Selecting engineering techniques using fuzzy logic based decision support

Peter Liggesmeyer
 Siemens AG, Corporate Research and Development, Software & Engineering
 Otto-Hahn-Ring 6, 81730 Munich, Germany

Abstract

The task of selecting software engineering methods, techniques, metrics, and tools is usually performed manually, based on the expertise of individuals. This paper presents a systematic tool supported approach, that bases its suggestions on the technical situation, the existing goals, and constraints of a specific organization or a particular project. A prototype of the decision support system supports the elaboration of test strategies. The approach uses information about the technical situation that is provided by answering predefined questions with fuzzy data. The objective is to assign "adequacy values" to combinations of test methods, techniques, metrics, tools, and quantified test situations. The priorities of goals and constraints are assessed by applying a technique that is based on comparing goals in pairs. This permits to check certain consistency criteria by static analysis. A hierarchy of the importance of goals and constraints is calculated, which provides the basis for the determination of the suitability of test methods, techniques, metrics, and tools with respect to goals and constraints.

Introduction

Many companies have installed corporate programs that are focused on quality or productivity improvement or reduction of the development time [1, 2, 3]. Improvement of software development means identification, evaluation and large scale introduction of appropriate practices and software engineering techniques. The task of selecting specific improvements in software development is usually performed manually, based on the expertise of individuals. This is neither a systematic approach nor is it likely that many experts that know enough about the various software engineering domains are available. This paper presents an approach for the systematic identification of suitable software engineering methods, techniques, metrics, and tools. It is based on a combination of automatic and interactive steps The automatic steps are performed by a decision support system that uses fuzzy logic. The suggestions are based on the technical situation, the goals, and the constraints of a specific organization or a particular project. A prototype of the system that supports improving testing is available.

Most software engineering domains are characterized by numerous methods, techniques, metrics, and tools that could be used. A method is the basic paradigm that is used to handle a certain task and that is usually refined into specific techniques. Techniques are characterized by a level of precision that allows to apply them to a particular problem and which can be supported by a tool. Methods for the software analysis phase are, e.g., functional decomposition or object oriented analysis, which are refined into techniques as Structured Analysis (SA), Real Time Analysis (RT) or Object Modeling Technique (OMT), Object Behavior Analysis (OBA), etc. Software testing methods are, e.g., functional testing, control flow testing, data flow testing or diversified testing that are refined into techniques as state-based testing, equivalence partitioning, cause effect graphing, branch testing, all uses testing, back-to-back testing, mutation testing etc. [4]. Metrics that are used to control the development process on a quantitative basis or to measure product properties are usually desired in addition to methods and techniques. In software testing, typical metrics are functional or structural coverage rates or the mean time to failure (MTTF). In many cases methods and techniques imply certain metrics. An appropriate combination of methods and techniques with respect to a certain development situation guarantees effectiveness, i.e., it guarantees that methods and techniques fit the task. Suitable metrics provide quantitative data as a mechanism for process control and project management. Tools usually support techniques and metrics. They guarantee efficiency concerning the application of methods, techniques, and metrics.

Technology selection

The majority of research in software engineering is focused on technology. Since a few years, the interests in process oriented approaches increase. The basic idea of process orientation is, that technical means alone cannot always provide solutions for problems, and that processes and people must be taken into account. The process describes how things are performed, controlled, and
managed. A suitable process is an appropriate basis for introducing techniques successfully.

A common approach for selecting software engineering techniques is to install a team that identifies modifications of the process and of the technology, e.g., based on the results of a software process assessment [5]. This is necessary, because the level of detail of the assessment results does usually not allow to determine specific improvements directly. The improvement suggestions can be organizational practices as well as technical modifications. There are a variety of methods, corresponding techniques, metrics, and various types of tools that could be used in most software engineering domains. Whether suitable solutions are selected, depends primarily on the expertise of the team, if the selection is performed manually. For this reason, a method for systematic decision support and a corresponding tool were developed. The tool partially automates technology selection. At present, the prototype of the tool supports selecting test methods, test techniques, corresponding metrics, and test tools. It is likely, that the approach can also be applied to other software engineering domains.

