A GENERIC FRAMEWORK FOR TEST EFFORT ESTIMATION
MODEL SELECTION

Abstract

Quality plays an important role in software development. Even so, in most software projects testing – as one quality improvement and control technique - is not planned properly and thus requires more time and resources than initially planned. This shortcoming can only be avoided if planning is thoroughly done and based on realistic assumptions and effort estimations.

Several approaches exist for test effort estimation, but all of them have limitations. There’s no systematic analysis to date that helps to decide if a model, method or framework can be applied successfully in a certain project. Once this analysis is done, a structured and well-founded selection process can be constructed.

This paper presents a methodically derived set of criteria for examining test effort estimation models and helps to make them comparable. Furthermore, it integrates these criteria in a generic framework that ensures a comprehensible selection. The framework usage is illustrated by a fictitious case study concerning a standard business application from a global software developing company.

Keywords: test effort, estimation, model selection, generic framework
1 INTRODUCTION

In software development projects, tests are frequently treated as subordinate to development. For achieving continuous quality improvements, however, test goals and the test procedure should be planned in more detail. The required test effort should be estimated more diligently. A problem in this context is the selection of the appropriate effort estimation model among the existing ones. A systematic selection process is needed, but is often not in place. It is the intention of this paper to develop such a systematic, generic selection framework.

1.1 Aims of this paper

The aim of this research paper is twofold: one regarding content and one regarding form. As it regards content, a framework is developed to evaluate the appropriateness of test effort estimation models for a given development situation of standard business software\(^1\). Formally, well-reasoned recommendations to solve practical problems will be given. In the context of pragmatic scientific research, these recommendations are called praxeological statements (cf. Kubicek 1975, p. 13).

Following the formal aim, the conceptual framework contains the four basic elements of praxeological statements (cf. Kubicek 1975, pp. 15-33): design goals, design restrictions, design parameters (or measures) and design efficiency and effects.

![Conceptual framework to evaluate test effort estimation models (according to Kubicek 1975, p. 22).](image)

The test in a software project, mostly distributed across several phases, represents the design object. Therefore, the conceptual framework consists of test goals, test restrictions, test parameters and effects of the test parameter (or measures). How these elements are connected is depicted in figure 1.

---

1. Enterprise Application Packages that automate and integrate information and processes (e.g. ordering and production engineering) of the essential functions within an organization are software systems also called Business Software.

2. Test restrictions are conditions in a project that cannot be altered in the short term, such as dependency on other projects.

3. Test parameters are factors affecting a project that can be changed in the short term, such as measures like test case reviews which can be decided upon on project level.

4. These effects arise from design restrictions and design parameters and affect design goals.
1.2 Definition of relevant terms

According to Stachowiak (1973) a model consists of three features: 1. reduction feature - the model contains only a relevant selection of the original's properties, 2. mapping feature - the model has to be based on some original and 3. pragmatic feature - the model can be used in place of the original by certain subjects within a certain time interval and restricted to certain theoretical or actual operations.

Effort refers to the required working hours spent by members of the project team. This represents by far the largest part of effort spent in a software project (cf., e. g., Leung & Fan 2002, p. 808).

Tests are measures of analytical software verification and validation carried out by executing code (see IEEE 1990).

Estimation in this context refers to an unbiased prediction of the most likely value which is qualified a) by upper and lower bounds, b) by the risk associated with the estimate, and c) by the assumptions that were made when the estimate was made. The estimation is transitory, i. e. it is valid at a certain point in time and will be outdated at later points in time (cf. Kitchenham 1996, p. 183)

1.3 Outline

This paper is structured as follows: In the next section related scientific work will be discussed briefly. In section 3 the used research method will be explained. Thereafter, the framework is developed. To this end first the relevant criteria for model evaluation are deducted and described (section 4). Then, in section 5, the steps necessary to reach a sound decision are explained. In section 6 the framework will be illustrated and evaluated by means of a case study. Finally, section 7 sums up lessons learned, mentions limits to the framework’s application and points to further research.

2 RELATED WORK

Model selection can be viewed as a decision problem under uncertainty\(^5\). A number of generic solutions can be found in the literature, such as prioritized checklists (Kenny & Raiffa 1976) or Analytical Hierarchy Process (AHP, Saaty 1980). However, these methods have to be adapted to the situation under study. This means defining the relevant criteria and adjusting the steps to take.

