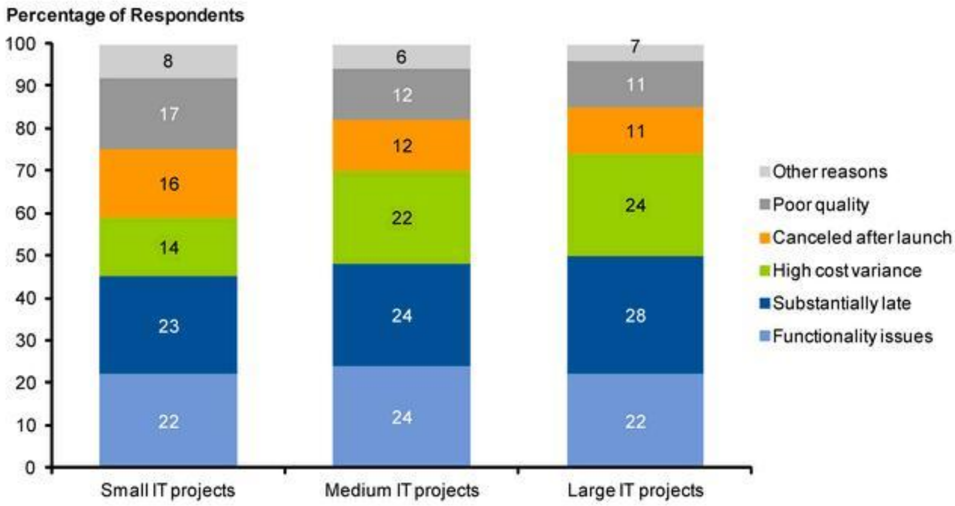


Figure 2: Reasons of projects failures [36]

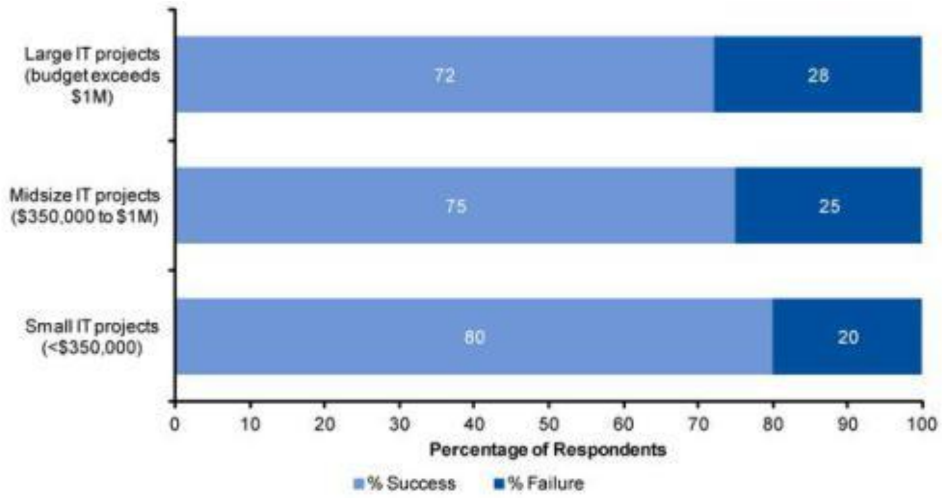


Figure 3: Distribution of success and failure across project size [35]

Other studies report on the reasons and on the distribution of successful and failed projects.

A study carried out by Molokken and Jorgensen [2] found that 60-80% of the projects are completed over budget and that 30%-40% of project plans are based on over-optimistic effort estimates. Moreover, the increase in term of project size leads to overruns estimates between 30-40%.

Moreover, according to Phan [37], cost overruns are related to over-optimistic estimates (51%), closely followed by changes in design or implementation (50%) and optimistic planning (44%), followed by frequent major (36%) and minor (33%) changes in the specifications. The main reason is that is usually hard to keep track of effort status, based on the effort estimated before the project.

1.1 The approach

In our work, we want to improve the effort estimation quality during the development process, by defining a lightweight iterative model for effort estimation during all development phases.

Our proposal can be used to predict and monitor project effort during ongoing projects for the next development phase or for the rest of the project. Our approach will help project managers react as soon as possible and reduce project failures due to estimation errors.

The research process adopted in this dissertation is organized in three main steps as described in Figure 4.

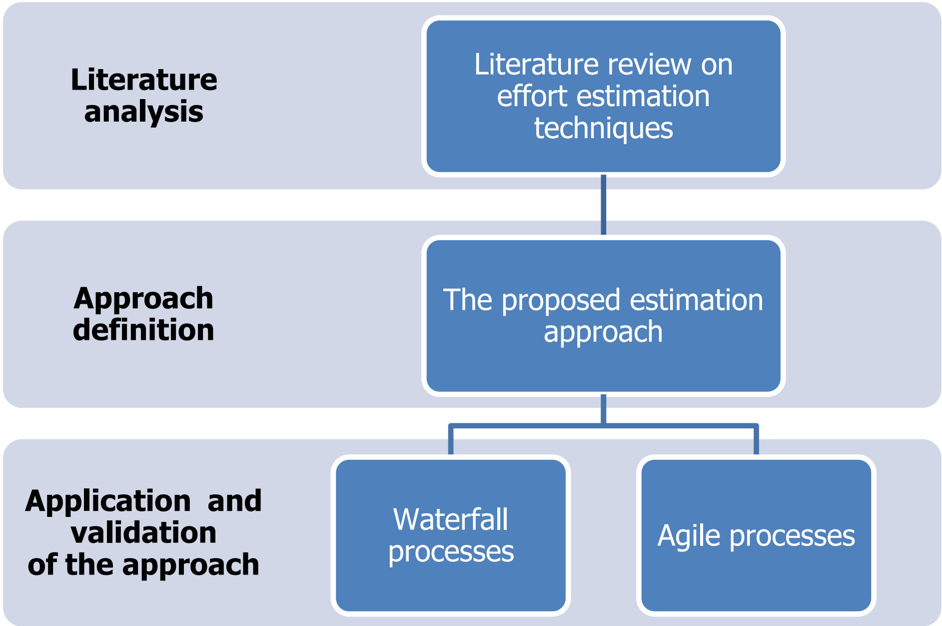


Figure 4: The approach

The first step focuses on the literature review on effort estimation techniques where we investigated the most relevant and used effort estimation models.

In the second step we describe the approach adopted for effort estimation. In the last step we applied our approach to two different development process: Waterfall and Agile life cycles and we describe all of the steps we carried out.

1.2 Document Structure

This dissertation is structured as follow.

- Chapter 1: introduction on the problem
- Chapter 2: related literature review on effort estimation
- Chapter 3: the proposed approach for effort estimation,
- Chapter 4: the application and validation of the proposed model description
- Chapter 5: conclusions and future work.

CHAPTER 2 EFFORT ESTIMATION TECHNIQUES

Software effort estimation is the process of estimating effort as accurately as possible for all development phases. Effort is expressed in terms of person-hours.

Software effort estimation is a complex and critical [27] task. A good deal of information should be taken into account to estimate total effort, such as project size, domain, and many other factors that may significantly influence the estimation. [28]

Analogy is one of the simple estimation techniques. A project's effort is estimated based on the effort of similar ones. However, but, estimation errors usually occur, since the development process is usually unique and hardly repeatable. Moreover, the measurement of parameters that could influence the effort is very complex, because software products are much less tangible than the products of classical engineering. In addition, the continuous change in requirements does not help estimation accuracy. [28]

For this reason, the research in effort estimation mainly focuses on improving the accuracy of the existing models.

Effort estimation models can be grouped in three main categories, based on the quantity of human expertise and historical data collected in previous projects: expert-based models (Section 2.1), hybrid models (Section 2.2), and data-driven models (Section 2.3), as shown in Figure 5.

Data-driven models can be applied only in those companies that collected historical quantitative data. The higher the influence of the quantitative data on effort estimation, the lower the needed human expertise.

In this chapter, we present some of the most common effort estimation techniques, describing strength and weaknesses and highlighting their differences with our approach.

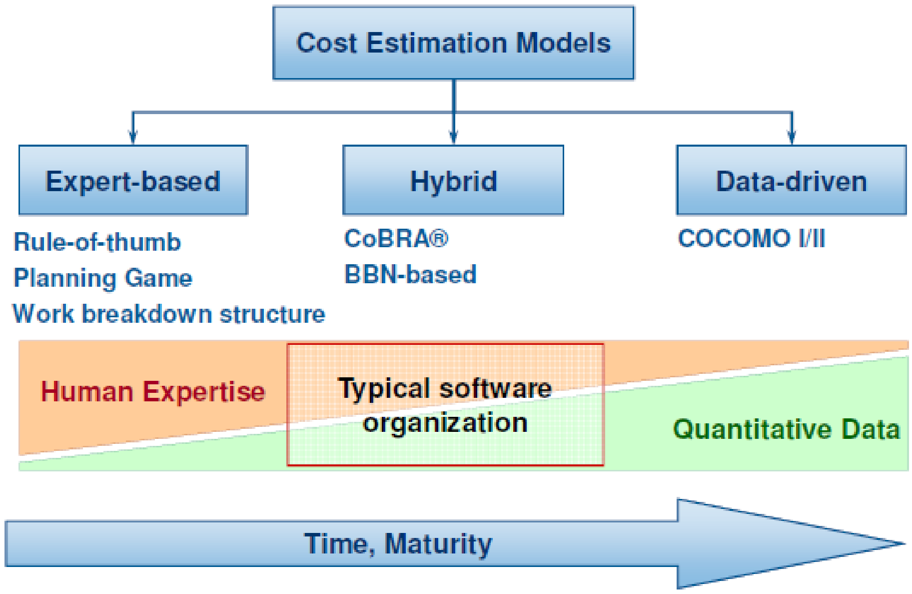


Figure 5: A classification of effort estimation models

2.1 Expert-Based Models

Expert-based estimation models are mainly based on people's experience. They are the oldest and most common effort estimation techniques, applied in any domain. Estimation is commonly carried out by analogy, comparing the project to be developed to similar projects, carried out in the past.

A simplified example of effort estimation can be the calculation of the taxi fare, from point A to point B. An experienced taxi driver could approximately estimate the cost of the trip, based on his experience of driving the same path at the same time of the day. The same estimation could be done with a data-driven approach. The taxi company can estimate the cost, based on the average of cost of similar trips at the same time, including information related to the actual traffic, temporary deviations, accidents and weather conditions, thus obtaining a result similar to the one obtained with the experience based estimation, carried out by the driver.

A survey carried out by Trendowicz et al. [26] reports that the large majority of software organizations adopt expert based models. This result is also confirmed by a study carried out by Molokken [2]. Moreover, a systematic literature review published by Molokken [3] shows different viewpoints: some articles report that some publication recommend expert-based effort estimation; some recommend the usage of data-driven models, while others are not able to identify which approach is better. Studies [2] and [3] also show that the improvement of software effort estimation does not necessarily require the introduction of sophisticated formal estimation models or expensive project experience databases. In this section we describe two of the most used expert based models: the planning game and the work breakdown structure.

2.1.1 The Planning Poker

The term Planning Poker was introduced by Grenning in 2002 [38] in agile software development. Planning Poker represents expert-based estimation with a structured group approach. The method originates from agile software development for providing a lightweight approach to estimating software development interactions. The model estimates the functional size and effort from historical data on development productivity. [36]

In Agile and especially in Extreme Programming (XP), Planning Poker is the common estimation technique.

In XP, development is structured as a set of iterations (sprints), where developers and customers elicit requirements and plan the next development steps. Requirements are then grouped in user stories, which are units of software functionalities that are understandable from customers, users and developers. A single user story is small enough to be developed in a single sprint and needs to be testable, based on a set of acceptance test agreed with the customer. The whole list of requirements, collected as user stories, is then stored in the so-called Product Backlog, a list of all product features required by the customers.

Since in XP requirements are defined iteratively, and the whole set of requirements is not available at the beginning of the project, classical effort estimation techniques are not applicable.

During the Planning Poker developers are required to estimate the effort of the user stories that will be implemented in the next sprint. Effort is usually assessed via “story points,” a number that ranges from 1 to 5 based on the complexity of the requirement. Story points are believed to be related to effort and complexity. Several iterations of contacts between developers and customers are needed to adequately estimate each story. Story points are determined on the basis of the Fibonacci number sequence, in the series 0, 1, 1, 2, 3, 5, 8, 13, and so on.

User story size estimation in planning poker is done by analogy, based on the similarity to the size of similar user stories already implemented in the past or already estimated in the same estimation session. Effort per story unit is called velocity and represents the development productivity of an agile team.

The estimated story point size is used for further project planning. In agile development, there are two levels of planning: iteration planning and release planning, where user stories and their estimated size are two of the inputs for project planning.

To the best of our knowledge, no studies report on the accuracy of the estimation carried out by means of the Planning poker approach or comparing the estimation power of Planning poker with other methods.

2.1.2 Work breakdown structure

The Work Breakdown Structure (WBS) is an effort estimation technique based on the decomposition of the project in several sub-components, whose related efforts are easier to estimate by a set of experts [25]. The WBS, as defined in the PMBOK® Guide [29], is a "deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables." WBS helps initiate, plan, execute, monitor and control processes used to manage projects. Figure 6 shows an example WBS hierarchy.

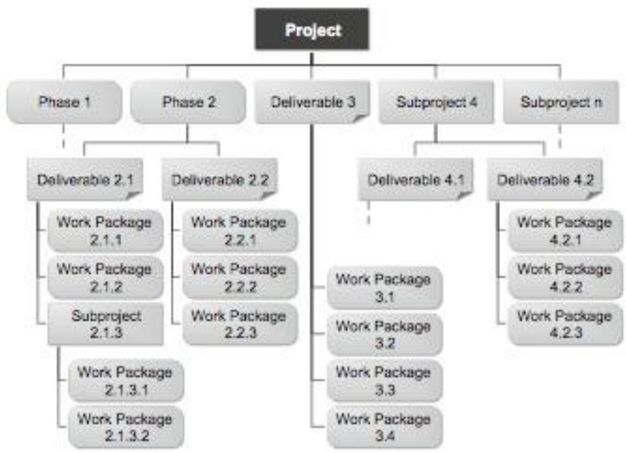


Figure 6: Work breakdown hierarchy

Since WBS decomposes projects in small components, estimating the effort for each component is usually based on the experience of experts who evaluate the required effort by analogy, by comparing the component to be developed to similar components they developed in the past. The sum of the sub-components includes 100% of the work to be carried out in the project, including project management.

The errors introduced by the expert estimation based on the WBS are due to errors in the project decomposition, such as missing components, and to errors in the expert based estimation of the single component.

2.2 Hybrid Models

The importance of the information coming from experts is taken into account in a sub-category of data-driven models, so-called “hybrid models,” which combine experience from experts and statistical techniques. Some relevant examples of hybrid models are CoBRA [10] and the BBN-Based model [11] [12].

Compared to our proposal, hybrid models are usually less accurate, even though they consider human expertise together with historical data.

2.2.1 CoBRA

CoBRA® (COst estimation, Benchmarking, and Risk Assessment) is a hybrid method defined by Briand et al. in 1998, that combines data and expert-based cost estimation approaches. [24] The model takes into account the most relevant constraints and capabilities of software engineering contexts and has the capability of combining insufficient measurement data with human expertise into an intuitive graphical effort model. Thanks to the lower requirements it sets on available measurement data, its capability of utilizing humans expertise, and a simple theoretical approach, CoBRA is an attractive software estimation process.

In CoBRA®, the development effort is calculated based on three basic components, as shown in Figure 7: nominal effort ,effort overhead and nominal productivity.

Nominal effort is the engineering and management effort spent on developing a software product of a certain size in the context of a nominal project, which is a hypothetical “ideal” project in a certain environment of a business unit. The value of nominal effort is based on data from similar historical projects about some characteristics such as development process or life cycle type. While such characteristics define the context of the project, the past project data determine the relationship between effort overhead and effort (see equation 2).

The effort overhead is the extra effort spent in addition to the nominal effort and quantified as the percentage of additional effort over nominal one due to the problems of the real project environment, such as skill lacks of the project team.

Nominal productivity (PNom) is the development productivity under optimal project conditions and it is related to the ratio between a project’s output and input. Development productivity is obtained from the ratio between the size of delivered software products and the effort consumed to develop these products as described in (1) and (2),

$$\text{Effort} = \text{Nominal Effort} + \text{Effort Overhead} \quad (1)$$

$$\text{Nominal Effort} = \text{Nominal Productivity} \cdot \text{Size} \quad (2)$$

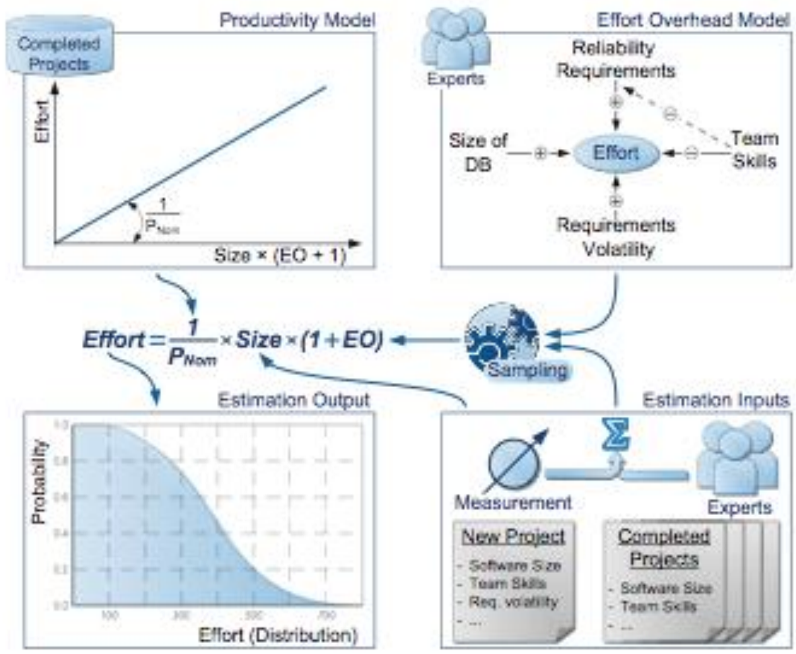


Figure 7: the CoBRA model approach

Compared to our model, CoBRA® relies mostly on software size in terms of lines of code or Function Points, while we provide much more flexibility that allows the users to select different project dimension. Moreover CoBRA® does not help keep track on the effort status or to predict the effort for each development phase.

2.2.2 BBN-Based Model

Bayesian Networks (BNs) is applied in effort estimation [30] for different types of projects, including web applications [31]. The model makes it possible to combine expert judgment for a flexible and informative estimation [32].

Such model presents rigorous mathematical aspects and, at the same time, is easy to understand. Also, BBNs allow a probabilistic mechanisms for representing uncertain information.

BBN models can be represented by Directed Acyclic Graphs (DAGs) composed of causal networks. Figure 8 shows an example of a BBN for estimating software effort.

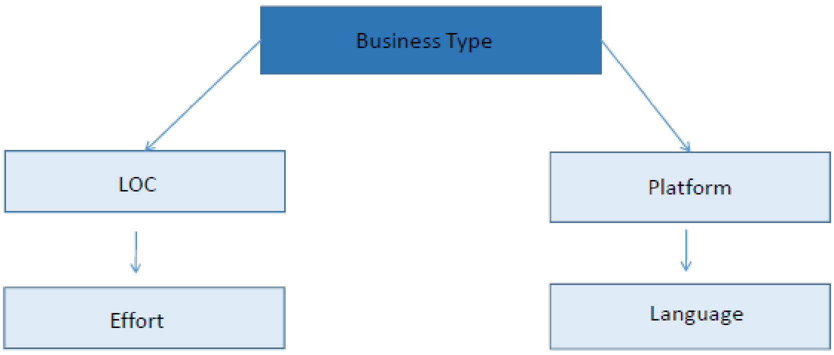


Figure 8: Bayesian Belief Network

In the example in Figure 8, we introduce four variables (business type of the application, the programming language, the development platform used and the lines of code (LOC)) and indicate the relation among them:.

Business type influences directly the development platform used and the size measured in LOC. The total effort required for the project is directly influenced by the size of the application. Consequently, the estimated effort depends on the value of LOC. If the LOC value is unknown, it is possible predict indirectly the value of effort from the value of business type predicting LOC and so the effort.

For each node, a node probability table (NPT) specify the values that the variable can assume, each predecessor nodes. A possible NPT is represented in Table 1.

LOC	≤12105	>12105
≤ 5.5	0.7	0.3
>5.5	0.3	0.7

Table 1: node probability table (NPT)

In this example, a project that is relatively small in size (≥12105 LOC) presents 70% probability of being in the low effort (≤5.5) interval and 30% possibility of belonging to the high effort interval (>5.5 months).

A recent study carried out by Shepperd and Macdonell [33] investigated the validity of the BBN method highlighting the low accuracy of this technique, reporting an MMRE that ranges from 900% to 90%,.

2.3 Data-Driven Models

Data-driven models are based on statistical or machine-learning approaches, with the goals of (1) reducing the amount of subjectivity inherently related to expert-based estimation and (2) automate the effort estimation as much as possible, thereby reducing the cost related to estimation itself.

Several studies have investigated the accuracy of effort estimation using modeling techniques such as ordinary least square regression and analogy-based estimation [5, 6, 7].

One of the most important data-driven estimation models based on regressions is the Constructive Cost Model in both its original (COCOMO 81) and second (COCOMO II) versions, which we now describe.

2.3.1 COCOMO 81

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm 1981 [34]. The model is based on a basic regression formula with parameters obtained from historical project data . It is based on and applicable to the Waterfall development process only.

COCOMO is composed of three levels:

- **Basic**
- **Intermediate**
- **Detailed**

Three different classes of software projects exist for each level, defined as

- **Organic:** the project size must be small, the team must have more experience in the project domain.
- **Embedded:** the project must be big, the team does not have more experience in the project domain.
- **Semi-detached:** this level is between the organic and the embedded.

Basic COCOMO

In the Basic COCOMO model, effort is a function of size, expressed in estimated thousand delivered source instructions (KDSI) :

$$\text{development effort (MM)} = a * \text{KDSI}^{\,b} \quad (3)$$

where MM (Man Months) is the total effort expressed in person months, considering a monthly effort of 152 hours per person month.
The coefficients a and b , defined in (3), depend on the different classes of software projects.
The coefficient for Basic COCOMO are reported in Table 2.

Intermediate COCOMO

This model also takes into account the cost drivers, so its accuracy is better than that of Basic COCOMO. In the effort estimation formula, a new factor is considered, called Effort Adjustment Factor (EAF), obtained by multiplying the values given to of fifteen cost drivers rated on a scale from “very low” to “very high.” The adjustment factor is 1 for a cost driver that is defined as normal. Development effort is then calculated as in (5):

$$\text{development effort (MM)} = a * KDSI^{b} + EAF (5)$$

The coefficient for Intermediate COCOMO are reported in Table 2

	Basic		Intermediate	
	a	b	a	b
Organic	2.4	1.05	3.2	1.05
Semi-detached	3.0	1.12	3.0	1.12
Embedded	3.6	1.20	2.8	1.20

Table 2: basic and intermediate COCOMO coefficients

The fifteen cost drivers EAF vary from 0.9 to 1.40, as reported in Table **3**.

EAF	**Rating**					
	Very low	**Low**	**Nominal**	**High**	**Very high**	**Extra high**
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.4	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Table 3: cost drivers EAF for intermediate COCOMO

Detailed COCOMO

In this level, the project size and the Cost Drivers are weighted according to account the influence of the development project phases. Advanced COCOMO model adopts the Intermediate model for the component level as defined in (5):

$$\text{development effort (MM)} = a * KDSI^{b} * EAF \quad (5)$$

Detailed COCOMO divides the development process in four phases, based on which, it identifies the different EAF coefficient, as show in Table 4 :

- requirements planning and product design (RPD)
- detailed design (DD)
- code and unit test (CUT)
- integration and test (IT)

	Rating	RPD	DD	CUT	IT
EAF	Very Low	1.80	1.35	1.35	1.50
	Low	0.85	0.85	0.85	1.20
	Nominal	1.00	1.00	1.00	1.00
	High	0.75	0.90	0.90	0.85
	Very High	0.55	0.75	0.75	0.70

Table 4: EAF coefficients for Detailed COCOMO

2.3.2 COCOMO II

COCOMO II was developed in 1995 and published in 2000 [34] as evolution of COCOMO 81 and it is applicable to different development processes and not only to the Waterfall one like COCOMO 81. Also, it provides more accurate results during the effort estimation process.

COCOMO II is composed of four models:

- Application composition model
- Early design model
- Post-architecture model
- Reuse model

Early Design model: the effort estimation is based on Function Points as Functional size measurement [34]. Function Points are defined by measuring the product functionality in terms of data and process.

This model is used in the early stages of the development process and when the requirements are defined without knowing the size of the product to be developed, the nature of the target platform, the nature of the personnel to be involved in the project, or the detailed specifics of the process to be used. It is often used for comparing different planning solution

$$\text{development effort (MM)} = a * KDSI\ b + EM \quad (7)$$

$$EM = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED \quad (8)$$

Coefficient a is equal to 2.94 and coefficient b ranges from 1.10 to 1.24 and depends on several factors such as flexibility, project innovation, risk management and process maturity. Size is measured in thousands lines of code obtained from the Function Point using a conversion table. The factors of the EM formula are:

- PERS: personnel capability
- RCPX: Product reliability and complexity
- RUSE: Required reuse
- PDIF: Platform difficulty
- PREX: Personnel experience
- FCIL: Facilities
- SCED: Schedule

Each factor above is evaluated based on a scale from 1 (very low) to 6 (very high).

The early design model in COCOMO 81 is an approach closely related to our study. Specifically, it provides the following effort distribution across product development phases: 60% for analysis and design, 15% for programming, and 25% for integration and test activities. This distribution somewhat disagrees with the well-known rule of ‘40/20/40’ [8]. Unlike in COCOMO, in our approach we do not only define ratios but also suggest how to calculate these ratios based on company projects.

Application composition model: it is a model used for estimating the effort needed for prototyping or for building software from some existing components.

$$\text{development effort (MM)} = (NAP \times (1 – \%reuse/100)) / PROD \quad (9)$$

where NAP is the number of Application Point and PROD is the productivity as reported in Table 5.

Rating	Value				
Developers experience and capability	Very low	Low	Nominal	High	Very high
Maturity and cabality of CASE tools	Very low	Low	Nominal	High	Very high
Productivity (prod/month)	4	7	13	25	50

Table 5: productivity value identification

Reuse model: such model considers both code reuse without changes (black box) and with changes for integrating with the new code (white box).

In the black box case, the model estimates the total effort spent as in the Early design model, while in the white box case the number of new lines of code are estimated from the number of reused ones. Such values are integrated with the new ones, as defined in (10):

$$\text{development effort (MM)} = (\text{ASLOC} * \text{AT}/100)/\ \text{ATPROD} \quad (10)$$

where ASLOC is the total number of line of code, AT is the percentage of automatically generated code and ATPROD is the developers productivity for the code integration.

The code reused and modified is estimated as defined in (11):

$$\text{ESLOC} = \text{ASLOC} * (1-\text{AT}/100) * \text{AAM} \quad (11)$$

where ESLOC is the number of lines of new code , ASLOC is the number of lines of reused code to be adapted, AT is the percentage of automatically generated code and AAM is the coefficient for the code adapting difficulty.

Post-architecture model: once the project is ready to develop and sustain a system it should have a life-cycle architecture, which provides more accurate information on cost driver inputs, and enables more accurate cost estimates.

The formula is the same as in the Early design model.

Size is determined as the sum of three components:

- Number of lines of code to be developed
- Number of lines of code calculated by the reuse model
- Number of lines of code to be adapted to the application requirements

In this case, coefficient b depends on five factors (with values on a 0 to 5 scale) while in the Early design model the factors are three.

Coefficient M depends on 17 factors related to Product attributes, Hardware attributes, Personnel attributes, and Project attributes.

As for COCOMO 81, also COCOMO II provides a distribution of effort for each development phase, as shown in Table 6.

Phase	Effort %
Plan and Requirement	7
Product design	17
Programming	64-52
• Detailed design	27-23
• Code and Unit Test	37-29
Integration and Test	19-31

Table 6: COCOMO II effort phase distribution [34]

2.4 Existing approaches to estimate effort in project phases

In this section, we introduce the most used and relevant model for estimating the effort in project phases. Such approaches are in general complex to use because they need several parameters that may be quite difficult to identify and adapt in every case. Moreover there are no studies to estimate the remaining effort of an ongoing project.

Few other works have investigated the distribution and prediction of effort among phases. MacDonnell et al. [20] studied the relationships of the efforts spent in each phase in 16 projects developed in the same organization, finding that there is no correlation of effort in project phases.

Jiang et al. [21] proposed a model for predicting development effort based on the software size estimated with Function Points. They computed the average amount of effort spent on each phase, based on the ISBSG R9 dataset [49], and obtained the following effort distribution: 7.2% for planning, 15.9% for specification, 12.9% for design, 37.8% for building, 17.6% for testing, and 8.6% for the deployment phases. A detailed description of each phase is provided in Section IIIB.

Another work [18] evaluated the effort distributions of two projects developed according to the Rational Unified Process. The goal of the paper was to carry out a post-mortem analysis to help project managers in future projects. Due to the low number of projects analyzed, no correlations were found among the efforts in the development phases, but the graphical visualization of effort per phase was found useful from the project managers' point of view.

Yang et al. [19] compared the effort distribution obtained from 75 projects from different Chinese software organizations with the COCOMO effort distribution to understand variations and possible causes of effort distribution. They identified the development lifecycle, development type, software size, and team size as the main influencing factors that cause variations in effort distribution.

Chatzipetrou et al. compared the effort distribution in ISBSG R11 to lifecycles activities, organization type, programming language, and function points, investigating one project phase at a time [22]. The main goal was the application of the Compositional Data Analysis (CoDA) technique. They proved that the technique is effective for graphically representing correlations. Moreover, they identified organization type as the main factor that differentiates the levels of effort distributed across each project phase.

CHAPTER 3 THE PROPOSED EFFORT ESTIMATION APPROACH

In this chapter we describe the effort estimation approach we used in our research, which is schematically represented in Figure 9.

The proposed approach is composed of the three steps presented in Figure 9. First, we started with data pre-processing, then we applied statistical techniques for the effort estimation phase, and finally the last step is concerns the validation of the results.

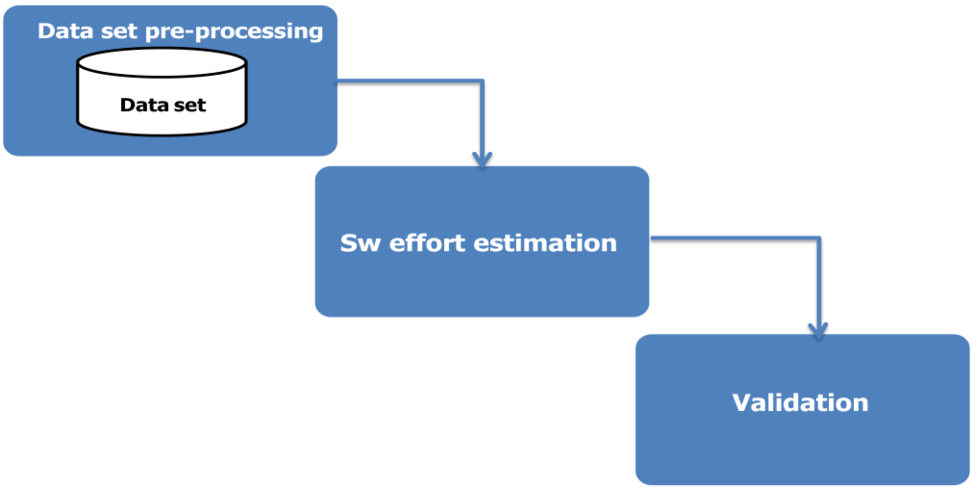


Figure 9: the proposed effort estimation approach

We process the data set identified for the software effort analysis. We investigate the data set in order to find possible attributes in terms of data frequency that can influence the effort during the analysis.

In order to better explain the process described in this section, here we define an example data set that will be used in the whole section, as shown in Table 7.

a	1	3	4	7	8	11	15	19	23	27	29
b	2.7	5	19	5	7	8.76	2	16	17	19	22

Table 7: example data set

We analyzed this data set using Ordinary Least Squares (OLS) regression with outlier elimination. OLS estimates unknown parameters in a linear regression model, by minimizing the sum of squared vertical distances between the observed responses in a real dataset and the responses predicted.

The linearity of the correlation between two variables X and Y is measured by the Pearson product-moment correlation coefficient. The value obtained is statistically significant if it is associated with a p-value less than 0.05. In the linear case, when we have one dependent and one independent variables, the resulting estimator can be expressed by a simple formula as follows:

$$Y = m*X + b$$

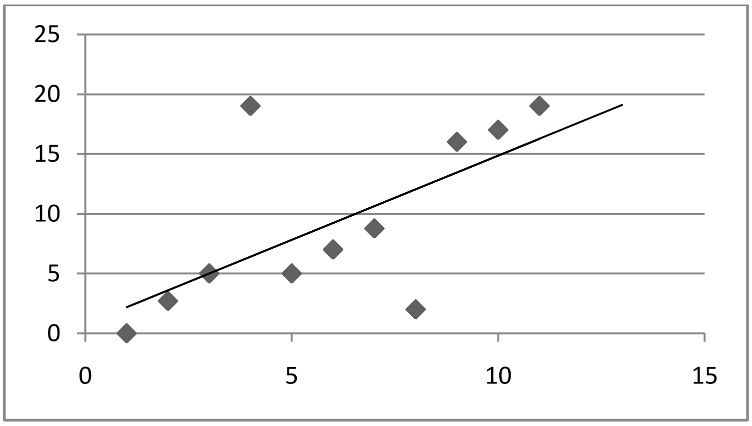


Figure 10: example of linear regression

In Figure 10: example of linear regression we show the data set in a scatter plot.

During the building of the OLS models, we analyze the data set and remove the outliers to prevent them from unduly influencing the OLS regression lines obtained. Specifically, we identify the outlier values that range more than 3 times standard deviation from the mean [14]. In the building of our models, we use a 0.05 statistical significance threshold, as customary in empirical software engineering studies

In the example in Figure 10, we remove two outliers, the data points for which a= 4 and a= 14.

After the outlier elimination we obtain a more accurate model, as reported in Figure 11. The new defined model is that allows the estimation of any value of "a" is y= 0.5x + 3.

