AN EMPIRICAL STUDY ON SOFTWARE TEST EFFORT ESTIMATION

LAVA PRASAD KAFLE

Kathmandu University, Nepal
E-mail: lkafle@gmail.com

Abstract- It is well known that software development projects tend to be based on over-optimistic cost estimates. Better knowledge about software cost estimation is necessary to improve realism in software development project bids and budgets. In my master thesis, I did a literature review that indicates that many research papers address software cost and effort estimation, but none of the 150 papers I reviewed addressed the software test effort and/or cost estimation. We therefore prepared a set of five research questions to address software test effort estimation, and conducted a case study and collected empirical evidence from software development companies in Nepal. The minimum company size was 30 while the maximum company size was 200. I performed the case study by conducting interviews with a set of structured questionnaires. I compared the results obtained from the case study with the literature review and found that there exists practice for empirical evidence based verification, validation, and testing cost/effort estimations. I also noted that test effort estimation follow the same pattern as software development project estimates. My results show that 1) all the companies prepare separate estimates for test effort, 2) empirical data is commonly used to estimate test effort, and 3) test effort estimation error seems to be closely correlated with development effort estimation error. A company that had estimated total of 3500 man-months had actually spent 4200 man-months implying 700 man-months of effort/cost overruns to complete the project. Another company that projected testing effort of 100 man-hour actually ended up in 120 man-hour at the end of project causing 20 man-hour effort/cost overruns. Therefore, our study indicates that test effort closely follows the development patterns. However, more studies in this area are clearly needed.

Keywords- Software effort estimation, Testing, Cost estimation

I. INTRODUCTION

Better knowledge about software cost estimation is necessary to improve realism in software development project bids and budgets. In my Master Thesis I have investigated software test effort estimation. This is an important area because it is well known that software testing is a major cost component in software development projects, typically 25-50% of the total cost. I evaluated in my Master Thesis by studying the 150 papers that existing papers do not sufficiently address our research goals of empirical testing estimations with the approach I was seeking, so I take refute in case studies. After series of reviews, I selected 50 papers that matched our criterion independently. I also concluded that search keywords/criterion is not a very good or relevant measure to conclude research results because quality of literature lacks generalization in conveying commonly used classification terms.

I intended to see empirical evidence based software effort cost estimation methods in testing as well rather than generic software development, but, the research found no innovative creative estimation models other than regular COCOMO models, function points, Expert-Judgments, and some already set formal models which implies their maturity in academician and industry. So, I considered it practical to rely on our Case Studies conducted across five software development companies in Nepal because there is no other empirical evidence of any research paper citing scenarios concerning Nepal and our research efforts. The papers, publications, research, and studies fail to address the single unified vision for empirical evidence based software effort cost estimation in testing, that this research is committed. The papers Study review is important because it systematically analyzes the vast pool of papers, innovations, and improvements in evidence based estimation of software cost effort in verification, validation, and testing. Evidence Based Software Engineering (EBSE) is scientific research conducted at real industrial setting to gather actual data to analyze the prospects of outcome as a result of the study. Random controlled experiments are also evidence based although they do not necessarily depict the actual practice scenario.

Software Cost/Effort Estimation is a vital activity in software development projects that allows developers, and managers to forecast, predict, and accurately quote the budget, schedule, and manpower effectively to save from overruns or underruns thereby attempting to optimize the crucial factors leading to project success. The effort and cost estimations in software development have evolved since 1950s and continually being researched as seen from papers by Jorgensen et al. Jorgenson et al., and Grimstad et al.

COnstructive Cost MOdel (COCOMO) was first coined in 1981 published Software Engineering Economics book by Barry Boehm that used Line of

Codes (LOC) as basic parameter used in estimating software development efforts or costs.

Agile software development is becoming increasingly popular in past and current decade as claimed by 36 empirical papers.