Systematic decision support for test improvement

Experience shows that software testing is a time-consuming task during software development. Furthermore, testing is located at the end of the development process, and is usually characterized by tight schedules. Almost every organization that develops software uses testing. On the one hand, companies spent much effort for testing. On the other hand, many of them observe insufficient software quality, which causes, e.g., high maintenance costs. Testing is an area that usually turns out to have a high potential for improvement, but it is difficult to identify those test methods, techniques, metrics, and tools that fulfill technical requirements as well as goals and constraints.

Previous work [6,7,8] describes a metric driven approach for the selection of unit test techniques that is primarily guided by technical properties of the program under test. To be applicable to real world testing, the concept must be extended with respect to integration testing and system testing. It is necessary to base the decision about test improvements also on the goals of the organization. Furthermore, it is essential to take into account the existing constraints. In practice, a real test environment contains methods (e.g., functional testing, structural testing, ...), techniques (e.g., equivalence partitioning, branch coverage testing, ...), metrics (e.g., coverage metrics, MTTF, ...), and tools (e.g., regression test tools, test coverage tools, ...).

The objective is to elaborate an approach that can be applied to all test phases, that covers methods, techniques, metrics, and tools for testing, and takes into account technical properties as well as goals and constrains.

The knowledge about the suitability of test methods, techniques, and about the appropriateness of test metrics and test tools is sparse. There exist of course some empirical [9, 10, 11, 12, 13, 14] and also some formal evaluations [15, 16, 17, 18, 19, 20, 21] of test adequacy criteria, but these data do not always fit the test situation in practice.

Proficient decision processes are not flat. They are structured; and usually they are hierarchical. For this reason, the decision process identifies test methods first and test techniques, test metrics, and test tools afterwards.

The majority of the relevant properties are not precise. If numerical data are available, their values are usually not very exact. Furthermore, some data are qualitative rather than available as metric values. The decision support method provides mechanisms to state the relevant properties of the specific test situation more precisely and to quantify them. This is performed separately for technical properties ($S_T$), goals ($S_G$) and restrictions ($S_R$). Furthermore, it is required, that relevant properties of the test methods ($P_{T}$), techniques ($P_{T}$), metrics ($P_{M}$), and tools ($P_{W}$) are represented systematically. This is the basis for the development of a function that assigns "adequacy values" to combinations of test methods, techniques, metrics, tools and formalized test situations:

$$f_{T}: ((\mu, T, M, W) \times S_T) \rightarrow [0; 1]$$
$$f_{G}: ((\mu, T, M, W) \times S_G) \rightarrow [0; 1]$$
$$f_{R}: ((\mu, T, M, W) \times S_R) \rightarrow [0; 1]$$

Decisions based on the current technical situation

Technical aspects, e.g., the completeness and quality of specifications, the structure and availability of the code, influence the suitability of the test procedure. Information about technical aspects is provided by the user as the input to the decision process. Since the data are usually fuzzy, they are processed by fuzzy logic rules to determine the appropriateness of the various possible test improvements. Using fuzzy logic has three major advantages:

- First, fuzzy logic can handle imprecise information, which is the case for the majority of the data used within the decision process.
- Second, the available knowledge about testing is more or less a set of rules, which are fuzzy by nature (e.g., measuring the structural test coverage is a good approach in early test phases).
- Third, the application of fuzzy logic rules produce a hierarchy of solutions. The result is not one optimal
test strategy; it is a hierarchy of strategies on an ordinal scale. This is beneficial, because a model-based approach as the one described here will never be able to take into account all necessary criteria. For this reason, it may happen, that practitioners refuse to use the best approach identified by the decisions support system. In this case, it is helpful if a hierarchy of solutions provides alternatives.

One fundamental question of the test engineer is: How adequate are the test methods, test techniques, metrics, and tools with respect to the current technical situation?

The first step of the decision process is to provide information about some specific pre-defined technical properties of the test situation. This is done by answering a number of questions on a scale from 0 and 1. Zero represents the Boolean value false, 1 represents the Boolean value true. Any value greater than 0 and less than 1 represents fuzzy information.

Example: The question "Is it possible to identify the relation between specification units and code units?" may be, e.g., assigned the value 0.9, because for 90% of the code units there exists a one-to-one relation to the corresponding specification units. The relation is unclear for the remaining code units. The value 0.9 can be interpreted as the degree to which the specification units and code units in this test situation are characterized by the property that it is possible to identify the relation between them. This information is usually called the value of the membership function [22].