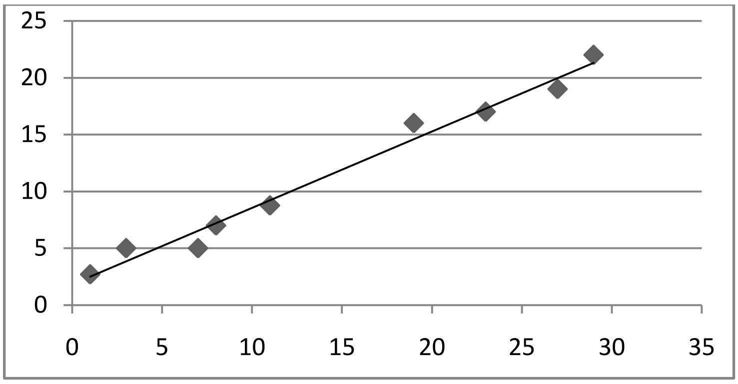


Figure 11: example of linear regression with outlier elimination

We validated the results obtained analyzing three accuracy indicators:

- R^2
- MMRE
- MdMRE

An effort estimation model must fit the data as well as possible. we consider as measures of model goodness of fit the coefficient of determination R^2 and magnitude of relative error MRE. R^2 indicates how well data fit a statistical model by providing the proportion of total variation of outcomes explained by the model. The coefficient ranges between 0 and 1: when R^2 1 the regression line perfectly fits the data and when $R^2 = 0$ the model does not provide any explanation for the data. Two of the most commonly used indicators of estimation accuracy are the Mean Magnitude of Relative Error MMRE and the Median Magnitude of Relative Error MdMRE. Both indicators are based on the Magnitude of Relative Error MRE for each estimate, defined as follow:

$$MRE = \frac{|\text{actual value} - \text{estimated value}|}{\text{actual value}}$$

where the actual effort is the effort really spent and the estimated effort is the effort obtained by the statistical analysis. A low value of MMRE and MdMRE indicate a high goodness of fit of the estimation.

a	1	3	7	8	11	19	23	27	29
b	2.7	5	5	7	8.76	16	17	19	22
b (estimated)	3.5	4.5	6.5	7	8.5	12.5	14.5	16.5	17.5
R^2	0.991								
MRE	0.30	0.10	0.30	0	0.03	0.22	0.15	0.13	0.2
MMRE	0.15								
MdMRE	0.16								

Table 8: validation example

CHAPTER 4 APPLYING THE APPROACH TO WATERFALL PROCESSES

In this chapter, we report on an empirical study we ran to investigate how to apply the model defined in Chapter 3, on a large set of projects contained in the International Software Benchmarking Standards Group (ISBSG) data set, release 11 [49].

In Section 4.1, we introduce the design of the empirical study and we describe the related research questions, development process, data set, analysis procedure and data collection and aggregation. In Section 4.2, we present the data analysis and discuss the results in Section 4.3.

4.1 Empirical study design

In this section, we specify the goal of the empirical study and we describe the design used for the study and the procedure followed for its execution.

4.1.1 Research Questions

The objective of our research is to understand if we can predict the effort of the next development phases based on the effort already spent. For this purpose, we design an empirical study. We start by investigating all the new development projects, analyzing the correlation between one phase and the next one. Since there are a variety of factors that may affect effort, we also want to understand if clustering projects by common characteristics helps obtain models that are more accurate than those obtained by using only effort data.
This leads to the following research questions:

RQ1: Is it possible to use the effort of **one phase** for estimating the effort of **the next development phase**?

RQ1.1: Does considering **other characteristics of the projects**, in addition to the effort for a phase, improve the effort prediction for the next phase in a statistically significant way?

We then investigate if, based on the effort spent in one phase, it is possible predict the effort for the remaing part of the project, as described in the second research questions:

RQ2: Is it possible to use the effort of **one phase** for estimating **the remaining project effort**?

RQ2.1: Does considering **other common caractheristics**, in addition to the effort for a phase, improve the effort prediction for the remaining project in a statistically significant way?

Next, we want to investigate if considering more than one previous development phase would improve the estimation accuracy of the next one. For this reason we investigate the following research questions:

RQ3: Is it possible to use the **effort spent up to a development phase** to estimate **its effort**?

RQ3.1: Does considering **other common caractheristics**, in addition to the effort spent up to a development phase, improve the its effort prediction in a statistically significant way?

Finally, we want to investigate if considering more than one development phase before would improve the estimation accuracy of the remaining effort up to the end of the project. For this reason we investigate the following research question:

RQ4: Is it possible to use **the effort spent up to a development phase** to estimate **the remaining project effort**?

4.1.2 The development process

We now concisely describe the waterfall development process, which is one of the most common processes of the projects represented in the ISBSG data set.

The waterfall process was the first organized process proposed in software engineering. Originated in the manufacturing and construction industries, the waterfall model is a linear-sequential life cycle model that is very simple and easy to understand and use.

The process, as shown in Figure 12, is composed as follows:

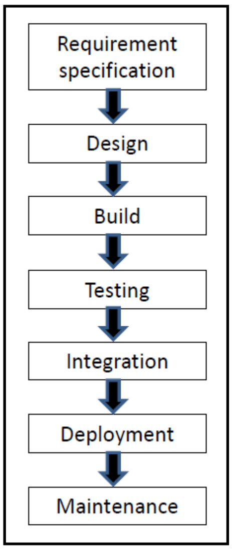


Figure 12: The waterfall process

Requirement specification: this phase includes the description of what the system does by defining the requirements, which are detailed in a requirements analysis and specification document.

Design : this phase describes the architecture of the software to be built by identifying its modules and defining the relations between them.

Build: this phase consists of the modules creation. Based on the design, the project is first developed in modules called units, which are later integrated.

Testing: each unit, developed in the Build phase, is tested for its functionality in order to find faults and failures.

Integration: after testing each unit, all units developed are integrated into a system.

Deployment: in this phase the product is deployed in the customer environment or released into the market.

Maintenance: this phase includes all the actions for corrective, adaptive, perfective and preventive maintenance.

Before beginning one phase, the previous phase must be completed. The output produced from each phase is the input for the next one. At the end of each phase the developers must documented what done and a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. Waterfall model works well for small projects, where requirements are clearly defined in the early phases.

4.1.3 Target dataset

The empirical study was carried out based on the International Software Benchmarking Standards Group (ISBSG) (release 11) data set. The data set allows ISBSG users to compare their projects for benchmarking and estimation purposes. It contains more than 5000 software projects collected worldwide from 1990 to 2006 from several business areas such as banking, financial, manufacturing, and others.

The data set contains several variables that can be useful for estimating the effort in different development phases. We now list the most important variables we consider in this work, along with their values:

Development Type:

- *New development projects:* projects developed following the complete development lifecycle from the beginning (planning / feasibility, analysis, design, construction, and deployment)
- *Enhancement projects:* changes made to existing applications where new functionality has been added, or existing functionality has been changed or deleted
- *Re-development projects*: re-development of an existing application.

Effort per development phase: This attribute contains the breakdown of the work effort reported via six categories:

- *Planning:* preliminary investigations, overall project planning, feasibility study, and cost benefit study
- *Specifications:* systems analysis, requirements, and architecture design specification
- *Design:* functional, internal, and external design
- *Building:* package selection, software coding and code review, package customization, unit testing, and software integration

- *Testing:* system, performance, acceptance testing planning and execution

- *Deployment:* release preparation for delivery, release installation for users, user documentation preparation. Note that this category is actually called “Implementation” in the ISBSG data set, but we renamed it “Deployment” here to better clarify its meaning and differentiate it from the “Building” phase in which the software code is actually written.

- *Effort unphased:* includes all projects that specify the whole effort without making distinctions among phases.

Primary Programming Language: This attribute describes the primary language used for the development. Some of the most common languages used by the projects are JAVA, C++, PL/1, Natural, Cobol.

Architecture: this attribute describe the organizational structure of a system and its implementation guidelines. The architectures used by the projects are Multi-tier, Client server, Stand alone and Multi – tier/Client server.

Development platform: This attribute describes the platform chosen for the development. Some of the most common platforms used by the projects are Multi, Main Frame (MF), PC, Mide Range(MR).

Development techniques: This attribute describes the development techniques chosen for the development. Some of the most common development techniques used by the projects are waterfall and data modeling.

Domain: This attribute describes the domain used for the development. Some of the most common domain used by the projects are banking, insurance, communication, manufacturing and public administration. Note that this category is actually called “Organization type” in the ISBSG data set.

Functional measurement approach: This attribute describes the count approach used for determining the size of the project. Some of the most common count approaches used by the projects are IFPUG, FiSMA, NESMA and COSMIC. Note that this category is actually called “Counth Approach” in the ISBSG data set.

Functional size: this attribute contain the value related to the Functional measurement approach used.

We do not consider “Effort unphased” since it does not provide any information on the phases whose effort is the main focus of our work. The ISBSG data set contains 5052 projects, 1975 of which are new developments; 2869 are enhancement projects, while the nature of 213 is not specified.

4.1.4 Analysis procedure

The data contained in ISBSG data set are analyzed by means the approach defined in Chapter 3.

Moreover, when we analyze RQ3 and RQ4, we consider univariate analysis of the sum of effort of the previous phases and the multivariate correlation, analyzing the contribute of each previous phase separately. Then we cluster each combination data for each attribute available on the database, such as development language, architecture and project domain), in order to understand if the effort estimation accuracy improves by considering more information.

For instance, we estimated the test phase effort based on the following combinations of independent variables (Table 9):

- Build plus Design effort (univariate model);
- Build and Design effort (multivariate model);
- Build plus Design plus Specification effort; (univariate model)
- Build and Design and Specification effort (multivariate model);
- ...

Combinations	Clusters
• Previous phase vs next phase • Previous phase vs remaining phases • Sum of effort up to a certain phase vs remaining phases • Previous phases vs next phase	• Dev. language • Architecture • Domain • Dev. process • Platform • Func. Approach >/< 1000

Table 9: set of analysis

4.1.5 Data collection and aggregation

The ISBSG data are preprocessed to obtain several data subsets for effort estimation. The selection is carried out in two steps and only projects containing effort values greater than or equal to zero for each phase considered are taken into account [49], i.e., we filter out projects that contain corrupt data.

We selected as Development Type just the new development projects and obtained 1975 projects. In Table 10 we show the descriptive statistics of the retrieved projects in terms of person/month (PM).

	Descriptive statistics			
	#Projects	**Mean (PM)**	**Std.dev (PM)**	**Median (PM)**
Planning	394	687.46	1775.65	160.00
Specification	627	1102.76	2945.44	242.00
Design	374	1094.,10	2743.63	330.00
Building	779	3121.34	5994.17	963,00
Testing	722	1314.44	2872.82	419.50
Deployment	482	661.10	2773.01	105.00

Table 10: effort descriptive statistics per phase

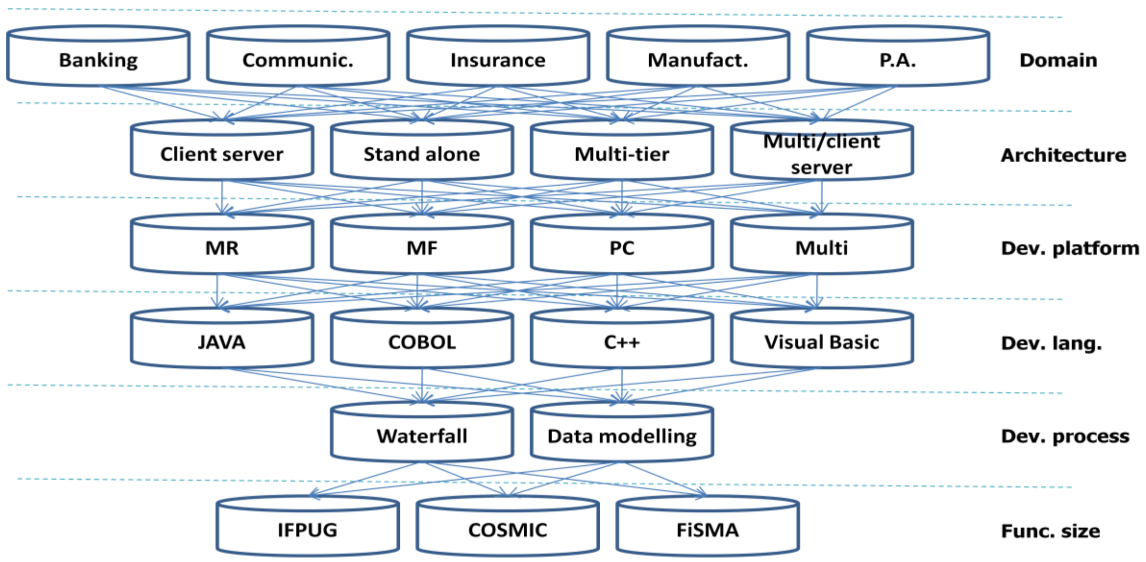


Figure 13: project cluster

We then selected the new development projects containing information on Architecture, Development techniques, Domain, Development platform and Functional measurement approach. In Figure 13 we show the project cluster that contains the combination of values investigated.

In Table 11 we show the number of projects identified for each phase divided per common characteristics.

Attributes		#projects	Plann. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
ALL		1972	329	306	372	692	453
FUNCTIONAL MEASUREMENT APPROACH	ALL	1872	232	137	162	470	275
	IFPUG	1310	211	50	60	330	164
	COSMIC	187	23	88	106	124	98
DOMAIN	ALL	1566	268	305	371	571	429
	BANKING	250	101	117	136	173	161
	INSURANCE	261	6	23	31	47	36
	COMMUNICATION	126	11	10	11	53	12
	MANUFACTURING	146	21	21	23	29	21
	PUBLIC ADMINISTRATION	115	11	37	42	54	44
PLATFORM	ALL	1724	266	294	353	557	433
	MULTI	532	16	67	97	108	74
	MAIN FRAME	497	96	91	99	202	176
	PC	472	130	118	135	181	142

Attributes		#projects	Plann. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
	MIDE RANGE	222	24	18	22	66	41
DEVELOPMENT	ALL	1172	211	69	88	340	184
	WATER FALL	451	21	41	46	46	34
	DATA MODELLING	221	78	0	4	107	83
ARCHITECTURE	ALL	944	140	135	168	279	149
	CLIENT SERVER	328	86	60	79	129	82
	STAND ALONE	379	70	9	11	114	77
	MULTI-TIER	44	12	4	7	19	17
	MULTI-TIER/STAND ALONE	159	0	0	0	0	0

Table 11: number of new development projects with valid data

4.2 Effort data analysis

Here, we report the results for each research question, analyzed as described in section 4.1.4

RQ1: estimating the effort for the next development phase based on the previous one

The ISBSG dataset contains 1975 new development projects. We obtained a good correlation and an acceptable goodness of fit only when estimating design effort based on the effort spent during the specification phase and predicting build phase effort based on design effort, as shown in Figure 12.There are no statistically significant correlations among the other phases. We suppose this is due to the lack of project clustering, since we did not group the projects based on their characteristics such as domain, programming language and others.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.370	0.137	1.700	0.840
design from spec.	0.682	0.465	0.975	0.389
build from design	0.682	0.465	0.510	0.460
test from build	0.576	0.331	1.220	0.610
deploy from test	0.361	0.131	1.980	0.950

Table 12: previous phase vs next phase

RQ 1.1 : Does considering other characteristics of the projects, in addition to the effort for a phase, improve the effort prediction for the next phase in a statistically significant way?

Here we investigated the correlation between a phase and the next one analyzing the project for each single common caractheristics.

Moreover, in order to investigate whether estimation accuracy improves by grouping by common caractheristics, we compare the estimation accuracy between RQ1.1. and RQ1 for each common caractheristics considered for clustering the projects.

In some cases, there are not enough projects to draw statistical significant conclusions. For this reason, we left the column empty.

Clustering by one characteristic

We investigate the correlation between a phase and the next one clustering the projects by one common caractheristic.

Clustering by domain

Here we report the results obtained for the first cluster where we select, as common characteristic, the domain: banking, communications, insurance, manufacturing and public administration.

Domain: Banking

When clustering by Banking domain, the data analyzed show a good correlation for the specification and for the build effort estimation instead of for the design phase the correlation is very low. The accuracy for all combinations is not acceptable, as shown in Table 13.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.733	0.537	0.470	0.330
design from spec.	0.208	0.043	0.880	0.490
build from design	0.674	0.454	1.300	0.620
test from build	0.583	0.339	1.330	0.620
deploy from test	0.499	0.249	1.530	0.770

Table 13: previous phase vs next phase - Banking

Comparing the model obtained for all projects with this cluster, we can see a dramatic improvement in estimating the effort for the specification phase based on the planning phase in terms of goodness of fit, while estimating the effort for the design phase based on the specification phase we can see a huge drop of the accuracy (see Table 12). For the other phases there are no improvements for the estimation accuracy (see Table 12).

Domain: Communications

In this case, the combinations that allow us to analyze the data are: (1) test versus build and (2) deployment versus test. In the both cases we obtain a high correlation even if the estimation accuracy is not acceptable, as shown in Table 14.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.				
build from design				
test from build	0.937	0.878	0.650	0.623
deploy from test	0.865	0.748	0.765	0.615

Table 14: previous phase vs next phase - Communications

Comparing the model obtained for all projects with this cluster, we could obtain show a dramatic improvement of the correlation and the estimation accuracy (see Table 12).

Domain: Insurance

Clustering by Insurance, the combinations that does not allow us to analyze the data is the specification phase versus planning one. We obtain a good correlation only for the deployment effort estimation even if the accuracy is not acceptable in every combinations, as shown in Figure 15.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.343	0.117	2.490	1.920
build from design	0.318	0.101	4.210	3.870
test from build	0.285	0.081	1.450	0.610
deploy from test	0.638	0.407	1.230	0.480

Table 15: previous phase vs next phase - Insurance

Comparing the model obtained for all projects with this cluster, we can see a huge drop of the accuracy except for the effort estimation of the deployment phase based on the test phase with a small improvement (see Table 12).

Domain: Manufacturing

In this cluster we obtain a high correlation for the deployment, test and design effort estimation even of the accuracy is not acceptable in every combinations, as shown in Table 16.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.451	0.204	1.060	0.620
design from spec.	0.782	0.612	1.280	0.540
build from design	0.441	0.194	0.790	0.440
test from build	0.790	0.577	0.830	0.360
deploy from test	0.810	0.007	1.260	0.550

Table 16: previous phase vs next phase - Manufacturing

Comparing the model obtained for all projects with this cluster, we can see a small improvement of the accuracy when we estimate the effort for the deployment phase based on the test phase. In the other phases there are a huge drop of the accuracy (see Table 12).

Domain: Public administration

Here we obtain a very good correlation for all the combinations except for the build effort estimation. However the accuracy is not acceptable in any cases, as shown in Figure 16.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.842	0.709	2.380	2.040
design from spec.	0.731	0.534	0.840	0.770
build from design	0.230	0.053	1.370	0.720
test from build	0.648	0.419	1.500	0.860
deploy from test	0.937	0.877	1.260	0.550

Table 17: previous phase vs next phase - Public administration

Comparing the model obtained for all projects with this cluster, the improvement of the accuracy is small estimating the effort for the design phase based on the specification phase and for the deployment phase based on the test phase. In the other phases there are a huge drop of the accuracy (see Table 12).

<u>Clustering by architecture</u>

Here we report the results obtained for the second cluster where we select, as common characteristic, the architecture: client server, stand alone, multi tier and multi tier/client server.

Architecture: Client server

In Client server cluster we obtain a good correlation for all analysis except if we consider as previous phase planning. Taking into account the estimation accuracy the results are not acceptable as shown in Table 18.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan	0.150	0.022	2.110	1.176
design from spec.	0.899	0.807	1.118	0.649
build from design	0.638	0.408	1.082	0.714
test from build	0.774	0.599	0.875	0.488
deploy from test	0.815	0.665	1.084	0.640

Table 18: previous phase vs next phase - Client server

Comparing the model obtained for all projects with this cluster, we can see a dramatic improvement of the accuracy estimating the effort for the design, test and deployment phases based on the previous (see Table 12).

Architecture: Stand alone

In this case, the combinations that don't allow us to analyze the data are: (1) design versus specification and (2) build versus design. in this cluster we obtain good results even if the goodness of fit is not acceptable as shown in Table 19.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan	0.675	0.465	0.891	0.640
design from spec.				
build from design				
test from build	0.668	0.446	1.071	0.620
deploy from test	0.480	0.231	1.371	0.737

Table 19: previous phase vs next phase - Stand alone

Comparing the model obtained for all projects with this cluster we can see a dramatic improvement estimating of the accuracy (see Table 12).

Architecture: Multi tier

In this case, the combinations that don't allow us to analyze the data are: (1) design versus specification and (2) build versus design. We find very good results even if the goodness of fit is acceptable just for the effort estimation of the test phase, as shown in Table 20.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.	0.699	0.488	0.729	0.558
design from spec.				

build from design				
test from build	0.939	0.882	0.273	0.291
deploy from test	0.774	0.600	1.344	0.650

Table 20: previous phase vs next phase - Multi tier

Comparing the model obtained for all projects with this cluster, we can see a dramatic improvement estimating all phases of the accuracy (see Table 12) especially when we estimate the test effort based on the previous one.

<u>Clustering by development platform</u>

Here we report the results obtained for the cluster where we select, as common characteristic, the development platform: MR, MF, PC and Multi.

Development platform: MR

In MR cluster we obtain a very good correlation when we estimate the effort for build, design and specification phases. Taking into account the estimation accuracy we find acceptable results just for the design phase, as shown in Table 21**.**

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.	0.453	0.205	1.300	1.270
design from spec.	0.831	0.690	0.840	0.780
build from design	0.973	0.946	0.350	0.270
test from build	0.814	0.663	0.480	0.390
deploy from test	0.407	0.166	1.130	0.650

Table 21: previous phase vs next phase - MR

Comparing the model obtained for all projects with this cluster, we can see an improvement of the accuracy (see Table 12) especially take in to account test and build phases.

Development platform: MF

Here we obtain a very good correlation when we estimate the effort for build, design phases. Taking into account the estimation accuracy we don't find acceptable results, as shown in Table 22.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.	0.373	0.139	1.530	1.230
design from spec.	0.303	0.092	0.750	0.560

build from design	0.665	0.443	0.510	0.530
test from build	0.781	0.610	0.720	0.520
deploy from test	0.575	0.331	1.390	0.780

Table 22: previous phase vs next phase – MF

Comparing the model obtained for all projects with this cluster, there are no improvement of the accuracy (see Table 12) where in several case the accuracy decrease.

Development platform: PC

In this cluster we obtain results a very good correlation when we estimate the effort for test, build and design phases. Taking into account the estimation accuracy we don't find acceptable results, as shown in Table 23.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.249	0.062	1.250	0.490
design from spec.	0.925	0.855	0.920	0.480
build from design	0.678	0.460	1.960	0.660
test from build	0.624	0.389	1.240	0.530
deploy from test	0.330	0.109	1.480	0.750

Table 23: previous phase vs next phase – PC

Comparing the model obtained for all projects with this cluster we obtain an improvement of the accuracy for specification and for deployment phases (see Table 12).

Development platform: Multi

In Multi cluster, we obtain a very good correlation except when we estimate the effort for design phase. Taking into account the estimation accuracy we don't find acceptable results, as shown in Table 24.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.944	0.891	3.140	3.120
design from spec.	0.661	0.437	1.780	0.770
build from design	0.320	0.103	1.880	1.190
test from build	0.825	0.680	1.230	0.850
deploy from test	0.915	0.837	1.230	0.850

Table 24: previous phase vs next phase – Multi

Comparing the model obtained for all projects with this cluster, we have an improvement of the accuracy for the deployment phase and a huge drop of the accuracy for the specification phase (see Table 12).

<u>Clustering by programming language</u>

Here we report the results obtained for the cluster where we select, as common characteristic, the programming language: Java, COBOL, C++ and Visual basic.

Programming language: Java

Selecting Java projects, we do not find a good correlation between the phases. Such results are confirmed taking into account the estimation accuracy where we don't find acceptable results, as shown in Table 25.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.418	0.002	1.360	0.820
design from spec.	0.273	0.064	0.490	0.520
build from design	0.349	0.114	1.040	0.620
test from build	0.591	0.344	0.880	0.520
deploy from test	0.394	0.146	1.350	0.730

Table 25: previous phase vs next phase - Java

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the accuracy, even if the correlation improve only for the specification, design and build effort estimation based on the previous one (see Table 12).

Programming language: COBOL

In COBOL cluster we find a good correlation only when we estimate the effort for test and build phases even if only the estimation accuracy are not acceptable, as shown in Table 26.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.482	0.215	1.490	1.000
design from spec.	0.139	0.495	0.550	0.550
build from design	0.802	0.638	1.720	1.420
test from build	0.678	0.454	0.950	0.550
deploy from test	0.460	0.204	1.770	0.930

Table 26: previous phase vs next phase - COBOL

Comparing the model obtained for all projects with this cluster, we find an improvement of the correlation when we estimate the build, test and deployment phases while the correlation decrease in specification and design phases. Taking into account the accuracy we obtain an improvement only for the specification, design and build effort estimation based on the previous one (see Table 12).

Programming language: C++

With the C++ cluster we find a good correlation only when we estimate the effort for test and build phases even if only the estimation accuracy are not acceptable, as shown in Table 27.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.572	0.315	1.380	1.023
design from spec.	0.259	0.595	0.650	0.510
build from design	0.772	0.578	1.670	1.350
test from build	0.758	0.474	0.810	0.690
deploy from test	0.530	0.310	1.680	0.980

Table 27: previous phase vs next phase - C++

Comparing the model obtained for all projects with this cluster we obtain an improvement of the accuracy for specification and for deployment phases (see Table 12).

Programming language: Visual basic

Here we obtain a high correlation between all the phases investigated. Moreover taking into account the estimation accuracy we don't find acceptable results, as shown in Table 28.

	pearson	R^2	mmre	mdmre
spec. from plan .	0.709	0.486	1.590	1.040
design from spec.	0.736	0.517	2.310	1.470
build from design	0.649	0.398	1.290	0.580
test from build	0.839	0.699	0.550	0.400
deploy from test	0.734	0.523	1.230	0.500

Table 28: previous phase vs next phase - Visual Basic

Comparing the model obtained for all projects with this cluster, we find an improvement of the correlation in every case. Taking into account the accuracy we find an improvement only for the specification, test and deployment effort estimation based on the previous one (see Table 12).

<u>Clustering by development process</u>

Here we report the results obtained for the following cluster where we select, as common characteristic, the development process: waterfall and data modelling.

Development process: Waterfall

In waterfall cluster we obtain good correlation for the estimation of build and design phases even if taking into account the estimation accuracy we don't find acceptable results, as shown in Table 29.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.551	0.303	0.610	0.510
design from spec.	0.643	0.414	0.630	0.600
build from design	0.418	0.175	0.900	0.500
test from build	0.848	0.720	0.510	0.410
deploy from test	0.337	0.114	1.190	0.780

Table 29: previous phase vs next phase - Waterfall

Comparing the model obtained for all projects with this cluster, we obtain an improvement for specification, design and test phases and a huge drop of the accuracy for the build one (see Table 12).

Development process: Data modelling

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. In this cluster we don't obtain results for build and design phases, while for the other phases we find a good correlation only for the estimation of test phase. Taking into account the estimation accuracy we don't find acceptable results, as shown in Table 30.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.479	0.229	2.230	1.570
design from spec.				
build from design				
test from build	0.811	0.657	0.510	0.380
deploy from test	0.589	0.347	1.140	0.730

Table 30: previous phase vs next phase - Data modelling

Comparing the model obtained for all projects with this cluster we find an improvement for design and deployment phases and a huge drop of the accuracy for the specification one (see Table 12).

<u>Clustering by functional measurement approach</u>

Here we report the results obtained for the cluster where we select, as common characteristic, the functional measurement approach: IFPUG and COSMIC. We consider for each case either the value minor to 1000 and major or equal to 1000.

Functional measurement approach: IFPUG < 1000

In this cluster the correlation is good in every case, even if taking into account the estimation accuracy we don't find acceptable results, as shown in Table 31.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.564	0.314	1.219	0.860
design from spec.	0.714	0.495	0.581	0.420
build from design	0.658	0.418	0.844	0.518
test from build	0.728	0.528	0.765	0.854
deploy from test	0.532	0.222	1.880	0.870

Table 31: previous phase vs next phase – IFPUG < 1000

Comparing the model obtained for all projects with this cluster, we can obtain an improvement of the accuracy (see Table 12).

For this cluster we investigate also the data considering multivariate model as shown in Table 32. Here the column "pearson" report in the first row the value related to the contribution of the previous phase and in the second row the contribution of IFPUG<1000.

Comparing the multivariate model with the univariate one we don't obtain an improvement of the correlation and the accuracy of the results.

<table>
<tr><th></th><th>pearson</th><th>R^2</th><th>mmre</th><th>mdmre</th></tr>
<tr><td rowspan="2">spec. from plan.</td><td>0.564</td><td rowspan="2">0.335</td><td rowspan="2">0.919</td><td rowspan="2">0.553</td></tr>
<tr><td>0.391</td></tr>
<tr><td rowspan="2">design from spec.</td><td>0.249</td><td rowspan="2">0.956</td><td rowspan="2">395.910</td><td rowspan="2">320.165</td></tr>
<tr><td>0.714</td></tr>
<tr><td rowspan="2">build from design</td><td>0.553</td><td rowspan="2">0.569</td><td rowspan="2">1.570</td><td rowspan="2">1.120</td></tr>
<tr><td>0.658</td></tr>
<tr><td rowspan="2">test from build</td><td>0.283</td><td rowspan="2">0.526</td><td rowspan="2">0.770</td><td rowspan="2">0.520</td></tr>
<tr><td>0.728</td></tr>
<tr><td rowspan="2">deploy from test</td><td>0.229</td><td rowspan="2">0.288</td><td rowspan="2">1.270</td><td rowspan="2">0.660</td></tr>
<tr><td>0,532</td></tr>
</table>

Table 32: previous phase vs next phase – IFPUG < 1000 – Multilinear regression

Functional measurement approach: IFPUG ≥ 1000

In this case the only combination that doesn't allow us to analyze the data is the specification phase versus planning one. In Table 33 we show high correlation between the phases especially when we considered the test effort for estimating the deployment one. Moreover the accuracy of all results are not acceptable.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.976	0.946	0.496	0.413
build from design	0.689	0.431	1.189	1.437
test from build	0.768	0.584	0.730	0.440
deploy from test	0.384	0.224	1.120	0.610

Table 33: previous phase vs next phase – IFPUG > 1000 – Linear regression

Comparing the model obtained for all projects with this cluster, we can obtain an improvement of the accuracy except for the build effort estimation(see Table 12).

Also comparing the multivariate model with the univariate one we don't obtain an improvement of the correlation and the accuracy of the results as shown in Table 34.

<table>
<tr><th></th><th>pearson</th><th>R^2</th><th>mmre</th><th>mdmre</th></tr>
<tr><td rowspan="2">spec. from plan.</td><td>0.267</td><td rowspan="2">0.039</td><td rowspan="2">1.701</td><td rowspan="2">0.806</td></tr>
<tr><td>0.167</td></tr>
<tr><td rowspan="2">design from spec.</td><td>0.976</td><td rowspan="2">0.956</td><td rowspan="2">0.544</td><td rowspan="2">0.456</td></tr>
<tr><td>-0.041</td></tr>
<tr><td rowspan="2">build from design</td><td>0.488</td><td rowspan="2">0.673</td><td rowspan="2">0.670</td><td rowspan="2">0.581</td></tr>
<tr><td>0.689</td></tr>
<tr><td rowspan="2">test from build</td><td>0.417</td><td rowspan="2">0.587</td><td rowspan="2">0.800</td><td rowspan="2">0.580</td></tr>
<tr><td>0.768</td></tr>
<tr><td rowspan="2">deploy from test</td><td>0.151</td><td rowspan="2">0.111</td><td rowspan="2">1.510</td><td rowspan="2">0.680</td></tr>
<tr><td>0,384</td></tr>
</table>

Table 34: previous phase vs next phase – IFPUG > 1000 – Multilinear regression

Functional measurement approach: COSMIC<1000

In this cluster the correlation is good in every case, even if taking into account the estimation accuracy we don't find acceptable results, as shown in Table 34

	pearson	R^2	mmre	mdmre
spec. from plan	0.504	0.414	1.119	0.960
design from spec.	0.619	0.595	0.681	0.526
build from design	0.598	0.503	0.832	0.512
test from build	0.651	0.596	0.769	0.766
deploy from test	0.492	0.325	1.670	0.979

Table 35: previous phase vs next phase - COSMIC<1000

Clustering by two characteristic

We investigate the correlation between a phase and the next one clustering the project by two common caractheristic.

Clustering by Domain and Architecture

Here we select Domain and Development process (see Figure 14) as common caractheristics. Following the results obtained.

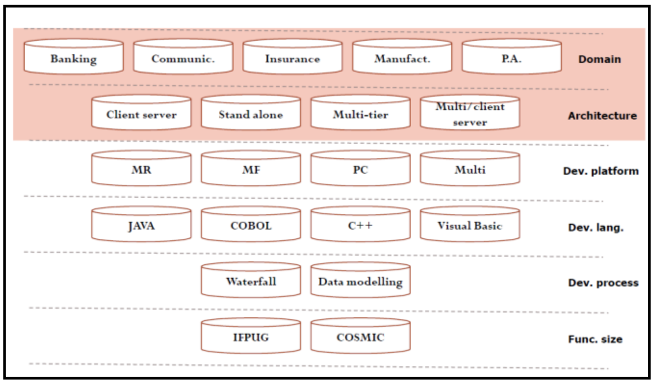


Figure 14: clustering by Domain and Architecture

Domain and Architecture: Banking and Stand alone

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. Here we don't obtain results for build and design phases, while for the other phases we find a high correlation. Taking into account the estimation accuracy we find acceptable results only for the effort test estimation, as shown in Table 36.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.919	0.845	0.524	0.638
design from spec.				
build from design				
test from build	0.949	0.901	0.358	0.295
deploy from test	0.859	0.738	0.913	0.894

Table 36: previous phase vs next phase - Banking and Stand alone

Comparing the model obtained for all projects with this cluster, we obtain a dramatic improvement of the correlation especially for the test phase (see Table 12 and Table 13)

Domain and Architecture: Communications and Stand alone

In this case the only combinations that allows us to analyze the data is the test phase. We obtain very good correlation and an acceptable goodness of fit results only estimating the test effort from the design one as shown in Table 37.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.				
build from design				

test from build	0.922	0.851	0.411	0.374
deploy from test				

Table 37: previous phase vs next phase - Stand alone and Communications

Comparing the model obtained for all projects with this cluster, we obtain a dramatic improvement of the accuracy (see Table 12 and Table 14).