Verification &Validation Testing cost effort estimations is not new in research or industry or among academicians, but I presume it's not widespread as overall software development projects estimations. Verification is software output conforming to user requirements while Validation is design, codes, and implementation conforming to expected outputs commonly known as V&V which encompasses commonly used term Testing that is an essential activity for all phases and processes of software development. Therefore, V&V Testing also needs to be estimated for cost and effort to save a software development project from overruns and under-runs.

Overall software quality is measured by output evaluation, client feedback, impression, and actual task fulfillment which are improved by implementing V&V Testing correctly using different approaches described in papers and practices made by industries. IEEE Fifth International Conference on Software Testing Verification, and Validation 2012 focused on models, fault localizations, database/GUI testing, constraint solving, search-based testing, web-applications, test evolution, domain-specific testing, white-box techniques, state-based testing, empirical studies, failure analysis, case studies, analysis and validation, test automation, and PHD Symposium, but we could not see any research paper committed to software cost effort estimation on testing. The Paper is organized as described here: Section II. Is "Research Questions, Section III. is "Papers Study Summary", Section IV. provides "Case Study Summary", Section V. illustrates the "Threats to Validity", and Section VI. describes the "Results/Suggestions". We can find latest papers on testing by Bandyopadhyay [4], and Wohlin [5].

II. RESEARCH QUESTIONS

The motivation behind the identification of five research questions below is to fulfill the goal of the thesis to find the conclusive evidence of testing effort cost estimation practices going on in academic scientific research and real software development by industries:

RQ1: Do Companies collect and use empirical data for the purpose of estimation of V&V effort?
Motivation RQ1: Finding about practices of verification, validation, and testing using empirical evidence.
RQ2: What are the current practices in effort and cost estimation?
Motivation RQ2: Finding about overall effort cost estimation scenario.
RQ3: Which papers address verification, validation, and testing estimations?
Motivation RQ3: Finding about research going on for estimation processes for test effort.
RQ4: How can we improve the current trends in estimations of V&V effort?
Motivation RQ4: Finding about measures suggested to improving current effort estimations in testing.
RQ5: What does the quality of experiments, simulations, and current industry data indicate about the evidence based software cost effort estimation in testing?

1. Motivation RQ5: Finding about real insight into overall quality of practice on testing estimations based on evidence based software engineering. I used the following keywords to select the papers: Action research evidence based software engineering common practices
2. Evidence based software engineering action research common practices
3. Evidence based software engineering practices case studies
4. Evidence based software engineering practices action research
5. Experiments evidence based software engineering practices
6. Evidence based software engineering common practices surveys case studies
7. Common evidence based software engineering practices
8. Evidence based software cost effort estimation testing industry data impact experiments simulations quality
9. Evidence based software cost effort estimation testing
10. Software Verification Validation V&V effort estimates current trends improvements
11. Improve current trends estimations Verification validation V&V effort
12. Estimations software Verification Validation V&V effort improvements current trends
13. Current Practices Testing Software Verification Validation

III. PAPERS STUDY SUMMARY

This section highlights Review goals, methods, results, and summary.

Review Goals
The goals of the literature review are to gain the real knowledge and status about the different papers that address the five research questions.

Review Method
I searched papers based on keywords, notion of keywords, using AND OR criterion, changing context

and words locations. I read the abstract, keywords, summary, and results to select the paper not only relying on keyword anchoring as small paper with less keywords can be of higher quality than a big paper with more keywords, so it needs manual intervention to assess the quality of papers to be included. I focused on surveys, case studies, experiments, and action research that are presented in the selected papers and will form a foundation for our research work. I extracted the bibliography, the relevance to our research needs pertaining to empirical evidence, verification, validation, testing, and above all cost effort estimation approaches related to main focus.