Most properties of the test situation are fuzzy by nature. The necessary information about the test situation is provided manually by selecting values between 0 and 1 as answers to the set of pre-defined questions. There are separate sets of questions for selecting methods, techniques, metrics, and tools. The objective of this step is to map technical properties of the test situation to numerical values: \( f_{ST} : ST \rightarrow [0 ; 1] \)

As a means for test decision support based on technical properties a function \( f_{ST} \) is required which assigns a "suitability value" to test methods, techniques, metrics, and tools with respect to technical properties of the test situation \( S_T \). Since the interval \([0 ; 1]\) is used to represent information about the test situation, using the same numerical representation for suitability is reasonable. Zero means "completely not suitable"; 1 means "ideally suitable".

\[
f_{ST} : (\{T \cup M \cup W\} \times S_T) \rightarrow [0 ; 1]
\]

Since the properties of test methods, techniques, metrics, and tools are constant and the only alterable aspects are properties of the test situation, we can expect, that all parameters of the technical suitability function are test situation properties. The suitability of a test method, technique, metric, or tool \( x \) can be determined from the \( k \) technical properties \( (S^k_T) \) of the test situation as:

\[
f_x = f(\{S^1_T, \ldots, S^k_T\})
\]

This function assigns a value to a test method, technique, metric, or tool \( x \) that represents its suitability with respect to aspects \( S^k_T \) of the test situation represented by numerical values between 0 and 1. Since the knowledge about testing exists basically as a set of rules, this function is implemented by a set of fuzzy logic rules. On the one hand, there are standard fuzzy logic operations that replace the Boolean operations "AND" and "OR" [22]. The membership function of the conclusion of a conjunctive if-then-rule is defined as the minimum of the premises' membership functions ("AND"). The conclusion's membership function of a disjunctive if-then-rule is defined as the maximum of the premises' membership functions ("OR").

Example: If the values of the membership functions of premises A and B are 0.3 and 0.8 the corresponding membership function of conclusion C is assigned the following values:

If A AND B then C; C is assigned the value 0.3  
IF A OR B then C; C is assigned the value 0.8

On the other hand, it is usually required that fuzzy logic operators are appropriate models of the corresponding "human operators". This includes the ability to assign a weight to premises as well as the capability to compensate for a low value of one premise by a high value of another premise. If we assume, that the two premises of a conjunctive rule are not equally important and that the more important premise has a higher membership function value than the less important premise, assigning the conclusion the minimum value seems to be unreasonable. Various fuzzy logic operators that take these aspects into account have been published. In this approach it has been decided to use the following function as the "AND"-operator:

\[
f_y ((w_1, a_1), (w_2, a_2)) = \gamma \text{MIN} (a_1, a_2) + (1-\gamma) (w_1 a_1 + w_2 a_2) / (w_1 + w_2)
\]

\( w_1 \) represents the weight of the premise with membership function value \( a_1 \). \( w_2 \) represents the weight of the premise with membership function value \( a_2 \). \( \gamma \) is a parameter that determines the type of the operator. If \( \gamma \) is assigned the value 1 then \( f_y \) becomes the standard fuzzy "AND"-operator and the premises' weights are ignored. If \( \gamma \) is assigned the value 0 then \( f_y \) becomes a combination of the standard fuzzy "AND" and the fuzzy "OR"-operation.

Since rules that determine the suitability of the same technique describe alternative situations, the standard
fuzzy "OR"-operator is used to calculate the suitability of the technique. This means, that the final value of a technique's membership function is the maximum of the values determined by the applicable rules. The decision support system contains, e.g., the following rule:

IF (the test phase is early (w = 1))
AND (the specification is available (w = 2))
AND (the software structure is modular (w = 1))
AND (there is a relation between specs and code (w = 1))
AND (the code is available (w = 2))
THEN
(A combination of functional testing and structural testing is suitable)

We assume that the following membership function values have been assigned:

Test phase is early: 1.0
The specification is available: 0.9
The software structure is modular: 0.9
There is a relation between specs and code: 1.0
The code is available: 1.0

Since for this rule the value of $\gamma$ is 0.75, the suitability of combining functional and structural testing is assigned the value 0.91.