Clustering by Domain and Developed platform

Here we select Domain and Development process (see Figure 15) as common caractheristics. Following the results obtained.

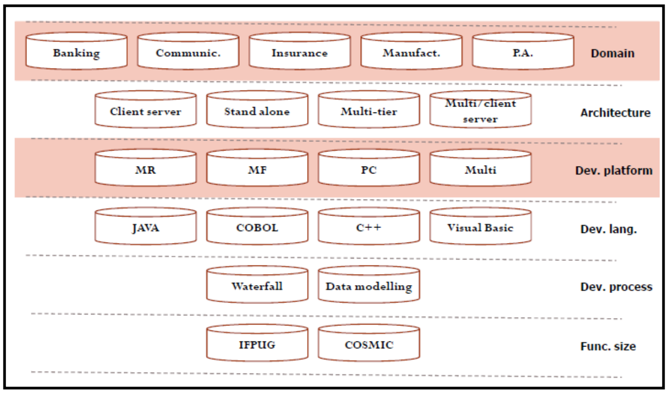


Figure 15: Clustering by Domain and Development platform

Domain and Development platform: Banking and MF

Selecting Banking and MF we obtain good results for specification and build phases, while for the other phases we find a low correlation. Taking into account the estimation accuracy we find a no acceptable results, as shown in Table 38.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.888	0.788	0.520	0.497
design from spec.	0.108	0.012	0.971	0.472
build from design	0.792	0.627	0.639	0.424
test from build	0.597	0.356	0.981	0.566
deploy from test	0.451	0.204	1.267	0.766

Table 38: previous phase vs next phase - Banking and MF

Comparing the model obtained for all projects with this cluster we don't find statistically variation of the accuracy (see Table 12 and Table 13).

Domain and Development platform: Banking and PC

Considering Banking and PC projects we obtain good results for specification, test and deployment phases, while for the other phases we find a low correlation. Taking into account the estimation accuracy we find a no acceptable results, as shown in Table 39.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.779	0.608	0.973	0.660
design from spec.	0.356	0.126	1.096	0.423
build from design	0.240	0.058	1.212	0.572
test from build	0.587	0.345	0.717	0.840
deploy from test	0.619	0.384	3.270	0.696

Table 39: previous phase vs next phase - Banking and PC

Comparing the model obtained for all projects with this cluster we obtain an improvement of the accuracy for the specification and test phases and for specification and design phases (see Table 12 and Table 13).

Domain and Development platform: Banking and Multi

In this case, the combinations that do not allow us to analyze the data are: (1) specification versus planning and (2) design versus specification. We obtain good results for specification and build phases, while for the other phases we find a low correlation. Taking into account the estimation accuracy we don't find acceptable results, as shown in Table 40.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.				
build from design	0.609	0.371	0.424	0.323
test from build	0.594	0.353	0.870	0.627
deploy from test	0.404	0.164	2.023	0.805

Table 40: previous phase vs next phase - Banking and Multi

Comparing the model obtained for all projects with this cluster we find that the improvement of the accuracy in is not significant (see Table 12 and Table 13).

Domain and Development platform: Communications and MR

In this case the only combination that allows us to analyze the data is test phase versus build one. The correlation obtained is very high with an acceptable accuracy, as shown in Table 41.

	pearson	R^2	mmre	mdmre
spec. from plan.				

design from spec.				
build from design				
test from build	0.965	0.932	0.310	0.270
deploy from test				

Table 41: previous phase vs next phase - Communications and MR

Comparing the model obtained for all projects with this cluster we have a dramatic improvement of the accuracy (see Table 12 and Table 14).

Domain and Development platform: Communications and PC

In this case the only combination that allows us to analyze the data is test phase versus build one. The correlation is very high with an acceptable accuracy, as shown in Table 42.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.				
design from spec.				
build from design				
test from build	0.938	0.879	0.290	0.210
deploy from test				

Table 42: previous phase vs next phase - Communications and PC

Comparing the model obtained for all projects with this cluster we shows an improvement of the accuracy (see Table 12 and Table 14).

Domain and Development platform: Insurance and MF

In this case the only combination that allows us to analyze the data is test phase versus build one. The correlation is not so high and the accuracy is not acceptable, as shown in Table 43.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.				
design from spec.				
build from design				
test from build	0.594	0.354	2.436	1.848
deploy from test				

Table 43: previous phase vs next phase - Insurance and MF

Comparing the model obtained for all projects with this cluster we shows a huge drop of the accuracy (see Table 12 and Table 15).

Domain and Development platform: Insurance and multi

In this case, the combinations that allow us to analyze the data are: (1) test versus build and (2) deployment versus test The correlation estimated is not so high and the accuracy are not acceptable, as shown in Table 44.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.				
design from spec.				
build from design				
test from build	0.582	0.338	0.730	0.878
deploy from test	0.201	0.004	1.785	1.931

Table 44: previous phase vs next phase - Insurance and Multi

Comparing the model obtained for all projects with this cluster, we don't find an improvement of the accuracy (see Table 12 and Table 15).

Domain and Development platform: Public administration and Multi

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. The correlation is very high for the design and deployment effort estimation, even if the accuracy is acceptable only in the first case, as shown in Table 45.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.				
design from spec.	0.802	0.644	0.420	0.440
build from design	0.243	0.056	0.650	0.570
test from build	0.348	0.121	0.560	0.490
deploy from test	0.961	0.924	0.970	0.800

Table 45: previous phase vs next phase - Public administration and Multi

Comparing the model obtained for all projects with this cluster, we find shows an improvement of the accuracy for all the analysis (see Table 12 and Table 17).

Clustering by Domain and Programming language

Here we select Domain and Programming language (see Figure 16) as common caractheristics. Following the results obtained.

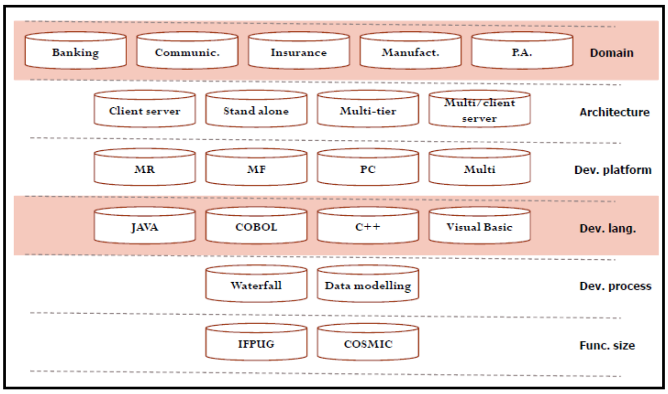


Figure 16:clustering by Domain and programming language

Domain and Programming language: Banking and Java

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. The correlation is good for the test and deployment effort estimation even if the accuracy is not acceptable, as shown in Table 46.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.152	0.023	1.987	2.028
build from design	0.311	0.096	1.076	0.865
test from build	0.652	0.425	0.784	0.897
deploy from test	0.572	0.327	1.456	0.821

Table 46:previous phase vs next phase - Banking and Java

Comparing the model obtained for all projects with this cluster, we don't find an improvement of the accuracy for all the analysis (see Table 12 and Table 13).

Domain and Programming language: Banking and COBOL

In this cluster the correlation estimated is not good only for the design effort estimation even if the accuracy is not acceptable, as shown in Table 47.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.962	0.962	0.450	0.460
design from spec.	0.135	0.135	0.654	0.654
build from design	0.803	0.646	0.567	0.671
test from build	0.604	0.365	0.842	0.765

deploy from test	0.456	0.208	1.143	0.832

Table 47: previous phase vs next phase - Banking and COBOL

Comparing the model obtained for all projects with this cluster, we find an improvement of the accuracy for all the analysis (see Table 12 and Table 13).

Domain and Programming language: Public administration and Java

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. The correlation estimated is not good only for the design effort estimation even if the accuracy is not acceptable, as shown in Table 48.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.777	0.604	0.873	0.781
build from design	0.539	0.291	1.354	0.983
test from build	0.617	0.380	1.435	0.921
deploy from test	0.801	0.642	1.234	0.451

Table 48: previous phase vs next phase - Public administration and Java

Comparing the model obtained for all projects with this cluster, we don't find an improvement of the accuracy for all the analysis (see Table 12 and Table 17).

<u>Clustering Domain and Functional measurement approach</u>

Here we select Domain and Functional measurement approach (see Figure 17) as common caractheristics. Following the results obtained.

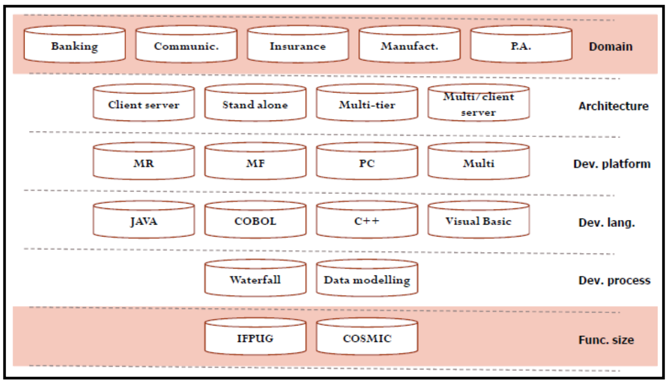


Figure 17: clustering by Domain and Functional measurement approach

Domain and Functional measurement approach: Banking and IFPUG<1000

Following we show the results obtained where the correlation is good for the specification and test effort estimation even if the accuracy is not acceptable, as shown in Table 49.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.727	0.528	0.517	0.400
design from spec.	0.035	0.001	0.703	0.680
build from design	0.234	0.055	0.208	0.150
test from build	0.657	0.432	0.502	0.390
deploy from test	0.333	0.111	0.872	0.700

Table 49: previous phase vs next phase - Banking and IFPUG<1000

Comparing the model obtained for all projects with this cluster, we find an improvement of the accuracy for all the analysis (see Table 12 and Table 13).

Domain and Functional measurement approach: Banking and COSMIC<1000

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. The correlation is good for the build and test effort estimation even if the accuracy is not acceptable, as shown in Table 50.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.340	0.116	0.904	0.880
build from design	0.589	0.347	1.097	0.950
test from build	0.422	0.178	0.902	0.830
deploy from test	0.388	0.150	1.872	1.700

Table 50: previous phase vs next phase - Banking and COSMIC<1000

Comparing the model obtained for all projects with the one clustered by Banking and this cluster, we don't find an improvement of the accuracy for all the analysis (see Table 12 and Table 13).

Domain and Functional measurement approach: Manufacturing and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) specification versus planning and (2) deployment versus test. The correlation is very high and the accuracy is acceptable as shown in Table 51.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.860	0.740	0.320	0.210
build from design	0.676	0.457	0.260	0.240

test from build	0.757	0.573	0.350	0.260
deploy from test				

Table 51: previous phase vs next phase - Manufacturing and IFPUG1000

Comparing the model obtained for all projects with the one obtained clustered by Manufacturing and this cluster, we find an improvement of the accuracy for all the analysis (see Table 12 and Table 16).

Domain and Functional measurement approach: Public administration and IFPUG<1000

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. The correlation estimated is very high except for the build effort phase but the accuracy is not acceptable in any case as shown in Table 51.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.				
design from spec.	0.731	0.534	0.760	0.550
build from design	0.230	0.053	1.270	0.720
test from build	0.648	0.419	0.860	0.780
deploy from test	0.937	0.877	1.700	1.050

Table 52: previous phase and next phase - Public administration and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Public administration and this cluster, we find an improvement of the accuracy for all the analysis (see Table 12 and Table 17).

Clustering by Architecture and Development platform

Here we select Architecture and Development platform (see Figure 18). Following the results obtained.

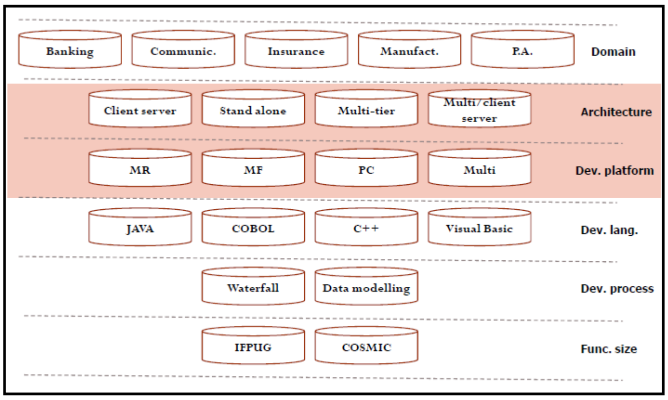


Figure 18: clustering by Architecture and Development platform

Architecture and Development platform: Client server and PC

In this cluster the correlation is good except for the specification and deployment effort phases but the accuracy is not acceptable in any case, as shown in Table 53.

	pearson	R^2	mmre	mdmre
spec. from plan	0.130	0.017	1.843	1.287
design from spec.	0.541	0.292	0.921	0.872
build from design	0.673	0.453	1.065	1.175
test from build	0.783	0.613	0.985	0.785
deploy from test	0.148	0.022	1.345	1.187

Table 53: previous phase vs next phase - Client server and PC

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we don't find an improvement of the accuracy for all the analysis (see Table 12 and Table 18).

Architecture and Development platform: Client server and Multi

In Table 54 we show the results obtained where the correlation is very high and the accuracy is acceptable in every case except in design and build phases.

	pearson	R^2	mmre	mdmre
spec. from plan	0.951	0.904	0.235	0.342
design from spec.	0.947	0.897	0.356	0.423
build from design	0.869	0.755	0.387	0.402

test from build	0.918	0.843	0.298	0.311
deploy from test	0.973	0.947	0.267	0.299

Table 54: previous phase vs next phase - Client server and Multi

Comparing the model obtained for the all projects with the one obtained clustered by Client server and this cluster, we find a dramatic improvement of the accuracy for all the analysis (see Table 12 and Table 18).

Architecture and Development platform: Stand alone and MF

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. In Table 55 we show the results obtained where the correlation is good but the accuracy is not acceptable in any case.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan	0.691	0.477	0.945	0.865
design from spec.				
build from design				
test from build	0.601	0.362	1.087	1.132
deploy from test	0.768	0.590	0.821	0.902

Table 55:previous phase vs next phase - Stand alone and MF

Comparing the model obtained for all projects and the one obtained clustered by Stand alone and this cluster, we don't find an improvement of the accuracy for all the analysis (see Table 12 and Table 19).

Architecture and Development platform: Stand alone and PC

In this case, the combinations that allow us to analyze the data are: (1) design versus specification and (2) build versus design. In Table 56 we show the results obtained where the correlation is good but the accuracy is not acceptable in any case.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan	0.685	0.469	0.985	0.832
design from spec.				
build from design				
test from build	0.720	0.519	0.987	0.832
deploy from test	0.766	0.587	0.897	0.954

Table 56: previous phase vs next phase - Stand alone and PC

Comparing the model obtained for all projects and the one obtained clustered by Stand alone and this cluster, we find an improvement of the accuracy for the test and deployment effort estimation (see Table 12 and Table 19).

__Clustering by Architecture and Development process__

Here we select Architecture and Programming language (see Figure 19). Following we show the results.

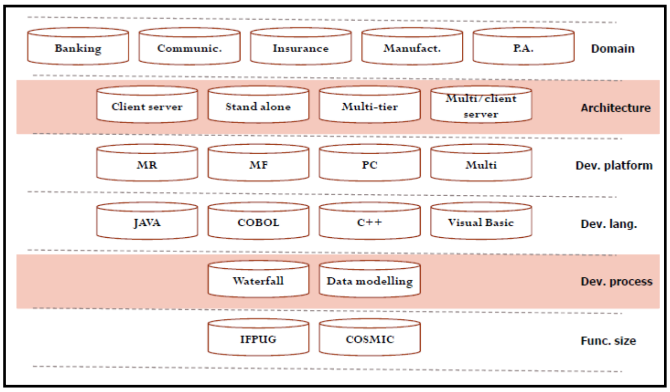


Figure 19: clustering by Architecture and Development process

Architecture and Development process: Client server and Waterfall

In this case, the combinations that do not allow us to analyze the data are: (1) specification versus planning and (2) deployment versus test. In Table 57 we show the results obtained, for the specification and deployment effort estimation there are not enough projects for the analysis.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.860	0.740	0.320	0.210
build from design	0.676	0.457	0.260	0.240
test from build	0.757	0.573	0.350	0.260
deploy from test				

Table 57: previous phase vs next phase - Client server and Waterfall

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we find an improvement of the accuracy in every case (see Table 12 and Table 19).

Architecture and Development process: Client server and Data modelling

In this case, the combinations that allow us to analyze the data are: (1) specification versus planning and (2) test versus build. In Table 58 we show the results obtained where the correlation is very high only for the test effort phase with an acceptable goodness of fit.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.333	0.111	2.530	0.890
design from spec.				
build from design				
test from build	0.936	0.876	0.360	0.340
deploy from test				

Table 58: previous phase vs next phase - Client server and Data modelling

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we find an improvement of the accuracy only for the test effort estimation (see Table 12 and Table 19).

Architecture and Development process: Stand alone and Data modelling

In Table 59 we show the results obtained show a very high correlation for the analysis except for the specification and design effort phase but the accuracy is not acceptable in any case.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.334	0.112	0.850	0.800
design from spec.	0.439	0.193	0.510	0.540
build from design	0.883	0.780	0.450	0.490
test from build	0.876	0.768	0.460	0.360
deploy from test	0.688	0.474	0.440	0.400

Table 59: previous phase vs next phase - Stand alone and Data modelling

Comparing the model obtained for all projects with the one obtained clustered by Stand alone and this cluster, we find an improvement of the accuracy only for the test effort estimation (see Table 12 and Table 19).

Clustering by Architecture and Programming language

Here we select Architecture and Programming language (see Figure 20). Following we show the results.

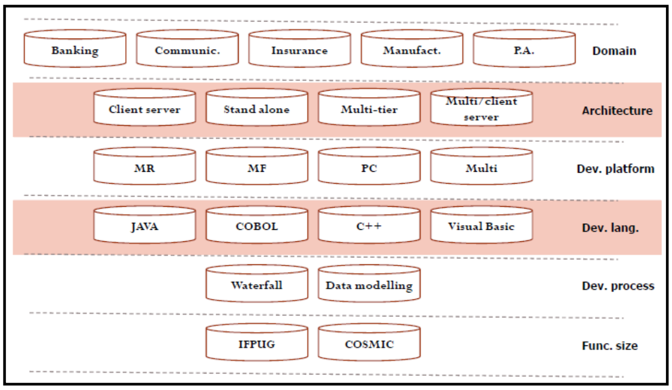


Figure 20: clustering by Architecture and Programming language

Architecture and Programming language: Client server and Java

In Table 60 we show the results obtained show a good correlation for the build and test phases but the accuracy is not acceptable in any case.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.089	0.008	2.098	1.256
design from spec.	0.469	0.220	1.155	0.983
build from design	0.863	0.744	0.821	0.673
test from build	0.637	0.405	0.955	0.621
deploy from test	0.100	0.325	1.189	0.832

Table 60: previous phase vs next phase - Client server and Java

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we don't find an improvement of the accuracy only for the test effort estimation (see Table 12 and Table 18).

Architecture and Programming language: Client server and Visual basic

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. In Table 61 we show the results obtained show a good correlation for the test and deployment phases but the accuracy is not acceptable in any case.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.486	0.236	2.045	1.054

design from spec.				
build from design				
test from build	0.855	0.732	0.732	0.564
deploy from test	0.792	0.627	0.654	0.554

Table 61: previous phase vs next phase - Client server and Visual basic

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we find an improvement of the accuracy for test and deployment phases (see Table 12 and Table 18).

Clustering by Architecture and Functional measurement approach

Here we select Architecture and Functional measurement approach (see Figure 21). Following the results.

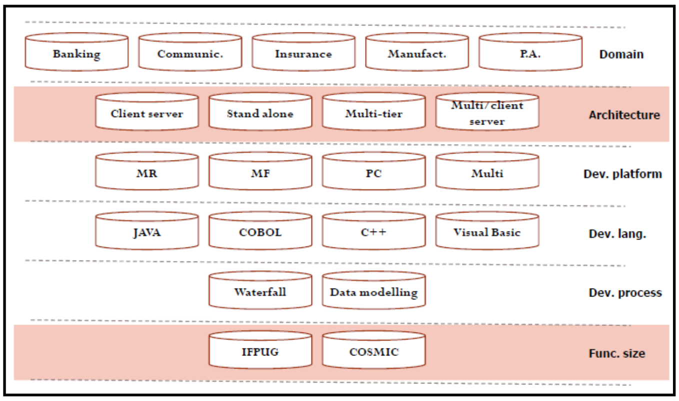


Figure 21: clustering by Architecture and Functional measurement approach

Architecture and Functional measurement approach: Client server and IFPUG<1000

The results obtained show a good correlation for the analysis only for the build and test effort estimation, but the accuracy is not acceptable in any case, as reported in Table 62.

	pearson	R^2	mmre	mdmre
spec. from plan.	-0.060	0.004	0.940	0.800
design from spec.	0.369	0.136	0.490	0.410
build from design	0.640	0.410	0.640	0.400
test from build	0.861	0.742	0.460	0.380
deploy from test	0.567	0.321	0.730	0.440

Table 62: previous phase vs next phase - Client server and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we find an improvement of the accuracy (see Table 12 and Table 18).

Architecture and Functional measurement approach: Stand alone and IFPUG<1000

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. The results obtained show a good correlation, but the accuracy is not acceptable in any case, as reported in Table 63.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.664	0.441	0.780	0.680
design from spec.				
build from design				
test from build	0.772	0.597	0.520	0.410
deploy from test	0.465	0.216	0.850	0.710

Table 63: previous phase vs next phase - Client server and COSMIC<100

Comparing the model obtained for all projects with the one obtained clustered by Stand alone and this cluster, we find an improvement of the accuracy (see Table 12 and Table 19).

<u>**Clustering by Development platform and Development process**</u>

Here we select Development platform and Development process (see Figure 22). Following the results.

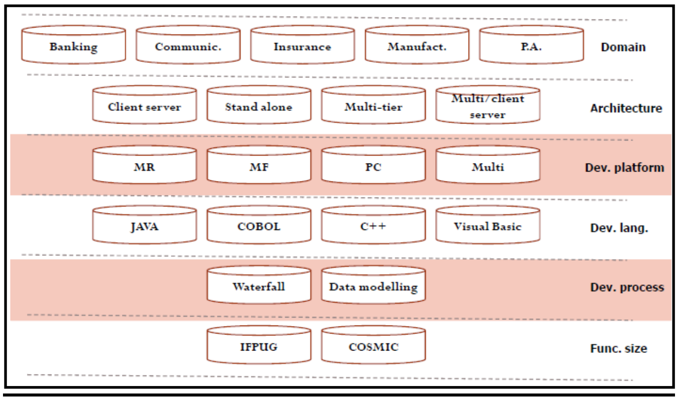


Figure 22: clustering by Development platform and Development process

Development platform and Development process: MF and Waterfall

In this case, the combinations that allow us to analyze the data are: (1) design versus specification and (2) deployment versus test. The results obtained show a good correlation, but the accuracy is not acceptable in any case, as reported in Table 64.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.584	0.341	0.520	0.460
build from design				

test from build	0.564	0.318	0.420	0.420
deploy from test				

Table 64: previous phase vs next phase - MF and Waterfall

Comparing the model obtained for all projects with the one obtained clustered by MF and this cluster, we find an improvement of the accuracy (see Table 12 and Table 22).

Development platform and Development process: MF and Data modelling

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. The results obtained show a good correlation, but the accuracy is not acceptable in any case, as reported in Table 65.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.464	0.216	0.940	0.720
design from spec.				
build from design				
test from build	0.845	0.714	0.570	0.470
deploy from test	0.647	0.418	0.810	0.660

Table 65: previous phase vs next phase - MF and Data modelling

Comparing the model obtained for all projects with the one obtained clustered by MF and this cluster, we find an improvement of the accuracy (see Table 12 and Table 22).

Development platform and Development process: PC and Waterfall

The results obtained show a good correlation, but the accuracy is not acceptable in any case, as reported in Table 66.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.579	0.335	0.510	0.450
design from spec.	0.709	0.503	0.370	0.250
build from design	0.567	0.321	0.280	0.260
test from build	0.825	0.681	0.470	0.390
deploy from test	0.294	0.086	1.540	0.860

Table 66: previous phase vs next phase - PC and Waterfall

Comparing the model obtained for all projects with the one obtained clustered by PC and this cluster, we find an improvement of the accuracy except for the deployment effort estimation (see Table 12 and Table 23).

Development platform and Development process: PC and Data modelling

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. The results obtained show a good correlation except for the deployment phase, even if the accuracy is not acceptable in any case, as reported in Table 67.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.740	0.547	1.630	1.680
design from spec.				
build from design				
test from build	0.769	0.591	0.540	0.510
deploy from test	0.265	0.070	0.660	0.660

Table 67: previous phase vs next phase - PC and Data modelling

Comparing the model obtained for all projects with the one obtained clustered by PC and this cluster, we find an improvement of the accuracy except for the specification effort estimation (see Table 12 and Table 23).

<u>Clustering b y Development platform and Functional measurement approach</u>

Here we select Development platform and Functional measurement approach (see Figure 23) as common caractheristics. Following the results obtained.

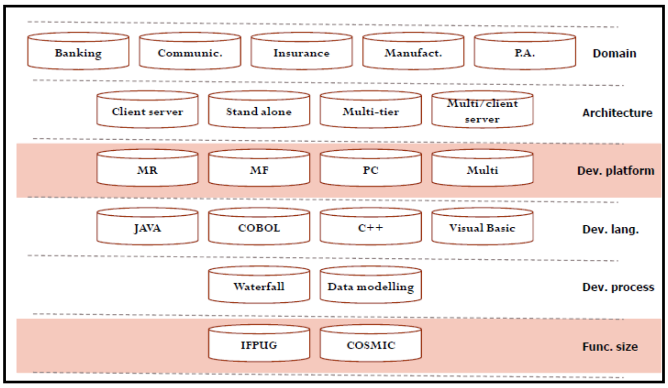


Figure 23: clustering by Development platform and Functional measurement approach

Development platform and Functional measurement approach: MF and IFPUG<1000

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. For the others combinations we have a good correlation only for the test and deployment phases, but the accuracy is not acceptable except for the test phase, as reported in Table 68.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.361	0.130	1.030	0.790
design from spec.				
build from design				

test from build	0.888	0.789	0.430	0.360
deploy from test	0.775	0.601	0.600	0.560

Table 68: previous phase vs next phase - MF and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by MF and this cluster, we find an improvement of the accuracy in every case (see Table 12 and Table 22).

Development platform and Functional measurement approach: Multi and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) build versus design and (2) test versus build. For the others we have a high correlation even if the accuracy is not acceptable except, as reported in Table 69.

	pearson	R^2	mmre	mdmre
spec. from plan .				
design from spec.				
build from design	0.745	0.555	0.740	0.740
test from build	0.979	0.958	0.660	0.170
deploy from test				

Table 69: previous phase vs next phase - Multi and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Multi and this cluster, we find an improvement of the accuracy in every case(see Table 12 and Table 24).

Development platform and Functional measurement approach: PC and IFPUG<1000

In this cluster we obtain a good correlation expect if we consider planning and test phases for estimation the next one. taking into account the accuracy the values are not acceptable in any case, as reported in Table 70.

	pearson	R^2	mmre	mdmre
spec. from plan	0.464	0.215	0.690	0.650
design from spec.	0.979	0.959	1.060	0.710
build from design	0.708	0.502	0.540	0.560
test from build	0.738	0.545	0.740	0.520
deploy from test	0.351	0.123	1.060	0.880

Table 70: previous phase vs next phase - PC and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by PC and this cluster, we find an improvement of the accuracy in every case(see Table 12 and Table 24).

Development platform and Functional measurement approach: MR and IFPUG<1000

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. We have a good correlation except for deployment phase estimation. Moreover the accuracy is acceptable only for the deployment effort estimation, as reported in Table 71.

	pearson	R^2	mmre	mdmre
spec. from plan	0.167	0.028	0.640	0.640
design from spec.				
build from design				
test from build	0.893	0.798	0.280	0.290
deploy from test	0.316	0.100	0.240	0.090

Table 71: previous phase vs next phase - MR and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by MR and this cluster, we find an improvement of the accuracy in every case(see Table 12 and Table 21).

<u>Clustering by Development process and Functional measurement approach</u>

Here we select is Development process and Functional measurement approach (see) as common caractheristics. Following the results obtained.

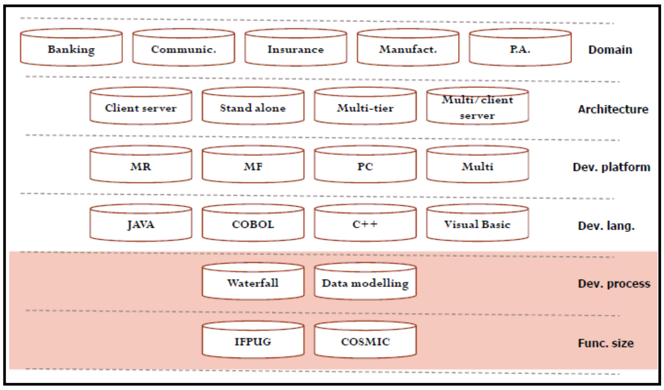


Figure 24: clustering by Development process and Functional measurement approach

Development process and Functional measurement approach: Waterfall and COSMIC<1000

In this case, the combinations that do not allow us to analyze the data are: (1) specification versus planning and (2) build versus design. We have a good correlation except for the build effort estimation where the correlation in negative. The result accuracy is not acceptable except in any case, as reported in Table 72.

	pearson	R^2	mmre	mdmre

spec. from plan.				
design from spec.	0.601	0.361	0.630	0.680
build from design				
test from build	0.511	0.261	0.560	0.550
deploy from test	0.414	0.172	1.440	1.240

Table 72: previous phase vs next phase - Waterfall and COSMIC<1000

Comparing the model obtained for all projects with the one obtained clustered by Waterfall and this cluster, we don't find an improvement of the accuracy in every case(see Table 12 and Table 29).

Development process and Functional measurement approach: Data modelling and IFPUG<1000

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. We have a good correlation even if the accuracy is not, as reported in Table 73.

	pearson	**R^2**	**mmre**	**Mdmre**
spec. from plan.	0.464	0.216	1.138	0.724
design from spec.				
build from design				
test from build	0.833	0.695	0.618	0.450
deploy from test	0.584	0.342	0.680	0.536

Table 73: previous phase vs next phase - Data modelling and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Waterfall and this cluster, we find an improvement of the accuracy for the deployment effort estimation (see Table 12 and Table 30).

Clustering by Development process and Programming language

Here we select Development process and Programming language (see Figure 25). Following the results.

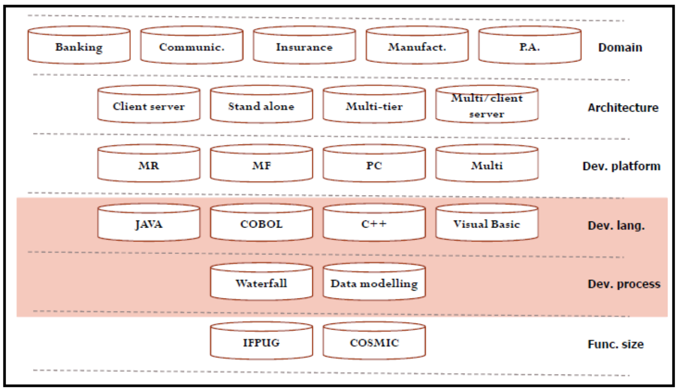


Figure 25: clustering by Development process and Programming language

Development process and Programming language: Data modelling and COBOL

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. We have a good correlation especially for the test phase, even if the accuracy is not, as reported in Table 74.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.416	0.173	2.143	1.654
design from spec.				
build from design				
test from build	0.817	0.667	0.532	0.411
deploy from test	0.648	0.420	1.054	0.822

Table 74: previous phase vs next phase - Data modelling and COBOL

Comparing the model obtained for all projects with the one obtained clustered by COBOL and this cluster, we don't find an improvement of the accuracy for the deployment effort estimation (see Table 12 and Table 30).

Clustering by Programming language and Functional measurement approach

Here we select is Programming language and Functional measurement approach (see Figure 26 as common caractheristics. Following the results obtained.

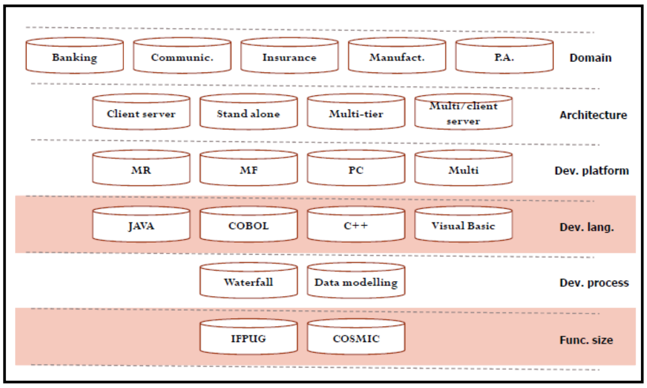


Figure 26: clustering by Programming language and Functional measurement approach

Programming language and Functional measurement approach: Java and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) build versus design and (2) test versus build. Here we have a high correlation and the accuracy is acceptable for the test phase, as reported in Table 75.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.				
build from design	0.877	0.769	0.543	0.432
test from build	0.907	0.823	0.324	0.398
deploy from test				

Table 75: previous phase vs next phase - Java and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Java and this cluster, we find an improvement of the accuracy (see Table 12 and Table 25).

Programming language and Functional measurement approach: Java and COSMIC<1000

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. In our results we have a high correlation for the design and build phases while for the test and deployment one is low, as reported in Table 76.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.741	0.549	0.678	0.509
build from design	0.899	0.809	0.345	0.301
test from build	0.192	0.037	1.098	0.932
deploy from test	0.197	0.039	1.238	1.092

Table 76: previous phase vs next phase - Java and COSMIC<1000

Comparing the model obtained for all projects with the one obtained clustered by Java and this cluster, we find an improvement of the accuracy only for build phase(see Table 12 and Table 25).

Programming language and Functional measurement approach: COBOL and IFPUG<1000

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. We have a high correlation for the specification phase while for other is low. The estimation accuracy is not acceptable for each case, as reported in Table 77.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.800	0.639	0.921	0.876
design from spec.				
build from design				
test from build	0.692	0.478	0.932	0.765
deploy from test	0.042	0.002	1.890	1.043

Table 77: previous phase vs next phase - COBOL and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by COBOL and this cluster, we find an improvement of the accuracy only for specification phase (see Table 12 and Table 26).