I rejected the papers that were: logically duplicate, biased, off-line, unclear, non-conforming, deviating from our main focus: testing, verification, validation, empirical evidence, and cost/effort estimation. Threats to validity in this review is induced from the irrefutable fact that there are omissions of papers due to human negligence, in-appropriate search results, non-indexed papers, unsearchable papers, non-locatable bibliography, unpublished thesis, irrelevant abstract/title/summary/results, and non-English. The time variant distribution of papers on research on cost estimation has been analyzed on the basis of regression, analogy, bayesian, expert judgment, work breakdown, function points, simulation, theory, classification /regression trees, combination of estimates , and others.

IV. REVIEW RESULTS

The 16% increase in other estimation approaches has been pointed to fuzzy logic, lexical analysis, genetic programming, and others. Only 15% of papers discuss expert judgment –based approaches. Authors argue that Boehm pioneered 1981 software engineering estimation model COCOMO is not relevant to current development practices but are used by 12 papers since 1995.

More than 60% of papers on cost estimation were identified by manual search and thus recommended to fellow researchers. The cross study of 5 research questions with 13 keywords among 45 papers showed that keyword 10. was contained highest number of hits while keyword 1. contained lowest number of matches in the selected reviewed papers. Similarly, RQ2 was answered in most of the papers while RQ4 was answered only in a few papers for all the thirteen keywords. I visualized that there are very few papers that could address the systematic literature review conducted for the five research questions versus thirteen keywords. The 59 out of 159 papers were in disagreements due to issues with recurring. Disagreements were also made on 12 papers due to descriptions and reclassifications occurred on problematic categories papers.

Review Summary

I noticed from a series of research conducted on effort and cost estimations based on different approaches from 1987 to 2012 that most of the papers are concentrated on reducing estimation errors arising out of estimated costs/efforts compared to actual cost and efforts. Accuracy, biases, and correctness of estimation models have been discussed. Effort is usually a function of project size but large project may be completed with lesser effort. Effort is calculated using parameters like LOC, FP, user stories, use cases that basically help in software sizing which help compute SLIM, COCOMO, COCOMO II, and other models to estimate efforts to be expended during software development tasks.

Papers addressing V&V testing effort estimations calculate as part of overall project estimations following similar pattern and conducting error estimations with similar behavior. The papers define verification as doing things right, validation as doing right thing, and testing as an inherent activity of V&V which is essential in maintaining quality of delivered software. There are also automated and manual tasks for enhancing the testing process so V&V is effective, productive, and valuable. Authors also have argued that testing may be skipped in case of emergency delivery to the client. There is also discussion on manual testing versus formal models and costs and risks associated with each one. The term Independent Verification and Validation (IV&V) is being constantly used so a third party is responsible to conduct SQA activities during SDLC. Many companies estimate testing efforts during verification validation phase of software development activities. Others do not separately estimate for testing tasks.

So, I decided to conduct the case study of five companies of Nepal to determine the actual status in verification, validation, and testing effort estimation tasks in software development.

V. CASE STUDY SUMMARY

Goals

The 50 paper reviews out of 150 total research papers showed polarized, vague, and incoherent sources of literature. So, we use our experience, knowledge, education, and technical skills in empirical evidence based software cost effort estimation, and testing into study of companies.

My goal is to assess the actual practical state of empirical testing estimations occurring in companies of Nepal. A lot of companies would not want to divulge their data so we needed to consider a minimum sample of companies willing to participate in the survey.

Methods

I designed 7 questions for company background, 9 questions for estimation /test methods, and 12 questions for last completed projects data with total of 28 questions that could provide insight into our research questions and contributing factors for the study. I conducted a case study of five software companies that are labeled A, B, C, D, and E to address the confidently issues and hiding their identity. Out of 100 top known companies of Nepal, only the five companies agreed to participate in the case study. I used interviews and questionnaires for the case study. I sent emails, made phone calls, and met CEOs and their top engineers, test engineers, product managers, and other members of the team. After they set the date and venue, I shared them the cause of my research and let them know in details about empirical evidence based testing estimations. After the training like session, they felt comfortable to fill the questionnaire. One of the companies did not have readymade data to fill in immediately but they later sent in email. All the studied companies were within 10 kilometer radius of Kathmandu valley. We could not reach out for companies outside of Kathmandu valley. The questionnaire was structured into three parts, namely: Company Background, Estimation and Test Methods, and Data from last completed projects.

