CHALMERS

Human Factors in Software Development: A Systematic Literature Review

Master of Science Thesis in Computer Science and Engineering

Laleh Pirzadeh

Department of Computer Science and Engineering
Division of Networks and Distributed Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2010

Human Factors in Software Development: A Systematic Literature Review

Laleh Pirzadeh

Examiner: Robert Feldt

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden September 2010

ABSTRACT

Background: Software Development Process is a human-centered activity. This fact highlights the impact of human factors on the development process and performance from different perspectives.
Aim: This thesis project aims to identify and characterize human factors influencing the software development process from development lifecycle and software management perspectives.
Method: The systematic literature review initial search resulted in 401 papers among which 67 journal papers (2000-2010) were picked based on two filtering phases. The selected papers were classified and analyzed according to software development phases, software engineering management, human roles and level of human factors focus.
Results: Data extracted from the selected papers was mapped to SWEBOK [125]. Among software development phases (waterfall model) the most common one in human factors studies is "Requirement Engineering" addressed by 67% of the inspected papers followed by "Design" (52%). "Software Project Enactment" is the most common software engineering management phase from human factors perspective (67 papers, 100%) followed by "Initiation and Scope Definition and Software Project Planning" (89%) and "Software Engineering Measurements" (56%). Human plays different roles in development process which has impact on the process performance and success. It can be Customer, Developer or Manager. The main human role was "Developer" covered by 94% of the inspected papers. The highest level of human factors focus among these papers was "Individual" level addressed by 76% papers followed by "Organizational" level. (70%) And finally the most common research method among the selected papers was case study (industrial) being applied in 78% of the empirical studies.
Conclusion: Despite of the human factors impact on software development process and its level of success/failure, there has not been enough focus by primary study researchers on the area. Human factors have been overlooked in late phases of development process as well as software engineering management. In contrast with all other areas of business which their highest concern is the customer, among the inspected papers this role has been overlooked by researchers. As transferring from preplanned to agile development process, there will be more "Interpersonal" level of human factors involved that has not been addressed well by researchers so far.

Keywords: Systematic Literature Review, Human Factors, Software Development Process, Software Engineering Management

ACKNOWLEDGEMENT

It’s an honor for me to show my gratitude to those who made this thesis possible. Thank you Ana for your supervision, guidance, patience and useful feedback. I’ve learnt a lot from our discussions and hope to apply them in my future research carrier as a PhD student. Thank you Robert for your wise guidance and helping me find my direction during the project.

My deepest gratitude goes to my family for being there for me during all the sad times and good times. Dad, without your help I could have never been able to achieve my goals and be where I am today. You helped me to believe in myself and become stronger in every step I take along the way. Mom, Thanks for understanding me and supporting me with your love and wisdom whenever I thought I cannot take it anymore. You gave me the spirit to hold on. Panti, Nima and my second soul Zhaleh, thanks for all your support and help during this journey. You were always there for me when I needed a shelter and helped me with your great advice and having my back.

Last but not the least I would like to show my deep appreciation to Erland for giving me the chance of continuing my dream as a researcher and PhD student. Thanks a lot for understanding and being so patient with me. You gave me the hope to take the final steps of my project stronger.

Thank you,
Laleh Pirzadeh

Table of Contents

1 INTRODUCTION

In this chapter a brief introduction of this project, overview of the Systematic Literature Review and some background are provided in order to provide a better understanding for the reader.

1.1 Introduction

During the last three decades, software development process has been the centre of attention for many software engineering researchers. [1-5] Regardless of the details of products and customers, software is an area where almost every industry is involved with. A successful software development process is therefore crucial for industries in various fields.

Based on software development being a human-centered process, human factors have a great impact on the process and its performance. This can be explained by the impact of human role in development process i.e. affecting the software development market by being a customer, influencing the development process phases by being a developer and finally having a noticeable impact on process performance and success by being a manager. Human factors in this process can be studied from different perspectives such as psychological, cognitive, management and technical aspects. Moreover, human factors have different levels of impact in the process varying from organizational and interpersonal to individual.

Although human factors have been proved to have impact on software development process, unfortunately they have been overlooked by the researchers in the software engineering and development research areas. Thus, there seems to be a need to identify and characterize human factors and their impact on development process. A systematic review over software development human factors could highlight the research needs and consequently improve the research in the SE field.

The idea behind this thesis project is to investigate and review studies related to software development, management and organizational/individual and interpersonal human factors via an extensive systematic review. The systematic review includes different categories; each covering specific human factor keyword and introducing new aspect that should be considered in Software Development Process. The main focus of the thesis is to analyze human factors from two main aspects i.e. Software Development and Software Engineering Management Phases. The former perspective is consisted of Requirement Engineering, Design, Implementation, Test and Maintenance while the latter includes Initiation and Scope Definition, Software Project Planning, Software Project Enactment/Execution, Review and Evaluation, Closure and Software Engineering Measurements. The term “Human Factor” is used very often in this project report and is indicating different aspects of human being involvement and impact in a software development process. It can vary from individual to organizational, managerial to psychological,

customer to developer and etc. “The human factor in software development is the ingredient that ultimately gives a project team its soul.”[29] Human factors related to each of these phases, human roles and level of focus are analyzed and extracted from the selected research papers resulting in a suggestion for human factor research framework.

This report is a description of the systematic review conducted on software journal papers published during 2000-2010. The structure of the report is as follows:

1. The first chapter gives a brief background to systematic review methodology and related work.
2. Chapter 2 describes the research methodology, research questions and project process steps in details.
3. Chapter 3 includes the results and analysis of the systematic review.
4. Chapter 4 identifies, analyzes and discusses validity threats of the systematic review.
5. Finally the conclusion of the systematic review accompanied with discussions and future work has been brought in Chapter 5.

1.2 Background

In this section a brief overview of the Systematic Literature Review method is accompanied with the project related work.

1.2.1 Systematic Literature Review

“As a research area matures there is often a sharp increase in the number of reports and results made available, and it becomes important to summarize and provide overview.” [6]According to Kitchenham, “Systematic Literature Review (SLR) is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.” [7]

During a typical SLR, the researcher uses primary studies i.e. individual papers published in the area as the source of the review for the related research questions. Systematic review is a form of secondary study inspired from medical related research and guidelines[11,110] which is based on published primary studies and has gained more attention recently by researchers in software development and software engineering related areas.[6,8-10]

Petersen [6] explains that SR reviews the existing primary studies related to the predefined systematic review research questions in depth, describes their methodologies, context and analyzes their results. Among various reasons for conducting a systematic review, the most common reasons that have been mentioned by Kitchenham[7]:

- “To summarize the existing evidence concerning a treatment or technology e.g. to summarize the empirical evidence of the benefits and limitations of a specific agile method.
- To identify any gaps in current research in order to suggest areas for further investigation.

- To provide a framework/background in order to appropriately position new research activities."

As systematic review has a predefined search strategy, which should allow assessment of search completeness, it's been seen to be a fair and scientific secondary study that has good potential of being referred by other researchers conducting primary or secondary studies. When comparing literature review and systematic review, the advantages and disadvantages of systematic review reveal. According to Kitchenham[7,12] the main rationale for conducting a systematic review is the lack of scientific value in most literature reviews of different researches.

As in systematic review, researchers have a predefined methodology; it reduces the bias while conducting the review. Another advantage is the allowance of more general conclusion based on a wider range of contexts and situations. [6,12] However the main disadvantage with systematic review is the considerable amount of effort it requires.[6,7,12]

The main features of a systematic review that differentiate it from a conventional literature review have been summarized as follows:

- A systematic review is conducted under a predefined review protocol specifying the project research questions and methods being used in the review procedure.
- The search strategy for each systematic review is designed to cover and detect as much relevant papers/documents as possible and is documented to ensure the readers about its completeness and reliability.
- Primary study selection in systematic review is based on a carefully/explicitly designed inclusion/exclusion criteria which meets the research questions of the review.
- Systematic review uses extraction forms or other reviewing tools to specify the information needed to be extracted from primary studies and evaluates them via a quality assessment section. [7]

1.2.2 Related work

As mentioned before, according to the increasing number of publications and reports in different areas of software engineering and development there is a need for summarizing and classifying those findings. Secondary studies such as systematic mapping and systematic review have been used to help researchers to have a quick and easy review on a special research topic without the need of searching for all the reports in that area. This has not been the case for Software Engineering (SE) field in spite of a growing need for it.[6]

This can be a result of not having enough information about this method in software related research. Systematic review is a very effective method for giving a structured summary of specific research field in SE such as software development process. One of SE related areas which is lacking researchers' focus is human factors/issues influencing software development

process. Based on the research review that was carried out in this systematic review, there are several categories that are concerned with human factors in the development process.
To the best of our knowledge, according to the search that has been carried out for this master thesis report and its research questions there have been only 5 papers [5,6,73,74,87] in Software Engineering that clearly have used or discussed systematic review and mapping research methodology in one way or another. There are several systematic reviews in different areas of software engineering but not really related to the human factors and this project research questions. [111, 114-124].
Peterson et al. [6] has discussed the use of systematic mapping, review and their procedure in software engineering related areas. While Michael Unterkalmsteiner et al. [5] have addressed evaluation and measurement of Software Process Improvement by reviewing 148 papers published between 1991 and 2008, Magne Jørgensen and Martin Shepperd [73] carried out a systematic review on Cost Estimation by reviewing 304 papers published in 76 journals. Tore Dybå et al. [87] systematically identified and analyzed all the existing studies on pair programming. Tracy Hall, et al[74] reviewed studies of motivation in software engineering published between 1980 and 2006. One can ask why this systematic review included more papers than this current one. The answer is that this paper reviewed 92 studies of motivation in software engineering that were published in the literature between 1980 and 2006. Moreover the databases used for this review are different from the current review (including ACM Digital library, EI Compendex, Google Scholar, IEEE Explore, Journal of Systems and Software, etc). Additionally this review has different inclusion and exclusion criteria such as including conference proceedings, or conference experience report).
All of these systematic reviews focus on a specific category of software engineering human factors such as Cost Estimation (bias, human judgment and anchoring)[73], investigation of software engineers' motivations[74], evaluation and measurements of Software Process Improvements [5] and human factors in pair programming(PP) such as communication and group work [87]. This reveals a need for a general systematic review on human factors in software engineering and development which is the idea behind this master thesis. By general systematic review, we mean a systematic review on human factors in development process in general and not digging deep in any specific phase or category.

To cover most human factors related research papers in software engineering, two main perspectives have been defined for this systematic review which are software development process(phases) and software development/engineering management. The SLR investigates the human role, level of human factor focus and research methodologies for each paper. More explanation about this has been brought in chapter 2 and chapter 3.

1.3 Purpose

The goal of this project is to clarify and characterize human issues and challenges within software engineering and development industry by offering a better understanding provided by

an extensive review over various related journal papers published within last ten years. To summarize, this thesis aims to find the answers to the following research questions:

1. What areas of human factors in software development have been addressed?
2. Which phases of software development have been addressed in SE human factors research? What are the most and the least studied phases and what is been missing?
3. Which software management phases have been addressed in SE human factors research? What are the most and the least studied phases and what is been missing?
4. What kind of research methods/tools (empirical, case study, survey, etc) have been used in this research area? What types of papers are published in the area of research and what was their approach to the topic?
5. What is the human role in these papers? Developer/manager/user, customer

2 RESEARCH METHODOLOGY

In this chapter the design and the procedure of Systematic Literature Review (SLR) of this project has been described. The main focus is on research question definition, selection procedure and data extraction phases.

2.1 Systematic Literature Review

Figure 1 is a schematic view of the SLR process and its detailed steps. This method has been used to conduct the current SLR in Software Engineering.

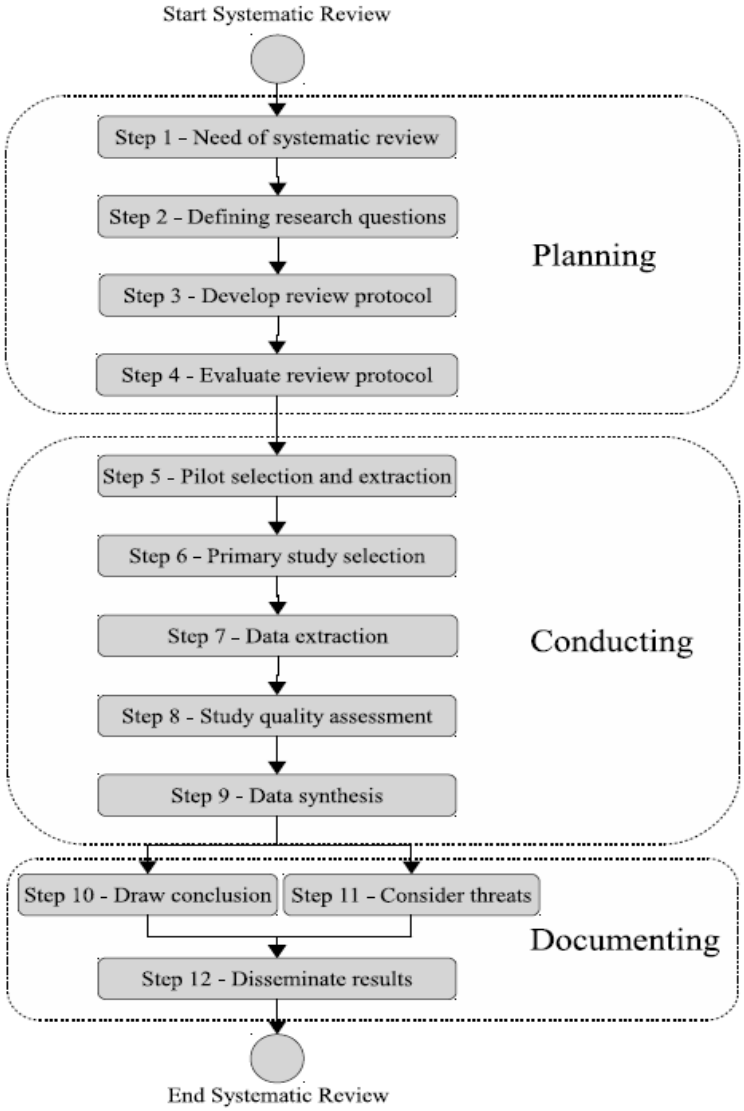

Figure 1, Systematic review steps,[5]

This methodology in general includes three main phases, planning, conducting and documenting. The first step (the need for systematic review) has already been discussed in introduction chapter of this report. To find similar SLR studies in software engineering and development research fields, special combinations of keywords were generated and searched through well known databases such as Scopus, ACM and IEEEXplore. The main keywords were Software Development Process or Software Engineering and Human Factors and Systematic Review (Table 3). To reduce the bias of the search, synonyms of Systematic Review (identified by Biolchini et. al. [13]) were added to the keywords (i.e. research review, research synthesis, research integration, etc). Thus, the final search string for similar studies was generated as follows:

(("Software Development Process" OR "Software Engineering") *AND ("Human Factors" OR "Human Issues") AND* ("systematic review" *OR* "research review" *OR* "research synthesis" *OR* "research integration" *OR* "systematic overview" *OR* "systematic research synthesis" *OR* "integrative research review" *OR* "integrative review"))

By searching this string in mentioned digital databases within the titles, abstracts and keywords (depending on the database options) none of the retrieved results were related to any of the research questions defined for this systematic review project.(ACM Digital Library and Compendex had no results, IEEEXplore had four results [103,104,105,106] and only one of them was related to the research questions of this systematic review [103] (on risk management in distributed projects)). There were two papers retrieved from ACM Guide that were related to agile software development [107] and software requirement errors [108]. But as mentioned earlier this data base was not considered in the study.

The second step was to design the research questions for which the systematic review should be carried out upon. Research questions in this systematic review are related to human factors involved in and influencing software development process or software engineering. Different perspectives have been considered for software development human factors which are addressed separately by each research question. These research questions and their aims have been brought in Table 1.

Table 1, Research questions for the Systematic Literature Review

Questions
RQ1: Which phases of software development have been addressed in SE human factors research? What are the most and the least studied phases and what is been missing?
RQ2: What are the project management/management issues that have been studied in software development process research? Which management issue is mostly focused in this area and which has been ignored? What is been missing and needs more research?
RQ3: What kind of research methods/tools (empirical, case study, survey, etc) have been used in this research area? What types of papers are published in the area of research and what was their approach to the topic?
RQ4. What is the human role in these papers? Developer/manager/user, customer

These research questions are later used for constructing the data extraction form to be used while carrying out the full text reading phase. For instance RQ1 investigates human factors from software development lifecycle perspective while RQ2 focuses more on managerial aspect of human factors in software development process. Thus when creating the extraction form, different categories were assigned to each research question based on the SWEBOK [125].

To reduce the bias of the search and systematic review there is a need for a specific review protocol (step 3) which also enables researchers to replicate the review for future use. The review protocol of this SLR has been checked and evaluated by a researcher in the area of software engineering and human factors with a good experience of conducting systematic reviews. Moreover, parts of this protocol has been previously used and established by several researchers in the area working on SPI systematic review [5] Specific small modifications have been applied after the pilot study phase to improve the protocol. This protocol and its further details have been brought in following sections of this chapter.

2.2 Search Strategy

To find the answers to this systematic review research questions and to conduct the review with a higher quality, the author decided to cover the papers from three well known digital databases which are Compendex (Scoupus), IEEEXplore and ACM Digital Library published within the period of 2000 to 2010. This was carried out to give a snapshot of current state of Human Factors in Software Engineering. The results of the search have has been brought in Table 2. The search only included journal papers and filtered books, early access and conference papers to give a snapshot of the current state of this research in SE.

Table 2, Source databases

Database	Retrieval date	Number of retrieved papers
IEEEXplore	April 2010	118
ACM Digital Library	April 2010	64
Compendex (Scopus)	April 2010	104

To increase the validity of the search and cover as much papers in the area as possible additional search was carried on Google which included papers from well known digital including ACM Guide, IEEE Computer Society. The search from Google also led to additional 115 papers which after filtering on publication year (remaining 61 papers), article type (remaining 31 journals) and removing Human Computer Interaction (remaining 30 papers) resulted in overall number of 30 papers which used for data extraction. Among these 30 papers one paper [94] was not published and excluded from the primary data base.

The overall view of the search strategy has been illustrated in Figure 2.

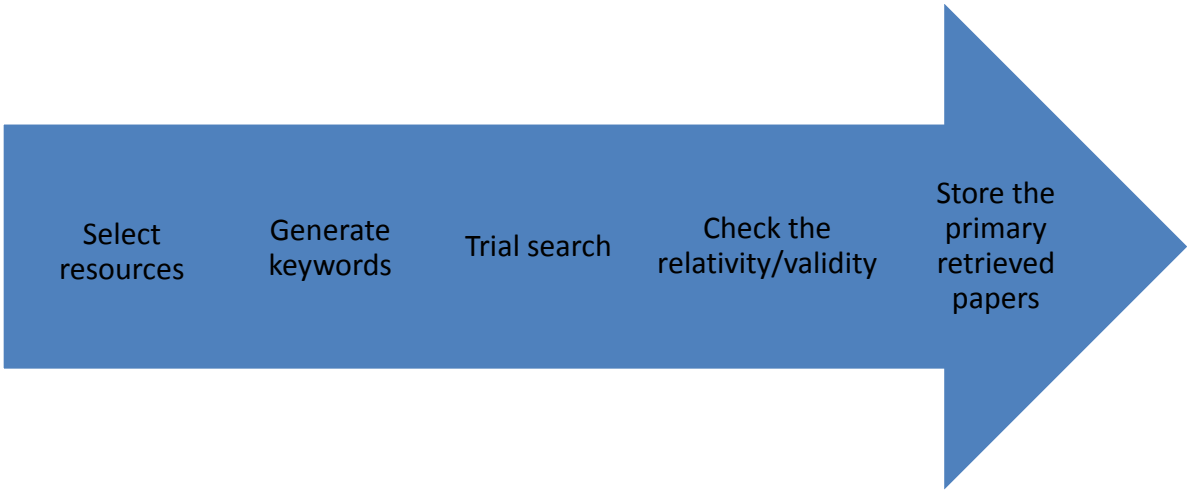

Figure 2, Search strategy

The keywords and search strings were carefully and specifically generated based on the research questions to reduce the bias and retrieve as many papers as possible. Table 3 includes the keywords that were used for the search. As the main focus of this systematic review is to study the human factors in software development process, the search strings include “human factors” or “human issues” and “software development process” or “software development” or “software engineering”. To cover the areas for each specific research question, related interventions were added. For instance to cover RQ1 (software development lifecycle) each of the development phases keywords e.g. requirement engineering, design, implementation, etc have been added to the search.

Table 3, Keywords and Search Strings

Main Body of Search Strings:	"Software Development Process" OR "Software Development" OR "Software Engineering" AND "Human factors" OR "Human issues"
Added Interventions	***Details of Interventions***
Lifecycle phases:	• "Requirement engineering" OR "Requirement elicitation" OR "Requirement analysis" OR "Requirement" • "Design", "Design phase", "architectural design", • "Implementation", "Implement" • "Maintenance", "Maintain" • "Validation and Verification", "Test", "Validation", "Verification"
Project management:	• Planning and feasibility study: o "Cost estimation" o "Scheduling", "Schedule" o "Risk study", "Risk" o "Activity planning", "Configuration management" o "Resource allocation", "Resource management", "Resource" • "People management", "People issues" • "Risk management", "Risk analysis"
Research methods/tools:	"Empirical", "Theory", "Theoretical", "Case study", "Survey", "Experiment", "Interview", "Questionnaire"

For trial, number of searches were carried out in IEEEXplore with the generated keywords and the first 5 papers for each keyword were checked by their abstracts to investigate if they are really related or not and the result was really convincing. The results of initial retrieved papers

by search with the generated keywords for each database were stored in excel forms and their amount were earlier brought in Table 2.

2.3 Study selection criteria

After finding the primary set of papers for the review, the abstract reading phase was conducted to exclude the papers not related to the research questions. For instance, there were many papers in the area of cost estimation and scheduling which were more related to the technical aspect and cost estimation model/methods and were concurrently excluded. Publications that were not full papers (2-3 pages), were not journal papers, were not English, and were not within the related area (software development, software engineering, cognitive issues, human factors, etc) were excluded from the databases.
Furthermore another exclusion criterion was added during the abstract reading phase which was to exclude papers related to Human Computer Interaction, Usability Design, Software Interface, Interface Design, etc that were more retrieved in design phase search. The main reason behind this exclusion criterion is the need of more focus on development process aspect of human factors instead of only focusing on the interaction process aspect.

2.4 Selection Procedure

To focus more on the research questions and the goal of the project, the whole data base of papers was reviewed for the second time by abstracts and keywords. The remaining categories, Process, Product, Organization and Management were chosen based on the main objective of this project and extracted using abstracts. The goal of the project was set to identify the human factors in software development process.
After reading the abstracts of the mentioned 90 papers they were put in a data base consisted of 4 columns representing each category and the rows including the papers titles. There were some overlaps among the categories (as some papers were analyzing development process from several points of view each in different category) but as the most relevant categories for this project were Management and Organization thus the papers related to only Process or Product (not having any overlaps with Management and/or Organization) were excluded. The reason is that this project aims on finding human related issues in development process from organizational and managerial perspective. Consequently papers that discuss only about human factors related to software products or other types of process are excluded. Number of papers in each category has been summarized in Table 3. These numbers are only based on the abstract reading phase, which were later revised after the full text reading phase.

Table 4, Categories of the Final Database

Categories	**Number of Papers**
Process	57
Product	1
Management	40
Organization	65

As it can be interpreted from Table 3 the main focus of the included papers has been on Management and the least focus on Product. More detailed information on the overlaps of these categories can be found in Table 5.

Table 5, Categories Overlaps

Overlapping Categories	**Number of Papers**
Process and Organization	21
Process and Management	31
Product and Process and Organization and Management (1 paper in Product category overlapping with all other categories)	1
Organization and Management	28

As it can be observed from Table 5, the main overlap has been among process and management and secondly among organization and management.

The final outcome of the overall extraction procedure led to the primary database of papers to be used for the data extraction phase and full text reading. This database included to the total number of 67 papers in the area of Management and Organization with overlaps with other two categories.

An overall schematic view of the whole selection procedure including the first search results, abstract reading, exclusion/inclusion criteria, etc has been illustrated in following figure (Figure 3).