There are also methods explicitly developed for use in a problem domain. For example, some authors describe approaches for selection of a Software Reliability Growth Model (SRGM). Stringfellow & Andrews (2002) present a method to select the optimal SRGM using historical project data and simulation. Bhawnani et al. (2005) describe an approach to support release decisions by selecting an appropriate SRGM. However, in order to apply it, the procedure contains too many factors, formulae and steps that are not explained. Menzies et al. (2006) develop and recommend a method to select best practices for effort estimation. Their intention is to avoid typical problems of software estimation, e. g. too small samples of completed projects.

The approaches mentioned require practical application of the analyzed models using historical data. This suitability test, however, is time-consuming and can only by done with the corresponding data.

Generic selection techniques, on the other hand, are not suited for model selection for test effort estimation because the decision’s dimensions are not trivial to determine. Moreover, a number of test effort estimation models (TEEMS) exist which cannot be compared trivially. In order to find an appropriate model objectively and systematically, analogous to the approach by Stringfellow &

\(^5\) Decision under uncertainty because when selecting a model, never all information is available, thus lending uncertainty to any decision and assumption (cf. Meyer et al. 2002).
Andrews (2002) a structured procedure for objective selection of test effort estimation models has to be developed.

3 RESEARCH APPROACH

This paper follows a constructive research approach. A framework is designed which can be used in software development without constraints and is thus called generic.

According to the pragmatic research aim, recommendations are deducted which provide a solution for the problem of selecting an appropriate model for test effort estimation.

Foundation is the conceptual framework (cf. figure 1) which describes the environment of the object under study (test effort) on an abstract level.

In the process, we revert to existing estimation models and methods from which a systematic, situational selection has to be made. The estimation models and methods in this paper are exemplary in nature and serve as illustration of the framework.

Model selection requires analyzing existing approaches. This is done by means of theoretically deducted criteria. Besides this theoretical basis, our research is illustrated by applying the developed framework in a fictitious case study. A case study is capable of investigating the estimation procedure (“how”) without the researcher being able to measure or control the variety of influencing factors that affect the object of analysis (estimation model selection, cf. Yin 2003, pp. 8 et seq.).

In a case study (see section 6), semi-structured interviews can be done with the project lead and the person in charge of quality assurance. Besides this, various documents can be evaluated (e. g., quality goals on a multi-project level, presentation of the development process, quality plan of the project etc.) in order to gather project restrictions and goals.

4 CRITERIA FOR EVALUATION OF EXISTING APPROACHES TO TEST EFFORT ESTIMATION

The catalogue of criteria for analysis of existing approaches consists of two parts:

1. Criteria to match models and project with respect to the conceptual framework (criteria A to C),
2. Requirements of a model for test effort estimation (criteria D to H).

The first part was derived from the conceptual framework (see section 1.1, figure 1). It serves for matching each model on one hand with the project to be estimated and the project’s organization and domain on the other. Criteria D to H are based on scientific work regarding effort estimation (see e. g. Noth & Kretzschmar 1986, Bundschuh et al. 1991) and scientific theory in general. Irrespective of project, organization or domain they are applied to the remaining models for evaluation and subsequent selection.

1. Matching of models and project with respect to the conceptual framework

A) How is the system of goals represented (is the process modeled goal-oriented?, which goals does the model contain? can the user choose between different goals?)

---

6 I. e., apart from answering predetermined questions, the interviewee has the opportunity to tell his experiences without addressing concrete questions.
7 Especially Popper (1968) is drawn on for identification of scientifically sound approaches.
According to the formal goal (see section 1.1) design goals are essential elements of a model for test effort estimation. For situational selection, the goal set in the given situation has to be selectable from the goals modeled.

B) Is a distinction made between restrictions and parameters regarding humans (e.g., mathematical knowledge), processes (e.g., a certain process model, tool support), regarding products (e.g., technology, data quality and quantity) and the project (e.g., number of locations or structure)?

Test restrictions and parameters accounted for allow for matching them with the situation given: the more modeled factors that can be measured in the project, and the less factors affecting test effort in the organization or project that are missing in the model, the better suited a model is in the situation.

C) Are assumptions concerning effects stated explicitly?

When assumptions about the effects and their relationships underlying the estimation approach are stated, the user can check whether they are valid also in the situation given. Besides, estimation results become traceable and the user can influence the estimation result according to the assumptions concerning effects.

2. Requirements of a model for test effort estimation

D) Falsifiable? Objective?