Results

One of the companies was 25 years old while others were fairly new including 8 years old to 2 years old. One of the five companies had 200 employees while one of the companies had only 30 employees. Almost all companies had multiple foreign clients and a few local Nepali clients while 2 of them have 0 local clients. Company C and D clearly reflected the type of clients they serve by making high quality deliveries while others stated average numbers by keeping details confident. Clearly, Company D develops software for critical components of society where quality is a must and cannot be compromised at any cost. When entering client-size data, the companies underestimate, so write medium mostly as they must be confused on estimating how large is a big client and how small is another provided some clients may not be transparent and proper knowledge in assessing clients size is lacking that effects estimation accuracy and correctness. We may need to assert rules for proper client sizing in different scenarios; let us consider developing a software module for Microsoft or Google, and then certainly it would definitely impact all estimates because papers claim estimates are dependent on client sizes. Big clients have larger requirements while smaller clients have lesser requirements that would impact the estimates. I noticed that software verification validation and testing cost effort estimation using empirical evidence is in practice and perceived as contributing factor in providing quality outcomes by preventing budget/schedule underruns or overruns. RQ1, RQ2, and RQ4 were suitable for companies.

After receiving feedback from the five selected case study respondent companies, we analyze each of the research questionnaire items as described below:

Table 1: Company Background

Questions	A	B	C	D	E
When was your company established	2006	2011	1998	2008	2008
What is the size of your company	80	30	55	31	200
What kind of businesses does your company support	Mobile Technology and Mobile based value added services	Multi-channel integrations	Bank, Financial Institutions, Capital Markets, Hospitals, Cooperatives, Industries, Government, NGO	Software Security Solutions known as Security Information and Event Management (SIEM) Solutions	US Health Care management software
What are the average sizes of the projects your company handles	4 man-months	6-8 man-months	Medium-large (5 man-months)	2 man-months	15 man-months
How many types of programming languages does your company use	4	3	5	6	2
How many types of platforms does your company support	5	6	6	3	2
How do you rank your company compared with others in 1-5 scale (5 being best)	4	4	5	5	4

We can see from the data that they cover almost entire common software development industry, maturity, experience, coverage, strength, and self-ranking and are the empirical evidence for their quality sensitive approach. One of the companies C is 25 years old (The CEO is also a professor of Computer Science) and is a remarkable existence in country like Nepal to survive that long as markets here are not strong and long-lasting local or foreign clients are rare. The company E with 200 employees is from healthcare domain in USA and boasts being one of the best job-giver, best-quality software developing company establishing own computer science college to produce required manpower , so dearth of good quality software developers in Nepal can be met easily, and competition to acquire qualified developers can be dealt with. Company A boasts existing in 6 countries and deals with varieties of clients locally and globally. Company B claims that their application is global and so quality is imperative for their Japanese client who outsources them software projects. Company D definitely is leading error-free services to their Denmark and other European clients because of the sensitive security based software that is critical for millions of people. Company E plans to employ as much as 1000 software developers in Nepal software development center. Company B is too young they said and still perspiring to lead the industry with quality initiative with best quality outcomes. They however have mature CEO who constantly guides them, and monitors the activities during software development and delivery. Among them, Company D pays the best salary to it’s software developers as we investigated using inside sources and they were reluctant to reveal their revenue and pay scales. In Nepal, the culture of tax evasion by software companies is reported by experienced experts who claim that many do not participate in case studies and surveys in Nepal fear being identified or marked lest their data reaches tax office. From their own self-ranking, we can surmise that they are over-estimating themselves as being best as there is no tangible ranking empirical evidence that can definitely point and say: this software company is best compared to other.