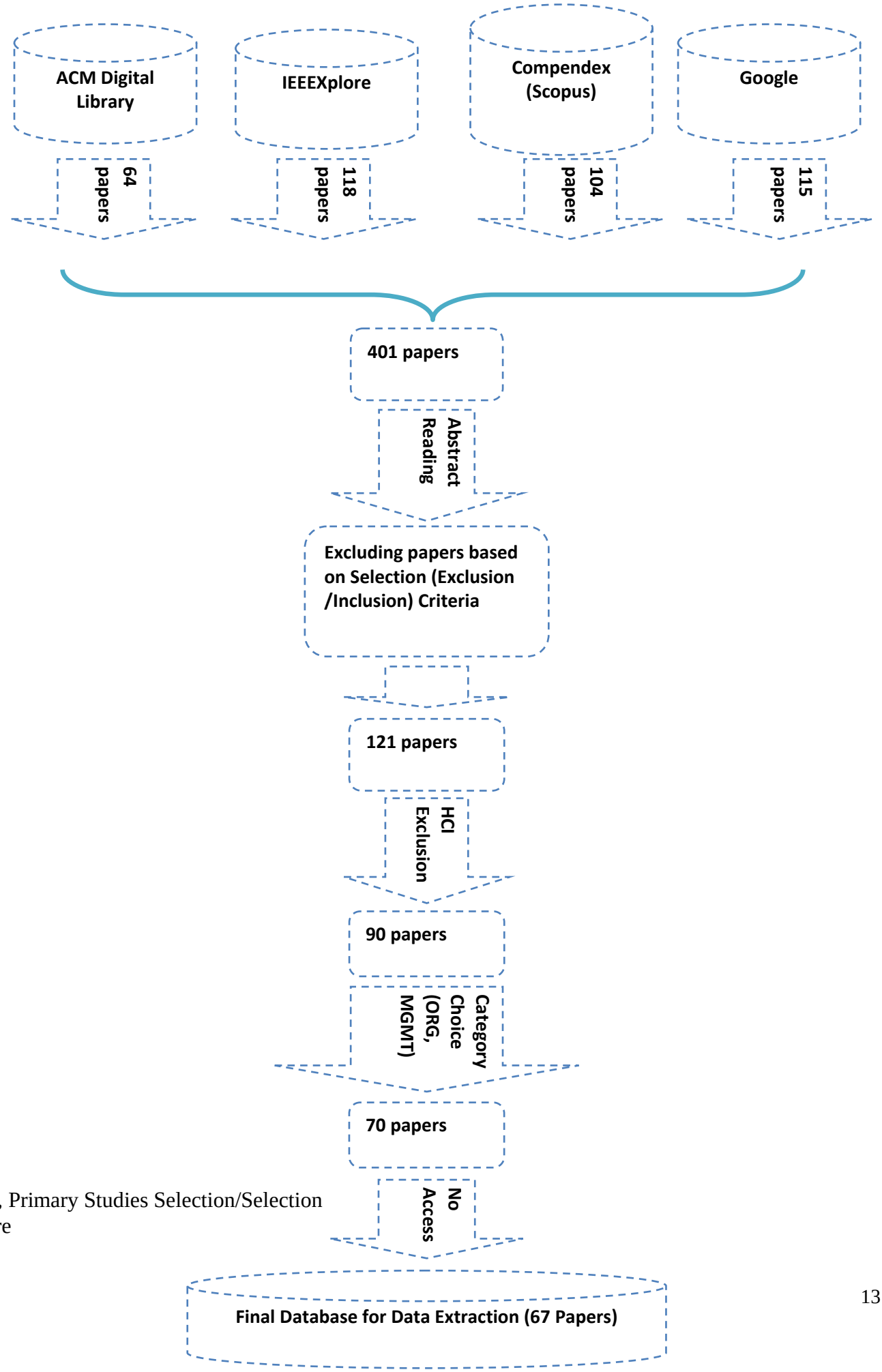

Figure 3, Primary Studies Selection/Selection procedure

2.5 Data Extraction

In this phase the remaining papers left after various/several filtering were collected for full text reading and data extraction (Step 7 of systematic review (Figure 1)). To enable the author to carry out this phase more efficiently, a data extraction form was designed based on the research questions of this project. In order to evaluate the form, a pilot review was carried out. 10 papers were selected randomly by an online random number generator (http://www.random.org) and the full text reading was carried out for each. Based on this, the form was modified to include more specific factors and become easier to use for the real data extraction. (Appendix A)
The first version of the extraction form included two main perspectives of this project, Software Development Lifecycle and Project Management. As Staples and Niazi [14] have mentioned in their studies, it was difficult to set values for all properties prior to the extraction phase as they were completely dependent to the papers and their contents. Hence, some values in the extraction form were expected to be detected during the extraction phase. These properties and their related research questions which have been considered in the first version of the form are brought in the following table. (Table 6)

Table 6, Extracted Properties

ID	**Property**	**Research question**
P1	Software Development Lifecycle	RQ1
P2	Project Management	RQ2
P3	Context	Overview of the studies
P4	Research method	RQ3
P5	Human Role	RQ4
P6	Level of Human Factor Focus	Overview of studies
P7	Publication year	Overview of studies

P1, P2, P3 and P4 were assigned with set of values based on the information gathered in abstract reading phase. P1 was assigned to the set of predefined values based on the Waterfall Model.[15,16] for software development lifecycle steps including requirement engineering, design, implementation, etc. For P2, assigned values included Cost Estimation, Planning, Risk management, People management, etc. As mentioned before, 4 major categories were defined for the papers collected (Process, Product, Organization, and Management) and P3 was assigned to these values. Forth property (research method) was to investigate the research methods and tools that have been commonly used in this research field and to highlight those which are not used but can be helpful for future research and studies in this area. For P4 the values were generally Empirical and Theory. Empirical category included Case study, survey, Industrial report, Grounded theory and ethnography. P5 clarified the human role discussed in the papers to identify

researchers' perspectives to human involved in the development process. The set of values assigned for this property was: Customer, Developer and Manager.P6 investigates the level of human factor focus for each paper to identify whether it's Individual, Interpersonal or Organizational. P7 indicates the publication year of the paper.
Based on the results of the full text reading pilot and, properties and their values were modified. For instance the research methodology property (P4) was modified to include more research methods that are common in software engineering papers which were included in full text reading pilot. These methods were extracted from several books on the area of experimentation in software engineering and relative systematic reviews [17,5] Other values which were modified among properties were related to Management (P2) and Development Phases (P1). These modifications were applied based on Software Engineering Body of Knowledge (SWEBOK) (Appendix B) [125] to include more relevant and reliable categories. These modifications resulted in an extraction form in Appendix A that was used for the full text reading phase. Categories in this form have been used for data synthesis (Step 9) and drawing conclusions (Step 10). More explanation of the properties/categories, and their values have been brought in Sections 2.5.1 to 2.5.7.

2.5.1 Software Development Lifecycle phases (P1)

The first property investigates software development phases. This property highlights which software development phases have not been studied from human factors perspective and generates some new research questions for future studies in the area.
The phases that should be investigated in this property are based on the waterfall model and SWEBOK [125] are:

- Software Requirement
- Software Design
- Software Implementation/Construction
- Software Testing (V&V)
- Software Maintenance

Very brief and general description of each phase and the related information that has been expected to be extracted follow in section 2.5.1.1 to 2.5.1.5.

2.5.1.1 Software Requirements

The earliest phase of the software development process is requirement engineering. "The systems services, constraints, and goals are established by consultation with systems users. They are then defined in details and serve as a system specification" [16] Researchers have studied different types of requirement engineering for various development processes and the effect of the process type on the requirement engineering methods. "Requirements engineering (RE)… is

a traditional software engineering process with the goal to identify, analyze, document and validate requirements for the system to be developed. [18]
As researchers have identified, RE process mainly consists of five major activities [19] which are Elicitation, Analysis and Negotiation, Documentation, Validation, and Management. However, according to [16] requirement engineering is consisted of four main activities: Feasibility study, Requirement elicitation and analysis, Requirement specification and requirement validation. Feasibility study "… considers whether the proposed system will be cost-effective from a business point of view and if it can be developed within budgetary constraints." [16] In Requirement Elicitation developers focus on system observation and negotiation with potential users, producers while in Requirement Specification they translate the information gathered to documents defining the requirements. The last part of requirement engineering is the validation phase where the developers check the requirements for its consistency, completeness and realism. [16]
These phases of RE, have been the focus of many studies and researchers have compared them in different types of development processes. For instance, there are lot of publications studying the differences of the methods and tools used in each RE activity for traditional and agile software development. [18] Furthermore there has been some studies on different aspects of requirement engineering including: strategies, heterogeneous engineering, political ecology [109], high level requirement engineering, geographic distribution of software development(multi-site development)[57][102] and its effects on RE(multi languages, cultures [57], stakeholders, time zone boundaries[57,102], RE practice in global software development.[57,70,102] But as mentioned earlier this project follows the SWEBOK[125] which has divided RE into two main categories (analysis and validation) and their related subcategories, almost all covered by the different definitions and categories explained above.

2.5.1.2 Design

The system design process allocates the requirements to either hardware or software systems by establishing an overall system architecture. Software design involves identifying and describing the fundamental software system abstraction and their relationships. [16] Moreover Pierson mentions "A software design is a description of the structure of the software to be implemented, the data models and structures used by the system, the interface between system components and, sometimes the algorithms used." [16] SWEBOK[125] considers three main categories for design phase that have been indicated in the extraction form which are Architectural styles, Design patterns and Software design methods.
Architectural style is more related to identifying "… the overall structure of the systems, the principal components (sometimes called sub-systems or modules), their relationships, and how they are distributed." [16] According to SWEBOK (Appendix B) this category describes the general structure of the software, and classifies them to Distributed, Interactive and Adaptable systems. Design pattern categorizes the design pattern of the software to Creational, Structural and Behavioral patterns. And finally design methods describe the methods behind the design of

the software that have been defined to be Function-oriented, Object-oriented, Data-structure-centered and Component-based. Most of the focus of the research in the design phase is technical and related to Interface Design, and researchers have not paid enough attention to other human aspects of this phase. The design related papers (few) were mostly about requirement based designs, design patterns and architectural design.

2.5.1.3 Implementation

The most technical phase of the software development is the Implementation phase. In this phase "the software design is realized as a set of programs or program units…" [16] This phase addresses human related issues connected to the technical aspect of the Implementation phase. Some examples are: programming language, data bases, functions, coding, pair programming, personality, productivity, dependability, reuse, end-user programming and methods. The SWEBOK [125] considers three main subcategories for Implementation (Constructing for verification, Coding and Reuse), that have been used in extraction form. The first category is more related to the early concerns of the implementation such as programming language choice, platforms and tools. Coding is the main part of the implementation phase where developers are concerned with linguistics and visual issues. The last but not the least phase of implementation is Reuse where developers conduct unit and integration testing in order to use the software by integrating it to their system.

2.5.1.4 Test

Test phase is also another Software Development phase which is related to human factors. Testing phase has different levels such as unit testing, integration testing and system testing. [16] The first level of testing is unit testing where the developer/tester checks each unit regarding specification. The second level of testing (integration testing) happens when integrating individual programs and units to create a complete system to check if software requirements have been met. Afterwards the software will be delivered to the user/customer. Also testing process has been defined to include three main stages, Development testing, System Testing and Acceptance Testing. [16] In Development Testing, developers test the system components while system testing is carried out when integrating components to create the system to find errors that are the results of unanticipated interactions between components and component interface problems. Finally Acceptance testing is conducted to check the system with the data supplied by customer to reveal the system engineering errors and requirement problems. [16] Based on the SWEBOK (Appendix B) [125], Test levels, Objectives of testing and Techniques based on the nature of the application are the higher level categories related to test phase. These phases have been covered by different test definitions explained here. Test levels categorizes different levels for testing (unite testing, integration testing and system testing) as already explained. Objectives category includes subcategories that explain the aim of the test process such as performance, usability, recovery and acceptance testing. In each developers focus on specific goal and test the

software against that. The last category includes different techniques for testing software such as object-oriented testing or GUI testing.

2.5.1.5 Maintenance

The last phase of software development Lifecycle is Maintenance which has close relations with human factors and has been studied from different aspects. Operation and Maintenance is called to be the longest development phase. “Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered.” [16]

One of the most common topics within this area that have been of focus for many researchers is Reengineering, its consequences, challenges, and related solutions. Modification implementation and process implementation are some other examples of research papers in this area. According to the SWEBOK [125], Maintenance covers three main categories (included in the extraction form), Categories of Maintenance, Process and Techniques for Maintenance. (Appendix A) The first category includes different types of maintenance such as corrective and adaptive maintenance while the second category (Process) characterizes different types of process related to the maintenance phase such as process implementation and modification implementation. The last category includes different techniques used in maintenance activity such as reengineering and reverse engineering.

As mentioned earlier it was decided to use the SWEBOK [125] for more details on each phase. Each subcategory of the SWEBOK in development lifecycle in addition has subcategories indicating more details of the phase. For those papers that these details could be extracted, the related filled form indicates the details. However, as this project is more related to the human issues of the development they are generally related to this phase. The Software development Lifecycle related tables (5 tables) of the SWEBOK have been brought respectively in appendix B. This part of the form is closely related to the people management and issues of the Project Management category as most researchers have discussed people communication, collaboration, team work issues, etc which will be discusses more in the analysis chapter of this report.

2.5.2 Project Management (P2)

Project management property is for clarifying management issues that have been discussed in software development. This helps readers to see if a specific management issue is discussed only in certain software development phases. Project management issues that have been considered when reading the papers are adapted from [20, 21]

- Cost estimation
- Scheduling
- Resource allocation/management

- Activity planning/ Configuration management
- People management
- Risk analysis and management
- Possibly steering and management challenges regarding change management, management skills and etc

The first three subcategories are very similar and it might be hard to separate them. When this was not possible all relevant subcategories were marked. For CE for instance the paper may talk about problems in scheduling for budget or time but doesn't include a specific CE method or explicitly specify CE terms. In these cases paper has been put in all three categories it concerns. These subcategories are drown from two main software engineering related books [20,21] to cover Management Issues as specifically as possible. But to follow the SWEBOK [125] in this perspective these were combined with the general Software Engineering Management table of the SWEBOK (Appendix c). This table considers 6 major subcategories for Management. The first two subcategories (Initiation and Scope definition, Software Project Planning) are related to the first 4 categories defined by [20,21] (CE, Scheduling, Resource Allocation/Management and Activity Planning/Configuration Management). As there is one subcategory in planning phase which has not been addressed by [20,21] this subcategory was added to the top of the list that is Feasibility study, risk study and RE. The rest of the categories mentioned above were assigned under each general SWEBOK subcategory. More details of these categories and their related subcategories can be found in the extraction form in Appendix A.
Papers in this category are also related to the soft side of the development process and mostly do not consider very technical development issues such as specific details of CE methods, models, mathematical computations, etc. By following in section 2.5.2.1 to 2.5.2.6 more explanation about each category has been brought.

2.5.2.1 Initiation and Scope Definition & Software Project Planning

This category includes the planning activities related to software development process/software engineering. These activities are mostly carried out very early in the project, consequently to a great concern dependent to human factors. These activities are:

- Feasibility study, risk study, requirement study
- CE
- Scheduling
- Resource allocation
- Activity planning/configuration management

Risk study, investigates the risk of carrying out the development project regarding the budget, time and resource limitations. Feasibility study is related to the first phase of software

specification "… where an estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies". [16] Requirement engineering and study is the activity of identifying the requirements of the project from different aspect such as time, resource and monitory perspectives.
Cost Estimation is the phase where the developers and project managers predict the cost of conducting the project. This cost is calculated according to the effort required to perform the project and the economical cost it generates.
Scheduling is a management phase which is more concerned with time limitations (deadlines) and source that has been set for the project and the efficient way of meeting those.
Resource allocation is a phase where managers and developers set the resource limits for the project in terms of money, time, people, and other types of resources that a software project needs.
Activity planning is more related to the role and activity assignment and scheduling (priorities, time schedules, etc). Most of the subcategories in this category are affected by human factors such as overestimation, human judgment, overconfidence, human errors and lack of experiences.

2.5.2.2 Software Project Enactment (Execution)

This category is related to the management issues that developers and managers are dealing with during the project execution phase. The first category is people management which is a very broad subcategory covering different aspects varying from psychological, such as communication issues, personality types, and mentality patterns to programming skills, performance, Personal Software Process improvement, work environment effect on human productivity, customer satisfaction and etc. The main subcategories of this Software Project Enactment management phase are:

- People management
- Risk analysis/management
- People issues / Human factors

People management category covers management issues related to people involved in the project during the enactment/execution. This category discusses different people management issues such as team management and group work, developers' communication in groups, multi culture group/project management, training staff and PSP, developers' personalities and it's relationship with role assignment. Moreover management issues closely combined with psychological issues are another aspect of this category. For instance resistance of people (developers and staff) to a new method and the management challenges to attract people to use those are considered in this category.
Risk analysis/management is more concerned with study of risk, and managing them during the project execution. This category is defined to cover papers that discuss the risk studies, analysis and management during the software development process. Failure risk management, configuration risk management, user participation related risk, scheduling and CE risk and risk

assessment training are some examples of risk analysis/management related topics. However risk study is also carried out in the first phase of software engineering management process with less information.

The last subcategory (people issues) has been added to include the human factors that are not among the mentioned ones. Cognitive issues such as learning styles, human information processing, client's confidence and decision making related psychological issues are some examples of the topics in this category.

2.5.2.3 Review and evaluation

This category has been added from the SWEBOK [125]. Mainly this category covers late management issues in development process and software engineering (time aspect) i.e. feedback management. This phase has impact on development process improvement. In other words this phase is conducted to identify any problems regarding the execution phase and to take corrective actions in an effective way. The review and evaluation of the software in this category is not related to the technical issues but mostly studies customers'/developers'/designers'/stakeholders' feedback in the development process. This category only has one subcategory, feedback, which leads to a more efficient and successful process by reviewing the feedback from participants and evaluating it. Feedback among different project phases also helps to improve the process by implementing corrective or preventive actions. [112] Feedback is also a customer centered process where developers try to meet the customer feedback and comments and gain their satisfaction. Subsequently, customers' and managers' feedback and satisfaction are the key points of evaluation and feedback phase. Feedback also affects the Customer Relationship Management among the developers (software industry) and customers.

2.5.2.4 Closure

This category is also related to the late management issue (from time aspect) in the development process; in this case is closing of the project after evaluating the feedback. Closure process is mainly the official acceptance of the project (by customers/managers) which is companied by the administrative activities such as documenting and archiving files. Software document design, management issues regarding the final management phases of the development process when closing a software development project are some examples of the topics in this category. Closure/closing phase also indicates two main steps in the final management phases, Project Closure and Contract Closure. Project Closure is when developers finalize all the project activities and close the project by archiving files and documents while Contract Closure is closing and completing each contract related to the project or a project phase. [112]

2.5.2.5 Software engineering measurement

Software engineering measurement is a phase in project management that highlights factors leading to project success or failure in qualitative and/or quantitative way. Measuring the project activities, monitoring the project variables (cost, time, etc) against the project management plan identifies the current state of the project. Some of the factors that this category covers are performance (group and individual), productivity, usability, success, failure, etc. As the nature of measurement in engineering fields is mostly quantitative, studies in this area are expected to be more quantitative based and have more statistical information.

Based on the pilot data extraction there are additional management challenges for developers and managers which are not addressed in the subcategories mentioned above. Thus this subcategory has been extended in the form (Appendix A) and is more related to the human role of managers. This subcategory is called “management challenges” and includes change management, management skills, etc.

2.5.3 Context (P3)

As mentioned earlier in the selection procedure, papers in the area of Software Development Human factors in this project have been categorized into four major groups: process, product, organization and management. As one of the goals of this project was to identify the type of human factors that have been studied in Software Engineering, these categories were defined. Process category is defined to cover human factors papers related to specific or general development process phase. In other words, this category focuses on the process aspect of the development and the human factors affecting it. This group has more technical details than the other groups of this property.

Product category is defined to include papers which focus on software products and do not consider other aspects and details of the development process.

Organization group is more related to organizational issues affecting the development process such as structure, policy and environment. Some topics related to this category are work place, regulations, communications among managers and developers or within a development team, multi-site development, multi stakeholder’s project and etc.

Finally the management group should include studies which focus on the management aspect of the development process such as risk management, people management, change management etc. In this category detailed management challenges regarding the staff, organization, project, risk, change and failure, employment (role assignment), etc should be discussed. This category is closely related to the Software Project Management category.

2.5.4 Research methods and tools (P4)

In this category papers have been categorized based on the research methods their authors have used when conducting the research projects. The papers in this category are divided in two types: empirical and theoretical. Empirical research methodology has several subcategories and the

ones that have been most commonly used in software experimentation have been considered in the extraction form. These methods are derived from a combination of methods adapted from three software engineering experimentation related references [5,17, 5] All the categories and their related subcategories that have been used in this systematic review extraction form are as follows:

1) Empirical

    a) Case study
        i) Industry case study
        ii) Non-industry case study
    b) Survey (interview, questionnaire, observation, systematic review?)
    c) Controlled experiment
    d) Industrial report
    e) Design of system/solution/model/method

2) Theoretical

The strategy for classifying the papers in these subcategories was to put the paper in the category that has been explicitly mentioned in the papers or it could be extracted from the paper. However, this is expected to be challenging in some cases. For instance, when a paper has not mentioned its methodology or it cannot be interpreted from the paper, it is very hard to distinguish which category it belongs to. This property of the papers can help to assess the quality of papers as well because there are many papers that do not mention their method explicitly in the report which is a quality issue for a paper being published in a journal. Also to reduce the error rate when deciding which category the papers belong to, categories and their criteria (accurate definition and criteria) have been set in advance. Some of these categories and their criteria have been adapted from [5] which is a systematic review in a similar area. In sections 2.5.4.1 and 2.5.4.2 each category and their related criteria has been explained.

2.5.4.1 Empirical

This category includes papers which have been a result of an empirical study such as case study, survey, controlled experiment within the industry or academia. “Lessons learned” also is considered as empirical research which is mostly categorized under case study and design methods.

Case study (Industry, Non-Industry): Case study is defined to be used when one wants to investigate a single entity or phenomenon in a specific period of time. [17] Another interesting definition/explanation for case study is offered by [23] explaining the central tendency of different types of case studies which is identified to be an endeavor to illuminate a decision or a set of decisions: why they were taken, how they were implemented, and with what result” [22]

The paper has been assigned this category if it meets one of the following criteria:

1) The author as addressed research question by conducting a case study in industry or academia [24]
2) The author has evaluated a theoretical concept via conducting a case study by which the results indicate the validity of a theory in an experimental environment and meeting a predefined research goal. [25]
3) It has been explicitly mentioned in a paper that its research method is case study, industrial experiment, academic case study or similar.

The industry studies are carried out in (collaboration with) industry and the rest are non-industry studies (in academy or not stated)

Survey: survey is a research method where the researcher collects quantitative and/or qualitative data by means of interviews and/or questionnaires. [5,17,24,26,27] conducting a survey using interview as its data collection tool has some advantages over the questionnaire based surveys such as higher response achievement, more precise responses (fewer number of responses such as "I don't know") and possibility of observing for interviewer.[17] " A survey is often an investigation performed in retrospect, when, a tool or technique, has been in use for a while." [17,28] Papers that will be assigned to this category might use different tools for their surveys. Even in some cases the survey is accompanied with another type of research method such as case study or controlled experiment. For instance it's possible to use a controlled experiment to collect information for a specific survey. Survey could be used to give a snapshot of the current status situation. [17] Surveys do not offer a better understanding for a specific sample; instead they help in understanding the population which the sample has been drawn. [113]

Controlled experiment: if the author has conducted an experiment with a predefined design and has control over the experiment (has set the factors of the experiment and wants to check the results) it should be categorized under this research method. "Experiments are normally done in a laboratory environment, which provides a high level of control. When experimenting, subjects are assigned to different treatments at random. The objective is to manipulate one or more variables and control all other variables at fixed levels" [17] One example could be dividing people in different groups and doing some experimental investigations by for instance teaching one group and not teaching the other and then asking both groups to carry out a similar task. Also if it has been explicitly mentioned in the paper that it uses controlled experiment, the paper belongs to this category.

Industry report: The paper is an industrial report if it meets one of the following conditions:

1. It has been explicitly mentioned in the papers that this is an industrial report or a report of results/lessons learned/experiences from and industrial experiment/project
2. The paper doesn't mention any research method explicitly but is result of an industrial experiment with no predefined research questions. "Therefore, instead of creating a

category “N/A”(research method is not applicable), we added this category as it complies with the “Project monitoring” method described by Zelkowitz and Wallace”. [25]

Design of system/solution/method/model: A paper belongs to this category if the author has suggested a framework, model, guideline, and method that can be used by other researchers or practitioners in the area to use in their research work. For instance one can give a framework for researchers working with a specific topic in the area which has been examined in several research projects earlier. This category mostly includes papers that have some practical/empirical background or evidence for their suggested system/solution/model or framework. For instance they might conduct a case study to evaluate their suggested model or implement it.

2.5.4.2 Theoretical

If the author has reviewed some theoretical concepts or suggested some theoretical solutions with no empirical support the paper belongs to this category. The difference between this category and Design of system/solution/method/model is the practicality of the suggestions. For instance it’s common that in the latter case authors give some practical examples of their implemented/partially implemented solutions while this is not the case for theoretical studies.

2.5.5 Human Role (P5)

Another aspect of the human factors studies in software development process is to investigate which role human has when researchers study and discuss human factors. This category is closely related to different software development phases, project management and research methods. For instance in Design and RE phase the focus is on user or customer perspective while the implementation phase focuses more on the developer role. Thus it’s good to identify these different perspectives and see how they are related to a specific software phase or management issue. The human roles that have been considered in the extraction form are: Customer/ user, Developer/Programmer/Designer/System analyst and Manager.

Customer role is defined to cover those papers where their focus is on customer, Customer Relationship Management, Customer satisfaction and etc. In this category customer have major roles in different phases of development process beginning from the early phases such as Requirement specification and ending with late phases such as maintenance and customer support.

Developers have a major role in development process and can be referred to with various terms such as programmers, system analyst, testers, designers and etc. In this category developers are the focus of the papers where researchers discuss their cognitive, psychological and managerial issues during the development process and how they affect the process and its performance. For instance training, learning styles and Software Process Improvement are closely connected to this category.