Falsifiability according to Popper is the criterion to be used for demarcation of empirical systems and statements\(^8\) (cf. Popper 1968, p. 41). It states that they have to be refutable by experience. Objectivity of an estimation approach means that the estimation has to be based on a formal process, and that different persons using the approach arrive at the same result. When both characteristics are present, portability to different application environments is assured in principle. Apart from this, the assumptions regarding effect relationships underlying the approach can be tested against reality. When the assumptions are corroborated in a case, this raises the credibility of the assumed effects and their relationships. This in turn strengthens or extends the scientific knowledge concerning test effort (cf. Popper 1968).

E) Model maturity

“Maturity” on one hand refers to the number of practical applications that have contributed to improving the model. A purely theoretical model has never been applied in practice. On the other hand, the diversity of the applications is of concern. Following Popper’s argumentation (1968), empirical scientific statements have to be tested in a large number of different environments in order to be corroborated empirically. In the context of software test effort, these tests should take place in different organizations\(^9\) and different domains\(^10\) to cover a broad spectrum of possible environments\(^11\).

F) Usage experience: user satisfaction

This criterion is applied to evaluate quality of the results (cf. Noth et al. 1986, p. 30, Bundschuh et al. 1991, p. 17). It is investigated whether the users state satisfaction with model application and whether they continue using it. This subjective evaluation confirms that the estimation model served its purpose in one case. Thus, it adds to testing the model (see criterion “model maturity”) and strengthens the empirical basis.

---

\(^{8}\) Statements are components of an empirical scientific system, cf. Popper (1968) p. 27.

\(^{9}\) Institutional meaning: specific orientation towards a purpose; well-arranged distribution of labor; persistent boundaries (Schreyögg 2004, pp. 969 et seq.), i.e. governmental and non-governmental organizations. In this sense, a project can be considered an organizational unit (Herzwurm 1998, p. 95)

\(^{10}\) A domain is characterized by a number of concepts and technologies in a knowledge area (Bass et al. 1997), or by common expertise, common design and a common market (Mili et al. 1999).

\(^{11}\) Scale: see legend for table 2 in the case study, section 6.4.
G) Usability

Usability (cf. Noth et al. 1986, Bundschuh et al. 1991) comprises understandability (G1) of the procedure, which is necessary for acceptance by everyone involved in the estimation. A very complex model would not only require thorough mathematical knowledge but would also be of little predictive value (cf. Grams 1999)\(^{12}\).

Adaptability (G2) is the second element of usability. Which model restrictions and parameters can be determined in which way has to be explained for the respective model\(^ {13}\).

The third element of usability is applicability (G3) (see Pol, Koomen, Spillner 2000, p. 29). Indicator for this is whether (commercial) tool support is available\(^ {14}\) (Noth et al. 1986, p. 30).

H) Project control

For one, the information necessary for control is of concern, i.e. whether a feedback loop is contained in the model (Kitchenham 1996, p. 183). Apart from this, according to Kitchenham (1996) any estimation is only valid at a certain point in time and thus has to be updated. This can be done once feedback has been obtained.

Also, it is investigated whether alternative actions to control the project are identified and their effects explained as part of the model’s description (Noth et al. 1986, p. 30, Bundschuh et al. 1991, pp. 17 et seqq.).

5 FRAMEWORK FOR MODEL SELECTION

As a prerequisite, several models for test effort estimation have to be under consideration\(^ {15}\).

To prepare for model selection, goals, restrictions and parameters (criteria A to C) have to be identified on project, organization and domain level. The procedure takes the project manager’s perspective. Thus, factors that can be changed on domain or organizational level, i.e. parameters on domain or organizational level, are represented as restrictions on project level. Table 1 shows which of the criteria A to C are analyzed on which level of model selection.

<table>
<thead>
<tr>
<th>Level</th>
<th>Goals</th>
<th>Parameters</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain level</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Organizational level</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Project level</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Levels of goals restrictions and parameters

Next, model selection is carried out on three levels (see figure 2).

The first level is the domain that the project belongs to.

\(^{12}\) The intermediate level of the scale used („partly”) means that not comprehensible for every influencing factor or not for each step of the estimation or not from every level of knowledge

\(^{13}\) The intermediate level of the scale used („partly”) is assigned to models for which not all parameters are indicated explicitly or in which not all parameters’ determination is explained.

\(^{14}\) Not the sole indicator of applicability, yet tool support is mentioned most frequently as characteristic of applicability in the literature concerning effort estimation.