Decisions based on goals

It is necessary to take into account the goals. They represent future requirements for the test process, as well as existing test goals in the current situation. Goals with respect to the test process are, e.g., low error rate, transparency of the test process, reproducibility, and comprehensive coverage of functionality and code. In this approach the priority of eleven potential test goals must be determined by the user. It is likely that practitioners have difficulties to assess the priorities of these goals with systematic support. For this reason, the approach used to determine the priority of the goals is similar to QFD which is a technique used to evaluate the importance of software requirements or system requirements [23, 24, 25, 26]. This task is similar to assessing the priorities of test goals, since the major problem in both cases is vagueness. QFD’s objective is to guarantee that in case of, e.g., resource problems the resources are invested into the important requirements. QFD determines priority on a numerical scale. The approach is to compare each pair of requirements and to determine their relation with respect to importance.

A modified version of this approach is used here to assess the importance of the test goals, which provides the basis for the determination of the suitability of test methods, techniques, metrics, and tools with respect to goals. This part of the decision process answers the question: Which test methods, test techniques, metrics, and tools achieve the goals?

A matrix representation is used to store these data (table 1). If a goal that labels a line of the matrix is judged to be more important as a goal in the column, the corresponding matrix element is assigned the value 2. If it is judged to be less important than the goal in the column, the matrix element is assigned the value 0. If the goals are judged to be equally important, the matrix element is assigned the value 1. The sum of the values of the matrix elements in one line can be used as a metric for the total importance of the corresponding goal. These weights are normalized to make sure they add up to 1.

Table 1. The matrix for assessing the priorities of goals

<table>
<thead>
<tr>
<th>Goal</th>
<th>Goal 1</th>
<th>Goal 2</th>
<th>Goal 3</th>
<th>Goal n</th>
<th>$\sum (g_0) \cdot 1$</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal 1</td>
<td>$g_{0}=1$</td>
<td>$g_{0}=2$</td>
<td>$g_{0}=2$</td>
<td>$g_{0}=2$</td>
<td>$(E_{G1})=1-18$</td>
<td>0.16</td>
</tr>
<tr>
<td>Goal 2</td>
<td>$g_{0}=2$</td>
<td>$g_{0}=0$</td>
<td>$g_{0}=0$</td>
<td>$g_{0}=2$</td>
<td>$(E_{G2})=1-8$</td>
<td>0.07</td>
</tr>
<tr>
<td>Goal 3</td>
<td>$g_{0}=0$</td>
<td>$g_{0}=1$</td>
<td>$g_{0}=1$</td>
<td>$g_{0}=2$</td>
<td>$(E_{G3})=1-6$</td>
<td>0.05</td>
</tr>
<tr>
<td>Goal n</td>
<td>$g_{0}=0$</td>
<td>$g_{0}=0$</td>
<td>$g_{0}=0$</td>
<td>$g_{0}=1$</td>
<td>$(E_{Gn})=1-2$</td>
<td>0.02</td>
</tr>
</tbody>
</table>

The data to which degree a test method, test technique, test metric, or tool fulfills each goal are represented as attributes of the methods, techniques, metrics, and tools in the corresponding decision support system. This is done using a numerical scale. By multiplying a normalized goal priority and the corresponding attribute value of a specific technique, the suitability of this technique with respect to the importance of that goal can be calculated. The sum of all these products concerning all goals represents the total suitability of that technique with respect to the goal priorities.

It is beneficial to recognize that the comparison of goals in pairs permits checking transitivity. If the user’s input is, e.g., goal B is more important than goal A, goal C is more important than goal B and goal A is more important than goal C, then this input violates transitivity. For this reason, the decision support system checks transitivity and accepts the goal priority matrix only if it fulfills transitivity completely.

Decisions based on constraints

In practice, the existing constraints cause a major influence on the test process. Their relative importance is determined by comparison of the importance of constraints and goals in pairs. Typical constraints are,
e.g., cost, time, and availability of tools. This part of the decision process answers the question: Which test methods, test techniques, metrics, and tools fulfill the goals and the constraints?

The degree to which a specific test technique violates a certain constraint is represented by assigning a value between 0 and 4 as an attribute. Zero represents no violation; 4 represents strong violation of a constraint. The assignment of attribute values has been based on, e.g., the relative strength relations between test techniques (i.e., a technique A that includes another technique B is stronger but requires more effort). Adding the products of the constraints' priorities and the corresponding attribute values of every test technique generates a metric for the degree to which each test technique violates the constraints.