They might also have got hindsight of competition with other software companies. We gave liberty to companies to fill in their own way so we see differences in filled data sets say for company size or project size according to their own interpretation. We see total of 396 developers when we sum all of them and it should represent 30% of Nepal’s software development industry although there is no formal or informal empirical evidence to claim the fact, but expert-opinion could make the prediction about overall Nepal’s software sizing estimation possible.

The CEO of Company E has software development experience since 1980sholding PHD in Economics while company D and A have younger CEOs of 2000s holding graduate level degrees.

Table 2: Estimation and Test methods

Questions	Choices	A	B	C	D	E
What kind of Model does your organization follow	A. Waterfall B. RUP (Rational Unified Process) C. Agile D. Others	B, C, D (own based on iterative incremental)	some are waterfall & some are agile	Water fall, Iterative	Agile SCRUM	Agile
Do you prepare a Project Plan	A. Y B. N	A	Yes	Yes	Yes	Y
Do you prepare an estimate for the project	A. Y B. N	A	Yes	Yes	Both Waterfall and Agile SCRUM Based Estimation, Waterfall: Size &Tasks, Time Boxing for Headline	Y

Questions	Choices	A	B	C	D	E
What kind of Model does your organization follow	A. Waterfall B. RUP (Rational Unified Process) C. Agile D. Others	B, C, D (own based on iterative incremental)	some are waterfall & some are agile	Water fall, Iterative	Agile SCRUM	Agile
					(aka Epic), User Stories and Tasks	
How do you prepare a project estimate	A. Expert-Judgment B. Estimation model C. Others	A	A, B & some are research based	A. Expert - Judgment	Estimation Model: Tasks Based and Time Boxing for User Stories (using estimations based on history data),Expert - Judgment	Expert - Judgment
In the estimation work, did you use historical data (data from previous projects)	No choices were provided but free text was expected	To some extent based on previous projects and expert judgment	Y	Yes we do use	Yes (previous releases)	Yes
Which levels of testing did you use	A. Regression Testing B. Unit Testing C. Others	functional testing, SIT, UAT	mixed	All A Regression Testing B. Unit Testing, Functional Testing	Unit Testing (by Developer),Integration Testing (by Developer),System Testing (by QA),Regression Testing (by QA), Manual, Automation: Siesta Framework (for UI) and, Automation: Robot Framework (for Engine level acceptance tests), User Acceptance Testing (by QA and Denmark Team)	Regression, Unit , integration, system test, performance test, security test, smoke test
What methods did you implem	A. Manual B. Automated C.	A	Mixed	Manual	A,B	A,B

Questions	Choices	A	B	C	D	E
What kind of Model does your organization follow	A. Waterfall B. RUP (Rational Unified Process) C. Agile D. Others	B, C, D (own based on iterative incremental)	some are waterfall & some are agile	Water fall, Iterative	Agile SCRUM	Agile
ent to conduct testing	Others					
Can you produce the evidence of those artifacts	A. Test Plan B. Test Case C. Test Script D. Test Data E. Test Report F. Others	A, B, D E. evidence in PM tool (Trac)	Email and Documents	Yes	Yes for Test Plan, Test Cases, Test Script, Test Data and Test Report	A,B,C, D,E
How Testing efforts are estimated	A. Time Boxing B. Percentage of development estimates C. Expert estimate D. No separate estimates for Testing E. Others	C and D	C	C	A,B, C (sometimes)	C