\(^{15}\) For illustration, care was taken to choose heterogeneous models (concerning research approach, underlying theories, time of publication etc) to provide an overview (not claiming completeness of any sort).

\(^{16}\) Goal or purpose is an imagined and intended future event or state, an anticipated idea of the effects of our actions (see Bidlingmaier 1964, p. 28). Thus, domains cannot have goals, but only organizations.
As first step, domain restrictions are matched against model restrictions. It is also checked whether the models were applied in this or similar domains. Relevant restrictions\textsuperscript{17}, e.g. are competitive environment (cf. Herzwurm 1998, pp. 88 et seq.), possibility of customer control, risk potential associated with use of the software (security requirements) and possible variation of usage.

In the second step on this level, models are discarded which do not comply with the domain restrictions, or whose restrictions or prerequisites are not fulfilled in the domain.

On the second level, criteria A to C are applied to the organizational level, i.e. organizational goals and restrictions are investigated.

As a first step organizational goals and restrictions are matched against those of each model. Organizational goals\textsuperscript{18} e.g. pertain to duration of a release cycle or sales that have to be reached in a certain time interval by a product division. Restrictions, e.g. concern resources (human, financial, schedule, spatial) and availability of technologies. Models whose goals and restrictions do not match those of the organization are discarded in the second step.

The third level is the project level.

As a first step, goals, restrictions and parameters (criteria A to C) of the project are matched against those of the remaining models, and models that are not compliant are discarded.

Goals on project level can be competitive goals or quality goals (e.g., number of tested scenarios, defects found and corrected). Restrictions concern mainly human resources (e.g., number of available testers and their respective experience with testing and with the functional environment) and historical project data (e.g., kind and priority of detected errors, planned and actual test effort). Parameters are basically the test method (requirements based; risk based; systematic versus operational testing) and tester selection (internal or external employees, developers or non-developing personnel, experience with the product as user, consultant, tester).

In the second step on the third level, the remaining models are analyzed according to criteria D to H. To do this, any technique for decision making can be applied, such as prioritized checklists, unweighted 0-1 factor / weighted factor scoring method (Kenny & Raiffa 1976) or the Analytical Hierarchy Process (AHP, Saaty 1980) Eliminating criteria can also be defined.

To select an appropriate estimation model in a given project, level 3 (see figure 2) is sufficient as long as selection on level 1 and 2 has been done for the domain and organization (initialization).

Selection on level 1 and 2 only has to be repeated in the following cases:
1. One or several additional models are to be included.
2. Domain restrictions have changed e.g. due to changing market conditions.
3. Organizational goals and restrictions have been shifted, e.g. quality goals are to be given higher priority.

\textsuperscript{17} For an overview of restrictions in the design of software development cf. Herzwurm 1998, pp. 68-111.
\textsuperscript{18} Cf. Kupsch 1979, pp. 20 et seq.
6 CASE STUDY

The fictitious case study describes a project dedicated to developing additional functionality for a human resources management system. The setting as well as the conclusions drawn represent realistic conditions. Five test effort estimation models are under consideration (see table 2), and the selection is done following the framework described in section 5.

6.1 Level 0: Preparation (identification of relevant goals, restrictions and parameters)

Domain restrictions, organizational goals and restrictions and goals, restrictions and parameters of the project are determined. To do this, the project lead, the person in charge of quality assurance and two long-time managers in the company are being interviewed. The interviewer lists possible answers and asks for agreement or disagreement and comments. The following goals, restrictions and parameters have been identified:

**Domain restrictions** (domain “standard business software”):
- The competitive situation (supplier oligopoly) demands keeping budget and schedule restrictions.
- New products are subject to customers’ market power.
- Bug corrections after market release are possible (in contrast to safety-critical software).
- Highly variable usage, thus not all process chains can be tested.

**Organizational goals (concerning test effort estimation):**
- The estimation is supposed to give input for resource and schedule planning, i.e. the model has to be applicable early in the product life cycle.
- Additional functionality is preferred over non-functional quality (e.g., performance).

**Organizational restrictions:**
- The development process is similar to the V-model. Development is object-oriented.
- Test process: There are requirements regarding test phases (number, time interval), test content (e.g. new and existing functionality, performance), and correction of found defects. There are no directives regarding test cases (scope, detail, content) and tester selection (internal/external; qualification, location). In addition, the test process is being changed continually in small increments, in particular requirements-based testing is being promoted.