Programming language and Functional measurement approach: COBOL and COSMIC≥1000

In this case the only combination that does not allow us to analyze the data is the specification phase versus planning one. In the results we have a good correlation for the build phase while for other is low. The estimation accuracy is not acceptable for each case, as reported in Table 78.

	pearson	R^2	mmre	mdmre
spec. from plan.				
design from spec.	0.194	0.038	1.021	0.974
build from design	0.623	0.388	0.921	0.732
test from build	0.439	0.193	1.132	0.965
deploy from test	0.508	0.258	1.090	0.843

Table 78: previous phase vs next phase - COBOL and COSMIC >=1000

Comparing the model obtained for all projects with the one obtained clustered by COBOL and this cluster, we find an improvement of the accuracy only for build and deployment phases (see Table 12 and Table 26).

Programming language and Functional measurement approach: Visual basic and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) specification versus planning and (2) test versus build. Here we have a good correlation for the specification phase and high for the test one. The estimation accuracy is acceptable only for test phase, as reported in Table 79.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.655	0.429	0.921	0.843
design from spec.				
build from design				
test from build	0.978	0.957	0.245	0.301
deploy from test				

Table 79: previous phase vs next phase - Visual basic and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Visual basic and this cluster, we find an improvement of the accuracy only for test phase (see Table 12 and Table 28).

Clustering by three characteristic

Then we investigated the correlation between a phase and the next one clustering the project by three common caractheristics.

Clustering by Domain and Architecture and Development platform

The first combination selected is Domain and Architecture and Development platform (see Figure 27) as common caractheristics. Following the results obtained.

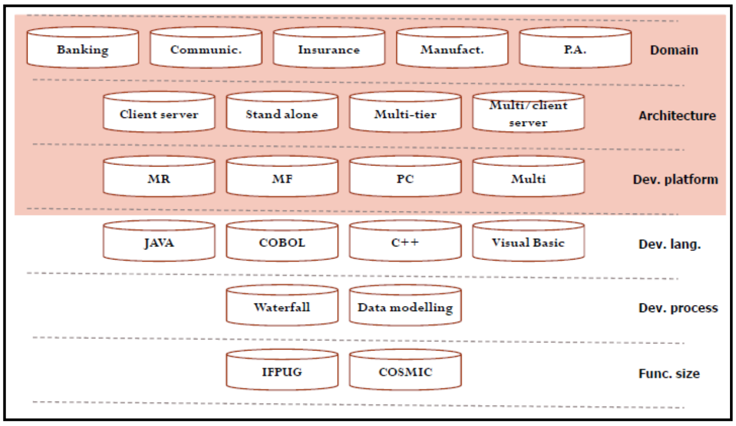


Figure 27: clustering by Domain and Architecture and Development platform

Domain and Architecture and Development platform: Banking and Stand alone and MF

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. in our results we have a high correlation with an estimation accuracy acceptable in every case as reported in Table 80.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.918	0.843	0.301	0.298
design from spec.				
build from design				
test from build	0.949	0.901	0.289	0.234
deploy from test	0.859	0.738	0.389	0.298

Table 80: previous phase vs next phase - Banking and Stand alone and MF

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) the one obtained clustered by Banking and Stand alone and (3) this cluster, we find an improvement of the accuracy (see Table 12, Table 13 and Table 36).

Clustering by Domain and Development platform and Development process

Clustering by Domain and Development platform and Programming language (see Figure 28) as common caractheristics. Following the results obtained.

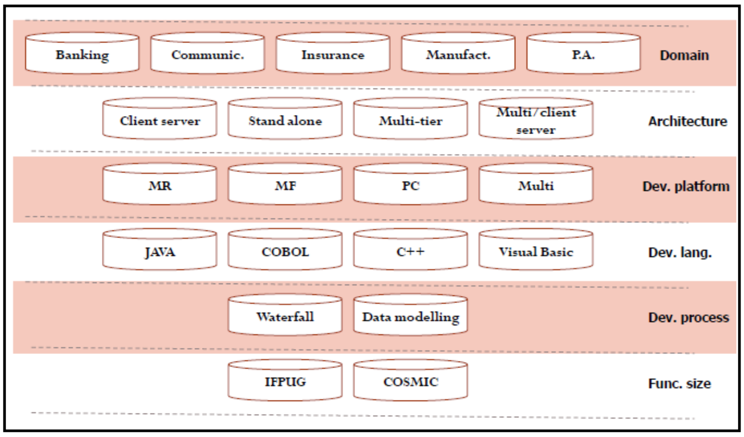


Figure 28: clustering by Domain and Development platform and Development process

Domain and Development platform and Development process: Banking and MF and Data modelling

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. We have a high correlation even if the accuracy is not acceptable in any case, as reported in Table 81.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.946	0.895	0.560	0.520
design from spec.				
build from design				
test from build	0.834	0.696	0.500	0.430
deploy from test	0.730	0.533	0.650	0.650

Table 81: previous phase vs next phase - Banking and MF and Data modelling

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) the one obtained clustered by Banking and MF and (3) this cluster, we find an improvement of the accuracy for the test and deployment phases (see Table 12, Table 13 and Table 38).

Clustering by Architecture and Development platform and Development process

Clustering by Architecture and Development platform and Functional measurement approach (see Figure 29) as common caractheristics. Following the results obtained.

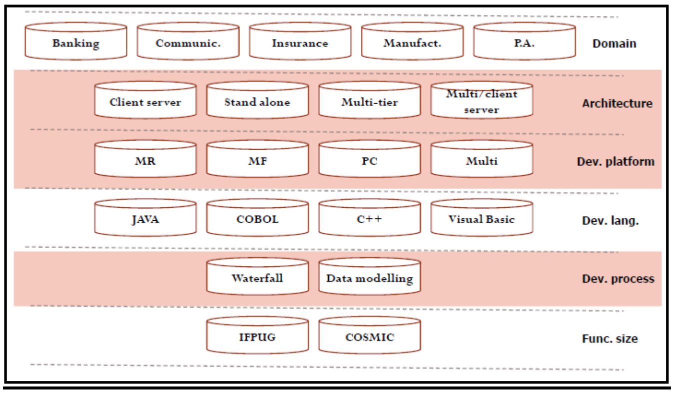


Figure 29: clustering by Architecture and Development platform and Development process

Architecture and Development platform and Development process: Stand alone and MF and Data modelling

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. In our results we have a high correlation except for test phase, even if the accuracy is acceptable only in deployment effort estimation, as reported in Table 82.

	pearson	**R^2**	**mmre**	**mdmre**
spec. from plan.	0.718	0.516	0.934	0.823
design from spec.				
build from design				
test from build	0.465	0.216	1.176	1.177
deploy from test	0.933	0.870	0.256	0.247

Table 82: previous phase vs next phase - stand alone and MF and data modelling

Comparing the model obtained for all projects with (1) the one obtained clustered by Stand alone, (2) the one obtained clustered by Stand alone and MF and (3) this cluster, we find an improvement of the accuracy for the deployment phase (see Table 12, Table 12 and Table 55).

Architecture and Development platform and Development process: Stand alone and PC and Data modelling

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. In our results we have a high correlation only for the deployment phase, even if the accuracy is not acceptable, as reported in Table 83.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.699	0.489	0.876	0.743
design from spec.				
build from design				
test from build	0.491	0.242	1.098	0.954
deploy from test	0.850	0.723	0.543	0.489

Table 83: previous phase vs next phase - Stand alone and PC and Data modelling

Comparing the model obtained for all projects with (1) the one obtained clustered by Stand alone, (2) the one obtained clustered by Stand alone and PC and (3) this cluster, we find an improvement of the accuracy for the deployment phase (see Table 12, Table 19 and Table 56).

Clustering by Architecture and Development platform and Programming language

Clustering by Architecture and Development platform and Programming language (see Figure 30) as common caractheristics. Following the results obtained.

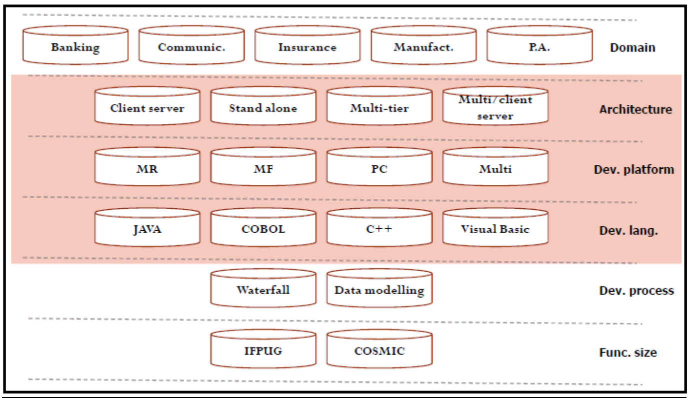


Figure 30: clustering by Architecture and Development platform and Programming language

Architecture and Development platform and Programming language: Stand alone and MF and COBOL

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design . in this cluster we have a high correlation only for the specification and deployment phases, even if the accuracy is acceptable only for the deployment phase, as reported in Table 84.

	pearson	R^2	mmre	mdmre
spec. from plan.	0.710	0.505	0.892	0.743
design from spec.				
build from design				
test from build	0.599	0.312	1.043	0.972
deploy from test	0.838	0.702	0.296	0.278

Table 84: previous phase vs next phase - stand alone and MF and COBOL

Comparing the model obtained for all projects with (1) the one obtained clustered by Stand alone, (2) the one obtained clustered by Stand alone and MF and (3) this cluster, we find an improvement of the accuracy for the deployment phase (see Table 12, Table 19 and Table 55).

<u>Clustering by Development platform and Development process and Functional measurement approach</u>

Clustering by Development platform and Development process and Functional measurement approach(see Figure 31) as common caractheristics. Following the results obtained.

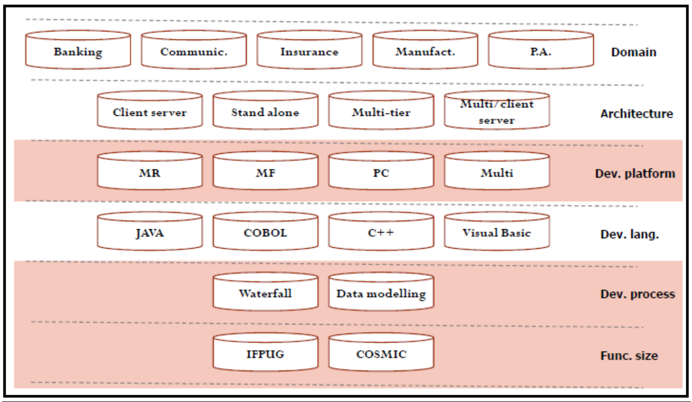


Figure 31: clustering by Development platform and Development process and Functional measurement approach

Development platform and Development process and Functional measurement approach: MF and Data modelling and IFPUG.

In this case, the combinations that allow us to analyze the data are: (1) design versus specification and (2) build versus design. In this cluster we have a high correlation only for the test phase, even if the accuracy is not acceptable in every case, as reported in Table 85.

	person	R^2	mmre	mdmre
spec. from plan.	0.459	0.211	0.940	0.690
design from spec.				
build from design				
test from build	0.844	0.713	0.430	0.340
deploy from test	0.645	0.416	0.880	0.690

Table 85: previous phase vs next phase - MF and Data modelling and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) the one obtained clustered by MF and Data modelling and (3) this cluster, we find an improvement of the accuracy for the test phase (see Table 12, Table 22 and Table 65).

Development platform and Development process and Functional measurement approach: MF and Data modelling and COSMIC<1000

In this case, the combinations that allow us to analyze the data are: (1) design versus specification and (2) build versus design. In our cluster we have a good correlation only for the design and test phases, for the build effort estimation the correlation is negative. The accuracy is not acceptable in any case, as reported in Table 86.

	person	R^2	mmre	mdmre
spec. from plan.	0.584	0.341	0.520	0.460
design from spec.				
build from design				
test from build	0.564	0.318	0.420	0.420
deploy from test	0.332	0.110	1.100	0.870

Table 86: previous phase vs next phase - MF and Data modelling and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) the one obtained clustered by MF and Data modelling and (3) this cluster, we find an improvement of the accuracy the specification and test phases (see Table 12, Table 22 and Table 65).

Development platform and Development process and Functional measurement approach: PC and Data modelling and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) design versus specification and (2) build versus design. In this cluster we have a good correlation except for the deployment phase, but the accuracy is acceptable only for the test phase, as reported in Table 87.

	person	R^2	mmre	mdmre
spec. from plan.	0.736	0.542	0.730	0.630
design from spec.				
build from design				
test from build	0.754	0.568	0.390	0.410
deploy from test	0.205	0.042	0.510	0.640

Table 87: previous phase vs next phase - PC and Data modelling and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by PC, (2) the one obtained clustered by PC and Data modelling and (3) this cluster, we find an improvement of the accuracy for every case (see Table 12, Table 23 and Table 67).

Clustering for four characteristic

We investigated the correlation between a phase and the next one analyzing the project for four common caractheristics.

<u>Clustering by Domain and Development platform and Development process and Functional measurement approach</u>

Clustering by Domain and Development platform and Development process and Functional measurement approach (see Figure 32) as common caractheristics. Following the results obtained.

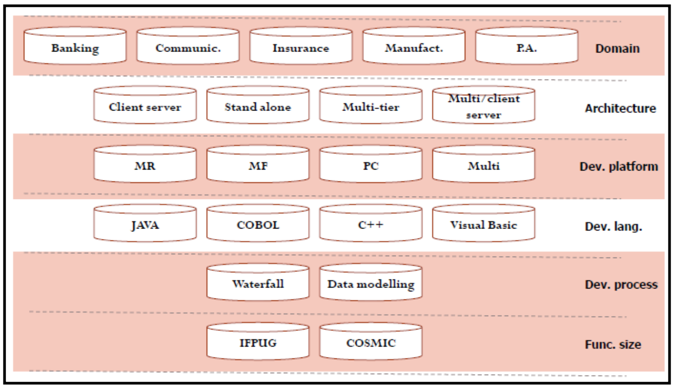


Figure 32: clustering by Domain and Development platform and Development process and Functional measurement approach

Domain and Development platform and Development process and Functional measurement approach: Banking and MF and Data modelling and IFPUG<1000.

In this case, the combinations that do not allow us to analyze the data are: (1) design versus specification and (2) build versus design. In our results we have a good correlation even if the accuracy is acceptable only for the specification and test phases, as reported in Table 88.

	pearson	R^2	mmre	mdmre
spec. from plan	0.919	0.845	0.330	0.310
design from spec.				
build from design				
test from build	0.841	0.707	0.340	0.310
deploy from test	0.628	0.394	0.430	0.440

Table 88: previous phase vs next phase - Banking and MF and Data modelling and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) the one obtained clustered by Banking and MF, (3) the one obtained clustered by Banking and MF and Data modelling and (4) this cluster, we find an improvement of the accuracy for every case (see Table 12,Table 13, Table 38 and Table 81).

Grouped by five common caractheristics

Clustering the project by five common characteristics doesn't allow to obtain any projects for the effort estimation.

Grouped by six common caractheristics

Clustering the project by six common characteristics doesn't allow to obtain any projects for the effort estimation.

RQ2: Is it possible to use the effort of one phase for estimating the remaining project effort?

RQ2 has been carried out without clustering project by type, domain or other characteristics, the results show promising models in certain phases as shown in

There is a good correlation between the effort of each phase and the sum of the efforts of the following ones, except if we consider as previous phase the design. The result shows good improvement of the prediction compared to the one obtained considering the estimation between a phase and the next one.

Considering the effort spent in one phase, it is also possible to estimate the remaining effort for the whole project with a similar error to that obtained when estimating only the next phase.

	pearson	**R^2**	**mmre**	**mdmre**
Sum remaining project from plan.	0.680	0.457	0.690	0.520
Sum remaining project from spec.	0.670	0.447	1.350	0.810
Sum remaining project from design	0.389	0.149	1.340	0.690
Sum remaining project from build	0.691	0.477	0.500	0,500

Table 89: previous phases vs remaing project

RQ 2.1: Does considering one common caractheristics, in addition to the effort for a phase, improve the effort prediction for the remaining project?

Here we want to use the effort of one phase for estimating the remaining project effort clustering the projects by each common caractheristics.

Moreover, in order to understand the estimation accuracy improves by grouping by one common caractheristics, we compare the R2, mmre and mdmre for RQ2.1.and RQ2 for each common caractheristics.

In some cases, there are not enough projects to draw statistical significant conclusions. For this reason, we left the column empty.

<u>**Clustering by one common characteristic**</u>

Here we investigated the effort correlation between a phase and the remaining project clustering by one common caractheristic.

Clustering by domain

Following we report the results obtained for the cluster where we select, as common characteristic, the domain: banking, communications, insurance, manufacturing and public administration.

Domain: Banking

Clustering by Banking domain, we obtain a good correlation when the previous phase are planning and design. The estimation accuracy is not acceptable in every case, as show in Table 90.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.735	0.540	0.881	0.943
Sum remaining project from spec.	0.392	0.154	1,290	0,919
Sum remaining project from design	0.704	0.495	0.675	0,520
Sum remaining project from build	0.548	0.300	0,880	0,736

Table 90: previous phase vs remaining project - Banking

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy when the previous phase is the design one. Instead when we consider specification phase for estimating the remaining project the correlation decrease (see Table 89).

Domain: Communications

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In this cluster we obtain a good correlation when the previous phase is the specification one even if the estimation accuracy is not acceptable in every case, as show in Table 91.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.917	0.841	0.654	0.578
Sum remaining project from design				
Sum remaining project from build	0.497	0.247	1.109	1.076

Table 91: previous phase vs remaining project - Communications

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy when the previous phase is the specification one (see Table 89).

Domain: Insurance

In this case, the only combination that allows us to analyze the data is the sum remaining project versus planning phase. We don't obtain a good correlation with an estimation accuracy not acceptable in every case, as show in Table 92.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.376	0.141	1.346	1.198
Sum remaining project from design	0.343	0.117	1.246	1.076
Sum remaining project from build	0.219	0.048	1.438	1.257

Table 92: previous phase vs remaining project - Insurance

Comparing the model obtained for all projects with this cluster, we don't obtain an improvement of the correlation and the accuracy (see Table 89).

Domain: Manufacturing

In Manufacturing cluster we obtain a good correlation if we consider specification and build phases for estimating the remaining effort project. Taking into account the estimation accuracy, it is not acceptable in every case, as show in Table 93.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.518	0.268	1.190	1.021
Sum remaining project from spec.	0.810	0.656	0.832	0.721
Sum remaining project from design	0.532	0.283	1.256	1.055
Sum remaining project from build	0.767	0.588	0.921	0.833

Table 93: previous phase vs remaining project - Manufacturing

Comparing the model obtained for all projects with the this cluster, we obtain an improvement of the correlation and the accuracy when the previous phase is the specification one(see Table 89).

Domain: Public administration

In this case the only combinations that does not allow us to analyze the data is the sum remaining project versus planning phase. We obtain a good correlation if we consider specification and design phases for estimating the remaining effort project. Taking into account the estimation accuracy, it is not acceptable in every case, as show in Table 94.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.896	0.803	0.674	0.789
Sum remaining project from design	0.670	0.448	0.943	0.821
Sum remaining project from build	0.477	0.288	1.176	1.098

Table 94: previous phase vs remaining project - Public administration

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation in every case and for the accuracy when the previous phase is the specification and design (see Table 89).

Clustering by architecture

Here we report the results obtained for the second cluster where we select, as common characteristic, the architecture: Client server, stand alone, Multi tier and Multi tier/Client server.

Architecture: Client server

In this case the only combination that doesn't allow us to analyze the data is the sum remaining project versus build phase. In our result we obtain a good correlation if we consider specification and design phases for estimating the remaining effort project. Taking into account the estimation accuracy, it is not acceptable in every case, as show in Table 95.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.197	0.038	1.187	1.085
Sum remaining project from spec.	0.690	0.477	1.043	0.921
Sum remaining project from design	0.819	0.670	0.732	0.655
Sum remaining project from build				

Table 95: previous phase vs remaining project - Client server

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation in every case and for the accuracy when the previous phase is the design. Moreover we have a huge of drop of the correlation with the planning phase(see Table 89).

Architecture: Stand alone

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus design and (2) sum remaining project versus build. In our result we obtain a good correlation if we consider specification phase even if the estimation accuracy, it is not acceptable in every case, as show in Table 96.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.482	0.232	1.156	1.088
Sum remaining project from spec.	0.811	0.657	0.832	0.655
Sum remaining project from design				
Sum remaining project from build				

Table 96: previous phase vs remaining project - Stand alone

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy in specification case. Also we have a huge of drop of the correlation with the planning phase(see Table 89)

Architecture: Multi tier

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus design and (2) sum remaining project versus build. In our result we obtain a high correlation even if the goodness of fit is not acceptable in every case, as show in Table 97.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.828	0.686	0.460	0.521
Sum remaining project from spec.	0.940	0.884	0.478	0.432
Sum remaining project from design				
Sum remaining project from build				

Table 97: previous phase vs remaining project - Multi tier

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy in every case. (see Table 89)

Clustering by Development platform

Here we report the results obtained for the cluster where we select, as common characteristic, the Development platform: MF, MR, PC and Multi.

Development platform: MR

In this case the only combination that doesn't allow us to analyze the data is the sum remaining project versus build phase. In our result we obtain a high correlation when we consider as previous phase the specification and the design, even if the goodness of fit is acceptable only for the second one, as show in Table 98.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.475	0.225	0.876	0.921
Sum remaining project from spec.	0.851	0.725	0.421	0.367
Sum remaining project from design	0.965	0.931	0.287	0.251
Sum remaining project from build				

Table 98: previous phase vs remaining project - MR

Comparing the model obtained for all projects with this cluster , we obtain a dramatic improvement of the correlation and the accuracy in specification and design phases. (see Table 89)

Development platform: MF

In this case the only combination that does not allow us to analyze the data is the sum remaining project versus build phase. In our result we obtain a high correlation except if we consider as previous phase the planning one. The goodness of fit is not acceptable in every case, as show in Table 99.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.328	0.108	1.234	1.022
Sum remaining project from spec.	0.752	0.565	0.932	0.811
Sum remaining project from design	0.690	0.476	0.973	0.893
Sum remaining project from build				

Table 99: previous phase vs remaining project – MF

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy in specification and design phases. (see Table 89)

Development platform: PC

In this case the only combination that does not allow us to analyze the data is the sum remaining project versus build phase. In our result we obtain a high correlation except if we consider as previous phase the planning one. The goodness of fit is not acceptable in every case, as show in Table 100.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.203	0.041	0.921	0.855
Sum remaining project from spec.	0.724	0.524	0.933	0.721
Sum remaining project from design	0.701	0.492	1.044	1.156
Sum remaining project from build				

Table 100: previous phase vs remaining project – PC

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy in specification and design phases. (see Table 89)

Development platform: Multi

In this case the only combinations that does not allow us to analyze the data is the sum remaining project versus build phase. In our result we obtain a high correlation except if we consider as previous phase the planning and the specification. The goodness of fit is acceptable only in specification phase, as show in Table 101.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.886	0.786	0.578	0.688
Sum remaining project from spec.	0.895	0.802	0.301	0.289
Sum remaining project from design	0.508	0.258	1.277	1.133
Sum remaining project from build				

Table 101: previous phase vs remaining project – Multi

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and the accuracy in planning and specification phases. (see Table 89)

Clustering by Development process

Here we report the results obtained for the cluster where we select, as common characteristic, the Development process: Waterfall and Data modelling.

Development process: Data modelling

In this case the only combinations that does not allow us to analyze the data is the sum remaining project versus planning phase. In our result we don't obtain a good correlation and the goodness of fit is acceptable, as show in Table 102.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.330	0.109	1.123	1.047
Sum remaining project from spec.				
Sum remaining project from design				
Sum remaining project from build				

Table 102: previous phase vs remaining project - Data modelling

Clustering by Programming language

Here we report the results obtained for the cluster where we select, as common characteristic, the Programming language: Java, COBOL; C++ and Visual basic.

Programming language: Java

In this case the only combinations that does not allow us to analyze the data is the sum remaining project versus build phase. In our result we obtain a good correlation if we consider as previous phase the specification one. The goodness of fit is not acceptable in every case, as show in Table 103.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.352	0.124	1.234	1.356
Sum remaining project from spec.	0.658	0.433	1.087	1.144
Sum remaining project from design	0.406	0.164	1.430	1.234
Sum remaining project from build				

Table 103: previous phase vs remaining project - Java

Comparing the model obtained for all projects with this cluster, we don't obtain an improvement of the correlation and the of accuracy. (see Table 89)

Programming language: COBOL

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus build. In our result we obtain a good correlation even if the goodness of fit is not acceptable in every case, as show in Table 104.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.754	0.569	0.965	0.877
Sum remaining project from design	0.746	0.556	0.934	0.811
Sum remaining project from build				

Table 104: previous phase vs remaining project - Cobol

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and of the accuracy. (see Table 89)

Programming language: C++

In this case the only combination that allows us to analyze the data is the sum of remaining project versus design phase. In our result we obtain a good correlation even if the goodness of fit is not acceptable, as show in Table 105.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.707	0.499	0.877	0.799
Sum remaining project from build				

Table 105: previous phase vs remaining project - C++

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and of the accuracy. (see Table 89)

Programming language: Visual basic

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus design. In our result we obtain a good correlation even if the goodness of fit is not acceptable, as show in Table 106.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.917	0.841	0.689	0.566
Sum remaining project from design	0.619	0.383	1.098	1.156
Sum remaining project from build				

Table 106: previous phase vs remaining project - Visual Basic

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and of the accuracy. (see Table 89)

Clustering by Functional measurement approach

Here we report the results obtained for the second cluster where we select, as common characteristic, the Functional measurement approach: IFPUG and COSMIC, we consider for each case either the value minor to 1000 and major or equal to 1000.

Functional measurement approach: IFPUG < 1000

In this case the only combinations that does not allow us to analyze the data is the sum of remaining projects versus build phase. In our result we obtain a good correlation even if the goodness of fit is not acceptable, as show in Table 107.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.609	0.371	0.987	0.866
Sum remaining project from spec.	0.475	0.235	1.188	1.098
Sum remaining project from design	0.663	0.440	0.921	0.845
Sum remaining project from build				

Table 107: previous phase vs remaining project – IFPUG < 1000

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and of the accuracy in design case. (see Table 89)

Functional measurement approach: IFPUG ≥ 1000

In this case the only combinations that does not allow us to analyze the data is the sum of remaining projects versus build phase. In our result we obtain a good correlation except for the planning phase even if the goodness of fit is not acceptable, as show in Table 108.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.222	0.049	1.178	1.033
Sum remaining project from spec.	0.704	0.495	0.943	0.807
Sum remaining project from design	0.688	0.473	0.941	0.871
Sum remaining project from build				

Table 108: previous phase vs remaining project – IFPUG > 1000

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and of the accuracy in specification and design case. (see Table 89)

Functional measurement approach: COSMIC < 1000

In this case the only combinations that doesn't allow us to analyze the data is the sum of remaining projects versus build phase. In our result we obtain a good correlation except for the planning phase even if the goodness of fit is not acceptable, as show in Table 109.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.377	0.142	1.218	1.130
Sum remaining project from spec.	0.603	0.363	1.087	0.867
Sum remaining project from design	0.667	0.444	0.974	0.802
Sum remaining project from build				

Table 109: previous phase vs remaining project – COSMIC < 1000

Comparing the model obtained for all projects with this cluster, we obtain an improvement of the correlation and of the accuracy in specification and design case. (see Table 89)

Grouped by two common characteristics

Based on the results obtained grouped by one common characteristics we refined the analysis clustering the project by two common characteristics.

Clustering by Domain and Architecture

Here we select Domain and Architecture (see Figure 14). Following the results.

Domain and Architecture: Banking and Stand alone

In this case the only combinations that does not allow us to analyze the data is the sum of remaining projects versus design phase. In our result we obtain a high correlation except for the planning phase even if the goodness of fit is acceptable only for the specification one, as show in Table 110.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.940	0.883	0.546	0.466
Sum remaining project from spec.	0.970	0.941	0.276	0.233
Sum remaining project from design				
Sum remaining project from build	0.925	0.857	0.521	0.409

Table 110: previous phase vs remaining project - Banking and Stand alone

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy in specification and design case. (see Table 89 and Table 90)

Domain and Architecture: Communications and Stand alone

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In our result we obtain a high correlation except for the planning phase with an acceptable accuracy, as show in Table 110.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.977	0.954	0.298	0.231
Sum remaining project from design				
Sum remaining project from build	0.923	0.852	0.278	0.262

Table 111: previous phase vs remaining project - Stand alone and Communications

Comparing the model obtained for all projects with (1) the one obtained clustered by Communications, (2) and this cluster, we obtain a dramatic improvement of the correlation and of the accuracy. (see Table 89 and Table 91)

Clustering by Domain and Developed platform

Here we select Domain and Development platform (see Figure 15). Following the results.

Domain and Development platform: Banking and MR

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In our result we don't obtain a good correlation without an acceptable accuracy, as show in Table 112.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.389	0.151	1.237	1.187
Sum remaining project from design				
Sum remaining project from build	0.464	0.215	1.054	1.178

Table 112: previous phase vs sum remaining project – Banking and MR

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we obtain a huge of drop of the correlation and of the accuracy. (see Table 89 and Table 90)

Domain and Development platform: Banking and MF

In this case the only combinations that does not allow us to analyze the data is the sum of remaining projects versus specification phase. In our result we obtain a good correlation even if the goodness of fit is acceptable only for the planning phase, as show in Table 113.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.906	0.906	0.245	0.290
Sum remaining project from spec.				
Sum remaining project from design	0.720	0.518	0.943	0.800
Sum remaining project from build	0.619	0.383	1.198	1.021

Table 113: previous phase vs remaining project - MF and Banking

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we obtain a huge of drop of the correlation and of the accuracy for the planning phase(see Table 89 and Table 90)

Domain and Development platform: Banking and PC

In this cluster we obtain a good correlation especially for the first analysis, even if the goodness of fit is not acceptable, as show in Table 114.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.691	0.478	0.831	0.798
Sum remaining project from spec.	0.404	0.163	1.432	1.398
Sum remaining project from design	0.560	0.314	1.098	1.177
Sum remaining project from build	0.449	0.201	1.334	1.177

Table 114: previous phase vs remaining project - Banking and PC

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy , while comparing with the one without cluster only for the design phase (see Table 89 and Table 90).

Domain and Development platform: Banking and Multi

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus design and (2) sum remaining project versus build. In this cluster we obtain a good correlation even if the goodness of fit is not acceptable, as show in Table 115.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0,624	0,389	0.986	0.888
Sum remaining project from build	0,563	0,317	1.348	1.290

Table 115: previous phase vs remaining project - Banking and Multi

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy , while comparing with the one without cluster only for the design phase (see Table 89 and Table 90).

Domain and Development platform: Communications and MR

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In this cluster we obtain a good correlation with an acceptable goodness of fit, as show in Table 116.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.955	0.913	0.289	0.210
Sum remaining project from design				
Sum remaining project from build	0.939	0.881	0.276	0.323

Table 116: previous phase vs remaining project - Communications and MR

Comparing the model obtained for all projects with (1) the one obtained clustered by Communications, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy especially when we consider the specification phase (see Table 89 and Table 91).

Domain and Development platform: Communications and PC

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In this cluster we obtain a good correlation with an acceptable goodness of fit, only for the build phase considered as show in Table 117.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.294	0.087	1.489	1.187
Sum remaining project from design				
Sum remaining project from build	0.937	0.877	0.298	0.379

Table 117: previous phase vs remaining project - Communications and PC

Comparing the model obtained for all projects with (1) the one obtained clustered by Communications, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only when we consider the build phase (see Table 89 and Table 91).

Domain and Development platform: Insurance and MF

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In this cluster we obtain a good correlation with an acceptable goodness of fit, only for the specification phase considered as show in Table 118.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.939	0.881	0.298	0.340
Sum remaining project from design				
Sum remaining project from build	0.009	0.008	1.987	1.567

Table 118: previous phase vs remaining project - Insurance and MF

Comparing the model obtained for all projects with (1) the one obtained clustered by Insurance, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only when we consider the specification phase (see Table 89 and Table 92).

Domain and Development platform: Manufacturing and PC

In this case the only combinations that does not allow us to analyze the data is the sum of remaining projects versus planning phase. In this cluster we obtain a good correlation only for the specification phase, without an acceptable goodness of fit, as show in Table 119.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.857	0.735	0.678	0.876
Sum remaining project from design	0.361	0.130	1.278	1.178
Sum remaining project from build	0.748	0.559	0.921	0.754

Table 119: previous phase vs remaining project - Manufacturing and PC

Comparing the model obtained for all projects with (1) the one obtained clustered by Manufacturing, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 93).

Clustering by Domain and Functional measurement approach

Here we select Domain and Functional measurement approach (see Figure 17). Following the results.

Domain and Functional measurement approach: Banking and IFPUG<1000

In this case the only combinations that does not allow us to analyze the data is the sum of remaining project versus design phase. In this cluster we obtain a good correlation with an acceptable goodness of fit, only for the build phase considered as show in Table 120.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.609	0.371	1.098	0.921
Sum remaining project from spec.	0.779	0.607	0.854	0.721
Sum remaining project from design				
Sum remaining project from build	0.470	0.221	1.198	1.289

Table 120: previous phase vs remaining project - Banking and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only when we consider the specification phase (see Table 89 and Table 90).

Domain and Functional measurement approach: Banking and COSMIC<1000

In this cluster we obtain a good correlation with an acceptable goodness of fit, only for the build phase considered as show in Table 121.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.377	0.142	1.234	1.076
Sum remaining project from spec.	0.516	0.266	1.189	0.931
Sum remaining project from design	0.741	0.549	0.953	0.886
Sum remaining project from build	0.497	0.247	1.143	1.008

Table 121: previous phase vs remaining project – Banking and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Banking, (2) and this cluster, we don’t obtain an improvement of the correlation and of the accuracy, while considering the one without cluster we have an improvement when we consider the design phase (see Table 89 and Table 90).

Domain and Functional measurement approach: Communications and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus specification and (2) sum remaining project versus build. In this cluster we obtain a good correlation but the goodness of fit is not acceptable, as show in Table 122.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.732	0.535	0.978	0859
Sum remaining project from design				
Sum remaining project from build	0.869	0.755	0.879	0.748

Table 122: previous phase vs remaining project - Communications and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Communications, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only when we consider the build phase (see Table 89 and Table 91).