Table *2* shows the strength of test estimations in software development companies under case study. Definitely, as anticipated, Expert-Judgment leads across all companies although alternative estimation practices have been indicated. Similarly, Agile based software development model establishes as leader. Company D has better details on preparing estimation using own tool. All companies claim using historical empirical evidence for estimation tasks. Company D has special automation testing levels using own framework, while Company E adds security testing as key feature compared to others. Automated testing is on rise as 3 companies claim using it. Those companies using manual testing methods claim automation is huge task in itself to implement and are considering adopting. We can see Company B not indicating any tool to keep test evidence but email and documents. Only one company D has knowledge of using Time Boxing effort estimation, while others had not heard the term too, they confessed during case study interviews. Again, Expert-estimates lead across estimating Tasks among companies and while averaging 25 year old expert knowledge, Company C took one week to fill the averages because they wanted to provide accurate and correct estimation data. Unit testing and regression testing are commonly followed as seen from the evidence presented by the companies. Only one Company A knew about RUP (Rational Unified Process) software development model, and rest were unaware of it. All companies prepare project plans and estimates which I see as a good practice.

Table 3: Data from last completed projects

Questions	A	B	C	D	E
How many types of clients do your projects serve	8	4	Government,International,local,public,Privatebusinesses,doctors,managers,officersetc 10+	Finance,Healthcare,Government,Telecom&ISPs,Defense&Aerospace,Utilities and Energy (SCADA) 7+	22
What is the average size of your clients that projects serve	Medium	1000	Medium-large	Medium	5
How many local clients do your projects serve	100	0	Almost all	0	2
How many foreign clients do your projects serve	2	50	3	300	30
Total number of Estimated Effort for your project	500 man-hour	12 man-months	3500 man-months	Major Release (Enhancement): Approximately 8 - 12 months, Minor Release (Maintenance): Approximately 3 - 6 months	160 hours
Total number of Estimated Effort for the Testing activities	100 man-hour	2 man-months	1800 man-months	Major Release (Enhancement): Approximately 6 months, Minor Release (Maintenance): Approximately 1 month	80 hours
Total number of Actual Effort for your project	600 man-hour	15 man-months	4200 man-months	Major Release (Enhancement): Approximately 8 - 12 months, Minor Release (Maintenance): Approximately 3 - 6 months	200 hours
Total number of Actual Effort for the Testing activities	120 man-hour	3 man-months	2500 man-months	Major Release (Enhancement): Approximately 6 months, Minor Release (Maintenance): Approximately 1 month	100 hours
Estimation Error for the Project	16.66667	37.66234	16.66667	0	20
Estimation Error	16.66667	33.33333	28	0	20

Questions	A	B	C	D	E
How many types of clients do your projects serve	8	4	Government,International,local,public,Privatebusinesses,doctors,managers,officersetc 10+	Finance,Healthcare,Government,Telecom&ISPs,Defense&Aerospace,Utilities and Energy (SCADA) 7+	22
for the Testing					
Reason of Estimation Error for the project	Change in Requirement, Delayed feedback from Client	Client delays, inconsistencies in requirements	Requirement not proper, lack of system knowledge	N/A	project complexity and unseen problems
Reason of Estimation Error for the Testing	Change in Requirement, Delayed feedback from Client	Client delays, inconsistencies in requirements	Requirement documents not in hand before QA and not clear sometimes	N/A	error in development estimates