Researchers tend to address different types of managers in development process carrying from CIO and project manager to team manager and project owners.
As human is an important factor in this systematic review, it's of interest to know how these different roles have been addressed by researchers and how researchers see human being involved in the development process. These subcategories can help to see how researchers see human involved in the development process and which is the most focused role.

2.5.6 Level of human factor focus (P6)

Another property is the level of human factors in each paper. SE researchers have published several papers, which can be divided in three subcategories:

- Individual: this subcategory covers individual related human issues to SE/software development.
- Interpersonal: this category is related to human issues among individuals affecting or being affected by SE or software development process.
- Organizational: Organizational subcategory includes human factors and issues related to organizations and working environments.

As mentioned earlier human has different roles in software development process. For instance some papers have studied individual aspect of human factors in software development such as development skills, satisfaction, productivity and SPI while others have discussed organizational factors having impact on the process, its productivity and success. Interpersonal level is more related to the communication, collaboration and cooperation among individuals in a development team and how it affects the overall success or productivity of the development process.
Individual subcategory has a narrowed down focus on individual aspect of human and organizational factors including individual characteristics, personalities, human psychology, cultures, decision making, managers' management skills (individuals), Personal Software Process(PSP), and individual learning and improvement.
Interpersonal subcategory is the domain of research where human factors in software development have been studied from group/pair working pint of view. Some of the main concepts/ issues that are of interest for this project pair programming, cooperation, team work, multicultural environments, group learning and evaluation, productivity, group innovation and efficiency.
Organizational subcategory covers the research related to human factors from organizational point of view such as management and decision making in organizations, staffing, organization environment (multi-culture), industrial issues, etc.
As some researchers consider all these levels in their studies a subcategory called "all" has been added to the form. Some papers of the database do not focus in any level or it cannot be derived from the paper which level of human factor it's addressing. Hence, "unknown" field has been added as well.

2.5.7 Publication Year (P7)

The last property helps to identify the search improvement in recent years as it's a snapshot of the current search on human factors in software development process and SE. This property has a predefined list of values from 2000 to 2010. This period has been chosen to give a sense of the current status of human factors in software engineering research. As mentioned before Journal papers are considered in this systematic review. This property helps the reader to see the improvement and lack of research in different stages of this period. The idea behind this property is to highlight the need of more studies in the area of human factors in SE by showing the research improvement over the recent years. Moreover this property highlights the gap of research in this period.

These properties (P1-P7) should be extracted from the papers and be categorized based on the details of each property.

2.6 Assessment of Searchability

Study Quality Assessment (Step 8) is conducted to evaluate the validity of the retrieved studies in the area. As it has been concluded by Staples and Niazi [14] it is very challenging to determine in what level the validity threats have been addressed by the authors of the papers. However, this is not the only subject of the quality assessment process. Subsequently, quality assessment is modified to another level of assessment in this project which is more based on the author's judgment about the studies based on papers keywords, abstracts, titles and contents (searchability assessment). To do so several questions were designed and added to the end of the extraction form. These questions were designed based on the results of the early readings in this project. For instance there were many papers in this area that did not have the expected keywords and could not be retrieved with the search strings designed for this systematic review. These questions (in the form) were yes/no questions but during the assessment some information were added which were useful for the analysis and conclusion phase. "Partially" column was added for those papers that were in the middle. Following table (Table 7) includes these assessment questions.

Table 7, Quality Assessment Questions and Results

Search Ability Assessment Questions	**Aim**
Q1. Has author keywords similar to human factors in software development/SE, human issues in software development/SE, human centered software development/SE cognitive/psychological/interpersonal/individual/organizational/managerial issues been used?	To check author's keywords and how they match the keywords of this systematic review. To determine the difficulty of search and retrieval of these papers when conducting a secondary study in a relevant area.
Q2. Is the abstract searchable about the topic: human factors/issues/impact in/on software development/ SE?	To evaluate the abstracts of the papers according to topic/area keywords
Q3. Does the abstract include the keywords: Human factors, human aspect, cognitive issues, software psychology, management issues, organizational factors/issues/impact, human factors in software development/SE, human issues in software development/SE, human centered software development/SE, and interpersonal/individual/organizational/managerial issues?	To check whether the abstract has expected/relevant keywords to enhance the other researchers when searching via various databases base on their abstracts.
Q4. Is the title searchable about the topic? : Does it include human factors/issues/impact in/on software development/ SE?	To evaluate the titles of the papers according to topic/area keywords
Q5. Does the title include the keywords: Human factors, human aspect, cognitive issues, software psychology, management issues, organizational	To check whether the topic includes any related

factors/issues/impact, human factors in software development/SE, human issues in software development/SE, human centered software development/SE, and interpersonal/individual/organizational/managerial issues?	keywords or not. This is also useful for other researchers and reviewers as generally their first search is by title, abstract and keywords.

2.7 Validity Threats

Last phase of the systematic review model suggested by Kitchenham [7, 12] is to study validity threats to the conducted systematic review by considering various threat sources and aspects. These threats vary from Investigator Bias, Publication Bias, to Primary Studies related threats and Data Extraction Consistency threats. This phase will be explained in more details in chapter 4.

3 ANALYSIS AND RESULTS

In this chapter the result gained by previously explained systematic review step (data extraction, Step 8) have been analyzed and discussed. First a short overview of the included papers and their general characteristics are discussed. This is followed by the results and analysis related to each research question.

3.1 Overview of the studies

As mentioned in previous chapter, total number of papers included in the data extraction phase was 70. Three papers [49, 59, 67] were excluded as the author couldn't get the access to the full text. The analysis in this chapter is based on the data extracted from 67 remained papers that were fully read.
In this section a general overview of studies related to properties of interest is presented.

3.1.1 Study context (P3)

As previously explained in sections 2.5.3, papers which were not related to organization or management were excluded. After the extraction, out of the remaining 67 papers, 42(62%) were related to process, 6(8%) were related to product, 46 (68%) were related to organization and finally 58 (86%) were related to management. (Figure 4) As it can be identified these figures do not add up to 100% which is a result of high overlaps among these categories. A schematic view of papers context and its categories has been illustrated in Figure 4.

Figure 4, Context and Categories

There are some overlaps among these categories. The pairs and triples of overlaps have been illustrated respectively in Figure 5 and 6. In this study we only look at the two mentioned categories (Organization and Management) and their overlaps with each other and the two excluded categories.

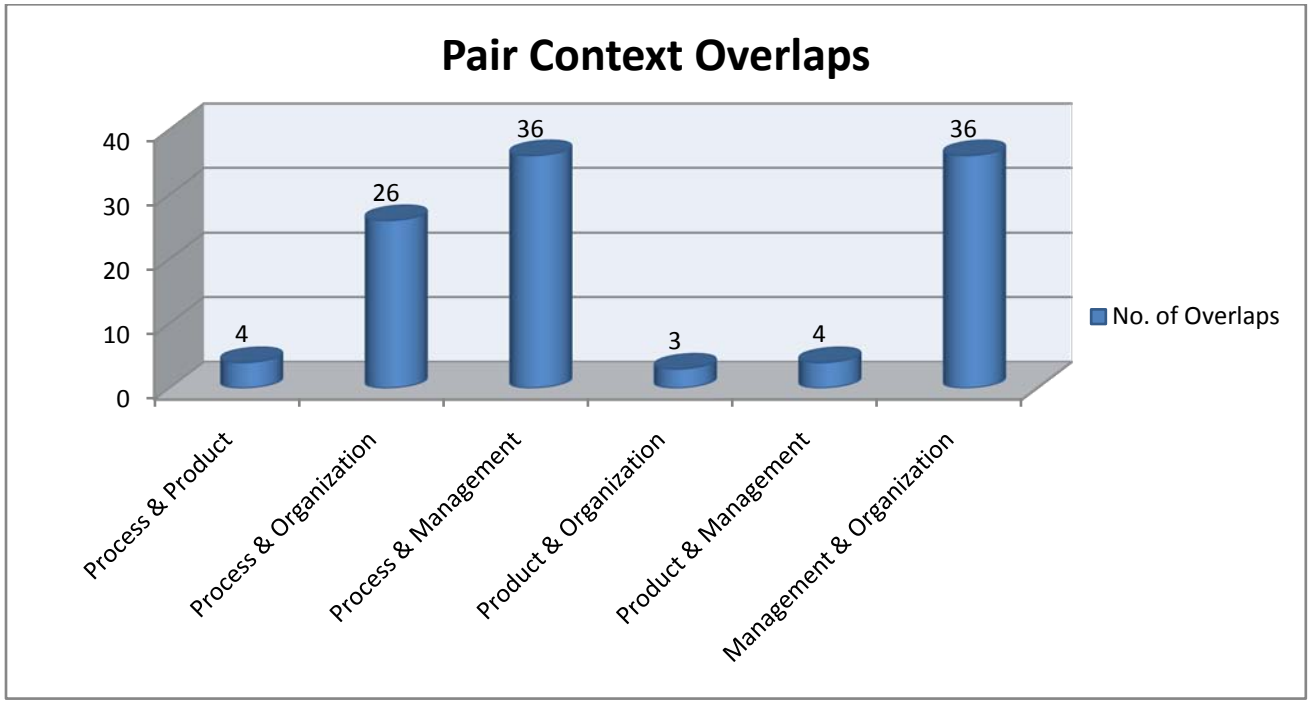

Figure 5, Overlaps among Context Pairs

As it can be observed from Figure 5, the highest overlap is between Process and Management which is similar to the one between Organization and Management. Also the lowest overlap is between Product and Organization. Following figure (Figure 6) illustrates context triples and their overlaps.

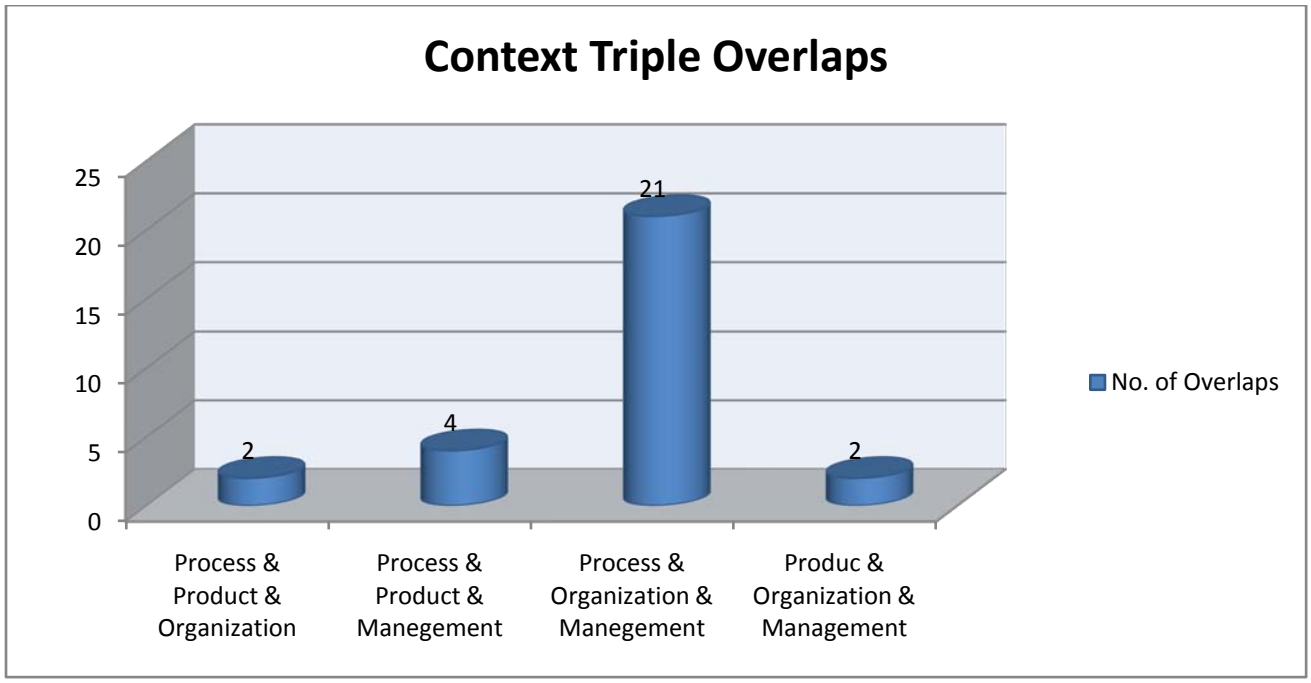

Figure 6, Overlaps among Context Triples

As it can be observed from figure above (Figure 6) the highest triple overlap is among Process, Organization and Management (20 papers) which is the focus of this project. Overlap among all of the categories is very low (2 papers). These two papers [43, 62] discuss more general terms of all categories and do not go into any details.

3.1.2 Level of Human Factor Focus (P6)

As it was explained in section 2.5.6 another interesting property is the level of human factors in each paper. This property was assigned to three predefined values: Individual, Interpersonal and Organizational. Based on the data extracted from 67 papers in the data extraction phase, the focus of most studies has been on individual aspect of human factors (76%) followed by organizational aspect (71%). Consequently the least focused level is interpersonal (38 papers, 56%) which is also a good percentage. However, 25 papers discussed about all these levels and as it can be observed in Figure 7, have been included in “all” category. Among these papers there were some studies where the level of focus could not be interpreted. As it can be observed the figures do not add up to 100% which indicates an existing overlap among pairs of this category.

4 papers between organizational and interpersonal aspects, 14 papers between organizational and individual aspect, and finally 5 papers between individual and interpersonal aspect have overlaps. They were categorized under “unknown”. Figure 7 illustrates the categories and the amount of papers in each one.

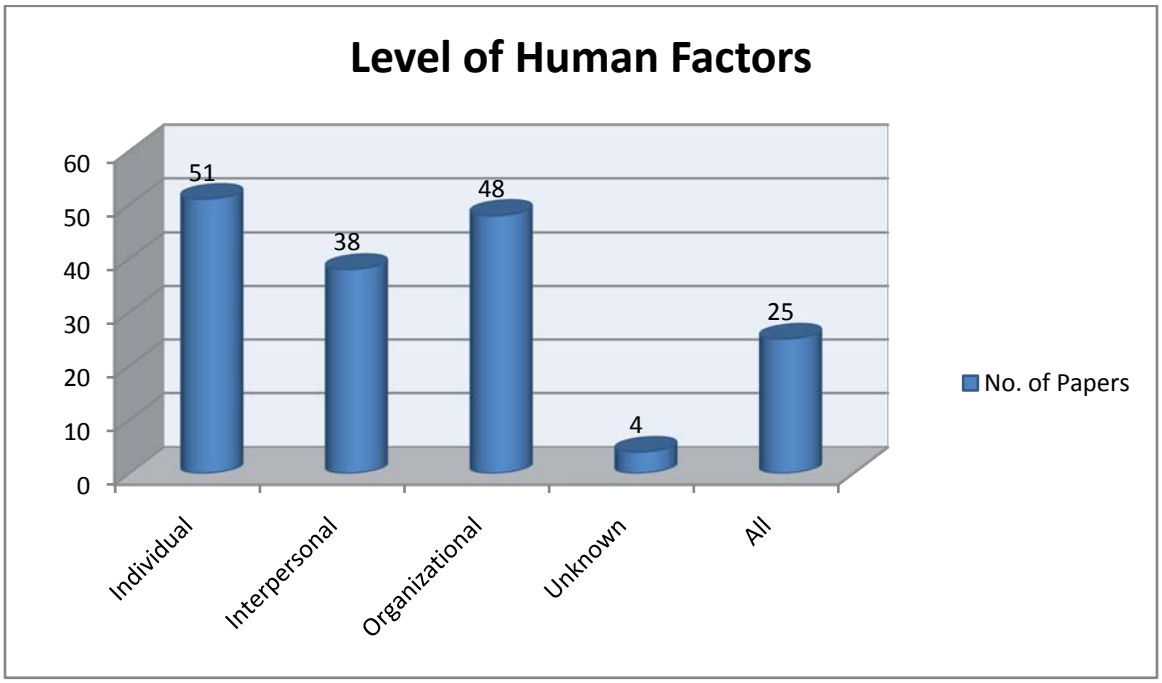

Figure 7, Level of Human Factor Focus

Papers assigned to *Individual* category, generally discuss individual aspects of human factors effect on the development process. Psychological issues such as personality types, characteristics, mentality patterns, judgment, bias and motivations, cognitive issues such as learning skills and styles, abstract/reflective thinking, knowledge sharing, PSP (Personal Software Process) training and individual process and improvement, estimation skills, customer satisfaction and management skills are the centre of focus in this level.

Interpersonal related papers address issues that are related to the communication, collaboration and cooperation of people when performing in a group, when working in an organization and how these communication issues influence the development process. Group work, Pair Programming, group psychology, knowledge sharing, group learning, multicultural environments and groups, group training, innovation and group performance and productivity are the most common discussed topics among these papers.

Papers in the *Organizational* level, discuss human factors involved in organizational level of the development process. Staffing, organization environment (work environment), training classes and workshops, management issues, cost estimation and scheduling, organization and project planning, traditional/ preplanned organizations and agile organizations are the most addressed organizational issues and human factors.

This category (context) is closely related to the management category specifically people management and people issues/human factors subcategories. More information about the details of these factors is explained in section 3.3.

3.1.3 Publication Year (P7)

When reviewing the publication year of the papers (P7 in section 2.5.7) some interesting figures and results show up which highlight the lack of research in some period and its growth in another. This can help as a good guideline for future research in the area. As it has been explained in section 2.5.7 studies that were selected for the data extraction have been published within the last ten years (2000-2010). Among 67 papers, a total of 28 papers (41%) were published during the first 5 years (2000-2004) and 37 papers (55%) during the last 6 years (2005-2010).
The highest number of papers (9) was published in 2002 and the least (2) was published in 2006. Figure 8 shows a schematic view of the distribution of papers according to their publication year.

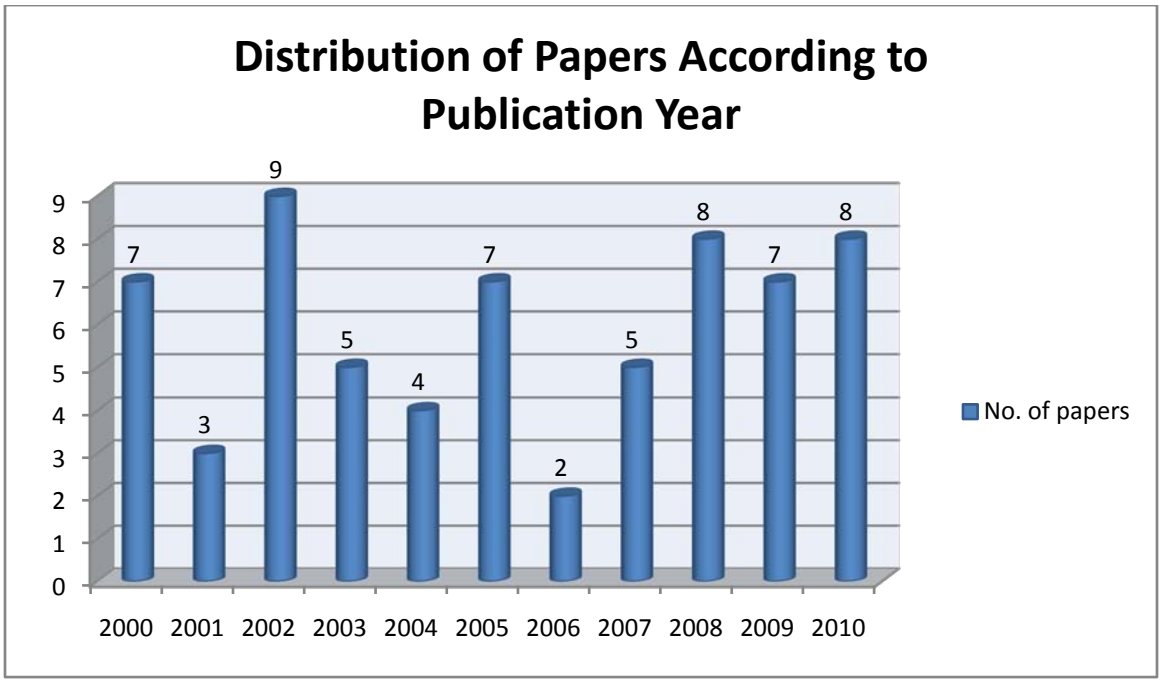

Figure 8, Distribution of Publication according to Publication Year

As it can be observed from figure above (Figure 8) during the last 4 years 28 papers have been published and the number of publications in each year has grown. These figures indicate a growing research interest in this area during recent years, which can invites more researchers to conduct research projects within the area of software development human factors.
In the following sections (3.2 to 3.6) the results of the data extracted for each research question has been summarized followed by the relevant analysis and discussion.

3.2 Software Development Lifecycle Phases (RQ1)

As mentioned in previous chapter first research question of this project is to investigate the human factors and their effects on different software development process phases. To do so this project used the SWEBOK [125] (Appendix A) and investigated human factors related to each phase and its subcategories and their effects in the covered journal papers. This section summarizes the results and analysis related to the first research question.

3.2.1 Results

As the goal of this project was to identify human factors in software development process, consequently one sub goal of the project was to find the most and least discussed development phases and their relevant human factors. The purpose of this research question is to help the researchers see the connection between the development phase and human factors and identify which phases focus least/most on HF. The results have been summarized in Table 8. Based on the SWEBOK [125], development process includes 5 main phases (Requirement Engineering, Design, Implementation, Test and Maintenance) which will be analyzed respectively in section 3.2.2.

Table 8, Development Phases

Software Development Phases	**Papers**	**Frequency**
Requirement Engineering	[33],[35],[37],[47],[50],[51],[52],[66],[72], [57],[32], [60],[48],[38],[70],[41],[43],[44], [53],[55],[63], [65],[68],[69],[73],[74],[75],[76] [77],[79],[80], [81],[84],[85],[88],[30],[91],[92],[93],[96],[97],[99], [100],[101],[102]	45
Design	[33],[35],[47],[51],[72],[32],[48],[38],[70],[39],[40] ,[41],[43],[44],[46] [53],[54],[55],[78],[56],[58],[61], [68],[69],[71],[74],[75],[76],[77],[81] [30],[93] ,[96],[99],[102]	35
Implementation	[35],[37],[47],[50],[51],[66],[72],[32],[60],[36],[38], [70],[39],[40],[41] [43],[44],[53],[55],[62],[68], [69],[74],[75],[76],[77],[79],[87],[99],[97],[100]	31
Test	[33],[35],[51],[36],[38],[70],[40],[41],[43],[53],[55],[5 6],[63],[68],[69] [71],[74],[75],[84],[88],[30],[93],[95] ,[99], [100]	25
Maintenance	[35],[50],[48],[60],[36],[38],[70],[39],[40],[42],[43] ,[53],[55],[62],[63][68],[69],[74],[75],[76],[83],[30], [100]	21

"Requirement Engineering" is the most studied development phase from human factors perspective (45 papers, 67%), followed by "Design" and "Implementation" respectively 35 papers (52%), and 31 papers (46%). "Maintenance" is the least studied development phase; covered by 21 papers (31%).

Following figure (Figure 9) illustrates the distribution of papers according to the development phases. 11 papers are in common among all the development phases indicating their general perspective to the development process. 26 papers overlap between Requirement Engineering and Design, 26 papers between RE and Implementation, 20 between RE and Test and finally 17 between RE and Maintenance overlap. 22 papers overlap between Design and Implementation, 20 between Design and Test and 16 overlap between Design and Maintenance. 16 papers between Implementation and Test, 18 between Implementation and Maintenance and finally 15 papers between Test and Maintenance overlap. As it can be observed the highest overlap is among both RE and Design and RE and Implementation.