Domain and Functional measurement approach: Manufacturing and IFPUG<1000

In this case the only combinations that does not allow us to analyze the data is the sum of remaining project versus specification phase. In this cluster we obtain a good correlation for design and build phases, but the goodness of fit is not acceptable in every case, as show in Table 123.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.313	0.098	1.436	1.359
Sum remaining project from spec.				
Sum remaining project from design	0.795	0.632	0.875	0.659
Sum remaining project from build	0.616	0.380	1.023	0.956

Table 123: previous phase vs remaining project - Manufacturing and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Manufacturing, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only when we consider the design phase (see Table 89 and Table 123).

Clustering by Architecture and Development platform

Here we select Architecture and Development platform (see Figure 18). Following the results.

Architecture and Development platform: Client server and MR

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus design and (2) sum remaining project versus build. In this cluster we don't obtain a good correlation with a not acceptable goodness of fit, as show in Table 123.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.475	0.225	1.187	0.981
Sum remaining project from spec.	0.475	0.225	1.234	1.047
Sum remaining project from design				
Sum remaining project from build				

Table 124: previous phase vs remaining project – Client server and MR

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 95).

Architecture and Development platform: Client server and PC

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus design and (2) sum remaining project versus build. In this cluster we obtain a good correlation only for the specification phase, even if the goodness of fit is not acceptable, as show in Table 123.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.065	0.004	1.987	1.436
Sum remaining project from spec.	0.690	0.476	1.098	0.963
Sum remaining project from design				
Sum remaining project from build				

Table 125: previous phase vs remaining project – Client server and PC

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see **Table 89** and **Table 95**).

Clustering by Architecture and Development process

Here we select Architecture and Development process (see Figure 19). Following the results.

Architecture and Development process: Client server and Waterfall

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus design. In this cluster we obtain a good correlation in every case, even if the goodness of fit is not acceptable, as show in Table 126.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.628	0.395	0.894	0.698
Sum remaining project from spec.				
Sum remaining project from design	0.792	0.627	0.798	0.699
Sum remaining project from build				

Table 126: previous phase vs remaining project – Client server and Waterfall

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) and this cluster, we don't obtain a correlation improvement, even if with the one without cluster we obtain an improvement of the correlation and of the accuracy for the planning phase (see Table 89 and Table 95).

Architecture and Programming language

Here we select Architecture and Programming language (see Figure 20). Following the results.

Architecture and Programming language: Client server and Java

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus design. In this cluster we obtain a good correlation for the design phase, even if the goodness of fit is not acceptable, as show in Table 127.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.479	0.229	1.036	0.985
Sum remaining project from spec.				
Sum remaining project from design	0.864	0.746	0.769	0.622
Sum remaining project from build				

Table 127: previous phase vs remaining project – Client server and Java

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) and this cluster, we don't obtain a correlation improvement, even if with the one without cluster we obtain an improvement of the correlation and of the accuracy for the planning phase (see Table 89 and Table 95).

Clustering by Architecture and Functional measurement approach

Here we select Architecture and Functional measurement approach (see Figure 21). Following the results.

Architecture and Functional measurement approach: Client server and IFPUG<1000

In this case the only combinations that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we obtain a good correlation for the design phase, even if the goodness of fit is not acceptable, as show in Table 128.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.874	0.763	0.759	0.601
Sum remaining project from build				

Table 128: previous phase vs remaining project – Client server and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) and this cluster, we don't obtain a correlation improvement, even if with the one without cluster we obtain an improvement of the correlation and of the accuracy for the planning phase (see Table 89 and Table 95).

Architecture and Functional measurement approach: Client server and COSMIC<1000

In this case the only combinations that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we obtain a good correlation for the design phase, even if the goodness of fit is not acceptable, as show in Table 129.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.756	0.571	0.796	0.602
Sum remaining project from build				

Table 129: previous phase vs remaining project – Client server and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) and this cluster, we don't obtain a correlation improvement, even if with the one without cluster we obtain an improvement of the correlation and of the accuracy for the planning phase (see Table 89 and Table 95).

Clustering by Development platform and Development process

Here we select Development platform and Development process (see Figure 22). Following the results.

Development platform and Development process: MF and Waterfall

In this case the only combinations that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we don't obtain a good correlation for the design phase, with a not acceptable goodness of fit, as show in Table 130.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.365	0.133	1.536	1.190
Sum remaining project from build				

Table 130: previous phase vs remaining project – MF and Waterfall

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) and this cluster, we don't obtain a correlation improvement (see Table 89 and Table 99).

Development platform and Development process: Multi and Waterfall

In this case the only combinations that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we obtain a good correlation for the design phase, with an acceptable goodness of fit, as show in Table 131.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.967	0.934	0.249	0.209
Sum remaining project from build				

Table 131: previous phase vs remaining project – Multi and Waterfall

Comparing the model obtained for all projects with (1) the one obtained clustered by Multi, (2) and this cluster, we obtain a dramatic improvement of the correlation and of the accuracy (see Table 89 and Table 99).

Development platform and Development process: PC and Waterfall

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus design. In this cluster we don't obtain a good correlation and the goodness of fit is not acceptable, as show in Table 132.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.446	0.199	1.103	0.963
Sum remaining project from spec.				
Sum remaining project from design	0.554	0.307	1.049	0.947
Sum remaining project from build				

Table 132: previous phase vs remaining project – PC and Waterfall

Comparing the model obtained for all projects with (1) the one obtained clustered by PC, (2) and this cluster, we don' obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 100).

Clustering by Development platform and Programming language

Here we select Development platform and Programming language (see Figure 23). We don't obtain projects containing value for the effort spent per each development phase.

Clustering by Development platform and Functional measurement approach

Here we select Development platform and Functional measurement approach (see Figure 23) Following the results

Development platform and Functional measurement approach: MF and IFPUG<1000

In this case the only combinations that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we don't obtain a good correlation and the goodness of fit is not acceptable, as show in Table 133.

	pearson	**R^2**	**mmre**	**mdmre**
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.703	0.495	0.943	0.801
Sum remaining project from build				

Table 133: previous phase vs remaining project – MF and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) and this cluster, we don' obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 99)

Development platform and Functional measurement approach: MF and COSMIC<1000

In this case the only combinations that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we obtain a good correlation and the goodness of fit is not acceptable, as show in Table 134.

	pearson	**R^2**	**mmre**	**mdmre**
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.625	0,391	0.921	0.856
Sum remaining project from build				

Table 134: previous phase vs remaining project – MF and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) and this cluster, we don' obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 99)

Development platform and Functional measurement approach: PC and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus design. In this cluster we don't obtain a good correlation and the goodness of fit is not acceptable, as show in Table 135.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.546	0.298	0.926	0.723
Sum remaining project from spec.				
Sum remaining project from design	0.553	0.306	0.963	0.800
Sum remaining project from build				

Table 135: previous phase vs remaining project – PC and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by PC, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only in design phase (see Table 89 and Table 100).

Development platform and Functional measurement approach: PC and COSMIC<1000

In this case the only combination that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we obtain a good correlation with an acceptable goodness of fit, as show in Table 136.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.959	0.921	0.246	0.324
Sum remaining project from build				

Table 136: previous phase vs remaining project – PC and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by PC, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy only in design phase (see Table 89 and Table 100).

<u>Clustering by Development process and Functional measurement approach</u>

Here we select Development process and Functional measurement approach (see Figure 24). Following the results.

Development process and Functional measurement approach: Data modelling and IFPUG<1000

In this case the only combination that allows us to analyze the data is the sum of remaining project versus planning phase. In this cluster we obtain a high correlation with an acceptable goodness of fit, as show in Table 137.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.821	0.674	0.240	0.298
Sum remaining project from spec.				
Sum remaining project from design				
Sum remaining project from build				

Table 137: previous phase vs remaining project – Data modelling and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Data modelling, (2) and this cluster, we obtain a dramatic improvement of the correlation and of the accuracy (see Table 89 and Table 102).

Development process and Functional measurement approach: Data modelling and IFPUG≥1000

In this case the only combination that allows us to analyze the data is the sum of remaining project planning phase. In this cluster we don't obtain a good correlation and not an acceptable goodness of fit, as show in Table 138.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.076	0.006	1.969	1.654
Sum remaining project from spec.				
Sum remaining project from design				
Sum remaining project from build				

Table 138: previous phase vs remaining project – Data modelling and IFPUG>=1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Data modelling, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 102).

Clustering by Programming language and Functional measurement approach

Here we select Programming language and Functional measurement approach (see Figure 26). Following the results.

Programming language and Functional measurement approach: Java and IFPUG<1000

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus design. In this cluster we obtain a high correlation even if the goodness of fit is acceptable only in design phase, as show in Table 139.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.875	0.765	0.645	0.421
Sum remaining project from spec.				
Sum remaining project from design	0.909	0.826	0.325	0.256
Sum remaining project from build				

Table 139: previous phase vs remaining project – Java and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Java, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 103).

Programming language and Functional measurement approach: Java and COSMIC<1000

In this case, the combinations that allow us to analyze the data are: (1) sum remaining project versus planning and (2) sum remaining project versus design. In this cluster we obtain a high correlation even if the goodness of fit is acceptable only in design phase, as show in Table 140.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.	0.840	0.705	0.569	0.421
Sum remaining project from spec.				
Sum remaining project from design	0.920	0.840	0.298	0.245
Sum remaining project from build				

Table 140: previous phase vs remaining project – Java and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Java, (2) and this cluster, we obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 102).

Programming language and Functional measurement approach: COBOL and COSMIC<1000

In this case the only combination that allows us to analyze the data is the sum of remaining project versus design phase. In this cluster we don't obtain a good correlation and not an acceptable goodness of fit, as show in Table 141.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.				
Sum remaining project from design	0.488	0.238	1.234	1.023
Sum remaining project from build				

Table 141: previous phase vs remaining project – COBOL and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by COBOL, (2) and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 89 and Table 104).

Grouped by three common caractheristics

Based on the results obtained grouped by two common characteristics we refined the analysis clustering the project by three common characteristics.

<u>Clustering by Architecture and Development platform and Functional measurement approach</u>

Here we select Architecture and Development platform and Functional measurement approach (see Figure 30). Following the results.

Architecture and Development platform and Functional measurement approach: Client server and Multi and COSMIC<1000

In this case the only combination that allows us to analyze the data is the sum of remaining project versus specification phase. In Table 143 we show the result for this cluster where we have a good correlation even if the accuracy is not acceptable.

	pearson	R^2	mmre	mdmre
Sum remaining project from plan.				
Sum remaining project from spec.	0.645	0.416	0.793	0.802
Sum remaining project from design				
Sum remaining project from build				

Table 142: previous phase vs remaining project – Client server and Multi and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Stand alone, (2) the one obtained clustered by Stand alone and Multi and (3) this cluster, we find a dramatic improvement of the correlation and the estimation accuracy.

Clustering by Architecture and Development process and Functional measurement approach
Here we select Architecture and Development process and Functional measurement approach (see). Following the results.

Architecture and Development platform and Functional measurement approach: Stand alone and Data modelling and IFPUG<1000

In this case the only combination that allows us to analyze the data is the sum of remaining project versus planning phase. In Table 143 we show the result for this cluster where we have an high correlation even if the accuracy is not acceptable.

	pearson	**R^2**	**mmre**	**mdmre**
Sum remaining project from plan.	0.937	0.038	1.395	1.428
Sum remaining project from spec.				
Sum remaining project from design				
Sum remaining project from build				

Table 143: previous phase vs remaining project – Stand alone and Data modelling and IFPUG<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by Stand alone, (2) the one obtained clustered by Stand alone and Data modelling and (3) this cluster, we don't find an improvement of the correlation but no for the estimation accuracy.

Grouped by four common caractheristics

Clustering the project by four common characteristics doesn't allow to obtain any projects for the effort estimation.

Grouped by five common caractheristics

Clustering the project by five common characteristics doesn't allow to obtain any projects for the effort estimation.

Grouped by six common caractheristics

Clustering the project by six common characteristics doesn’t allow to obtain any projects for the effort estimation.

RQ3: Is it possible to use the effort spent up to a development phase to estimate its effort?

RQ3 has been carried out without clustering project by type, domain or other characteristics, the results show promising models in certain phases as shown in Table 144. Considering more than one phase before allow to obtain more accuracy results in some cases but this approach proves to be unless in the other cases.

When we estimate the effort for the design phase, taking in to account the two phases before is the best solution in term of correlation and estimation accuracy. Instead for the build phase reckoning with all the previous phases make higher the correlation even if the estimation accuracy became worse. Contrarily estimating the effort for the test phase the correlation became worse with an improvement of the accuracy. Caractheristics results can be seen in case of deployment effort estimation, here looking for the correlation we obtain a dramatic improvement, while the accuracy enhances only if the previous phase are test plus build and test plus build plus design.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.705	0.497	0.660	0.530
build from design+spec.	0.525	0.275	2.930	2.090
build from design+spec.+plan.	0.760	0.577	2.650	1.830
test. from build+design	0.460	0.211	0.720	0.670
test. from build+design +spec.	0.500	0.250	0.720	0.710
test. from build +design+spec.+plan.	0.440	0.197	0.680	0.730
deploy. from test.+build	0.777	0.604	0.860	0.920
deploy. from test.+build+design	0.715	0.511	0.930	0.960
deploy. from test.+build+design +spec.	0.662	0.439	1.300	1.160
deploy. from test. +build+design+spec.+plan.	0.721	0.520	1.300	1.250

Table 144: previous phases vs next phase

Based on the results above, we investigate the same correlation with the Multilinear regression analysis. We want understand if it is possible increase the correlation and the estimation accuracy and identify which previous phase influence more the effort of the next one.

In Table 145 we show the results obtained. Starting from the design effort estimation the Multilinear regression doesn’t allow an improvement of the correlation and the accuracy. We find that the planning phase has more influence rather than the specification one and it explain the improvement obtain in RQ3 (see Table 144). In build effort estimation we obtain an improvement of the estimation accuracy even it is not acceptable. We find that more or less each previous phase influence in the same way the estimation (see Table 144). Contrarily in test effort estimation we discover a huge of drop in the accuracy rather than the results obtained in RQ3 and we identify the design phase the one more influencing while the build phase i8s le lessing one (see Table 144). As above also in deployment effort estimation we don't obtain positive accuracy improvement. Considering all previous phases, specification and test phases provide the major influence, instead of design phase which doesn't influence more (see Table 144).

	pearson	R^2	mmre	mdmre
design from spec +plan	0.221	0.648	0.633	0.498
	0.805			
build from design+spec.	0.434	0.325	1.480	0.750
	0.567			
build from design+spec.+plan.	0.647	0.618	0.610	0.380
	0.628			
	0.563			
test. from build+design	0.395	0.224	1.359	0.624
	0.376			
test. from build+design+spec.	0.390	0.325	1.42	0.795
	0.370			
	0.454			
test. from build+design +spec.+plan.	0.360	0.310	1.374	0.923
	0.400			
	0.540			
	0.270			
deploy. from test.+build	0.362	0.378	1.980	1.06
	0.614			
deploy. from test.+build+design	0.350	0.587	3.250	1.960
	0.760			
	0.230			
deploy. from test.+build+design +spec.	0.360	0.613	3.670	2.350
	0.780			
	0.230			
	0.390			
deploy. from test.+build +design+spec.+plan.	0.304	0.846	3.160	2.510
	0.830			
	0.200			
	0.376			
	0.725			

Table 145: previous phases vs next phase - Multilinear regression

RQ 3.1: Does considering one common caractheristics in addition to the effort for the phases, improve the effort spent up to a development phase to estimate its effort?

Here we want to use the effort of the previous phases for estimating the next one clustering the projects by each common caractheristics.

Grouped by one common characteristics

We investigated the correlation between the previous phases and the next one clustering the project by one common caractheristic.

Clustering by Domain

Here we report the results obtained for the second cluster where we select, as common characteristic, the domain: banking, communications, insurance, manufacturing and public administration.

Domain: Banking

The database do not contain enough projects in this cluster for estimating the remaing project from the planning, specification and build phases. In this cluster we don't obtain a good correlation and not an acceptable goodness of fit, as show in Table 146.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.093	0.336	1.567	1.489
build from design+spec.	0.728	0.530	1.023	1.145
build from design+spec.+plan.	0.748	0.560	0.989	0.848
test. from build+design	0.698	0.487	1.023	1.178
test. from build+design +spec.	0.556	0.309	1.356	1.478
test. from build +design+spec.+plan.	0.606	0.367	1.369	1.547
deploy. from test.	0.464	0.215	1.278	1.025
deploy. from test.+build	0.491	0.241	1.369	1.124
deploy. from test.+build+design	0.481	0.231	1.289	1.359
deploy. from test.+build+design +spec.	0.481	0.231	1.124	1.169
deploy. from test. +build+design+spec.+plan.	0.348	0.121	1.347	1.288

Table 146: previous phase vs next phase - Banking

Comparing this cluster with the one without cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144).

Clustering by Architecture

We report results for the first cluster selecting the architecture: Client server, Stand alone, Multi tier and Multi tier/Client server.

Architecture: Client server

The database do not contain enough projects in this cluster for estimating the remaing project from the planning, specification and build phases. In this cluster we don't obtain a good correlation and not an acceptable goodness of fit, as show in Table 147.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.801	0.642	0.987	0.875
build from design+spec.	0.588	0.450	0.969	0.845
build from design+spec.+plan.	0.597	0.357	1.245	1.199
test. from build+design	0,832	0.692	0.945	0.855
test. from build+design+ spec.	0.764	0.584	1.36	1.189
test. from build+design+spec.+plan.				
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 147: previous phase vs next phase - Client server

Comparing this cluster with the one without cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144).

Clustering by Development platform

Here we report the results obtained for the second cluster where we select, as common characteristic, the development platform: MR, MF, PC and Multi.

Development platform: MR

Since there are not enough data we can not analysis results for predicting the effort for design and deployment phases. In this cluster we don't obtain a high correlation in every cases and the accuracy is acceptable only for the build effort estimation, as show in Table 148.

	pearson	R^2	mmre	mdmre
design from spec +plan				
build from design+spec.	0.985	0.970	0.290	0.280
build from design+spec.+plan.	0.985	0.970	0.290	0.280
test. from build+design	0.775	0.600	0.580	0.670
test. from build+design+ spec.	0.781	0.610	0.610	0.700
test. from build+design+spec.+plan.	0.781	0.610	0.610	0.700
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 148: previous phase vs next phase - MR

Comparing the model obtained for all projects, we don't obtain a dramatic improvement of the correlation and of the accuracy especially for the build phase (see Table 144).

Development platform: MF

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a good correlation even if the accuracy is not acceptable, as show in Table 149.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.652	0.425	0.520	0.440
build from design+spec.	0.767	0.588	0.370	0.340
build from design+spec.+plan.	0.767	0.588	0.370	0.340
test. from build+design	0.621	0.386	0.350	0.320
test. from build+design+ spec.	0.565	0.319	0.550	0.610
test. from build+design+spec.+plan.	0.565	0.319	0.550	0.610
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 149: previous phase vs next phase – MF

Comparing the model obtained for all projects, we obtain an improvement of the accuracy especially for the build effort estimation (see Table 144).

Development platform: PC

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation even if the accuracy is not acceptable, as show in Table 150.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.839	0.704	0.600	0.470
build from design+spec.	0.682	0.466	0.550	0.400
build from design+spec.+plan.	0.682	0.466	0.550	0.400
test. from build+design	0.648	0.420	0.440	0.410
test. from build+design+ spec.	0.720	0.518	0.660	0.710
test. from build+design+spec.+plan.	0.720	0.518	0.660	0.710
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 150: previous phase vs next phase – PC

Comparing the model obtained for all projects, we obtain an improvement of the accuracy especially for the build effort estimation (see Table 144).

Development platform: Multi

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we don't obtain a high correlation especially for design and test phases, even if the accuracy is not acceptable, as show in Table 151.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.950	0.903	0.630	0.760
build from design+spec.	0.484	0.234	0.580	0.420
build from design+spec.+plan.	0.484	0.234	0.580	0.420
test. from build+design	0.851	0.725	0.430	0.380
test. from build+design+ spec.	0.901	0.811	0.910	0.950
test. from build+design+spec.+plan.	0.901	0.811	0.910	0.950
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 151: previous phase vs next phase – Multi

Comparing the model obtained for all projects whit this cluster, we obtain an improvement of the accuracy for build effort estimation (see Table 144).

Clustering by Programming language

Here we report the results obtained for the second cluster where we select, as common characteristic, the Programming language cluster, selecting : Java, COBOL, C++ and Visual basic.

Programming language: Java

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we do not obtain a good correlation even if the accuracy is not acceptable, as show in Table 152.

	pearson	R^2	mmre	mdmre
design from spec +plan	0,496	0,246	0,480	0,490
build from design+spec.	0,484	0,234	0,410	0,400
build from design+spec.+plan.	0,484	0,234	0,410	0,400
test. from build+design	0,461	0,213	0,430	0,410
test. from build+design+ spec.	0,421	0,177	0,620	0,680
test. from build+design+spec.+plan.	0,421	0,177	0,620	0,680
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 152: previous phase vs next phase - Java

Comparing the model obtained for all projects whit this cluster, we obtain an improvement of the estimation accuracy (see Table 144).

Programming language: COBOL

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation even if the accuracy is acceptable only for the build phase, as show in Table 153.

	pearson	R^2	mmre	mdmre
design from spec +plan				
build from design+spec.	0.811	0.658	0.330	0.290
build from design+spec.+plan.	0.811	0.658	0.330	0.290
test. from build+design	0.732	0.536	0.380	0.310
test. from build+design+ spec.	0.713	0.508	0.490	0.500
test. from build+design+spec.+plan.	0.713	0.508	0.490	0.500
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 153: previous phase vs next phase - Cobol

Comparing the model obtained for all projects whit this cluster, we obtain an improvement of the correlation and of the accuracy (see Table 144).

Programming language: C++

Since there are not enough data we can not analysis results for predicting the effort for design and deployment phases. In this cluster we obtain a good correlation especially for the test effort estimation based on build plus design phases, even if the accuracy is not acceptable as show in Table 154.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.642	0.412	0.770	0.460
build from design+spec.	0.642	0.412	0.770	0.460
build from design+spec.+plan.	0.966	0.933	0.420	0.380
test. from build+design	0.966	0.933	0.420	0.380
test. from build+design+ spec.	0.966	0.933	0.420	0.380
test. from build+design+spec.+plan.				
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 154: previous phase vs next phase - C++

Comparing the model obtained for all projects whit this cluster, we don't obtain an improvement of the estimation accuracy (see Table 144).

Clustering by Development process

Here we report the results obtained for the cluster where we select, as common characteristic, the Development process cluster, selecting : Waterfall and Data modelling. We obtain results only for Waterfall process.

Development process: Waterfall

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation even if the accuracy is acceptable only for design and test phase based on build plus design one, as show in Table 155.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.784	0.615	0.330	0.290
build from design+spec.	0.651	0.423	0.450	0.340
build from design+spec.+plan.	0.651	0.423	0.450	0.340
test. from build+design	0.939	0.876	0.290	0.240
test. from build+design+ spec.	0.920	0,846	0.940	0,940
test. from build+design+spec.+plan.	0.920	0.846	0.940	0.940
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 155: previous phase vs next phase - Waterfall

Comparing the model obtained for all projects whit this cluster, we don't obtain a dramatic improvement of the correlation and of the accuracy (see Table 144).

<u>Clustering by Functional measurement approach</u>

Here we report the results obtained for the second cluster where we select, as common characteristic, the Functional measurement approach cluster, selecting : IFPUG and COSMIC.

Functional measurement approach: IFPUG<1000

Since there are not enough data we can not analysis results for predicting the effort for design and deployment phases. In this cluster we obtain a high correlation for the test effort estimation, even if the accuracy is not acceptable, as show in Table 156.

	pearson	R^2	mmre	mdmre
design from spec				
design from spec +plan	0.489	0.240	0.490	0.280
build from design+spec.	0.457	0.209	0.690	0.530
build from design+spec.+plan.	0.457	0.209	0.690	0.530
test. from build+design	0.919	0.845	0.410	0.340
test. from build+design+ spec.	0.854	0.729	1.030	1.040
test. from build+design+spec.+plan.	0.854	0.729	1.030	1.040
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 156: previous phase vs next phase – IFPUG < 1000

Comparing the model obtained for all projects whit this cluster, we don't obtain an improvement of the accuracy for build phase and for test phase based on build and design one (see Table 144).

Functional measurement approach: COSMIC<1000

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a good correlation for build effort estimation even if the accuracy is not acceptable, as show in Table 157.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.580	0.336	0.960	0.930
build from design+spec.	0.741	0.549	0.340	0.310
build from design+spec.+plan.	0.741	0.549	0.340	0.310
test. from build+design	0.256	0.065	0.620	0.540
test. from build+design+ spec.	0.241	0.058	0.500	0.520
test. from build+design+spec.+plan.	0.241	0.058	0.500	0.520
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 157: previous phase vs next phase – COSMIC < 1000

Comparing the model obtained for all projects whit this cluster, we don't obtain an improvement of the estimation accuracy (see Table 144).

Clustering by two common characteristics

Based on the results obtained grouped by one common characteristics we refined the analysis clustering the projects by two common characteristics.

Clustering by Architecture and Development platform

Here we combined the Architecture and the Development process for each project as shown in Figure 19. Following we reported the results.

Architecture and Development platform: Client server and PC

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a good correlation except for the design effort estimation, even if the accuracy is not acceptable, as show in Table 158.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.394	0.155	0.430	0.460
build from design+spec.	0.731	0.534	0.390	0.410
build from design+spec.+plan.	0.731	0.534	0.390	0.410
test. from build+design	0.791	0.625	0.480	0.450
test. from build+design+ spec.	0.788	0.620	0.720	0.740
test. from build+design+spec.+plan.	0.788	0.620	0.720	0.740
deploy. from test.				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 158: previous phase vs next phase – Client server and PC

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144 and Table 147).

Architecture and Development platform: Client server and Multi

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation even if the accuracy is acceptable only for the test effort estimation, as show in Table 159.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.953	0.908	1.980	1.380
build from design+spec.	0.834	0.696	0.960	0.820
build from design+spec.+plan.	0.834	0.696	0.960	0.820
test. from build+design	0.953	0.908	0.360	0.240
test. from build+design+ spec.	0.953	0.908	0.360	0.240
test. from build+design+spec.+plan.	0.953	0.908	0.360	0.240
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 159: previous phase vs next phase – Client server and Multi

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we obtain a dramatic improvement of the correlation in every case and of the accuracy only for the test effort estimation (see Table 144 and Table 147

Clustering by Architecture and Development process

Here we combined the Architecture and the Development process for each project as shown in Figure 19. Following we reported the results.

Architecture and Development process: Client server and Waterfall

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we do not obtain a good correlation even if the accuracy is acceptable only for the design effort estimation, as show in Table 160.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.784	0.615	0.330	0.290
build from design+spec.	0.851	0.724	0.520	0.610
build from design+spec.+plan.	0.851	0.724	0.520	0.610
test. from build+design	0.634	0.401	0.400	0.280
test. from build+design+ spec.	0.635	0.403	0.780	0.840
test. from build+design+spec.+plan.	0.635	0.403	0.780	0.840
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 160: previous phase vs next phase - Client server and Waterfall

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144 and Table 147).

Clustering by Architecture and Programming language

We combined the Architecture and the Programming language for each project as shown in Figure 20. Following we reported the results.

Architecture and Programming language: Client server and Java

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a good correlation for the build effort estimation even if the accuracy is acceptable only for the build phase, as show in Table 161.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.556	0.309	0.550	0.580
build from design+spec.	0.792	0.626	0.300	0.310
build from design+spec.+plan.	0.792	0.626	0.300	0.310
test. from build+design	0.444	0.197	0.670	0.590
test. from build+design+ spec.	0.557	0.314	0.640	0.700
test. from build+design+spec.+plan.	0.557	0.314	0.640	0.700
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 161: previous phase vs next phase – Client server and Java

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144 and Table 147).

Clustering by Development platform and Development process

After we combined the Development platform and Development process for each project as shown in Figure 22. Following we reported the results.

Development platform and Development process: PC and Waterfall

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a good correlation even if the accuracy is acceptable only for the design phase, as show in Table 162.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.788	0.620	0.250	0.230
build from design+spec.	0.851	0.724	0.520	0.610
build from design+spec.+plan.	0.851	0.724	0.520	0.610
test. from build+design	0.545	0.298	0.340	0.290
test. from build+design+ spec.	0.635	0.403	0.780	0.840
test. from build+design+spec.+plan.	0.635	0.403	0.780	0.840
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 162: previous phase vs next phase – PC and Waterfall

Comparing the model obtained for all projects with the one obtained clustered by Client server and this cluster, we don't obtain an improvement of the correlation and of the accuracy for the design effort estimation (see **Table 144** and **Table 150**).

Clustering by Development platform and Programming language

After we combined the Development platform and Programming language for each project as shown in Figure 22. Following we reported the results.

Development platform and Programming language: MF and COBOL

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we do not obtain a good correlation and not an acceptable goodness of fit, as show in

	pearson	R^2	mmre	mdmre
build from design+spec.	0.643	0.492	0.350	0.440
build from design+spec.+plan.	0.643	0.492	0.350	0.440
test. from build+design	0.301	0.259	0.570	0.630
test. from build+design+ spec.	0.301	0.259	0.570	0.630
test. from build+design+spec.+plan.	0.301	0.259	0.570	0.630
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 163: previous phase vs next phase – MF and COBOL

Comparing the model obtained for all projects with the one obtained clustered by MF and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144).

Development platform and Programming language: PC and Java

Since there are not enough data we cannot analysis results for predicting the effort for deployment phase. In this cluster we don't obtain a good correlation except for the design effort estimation, even if the accuracy is not acceptable, as show in Table 164.

	pearson	R^2	mmre	mdmre
design from spec +plan	0.261	0.068	0.410	0.300
build from design+spec.	0.581	0.338	0.400	0.330
build from design+spec.+plan.	0.581	0.338	0.400	0.330
test. from build+design	0.581	0.338	0.400	0.330
test. from build+design+ spec.	0.581	0.338	0.400	0.330
test. from build+design+spec.+plan.	0.581	0.338	0.400	0.330
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 164: previous phase vs next phase – PC and Java

Comparing the model obtained for all projects with the one obtained clustered by PC and this cluster, we don't obtain an improvement of the correlation and of the accuracy for the design and build effort estimation (see Table 144 and Table 150).

<u>Clustering by Development platform and Functional measurement approach</u>

After we combined the Development platform and the Functional measurement approach for each project as shown in Figure 23. Following we reported the results.

Development platform and Functional measurement approach: MF and COSMIC<1000

Since there are not enough data we can not analysis results for predicting the effort for deployment and design phases. In this cluster we obtain a good correlation only for the build phase with an acceptable goodness of fit, as show in Table 165.

	pearson	**R^2**	**mmre**	**mdmre**
design from spec +plan.				
build from design+spec.	0.723	0.522	0.250	0.240
build from design+spec.+plan.	0.723	0.522	0.250	0.240
test. from build+design	0.230	0.053	0.390	0.330
test. from build+design+ spec.	0.230	0.053	0.390	0.330
test. from build+design+spec.+plan.	0.230	0.053	0.390	0.330
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 165: previous phase vs next phase – MF and COSMIC<1000

Comparing the model obtained for all projects with the one obtained clustered by MF and this cluster, we don't obtain an improvement of the accuracy when we estimate the build effort, and we have a huge of drop for the correlation in test phase (see Table 144 and Table 149).

Development platform and Functional measurement approach: PC and COSMIC<1000

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation even if the goodness of fit is not acceptable, as show in Table 166.

	pearson	R^2	mmre	mdmre
design from spec +plan				
build from design+spec.	0.933	0.871	0.410	0.420
build from design+spec.+plan.	0.933	0.871	0.410	0.420
test. from build+design	0.933	0.871	0.410	0.420
test. from build+design+ spec.	0.933	0.871	0.410	0.420
test. from build+design+spec.+plan.	0.933	0.871	0.410	0.420
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 166: previous phase vs next phase – PC and COSMIC<1000

Comparing the model obtained for all projects with the one obtained clustered by PC and this cluster, we don't obtain an improvement of the correlation but not for the accuracy (see Table 144 and Table 150).

Clustering by Programming language and Functional measurement approach

After we combined the Programming language and the Functional measurement approach for each project as shown in Figure 26. Following we reported the results.

Programming language and Functional measurement approach: Java and IFPUG<1000

Since there are not enough data we can not analysis results for predicting the effort for design and deployment phases. In this cluster we do not obtain a high correlation with an acceptable goodness of fit, as show in Table 167.

	pearson	R^2	mmre	mdmre
design from spec +plan				
build from design+spec.				
build from design+spec.+plan.				
test. from build+design	0.927	0.859	0.320	0.240
test. from build+design+ spec.	0.927	0.859	0.320	0.240
test. from build+design+spec.+plan.	0.927	0.859	0.320	0.240
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 167: previous phase vs next phase – Java and IFPUG<1000

Comparing the model obtained for all projects with the one obtained clustered by Java and this cluster, we don't obtain an improvement of the correlation and of the accuracy (see Table 144 and Table 152).

Programming language and Functional measurement approach: Java and COSMIC<1000

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we do not obtain a high correlation with an acceptable goodness of fit only for the build phase, as show in Table 168.

	pearson	R^2	mmre	mdmre
design from spec				
design from spec +plan				
build from design+spec.	0.817	0.668	0.340	0.320
build from design+spec.+plan.	0.817	0.668	0.340	0.320
test. from build+design	0.206	0.042	0.850	0.640
test. from build+design+ spec.	0.206	0.042	0.850	0.640
test. from build+design+spec.+plan.	0.206	0.042	0.850	0.640
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 168: previous phase vs next phase – Java and COSMIC<1000

Comparing the model obtained for all projects with the one obtained clustered by Java and this cluster, we don't obtain an improvement of the correlation and of the accuracy for the build phase (see Table 144 and Table 152).

Programming language and Functional measurement approach: COBOL and COSMIC<1000

Since there are not enough data we can not analysis results for predicting the effort for design and deployment phases. In this cluster we obtain a good correlation and not an acceptable goodness of fit only for build effort estimation, as show in Table 169.

	pearson	R^2	mmre	mdmre
design from spec				
design from spec +plan				
build from design+spec.	0.811	0.658	0.330	0.290
build from design+spec.+plan.	0.811	0.658	0.330	0.290
test. from build+design	0.345	0.119	0.470	0.410
test. from build+design+ spec.	0.345	0.119	0.470	0.410
test. from build+design+spec.+plan.	0.345	0.119	0.470	0.410
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 169: previous phase vs next phase – COBOL and COSMIC<1000

Comparing the model obtained for all projects with the one obtained clustered by COBOL and this cluster, we don't obtain an improvement of the correlation and of the accuracy for build phase (see Table 144 and Table 153).