Table *3* provides insight into last completed projects of the 5 companies who participated in case study display effort estimation by expert-judgment in development as well as testing efforts both estimation errors of approximating to 20% to complete the software development projects. There might be lack of understanding and knowledge to segregate client types in some cases that impacts their estimation errors. All the respondents involved in Case Studies claim that clients definitely are major cause of project schedule and cost motivators that has been extensively researched by Grimstad et al. [1] based on literature review and survey of 300 software professionals. Company C and D clearly reflected the type of clients they serve by making high quality deliveries while others stated counts by keeping details confident. Clearly, Company D develops software for critical components of society where quality is a must and cannot be compromised at any cost. In filling clients size data, the companies underestimate so write medium mostly as they must be confused on estimating how large is a big client and how small is another provided some clients may not be transparent and proper knowledge in assessing clients size is lacking that effects estimation accuracy and correctness. We may need to assert rules for proper client sizing in scenarios like, developing a software module for Microsoft or Google and how to put client size then, that would definitely impact all estimates. Two companies B and D reported 0 local clients that shows their dependency on foreign outsourcing opportunities while displays their competence , quality initiative , and client happiness. Actually Nepal’s software market is poorly entangled in clients not being able to pay, unmotivated for software solutions, and looking for penny wise pound foolish solutions from lone consultants where quality is not guaranteed, but solves purpose of most of the market. Company E is unable to get millions from a hospital because the hospital cannot pay, although uses their software system while Company C has 100s of software in the shelf as local market is not able to grasp the trove of software. Company A faces tough competition in mobile based and other web based software share, but is leading the market since many years. In one case, even some foreign clients ran away without paying after using millions of dollars worth software. Company C has mentioned “almost all” for local clients and really they are a well-established brand since last 25 years serving almost all arenas by providing software and winning almost all bidding. Regarding serving foreign clients, we can see Company D leading although other companies who put smaller numbers there confided that they serve more but cannot estimate properly due to complexities. Company E could not get data from all projects due to busy schedule of project managers, product owners, so filled only 30 from average of one project only. Estimated projects/testing compared to actual values show varieties in filling the same thing: man-hour, man-months, hours, months and we did not think it a good idea to pressure them to fill it making suitable for same unit project estimation but converting them later into same estimation error calculations. Company D is of interest particularly because they operate differently with their clients and their estimates are time based, and there is no chance of error for criticality of their clients or, their billing process with clients depends upon month based major and minor releases, so they always meet the deadline. It however poses risks because they must be estimating using other parameters too instead of relying on release dates only and assuming 0% error in estimation because it is not possible to obtain 0% error as the empirical model would not support that. Another interesting study is made on companies A, C where both have same value of Project estimation errors 16.67. However, Company C has poorer estimation error 28% compared to 16.67 of Company

A. Real-world evidence collected from experts in quality software development in Nepal would agree with the data filled here because of the answers filled on testing methods logic, and evidences. Company B suffers from heavy project and testing errors 37.66% and 33.33% respectively, in estimation because of their clients' random needs, ideas and quality requirements that is constantly new and experience does not work because of clients. They confided estimation errors up to 200% when fulfilling wild dreams of clients. Company E demonstrates 20% error in estimation which may be realistic depending upon their business success, software development growth, and huge company size, or, they might have underestimated fearing not being able to assess all the parameters properly for empirical evidence based computations for projects and testing. Inside Sources say, Company E has dedicated experienced, mature 25% of total employees in verification, validation, and testing that proves it's commitment to superior quality of deliveries to clients. Regarding cause of projects/testing estimation errors, companies A, B, C blame clients related issues while, D has nothing to say and E pointed to project complexity, unforeseen factors and error in estimation itself.

Discussion

From the case studies we conducted on five software companies of Nepal, we find that software verification validation and testing cost effort estimation using empirical evidence is in practice and perceived as contributing factor in providing quality outcomes by preventing budget/schedule underruns or overruns. The result motivates fellow researchers, academicians, and industries to follow the practice to improve their estimation accuracy and predictability in testing. Expert opinion is regarded as the major method for estimating software cost effort apart from the empirical evidence present across projects claimed by all 5 companies that participated in research. We found the following regarding five research questions discussed in section B from the case study:

RQ1: Do Companies collect and use historical empirical data for the purpose of estimation of V&V effort?

Yes, all the five companies claim to collect and use historical empirical data for estimating V&V effort although they admitted expert opinion as the key factor.

RQ2: What are the current practices in effort and cost estimation?