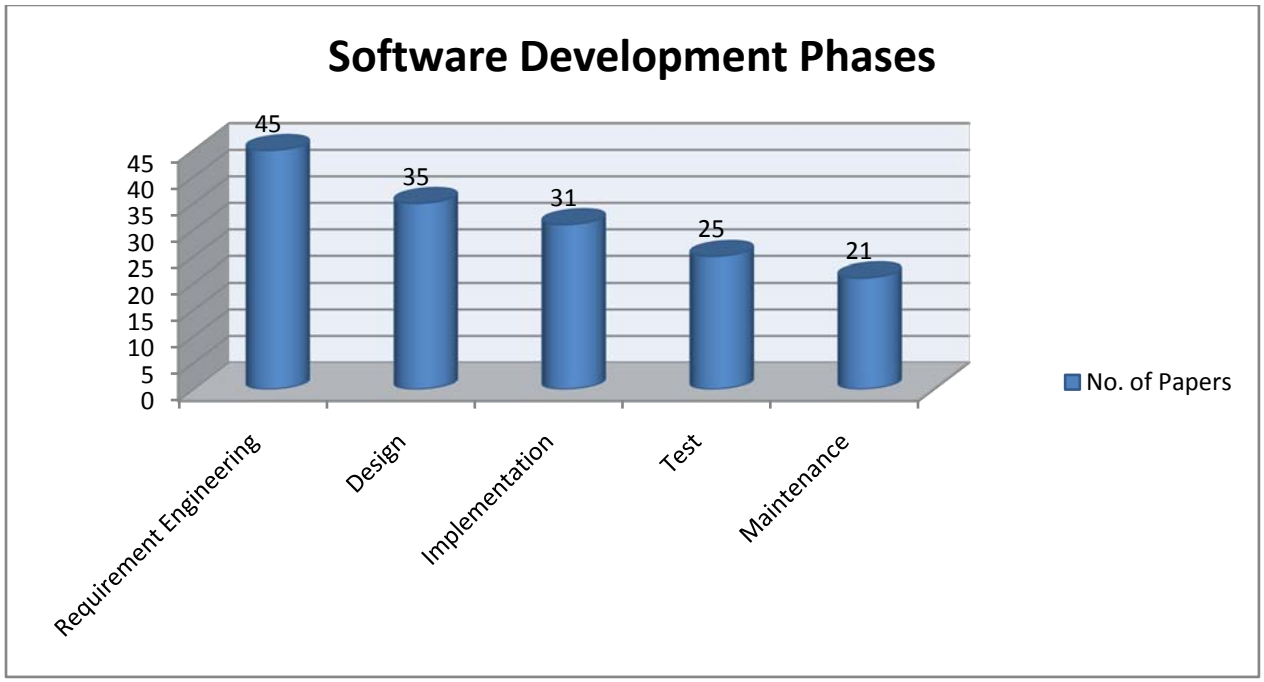

Figure 9, Software Development Phases

3.2.2 Analysis and Discussion

Development phases are more often discussed in general terms within the papers reviewed by this project. For instance a paper by Luiz Fernando Carpets et al. with a title of "Making Sense of Software Development and Personality Type"[32] discusses different personality types and how they match a specific development phase or career. This paper discusses development phases but does not specifically go into any detail for these phases. A high proportion of the papers in this category (Software Development Phases) are of this type. Although this is almost what has been

expected from the beginning of the project as the research questions of this review do not aim for technical details of development process.

Requirement Engineering: Requirement engineering phase was divided into two main subcategories based on the SWEBOK [125] that the extraction form was developed upon. These subcategories are Requirement Analysis and Requirement Validation. The papers in this area were mostly about *requirement specification* (22 papers), [57,33,50,51,52,44,53,66,76,77,85,88,30,91,92,93,96,99,100,101,102] *elicitation* (10 papers) [48,38,70,44,63,65,73,80,81,30] and *negotiation* (4 papers), [37,51,57,102] which are more related to the Requirement Analysis subcategory. The rest of the "Requirement Engineering" related papers were about change requirements (4 papers), [30, 49, 60,101] that can be considered as requirement analysis as well. There is one other group of papers in Requirement Engineering (16 papers) category which does not include any detail for this phase but as the paper has a general view over most of the development phases, this paper has been marked to be under this category as well.

Requirement Elicitation is the first phase of Requirement analysis which addresses the process of deriving the system requirements by observations, discussions with users and task analysis [16] The most discussed issues in Requirement Elicitation are cognitive/psychological issues (7 papers) followed by group work, collaboration and communication aspect of requirement engineering (5 papers). Communication issues are not only related to the communication among developers and system analysts but also consider communication with the customer and their participation while conducting requirement engineering. Generally papers related to this phase discuss customer centered development process and how to meet their demands.
During the *Requirement Specification*, the collected information is translated into a document used for the rest of the development process [16] Table 9 gives more information on Requirement Specification related topics.

Table 9, Requirement Specification

Topics	**Papers**	**Number**
Meeting user requirement specification	[50], [51]	2
cross organizational requirement specification	[76]	1
Multi site Requirement engineering challenges	[57],[102], [76]	3
Early requirement study in planning (initiation and scope definition of projects)	[44],[52],[53],[33]	4
Personnel training , cost and time requirements	[77],[50],[52]	3
Human resource requirements	[66],[88]	2
Psychological and management issues in requirement specification	[44],[53],[85], [96], [37], [88], [30], [91], [92], [99], [100]	11
Team work, collaboration and communication	[99], [33],[37], [51],[57],[88]	5
Cultural issues in requirement specification and engineering	[57],[77]	2
Documenting, user participation in requirement specification	[57],[50],[53]	3
Agile requirement engineering	[30],[88],[75],[80]	4
Design requirements	[93]	1

To summarize most papers related to requirement specification cover management and psychological issues involved in requirement engineering documentation and specification (11 papers) followed by team work and communication (5 papers).

Requirement Negotiation is another subcategory of the “Requirement Analysis” that it’s relevant papers cover requirement engineering issues related to the negotiation process of the requirement engineering. Papers and their main focused topics have been listed in table 10.

Table 10, Requirement Negotiation

Topics	**Papers**	**Number**
Communication medias and requirement negotiation	[57],[102],[37],[51]	4
psychological issues	[37],[57],[102]	3
Distributed group meetings and negotiations	[37]	1
Negotiation in Geographical distributed development	[57],[102]	2
Common understanding	[57]	1
Users participation in requirement engineering	[57]	1
Cultural diversity	[57],[102]	2
Management issues and steering	[57],[102]	2

As it can be observed the most discussed topic in this subcategory is communication and negotiation tools (4 papers) among stakeholders distributed geographically, etc. This is followed by psychological issues (3 papers) mostly related to the effects of (virtual) media enhanced communication and face-to-face communications. Cultural diversity is another affecting issue when negotiating the requirements among stakeholders. Perceptions and interpretations vary in different cultures.

As explained in previous chapter *Requirement Validation* is the process of checking the requirements documents and specification with users to reveal the errors and confusions. There was only one paper to the best of our knowledge that could be assigned to this category and discussed the goal review and validation and acceptance test [86].

Design: Design phase is the second most discussed phase in the area of Software Development Human Factors based on this systematic review (35 papers, 52%). This is indicating the effect of human factors involved in design phase from researchers' point of view. One of the most published topics in the area of design and human factors is HCI which was extracted earlier in the primary study selection as it was mainly about the interaction of software/ system with human. The remaining papers were divided into three subcategories: Architectural styles, Design patterns and Software design methods. As some papers are of general nature a high number of the papers in Design category has not explicitly mentioned details about the three mentioned subcategories. Data that could be categorized is explained in more details in Table 11.

Table 11, Design

Topics	Papers	Number
Architectural Styles		
Use of design documents during the design phase	[81]	1
Design and software related courses	[43]	1
Customers satisfaction	[43],[61]	2
Architecture level reliability	[71]	1
Design Patterns		
Cross organizational business process design	[76]	1
Usability, user centered design	[93],[78],[46],[56],[72],[40]	6
Test driven design patterns	[30]	1
User involvement and iterative design	[33]	1
Design Methods		
Object Oriented design methods	[33],[39]	2
Component based design	[38]	1
Design methods (OO, procedural design)	[39],[33]	2
Use of components in reuse and integration	[38]	1

Implementation: Implementation is the third phase of software development which according to the SWEBOK includes three main subcategories: Construction for verification, Coding and Reuse. As it can be interpreted from the subcategories, this phase includes technical issues such as programming languages, integration testing, platforms, etc. This phase is the third most discussed phase in resource papers of this systematic review (31 papers).

Some of the articles in this phase are guidelines for practitioners and developers, case studies carried out while developing special software in an organizations, proposals, reuse guides related to problems identification while implementing software in an organization. 16 papers of the implementation category had overall view about this phase and did not include any detail to enhance categorizing them under these mentioned subcategories.

The least discussed implementation subcategory/topic was *Constructing for Verification* (4 papers). Among the remaining papers the most discussed subcategory was *Coding* (13 papers). Team work, collaboration and psychological issues are the most common topics in Coding

subcategory. Reuse is the second frequent topic in implementation category (8 papers). More details on papers in Coding and Reuse categories and their addressed topics are listed in Table 12.

Table 12, Coding and Reuse

Topics	**Papers**	**Number**
Coding		
ERP customization, adaptation and implementation	[35][62]	2
Personal Software Process	[77], [41]	2
Teamwork and pair programming	[79],[87],[100],[62],[66],[72]	6
Component based coding	[38]	1
Cultural issues	[38][62]	2
People characteristics, psychological and cognitive issues	[33],[44]	2
Reuse		
ERP reuse, customization, adaptation, flexibility	[35][62][60]	3
Cognitive issues and psychology	[35][38][41][44][62]	5
Management issues	[74]	1
Reuse and integration management	[35][38] [60]	3
PSP modifications for reuse	[41]	1
Software reuse in organizations and legacy issues	[51]	1

Test: Most papers in this phase look at human as imperfect creature making several mistakes and errors in all process. Consequently human error rate, performance test, usability test, integration test are the most common topics in this area. 25 papers in software development phases were related to testing. Among these, 13 papers had more general description and did not include any specifications/details about this phase. According to the SWEBOK[125] (Appendix B) Test phase has three main subcategories: *Test levels, Objective of Testing and Techniques based on the nature of the application*.

Test Levels: This subcategory discusses more about the level of test in the development process. It can be unit testing, integration testing, early testing or system testing. Papers in this subcategory mostly discuss unit testing and its related acceptance testing. Other levels of testing addressed by studies are integration testing, early phase and configuration testing.

Objectives of testing: This subcategory is the most common test subcategory (8 papers) including papers on testing quality and performance, acceptance testing, reliability testing and strategic risk management, usability testing and regression testing.

Techniques based on the nature of the application: only two papers were related to this subcategory. These papers discussed component based testing and object oriented testing. List of papers related to test category and their topics is presented in Table 13.

Table 13, Test

Topics	**Papers**	**Number**
Test Level		
Acceptance testing	[35]	1
Integration testing and risk management related to Of The Shelf components	[38]	1
Early phase testing and configuration testing	[36]	1
Objectives of Testing		
Testing quality and performance	[84][99][56]	3
Acceptance testing, regression testing	[63][30][95]	3
managerial and psychological issues	[30] [99][95][56][63][40]	6
reliability testing and strategic risk management	[71]	1
Configuration testing	[36]	1
Usability testing	[40]	1
Techniques based on the nature of the application		
Component based testing	[38]	1
Object oriented testing	[33]	1
Rapid software development	[38][33]	2

Maintenance: The last and the least discussed phase is maintenance (21 papers). According to SWEBOK [125], Maintenance covers three subcategories: Categories of Maintenance, Process and Techniques for maintenance. Among the papers in this category 11 papers have overall view about the maintenance phase and do not specify any details related these subcategories. The remaining papers mainly cover maintenance issues related to reengineering, maintenance acceptance, modification implementation, evolution related maintenance, change management, etc.

There were several papers related to *Process and Techniques for maintenance* subcategories discussing various maintenance issues. These topics and their related papers are listed in Table 14.

Table 14, Maintenance

Topics	**Papers**	**Number**
Process		
Modification implementation	[35][36][60][62]	4
Maintenance acceptance	[63][30]	2
RE process implementation	[50]	1
ERP modification and adaptation	[35][62][80]	3
Configuration management and its similarity to maintenance	[36]	1
Management and psychological	[62][60][63][30]	4
Process implementation	[60]	1
Agile implementation in traditional organizations	[30],[75][80]	3
Software process improvement	[50]	1
Techniques for Maintenance		
Reengineering and change management : ERP reengineering, preplanned and agile organization change management	[39][48][60][30]	4

3.3 Project Management (RQ2)

According to the goal of this project and based on the papers retrieved in this systematic review another aspect of human factors in software development is project management. SWEBOK (Appendix B) was used to identify the main subcategories related to management issues in software development. SWEBOK defines 6 main categories for Software Engineering Management: Initiation and Scope Definition, Software Project Planning, Software Project Enactment, Review and Evaluation, Closure and Software Engineering measurement. Each of these categories was related to project management phases extracted from [20,21] supporting more details and information about the management issues involved in software development. In this section first the results extracted from papers related to management issues are reviewed (section 3.3.1) and later each subcategory is analyzed individually (3.3.2).

3.3.1 Results

Results from the data extraction of papers show that the main project management subcategory discussed by researchers is Software Project Enactment/execution (67 papers, 100%). The second most discussed project management subcategory is the combination of Initiation and Scope definition and Software Project Planning (60 papers, 89%). These two subcategories were decided to be merged in one phase in this systematic review based on the discussions with an expert in the area and difficulties of distinguishing between planning and initiation related activities. The least discussed project management subcategory is Closure (1 paper, 1%) which shows that more research is needed in this area software engineering management. Table 15 includes the list of papers for each category of project management and their frequency.

Table 15, Software Engineering Management Papers Distribution

Software Engineering Management	**Papers**	**Frequency**
Initiation and Scope Definition & Software Project Planning	[33],[34],[51][52][57][66][32][70][86][41][54][55][61][65][69][71][73][75][79][80][81][83][84][85][88][30][90][92][93][95][96][97][98][99][100][101][102][50][60][36][39][46][53][78][62][63][74][76][91][38][44][58][77][82][87][35][37][72][58][77][48]	60 (89%)
Software Project Enactment/Execution	[33], [34] [35] [37] [47][50][51][52] [66] [72][57][32][48][60][36][38][70][39][86][41][43][44][46][31][53][54][55][78][56][58][61][62][63][65][68][69][74][75][76][77][79][80][81][82][84][85][87][88][30][90][91][92][93][95][96][97][99][102][71][40][42][64][71][73][98][101]	67 (100%)
Review and Evaluation	[47][50][51][65][68][79][90][97][101]	8 (11%)
Closure	[42]	1 (1%)
Software Engineering Measurement	[33] [34][37][47][52][60][39][86][40][41][42][43][46] [31][53][54][56][58][61][65][69][71][74][75][77][79][81] [83][84][85][87][30][90][91][98][99][100][101]	38 (56%)

Following figure (Figure 10) has a schematic view of the distribution of papers among the project management subcategories.

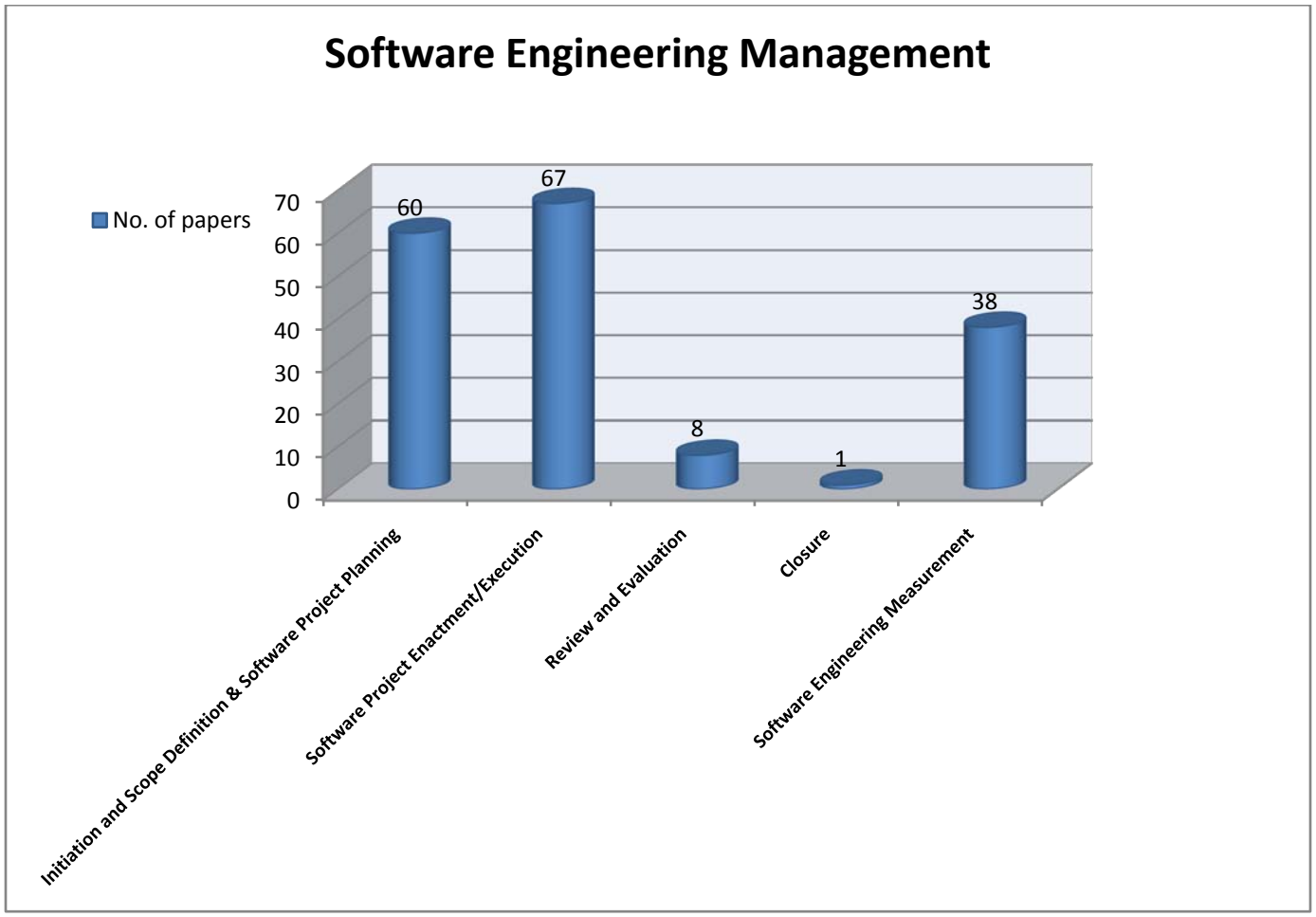

Figure 10, Software Engineering Management

These figures show a need for research in the area of Review and Evaluation and Closure. As the two mentioned software engineering management categories can help in improvement of the development process and its efficiency, there is a need for more studies on those phases and their effects on the process improvement and customer management (marketing).
By following section (3.3.2) each category and its related subcategories, covered topics will be explained and analyzed.

3.3.2 Analysis and Discussion

Software Engineering management /Project management is a human related factor in software development process and has been of great focus by researchers in the area of software engineering human factors. In this section each management category, subcategories and their area of focus will be analyzed. This can be used as a guideline for future research work in the area of Software Engineering Management.
Initiation and Scope Definition & Software Project Planning: As explained earlier due to the proximity of subcategories in these two categories they were merged and considered as one. This category is about the early management issues related to software development. 60 papers out of

67 were related to this category. The main subcategories in this subcategory are *Feasibility Study, Risk Study, and Requirement Study/engineering, Cost Estimation, Scheduling, Resource Allocation, Activity Planning/ Configuration Management*.

Feasibility studies/Risk Studies/ Requirement Studies : 60% of the papers (35 out of 58 papers related to the fist Software Engineering Management category) were related to Feasibility studies/Risk Studies/ Requirement Studies. This makes this subcategory the second most discussed topic in Initiation and Scope Definition and Project Planning. The main discussed topic in this subcategory was risk study (19 papers), followed by requirement study/engineering (11 papers) and finally feasibility study (6 papers). Table 16 is a list of topics addressed by papers in this subcategory.

Table 16, Feasibility Study, Risk Study, and Requirement Study/Engineering

Topics	**Papers**	**Number**
HRM, planning, professional requirements, training needs	[52]	1
Feasibility study and planning capabilities, team work, communication and leadership	[36][51][79][80][75][81][88] [101][66]	9
Risk and requirement studies in agile and preplanned organizations	[30][75][80][83][88]	5
Risk of change, flexibility	[30][83][101]	3
Providers selection risk	[91]	1
Cognitive and cultural issues	[80][81][76][85][97][62][66] [69]	8
Psychological issues	[46][91][92][85][88][97] [101][63][69][61][90]	11
Effects of risk and feasibility study on cost estimation accuracy	[92][97][100][101][63][90]	6
Cross organization requirement studies	[76]	1
Software bidding risk study	[84]	1
Risk study related to information and knowledge management	[85]	1
Risk study and awareness for software reuse	[39]	1
Optimization based scheduling	[62]	1
Process implementation risk study	[50]	1

User participation risks, usability risk and user management	[53],[78]	1
ERP implementation feasibility study	[62]	1
Priority and scope definition	[65]	1
Early(architectural) risk study	[71]	1

Cost estimation: CE is the most focused topic in the first software engineering management category including 63% (37 out of 58 papers related to the first category). This can be due to the impact of cost/effort and budget management and estimation on a successful software development process. Only 15 papers are not common between the first subcategory (feasibility, etc) and cost estimation. Most researchers discuss management and psychological issues leading to wrong/over optimistic cost estimations which make the development process to be a failure. Table 17 gives more details on these papers.

Table 17, Cost Estimation

Topics	**Papers**	**Number**
psychological and cognitive issues (bias, anchoring, judgment, thinking styles, experience, optimistic decisions, over confidence)	[51][46][66][63][65][76][81] [83][85][30][90][91][92] [93][97][44][73][82][95][96][98]	21
Management issues	[46][33][78][63][65][85][30] [97][101][34][44][58][33][38] [77][95][99][60]	18
ERP implementation cost estimations	[60]	1
Productivity, performance	[80][98][93][87][81][34]	6
Collaboration, communication and team working, pair programming	[80][99][87][51][65][100][34] [33][60]	9
Personal Software Process and its effects on CE	[41][77]	2
Individual assessment and self training	[41]	1
Cultural and organizational issues	[66][78][83] [38][95]	5
Cost reduction and risk study	[71][84][97]	3
Software biding	[91]	1
COCOMO model and improving client confidence	[101][34]	2

Scheduling: 30 papers among 58 initiation and scope definition & software project plan related papers discuss scheduling issues. All of these papers have overlap with previous mentioned subcategories (a, b). Subsequently, they discuss similar planning and scheduling issues that have been mentioned in two previous categories. Table 18 is a list of these papers and their topics.

Table 18, Scheduling

Topics	**Papers**	**Number**
Optimization based scheduling	[61]	1
Team work and communication	[80][99][87][65][100] [33][60]	7
Cognitive and psychological issues	[56][44][82][86][87][117][120] [121][123][127][129]	11
Management issues	[33][81][40][70][75][86][87] [105][117]	9
PSP	[27][105][131]	3
Performance	[115][123]	2
Scheduling and planning issues in agile development	[133]	1

As it can be interpreted from the topic mentioned above, cognitive issues, different management and psychological issues and team work have been on focus in the papers related to scheduling.

Resource Allocation: 20 among 58 papers discuss resource allocation management issues. The most common topic related to resource is human resource management and how managers should improve developers' performance and productivity by using different training methods, increasing communication and group spirit among developers and their related management challenges. More details on these papers is listed in Table 19.

Table 19, Resource Allocation

Topics	**Papers**	**Number**
Human Resource Management (HRM)	[52][66][46][55][58] [69][75][88]	8
Professional requirements	[52][66][55][69][81]	5
Training needs and resources	[50][51][52][66][46] [62][77][30]	8
Communication resources, resource constraints	[37]	1
Collaboration, communication needs	[51][72][55][65][75] [88][81]	7
Role assignment	[58]	1
Optimization based scheduling	[61]	1
ERP	[62]	1
Priority management and conflicts in these priorities(cost, time, etc)	[65]	1
Work place/environment	[75]	1

Activity Planning/Configuration Management*:* This subcategory of project planning is more related to task and activity planning in early phases of software development process. This phase leads to a more efficient and less time consuming development process.
25 papers have mentioned some configuration management related issues. Team work, communication and collaboration are the most common factors when researchers discuss configuration management or activity planning. More details on papers in this subcategory are presented in Table 20.

Table 20, Activity Planning/Configuration Management

Topics	**Papers**	**Number**
Task/role assignment	[34][32][48][58][79]	5
Team work and communication	[34][35][37][51][72][3][102][36][70][65][79] [80][83][88][30]	15
Virtual team working	[51]	1
Management/Human Resource Management	[66][70][39][41][58][65][69][80][83][88][30] [98]	12
Personal participation	[57][102]	2
Configuration management in Open Source Software companies	[36]	1
Configuration management in software reuse	[39]	1

Figure11 summarizes the distribution of papers among Initiation and scope definition and Software project planning.