**Project goals:**
- Traceability (to comply with SOX and audits), i.e. test cases have to be linked to requirements.
- Errors found in test have to be eliminated before the next test phase begins.
All existing test cases have to be run once without error.

**Project restrictions:**
- New development
- Developer experience: considerable with programming (seven years on average), none with the product, considerable (five years on average) with the process model and processes.
- Tester experience: none with the product, rather considerable with testing (three years on average)
- Test cases: two reviews take place; developers from one subproject test in another subproject; developers, quality and product manager create test cases; no automation
- High dependency (on schedule of a higher-ranking project, technical: on two middleware products, conceptual: on existing scenarios, into which the component will be integrated)
- Schedule restriction: given by higher-ranking product (test phases: covering three development phases 10, 2 and 2 weeks module test; 6, 4 and 2 weeks integration test; 2 weeks scenario test).
- Complexity: functional: low; technical: rather high (due to large number of interfaces and relatively new technology)

**Project parameters:**
- Code inspections and design inspections are carried out.
- Method: requirements based (30% of test cases), development-based (i.e. solution-based: 70% of test cases), systematic testing (not operational).
- Tester: few externals, mostly internals (1:10); developers test 100% of module tests, testers carry out 100% of the integration tests.

### 6.2 Level 1: Pre-selection on domain level (results)

*Boehm et al. (2000)*: Domain restrictions are supported by the model, model prerequisites are met in the domain “standard business applications”, so that the model seems appropriate.

*Calzolari et al. (1998)*: Domain restriction “competitive situation” is not matched because it is neither modeled as goal nor as restriction. Thus, the model is discarded.

*Cangussu et al. (2002)*: Domain restrictions are not violated. Model prerequisites are met in the domain “standard business applications”, so that the model seems appropriate.

*NASA (1993)*: Domain restriction “competitive situation” is not matched because the corresponding restrictions aren’t part of the model. Thus, the model is discarded.

*Sneed & Jungmayr (2006)*: Domain restrictions are not violated (competitive situation is taken into consideration by degree of coverage), model prerequisites are met in the domain “standard business applications”, so that the model seems appropriate.
6.3 Level 2: Pre-selection on organizational level (results)

Boehm et al. (2000): Assumptions and features of the V-model are mostly covered by the waterfall model; however, no historical project data is available in the organization and function points or lines of code are not estimated or counted, therefore, inappropriate.

Cangussu et al. (1998): Prerequisite of historical project data for parameter estimation is not met in the organization; data is only available on project level; therefore, inappropriate.

Sneed & Jungmayr (2006): Comparability of test cases is not required in the organization, yet possible on project level; V-model and oo-development are given; tester productivity measured in completed test cases per day varies extremely, but may be constant on project level; measurement of module complexity and interface density is not required in the organization but possible on project level; tool support not available for this, yet possible on project level. Therefore, the model may be appropriate.

<table>
<thead>
<tr>
<th>Model is falsifiable</th>
<th>Objective</th>
<th>Usage Experience</th>
<th>Maturity</th>
<th>Usability</th>
<th>Tool Support</th>
<th>Feedback loop</th>
<th>Alternative measures given</th>
<th>Language independent</th>
<th>Project control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boehm (2000)</td>
<td>hardly</td>
<td>no</td>
<td>high</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Calzolari (1998)</td>
<td>no</td>
<td>yes</td>
<td>low</td>
<td>3²</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Cangussu (2002)</td>
<td>no²</td>
<td>no²</td>
<td>low³</td>
<td>4¹⁰</td>
<td>no</td>
<td>partly</td>
<td>no¹⁵</td>
<td>yes¹⁸</td>
<td>yes²¹</td>
</tr>
<tr>
<td>NASA (1993)</td>
<td>partly⁷</td>
<td>---²</td>
<td>high</td>
<td>2¹¹</td>
<td>no</td>
<td>yes</td>
<td>²¹²⁶</td>
<td>no</td>
<td>---²⁷</td>
</tr>
<tr>
<td>Sneed (2006)</td>
<td>no⁴</td>
<td>no</td>
<td>med.⁵</td>
<td>²¹²¹²</td>
<td>yes</td>
<td>no³</td>
<td>no³¹</td>
<td>no²²⁰</td>
<td>no²³</td>
</tr>
</tbody>
</table>

*a*Because of space reasons, only the first author or the organization is mentioned. *b*1: theoretical model, 2: practically applied within one organization, 3: practically applied within more than one organization, but only in one domain, 4: practically applied within more than one organization and domain, and 5: practically applied many times and not only by the author(s). *c*yes/no: is comprehensible how the estimation result is obtained? (for higher acceptance, credibility). *d*yes/no: is explained which parameters have to be determined and how?; partly: not all parameters are identified for determination or not for each is explained how to determine its value; *e*not mentioned.