The companies involved in the case study use agile and waterfall models both depending on the nature of projects while all claimed to use agile only one company A claimed to use Rational Unified Process (RUP). All companies claimed preparing project plans and estimates. All companies prepare project estimates primarily using expert-judgment and Company B claimed using other research estimation models also not directly specifying which one. Only Company D claimed using time boxing model. All companies claimed using historical empirical evidence not disclosing how they used it. All five companies claimed they use regression testing unit testing and others. Companies A and C use manual testing method while companies B, D, and E claimed using manual as well as automated testing. V&V Testing artifacts like Test Plans, Test Scripts, Test Data, and Test Reports are claimed to be prepared by all companies as evidence while Company A claims using Project Management tool and Company B claims keeping evidence in email/documents specifically.

RQ4: How can we improve the current trends in estimations of V&V effort?

All companies replied that they need to invest more on research in improving current V&V effort estimation trends continually and by studying constantly the research papers to increasing the awareness, and knowledge among team members.

VI. THREATS TO VALIDITY

Threats to validity have been pointed to publication bias, vested interest of authors, and unfamiliarity with other fields. Five research questions have been projected to improve software verification, validation, and testing cost estimation. Only 5 companies could be used in Case Studies. The time variant distribution of papers on research on cost estimation has been analyzed on the basis of regression, analogy, bayesian, expert judgment, work breakdown, function points, simulation, theory, classification /regression trees, combination of estimates , and others. The 16% increase in other estimation approaches has been pointed to fuzzy logic, lexical analysis, genetic programming, and others. Only 15% of papers discuss expert judgment –based approaches. Authors argue that Boehm pioneered 1981 software engineering estimation model COCOMO is not relevant to current development practices but are used by 12 papers since 1995. More than 60% of papers on cost estimation were identified by manual search and thus recommended to fellow researchers. The papers stress on the need to focus on basic software cost estimation research topic, common industry practices, real-life data, and to conduct lesser studies in arbitrary datasets.

VII. RESULTS/SUGGESTIONS

From the case study of 5 companies and review of 150 papers, we suggest that testing data estimations need to be recorded, revised and historical evidence be taken into account for avoiding underruns or overruns in testing. We conclude that the reviewed 50 papers out of 150 selected papers spanning 30 year time period fail to provide conclusive evidence of research in evidence based software cost effort estimation for V&V Testing that indicates more effort on the education and knowledge is expected to spread the research arena. The papers that were studied focused on quality of software development but attributed less to cost effort estimation of V&V testing using empirical evidence. Therefore, we conducted a case study of 5 companies in Nepal to assess the industrial scenario and found that practically the companies are estimating V&V Testing cost efforts using empirical evidence and expert-judgment based models. We summarize the main findings of the case study by following observations:

1) Test effort is similarly calculated as Total Project effort estimation using expert-judgment

2) The estimation error of Testing effort seems to correlate closely to the estimation error of Total Project.

However, the companies can still improve the cost effort estimation errors of V&V Testing and projects by detailed analysis of their process by taking help from consultants. We request more research in diverse demographics, culture, client base, application type, and clients.

REFERENCES

[1] Magne Jørgensen and Stein Grimstad, "Software Development Estimation Biases: The Role of Interdependence," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 3, MAY/JUNE 2012, pp. 677-693, 2012.

[2] Magne Jørgensen and Martin Shepperd, "A Systematic Review of Software Development Cost Estimation Studies," Software Engineering, IEEE Transactions on, pp. 33-53, 2007.

[3] Stein Grimstad, Magne Jørgensen, and Kjetil Moløkken-Østvold, "The Clients' Impact on Effort Estimation Accuracy in Software Development Projects," in Software Metrics, 2005. 11th IEEE International Symposium, Como, 2005, pp. 10 pp-10.

[4] Aritra Bandyopadhyay, "Mitigating the Effect of Coincidental Correctness in Spectrum Based Fault Localization," 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, pp. 479-482, 2012.

[5] Claes Wohlin, "An Evidence Profile for Software Engineering Research and Practice," in Perspectives on the Future of Software Engineering - Essays in Honor of Dieter Rombach (Eds. Münch, Jürgen; Schmid, Klaus).: Springer Berlin Heidelberg, 2013, pp. 145-158.

★★★