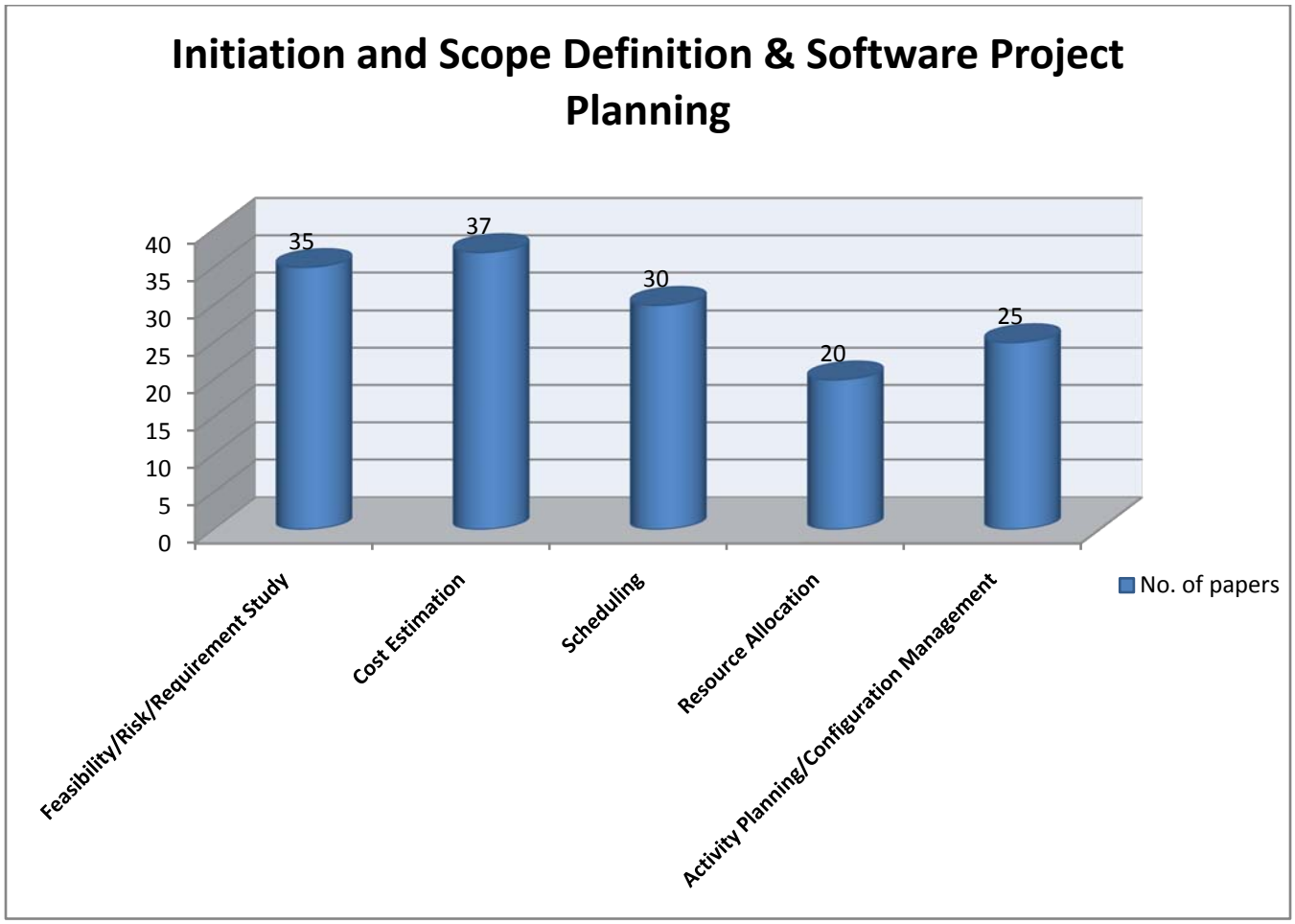

Figure 11, Initiation & Scope Definition and Project Planning (Software Engineering Management first two phases)

As it can be observed from the figure above the most common topic is Cost Estimation followed by Feasibility/Risk/Requirement studies. The least discussed topic is resource allocation which indicates the need for more research in this area. However, there are different levels of resource management to study such as Human Resource Management, Training Needs, Professional Requirements, etc which have impact on the success level of a development process.

Software Project Enactment/Execution: This phase is the most discussed one among papers reviewed in this study covering *People Management*, *Risk Analysis and People Issues* categories. ***People Management*:** this level is the most common topic among the execution/enactment related papers. This category includes 59 papers (88%) discussing different people management issues varying from people/developers communication and relation in development groups and teams, communication among different stakeholders in development process (developers, managers, project owners, etc), people cognitive issues (thinking styles, learning, development skills, experience), training, human resource management and psychological issues (mentality

patterns, personality characteristics, change resistance, motivation, commitment, independence, rationality), to management issues (management challenges in implementing new strategies/processes, steering, change management, risk management), role/task assignment, Personal Software Process, Software Process Improvement, people performance in groups and individually, efficiency, pair programming and impact of work/organization environment on developers process. All these topics were almost related to all other categories defined here (development phases, software engineering management, human role and research methods), as people management is one of the main aspects of software development human factors and its success. For instance the analysis reveals that in people management subcategory people can be divided into three major groups: people as employees or developers, managers and customers. For the latter case researchers are more concerned about the software development as a product and focus on how to satisfy a customer but for the former case, management is more focused on managing the group, individuals and improving people productivity.

Risk analysis/management: 13 papers of the total number of 67 Software Project Enactment/Execution belong to the risk analysis/management category. This category includes papers that discuss the risk studies, analysis and management during the software development process. This category is closely related to the risk study category in the first two Software Engineering Management phases. These papers and their details are listed in Table 21.

Table 21, Risk Analysis/Management

Topics	**Papers**	**Number**
Early risk analysis and management, Risk study and cost estimation, change risk management	[60][75][30]	3
Failure risk management	[60]	1
Configuration management risk analysis	[36]	1
Risk analysis in choosing Of The Shelf components	[38]	1
Risk analysis related to process management	[30]	1
User participation risk analysis	[78]	1
Risk analysis related to cultural issues	[78]	1
Cost Estimation risk studies	[63][30][90][92][97]	5
Scheduling related risk management	[63][30][90]	3
Group and individual risk management	[69][75]	2

Uncertainty management and analysis related to human judgment	[63][69][30][90][92][97]	6
psychological an cognitive issues	[69][75]	2
software bidding risk management	[62]	1
Effects of lessons learned in risk and uncertainty assessment	[97]	1

As it can be observed among the risk related topics the most common one is Cost Estimation/Effort estimation, its risks and the negative impact on the estimation accuracy [122].

***People issues/human factors*:** this category covers management human factors during the software project enactment/execution that have not been covered by the previously mentioned categories (people and risk management). However, these three categories have a very high overlap. Only 6 papers in this category are not covered by people management and risk management categories, 57 papers are common between people management and people issues and 12 papers are common between risk management and people issues. Hence, in this part the uncommon papers (6 papers) among these categories have been brought and referred to, for decreasing the repetition of the similar topics. Table 22 includes a list of these papers with more details.

Table 22, People Issues

Topics	**Papers**	**Number**
Human Cognition in Technical Communication	[40]	1
Learning styles	[42]	1
Behavioral and cognitive issues, human information processing	[64][73]	2
Psychological issues in planning and decision making	[98]	1
Client's confidence	[101]	1

To summarize people management is the most common software project execution management issues that has been of focus for many researchers. However there are not enough research papers related to risk management which highlights the need for more research in this area.

Review and Evaluation: this phase covers late management issues in development process and software engineering. Reviewing feedback from different stakeholders and participants leads to

improvement in the development process. Customer satisfaction, customer feedback, manager satisfaction, customer confidence and similar issues are the main keys in the evaluation and feedback phase. Although this is an important phase in management only 9 papers have discussed it. A successful review and evaluation phase can lead to a great improvement in Customer Relationship Management, marketing and development process and can offer lessons learned for the rest of the developers and organizations. Among these 8 papers 2 papers could be assigned to *feedback* category. [68][90] Topic covered by the papers in feedback category are: *Mentality feedback activity* in which "team members provide evaluation and corrective information to each other with respect to the mentality patterns and mentality principles already established"[68], lack of feedback and its negative impact on effort estimation(overconfidence)[90].

The rest of the papers in the review and evaluation phase discuss different issues such as impact of employees' participation in workshops and its effect on their use of electronic guides[47], evaluation of implementing RE processes in organizations and meeting customer demands and needs[50], evaluation of user requirements and cost-benefit analysis[51], reviewing and checking the documents (requirement) with customers and evaluating their satisfaction before starting the design and implementation process[79], evaluating the lessons learned session on effort estimation and their comparison with on the job learning[97], software evaluation framework[65] and evaluation of a 2stage model for effort estimation and project duration.

To summarize papers in this category are often concerned with customers' needs, requirements and a way of meeting them to increase customer satisfaction and confidence.

Closure: Among 67 papers only one paper [42] was related to closure management phase. This paper is about software document design and considering users understanding and learning styles in this process. This paper discusses improvement of usability of software documents and guidelines for users with different knowledge/information in the area and learning styles. This paper is theoretical which shows a great need for more empirical research on this phase.

Software Engineering Measurement: This phase is the last management phase according to SWEBOK. (Appendix B) This phase is a management phase highlighting quantitative/qualitative factors in development process leading to more successful, efficient and high performance process. 38 papers have discussed measurement factors varying from quality, performance and productivity to decision making, judgment, etc.

Some of the most focused measurement factors among these papers are: performance (20 papers, 52%), developers' productivity (7 papers, 18%), quality (12 papers, 31%), usability (2 papers, 5%), effectiveness/efficiency (5 papers, 13%), control (1 paper, 2%), process improvement (1 paper, 2%), success and failure (4 papers, 10%), progress (1 paper, 2%) and motivation (1 paper, 1%). The most discussed measurement factor is performance which has been addressed from individual and team work perspective. More details on these papers is listed in Table 24.

Management Challenges: This category has been added in addition to SWEBOK (appendix B) to include the management issues and skills that have been discussed in management related

papers. This category includes 54 out of 67 papers. A list of these papers with more details is presented in Table 23.

Table 23, Management Challenges

Topics	**Papers**	**Number**
Risk management	[36][71][84]	3
Configuration management, process management	[47]	1
Organizational training management	[66][70][62]	3
Personal management(group and individual)	[66][37][55][69][81][87][88]	7
Management strategies (rewarding, etc)	[66]	1
Quality management	[33][77][83]	3
Cost and scheduling management	[33][46][63][65][84][91]	6
Steering/management skills, leadership, decision making, analysis skills	[52][57][102][48][38][39][40][43] [46][62][63][79][80]	13
Organization environment/ office management	[72][54][55][78][62][69]	6
Change management and customization	[35][50][66][60][36][70][33][41] [55][78][62][75][76][80][83] [30][97][100][101]	19
Innovation management	[66]	1
Project management	[48][63][74][76][99][100]	6
Distributed work management	[36]	1
Intellectual capital management	[36]	1
Knowledge management	[57][70][102]	3
Global management	[57][102]	2
Managers' believes and awareness	[39]	1
Procurement management	[46]	1
Customer Relationship Management	[53]	1
Priority management	[63]	1
Conflict management	[87]	1
Bidding management	[91]	1

To summarize the most discussed management challenges are related to change management and steering/management skills.

3.4 Research Methods (RQ3)

This research question (RQ3) was designed to investigate types of research papers in the area of Software Development Human Factors and the methods researchers use for conducting their research projects. As this area is related to psychology, management and human factors, it was expected to see more empirical methods that are common in these research fields. Research methods in this research question were designed to cover most of the research methods in the area. This was pursued based on the pilot extraction in this systematic review, use of research methods description in a systematic review close to this project area of research [5] and experimentation methods in software engineering [17]. These methods are mainly divided into two main categories (Empirical and Theoretical). The Empirical category included Case Study, Survey, Controlled Experiment, Industrial Report and Design of System/Solution/Model/Method/Guideline. Table 24 includes a list of these papers related to each method and their frequency.

Table 24, Research Methods

Research Methods	**Papers**	**Frequency**
Empirical (55 papers, 82%)		
a) Case study	[33][34][47][50][51][66][57][60][36][38][70] [40][41][43][44][31][54][64][78][58][61][63][6 8][69][71][76][77][79][81][82][83][85][88][30] [90][92][93][96][98][99][100][101][102]	43
b) Survey	[33][35][37][47][51][52][66][72][57][60][36] [38][70][39][86][31][53][54][55][56][62][65] [68][69][91][95]	26
c) Controlled experiment	[37][44][54][92][93][95][96][97][99]	9
d) Industrial report	---	0
e) Design of system/solutio n/model/ Method	[50][52][66][72][57][36][38][70][39][41][43][3 1][54][78][56][58][61][68][69][71][73][76][77] [79][80][81][82][83][84][88][30][91][92][93][1 00][101]	36
Theoretical	[35][32][48][40][42][43][46][64][68][73][74][7 5][76][80][82][87][91]	17(25%)

3.4.1 Results

Table 24 shows that Empirical research method (55 papers, 82%) is the most common one. Several papers did not explicitly mention their research method but it was possible to interpret from the paper. Some of the papers in each research method category are interpreted from the text. Among the Empirical papers (55 papers), Case Study (43 papers, 78%) was the most common type of research method. The second most common method was Design of system/solution/model/method (36 papers, 65%). The least used research method is Industrial report (0 papers) followed by Controlled experiments (9 papers).

3.4.2 Analysis and Discussion

Case study is the most common method among the papers used for data extraction in this systematic review. This method was divided into two groups: industrial case study and non-industrial case study. Based on the paper content and some information about the case study also were extracted during the full text reading phase such as the number of case studies conducted in each project/paper, Number of companies or people involved in the case studies and occasionally duration of the case study. Among 43 case studies in this systematic review, 32 papers (74%) used industrial case studies and only 8 papers (18%) were non-industrial case studies. There were three papers which the type of case study could not be extracted or interpreted from [54][71][93]. The results show that almost ¾ of the case studies had industrial background and support which is very promising. However the lack of non-industrial case studies in this area shows that the human factors in software development process has been overlooked by academic researchers. The most common topics investigated by case studies are psychological/management, cognitive issues and team work, communication and collaboration.
Number of cases/companies, their size, names, and duration of these industrial case studies were extracted if it was mentioned in the paper. This information has been summarized for each paper in a table (Appendix D). Also if there was confusion in deciding about the paper research method (case study, controlled experiment and survey) it has been mentioned in the table.

As it can be interpreted from the table most of the case studies have been conducted in several companies, environments or multiple case studies have been carried out in a single organization.
As mentioned earlier only 8 papers were non-industrial case studies. Among these non-industrial case studies 4 cases were multi cases or multi company cases [44][77][98][81] and two were single cases in University of Southern California[30] and Southeastern Louisiana University faculty[79].
For both industrial and non industrial case studies it's more reliable when a case is conducted in several/multi companies or several cases have been carried out in a single paper. This increases the readers' confidence when they refer to these papers or use them as a basis of their research.

Survey: Survey is another common research method in empirical studies. When a researcher uses questionnaire or interviews (semi structured, structured, web based, emails, etc) to collect quantitative or qualitative information about his/her research question the papers is assigned to survey category. [24,26,27]

As it can be observed in Table 10 survey is the third most common empirical research method that has been used in 26 papers (47%). Among these papers 16 papers have explicitly mentioned they have used *interviews* as their research tool for the survey. And among these 16 papers 3 papers have explicitly mentioned they have used semi-structured interviews [57][47][56]. For one paper it can be interpreted that semi structured interview has been used by the researcher [68] as it mentions the interviews were carried out occasionally during the coffee breaks and meetings. There are one structured interview [39] and one consistent interview [60] among these papers as well. The rest of the papers have not explicitly mentioned type of the interview they conducted during their survey. These figures show that the most common type of interviewing is the semi-structured interview.

Questionnaire is the other survey research tool used by 9 papers which had different types such as web based questionnaire [31] and self administered questionnaire [55], scaling questionnaire[66], and open ended questionnaires[60].

The number of respondents/organizations/ cases involved for the survey was not possible to be extracted for all the papers as they didn't mention any details of their survey. Although there were several papers for which this data has been extracted and summarized in a table (Appendix E)

As it can be observed in appendix E, most papers mentioned details about their surveys and only 4 papers didn't. This can be a good factor for evaluating the reliability of these papers according to the level of reference other researchers can make to the results of these papers as they have explained details of their survey process and their sources.

Controlled experiment: This category was the hardest to find in papers as it was not explicitly mentioned whether they have used this method or not. Subsequently most of the papers assigned to this category have been decided by interpretation. Thus there is a possibility of wrong assignment in this category. For instance there are some papers that could not be decided whether they are case studies or controlled experiments. 9 papers were assigned to this research method. Among these, 8 papers overlap with case studies and survey methods. Only one paper [128] just uses this research method. In this papers the author has empirically investigated the impacts of lessons learned sessions by dividing people into two groups, one participating in these sessions and the other doesn't. Then the author has asked both groups to carry out 5 similar tasks and the paper is about the differences and similarities that have been made by the training sessions. Although there are some papers based on the content of the paper assigned to both controlled experiment and case study (5 papers) and controlled experiment and survey (3 papers) This is indicating the need of more controlled experiments research in the area of Software Development Human Factors. The most common subject that has been studied by controlled experiment is Cost Estimation and its related psychological, cognitive and management issues.

Industrial report: The papers in this category must be a report from industrial experience which explains the industrial case with no predefined research questions or do not mention any research method at all. [25] Among the primary study selected papers in this systematic review, no paper was found to use this research method.

Design of system/solution/method/model: This research method is the second most common empirical research method used in 36 papers (65%). Papers using this research method offer some type of solution, method, framework, guideline or model that can be used by other researchers in the area. Among 36 papers, the most common type of solution is model (9 papers, 25%) followed by guideline (8 papers, 22%) and the least common solution type is method (2 papers) and metaphor (1 papers, 2%). A list of these papers (system/model/framework/guideline) has been brought in Table 25.

Table 95, Types of offered solutions in Design research method

Paper	**Type of solution**	**Frequency**
[50][52][72][31][58][68][71][79][84][100]	Model	9
[52][36][83][30][91]	Framework	6
[61][73][76][91][66][39][43][54]	Guideline	8
[36][70][82]	Lessons learned	3
[38][78][92][93]	Hypothesis	4
[39][30][101]	Approach/proposal	3
[56]	Metaphor	1
[68][71]	Method	2

Theoretical: papers using this research method mainly discussed different concepts theoretically. As mentioned earlier, among 67 papers, 17 papers (25%) have used theoretical research method. This shows that most of the researchers in the area prefer empirical research methods and support their research project with empirical data which increases the reliability of their projects. These theoretical papers discuss various issues and concepts such as personality types of software developers [32], IT, work centered organizations and leadership [48], technical communications [40], cognitive issues [40,42,43],systematic review on CE[73],systematic review about motivations of software engineers[74], agile software development[75][80], individual professionalism[82] and systematic study on pair programming human factors[87].

To summarize the answer to this research question (Research methods), the most common research method is empirical and the most common empirical research method is case study. Lack of empirical research methods such as controlled experiment and industrial reports highlights the need for more research project using these methods.

3.5 Human Role (RQ4)

Human role is an interesting factor in software engineering human factor research as it's interesting to investigate how researchers see human being involved in the development process. This category is closely related to different software development phases and project management. As it was explained in previous chapter the predefined roles were: Customer/user, Developer and Manager. More details about the last role (manager) and other roles were added during the extraction phase. Table 26 includes the papers related to each role and their frequency.

Table 26, Human Roles and related papers

Human Role	**Papers**	**Frequency**
Customer/User	[79][80][85][88][100][101][40][42][43][57][44][46][50][51] [53][54][78][33][60][61][62][65][72]	23
Developer	[73][74][75][76][77][79][80][81][82][83][84][85][35][87][88][30][90][92][93][95][96][97][98][99][100][36][101][102] [37][32][39][86][40][41][57][42][43][44][46][47][48][50] [51][52][31][53][54][55][78][56][33][58][60][61][63][65] [66][68][69][71][72][34]	63
Manager	[74][75][77][79][80][83][84][88][30][91][92][93][96][98] [36][99][100][102][39][86][40][41][57][46][47][48][49][51] [52][54][64][55][56][58][33][60][61][62][63][65][66][68] [69][34]	45

3.5.1 Results

Results gained from the 67 papers indicate that the most discussed human role is developer (63 paper, 94%) and the least discussed role is customer/user covered by 23 papers (34%). Figure 12 illustrates the distribution of papers according to human roles.

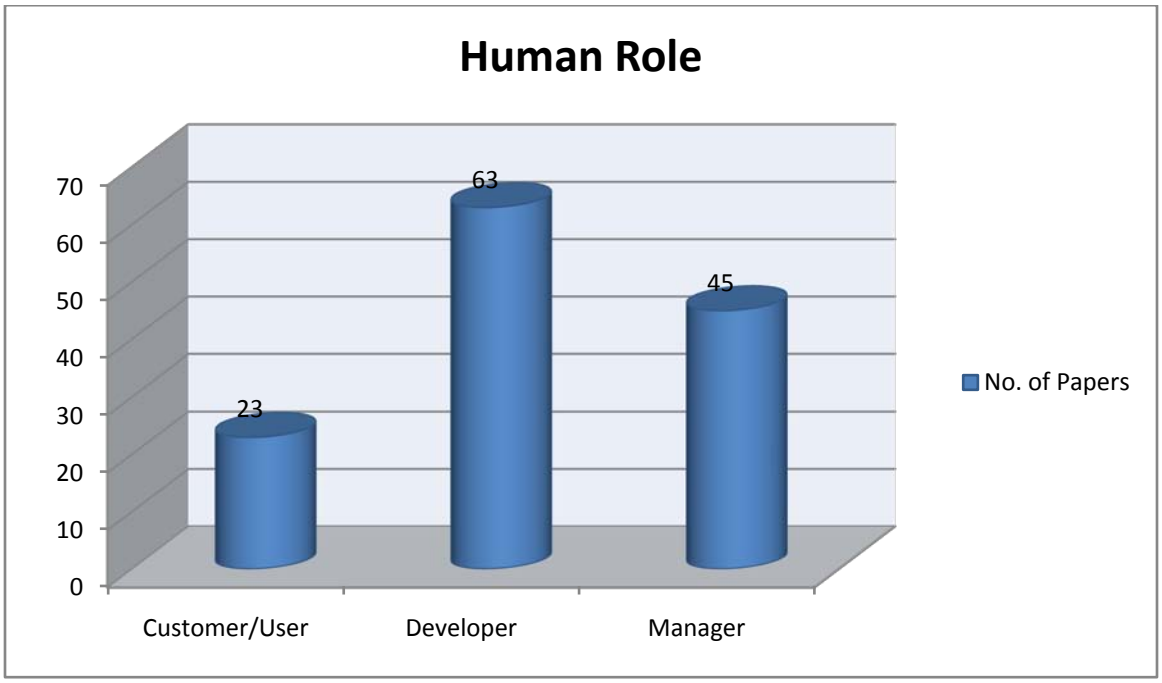

Figure 12, Human Roles

There were two papers [38][70] in which the human role couldn't be extracted (manager or developer were assigned to these two papers), therefore they are not included.

3.5.2 Analysis and Discussion

Customer: Customer related papers cover various customer related issues that have been listed in Table 27.

Table 27, Customer

Topics	Papers	Number
User participation in software development process and RE	[33][50][51][72][57][100] [102]	7
Customer satisfaction	[53][102]	2
Customer behavior and related design issues	[38][70][60][40][42][44] [46][78][56]	9

As it can be observed, customer role is more discussed in papers focusing on early Software Development Phases (requirement engineering, design), middle and final phases of Software Engineering Management (Execution, Review and Evaluation and feedback). This is due to the user/customer involvement in the early phases of development to enable understanding of customer needs and emergence of user centered development and design. Consequently this leads to more customer satisfaction and confidence in the development process and better Customer Relationship Management. From business and economical point of view Software Development is successful if it meets customer demands and achieves customer

satisfaction/confidence. However this role is the least discussed role (34%) and should be considered by researchers.

Developers: Developers are the most focused role in software development human factors research papers (94%) which is almost expected. This is due to dependence of software development process on the individuals involved in developing the software. There are important technical issues in software development but if the human aspects are not paid enough attention the process won't succeed. Researchers have focused on developer role in different phases of software development process and software engineering management. Specifically middle phases of software development (Design, Implementation) and Software Engineering Management (Software Project Enactment/Execution) consider developer role. Design and Implementation in development processes are completely dependent on developers and how they conduct these phases affect the development performance, process productivity, effectiveness and success. [31,32,55] Some studies have been carried out to investigate which personality type matches each of the software development tasks. "By mapping soft skills and psychological traits to the main stages of the software life cycle, the authors claim that assigning people with personality types best suited to a particular stage increases the chances of the project's successful outcome."[32] Also links between the personalities, views and attitudes of software engineers have been studied which is more in individual level of human factors.[31] As these two are more related to the execution management, software project enactment and management is the most common management phase in papers which basically includes people management, risk management and people issues. From management point of view it's useful to know how to manage developers to work individually or in groups with high performance, efficiency, while considering communication, collaboration[36][53][54][74][75][87][37][66], sharing and learning[57, 42], self assessment, cultural diversities, cognitive and psychological issues (learning, thinking, judgment, resistance, personalities, etc). [31][32][55][96][42]These are the most common discussed people management issues related to Developers.

Manager: Manager is the other human role that has been discussed by many researchers in this area of research (45 papers, 58%). As manager acts during all phases of development process and different activities from feasibility study, cost estimation and scheduling to change management and steering, this role is expanded all over these phases. But as mentioned earlier in this section, manager role has different types, which was expected to be extracted from each paper during the full text reading phase. These management roles are Senior manager, Chief Information Officer, Project manager, Function Manager, System Manager, Senior Manager, Product Manager, Organization Manager, Team/Office Manager, Coordinator, Sponsor, Organization/Project Owner, Support Manager (Customer Manager), Employer, Director, HR Manager, Database Manager and IT Manager. Figure 13 illustrates distribution of papers among different types of manager roles.