Clustering by three common characteristics

Based on the results obtained grouped by two common characteristics we refined the analysis clustering the project by three common characteristics.

Clustering by Architecture and Development platform and Development process

Here we select Domain and Architecture (see Figure 14). Following the results.

Architecture and Development platform and Development process: Client server and PC and Waterfall

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation, even if the goodness of fit is acceptable only for the design phase, as show in Table 170.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.784	0.615	0.330	0.290
build from design+spec.	0.851	0.724	0.520	0.610
build from design+spec.+plan.	0.851	0.724	0.520	0.610
test. from build+design	0.634	0.401	0.400	0.280
test. from build+design+ spec.	0.635	0.403	0.780	0.840
test. from build+design+spec.+plan.	0.635	0.403	0.780	0.840
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 170: previous phases vs next phase – Client server and PC and Waterfall

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) the one obtained clustered by Client server and PC and (3) this cluster, we don't obtain an improvement of the correlation and of the accuracy for design phase (see Table 144 and Table 147 and Table 150).

<u>Clustering by Architecture and Development platform and Programming language</u>

Here we select Domain and Architecture (see Figure 14). Following the results.

Architecture and Development platform and Programming language: Client server and PC and Java

Since there are not enough data we can not analysis results for predicting the effort for deployment phase. In this cluster we obtain a high correlation, even if the goodness of fit is acceptable only for the build phase, as show in Table 171.

	pearson	R^2	mmre	mdmre
design from spec +plan.	0.407	0.165	0.400	0.360
build from design+spec.	0.792	0.627	0.310	0.310
build from design+spec.+plan.	0.792	0.627	0.310	0.310
test. from build+design	0.537	0.288	0.730	0.780
test. from build+design+ spec.	0.557	0.311	0.640	0.700
test. from build+design+spec.+plan.	0.557	0.311	0.640	0.700
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 171: previous phases vs next phase – Client server and PC and Java

Comparing the model obtained for all projects with (1) the one obtained clustered by Client server, (2) the one obtained clustered by Client server and PC and (3) this cluster, we obtain an improvement of the correlation and of the accuracy in build case. (see Table 144 and Table 147 and Table 150).

<u>Clustering by Development platform and Programming language and Functional measurement approach</u>

Here we select Domain and Architecture (see). Following the results.

Development platform and Programming language and Functional measurement approach: MF and Java and COSMIC <1000

Since there are not enough data we can analysis results only for predicting the effort for build phase. In our result we obtain a high correlation and an acceptable goodness of fit , as show in Table 172.

	pearson	R^2	mmre	mdmre
design from spec +plan.				
build from design+spec.	0.963	0.928	0.240	0.230
build from design+spec.+plan.	0.963	0.928	0.240	0.230
test. from build+design				
test. from build+design+ spec.				
test. from build+design+spec.+plan.				
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 172: previous phases vs next phase – MF and Java and COSMIC<1000

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) the one obtained clustered by MF and PC and (3) this cluster, we obtain an improvement of the correlation and of the accuracy for build phase (see Table 144 and Table 149 and Table 152).

Development platform and Programming language and Functional measurement approach: MF and COBOL and COSMIC<1000

Since there are not enough data we can not analysis results for predicting the effort for design and deployment phases. In our result we obtain a high correlation except for the build phase with an acceptable accuracy, as show in Table 173.

	pearson	R^2	mmre	mdmre
design from spec +plan.				
build from design+spec.	0.723	0.522	0.250	0.240
build from design+spec.+plan.	0.723	0.522	0.250	0.240
test. from build+design	0.345	0.119	0.470	0.430
test. from build+design+ spec.	0.353	0.125	0.370	0.350
test. from build+design+spec.+plan.	0.353	0.125	0.370	0.350
deploy. from test.+build				
deploy. from test.+build+design				
deploy. from test.+build+design+spec.				
deploy. from test.+build+design+spec.+plan.				

Table 173**: previous phases vs next phase – MF and COBOL and COSMIC<1000**

Comparing the model obtained for all projects with (1) the one obtained clustered by MF, (2) the one obtained clustered by MF and COBOL and (3) this cluster, we obtain an improvement of the correlation and of the accuracy for build phase (see Table 144 and Table 149 and Table 153).

Clustering by four common caractheristics

Clustering the project by four common characteristics doesn’t allow to obtain any projects for the effort estimation.

Clustering by five common caractheristics

Clustering the project by five common characteristics doesn’t allow to obtain any projects for the effort estimation.

Clustering by six common caractheristics

Clustering the project by six common characteristics doesn’t allow to obtain any projects for the effort estimation.

RQ4: Is it possible to use the effort spent up to a development phase to estimate the remaining project effort?

In RQ4 we investigate the correlation without clustering projects in order to understand if the effort spent up a phase has a higher prediction power to estimate the remaining project effort, compared to the effort of the previous phase, as in RQ2. In Table 183 we show the results for this research questions.

In #1 we estimated the effort for the remaing project based on the sum of planning and specification phases. Comparing this result with the one obtained in RQ2, we obtain an improvement of the correlation and the estimation accuracy.

Then where we consider as previous phases in #2 planning plus specification plus design and in #5 only specification plus design, taking into account more than one previous phase we obtain an improvement of the correlation and of the accuracy. The best results is considering as previous phases: planning plus specification plus design.

In the #3, #6 and #8 analysis we estimate the same part of the project as in RQ3 when we take in to account only the build phase. Also in this case considering more than one previous phase allow to improve the correlation and the estimation accuracy.

Finally we have four different analysis (#4, #7, #9 and #10) for estimating the deployment effort to compare with RQ2. In term of correlation we have a dramatic improvement with more previous phases considered, but the estimation accuracy is better taking in to account as previous phases all except the planning one.

#	phase	pearson	R^2	mmre	mdmre
1	plan.+spec. _ remaining	0.828	0.683	0.460	0.710
2	plan.+spec.+design _ remaining	0.774	0.594	0.500	0.650
3	plan.+spec.+design+build _ remaining	0.731	0.529	0.550	0.980
4	plan.+spec.+design+build+test _deployment	0.721	0.515	2.990	4.960
5	spec.+design _ remaining	0.503	0.250	0.670	1.440
6	spec.+design+build _ remaining	0.712	0.505	0.670	1.440
7	spec.+design+build +test_deployment	0.663	0.437	0.789	0.934
8	design+build_ remaining	0.694	0.480	0.520	1.100
9	design+build+test_deployment	0.664	0.439	1.790	3.410
10	build+test_deployment	0.594	0.351	1.230	1.456

Table 174: previous phases vs remaining projects

Here we replace the analysis with the Multilinear regression, as shown in Table 184. In general we don't obtain an improvement of the correlations and of the goodness of fit. The accuracy improves marginally basing on all previous phases and when we consider only the design plus build plus test for estimating the deployment effort. Moreover in the deployment effort estimation the accuracy decreases such as in if we consider all the previous phases.

phase	pearson	R^2	mmre	mdmre
plan&spec_sum remaining	0.700	0.693	0.460	0.680
	0.643			
plan&spec&design_sum remaining	0.723	0.699	0.460	0.520
	0.601			
	0.512			
plan&spec&design&build_sum remaining	0.625	0.622	0.790	1.030
	0.565			
	0.356			
	0.722			
plan&spec&design&build&test_deployment	0.725	0.847	3.270	4.990
	0.376			
	0.200			
	0.830			
	0.304			
spec&design_sum remaining	0.582	0.314	0.700	1.540
	0.386			
spec&design&build_sum remaining	0.571	0.556	0.570	1.210
	0.363			
	0.700			
spec&design&build&test_deployment	0.390	0.621	0.689	0.896
	0.229			
	0.781			
	0.355			
design&build_sum remaining	0.366	0.495	0.540	1.350
	0.691			
design&build&test_deployment	0.232	0.587	1.790	2.920
	0.760			
	0.354			
build&test_deployment	0.614	0.378	1.890	1.698
	0.362			
	0.362			

Table 175: previous phases vs remaining projects – multivariate regression

4.3 Results discussion

To answer our research questions, we applied the estimation approach that we defined in Chapter 3 to the ISBSG dataset. We investigated the correlation among the efforts related to different development phases, considering first the development projects without clustering by common caractheristics and then clustering for one or more common characteristics, as shown in Figure 13.

As for RQ1, which is about estimating the effort for one phase based on the previous one, we obtain a good correlation for the effort estimation of the build phase based on the design phase and of the design phase based on the specification one (see in Figure 33). Moreover mmre and mdmre show an acceptable goodness of fit as shown in Table 176.

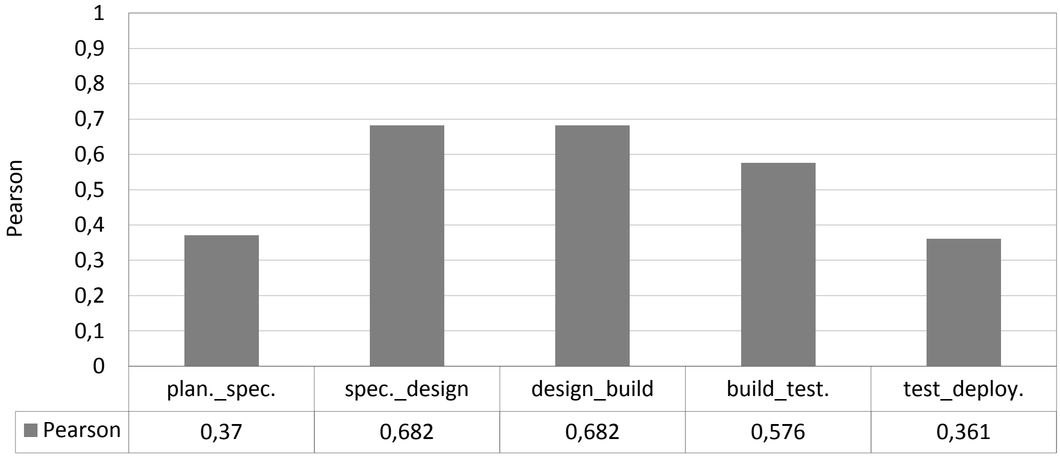


Figure 33: previous phase vs next phase (RQ1) – results

phase	mmre	mdmre
plan_spec.	1.700	0.840
spec._design	0.970	0.390
design_build	0.510	0.460
build_test	1.220	0.610
test_deploy	1.980	0.950

Table 176: previous phase vs next phase (RQ1) - goodness of fit

When analyzing the effort estimation of the remaining project phases based on the previous phase (RQ2), we obtain a good correlation for all the analyses except for the effort estimation of the remaining project phases based on the design phase, as shown in Figure 34. As for RQ1, However mmre and mdmre show an acceptable goodness of fit as shown in Table 177.

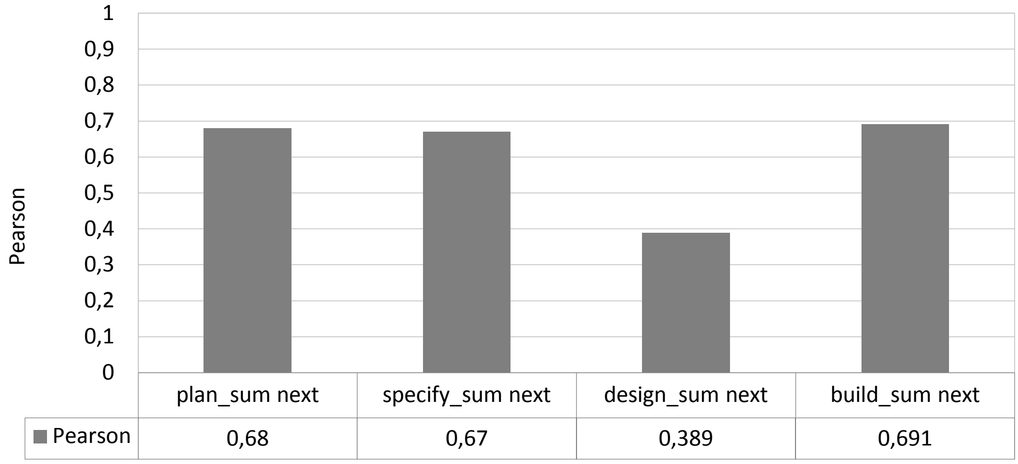


Figure 34: previous phase vs sum next phases (RQ2) – results

phase	mmre	mdmre
plan_sumnext	0.690	0.520
spec._sumnext	1.350	0.810
design_sumnext	1.340	0.690
build_sumnext	0.840	0.500

Table 177: previous phase vs sum next phases (RQ2) - goodness of fit

In order to improve the correlation between one phase and the next one, we try to answer to the RQ3, where we take into account deployment effort estimation based on all previous phases.

Unlike with RQ1 and RQ3, we obtain a dramatic improvement of the correlations for all the analysis, as shown in Figure 35. However, we also have an acceptable goodness of fit as shown in Table 177.

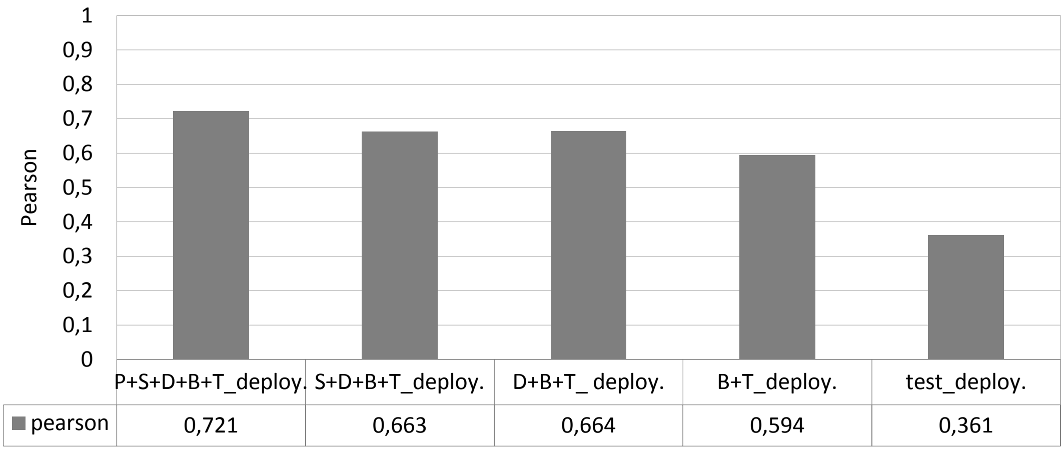


Figure 35: previous phases vs next phase (RQ3) – results

phase	mmre	mdmre
P+S+D+B+T_dep.	0.690	0.520
S+D+B+T_dep.	0.950	0.930
D+B+T_dep.	0.900	0.960
B+T_dep.	0.860	0.920
test_dep.	1.980	0.950

Table 178: previous phases vs next phase (RQ3) - goodness of fit

Here we want to assess whether we can improve the estimation of the remaining project effort compared to the one obtained in RQ2.

As for RQ3, a similar trend can be seen in the results of RQ4. Taking into account more than one of the previous phases effort we obtain a dramatic improvement of the correlation for all the analysis as shown in Figure 36 and of the goodness of fit as shown in Table 179.

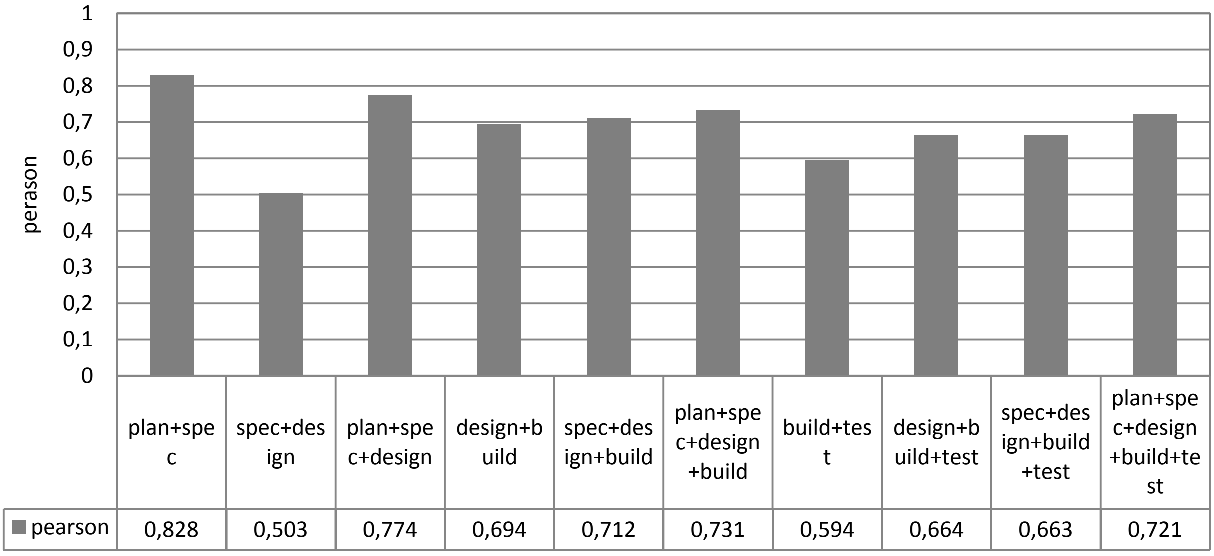


Figure 36: previous phases vs sum next phases (RQ4) – result

Effort estimation of the remaining project	mmre	mdmre
plan. + spec.	0.460	0.710
spec. + design	0.670	1.440
plan. + spec. + design	0.500	0.650
design + build	0.520	1.100
spec. + design + build	0.670	1.440
plan. + spec. + design + build	0.560	1.230
build + test	1.900	3.560
design + build + test	1.790	3.410
spec. + design + build + test	1.080	1.340
plan. + spec. + design + build + test	0.800	0.900

Table 179: previous phases vs sum next phases (RQ4) - goodness of fit

Finally we summarize the results for RQ1, RQ2, RQ3 and RQ4 obtained without clustering projects by common caractheristics. Considering more than one previous phase allows to obtain an improvement of the correlation and of the goodness of fit either in the next development phase effort and in the remaining project effort estimation.

In Figure 37 and in Figure 38 we show the correlation improvement and of the goodness of fit (see Table 180 and Table 181) taking into account all previous phases.

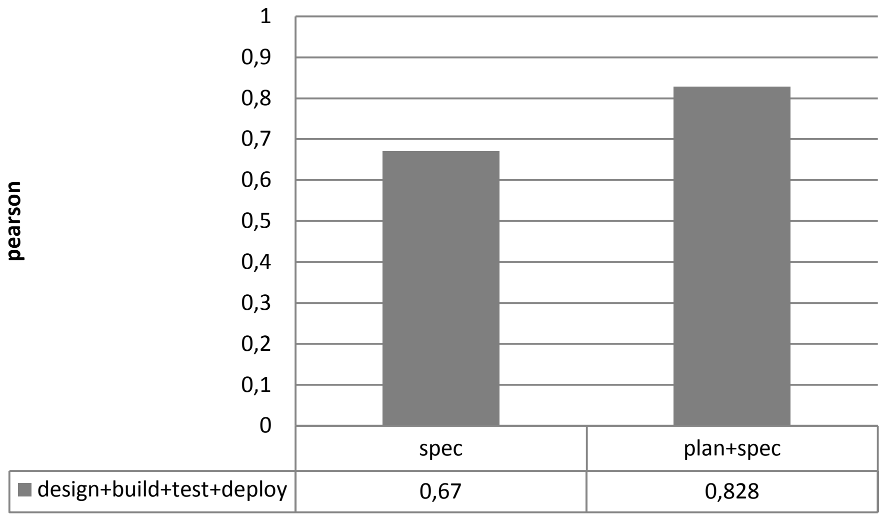


Figure 37: effort estimation of remaining project based on the previous phases (example 1) - results

design + build + test + deploy effort estimation	mmre	mdmre
plan. + spec.	0.460	0.710
spec.	1.350	0.810

Table 180: effort estimation of remaining project based on the previous phases (example 1) - goodness of fit

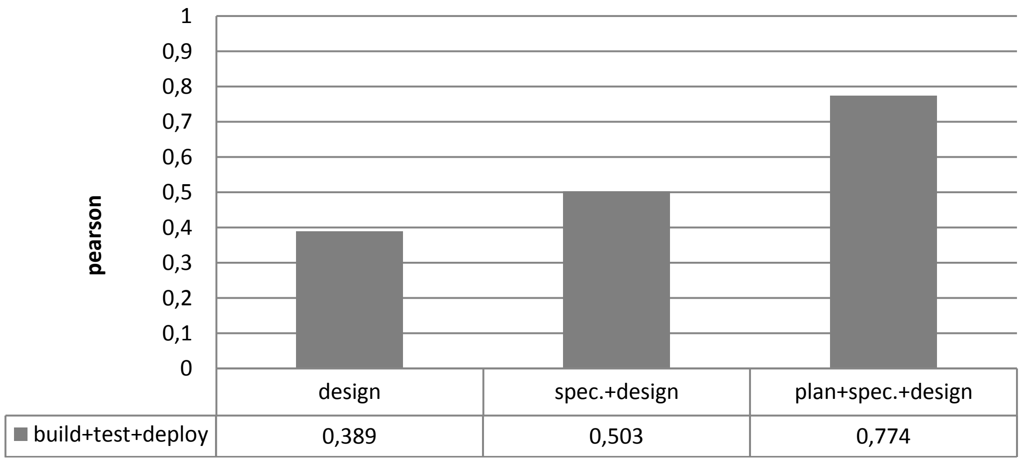


Figure 38: effort estimation of remaining project based on the previous phases (example 2) - results

build + test + deploy effort estimation	mmre	mdmre
plan. + spec. + design	0.500	0.650
spec. + design	0.670	1.440
design	1.340	0.690

Table 181: effort estimation of remaining project based on the previous phases (example 2) - goodness of fit

Since we want to investigate if we can improve the estimation accuracy obtained in RQ1, RQ2, RQ3 and RQ4, we cluster all new development projects considering 6 different attributes (domain, architecture, development platform, development process, programming language and functional measurement approach) with 21 attributes (see Figure 13), analyzing a total of 1280 data sets.

Clustering by common caractheristics allows, in several cases, an improvement of the correlation (from 0.6 to 0.9) and of the goodness of fit (mmre goes from 1.2 to 0.2).

Taking into account the clusters built by one common caractheristic (RQ3.1) we obtained the best results when we answer to RQ3.1 whit good correlations and an acceptable goodness of fit (from 0.2 to 0.8).

The best common caractheristics identified for effort estimation purposes are: client server architecture, PC and MF development platform, Java and COBOL development language, and COSMIC<1000 functional measurement approach.

For illustration purposes, we here report one of the best results, on the estimation of the build phase effort based on the effort for the previous two phases (design and specification). We compare the results obtained with clustering by client-server architecture, MF development platform and COSMIC<1000 functional measurement approach as shown in Figure 39. As we can see in Table 182 we have a significant improvement of the goodness of fit compared with the results obtained without cluster.

Figure 39: previous phases vs next phase clustering by one common caractheristic (RQ3.1) - example results

	mmre	mdmre
no cluster		
client server	0.390	0.350
MF	0.370	0.340
COSMIC<1000	0.260	0.200

Table 182: previous phases vs next phase clustering by one common caractheristic (RQ3.1) - goodness of fit

Clustering by two common caractheristics we obtained the best results when we answer to RQ3.1 with a good correlation in the most of the analysis and an acceptable goodness of fit (from 0.2 to 0.8).

The best pairs of caractheristics identified are client server architecture and PC development platform, client server architecture and java programming language, client server architecture and COSMIC<1000 functional measurement approach, PC development platform and waterfall development process. Also we identify two MF development platform combination with COBOL and java programming language and finally COSMIC<1000 functional measurement approach combined with COBOL or java programming language.

We take into account the same combination used above as one of the best results, we compare the results obtained with the client server architecture and PC development platform, MF development platform and COBOL programming language and COSMIC<1000 functional measurement approach with java programming language as shown in Figure 40. In Table 183 we have a significant improvement of the goodness of fit compared with the results obtained without cluster.

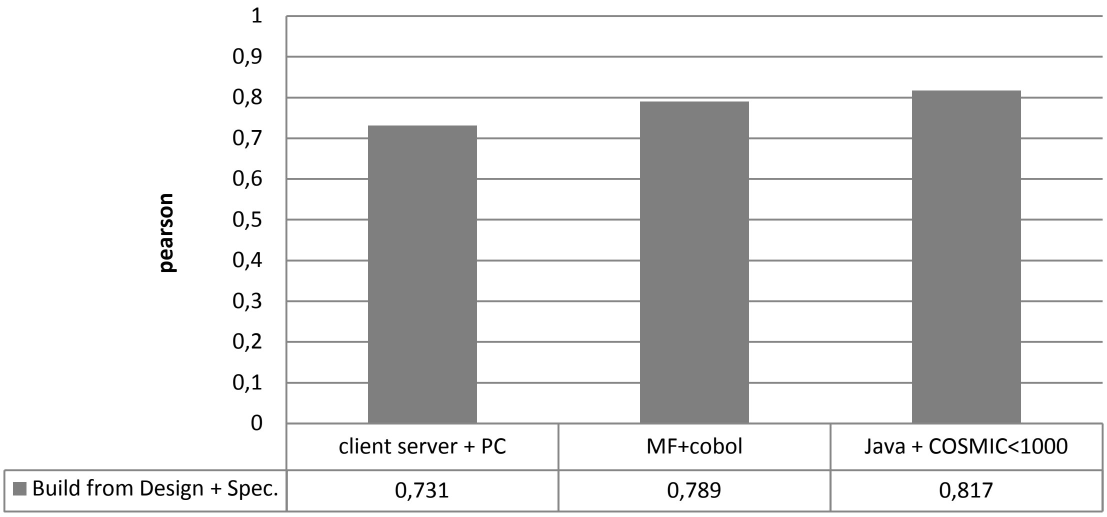


Figure 40: previous phases vs next phase clustering by two common caractheristics (RQ3.2) – results

	mmre	mdmre
no cluster		
client server + PC	0.390	0.410
MF + COBOL	0.340	0.290
Java + COSMIC<1000	0.250	0.240

Table 183: previous phases vs next phase clustering by two common caractheristics (RQ3.2) - goodness of fit

Clustering by three common caractheristics we obtained the best results when we answer to RQ3.1 with a good correlation in the most analysis, an acceptable goodness of fit (0.2 ÷ 0.8).

The best common caractheristics identified are client server architecture and PC development platform and waterfall development process, client server architecture and PC development platform and java programming language, MF development platform and java programming language and COSMIC<1000 functional measurement approach, MF development platform and COBOL programming language and COSMIC<1000 functional measurement approach.

We take into account the same example used above as one of the best results, we compare the results obtained with the client server architecture and PC development platform and COBOL programming language, and MF development platform and java programming language and COSMIC<1000 functional measurement approach as shown in Figure 41. In Table 184 we have a significant improvement of the goodness of fit compared with the results obtained without cluster.

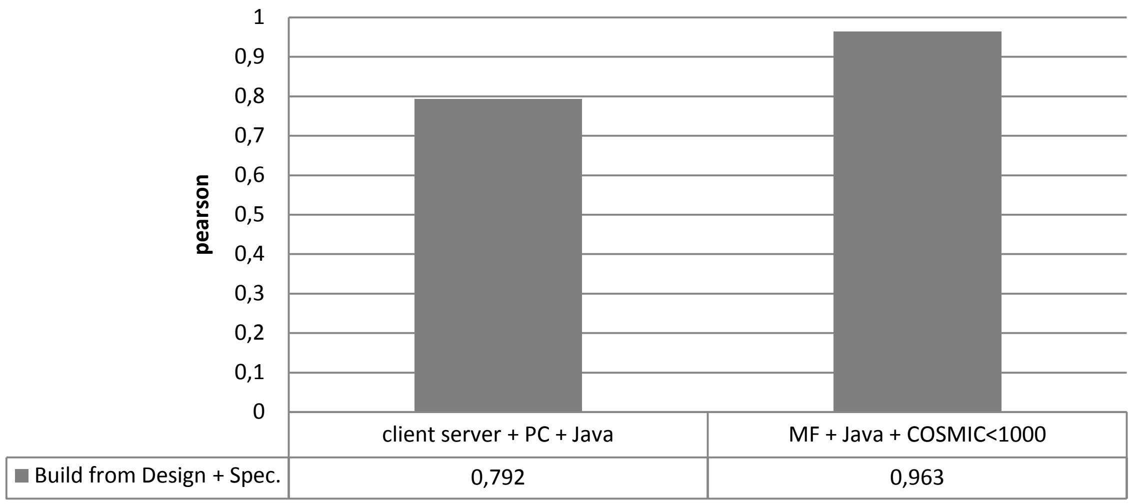


Figure 41: previous phases vs next phase clustering by two common caractheristics (RQ3.3) – results

	mmre	mdmre
no cluster		
client server + PC + java	0.390	0.350
MF + Java + COSMIC<1000	0.250	0.240

Table 184: previous phases vs next phase clustering by two common caractheristics (RQ3.3) - goodness of fit

As an example, here we report the comparison of one of the most significant results, where we compare the build phase based on the design and specification phases without clustering and clustering by one, two and three common caractheristics. We report an example where we compare the results obtained with the client server architecture and PC development platform and COBOL programming language, and MF development platform and java programming language and COSMIC<1000 functional measurement approach as shown in Figure 41. In Table 184 we report the dramatic improvement of the goodness of fit with one or more cluster attributes.

As expected, the more are the attributes considered, the higher is the accuracy of the estimation.

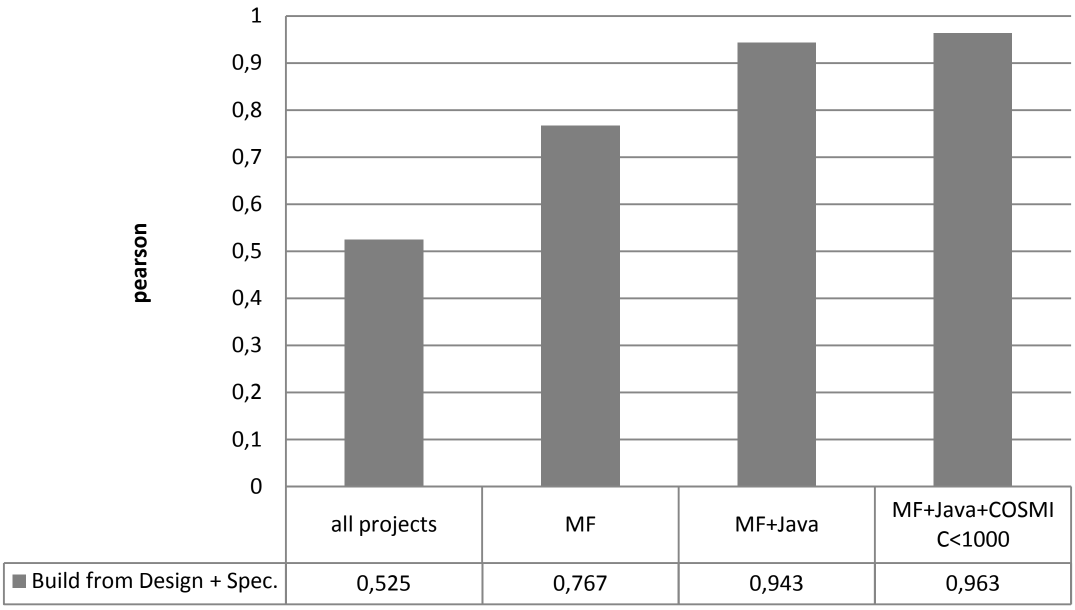


Figure 42: comparison between all project and clustering by common caractheristics - results

	mmre	mdmre
All projects	2.930	2.090
MF	0.370	0.340
MF + java	0.340	0.290
MF + java + COSMIC<1000	0.250	0.240

Table 185: comparison between all projects and clustering by common caractheristics - goodness of fit

We also want to underline that in RQ3.1. taking into account more than one previous phases not improve the accuracy and the correlation as in RQ3.

CHAPTER 5 – APPLYING THE APPROACH TO AGILE PROCESSES

Since the approach performed positively with the waterfall process, we wanted to understand whether the approach can be applied in Agile processes [45].

Here we want to understand if OLS techniques, can help to estimate the effort of user stories, based on the effort spent on similar previous stories.

For this reason, we conducted an empirical study on a SCRUM project developed with Moonlighting SCRUM [43], a version of SCRUM adapted for part-time developers working in non-overlapping hours.

In the following sections, we describe the empirical study carried out. First in the section 5.1 we introduce the context, describing the development process and the application developed, and then, in the section 5.2, we introduce the study design. Finally, we present and discuss the results.

5.1 Context

5.1.1 The development process

Agile software development is a category of development methods aimed to quickly and easily react to requirements changes.

The vast majority of Scrum's practices are not new to SE. Scrum was developed at Easel Corporation in 1993 [46], basically with the same idea behind Barry Boehm's Spiral Model [47].

Scrum speeds up the requirements adaptability of the spiral model with some agile practices from Extreme Programming [48], such as pair programming and daily meetings.

Scrum is a lightweight, iterative, and incremental development model based on three principles:

- Transparency: Any significant aspects of the process must be visible to those responsible for the outcome.
- Inspection: Artifacts must be frequently inspected by skilled persons.
- Adaptation: The process must allow its adjustment in case of negative inspection results.

Moreover, Scrum prescribes formal practices for inspection and adaptation (Figure 43):

- Sprint Planning Meeting: takes place at the beginning of each sprint. The product owner discusses with the developers the stories on which to focus during the next sprint.
- Daily Scrum: daily meeting where each member answers three questions:
    - What did I do yesterday?
    - What will I do today?
    - What prevents me from performing my work as efficiently as possible?
- Sprint Review: runs at the end of each sprint to show the work done to the product owner.
- Sprint Retrospective: runs after the sprint review. Teams discuss what went well, what did not, and what improvements could be made in the next sprint.