**Tool support and case study**

Some key features of the decision support tool TDS (Techniques Decision Support) which at present supports selecting test strategies are described in the following section. The situation is fictitious but nonetheless realistic. The first step is providing the necessary information for determining the suitability of test methods based on the technical situation. Figure 1 displays a part of the tool's user interface which contains the input data on the left and the conclusions concerning the suitability of the test methods drawn from these data on the right.

In this case, a combination of functional and structural testing is the best approach. The tool is capable to explain which rules have been used to determine suitability values. For this reason, it is very easy to figure out, that the suitability of functional testing combined with structural testing (0.91 or 91%) has been determined from the information about the test phase, the availability of appropriate specifications, the modularity and availability of the code, and the relation between code and specifications.

If the user accepts the combination of functional testing and structural testing, selecting test techniques is the next step. Investigating the suitability of test techniques requires additional input data. Since techniques are usually refinements of methods, their suitability is also influenced by the suitability of the corresponding method. Figure 2 displays a section of the tool's user interface which deals with determining the suitability of techniques. The upper left part of figure 2 contains the suitability of the test methods determined in the previous step as one input of this decision step. The lower left part of figure 2 contains additional input data provided by the user. The upper right part of the figure displays the suitability of the test techniques as the result of this step. Based on this information, branch coverage testing could be selected as an appropriate structural technique. Equivalence partitioning could be used as the functional technique.

Developing a complete technical solution that consists of methods, techniques, metrics, and tools.
The decision support tool does not automate such decisions, but it provides information to base the decisions on.

Experience and conclusions

The approach allows to identify complete solutions for testing, because the suitability of methods, techniques, metrics, and tools can be determined for all test phases. Furthermore, the current technical situation influences the selection as well as goals and constraints. It is visible, why certain techniques are not suitable, and which modifications have to be made to make them suitable. It is clear, whether goals are unrealistic under given constraints, and the decision, whether goals or constraints have to be modified, can be based on these data. At present, the tool TDS is evaluated in test improvement projects.

The method and the tool for the selection for test method, test techniques, metrics, and test tools are focused on supporting the decision process. For this reason, it is not desired that the decision is completely automated. The goal is to achieve a systematic decision process that strives to develop a good test strategy efficiently. Furthermore, the method permits to explore various scenarios. Another advantage is that due to quantifying the importance of goals and constraints unrealistic constellations of goals and constraints are revealed. Industrial applications of the tool demonstrate that selecting appropriate test methods, test techniques, test metrics can be performed at short notice. Going through the whole procedure is usually possible in a single day. This is a significant reduction of the required time and effort compared to the manual decision process, that usually requires numerous days. This effect is, e.g., caused by avoiding redundant discussions. The tool divides the complex task of selecting methods, techniques, metrics, and tools into a sequence of simple and small steps. This provides a guideline for the decision process. Numeric results are an additional key to enhancing transparency, as well as the tool's capability to explain its conclusions, and the possibility to explore scenarios.

Generally, people have fewer difficulties to provide the information about the technical situation as input to the decision support system. The information about the priorities of goals and constraints is usually appreciated, since it was generally not available before. The experience of applying the approach in industrial practice demonstrates that people are willing to accept the systematic approach. The major advantages of the method are:

- efficiency (i.e., only minor amount of time and effort required),
- transparency (it is clear why certain suggestions have been generated),
- interaction to the tool,
- the ability to simulate various scenarios (hypothesis can be explored),
- the quantified importance of goals and constraints.

Further investigations will focus on adding additional techniques to the existing decision support system (e.g., reviews), and on adapting the method to other engineering domains. Most engineering domains are characterized by methods, techniques, metrics, and tools. In every real project technology must fit the technical situation as well as goals and constraints. At present, TDS covers all test phases including system testing. In general, this requires to develop test strategies for systems that are composed of software and hardware. At present, experience does not indicate any specific problems concerning these applications of the method.

Additional investigations focus on enhancing the tool to be able to generate, e.g., cost-benefit data. This means that predictions about the effects of a new technique should be possible on a numerical basis.

References

5. Humphrey W.S., Managing the software process, Addison-Wesley, Reading 1989, 1990
6. Liggesmeyer P., Selecting adequate test techniques in the real world, Proc. 11th International Conf. on Testing

433


10. Gannon C., Error detection using path testing and static analysis, Computer, vol. 12, no. 8, 26-31, August 1979