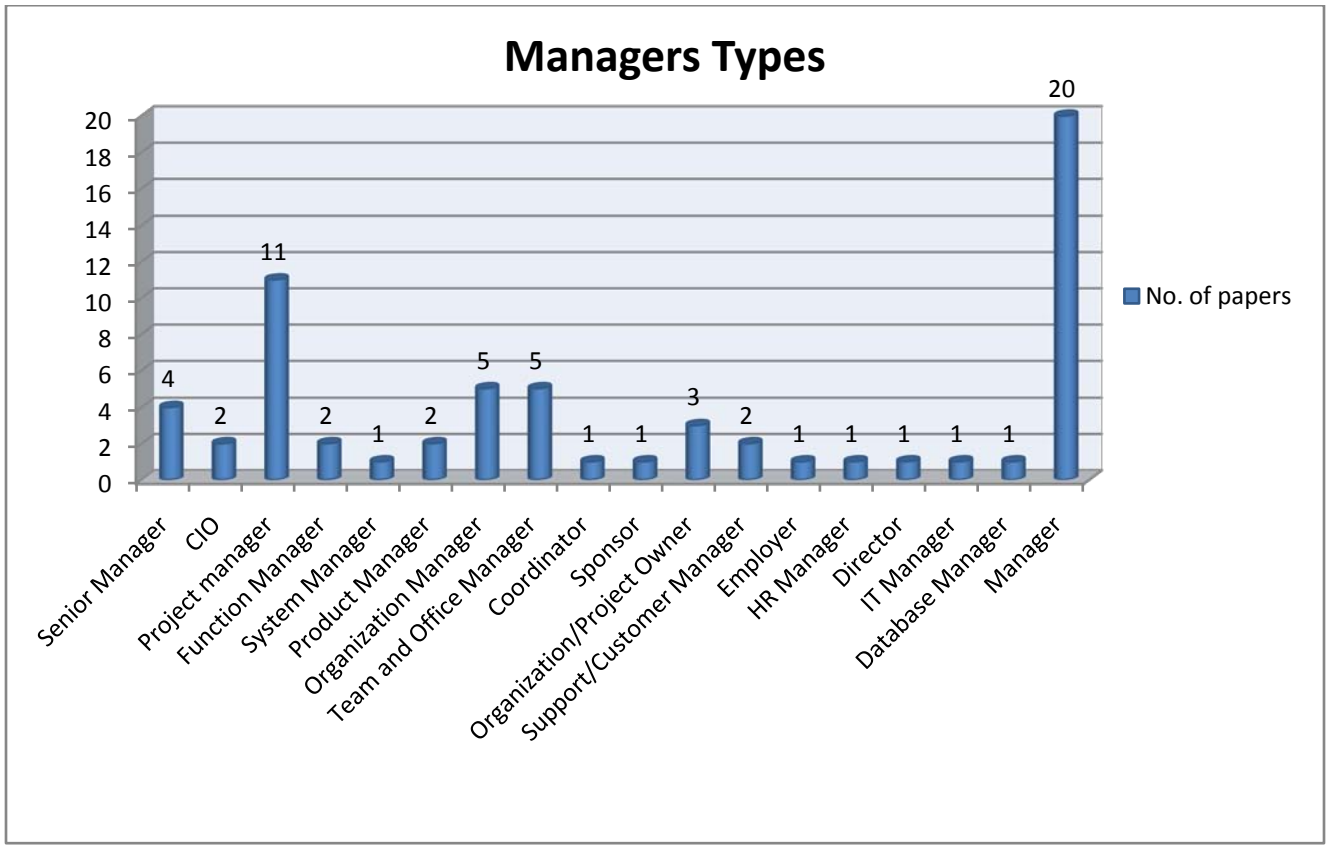

Figure 13, Manager Types

Organization Manager and team/office manager are the second most mentioned (5 papers) manager types in software development human factors research papers. As it can be observed from Figure 13 High proportion of papers (20) are in "manager" column where the type of manager could not be extracted and was concluded to be "manager".

To summarize Customer/User is a common role in studies related to early phases of the development (RE, Design) and also late phases of the project management (evaluation, feedback). Developer role is mostly related to the middle phases of both project management and development and finally manager role is a broad category covering most of the phases of both categories.

3.6 Assessment of Papers Searchability

In this section the searchability assessment phase conducted during the full text reading has been discussed. As mentioned instead of assessing the quality of papers, this phase is based on the author's judgment of papers' keywords, abstracts, titles, thus assessing their searchability. This is a carried out to examine other researcher's possibility to search papers in the field. Five assessment questions were designed to investigate the title, abstract and keyword searchability papers of this systematic review. As explained in previous chapter, these questions were added to the bottom of the extraction form and were answered while reading each paper. The answers to

each of these questions could be yes, partially and no. Partially was used when the answer was not even exactly 100% yes nor 100% no. Results of the searchability assessment for each paper are summarized in table 28.

Table 28, Searchability Assessment Questions and Results

Quality Assessment Questions	**Yes**	**Number**	**Partially**	**Number**	**No**	**Number**
Q1. Has the correct author keyword (human factor in software development and similar) been used?	[44][47][48][31] [64][65][68][74] [93][97]	10	[38][70][86][52][55] [78][56][58][60][66] [76][77][96][99][100]	15	[33][39][40] [42][50][51] [61][62][69] [71][72][73] [84][91][101]	15
Q2. Is the abstract searchable about the topic?	[57][38][32][39][40][43][44][46][49][50][51][31][64][55][60] [66][68][69][72][81][83][87][88][97]	24	[36][37][70][86][41] [54][78][56][58][61] [62][63][65][71][74] [76][77][79][80][85] [30][90][92][95][96] [100][102]	27	[33][34][35] [42][47][52] [53][73][75] [82][84][91] [93][98][99] [101]	16
Q3. Does the abstract include the keywords (Human factors, human aspect, cognitive issues, software psychology, human centered, etc.)	[57][38][32][39][50][51][66][72][48][60][40][43][44][46][31] [64][55][56][68][69][81][83][87][88][97]	25	[37][52][36][70][86] [41][42][54][78][58] [61][62][63][65][71] [74][76][77][79][80] [85][30][90][92][95] [96][93][100][102]	29	[33][34][35] [47][53][73] [58][82][84] [91][98][99] [101]	13
Q4. Is the title searchable?	[32][51][66][60][43][44][46][56][69][82]	10	[57][36][40][41][42] [31][54][78][68][74] [75][77][87][90][102]	15	[33][34][37] [47][50][52] [72][48][48] [70][39][86] [53][64][55] [58][61][62] [63][65][71] [73][76][79] [80][81][83] [84][85][88] [30][91][92] [93][95][96] [97][98][99] [100][101]	42
Q5. Does the title include the relevant	[32][51][60][66][43][44][46][3	11	[57][36][40][41][42] [54][78][68][74][75]	14	[33][34][37] [47][50][52]	42

keywords?	1][56][69][82]		[77][87][90][102]		[72][48][38] [70][39][86] [53][64][55] [58][61][62] [63][65][71] [73][76][79] [80][81][83] [84][85][88] [30][91][92] [93][95][96] [97][98][99] [100][101]	
Total frequency	80		100		128	

As it can be observed from Table 27 the highest proportion of the answers are no for all the questions. This shows that authors did not use research keywords in their titles or abstracts which could increase the searchability of their papers. The questions with highest number of "no" answers are 4 and 5 (42 papers, 62%) which are about title and title keywords. This is indicating that more than half of the papers that selected for this systematic review did not have searchable titles which increase the difficulty of finding them. Figure 13 illustrates the results for each research question regarding the searchability assessment of the papers.

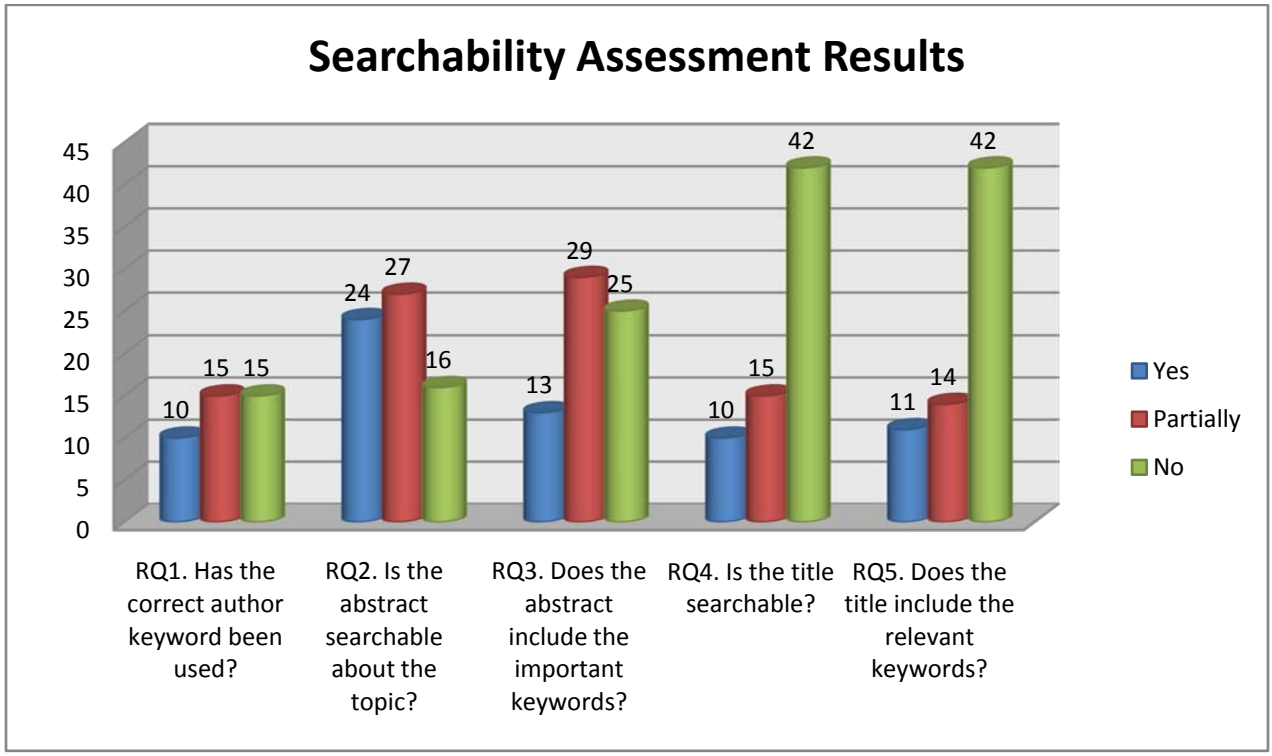

Figure 13, Searchability Assessment Results

As it can be observed from figure 13 the highest amount of "yes" answers is related to abstract searchability and the least number of "yes" is related to both authors' keywords and titles searchability. Thus researchers aiming to conduct systematic reviews in this area have to consider these searching issues in their search procedure.
The results of this phase show that a high percentage of the researchers in this area

- do not have keywords that would increase searchability
- do not include searchable keywords in their abstracts,
- do not consider these keywords when generating their papers titles

3.7 Summary of results

To summarize this chapter, the connection between different categories of the research questions has been illustrated in Figure 14, 15 and 16 to illustrate how software development phase, management and human roles are related. Figure 14 illustrates the connection between software development and management phases.

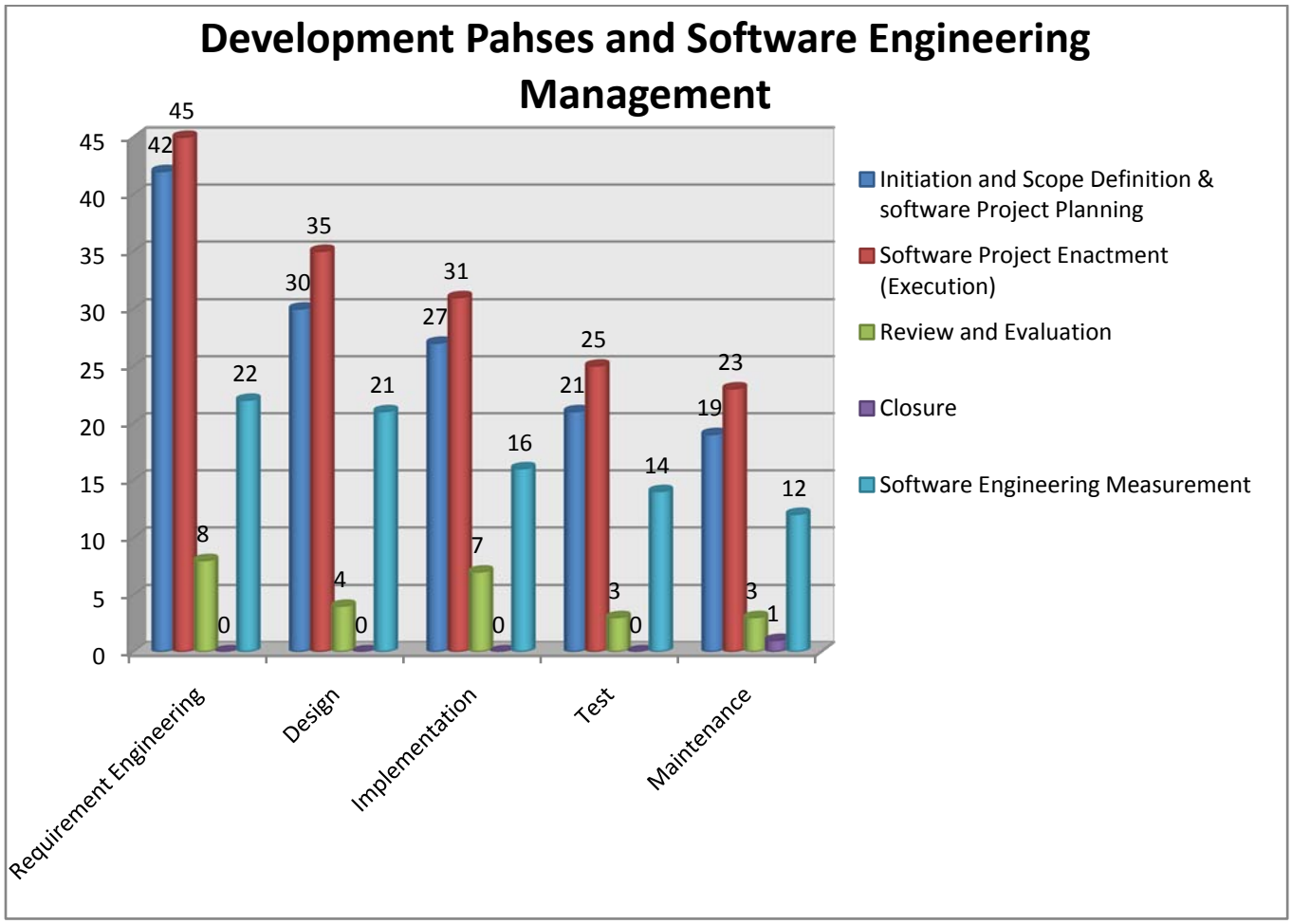

Figure 14, Software Development Lifecycle and Software Engineering Management

As it can be observed the most common software development phase studied from management point of view is Requirement Engineering and the least studied phase is Maintenance. Also the least addressed management phase in different development phases is Closure. The most common management level discussed in various development phases is Software Project Enactment (Execution).
Following figure (Figure 15) illustrates the relation between human role and software development phases.

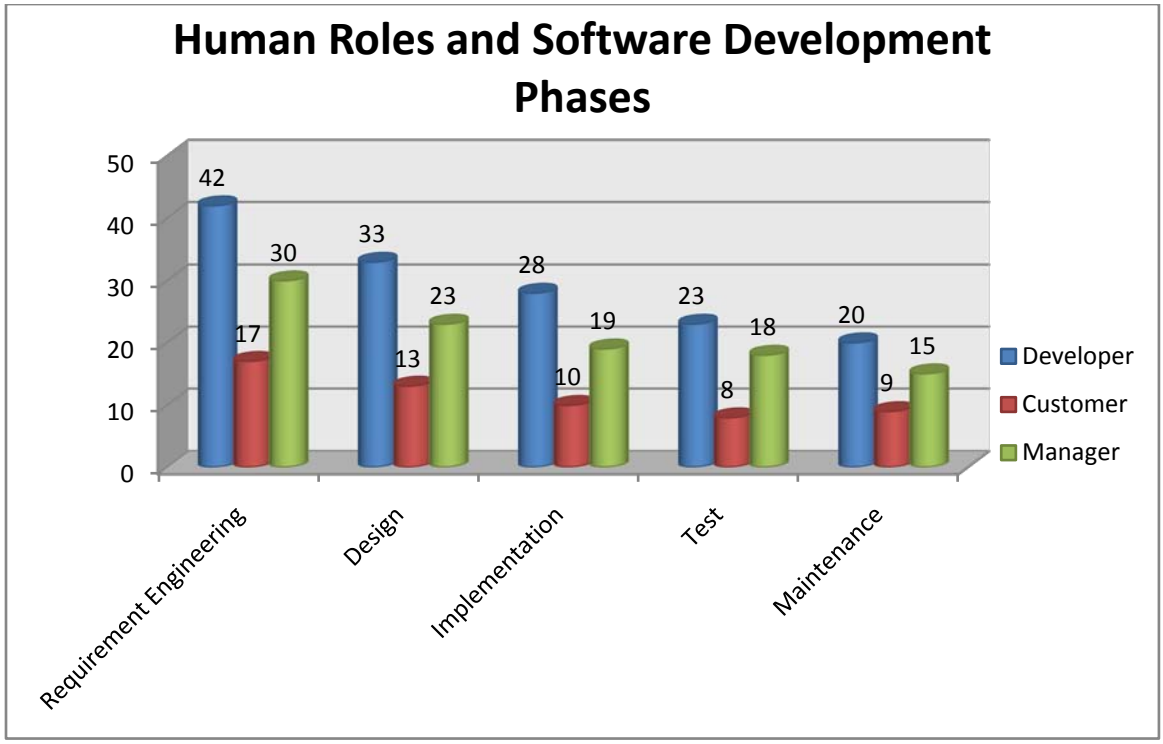

Figure 15, Human Roles and Development Phases

As concluded before developer is the most discussed human role in software development phases. The highest level of "Developer" role has been addressed by papers is in Requirement Engineering phase. The least discussed role is "Customer" which has been addressed in Test phase.
Finally the last figure (Figure 16) illustrates the connection/relation between software management and human roles.

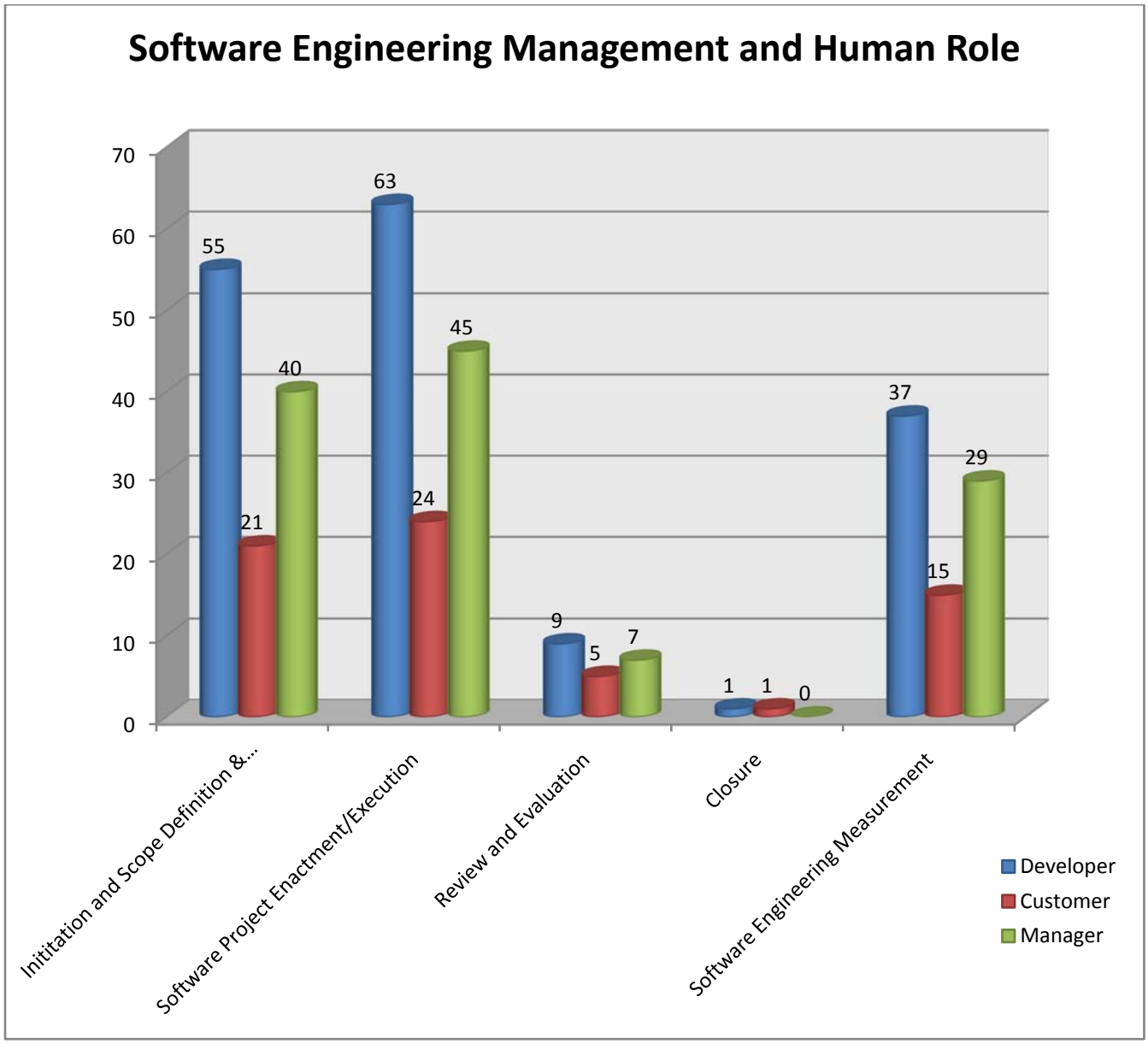

Figure 16, Software Engineering Management and Human Roles

As it can be observed from the figure the most common role among software engineering management phases is also “Developer” which has been mostly addressed by Software Project Enactment (Execution). The least discussed role is Customer within the Closure management phase.

4 VALIDITY THREATS

In this chapter validity threats to this systematic review and its results will be identified and discussed. The threats divided into four main categories Investigator Bias, Publication Bias, Primary Studies Related Threats and Bias and finally Threats to Data Extraction Phase and its Results. Each of these threats will be identified and discussed respectively in this chapter.

4.1 Investigator Bias

As this systematic review was conducted by an individual researcher, there is a potential of more validity threats in comparison with systematic reviews conducted by several researchers, due to the effects of individual judgment, bias, anchoring, and other types of investigator related biases [101][40] on different phases of this systematic review. These biases could have affected primary study selection and the data extraction phase, as well as its results which will be explained in following section. However, different phases and results gained in this systematic review were checked and approved by another researcher who has previously conducted systematic reviews in similar areas to reduce this bias and threat. Every task has been carried out several times (if possible) to make sure the results are similar to previous identified results. For instance abstract reading phase was conducted several times to reduce the probability of making mistakes by investigator. However it cannot be confidently claimed that these have solved the investigator bias/threats but to some extent has reduced it.

4.2 Publication Bias

This bias is related to the low publication probability for negative research outcomes and high probability of positive outcomes publication.[12] This bias is low in this systematic review since none of the research questions have been designed to evaluate a specific phase performance or specific concept such as management phases or development phases. Due to the general nature of this systematic review, papers studying human factors challenges in software development have been as equally analyzed as those discussing the positive effects of human related factors on development process. Also the results indicate that influential organizations or publishers have not promoted any specific development or management phase as each of them have various papers under both topics. As this systematic review was conducted based on the papers retrieved from three major databases and it was restrict to the full text journal papers, there might be a bias related to the papers being published in journals and not other types of papers(proceedings, conferences, etc).This restriction was made to increase the reliability of the accumulated information and make a trade-off between the amount of retrieved information and its reliability.[12] Subsequently there might have been some journal biases in these databases. For instance IEEEXpolre includes more engineering related journals; thus there might have been

some papers in other journal categories which have not been included in IEEEXplore. Finally as mentioned in methodology chapter, to reduce this publication/journal bias even further, Google search was conducted which added 30 papers also included the data extraction. The interesting result was that many of these papers were published in ACM guide and IEEE computer Society. This shows that there has been some journal bias which was covered by the Google search.

4.3 Threats to Identification of Primary Studies

Search Strategy used in this systematic review was designed to cover as many papers as possible related to Human Factors in Software Engineering. It was decided to use IEEEXplore, ACM Digital Library and Compendex as the main sources of this systematic review. These three databases are among the most well known and common databases used in similar areas and a systematic review on SPI[5]. Also these data bases were discussed with other researchers. As explained in section 3.1 and table 3 to reach this goal, the search string was designed very carefully considering all possible aspects of human factors in software engineering. For instance ("Software Development Process" OR "Software Engineering") AND ("Human Factors" OR "Human Issues") AND ("Requirement Engineering" OR "Requirement Elicitation" OR "Requirement Analysis") is the search string used for the human factors in requirement engineering. Therefore similar words to the ones that were used commonly were considered not to miss papers using different terms for the same issue. Also after searching, the abstracts of the papers were carefully read to evaluate the relevance of the paper to the research questions of this systematic review. This step was conducted several times to reduce the investigator bias and judgment when excluding and including papers. After this phase papers were categorized based on their context (product, process, organization and management) by abstract, keywords and title. To focus on the area which was related to the research questions, papers which were only related to product or process were excluded. Although after doing the full text reading there was one paper which was only related to product and was excluded from the data base used for statistical analysis. To decrease the probability of primary study selection bias, the author searched Google using similar search strings which resulted in a high number of papers that were already retrieved from the three source databases used in this systematic review and 30 more papers were added. Another threat to the primary study selection is the limited access to the full text of certain papers retrieved from the databases. There were three papers [49][59][67] that their full text could not be accessed via Chalmers University accessible databases and were ordered by this researcher but didn't succeed therefore were excluded from the primary study database, although they were relevant and related to the research questions of this systematic review. However, as the study was carried out by one person, the investigator bias could have affected this phase. For instance there were some papers that were considered in the primary study data base but after the second round the abstract reading phase were excluded. If several researchers had been involved, the paper selection could be compared and discussed leading to a higher precision and validity of the search. To summarize, the author has done her best to cover

the interesting papers for this systematic review but some primary study selection threats still exist and can be improved by reviews with other researchers and systematic reviewers.