*additional parameters can be added and some mentioned parameters can be irrelevant (p. 181 & 191). *²*parameter quality of the test process is undefined and has to be estimated. *³*in general not falsifiable, only with concrete data. *⁴*not defined how the used metrics should be measured and used terms have no clear definition. *⁵*see footnote 2. *⁶*two case studies with different results. *⁷*success questionable (p. 37). *⁸*COCOMO was and is applied to many different domains and within many different organizations. *⁹*applied to two not detailed described projects – domain unknown. *¹⁰*applied to two projects: text editor development and porting of a business application from COBOL to ABAP. *¹¹*applied to NASA projects only. *¹²*applied to financial application. *¹³*project-specific scaling factors are not described in detail (p. 27). *¹⁴*commercial tools available. *¹⁵*tool support would be very helpful (p. 795). *¹⁶*no commercial tools available. *¹⁷*high effort for manual counting. *¹⁸*p. 783f. *¹⁹*several re-estimates are prescribed, yet no comparison estimate-actual. *²⁰*entering remaining number of testcases, effort can be re-estimated, p. 32. *²¹*p. 793. *²²*but number of statements or functions points needed. *²³*applicable to object-oriented software only.

Table 2. Models analyzed.
6.4 Level 3: Selection on project level using a prioritized checklist (results)

*Sneed & Jungmayr (2006):* The project’s test cases have been created without guidelines by people in different roles and are run by developers and testers with different experiences. Since the model’s restriction “Comparability of test cases regarding effort” is not met\(^{19}\), the model is discarded.

Step 2 is not carried out because no model is compatible with domain restrictions, organizational goals and restrictions and project goals, restrictions and parameters.

When discussing the results, all four participants in the selection process decide to try the framework again in another pilot project as soon as additional test effort estimation models can be considered which have been applied successfully in practice. Moreover, instead of the first level, pre-selection shall be based on elimination criteria.

To demonstrate the framework’s second step on the third level, the five models have been analyzed by means of the criteria D to H (see table 2). If, for example, “usability” is chosen as eliminating criterion in the final selection, the model by *Boehm et al. (2000)* has to be selected. If instead “project control” is chosen as eliminating criterion, the consistent decision is to select the model by *Cangussu et al.* \(^{20}\).

7 CONCLUSION AND FURTHER RESEARCH

Despite the exemplary confirmation of applicability, the framework exhibits several limitations.

1. Theoretical: Identification of existing models and requirements of such models is by no means complete, and the models’ analysis contains subjective standards.

2. Practical: The time required is considerable. Besides, complexity, dynamics and humans as non-deterministic main influencing factor of the test effort give rise to skepticism that a generally valid test effort estimation model and thus its portability are feasible.

However, the focus here is on the selection procedure, and no claim of completeness regarding the components of the selection (models and requirements) is stated. Instead, application of the framework requires situational adaptation of these components and multiple applications, so that the required time can be reduced.

The case study represents a possible first trial of the framework. It describes a typical project by a global standard business software supplier, and the data can be gathered based on interviews as well as careful evaluation of various documents. For corroboration of the framework’s fitness for purpose empirical testing is necessary. As indicated in section 6, especially models that are falsifiable and have been successfully applied need to be available for selection.

Thus, need for further research covers analytical models that try to explain the test effort, as well as forecasting models that are based on analytical ones and can be applied in different organizations.

References


\(^{19}\) Developers as authors frequently intend to prove correctness of the implementation of the requirements, whereas quality assurance representatives aim at uncovering errors. Moreover, the effort for running a test case depends among other things on automation. Complexity of the test case, skills of the tester, necessary test data, stability of the application, kind of errors present, delay until an error is eliminated, etc.

\(^{20}\) Due to limited space we refrain from exemplary illustration using another decision making technique.


PMBOK (2000), Project Management Institute Body of Knowledge (PMBOK). Project Management Institute, Four Campus Boulevard, Newton Square, PA, USA.