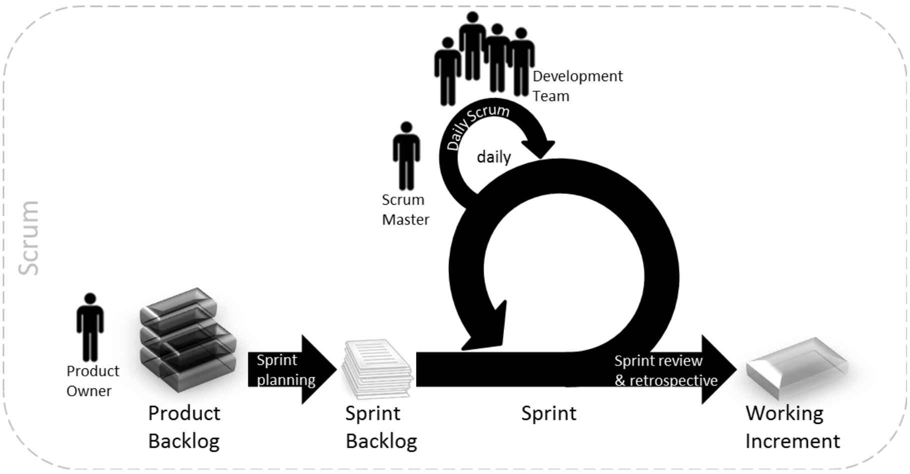


Figure 43: the SCRUM process

Moonlighting Scrum is a Scrum extension that helps developers structure the development process with the goal of releasing the best product possible with the available resources.

Just like Scrum, Moonlighting Scrum requires sprint planning meetings, sprint reviews, and retrospectives. During the meetings, the whole team and the product owner must meet in person or via video conference.

In Scrum, sprints last from two to three weeks, whereas in Moonlighting Scrum they last from three to four weeks. Because of the physical distribution and the non-overlapping time for the developers, pair programming cannot be applied and the daily meetings prescribed by Scrum cannot be attended in person. As a consequence, inspection are the responsibility of the Scrum master, who is in charge of checking the entire quality and help the developers preserve a minimum amount of code quality. Moonlighting Scrum is thought to deliver the highest quality possible with limited resources available.

Therefore, morning meetings are replaced with an online forum by creating a thread for every six working hours where each developer writes his/her comments by replying to three questions:

- What have you completed, with respect to the backlog, since the last daily meeting?
- What specific tasks, with respect to the backlog, do you plan to accomplish until the next report?
- What obstacles got in the way of completing this work?

The Scrum Master also has to take care of communication efficiency by adapting the online reporting interval, and is in charge of increasing or decreasing the reporting time based on the team's efficiency.

For this reason, the team members must also answer two further questions in their online report:

- When did you work (start-end)?
- How long did you work on writing this report?

The developers are working for at most ten hours per week and must work for a minimum of two continuous hours. Consequently, the time needed to write the report at the beginning and at the end of their work might take up an important percentage of their working time.

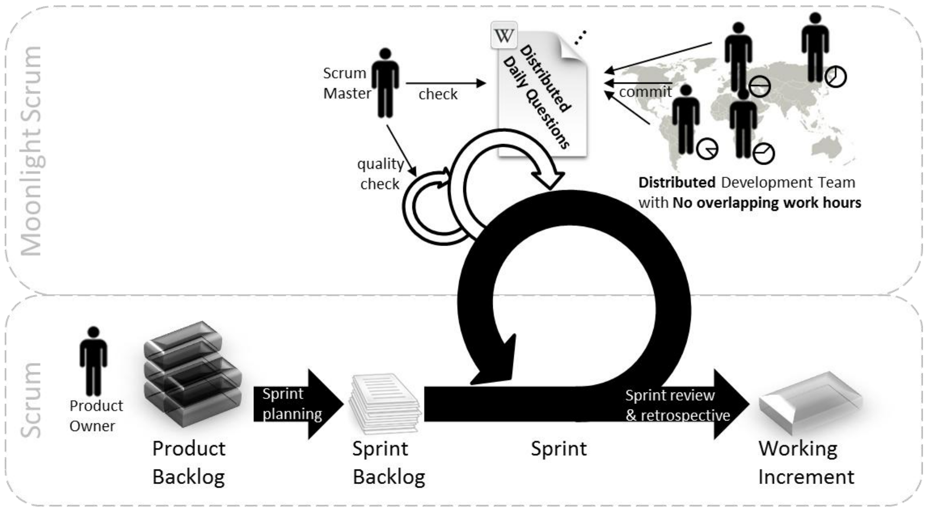


Figure 44: the Moonlight SCRUM process

Moonlighting Scrum is applicable to a wide range of projects, from university and research-based projects to open source projects. In general, the process requires around 15% more of effort for communication than Scrum [43] but allows the development code in a controlled and structured way.

5.1.2 The application developed

We analyzed the development of Process Configuration Framework (PCF), an online tool to classify software technologies and identify tool chains in specific domains [44]. PCF is a relatively small application, composed of 12,500 effective lines of code, calculated without considering comment lines, empty lines, and lines containing only brackets. The development started in February 2013, based on an existing prototype, and the first version of the tool was released at the end of May 2013. PCF is developed in C#/Asp.net with a simple 3-tier architecture that allows the development of independent features among developers. This allows developers to work independently on the data layer, on the business layer, and on the presentation layer. We deal with a special case of SCRUM process. In fact, special development conditions called for some changes of the SCRUM process.

The development was carried out by four part-time developers (Master's students) with 2 to 3 years' experience in software development. Developers work in non-overlapping hours and, to manage a good level of communication, an online forum is used for the daily meeting, as prescribed by Moonlight SCRUM [43]. Moreover, sprint retrospectives, planning, and retrospective discussions are led by means of an online integrated tool (http://www.rallydev.com), which allows us to record sprint reports, manage product backlog, and draw burn-down charts.

5.2 Case study design

5.2.1 the goal

We formulate the goal for our study following the GQM approach [39] as:

analyze the development process
for the purpose of evaluating the effectiveness of estimation measures
from the viewpoint of the developers
in the context of a moonlight SCRUM development process

5.1.2 measures

Since we collect measures to predict effort, a characteristic of the measures we use is that they can be measured before development. So, in principle we expect that it is possible to build a model that, by linking the development effort to the measures, provides an estimation tool that can be used in conjunction with, and possibly even in place of, the usual agile estimation techniques.

Another characteristic of the measures is that they must be fast and easy to collect, since they have to fit in an agile process, where little time and effort can be dedicated to measurement activities. Moreover, the proposed measures are easy to collect, so that any developer can perform the measurement without problems.

To measure user stories, we considered the usage of traditional functional size measures, possibly adapted to the agile context. However, plain function points such as IFPUG (International Function Point User Group) [40] or COSMIC function point [41] measures could not be used. In fact, we noticed several problems, including the following ones:

- The most popular functional size measures use processes (Elementary process or Functional process) as the element to be measured. This is reasonable when the smallest development step (for instance, a sprint in a regular SCRUM process, or an iteration in a RUP process) addresses several processes.

However, in our case, the development of a single process could span multiple sprints. Accordingly, knowing the size of a process could hardly help estimate the work to be done in a single sprint.

- Several sprints involved working mainly on the Graphical User Interface (GUI) of the application. So, functional size measures would not help estimate the effort required.
- Implementation-level details (like the number of interactions with the server or the number of database tables involved in the operations).

Based on the aforementioned constraints, we collected the following measures during the planning game:

- Actual effort: number of hours spent per user story. This information is tracked by developers
- Story Type: we collect this information so as to classify the user stories based on the type of development.
  - New feature: user stories that involve the creation of a new feature.
  - Maintenance: bug fixing or requirement changes for an existing feature.
- Functional measures. Since standard Function Points such as IFPUG or FISMA require a lot of effort to be collected, and most of required information is not available in our context, we opt for the Simplified Function Points (SiFP) [42].

SiFP are calculated as SiFP= 7 * #DF + 4.6 * #TF where #DF is the number of data function (also known as logic data file) and #TF is the number of elementary processes (also known as transactions).

We collect SiFP instead of IFPUG Function Points, since SiFP provides an “agile” and simplified measure, compatible with IFPUG Function Points [42]. Moreover, before running this study, we asked our developers what information they take into account when estimating a user story. All developers answered that they consider four pieces of information, based on the complexity of implementing the GUI and the number of functionalities to be implemented. They usually consider each GUI component as a single functionality that requires the sending or receiving of the information to the database. The complexity of the communication is related to the number of tables involved in the SQL query. For these reasons, we also consider the following measures:

- GUI Impact: null, low, medium, high: complexity of the GUI implementation. It is a subjective measure whose value is provided by the developers.
- # GUI components added: number of data fields added (e.g., Html input fields)

- # GUI components modified: number of data fields modified
- # database tables: number of database tables used in the SQL query.

We can consider this last measure as a functional size measurement with a very low level of granularity, even though not directly comparable to SiFP or IFPUG Function Points.

The measures identified are collected during each sprint meeting by the SCRUM master, in an Excel spreadsheet. After each sprint we collect the actual effort spent for each story, in order to validate results.

Measures must be collected in a maximum of 5 minutes per user story, at the end of the usual SCRUM planning game, so as to not influence the normal execution of the required SCRUM practices. Developers were informed, through an informed consent that the information is collected for research purposes and will never be used to evaluate them.

5.3 Study results

We ran the study analyzing the data for 4 months. We ran 6 sprints of three weeks, each with 4 developers working part-time for the entire period.

Table 186 reports descriptive statistics on the user stories per story type: the vast majority of the user stories are related to the development of new features (73%) while only 27% to maintenance.

Considering GUI impact as shown in Table 186 we can see that most of the user stories are related to the development of graphical features with high or medium complexity.

Functional measures have been collected only for 55 user stories (40.4%) since the remaining user stories do not contain enough information for functional size measurement (e.g., GUI features do not deal with data transactions).

As expected, the number of GUI components added or modified increase paired with the GUI impact while unexpectedly, the higher the GUI impact, the lower is the average number of hours required for implementing a user story.

		All	New Feature	Maintenance
# User stories		136	99(73%)	37 (27%)
Effort per user story (hours)	**Avg**	3.16	3.68	1.96
	Median	2.00	2.00	2.00
	Std. Dev	2.91	3.28	1.01

Table 186: Actual effort per story type

GUI Impact		Story Type		
		All	New	Maintenance
Null	#User	11	6	5
	AVG (hours)	3.12	1.91	1.6
	AVG	5.27	3.67	0.2
Low	#User	30	26	4
	AVG (hours)	3.68	2.46	1
	AVG	1.33	1.44	1
Medium	#User	40	30	10
	AVG (hours)	1.96	3.50	1.70
	AVG	5.02	6.13	0
High	#User	55	37	18
	AVG (hours)	1.30	4.90	2.20
	AVG	8.28	7.89	9.05

Table 187: Effort and GUI component added or modified (GUI components) per user story per GUI impact

Descriptive statistics for the SiFP collected for the user stories show that user stories with a null GUI Impact (user stories that do not deal with the user interface) have the higher number of SiFP, followed by the stories with a high GUI impact. The value are reported in Table 188.

GUI Impact		Story Type		
		All	New Feature	Maintenance
All	#User Stories	55	47	8
	AVG (SiFP)	6.1	5.76	8.58
Null	#User Stories	7	2	5
	AVG (SiFP)	9.12	6.4	12.51
Low	#User Stories	19	18	1
	AVG (SiFP)	4.66	4.8	2.2
Medium	#User Stories	22	20	2
	AVG (SiFP)	5.69	6.06	1.96
High	#User Stories	7	7	0
	AVG (SiFP)	8.79	8.79	/

Table 188: SiFP per user story per GUI impact

After the analysis of descriptive statistics, we investigated the correlations between actual effort and:

- GUI components added, modified and database tables
- GUI components (added + modified)
- SiFP

Here we report the results for all user stories and for each GUI impact and story type, to understand if this information can improve effort estimation accuracy.

Correlation between effort and SiFP

The analysis of correlations between SiFP and effort reported in all user stories does not provide any statistically significant result (Table 191– column “All Projects” and Figure 45), showing a very low goodness of fit (MMRE=81.4%, MdMRE=135.3%).

	All Projects	GUI Impact											
		Null			Low			Medium			High		
Story Type		All	Feat.	Main.	All	Feat.	Main.	All	Feat.	Main.	All	Feat.	Main.
#User Stories	55	7	2	5	19	18	1	22	20	2	7	7	0
pearson	0.065	0.391	/	0.383	0.660	0.669	/	-0.068	-0.073	/	-0.370	-0.370	/
p-value	0.320	0.193	/	0.262	0.001	0.001	/	0.382	0.380	/	0.207	0.207	/
R^2	0.004	0.153	/	0.147	0.436	0.448	/	0.005	0.005	/	0.137	0.137	/

Table 189: Correlations among effort and SiFP

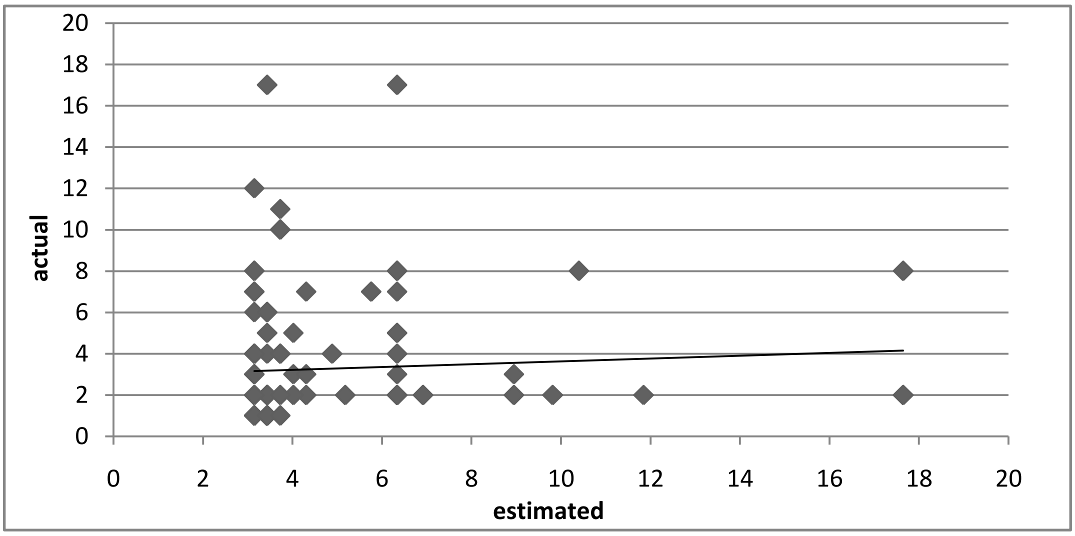


Figure 45: Actual effort vs estimated effort with SiFP

The analysis was then carried out by clustering stories per story types and GUI impact. Results obtained after the clustering show the same behavior, except for stories implementing new features with a low GUI impact (Column “GUI Impact Low – Features”). In this case, results are statistically relevant but with a very low goodness of fits. (MMRE=147%, MdMRE=111%).

Correlation between effort and number of GUI components added or modified

The correlation between the actual effort and the number of GUI components added or modified shows a similar pattern to the previous one in Table 190.

	All Projects	GUI Impact											
		Null			Low			Medium			High		
Story Type		All	Feat.	Main.	All	Feat.	Main.	All	Feat.	Main.	All	Feat.	Main.
#User Stories	136	11	6	5	30	25	5	40	30	10	55	36	19
pearson	0.071	-0.138	0.146	-0.211	0.191	0.190	/	0.436	0.396	0.588	-0.196	-0.217	0.040
p-value	0.207	0.343	0.391	0.366	0.156	0.181	/	0.002	0.015	0.037	0.076	0.102	0.437
R^2	0.005	0.019	0.021	0.045	0.037	0.036	/	0.190	0.156	0.346	0.038	0.047	0.002

Table 190: Correlations among effort and GUI components added or modified

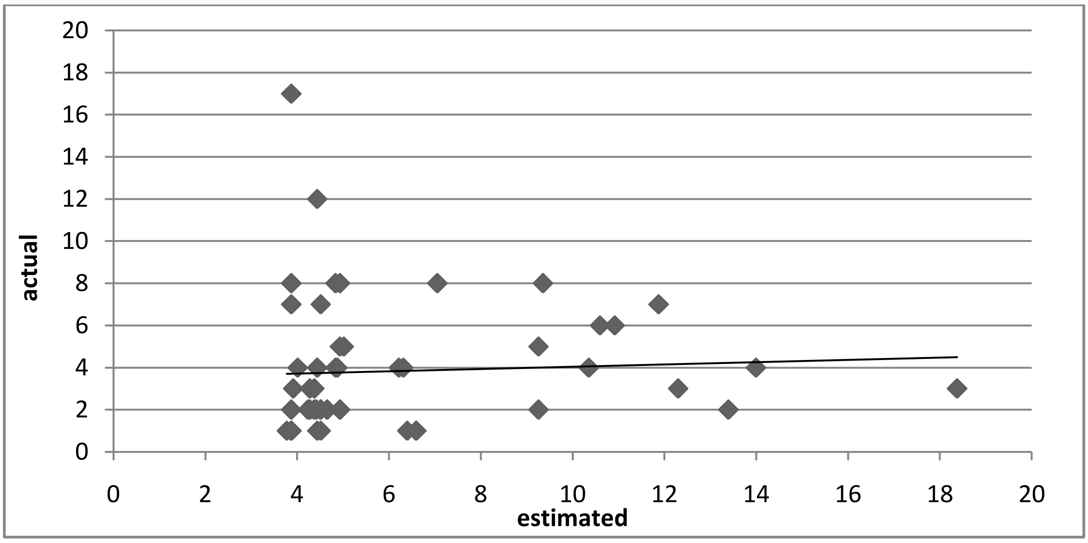


Figure 46: Actual effort vs estimated effort with GUI components added + modified

Only the analysis of stories with a medium GUI impact provides statistically significant results but, together with the analysis of the other types of stories, there is a very low correlation with a very low goodness of fit. (MMRE=71.3%, MdMRE=140.1%). Results are also confirmed by grouping user stories by story type and impact.

Finally, the multivariate correlations among GUI components added, modified and database tables provides statistically significant results paired with a low correlation. Moreover, multivariate correlation does not increase the goodness of fit either Table 191

		GUI Comp Added	GUI Comp Modified	Database Tables
	#Projects	138	138	138
Pearson	Actual Effort	0.212	-0.033	0.130
	GUI Comp Added	1.000	0.272	0.391
	GUI Comp Modified	0.272	1.000	0.377
	Database Tables	0.391	0.377	1.000
p-value	Actual Effort	0.006	0.351	0.0064
	GUI Comp Added		0.001	0.000
	GUI Comp Modified	0.001		0.000
	Database Tables	0.000	0.000	
	R^2		0.061	

Table 191:Multivariate correlation among actual effort and GUI components added, modified and data tables

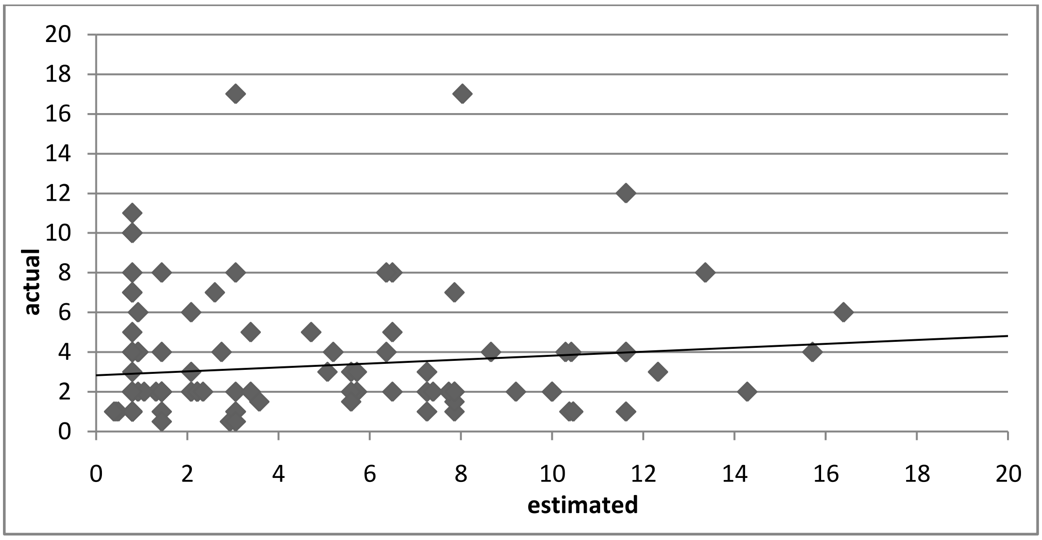


Figure 47: Actual effort vs estimated effort with GUI components added, modified and database tables involved

Correlation between effort and developers' estimated effort

To understand if the results are due to errors in the effort estimation made by our developers, we finally analyze the accuracy of the effort estimation carried out by our developers. We compared the actual effort with the effort estimated before implementing the user story (see in Table 191

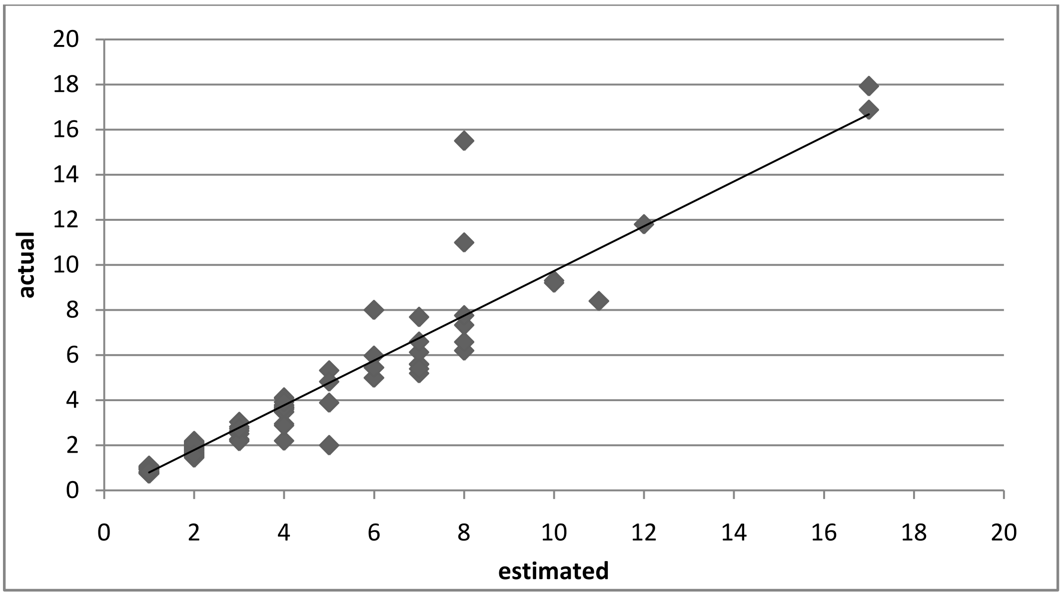


Figure 48: Actual effort vs developers' estimated effort

Results show very accurate estimates, with a very low average error (MMRE=13.5% MdMRE=9.35%). The low error is probably due to the nature of the user stories in Moonlight Scrum, usually smaller than common user stories in SCRUM. However, as expected, the accuracy decreases when the effort planned per user story is higher. This confirms that in our project context, expert estimation is still much better than data-driven estimation, based on functional measurement.

5.4 Results discussion

The immediate result of this study is the low prediction power of functional size measures in SCRUM. Unexpectedly, the prediction accuracy of SiFP compared to the accuracy of experience-based predictions is dramatically low.

Since SiFP can easily replace the more common IFPUG function points with very low error [42], it appears that functional size measures are not suitable for predicting the effort in Moonlight Scrum. Moreover, no correlations are found between the effort and the information commonly used by our developers to estimate user stories (GUI components and database tables). Again, the lack of correlation is probably due to the low complexity and the small effort needed to implement a story. Results are based only on the analysis of one development process, based on a relatively small codebase (12500 effectives lines of code). Concerning internal validity of the study, developers are Master's students, with a limited experience (2-3 years) in software development with at least one year of experience in SCRUM. As for external validity, this study focuses on Moonlight SCRUM, a slightly modified version of SCRUM. We expect some variations when applying the same approach to a full-time development team, working on a plain SCRUM process. Regarding the reliability of this study, results are not dependent on the subjects or the application developed. We expect similar results for the replication of this study with a Moonlight SCRUM process.

In this work, we analyzed the development of a Moonlight SCRUM process so as to understand if it is possible to introduce agile metrics to the SCRUM planning game. With this study, we contribute to the body of knowledge by providing an empirical study on the identification of measures for Agile, and in particular SCRUM, effort estimation. Results of our study show that SiFP do not help improve the estimation accuracy in Moonlight SCRUM. Moreover, the accuracy does not increase considering other measures usually considered by our developers when they evaluate the effort required to develop a user story.

CHAPTER 6 - CONCLUSION AND FUTURE WORK

Cost and effort estimation in software projects have been investigated for several years. Nonetheless, compared to other engineering fields, there are still a huge number of projects that fail in different phases due to effort prediction errors.

Several effort estimation models have been defined based on user experience or on previous project results but, to the best of our knowledge, no studies have tried to estimate the remaining effort after some phase based on the effort spent up to that phase, so as to easily track the effort status in ongoing projects.

The goal of our work is to improve existing estimation models by monitoring and estimating project costs after each development phase. Our approach can be used to predict and monitor project effort during ongoing projects for the next development phase or for the rest of the project. The result of this work is a set of estimation models for effort estimation, based on a set of context factors, such as the domain of application developed, size of the project team and other characteristics. Moreover, in this work we do not aim at defining a model with generic parameters to be applied in similar context, but we define a mathematical approach so as to customize the model for each development team.

We started our work with a literature review, to understand strengths and weaknesses of the existing effort estimation models in re-estimating the remaining project effort. The result of this review show that existing models support the estimation in the early phases. A follow up estimation, to track the effort status requires the re estimation of the whole project.

After the analysis of existing estimation models we propose the approach adopted in this work. We propose to apply Ordinary Least Squares regression (OLS) to investigate the existence of correlations between project phases in the company data-set. While building the OLS models we analyzed the data set and removed the outliers to prevent them from unduly influencing the OLS regression lines obtained. In order to validate the result we apply a 10-fold cross-validation assessing the accuracy of the results in terms of R^2, MMRE and MdMRE.

The model has been applied to two different case studies. First, we analyzed a large number of projects developed by means of the waterfall process. Then, we analyzed an Agile process, so as to understand if the developed model is also applicable to agile methodologies.

Then, we investigated if the prediction accuracy can be improved considering other common context factors such as project domain, development language, development platform, development process, programming language and number of Function Points.

We analyzed projects collected in the ISBSG dataset and, considering the different context factors available, we run a total of 4500 analysis, to understand which are the more suitable factors to be applied in a specific context. The results of this first case study show a set of statistically significant correlations between: (1) the effort spent in one phase and the effort spent in the following one; (2) the effort spent in a phase and the remaining effort; (3) the cumulative effort up to the current phase and the remaining effort. However, the results also show that these estimation models come with different degrees of goodness of fit. Finally, including further information, such as the functional size, does not significantly improve estimation quality.

As for internal validity for this first study, we tried to remove threats as much as possible by filtering data and removing all of those data that did not appear to be complete in the values available for phase effort. As for external validity, the sample is somewhat heterogeneous, so the results we obtained may not be entirely applicable for specific subsets of projects, e.g., projects that use the same programming language or projects belonging to the same application domain.

In the second case study, a project developed with an agile methodology (Moonlight Scrum) has been analyzed. In this case, we want to understand if is possible to use our estimation approach, so as to help developers to increase the accuracy of the expert based estimation.

Since in SCRUM, effort estimation is carried out at the beginning of each sprint, the usage of functional size measures, specifically selected for the type of application and development conditions, is expected to allow for more accurate effort estimates. The study shows that developers' estimation is more accurate than those based on functional measurement showing that, easy to collect functional measures do not help developers in improving the accuracy of the effort estimation in Moonlight SCRUM.

Results of this second case study are based only on the analysis of one development process, with a relatively small codebase (12500 effectives lines of code). Concerning internal validity of the study, developers are master students, with a limited experience (2-3 years) in software development with at least one year of experience in SCRUM. As for external validity, this study focuses on Moonlight SCRUM, a slightly modified version of SCRUM. We expect some variations in applying the same approach to a full time development team, working on a plain SCRUM process. Regarding the reliability of this study, results are not dependent by subjects or by the application developed. We expect similar results for the replication of this study with other Moonlight SCRUM processes.

The result of the application of our proposed model show that OLS could be successfully applied to iteratively estimate the effort in projects developed with Waterfall process while, cannot be used in the context of Moonlight Scrum processes.

These models derived in our work can be used by project managers and developers that need to estimate or control the project effort in a development process. Moreover, these models can also be used by the developers to track their performances and understand the reasons of effort estimation errors.

Finally the model help can be used by project managers to react as soon as possible and reduce project failures due to estimation errors.

APPENDIX A: DATA ANALYSIS

	Plann. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	759.955	437.510	432.784	464.238	155.345
Interc.	0.593	1.126	1.126	0.282	0.330

Table 192: previous phase vs next phase

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	759.955	93.980	2164.554	313.685	220.039
Interc.	0.593	1.118	2.038	0.333	0.208

Table 193: previous phase vs next phase – IGPUG<1000

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.		0.710	1.315	0.310	0.341
Interc.		605.526	928.992	324.396	111.508

Table 194: previous phase vs next phase – Banking

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	0.472	2.324	0.926	0.078	0.196
Interc.	3113.133	710.298	1165.577	530.051	15.228

Table 195: previous phase vs next phase – Communications

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	0.773	0.621	0.560	0.485	0.041
Interc.	547.765	317.277	1369.677	70.470	355.771

Table 196: previous phase vs next phase – Manufacturing

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	2.828	2.167	0.091	0.817	0.668
Interc.	73.386	-208.379	1786.813	-270.891	-237.502

Table 197: previous phase vs next phase – Public administration

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	2.828	2.167	0.091	0.817	0.668
Interc.	73.386	-208.379	1786.813	-270.891	-237.502

Table 198: previous phase vs next phase – MF

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	6.046	1.303	0.321	0.719	0.418
Interc.	-1093.322	737.271	2023.654	-573.426	-79.023

Table 199: previous phase vs next phase – Multi

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	0.419	1.074	2.048	0.177	0.325
Interc.	527.269	72.363	1380.596	560.567	74.204

Table 200: previous phase vs next phase – PC

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	1.912	3.754	1.765	0.414	0.164
Interc.	317.960	-274.368	302.398	49.426	151.024

Table 201: previous phase vs next phase – MR

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	0.877	0.845	1.134	0.394	0.167
Interc.	358.831	290.778	755.155	160.327	130.163

Table 202: previous phase vs next phase – Waterfall

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	1.510		4.296	0.257	0.197
Interc.	780.312		1135.846	193.118	144.804

Table 203: previous phase vs next phase – Data modelling

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	639.097	520.986	1680.334	499.968	143.97
Interc.	0.593	0.522	1.247	0.248	0.221

Table 204: previous phase vs next phase – Java

	Plan. vs Spec.	Spec. vs Design	Design vs Build.	Build. vs Test.	Test. vs Deploy.
Coeff.	-633.924	1058.997	1453.96	3.261	72.411
Interc.	4.846	1.275	0.898	0.382	0.096

Table 205: previous phase vs next phase – Visual basic

	Plan.+ IFPUG vs Spec.	Plan.+ IFPUG <1000 vs Spec.	Plan.+ IFPUG >1000 vs Spec.	Plan.and IFPUG <1000 vs Spec.	Plan.and IFPUG >1000 vs Spec.
Coeff.	0.670	0.618		0.533	0.427
				0.825	0.200
Interc.	846.279	417.306		185.441	2392.714

Table 206: previous phase vs next phase – IFPUG multivariate

	Spec.+ IFPUG vs Design	Spec.+ Plan.+ IFPUG vs Design	Spec.+ IFPUG <1000 vs Design	Spec.+ IFPUG >1000 vs Spec.
Coeff.	1.118	0.956	0.722	1.174
Interc.	93.98	-393.178	351.018	142.57
	Spec. and IFPUG <1000 vs Spec.	**Spec and IFPUG >1000 vs Spec.**		
Coeff.	0.494	1.182		
	0.701	-0.254		
Interc.	176.697	785.501		

Table 207: previous phases vs next phase – IFPUG multivariate

	Design.+ IFPUG vs Build	Design. + Spec. + IFPUG vs Build	Design. + Spec. + Plan. + IFPUG vs Build	Design.+ IFPUG <1000 vs Build.
Coeff.	2.038	1.095	0.619	1.940
Interc.	2164.554	2504.332	206.188	755.123
	Design.+ IFPUG >1000 vs Build	**Design and IFPUG <1000 vs Build**	**Design and IFPUG >1000 vs Build**	
Coeff.	1.794	5.966	3.212	
		1.631	1.816	
Interc.	7683.718	-1207.653	-158.18	

Table 208: previous phases vs next phase – IFPUG multivariate

	Build.+ IFPUG vs Test	Build + Design + IFPUG vs Build	Build + Design + Spec. + IFPUG vs Build	Build + Design + Spec. + Plan. + IFPUG vs Build
Coeff.	0.333	0.190	0.166	0.172
Interc.	313.685	1257.913	1365.535	1362.097
	Build + IFPUG <1000 vs Test	**Build + IFPUG >1000 vs Test**	**Build and IFPUG <1000 vs Test**	**Build and IFPUG >1000 vs Test**
Coeff.	0.379		-0.108	0.195
			0.382	0.280
Interc.	0.379		-0.108	0.195

Table 209: previous phases vs next phase – IFPUG multivariate

	Test.+ IFPUG vs Deploy.	Test + Build + IFPUG vs Deploy.	Test + Build + Design + IFPUG vs Deploy.	Test + Build + Design + Spec. + IFPUG vs Deploy.
Coeff.	0.368	0.334	0.617	0.113
Interc.	0.208	-207.456	-1492.842	-130.11
	Test + Build + Design + Spec. + Plan. + IFPUG vs Deploy.	**Test + IFPUG <1000 vs Deploy.**	**Build + IFPUG <1000 vs Deploy.**	**Build and IFPUG >1000 vs Deploy.**
Coeff.	0.116	0.127	0.346	-0.159
			0.211	0.255
Interc.	-183.776	506.552	4.845	734.187

Table 210: previous phases vs next phase – IFPUG multivariate

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.978			0.124	0.539
Interc.	-5.855			269.547	-68.147

Table 211: previous phase vs next phase – Banking and Stand alone

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.				0.453	
Interc.				-0.168	

Table 212: previous phase vs next phase – Communications and Stand alone

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	2.133			0.129	0.421
Interc.	150.217			288.830	-9.136

Table 213: previous phase vs next phase – Banking and MF

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.876	0.838	0.491	0.429	0.406
Interc.	-40.929	223.982	1472.995	81.556	48.857

Table 214: previous phase vs next phase – Banking and PC

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.			1.924	0.179	0.132
Interc.			1210.913	209.522	56.117

Table 215: previous phase vs next phase – Banking and Multi

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.				0.529	
Interc.				-154.887	

Table 216: previous phase vs next phase – Communications and MR

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.				0.695	
Interc.				125.080	