4.4 Threats to Data extraction and its Results Consistency

As suggested by Kitchenham [7,12] to have a more efficient and effective systematic review, a review protocol is necessary. As explained in section 3.2 of chapter 3, the inclusion and exclusion of papers based on the selection criteria (journal papers, English papers, human factors and cognitive related papers, non HCI related papers, etc) resulted in a consistent database used in data extraction phase. This protocol and its criteria were based on discussions with two software engineering researchers. To increase the consistency in the data extraction phase a data extraction form was designed and evaluated in a pilot extraction of 10 papers randomly chosen from the primary studies database. Based on the results of the pilot review the form was modified. This was also checked by two other researchers. For instance by the beginning of the extraction no details for software development phases were considered (e.g. maintenance or requirement engineering categories, etc). After conducting the pilot extraction and reviewing the results it was decided to use the SWEBOK for both Software Development Process and Software Engineering Management. This increased the reliability and consistency of the extraction phase. During the data extraction phase, this form was used and in confusion cases regarding the data extraction it was referred to and checked with the supervisor. However, this phase is also affected by investigator bias that could be reduced if there were several researchers working together. Although this (group work) is a very reliable way of increasing the validity of this phase, researchers try to pilot the selection and extraction and iteratively improve the extraction which is what has been done in this systematic review as well. This strategy has been proposed by Brereton et. al[29] to increase the efficiency, reliability and validity of these phases while considering the time/scheduling limitations.

5 DISCUSSION AND CONCLUSION

This chapter gives a brief discussion on major findings of this thesis. The results suggest some areas for future primary and secondary research studies. Finally some highlights of the research need for special fields of human factors in software development have been suggested.

5.1 Summary of Results

As software development process is a human dependent activity the human factors have great impact on its performance, success and failure. Main contribution of this thesis project was to investigate, identify and characterize these human factors and their impact on development process from different perspectives. Development Phases, Management Phases, Human Roles and Level of Human Factors were the main categories that this systematic review considered while covering 67 journal papers published in 2000-2010.

The retrieved papers were divided into four main categories of product, process, organization and management (section 3.1.1). Based on the research questions, the last two categories were focused on. Among 67 papers there were many papers overlapping with process (42 papers, 62%) and only 6 papers (8%) overlapping with product. The most focused category of human factors by researchers was management including 58 papers (86%). 46 papers (68%) were related to the Organizational context while 42 papers (62%) were assigned to product category and only 6 papers (8%) to the product category. The finding regarding the Study Context property (P3) is that Management followed by Organization are the most common context in the area of Human Factors in SE. Moreover the result shows that most of the research in human factors in software engineering is concerned with managerial issues addressed by the second research question of this systematic review.

Level of human factors was the sixth property (P6) defined to be investigated among the papers this systematic review covers. This was set to identify the researchers' perspective when they investigate human factors in software development and engineering .In other words this identifies the level of human factors impact on the development process. Research papers in this systematic review have discussed different levels of human impact on development and management phases. The highest level of human factors focus among these papers was "Individual" level addressed by 76% papers followed by "Organizational" level. (70%) Consequently the least discussed level of human factors is "Interpersonal" covered by 38 papers (56%). Although the effects of interpersonal relationships and its effect on development efficiency has been proved specifically on new development approaches (agile development process), the individual level of human factor has been of more interest for most researchers. As the agile development process implementation grows in traditional/preplanned development

organizations, the level of human factor focus will be converged to interpersonal relations and issues. Thus, it's expected that the interpersonal aspect of human factors gains more attention by researchers and practitioners in future.

Another interesting property of these papers covered in this project was the publication year distribution (P7). Among 67 papers, 28 papers (41%) were published the first 5 years (2000-2004) and 37 papers (55%) were published the last 6 years (2005-2010).
The highest number of papers (9) was published in 2002 and the least (2) was published in 2006. However the increment of the publications over the last six years shows some improvement in this research area.

Data extracted from the selected papers was mapped to SWEBOK [125]. According to the first research question (P1), papers were classified under software development phases (Requirement Engineering, Design, Implementation, Test and Maintenance). Among software development lifecycle phases the most common one in human factors studies is "Requirement Engineering" addressed by 67% of the inspected papers followed by "Design" (52%). The least discussed development phase is "Maintenance" covered only by 21 papers (31%) which highlights the need for more research in this area. Moreover, "Implementation" was covered by 31 papers (46%) and "Test" was addressed by 25 papers (37%).

The second research question was designed to investigate human factors involved in the software project management process (P2). According to SWEBOK (Appendix C), this process includes 5 main phases(Initiation and Scope Definition, Software Project Planning, Software Project Enactment, Review and Evaluation, Closure and Software Engineering Measurement).Among them "Software Project Enactment" is the most common software engineering management phase from human factors perspective (67 papers, 100%) followed by "Initiation and Scope Definition and Software Project Planning" (89%) and "Software Engineering Measurements" (56%). The least discussed management phase was "Closure" indicated only by one paper (1%) followed by "Review and Evaluation" addressed by 8 papers (11%).

Research methodology used in the papers published in the area of Human Factors in Software Development was another research question (RQ3, P4) of this SLR. Results of data extraction regarding the research methods show that Empirical research (55 papers, 82%) is the most common. Among the Empirical papers (55 papers), Case Study (43 papers, 78%) was the most common type of research method. The second most common method was Design of system/solution/model/method (36 papers, 65%). Survey was used by 26 research papers (47%). The least used research method is Industrial report (0 papers) followed by Controlled experiments (9 papers). However, among 67 papers only 17 papers (25%) used theoretical research method which is very low in comparison with the amount of empirical papers.

Moreover, Human Role was the fourth research question (RQ4, P5) in this systematic review. It was of interest to know which human roles have been addressed by researchers in SE area. These roles were of three types of Customer, Developer and Manager. Results of this systematic review, reveal that the most discussed human role by researchers is Developer (65 papers, 94%) and the least focused role is Customer (23 papers, 34%) which is in contrast with its effect in software business industry. One main reason to this result can be the exclusion of Human Computer Interaction/User Centered design papers which mainly focus on user/customer behavior and interface design. Systematic review results regarding the development phases show that early and middle development phases (requirement engineering, design and implementation) have been more addressed by researchers from human factors perspective. The main reason for this conclusion is the higher level of human participation in early development phases. For instance in requirement engineering related papers, managers, developers and customers have been addressed by great amount of papers while in test phase few papers have addressed customers or managers. However it doesn't indicate that human factors affect these phases less, but they have been overlooked by many researchers. For instance human factors such as customer support and satisfaction which are among factors affecting maintenance phase have been almost neglected by researchers. Manager role was addressed by 45 papers (58%) with a high variety of management tasks such as project management, function management, Human Resource management, product management and etc. Among these different types of management, Project Management was the most common one addressed by 11 papers followed by Organization and Team Management (5 papers).

Another goal set for this project was to assess the searchability of the papers published in the area of Human Factors in Software Development. This was carried out to examine other researcher's possibility when searching primary and secondary studies in the field. As already explained, to achieve that five assessment questions were defined to assess the title, abstract and keyword searchability of the papers reviewed in this SLR. The answers to each of these questions could be yes, partially and no. Results of the searchability assessment of papers reveal that the negative marks are more than the positive ones. The most negative marks (42 papers, 62%) are related to the fourth and fifth assessment question assessing the titles searchability. However the question with the most positive marks (24 papers, 35%) was the second research question assessing the searchability of abstracts regarding the topics. This shows that even the amount of most positive marks is a very low number. The results for the first research question regarding the author keywords include 10 positive and 15 negative and partially marks. The second research question assessing the searchability of the abstracts had 24 positive, 27 partially and 16 negative marks. The third research question that was designed to assess the keywords used in abstracts resulted in 13 positive, 29 partially and 25 negative answers. The forth research question resulted in 10 positive, 15 partially and 42 negative replies regarding the title searchability. Finally the last research question assessing the title keywords revealed that 11 papers had relevant keywords in their titles, while 14 papers partially did and 42 papers didn't.

5.2 Discussion

As it has been addressed by Kitchenham [7] some of the main goals of SLR are to:

- Summarize the existing evidence regarding a technology or a specific method
- Clarify the gaps of the research field and to give suggestion for suture studies
- To give a framework or background for future related research projects

Moreover as defined in the "Introduction" chapter the purpose of this SLR was to find the answers to the following research questions.

- What areas of human factors in software development have been addressed?
- Which phases of software development have been addressed in SE human factors research? The most and the least studied phases and what is been missing?
- Which software management phases have been addressed in SE human factors research? The most and the least studied phases and what is been missing?
- What kind of research methods/tools (empirical, case study, survey, etc) have been used in this research area? What types of papers are published in the area of research and what was their approach to the topic?
- What is the human role in these papers? Developer/manager/user, customer

Additionally it was set to find the level of human factors focus, research publication distribution and types of study context in the area of Human Factors in Software Development.
This SLR addressed the goals defined by Kitchenham, by giving a snapshot of the current status of Human factors in Software Engineering field covering 67 journal papers published during the last ten years. The project contributed in finding the answers to the research questions defined and revealed the gaps of research in this area as summarized in previous section (Summary of Results). Hopefully the revealed results can contribute in highlighting the areas that lack primary studies and invite researchers to conduct more studies on specified fields.

Despite of the human factors great impact on process success, failure, performance and quality, lack of primary and secondary studies on human factors related to software development process reveals the fact that software engineering studies still focus on technical context. Most researchers in this area overlook the human factors that are the center of development process.

Most of the topics that have attracted researchers in software engineering are related to technical software engineering, programming languages, architectural design, software reuse, etc. Subsequently there is a very small proportion of the papers that have mentioned and discussed human factors. The main reason behind this fact is that software development is an everyday updating process which needs numerous papers introducing these new technologies, their practical issues and challenges. This leads most researchers to investigate, introduce and discuss the technical side of software development process, leaving the human side of the area aside. For instance motivation of software developers is a factor that affects the development process

efficiency and group performance. Moreover cognitive issues (learning and thinking styles, abstract/reflective thinking, etc.) psychological issues (mentality patterns, personality and characteristics, their match with different development tasks, etc.) managerial issues (people management, risk management, feasibility study, etc.) have great impact on software development process.

This systematic literature review introduces the human factors that have been addressed by several researchers in the area of Software Engineering. However, the interesting finding is their impact on the process. In this systematic review we look at human factors from different perspectives such as human related Organizational and Managerial factors, Human Role's impact on the process, their level of depth (Individual, Interpersonal and Organizational). Moreover, these factors were classified under subcategories such as psychological and cognitive issues, collaboration and team working, cultural diversity, management issues, etc. All papers discussing these issues, investigated the impact/potential effects of these human factors on the development process. For instance, the systematic review by [74] discussed the impact of software engineers' motivation on their development process efficiency, success, quality, etc. Moreover, thinking and learning styles of developers/customers have great impact on different activities in the project. For instance the thinking styles and personal characteristics (confidence, optimism, judgment, etc) of developers and manager has a great impact on their planning regarding the resource allocation, cost estimation, scheduling and requirement studies. Additionally management skills such as steering, communication, staff relationship management, confidence and management strategies affect the process success level. Although development process is very technical but existence of good management strategies and communication has great impact on its success rate. For instance as the development process is transferring from traditional to agile methods, management, steering and communication will play major roles in the process success. Agile development process is a group dependent method where all members collaborate to deliver the desired outcome. In such environment, a good management strategy, steering skills and feedback is crucial for the process to succeed otherwise.

However according to the results of this SLR, early phases of development process and software project management have gained more attention than later phases. Among these phases the most discussed level of human factors impact was revealed to be individual level which is also connected to the human roles in the development process. Developers have the major role in the process and researchers are more interested to investigate their individual impact on the process such as their learning skills, cultural impacts, motivation, performance and etc. Another factor that needs more attention is the research methods used in primary studies. As it was explained in section 3.3 the most common research method is case study which has been conducted mostly in industrial context (74%). However, it's also interesting to study software development human factors in academic context such as human factors related courses in software engineering. Moreover, the problems in identifying research methods in papers that did not explicitly clarify their methods is challenging, specially for researchers who want to conduct systematic reviews, and mappings in this area. The results invite researchers to consider clarifying their research

methods in papers to reduce the difficulty of systematic reviewers and other researchers referring to their papers.

According to this systematic review, only 70 journal papers were published during last ten years in the area of this SLR interest which is a very low amount. The low amount of relevant primary studies affects the systematic reviews conducted in the area. This is due to the direct relationship between the number of primary studies and the need for the systematic reviews, as when the amount of primary study becomes high; the need for conducting systematic review becomes more. Therefore the need for systematic review might have been overlooked, despite of its great impact. As the number of systematic reviews increases, the lack of the primary studies will be highlighted, hopefully leading to improvement and increment of research. Based on the findings of this systematic review, the number of primary studies has been increased during last three years which is promising for both primary and secondary study researchers.

5.3 Conclusion

Despite of the human factors impact on software development process and its level of success, failure and performance, there has not been enough focus by primary study researchers on the area. Human factors have been overlooked in late phases of development process as well as software engineering management. Although both late development process and project management have major effect on development process success and improvement by means of improving the maintenance and test accordingly customer satisfaction, improving the process according to feedback either by managers and developers or customers. Closure of the projects is another management phase which lacks attention and more research. The more research is conducted on this phase, the more improvement will appear in other software development projects. This is a result of the documentation of software project failures and success and their lessons learnt.

Empirical research papers in this area mostly use case studies as their tool in conducting their projects. However there are many other useful empirical research methods in SE field such as survey, industrial report and other types of qualitative and quantitative studies. The reason is that although case studies are very useful methods but they mostly help to investigate a single entity or phenomenon in a specific period of time. [17] For giving more general and applicable research project in the area we need more use of other types of empirical and theoretical research methods.

In contrast with all other areas of business which their highest concern is the customer, among the inspected papers this role has been overlooked by researchers. Customer/user participation in development process has various advantages leading to a better market for the software product, improving the CRM while helping developers to know what customers are looking for. For instance the more users participate in requirement engineering and specification phase the better results will be gained in acceptance testing and later providing better market for their software products.

Human factors have different levels of impact on development process and the most discussed ones are Individual. However, a transferring from preplanned to agile development process, there will be more "Interpersonal" level of human factors involved which has not been addressed well by researchers so far. As agile development process becomes more common in traditional software development organizations, interpersonal aspect of human factors will be more focused. This is due to the human-centered nature of agile software development and great impact of team work and communication in its success. Software development process is transferring from an individual based activity to a more team/group oriented process where the success and performance of the process is dependent to the group work and communication skills of the group. Consequently it is expected to have more research projects investigating the interpersonal aspect of the human factors affecting the process.

Management and Organization were the main perspectives researchers had when discussing the human factors in development process. Product is the least discussed perspective as expected. This is due to the research questions defined for this systematic review which basically didn't consider the product aspect of software and was more interested in its development process.

Furthermore, as this SLR reveals that there is a significant improvement of publications in this research field, it highlights the recent attention this field has gained by researchers in academia and industry. This also encourages more researchers to conduct research project where there is a significant gap in the field as specified by this systematic review.

Human factors study in software development process has been partially overlooked by both primary study researchers. Among papers that have been published during last ten years, most do not have searchable title/keywords/abstracts which increase the challenges for systematic reviews. Also, there are many papers in the area that do not explicitly mention their research method which further increases the systematic review related challenges. This systematic review results invite researchers to consider choosing clearer, more searchable keywords, titles and writing more searchable abstracts.

Another factor that can be interpreted by reviewing these results is the validity threats in systematic reviews being conducted in similar areas. As these results show high number of papers that have not used searchable keywords in their abstracts, titles and the base of the first search in systematic review is on keywords/abstracts/titles search, there is a probability of missing relevant papers of the area due to their searchability issues.

Furthermore, this systematic review highlights the need of additional systematic reviews by several researchers. This increases the validity and quality of the systematic review as indicated in the validity threats chapter. These findings will hopefully help to identify areas for future primary and secondary research search.

5.4 Future work

The results of this systematic review, highlights research areas for further primary and secondary studies and research. These areas can improve the understanding of human factors impacts and

roles in software development process. The main areas this systematic review suggests for more studies have been summarized by following.

- Investigation and study of human factors related to later phases of software development process (test, maintenance)
- Studies of late phases of software engineering management and their connection with human factors (Review and Evaluation, Closure and software engineering measurements)
- Conducting more research on human factors in software development in academic and education related context
- Investigating more interpersonal issues involved in software development process and software engineering management
- And finally conducting more systematic reviews of general type related to human factors in software development for highlighting the research needs in the area.

REFERENCES

[1] N. Wirth, “A brief history of software engineering,” IEEE Annals of the History of Computing, vol. 30, no. 3, pp. 32–39, 2008.

[2] D. I. K. Sjoberg, T. Dyba, and M. Jorgensen, “The future of empirical methods in software engineering research,” in International Conference on Software Engineering, 2007, pp. 358–378.

[3] M. Shaw, “Prospects for an engineering discipline of software, IEEE Software, vol. 7, no. 6, pp. 15–24, 1990..

[4] W. Scacchi, “Process models in software engineering,” Encyclopedia of Software Engineering, pp. 993–1005, 2001.

[5] M.Unterkalmsteiner, T. Gorschek, A.K.M. Moinul Islam, C.K. Cheng, R. B.Permadi, R. Feldt, “Evaluation and Measurement of Software Process Improvement - A Systematic Literature Review”, 2010

[6] K.Petersen, R.Feldt, S.Mujtaba, M. Mattsson, “Systematic Mapping Studies in Software Engineering”, 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), June 2008

[7] B.Kitchenham, “Procedures for Performing Systematic Reviews”, Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia, 2004

[8] Dybå, T., Kampenes, V. B. & Sjøberg, D. I. K. (2006), ‘A systematic review of statistical power in software engineering experiments’, Information & Software Technology 48(8), 745–755.

[9] Hannay, J. E., Sjøberg, D. I. K. & Dybå, T. (2007), “A systematic review of theory use in software engineering experiments”, IEEE Trans. Software Eng. 33(2), 87–107.

[10] Kampenes, V. B., Dybå, T., Hannay, J. E. & Sjøberg, D. I. K. (2007), ‘Systematic review: A systematic review of effect size in software engineering experiments’, Information & Software Technology 49(11-12), 1073–1086.

[11] Australian National Health and Medical Research Council. How to review the evidence: systematic identification and review of the scientific literature, 2000.
IBSN 186-4960329.

[12] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” Software Engineering Group, Keele University and Department of Computer Science, University of Durham, UK, Technical Report EBSE-2007- 01, July 2007.

[13] J. C. de Almeida Biolchini, P. G. Mian, A. C. C. Natali, T. U. Conte, and G. H. Travassos, “Scientific research ontology to support systematic review in software engineering,” Advanced Engineering Informatics, vol. 21, no. 2, pp. 133–151, Apr. 2007.

[14] M. Staples and M. Niazi, “Experiences using systematic review guidelines,” Journal of Systems and Software, vol. 80, no. 9, pp. 1425–1437, Sep. 2007.

[15] Royce, W.W, “Managing the development of large software systems: Concepts and Techniques”, IEEE WESTCON, Los Angeles CA:1-9, 1970
[16] Ian Sommerville , Software Engineering, 9th ed, International Edition, Boston, Published by Addison-Wesley, March 2010, ISBN-10: 0-13-703515-2, ISBN-13: 978-0-13-703515-1, Chapter 2,Software Processes; pp 27-55
[17] C. Wohlin, P. Runeson, M. Höst, M.C.Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering: An Introduction, p.cm.- (The Kluwer international series in software engineering), Foreword by A.V. Mayrhauser, Kluwer Academic Publishers, Lund University, Sweden, 2000, ISBN 0-7923-8682-5
[18] F. Paetsch, A. Eberlein, and F. Maurer, “Requirements Engineering and Agile Software Development,” Proc. 12th IEEE Int’l Workshops Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 03), IEEE CS Press, 2003, pp. 308–313.
[19] G. Kotonya, I. Sommerville, "Integrating safety analysis and requirements engineering," apsec, pp.259, Fourth Asia-Pacific Software Engineering and International Computer Science Conference (APSEC'97 / ICSC'97)
[20] B.Hughes, M. Cottere, “Software Project Management”, Third Edition, Tata McGraw-Hill,
Or Mike Cotterell, Bob Hughes, Software project management, International Thomson Computer Press, Boston, MA, 1995
[21] H.V. Vliet, Software Engineering: Principles and Practice, 3rd edition, John Wiley & Sons, 2008
[22] R.K. Yin, Case Study Research Design and Methods, Third Edition, p.cm.-(Applied Social Research Methods Series, vol. 5), USA, 2003, ISBN 0-7619-2553-8, ISBN 0-7619-2552-X,
[23] Schramm, W., Notes on case studies of instructional media projects, Working paper for the academy for Educational Development, Washington, DC, December 1971
[24] S. Easterbrook, J. Singer, M. Storey, and D. Damian, “Selecting empirical methods for software engineering research,” in Guide to Advanced Empirical Software Engineering, 2008, pp. 285–311.
[25] M. V. Zelkowitz and D. Wallace, “Experimental validation in software engineering,” Information and Software Technology, vol. 39, no. 11, pp. 735–743, 1997.
[26] S. L. Pfleeger and B. Kitchenham, “Principles of survey research: part 1: turning lemons into lemonade,” ACM SIGSOFT Software Engineering Notes, vol. 26, no. 6, pp. 16–18, 2001.
[27] C. Seaman, “Qualitative methods in empirical studies of software engineering,” IEEE Transactions on Software Engineering, vol. 25, no. 4, pp. 557–72, Jul. 1999.
[28] S.Pfleeger, Software Engineering: Theory and Practice, Prentice-hall, 1998
[29] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” Journal of Systems and Software, vol. 80, no. 4, pp. 571–583, Apr. 2007
[29] Phillip G. Armour, “The spiritual life of projects”, Communications of the ACM, Vol. 45, Issue 1, pp.11 – 14, 2002

Systematic Review References

[30] B.Bohem, R.Tuner, “Management Challenges to Implementing Agile Processes in Traditional Development Organizations”, IEEE Software, vol. 22 , Issue 5, pp. 30 - 39 , September 2005

[31] R. Feldt, L. Angelis, R. Torkar, M. Samuelsson, “Links Between the Personalities, Views, and Attitudes of Software Engineers”, Information and Software Technology, vol. 52 , Issue 6, pp. 611-624, June 2010

[32] Luiz Fernando Capretz, Faheem Ahmed, "Making Sense of Software Development and Personality Types," *IT Professional*, vol. 12, no. 1, pp. 6-13, Jan./Feb. 2010

[33] S.D. Teasley, L.A. Covi, M.S. Krishnan and J.S. Olson, “Rapid Software Development through Team Collocation”, IEEE Transactions on Software Engineering, vol. 28, no. 7, pp.671-683, July 2002

[34] Hale, J.; Parrish, A.; Dixon, B.; Smith, R.K., “Enhancing the COCOMO Estimation Models”, Software, IEEE, vol. 17, Issue 6, pp. 45 – 49, Novermber2000

[35] Dittrich,Y.; Vaucouleur, S.; Giff, S.; “ERP Customization as Software Engineering: Knowledge Sharing and Cooperation”, Software, IEEE , vol. 26, Issue: 6 ,pp. 41 – 47, November 2009

[36] U.Asklund, L.Bendix, “A study of configuration management in open source software projects”, IEEE proceedings, Software, vol. 149, no.1, February 2002

[37]Damian, D.E.H.; Eberlein, A.; Shaw, M.L.G.; Gaines, B.R.; “Using different communication media inrequirements negotiation”, Software, IEEE, vol.17, Issue: 3, pp. 28 – 36, June 2000

[38] J.Li, R.Conradi, O.P.N. Slyngstad, M.Morisio, “A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components”, IEEE Transactions on Software Engineering, vol. 34, no.2, March/April 2008

[39] M. Morisio, M. Ezran, C. Tully, “Success and Failure Factors in Software Reuse”, IEEE Transactions on Software Engineering, vol. 28 , Issue 4, pp. 340 - 357 , April 2002

[40] K.SIAU, X.TAN, “Technical Communication in Information Systems”, IEEE Transactions on Professional Communication, vol. 48, no. 3, pp.269-284, September 2005

[41] M. Morisio, “Applying the PSP in Industry”, IEEE Software, vol. 17, Issue 6, pp. 90 – 95, November 2000

[42] Delanghe, S, “Using Learning Styles in Software Documentation”, IEEE Transactions on Professional Communication, vol. 43, Issue 2, pp. 201-205, June 2000

[43] Hazzan, O. ; Tomayko, J.E. ; , “Reflection and Abstraction in Learning Software Engineering’s Human Aspects”, IEEE Computer, vol. 38, Issue 6, pp. 39-45, May 2005