Table 217: previous phase vs next phase – Communications and PC

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.				0.808	
Interc.				-338.313	

Table 218: previous phase vs next phase – Insurance and MF

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.				0.499	0.049
Interc.				-50.896	43.567

Table 219: previous phase vs next phase – Insurance and Multi

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		2.755	0.076	0.542	0.721
Interc.		-322.655	1480.618	269.363	-238.023

Table 220: previous phase vs next phase – Public administration and Multi

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		0.136	0.985	2.369	1.549
Interc.		-569.362	136.545	-98.125	149.258

Table 221: previous phase vs next phase – Banking and Java

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	5.263	1.236	4.125	0.789	0.985
Interc.	-12.562	1369.596	985.125	235.125	54.236

Table 222: previous phase vs next phase – Banking and Cobol

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		2.569	0.213	0.895	3.254
Interc.		1.569	45.2360	459.236	1265.236

Table 223: previous phase vs next phase – Public administration and Java

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.136		0.084		
Interc.	325.747		322.557		

Table 224: previous phase vs next phase – Banking and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		0.125	2.569	0.129	2.147
Interc.		125.369	-45.236	369.789	1254.569

Table 225: previous phase vs next phase – Banking and COSMIC<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		2.569	0.569	0.147	
Interc.		125.369	1247.589	-35.236	

Table 226: previous phase vs next phase – Manufacturing and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		2.789	0.128	0.639	0.859
Interc.		-126.369	4.239	125.789	1893.254

Table 227: previous phase vs next phase – Public administration and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.236	1.569	0.459	0.895	0.125
Interc.	142.896	-48.215	1478.896	825.123	125.345

Table 228: previous phase vs next phase – Client server and PC

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.789	0.145	0.369	1.569	0.023
Interc.	2.369	12.596	458.562	-47.123	1489.510

Table 229: previous phase vs next phase – Client server and Multi

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.236			-0.120	0.784
Interc.	148.256			1459.263	-142.126

Table 230: previous phase vs next phase – Stand alone and MF

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.123			0.695	0.458
Interc.	12.145			236.520	-12.036

Table 231: previous phase vs next phase – Stand alone and PC

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		0.145	0.632	0.874	
Interc.		1458.203	0.895	14.012	

Table 232: previous phase vs next phase – Client server and Waterfall

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	3.816			0.268	
Interc.	1627.05			118.329	

Table 233: previous phase vs next phase – Client server and Data modelling

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.005	0.751	3.553	0.422	0.247
Interc.	511.882	613.701	-94.576	-89.505	268.029

Table 234: previous phase vs next phase – Stand alone and Data modelling

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.126	0.458	0.963	1.365	2.147
Interc.	147.203	-78.250	1456.251	2365.210	8.254

Table 235: previous phase vs next phase – Client server and Java

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.149			1.023	2.102
Interc.	12.478			145.201	458.874

Table 236: previous phase vs next phase – Client server and Visual basic

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.102	0.146	0.365	2.478	0.569
Interc.	895.236	1254.789	2365.780	456.014	-125.478

Table 237: previous phase vs next phase – Client server and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		1.414			0.430
Interc.		183.440			215.271

Table 238: previous phase vs next phase – MF and Waterfall

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.458			0.269	0.206
Interc.	1003.680			219.614	188.199

Table 239: previous phase vs next phase – MF and Data modelling

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.989	0.777	1.825	0.562	0.142
Interc.	353.450	289.744	1028.433	-126.276	162.961

Table 240: previous phase vs next phase – PC and Waterfall

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.202			0.197	0.127
Interc.	401.652			70.619	114.640

Table 241: previous phase vs next phase – PC and Data modelling

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.976			0.346	0.218
Interc.	1109.777			129.064	193.945

Table 242: previous phase vs next phase – MF and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.			3.398	0.603	
Interc.			523.650	-526.584	

Table 243: previous phase vs next phase – Multi and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.768	1.174	1.988	0.193	0.656
Interc.	173.353	-283.995	2956.549	448.909	64.449

Table 244: previous phase vs next phase – PC and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.320			0.466	0.236
Interc.	1023.589			-153.939	151.084

Table 245: previous phase vs next phase – MR and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		1.403		0.349	0.436
Interc.		174.061		128.535	43.551

Table 246: previous phase vs next phase – Waterfall and COSMIC<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.348			0.270	0.195
Interc.	804.006			233.545	164.383

Table 247: previous phase vs next phase – Data modelling and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	2.563			1.245	0.984
Interc.	145.636			478.597	-69.236

Table 248: previous phase vs next phase – Data modelling and Cobol

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.			0.236	0.145	
Interc.			.456.256	125.478	

Table 249: previous phase vs next phase – Java and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		0.256	-0.250	1.235	0.526
Interc.		458.236	1253.654	985.563	54.856

Table 250: previous phase vs next phase – Java and COSMIC<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.965			0.859	0.245
Interc.	1236.895			452.215	895.478

Table 251: previous phase vs next phase – Cobol and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		1.236	0.985	1.458	.0125
Interc.		3265.256	214.569	985.526	254.123

Table 252: previous phase vs next phase – Cobol and COSMIC>=1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		2.125	0.236		
Interc.		12.569	546.236		

Table 253: previous phase vs next phase – Visual basic and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.		1.236	0.215		
Interc.		550.693	125.366		

Table 254: previous phase vs next phase – Banking and Stand alone and MF

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	2.133			0.129	0.421
Interc.	150.217			288.830	-9.136

Table 255: previous phase vs next phase – Banking and MF and Data modelling

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.638			0.133	0.345
Interc.	264.396			315.373	12.519

Table 256: previous phase vs next phase – Stand alone and MF and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.526			0.5126	0.412
Interc.	1458.369			956.236	125.255

Table 257: previous phase vs next phase – Stand alone and PC and Data modelling

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.569			0.120	0.963
Interc.	879.22			12.589	1254.236

Table 258: previous phase vs next phase – Stand alone and MF and Cobol

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.443			0.269	0.205
Interc.	1064.272			232.564	196.050

Table 259: previous phase vs next phase – MF and Data modelling and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	0.969			0.197	0.369
Interc.	145.693			125.456	326.548

Table 260: previous phase vs next phase – MF and Data modelling and COSMIC<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.177			0.197	0.098
Interc.	466.491			72.052	145.365

Table 261: previous phase vs next phase – PC and Data modelling and IFPUG<1000

	Plan. vs Spec	Spec. vs Design	Design vs Build	Build vs Test	Test vs Deploy.
Coeff.	1.638			0.133	0.345
Interc.	264.396			315.373	12.519

Table 262: previous phase vs next phase – Banking and MF and Data modelling and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	3578.87	2670.119	3008.702	310.617
Interc.	6.767	4.938	1.378	0.561

Table 263: correlation coefficients - previous phase vs remaining project

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	1900.142	2883.944	1570.661	437.242
Interc.	5.157	2.492	2.401	0.420

Table 264: correlation coefficients - previous phase vs remaining project – Banking

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		1647.359		89.148
Interc.		1.395		0.519

Table 265: correlation coefficients - previous phase vs remaining project – Communication

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		3866.660	1543.187	527.197
Interc.		1.090	1.072	0.055

Table 266: correlation coefficients - previous phase vs remaining project – Insurance

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	104.278	1642.118	2094.727	39.909
Interc.	0.091	1.907	0.956	0.556

Table 267: correlation coefficients - previous phase vs remaining project – Manufacturing

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		-309.439	2095.932	-87.816
Interc.		4.629	0.602	0.759

Table 268: correlation coefficients - previous phase vs remaining project – Public Administration

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	578.419	4449.064	981.384	
Interc.	0.020	1.962	5.297	

Table 269: correlation coefficients - previous phase vs remaining project – Client server

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	150.338	1914.221		
Interc.	0.107	1.866		

Table 270: correlation coefficients - previous phase vs remaining project – Stand alone

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	257.437	1272.836		
Interc.	0.022	4.050		

Table 271: correlation coefficients - previous phase vs remaining project – Multi tier

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	303.567	2471.516	627.547	
Interc.	0.037	1.309	2.315	

Table 272: correlation coefficients - previous phase vs remaining project – MR

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	460.412	2981.973	1396.698	
Interc.	0.050	2.210	2.749	

Table 273: correlation coefficients - previous phase vs remaining project – MF

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	387.625	1576.426	2286.946	
Interc.	0.022	3.785	2.570	

Table 274: correlation coefficients - previous phase vs remaining project – PC

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	145.522	1897.585	2524.978	
Interc.	0.024	4.855	1.077	

Table 275: correlation coefficients - previous phase vs remaining project – Multi

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	432.829			
Interc.	0.054			

Table 276: correlation coefficients - previous phase vs remaining project – Data modelling

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	165.658	3514.720	2575.655	
Interc.	0.079	1.332	1.786	

Table 277: correlation coefficients - previous phase vs remaining project – Java

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		3261.480	1212.363	
Interc.		2.204	2.505	

Table 278: correlation coefficients - previous phase vs remaining project – Cobol

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			3418.519	
Interc.			1.705	

Table 279: correlation coefficients - previous phase vs remaining project – C++

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			2166.840	
Interc.			1.074	

Table 280: correlation coefficients - previous phase vs remaining project – Visual basic

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	33.408	1743.242	951.637	
Interc.	0.102	2.199	3.270	

Table 281: correlation coefficients - previous phase vs remaining project – IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	1150.778	7138.626	10217.873	
Interc.	0.034	1.866	2.168	

Table 282: correlation coefficients - previous phase vs remaining project – IFPUG>=1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	133.510	2617.705	796.698	
Interc.	0.127	5.392	4.423	

Table 283: correlation coefficients - previous phase vs remaining project – COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	2.343	646.309		263.862
Interc.	0.095	3.521		0.191

Table 284: correlation coefficients - previous phase vs remaining project – Banking and Stan alone

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		981.560		74.595
Interc.		1.741		0.450

Table 285: correlation coefficients - previous phase vs remaining project – Communication and Stan alone

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		3026.966		70.253
Interc.		2.520		0.171

Table 286: correlation coefficients - previous phase vs remaining project – Banking and MR

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	850.268			595.055
Interc.	8.964			0.470

Table 287: correlation coefficients - previous phase vs remaining project – Banking and MF

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	30.667	2549.235	1899.136	291.875
Interc.	0.144	2.071	1.942	0.458

Table 288: correlation coefficients - previous phase vs remaining project – Banking and PC

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			1676.433	277.650
Interc.			2.420	0.199

Table 289: correlation coefficients - previous phase vs remaining project – Banking and Multi

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		1452.879		-132.316
Interc.		1.399		0.529

Table 290: correlation coefficients - previous phase vs remaining project – Communications and MR

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		2063.004		215.721
Interc.		2.458		0.702

Table 291: correlation coefficients - previous phase vs remaining project – Communications and PC

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		585.371		698.976
Interc.		2.850		0.041

Table 292: correlation coefficients - previous phase vs remaining project – Insurance and MF

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		1762.973	2653.971	285.144
Interc.		1.777	0.646	0.454

Table 293: correlation coefficients - previous phase vs remaining project –Manufacturing and PC

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	33.408	877.679		292.193
Interc.	0.102	2.285		0.155

Table 294: correlation coefficients - previous phase vs remaining project – Banking and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	133.510	2973.558		762.724
Interc.	0.127	5.587		0.469

Table 295: correlation coefficients - previous phase vs remaining project – Banking and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.		1046.642		67.330
Interc.		1.018		0.451

Table 296: correlation coefficients - previous phase vs remaining project – Communications and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	214.611		1906.526	160.374
Interc.	0.022		1.030	0.405

Table 297: correlation coefficients - previous phase vs remaining project – Manufacturing and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	567.645			
Interc.	0.012			

Table 298: correlation coefficients - previous phase vs remaining project – Client server and PC

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	17.125		1330.188	
Interc.	0.048		2.867	

Table 299: correlation coefficients - previous phase vs remaining project – Client server and Waterfall

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	-804.187		-258.469	
Interc.	0.376		7.716	

Table 300: correlation coefficients - previous phase vs remaining project – Client server and Java

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			-708.290	
Interc.			6.611	

Table 301: correlation coefficients - previous phase vs remaining project – Client server and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			-461.762	
Interc.			6.017	

Table 302: correlation coefficients - previous phase vs remaining project – Client server and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			854.965	
Interc.			0.384	

Table 303: correlation coefficients - previous phase vs remaining project – MF and Waterfall

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.				
Interc.				

Table 304: correlation coefficients - previous phase vs remaining project – Multi and Waterfall

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	53.671		1632.542	
Interc.	0.038		2.963	

Table 305: correlation coefficients - previous phase vs remaining project – PC and Waterfall

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			1123.554	
Interc.			3.627	

Table 306: correlation coefficients - previous phase vs remaining project – MF and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	55.217		1914.925	
Interc.	0.106		0.955	

Table 307: correlation coefficients - previous phase vs remaining project – PC and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			-129.457	
Interc.			9.959	

Table 308: correlation coefficients - previous phase vs remaining project – PC and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	-332.294			
Interc.	0.245			

Table 309: correlation coefficients - previous phase vs remaining project – Data modelling and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	2213.073			
Interc.	0.013			

Table 310: correlation coefficients - previous phase vs remaining project – Data modelling and IFPUG>=1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	-791.974		1200.376	
Interc.	0.348		8.096	

Table 311: correlation coefficients - previous phase vs remaining project – Java and IFPUG<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			-694.239	
Interc.			8.387	

Table 312: correlation coefficients - previous phase vs remaining project – Java and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.			1776.223	
Interc.			2.262	

Table 313: correlation coefficients - previous phase vs remaining project – Cobol and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.				
Interc.				

Table 314: correlation coefficients - previous phase vs remaining project – Client server and Multi and COSMIC<1000

	Plann. vs Sum next phases	Spec. vs Sum next phases	Design vs Sum next phases	Build vs Sum next phases
Coeff.	-516.343			
Interc.	0.339			

Table 315: correlation coefficients - previous phase vs remaining project – stand alone and Data modelling and IFPUG<1000

	Spec vs Design	Spec. + Plan. vs Design	Spec. and Plan. vs Design
Coeff.	1.126	0.607	0.015
			0.064
Interc.	437.51	212.476	318.169

Table 316:previous phases vs next phase

	Design vs Build	Design +Spec. vs Build	Design + Spec. + Plan. vs Build	Design and Spec. vs Build	Design and Spec. and Plan. vs Build
Coeff.	1.126	0.780	1.229	0.196	1.267
				1.840	0.756
					2.310
Interc.	432.78	1473.246	1098.473	1463.113	732.423

Table 317: previous phases vs next phase

	Build vs Test	Build + Design vs Test	Build + Design + Spec. vs Test	Build + Design + Spec. + Plan. vs Test
Coeff.	0.282	0.196	0.19	0.164
Interc.	464.238	572.503	526.686	752.882
	Build and Design vs Test	**Build and Design and Spec. vs Test**	**Build and Design and Spec. and Plan. vs Test**	
Coeff.	0.327	0.090	0.007	
	0.157	0.780	0.151	
		0.930	1.114	
			0.296	
Interc.	541.754	561.569	588.134	

Table 318: previous phases vs next phase

	Test vs Deploy.	Test + Build + vs Deploy.	Test +Build + Design + vs Deploy.
Coeff.	0.330	0.297	0.263
Interc.	155.345	80.779	30.478
	Test and Build and Design vs Deploy	**Test and Build and Design and Spec. vs Deploy**	**Test and Build and Design and Spec. and Plan. vs Deploy**
Coeff.	0.225	0.256	0.045
			0.276
Interc.	-538.621	-1098.48	-256.85
	Test and Build and Design vs Deploy	**Test and Build and Design and Spec. vs Deploy**	**Test and Build and Design and Spec. and Plan. vs Deploy**
Coeff.	0.102	0.108	0.149
	0.457	0.490	0.660
	-0.076	-0.082	-1.138
		-0.032	0.113
			0.220
Interc.	-549.56	-536.07	-835.68

Table 319: previous phases vs next phase

	Interc.	Coeff.
design from spec +plan.	343.378	0.075
build from design+spec.	631.920	1.273
build from design+spec.+plan.	845.284	1.323
test. from build+design	165.766	0.331
test. from build+design +spec.	73.459	0.096
test. from build +design+spec.+plan.	-6.945	0.288
deploy. from test.+build	81.414	0.122
deploy. from test.+build+design	2660.607	2.035
deploy. from test.+build+design +spec.	72.008	0.113
deploy. from test. +build+design+spec.+plan.	72.008	0.113

Table 320:previous phase vs next phase – Banking

	Interc.	Coeff.
design from spec +plan.	334.245	0.509
build from design+spec.	651.671	0.283
build from design+spec.+plan.	1217.729	0.367
test. from build+design	2489.853	1.792
test. from build+design +spec.	-120.045	0.309
test. from build +design+spec.+plan.		
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 321: previous phase vs next phase - Client server

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.		
build from design+spec.+plan.		
test. from build+design	251.712	0.145
test. from build+design +spec.	232.112	0.117
test. from build +design+spec.+plan.	232.112	0.117
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 322: previous phase vs next phase - MR

	Interc.	Coeff.
design from spec +plan.	125.905	1.695
build from design+spec.	377.389	0.405
build from design+spec.+plan.	377.389	0.405
test. from build+design	1078.080	1.525
test. from build+design +spec.	578.677	0.205
test. from build +design+spec.+plan.	578.677	0.205
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 323: previous phase vs next phase – MF

	Interc.	Coeff.
design from spec +plan.	-170.568	0.800
build from design+spec.	292.747	0.412
build from design+spec.+plan.	292.747	0.412
test. from build+design	533.047	0.881
test. from build+design +spec.	691.174	2.417
test. from build +design+spec.+plan.	691.174	2.417
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 324: previous phase vs next phase – PC

	Interc.	Coeff.
design from spec +plan.	491.688	0.685
build from design+spec.	-454.800	1.319
build from design+spec.+plan.	-454.800	1.319
test. from build+design	1981.786	1.570
test. from build+design +spec.	-464.482	0.389
test. from build +design+spec.+plan.	-464.482	0.389
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 325: previous phase vs next phase – Multi

	Interc.	Coeff.
design from spec +plan.	260.256	1.673
build from design+spec.	773.083	0.184
build from design+spec.+plan.	773.083	0.184
test. from build+design	1809.356	1.507
test. from build+design +spec.	585.432	0.119
test. from build +design+spec.+plan.	585.432	0.119
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 326: previous phase vs next phase – Java

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	281.193	0.459
build from design+spec.+plan.	281.193	0.459
test. from build+design	830.251	1.768
test. from build+design +spec.	312.409	0.288
test. from build +design+spec.+plan.	312.409	0.288
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 327: previous phase vs next phase – Cobol

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	519.458	1.010
build from design+spec.+plan.	519.458	1.010
test. from build+design	1568.081	1.580
test. from build+design +spec.	1568.081	1.580
test. from build +design+spec.+plan.	1568.081	1.580
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 328: previous phase vs next phase – C++

	Interc.	Coeff.
design from spec +plan.	-4.737	4.053
build from design+spec.	692.155	0.234
build from design+spec.+plan.	692.155	0.234
test. from build+design	696.381	1.909
test. from build+design +spec.	-336.171	0.433
test. from build +design+spec.+plan.	-336.171	0.433
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 329: previous phase vs next phase – Waterfall

	Interc.	Coeff.
design from spec +plan.	698.349	0.732
build from design+spec.	956.248	0.285
build from design+spec.+plan.	956.248	0.285
test. from build+design	1257.491	1.982
test. from build+design +spec.	-448.465	0.356
test. from build +design+spec.+plan.	-448.465	0.356
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 330: previous phase vs next phase – IFPUG<1000

	Interc.	Coeff.
design from spec +plan.	172.632	2.343
build from design+spec.	488.413	0.201
build from design+spec.+plan.	488.413	0.201
test. from build+design	2158.634	1.009
test. from build+design +spec.	856.232	0.056
test. from build +design+spec.+plan.	856.232	0.056
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 331: previous phase vs next phase – COSMIC<1000

	Interc.	Coeff.
design from spec +plan.	583.504	0.886
build from design+spec.	1101.548	0.117
build from design+spec.+plan.	1101.548	0.117
test. from build+design	30.644	4.400
test. from build+design +spec.	427.143	0.135
test. from build +design+spec.+plan.	427.143	0.135
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 332: previous phase vs next phase – Client server and PC

	Interc.	Coeff.
design from spec +plan.	-667.711	1.341
build from design+spec.	-956.127	0.757
build from design+spec.+plan.	-956.127	0.757
test. from build+design	2501.730	1.491
test. from build+design +spec.	2501.730	1.491
test. from build +design+spec.+plan.	2501.730	1.491
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 333: previous phase vs next phase – Client server and Multi

	Interc.	Coeff.
design from spec +plan.	-4.737	1.053
build from design+spec.	1.801	0.199
build from design+spec.+plan.	1.801	0.199
test. from build+design	1755.508	1.438
test. from build+design +spec.	183.594	0.224
test. from build +design+spec.+plan.	183.594	0.224
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 334: previous phase vs next phase – Client server and Waterfall

	Interc.	Coeff.
design from spec +plan.	-114.219	2.869
build from design+spec.	813.267	0.122
build from design+spec.+plan.	813.267	0.122
test. from build+design	1778.095	2.708
test. from build+design +spec.	626.751	0.044
test. from build +design+spec.+plan.	626.751	0.044
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 335: previous phase vs next phase – Client server and Java

	Interc.	Coeff.
design from spec +plan.	31.019	1.050
build from design+spec.	634.722	0.298
build from design+spec.+plan.	634.722	0.298
test. from build+design	1755.508	1.438
test. from build+design +spec.	-154.322	0.369
test. from build +design+spec.+plan.	-154.322	0.369
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 336: previous phase vs next phase – PC and Waterfall

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	288.822	0.463
build from design+spec.+plan.	288.822	0.463
test. from build+design	777.852	1.708
test. from build+design +spec.	777.852	1.708
test. from build +design+spec.+plan.	777.852	1.708
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 337: previous phase vs next phase – MF and Cobol

	Interc.	Coeff.
design from spec +plan.	451.882	0.995
build from design+spec.	629.363	0.129
build from design+spec.+plan.	629.363	0.129
test. from build+design	629.363	0.129
test. from build+design +spec.	629.363	0.129
test. from build +design+spec.+plan.	629.363	0.129
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 338: previous phase vs next phase – PC and Java

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	371.518	0.326
build from design+spec.+plan.	371.518	0.326
test. from build+design	2033.866	0.612
test. from build+design +spec.	2033.866	0.612
test. from build +design+spec.+plan.	2033.866	0.612
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 339: previous phase vs next phase – MF and COSMIC<1000

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	234.713	0.128
build from design+spec.+plan.	234.713	0.128
test. from build+design	234.713	0.128
test. from build+design +spec.	234.713	0.128
test. from build +design+spec.+plan.	234.713	0.128
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 340: previous phase vs next phase – PC and COSMIC<1000

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.		
build from design+spec.+plan.		
test. from build+design	1117.146	1.754
test. from build+design +spec.	1117.146	1.754
test. from build +design+spec.+plan.	1117.146	1.754
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 341: previous phase vs next phase – Java and IFPUG<1000

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	481.587	0.193
build from design+spec.+plan.	481.587	0.193
test. from build+design	3250.037	1.258
test. from build+design +spec.	3250.037	1.258
test. from build +design+spec.+plan.	3250.037	1.258
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 342: previous phase vs next phase – Java and COSMIC<1000

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	281.193	0.459
build from design+spec.+plan.	281.193	0.459
test. from build+design	1786.028	0.636
test. from build+design +spec.	1786.028	0.636
test. from build +design+spec.+plan.	1786.028	0.636
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 343: previous phase vs next phase – Cobol and COSMIC<1000

	Interc.	Coeff.
design from spec +plan.	-4.737	1.053
build from design+spec.	1.801	0.199
build from design+spec.+plan.	1.801	0.199
test. from build+design	1755.508	1.438
test. from build+design +spec.	183.594	0.224
test. from build +design+spec.+plan.	183.594	0.224
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 344: previous phase vs next phase – Client server and PC and Waterfall

	Interc.	Coeff.
design from spec +plan.	218.358	2.170
build from design+spec.	813.267	0.122
build from design+spec.+plan.	813.267	0.122
test. from build+design	-953.049	6.638
test. from build+design +spec.	-953.049	6.638
test. from build +design+spec.+plan.	-953.049	6.638
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 345: previous phase vs next phase – Client server and PC and Java

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	371.518	0.326
build from design+spec.+plan.	371.518	0.326
test. from build+design		
test. from build+design +spec.		
test. from build +design+spec.+plan.		
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 346: previous phase vs next phase – MF and Java and COSMIC<1000

	Interc.	Coeff.
design from spec +plan.		
build from design+spec.	371.518	0.326
build from design+spec.+plan.	371.518	0.326
test. from build+design	1783.028	0.636
test. from build+design +spec.	774.047	0.179
test. from build +design+spec.+plan.	774.047	0.179
deploy. from test.+build		
deploy. from test.+build+design		
deploy. from test.+build+design +spec.		
deploy. from test. +build+design+spec.+plan.		

Table 347: previous phase vs next phase – MF and Cobol and COSMIC<1000

	Plan.and Spec. vs sum next	Plan.and Spec. and Des. vs sum next phases	Plan. and Spec.+ Des. and Build. vs sum next phases	Plan.and Spec. and Des. and Build. and Test vs Deploy
Coeff.	5.097	4.963	0.918	0.219
	3.878	2.358	1.518	0.112
			-0.836	-1.139
			0.517	0.655
				0.149
Interc.	731.567	1.124	-206.062	-830.147
	Spec. and Des. vs sum next	**Spec. and Des. and Build. vs sum next**	**Spec. and Des. and Build. and Test. vs sum next**	**Des. and Build vs sum next**
Coeff.	3.607	1.090	-0.077	0.274
		-0.000	-0.085	0.642
		0.568	0.489	
			0.108	
Interc.	2155.166	43.521	-542.785	
	Des. and Build and Test vs Deploy.	**Build and Test vs Deploy.**		
Coeff.	-0.077	0.276		
	0.458	0.045		
	0.102			
Interc.	-542.785	-256.854		

Table 348: previous phases vs remaining project

	Plan. + Spec. vs sum next	Plan.+ Spec. + Des. vs sum next phases	Plan. + Spec.+ Des. + Build. vs sum next phases	Plan.+ Spec. + Des. + Build. + Test vs Deploy
Coeff.	759.755	238.684	-311.226	-1102.383
Interc.	4.434	2.463	0.489	0.257
	Spec. + Des. + Build. vs sum next	**Spec. + Des. + Build. + Test. vs sum next**	**Des. + Build. + Test. vs sum next**	**Design + Build.+ Test. vs Deploy.**
Coeff.	4.545	4165.807	52.013	-561.254
Interc.	0.494	1.949	0.541	0.245

Table 349: previous phases vs remaining project

REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.

[2] Molokken and Jorgensen, "A review of surveys on software estimation. Empirical Software Engineering". ISESE 2003

[3] Molokken, "A review of studies on expert estimation of software development effort. The journal of system and software" vol.70, pp.37-60, 2004

[4] Jorgensen, "Practical Guidelines for expert-judgment based software effort estimation". IEEE Software, vol 22, issue 3, pp. 57-63, 2005.

[5] Grimstad and Jorgensen, "A preliminary study of sequence effects in judgment-based software development work effort estimation". Software IEE Transactions on Software Engineering, vol.38 no. 3, pp. 677 – 693, 2012.

[6] Jeffrey, Ruhe and Wieczorek, "Using Public Domain Metrics to Estimate Software Development Effort". IEEE, pp. 16-27, 2001.

[7] Jorgensens and Sjoberg, "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy". Information and software Technology, vol 45, pp.123-136, 2003

[8] R.S. Pressman, Software Engineering: a Practitioner's Approach, 5th ed., McGraw-Hill, 2000, pp. 172.

[9] Sentas and Agelis, "Software productivity and effort prediction with ordinal regression. Information and software technology" , vol. 47, pp.17-29, 2005

[10] Mendes, " Cross-company and single-company effort models using the ISBSG Database: a further replicated study". IEEE international symposium on Empirical software engineering, pp.75 – 84, 2006

[11] Seo, Yoon and Bae, "An empirical analysis of software effort estimation with outlier elimination". PROMISE 08, 2008

[12] Ayman Al Ahmar, " Rule based expert system for selecting software development methodology". Journal of technology and applied information technology, vol. 19 no. 2, 2005.

[13] Dygas and Breiter, " Variance of errors and elimination of outliers in the least squares analysis of impedance spectra". Electrochimica Acta, vol. 44, pp. 4163-417, 1999.

[14] Lars Mieritz, “Survey Shows Why Projects Fail”. Gartner report.Retrieved from Gartner Database, 2012

[15] D. Déry, and A. Abran, “Investigation of the Effort Data Consistency in the ISBSG Repository,” Proc. of the 15th International Workshop on Software Measurement (IWSM 05), Montreal, Canada, Shaker-Verlag, pp. 123–136.

[16] M.C. Ohlsson, and C. Wohlin, “An Empirical Study of Effort Estimation During Project Execution,” Proc. of the 6th International Symposium on Software Metrics (SM 99), Boca Raton FL, pp. 91-98.

[17] P. Yiftachel, I. Hadar, D. Peled, E. Farchi, and D. Goldwaser, “The Study of Resource Allocation among Development Phases: An Economics-Based Approach,”Advances in Software Engineering, vol. 2011, 2001, pp. 1-21.

[18] W. Heijstek, and M.R.V. Chaudron, “Evaluating RUP Software Development Processes Through Visualization of Effort Distribution,” Proc. of EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 08), 2008, pp.266-257.

[19] Y. Yang, M. He, M. Li, Q. Wang, and B.W. Boehm, “Phase Distribution of Software Development Effort,” Proc. of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 08), ACM-IEEE, Kaiserslautern, Germany, 2008, pp. 61-69.

[20] S.G. MacDonell, and M.J. Shepperd, “Using Prior-Phase Effort Records for Re-estimation during Software Projects,” Proc. of the 9th International Symposium on Software Metrics, Washington: IEEE Computer Society, 2003, pp. 1-13.

[21] Z. Jiang, P. Naudé, and B. Jiang, “The Effects of Software Size on Development Effort and Software Quality,” International Journal of Computer and Information Science and Engineering, vol. 1, no. 4, 2007, pp. 230-234.

[22] Chatzipetrou, P.; Papatheocharous, E.; Angelis, L.; Andreou, A.S., "An Investigation of Software Effort Phase Distribution Using Compositional Data Analysis," Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on , vol., no., pp.367,375, 5-8 Sept. 2012

[23] Lenarduzzi V. and Taibi D. “Can Functional Size Measures Improve Effort Estimation in SCRUM?” ICSEA 2014. Nice (France)

[24] Briand, L.C., El Emam, K., Bomarius, F. COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment, Proceedings of the 20th International Conference on Software Engineering, 1998, pp. 390-399.

[25] PMBOK - Work Breakdown Structure

[26] Trendowicz, A; Munch, J. and Jeffrey, R. State of the Practice in Software Effort Estimation: A Survey and Literature Review. Software Enginenring Techiniques, 2011, vol. 4980, pp. 232-245

[27] Saleem, B. and Dhavachelvan, P. Analysis of Empirical Software Effort Estimation Models.International Journal of Computer Science and Information Security. 2010, vol.7, no 3, pp.68-77

[28] Khatibi, V and Jawawi, D. Software Cost Estimation Methods: A Review. Journal of Emerging Trends in Computing and Information Sciences. 2010, vol. 2, no 1, pp. 21-29

[29] PMBOK® Guide

[30] Stamelos I, Angelis L, Dimou P, Sakellaris E. 2003. On the use of Bayesian belief networks for the prediction of software productivity. Information and Software Technology 45: 51–60.

[31] Sentas P, Angelis L, Stamelos I. 2005. Software productivity and effort prediction with ordinal regression. Journal of Information & Software Technology 47(1): 17–29.

[32] BBN based approach for improving the software development process of an SME - a case study S. Bibi1,∗,†,I. Stamelos1,G. Gerolimos2 andV. Kollias2

[33] Shepperd, M. and Macdonell, S.G. Evaluating prediction systems in software project estimation. Information and Software Technology, 2012, vol.54, no 8, pp.820-827

[34] COCOMO II model definition manual

[35] http://www.standishgroup.com/

[36] Trendowicz A. Software Cost Estimation, Benchmarking, and Risk Assessment. Springer.

[37] Phan, D., Information Systems Project Management: an Integrated Resource Planning Perspective Model in Department of Management and Information Systems. 1990, Arizona: Tucson.

[38] J. Grenning, “Planning poker or how to avoid analysis paral- ysis while release planning,” Apr. 2002 [Online]. Available: http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf

[39] Caldiera, Victor R Basili1 Gianluigi and Rombach, H. D. 1994. The goal question metric approach. Encyclopedia of software engineering 2, 1994, 528–532.

[40] International Function Point Users Group. “Function Point Counting Practices Manual”, 2004

[41] COSMIC - Common Software Measurement International Consortium. “The COSMIC Functional Size Measurement Method - version 3.0 Measurement Manual” September 2007

[42] L. Lavazza and R. Meli, “An Evaluation of Simple Function Point as a Replacement of IFPUG Function Point”, IWSM - Mensura 2014, Rotterdam, October 2014.

[43] D. Taibi, P. Diebold, and C. Lampasona. “Moonlighting SCRUM: An Agile Method for Distributed Teams with Part-Time Developers Working during Non-Overlapping Hours” Proceedings of the Eighth International Conference on Software Engineering (ICSEA), pp. 318-323, 2013

[44] P. Diebold, L. Dieudonné, and D. Taibi, “Process Configuration Framework Tool”, Euromicro Conference on Software Engineering and Advanced Applications , 2014

[45] K. Schwaber. “Agile Project Management with SCRUM” Microsoft Press, ISBN 9780735619937, 2004

[46] Beck, K., Andres, C.: Extreme programming explained. Embrace change. 2nd ed. Boston: Addison-Wesley, 2005.

[47] Boehm, B.W.: A spiral model of software development and enhancement. IEEE Computer, 21(5), 61-72, (1988).

[48] Sridhar, N, RadhaKanta M., and George Mangalaraj. “Challenges of migrating to agile methodologies. Commun. ACM 48, 5 (May 2005)

[49] The International Software Benchmarking Standards Group (ISBSG). http://www.isgsg.org. Last seen may 2014

[50] Benington, Herbert D. (1 October 1983). "Production of Large Computer Programs". IEEE Annals of the History of Computing (IEEE Educational Activities Department) 5 (4): 350–361. doi:10.1109/MAHC.1983.10102. Retrieved 2011-03-21.