[44] Parsons, J.; Saunders, C., “Cognitive Heuristics in Software Engineering: Applying and Extending Anchoring and Adjustment to Artifact Reuse”, IEEE Transactions on Software Engineering, vol. 30, Issue 12, pp.873-888, January 2005

[45] Martin, R.C., “Extreme programming development through dialog”, Software, IEEE vol.17,pp.12-13,Jul/Aug2000
[46] Zachary, Wayne; Hoffman, Robert R.; Neville, Kelly; Fowlkes, Jennifer; , “Human Total Cost of Ownership: The Penny Foolish Principle at Work”, IEEE Intelligent Systems, vol. 22, Issue 2, pp. 88-92, March 2007
[47] Dingsoyr, T.; Moe, N.B.;” The Impact of Employee Participation on the Use of an Electronic Process Guide: A Longitudinal Case Study”, IEEE Transactions on Software Engineering, vol. 34, Issue: 2, pp. 212-225, March 2008
[48] Patricia A. Carlson, “Information technology and the emergence of a work-centered organization”, ACM Journal of Computer Documentation (JCD), vol. 24 , Issue 4 , pp. 204 - 212 , November 2000
[49] J.F. Haroon, E.A.Konijn,H.V.Vliet, G.V.D Veer , “Requirements Change: Fears Dictate the Must Haves; Desires the Won’t Haves”, Journal of Systems and Software, vol. 80, Issue 3, pp.328-355, March 2007
[50] M. Kauppinen, M. Vartiainen, J. Kontio, S.Kujala, R. Sulonen, “Implementing Requirements engineering processes throughout organizations: success factors and challenges”, Information and Software Technology 46(14), pp. 937–953, April 2004 available online at: <www.sciencedirect.com >
[51] A. May, C. Carter, “A case study of virtual team working in the European automotive industry”, International Journal of Industrial Ergonomics 27, pp. 171–186, October 2000, available online at: <www.sciencedirect.com >
[52] W. Keogh,S. Cooper, “The Identification and application of knowledge capital within small firms”, *Journal of Small Business and Enterprise Development*, pp. 76, 2005
[53] R. Subramanyam, F.L. Wiesstein, M.S. Krishnan, “User participation in Software Development Projects”, Communications of the ACM, vol. 53 , Issue 3, pp. 137-141, March 2010
[54] D.Mishra, A.Mishra, “Effective Communication in Collaboration and Coordination in Extreme Programming: Human-Centric Perspective in a Small Organization”, Human Factors and Ergonomics in Manufacturing, vol. 19, Issue 5,pp. 438–456, 2009
[55] V. Wickramasinghe, S. Jayabandu, “Towards workplace flexibility: flexitime arrangements in Sri Lanka”, Employee Relations, vol. 29 no. 6, pp. 554-575, 2007
[56] H. SHARP, M. Woodman, F. HOVENDEN, “Using Metaphor to Analyse Qualitative Data: Vulcans and Humans in Software Development”, Empirical Software Engineering, vol. 10, pp. 343–365, 2005
[57] D.E. Damian and D. Zowghi, “The impact of stakeholders’ geographical distribution on managing requirements in a multi-site organization”, IEEE computer society, Proceedings of the IEEE Joint International Conference on Requirements Engineering, 2002

[58] Tsai H.-T., Moskowitz H., Lee L.-H., “Human resource selection for software development projects using Taguchi’s parameter design”, European Journal of Operational Research, vol.151, Issue 1, pp. 167-180, November 2003
[59] Z. Irani, “Empirical testing of an information system evaluation framework”, International Journal of Information Technology and Management, vol. 1 , Issue 2-3, pp. 298-323, July 2002
[60] Charlie C. Chen, Chuck C. H. Law, and Samuel C. Yang, Managing ERP Implementation Failure: A Project Management Perspective, IEEE Transactions on Software Engineering Management, vol. 56, no. 1, February 2009
[61] Michael C. Georgiadis, Aaron A. Levis, Panagiotis Tsiakis, Ioannis Sanidiotis, Constantinos C. Pantelides, Lazaros G. Papageorgiou, “Optimisation-based scheduling: A discrete manufacturing, case study”, Computers & Industrial Engineering , vol. 49, pp. 118–145, 2005
[62] A. Gupta, “Enterprise Resource Planning: The emerging Organizational System Values”, Industrial Management and Data Systems, vol.100, Issue 3, pp. 114-118, 2000
[63] M. JØRGENSEN, “Regression Models of Software Development Effort
Estimation Accuracy and Bias”, Empirical Software Engineering, vol. 9, pp. 297–314, 2004
[64] J. Evermann, “An Exploratory Study of Database Integration Processes”, IEEE Transactions on Knowledge and Data Engineering, vol.20, no.1,pp. 99-115, January 2008
[65] B.WONG, “Understanding Stakeholder Values as a Means
of Dealing with Stakeholder Conflicts”, Software Quality Journal, vol. 13, pp. 429–445, 2005
[66] T. Koc, “Organizational determinants of innovation capacity in software companies”, Computers and Industrial Engineering, vol. 53, Issue 3, pp. 373-385, October 2007
[67] M.J. Ashworth , K. M. Carley, “Who you know vs. what you know: The impact of social position and knowledge on team performance”, The Journal of Mathematical Sociology, vol. 30, Issue 1, pp 43 – 75, January 2006
[68] G. Koutsoukos, “Mentality patterns: recurring turns of mind as first-class concerns in software projects”, Int. J. Technology, Policy and Management, vol. 8, no. 1 pp. 5-21, 2008
[69] Acuna, S.T. Juristo, N. Moreno, A.M. “Emphasizing Human Capabilities in Software Development”, IEEE Software, vol. 23, Issue 2, pp. 94-101, March/April 2006
[70] A.Boden, B.Nett, V.Wulf, "Operational and Strategic Learning in Global Software Development - Implications from two Offshoring Case Studies in Small Enterprises," *IEEE Software*, 05 Aug. 2009. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/MS.2009.113>
[71] S.M. Yacoub, H.H. Ammar, “A Methodology for Architecture-Level Reliability Risk Analysis”, IEEE Transactions on Software Engineering, vol. 28, no. 6, pp. 529-547, June 2008
[72] P. Barthelmess, K. M. Anderson, “A view of software development environments based on activity theory”, Computer Supported Cooperative Work, vol.11 , Issue 1-2, pp. 13 – 37, 2002
[73] M. Jørgensen, M. Shepperd, “A Systematic Review of Software Development Cost Estimation Studies”, IEEE Transactions on Software Engineering, vol. 33, Issue 1, pp. 33-53, 2007

[74] T. HALL, N.BADDOO, S.BEECHAM, H.ROBINSON, H.SHARP, “A Systematic Review of Theory Use in Studies Investigating the Motivations of Software Engineers”, ACM Transactions on Software Engineering and Methodology, vol. 18,no. 3, Article 10, May 2009.
[75] A.Cockburn, J.Highsmith, “Agile Software Development: The People Factor”, Computer, vol. 34 no. 11, pp. 131-133, November 2001
[76] N.DESAI, A.K. CHOPRA, M. SINGH, “Amoeba: A Methodology for Modeling and Evolving Cross-Organizational Business Processes”, ACM Transactions on Software Engineering and Methodology, vol. 19, no. 2, Article 6, October 2009
[77] Lutz Prechelt, Barbara Unger, "An Experiment Measuring the Effects of Personal Software Process (PSP) Training," IEEE Transactions on Software Engineering, vol. 27, no. 5, pp. 465-472, May 2001
[78] N. Iivari, “Representing the User' in software development-a cultural analysis of usability work in the product development context”, Interacting with Computers, vol.18, no.4, pp.635-664, July 2006
[79] G. Alkadi, T.Beaubouef, “An experimental group for researching and developing software” ,Journal of Computing Sciences in Colleges, Volume 23 , Issue 6, pp. 104-109, June 2008
[80] S.Nerur, R.Mahapatra, G.Mangalaraj, “Challenges of migrating to agile methodologies: Organizations must carefully assess their readiness. before treading the path of agility”, Communication of The ACM, vol. 48, no. 5, pp. 72-78, May 2005
[81] T.L. Lewis, W.J. Smith, “Creating high performing software engineering teams: the impact of problem solving style dominance on group conflict and performance”, Journal of Computing Sciences in Colleges, vol. 24 , Issue 2, pp. 121-129, December 2008
[82] G.MacCalla, “Software engineering requires individual professionalism”, Communications of the ACM, vol 45, Issue 11, no.11, pp. 98 – 101, November 2002
[83] M.Lycett, R.D. Macredie, C.Patel, R.J. Paul, "Migrating Agile Methods to Standardized Development Practice," Computer, vol. 36, no. 6, pp. 79-85, June 2003
[84] B.Kitchenham, S.Linkman, P.W. Jones , “Modeling Software Bidding Risks”, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. 29, no. 6, June 2003
[85] M.Jørgensen, S.Grimstad , “Avoiding Irrelevant and Misleading Information
When Estimating Development Effort”, IEEE Software, vol.25, Issue 3, pp. 78 – 83, May/June 2008
[86] Choi, N; Chengalur-Smith, I; Whitmore, A; Software, IEEE, IEEE Computer Society, pp. 1: issue: 99, ISSN: 0740-7459, DOI: 10.1109/MS.2010.26, January 2010
[87] T. Dybå, E.Arisholm, D.I.K. Sjøberg, J.E. Hannay, F.Shull , “Are Two Heads Better than One? On the Effectiveness of Pair Programming”, IEEE Software, vol. 24, Issue 6, pp. 12-15, 2007
[88] M. Cohn, D. Ford, "Introducing an Agile Process to an Organization," Computer, vol. 36, no. 6, pp. 74-78, June 2003

[89] H.C.Benestad, B.Anda, E.Arisholm1, “Are We More Productive Now? Analyzing Change Tasks to Assess Productivity Trends During Software Evolution”, Evaluation of Novel Approaches to Software Engineering (ENASE), 2009
[90] M. Jørgensen, K. H. Teigen, and K. J. Moløkken-Østvold, “Better sure than safe? Overconfidence in judgment based software development effort prediction intervals”, Journal of Systems and Software, vol. 70, Issue 1-2, pp.79-93, 2004
[91] M.Jørgensen, "How to Avoid Selecting Bids Based on Overoptimistic Cost Estimates," IEEE Software, vol. 26, no. 3, pp. 79-84, May/June 2009
[92] M.Jørgensen, “Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates”, Information and Software Technology, vol.52 , Issue 5, pp. 506-516, May 2010
[93] M. Jørgensen, D.I.K. Sjøberg, “Impact of effort estimates on software project work” , Information and Software Technology
vol. 43, Issue 15, pp. 939-948, December 2001
[94] K. Børte, and M. Nerland, “software effort estimation as collective accomplishement: an analysis of estimation practice in a multi-specialist team”, 2009
[95] M. Jørgensen, S. Grimstad, “Software Development Estimation Biases: The Role of Interdependence”, IEEE Transactions on Software Engineering, 2010
[96] M. Jørgensen, T.Halkjelsvik, “The effects of request formats on judgment-based effort estimation”, Journal of Systems and Software, vol. 83 , Issue 1, pp. 29-36, January 2010
[97] M. Jørgensen, T.M. Gruschke, "The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments," IEEE Transactions on Software Engineering, IEEE computer Society Digital Library. IEEE Computer Society, January 2009, Available at: http://doi.ieeecomputersociety.org/10.1109/TSE.2009.2
[98] T. Halkjelsvik, M. Jørgensen, and K. H. Teigen , “To Read Two Pages, I Need 5 Minutes, but Give Me 5 Minutes and I will Read Four: How to Change Productivity Estimates by Inverting the Question”, APPLIED COGNITIVE PSYCHOLOGY, 2010, DOI: 10.1002/acp.1693, Published online in Wiley InterScience, Available at:
http://www3.interscience.wiley.com/journal/123341967/abstract
[99] K. Moløkken-Østvold, N.C. Haugen , H.C.Benestad , “Using planning poker for combining expert estimates in software projects”, Journal of Systems and Software, vol. 81 , Issue 12, pp. 2106-2117, December 2008
[100] E.J. BARRY, T. MUKHOPADHYAY, S. A. SLAUGHTER, “Software Project Duration and Effort: An Empirical Study”, Information Technology and Management, vol. 3, pp. 113–136, 2002
[101] J.MOSES, “Measuring Effort Estimation Uncertainty
to Improve Client Confidence”, Software Quality Journal, vol. 10, pp. 135–148, 2002
[102] D.E. Damian, D. Zowghi, “Requirements Engineering challenges in multi-site software development Organizations”, Requirements Engineering Journal, 8, pp. 149-160, 2003

[103] J.S Persson, L.Mathiassen, J. Boeg,T. Stenskrog Madsen, F. Steinson, " Managing Risks in Distributed Software Projects:An Integrative Framework", IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 56, NO. 3, pp. 508-532, AUGUST 2009
[104] N.Kasabov, "Evolving Intelligence in Humans and Machines: Integrative Evolving Connectionist Systems Approach", IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, pp. 23-37, AUGUST 2008
[105] G. Steven McMillan and Robert D. Hamilton, "The Impact of Publicly Funded Basic Research: An Integrative Extension of Martin and Salter", IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 50, NO. 2, pp. 184-191, MAY 2003
[106] N. Chonacky, D.Winch, "3MS for Instruction Reviews of MAPLE, MATHEMATICA, and MATLAB", IEEE CS and the AIP, pp. 7-13, MAY/JUNE 2005
[107] T. Dybaˊ, T. Dingsøyr, "Empirical studies of agile software development: A systematic review", Inform.
Softw. Technol. (2008), doi:10.1016/j.infsof.2008.01.006
[108] G. Singh Waliaˊ J.C. Carve, "A systematic literature review to identify and classify software requirement errors", Information and Software Technology, Volume 51, Issue 7, pp.1087-1109, July 2009
[109] King, John Leslie Bergman, Mark Lyytinen, Kalle J., "Large-Scale Requirements Analysis Revisited: The need for Understanding the Political Ecology of Requirements Engineering", 2002, doi: 10.1007/s007660200011
[110] Australian National Health and Medical Research Council. How to use the
evidence: assessment and application of scientific evidence. February 2000,
ISBN 0 642 43295 2
[111] S. Mujtaba, K.Petersen, R.Feldt, M.Mattsson1 "Software Product Line Variability: A Systematic Mapping Study",School of Engineering, Blekinge Institute of Technology, Sweden
[112] VA Office of Information and Technology (2003) Project Management Guide US DEPARTMENT OF VETERANS AFFAIRS. March 3, 2005
[113] E.Babbie, Survey Research Methods, Wadsworth, ISBN 0-524-12672-3,1990
[114] *M. Svahnberg, T. Gorschek, R. Feldt, R. Torkar, S. B. Saleem, Mfiqu Sha* , "A Systematic Review on Strategic Release Planning Models", Information & Software Technology, vol. 52, num. 3, pp. 237-248, 2010
[115] *Smite, C. Wohlin, T. Gorschek and R. Feldt*, "Empirical Evidence in Global Software Engineering: A Systematic Review", Empirical Software Engineering, vol. 15, num. 1, pp. 91-118, February 2010
[116] *W. Afzal, R. Torkar and R. Feldt*, "A Systematic Review of Search-Based Testing for Non-Functional System Properties", Information & Software Technology, vol. 51, num. 6, pp. 957-976, 2009.
[117]Kitchenham, B.A.; Mendes, E.; Travassos, G.H.; , "Cross versus Within-Company Cost Estimation Studies: A Systematic Review," Software Engineering, IEEE Transactions on , vol.33, no.5, pp.316-329, May 2007
[118] Dingsoyr, T.; Bjornson, F.O.; Shull, F.; , "What Do We Know about Knowledge Management? Practical Implications for Software Engineering," Software, IEEE , vol.26, no.3, pp.100-103, May-June 2009

[119] Hannay, J.E.; Sjoberg, D.I.K.; Dyba, T.; , "A Systematic Review of Theory Use in Software Engineering Experiments," Software Engineering, IEEE Transactions on , vol.33, no.2, pp.87-107, Feb. 2007
[120] Budgen, D.; Bailey, J.; Turner, M.; Kitchenham, B.; Brereton, P.; Charters, S.; , "Cross-domain investigation of empirical practices," Software, IET , vol.3, no.5, pp.410-421, October 2009
[121] Villalba, M.T.; Fernandez-Sanz, L.; Martinez, J.J.; , "Empirical support for the generation of domain-oriented quality models," Software, IET , vol.4, no.1, pp.1-14, Feb. 2010
[122] von Wangenheim, C.G.; Hauck, J.C.R.; Zoucas, A.; Salviano, C.F.; McCaffery, F.; Shull, F.; , "Creating Software Process Capability/Maturity Models," Software, IEEE , vol.27, no.4, pp.92-94, July-Aug. 2010
[123] Dyba, T.; Dingsoyr, T.; , "What Do We Know about Agile Software Development?," Software, IEEE , vol.26, no.5, pp.6-9, Sept.-Oct. 2009
[124] Kitchenham, B.; Budgen, D.; Brereton, P.; Turner, M.; Charters, S.; Linkman, S.; , "Large-scale software engineering questions - expert opinion or empirical evidence?," Software, IET , vol.1, no.5, pp.161-171, October 2007
[125] Alain Abran, , James W. Moore, Pierre Bourque, Robert Dupuis, “Guide to the Software Engineering Body of Knowledge”, SWEBOK, A project of the IEEE Computer Society Professional Practices Committee, 2004

APPENDICES

Appendix A

Data Extraction Form

Paper assessed: _ Time for assessment: _____

1. SE Perspective(s):
A, Product
B, Process
C, Organization
D, Management

2. SE phases(s):

A, Requirements Eng:
requirement analysis ----- requirement validation-----

B, Design/Architecture
Architectural designs------ design patterns------- software design methods------

C, Implementation
Constructing for verification?-------Coding---- reuse------

D, Testing/V&V
Test levels---- objectives of testing-----techniques based on the nature of the application-----?

E, maintenance
Categories of maintenance?----- process----- techniques for maintenance: -----

Use of SWEBOK tables for this part for more details:

Table 1 (software requirements), Table 2 (software design), Table 3 (software construction), Table 4 (software testing), Table 5 (software maintenance) in Appendix B

3. Management issues:
Initiation and scope definition & Software project planning:
A, Feasibility study, risk study, RE
B, CE
C, scheduling
D, resource allocation
E, activity planning/configuration management
Software project enactment/execution:
E, people management
F, risk analysis/management
G, People issues / Human factors

H, Review and evaluation:
Feedback

I, Closure

J, Software engineering measurement: performance

K, Management challenges: change management, management skills, etc

L, Other: _____

Use of SWEBOK tables:

Table 1: software engineering management in appendix c

4. Type of research method and tool:
A, empirical ---------
A1, case study:------
Industry case study----------
Non-industry case study----------
Num cases / companies / systems: _____
Other: _____
A2, survey: Number of “cases” / respondents: _____ Other: _____

A3, controlled experiment:---------
A4, industrial report----------

A5, Design of system/solution/model/method - Type of system/solution: _____ Other: _____

B, theoretical

5. Human role:
A, customer/user
B, developer
C, manager: -------
E, other

6. Level of human factor focus:
A: Individual
B: Interpersonal
C: Organizational
D: all
E: unknown

7. Publication date: ----

9. Quality assessment: keyword assessment

Q1. Has the correct author keyword (human factor in software development and similar) been used?
Q2. Is the abstract searchable about the topic?
Q3. Does the abstract include the keywords (Human factors, human aspect, cognitive issues, software psychology, human centered, etc?
Q4. Is the title searchable?
Q5. Does the title include the right keywords?

Appendix B

SWEBOK: Software Development Phases

Table 1, Software Requirements

Category	Sub-class
Requirement Analysis	• **Requirement Classification**
	• **Conceptual Modeling**
	• **Architectural Design and Requirements Allocation**
	• **Requirement Negotiation**
Requirement Validation	• **Requirement Reviews**
	• **Prototyping**
	• **Model Validation**
	• **Acceptance Test**

Table 2, Software Design

Category	Sub-class
Architectural Styles	• **General structure**
	• **Distributed systems**
	• **Interactive systems**
	• **Adaptable systems**
Design Patterns	• **Creational patterns**
	• **Structural patterns**
	• **Behavioral patterns**
Software Design Methods	• **Function-oriented design methods**
	• **Object-oriented design methods**
	• **Data-structure-centered design**
	• **Component-based design**

Table 3, Software Construction

Category	Sub-class
Constructing for verification	• **Communication methods**
	• **Programming languages**
	• **Platforms**
	• **Tools**
Coding	• **Linguistic**
	• **Formal**
	• **Visual**

Reuse	• **Unit testing**
	• **Integration testing**

Table 4, Software testing

Category	**Sub-class**
Test levels	• **Unit testing**
	• **Integration testing**
	• **System testing**
Objectives of testing	• **Acceptance/qualification testing**
	• **Installation testing**
	• **Alpha and beta testing**
	• **Conformance testing/functional testing/correctness testing**
	• **Reliability achievement and evaluation**
	• **Regression testing**
	• **Performance testing**
	• **Stress testing**
	• **Back-to-back testing**
	• **Recovery testing**
	• **Configuration testing**
	• **Usability testing**
	• **Test-driven development**
Techniques based on the nature of the application	• **Object-oriented testing**
	• **Component-based testing**
	• **Web-based testing**
	• **GUI testing**
	• **Testing of concurrent programs**
	• **Protocol conformance testing**
	• **Testing of real time systems**
	• **Testing of safety-critical systems**

Table 5, Software Maintenance

Category	**Sub-class**
Categories of maintenance	• **Corrective maintenance**
	• **Adaptive maintenance**
	• **Perfective maintenance**
	• **Preventive maintenance**

Process	• **Process implementation**
	• **Problem and modification analysis**
	• **Modification implementation**
	• **Maintenance review/acceptance**
	• **Migration**
	• **Software Retirement**
Techniques for maintenance	• **Program Comprehension**
	• **Reengineering**
	• **Reverse engineering**

Appendix C

SWEBOK: Software Engineering Management

Table 1, Software Engineering Management

Category	Sub-class
SEM activities	• **Initiation and scope definition**
	• **Software project planning**
	• **Software project enactment/execution**
	• **Review and evaluation**
	• **Closure**
	• **Software engineering measurement**

Appendix D

Information extracted from Case studies in the Human Factors in Software Engineering

Paper	Case study Related Extracted Information
33	1 company, a Fortune 100 automobile company
34	client server app for 400 users and 16 contractor
47	19 months longitudinal case study at a medium sized software company
50	3 Finish orgs, action research method
51	2 year user centered project European Automotive Manufacturer project, 7 cases/activities
57	grounded theory on case study, seven month at org development site Sydney Australia, jul2001-feb 2002
66	91 small and medium sized software development companies
60	one company: California bases multinational company
36	Case study and survey: 3 open source software projects
38	133 companies in Norway, Italy, Germany
70	2 German software enterprises
40	3 cases
41	1 FIDA ISO 9001 company in Italy
31	Case study and survey: 47 engineering, in 10 Swedish development comp
78	5 cases in three development organizations
58	1 company 2 cases in Taiwan
61	2 cases one company MORRISSA, north Greece
63	49 cases in one medium sized Norwegian web development company, 10 months
68	several SME projects
69	some SME companies
76	one aero space case study and one company: European union Project
82	Canadian experience?
83	1 company, AGCO
85	4 studies
90	4 case studies/controlled experiment
92	4 experiments/ controlled experiment
96	controlled ex: 4 cases
99	controlled or case: one Norwegian SME software company
100	one company between 1989-1994 , 18 projects: survey or case
101	pilot study in Electronic data system Europe, middle east, Asia

102	grounded theory on case study, seven month at org development site Sydney Australia, jul2001-feb 2002

Appendix E
Survey related extracted information

Paper	Survey related information
35	Review video recorded, face to face interviews with managers and leader developers in three consultancies
47	Three rounds interviews, each round 8 people in a medium sized software company
51	40 engineers from 4 countries
52	Semi structured Interview with senior staff, owner manager of 20 software related companies (5 micro, 12 small, 3 medium sized)
66	Two types of interviews, and scales among 91 SME s in the area of CRM, ERP, etc
72	Survey of representative set of Process Centered Software Development Environment
57	Semi-structured interview with 24 people from 4 sites in USA, 3 sites in Australia, 1 site in New zealand, 1 site in Europe. These people have different roles in each site including producer, developer, engineer, customer support manager, software engineers and team leaders
60	Semi-structured and consistent interview, open ended questionnaire with MIS director of California based multinational company. 2 directors of MIS and customer service and six employees
36	Interview with key people in 3 Open Source Software projects (KDE, Mozilla, Linux)
38	Interview with 16 developers of 13 IT companies
39	24 project in 1994 -1997, structured interviews in 19 companies
86	Interview with 6 computer science major students, 3 senior undergraduates and 3 graduates
53	Web based questionnaires, 2 sections, 56

	questions, 47 engineers in 10 different Swedish software companies
91	Self administered questionnaire, 145 questionnaire distributed among 40 companies, 108 employees responses from 30 different software companies were used.
62	Several ERP companies survey for ERP implementation
65	403 subjects from 22 Australian organizations in different areas such as Telecommunication, banking, insurance, etc. Pilot survey questionnaire included 23 users and developers. Real survey included 600 users and developers
68	Interview, project meetings, coffee breaks
69	Interview with software managers of 8 projects in SMEs
91	Interview and discussions with software providers
95	Questionnaires for 347 developers working in 6 outsourcing companies