www.ietdl.org

Published in IET Software
Received on 20th September 2011
Revised on 21st September 2012
Accepted on 22nd October 2012
doi: 10.1049/iet-sen.2011.0170

ISSN 1751-8806

Does software error/defect identification matter in the Italian industry?

Giuseppe Scanniello[1], Fausto Fasano[2], Andrea De Lucia[3], Genoveffa Tortora[3]

[1]Dipartimento di Matematica, Informatica e Economia, University of Basilicata, Potenza, Italy
[2]Department of Bioscience and Territory (DIBT), University of Molise, Pesche, Italy
[3]Department of Mathematics and Informatics, University of Salerno, Fisciano (SA), Italy
E-mail: giuseppe.scanniello@unibas.it

Abstract: The authors present the results of a descriptive survey to ascertain the relevance and the typology of the software error/defect identification methods/approaches used in the industrial practice. This study involved industries/organisations that develop and sell software as a main part of their business or develop software as an integral part of their products or services. The results indicated that software error/defect identification is very relevant and regard almost the totality of the interviewed companies. The most widely used and popular practice is testing. An increasing interest has been also manifested in distributed inspection methods.

1 Introduction

In traditional engineering disciplines, the artefacts built in the production process are checked to guarantee a level of quality for the final developed product. In particular, verification/validation activities are performed to identify and then remove defects. Similarly, in the software engineering field artefacts are checked during the whole life cycle to produce and deliver high-quality software products [1].

Today several methods and techniques are available and used to assess and maintain the quality of software artefacts and to check that they meet specifications and fulfill their intended purposes. Many of these methods and techniques have been largely experimented in case studies and controlled experiments [2, 3]. Systematic reviews and state of the art surveys are also available in the literature [4–7]. Only a few numbers of descriptive and explorative surveys have been conducted (e.g. [8, 9]) to ascertain error/defect identification methods.

Survey research is an important and useful method for collecting data from respondents (e.g. companies, unemployed, customers of a company and so on) [10]. Among the widely used types of surveys, a descriptive survey takes a picture or document the current conditions or attitudes to discover the situation in a fixed period for a given area [11]. Descriptive surveys are useful because allow selecting from a huge number of industries a smaller and focused subset, and observing them with 'insight' that might tell more than all the multitudes together. Conversely, this kind of research cannot describe what caused a given situation and results are always open to question and to different interpretations. For example, an unemployment survey might measure the number of people without jobs or that have looked for work in the past month, and are able to start working in the next 2 weeks. What caused an unemployment cannot be answered from this kind of study. In spite of the limitations above, a descriptive survey is an invaluable scientific tool [11].

In this paper, we present the results of a descriptive survey organised by three Italian Universities, that is, University of Basilicata, University of Molise and University of Salerno, to ascertain the state of the practice of software error/defect identification in the Information Technology (IT) Italian industry. The survey was conducted between spring 2008 and winter 2009. We invited 70 companies/organisations that develop and sell software as a main part of their business or develop software as an integral part of their products. We received by e-mail 30 fully completed questionnaires from key people of the invited companies/organisations. The main goal of our survey was to comprehend the relevance of: testing, inspection, distributed inspection, pair inspection and walkthrough. We were also interested in analysing the main problems that may limit the adoption of these methods/approaches.

The remainder of the paper is organised as follows. We discuss background and related work in Section 2. In Section 3, we present the design of the study, whereas the analysis of the gathered data is highlighted in Section 4. The threats that may affect the validity of the achieved results are discussed in Section 5. A discussion of the results and possible future directions for our research conclude the paper.

2 Background and related work

Detecting defects in software artefacts requires serious effort, so it is important to use the most efficient and effective methods [12]. Empirical studies can help software practitioners to comprehend the relevance of these methods and to decide the methods to use and for what purpose [13]. For example, we presented in [14] the results of two controlled experiments to compare the Fagan's software inspection process [15], distributed inspection [16] and pair inspection [17] in the identification of defects in source code. The results suggested that: using pair inspection significantly reduces the time to detect defects, the correctness of the identified defects improves when employing the Fagan's process, and the completeness of the recovered defects is mostly the same whatever the method used is. In the descriptive survey presented here, we were interested in comprehending the relevance of these methods (considering also testing and walkthrough) for the IT Italian industry. We conducted this study to obtain indications on whether and how to transfer the considered methods (including that developed in our research laboratory, namely distributed inspection) to the industry. In the following subsections, we first provide background and then related work concerning the use of survey to analyse the adoption of processes, methods and approaches for the identification of error/defect in software artefacts [8, 9].

2.1 Background

The choice of a method depends on several factors (e.g. the artefacts, the types of defects they contain and so on) [12]. For example, testing is a practice aimed at identifying defects within software [1]. In particular, this technique is considered as the process of validating and verifying that a software: meets the requirements, satisfies the needs of the stakeholders and works as expected. Testing process is composed of many phases from unit to system testing. Whatever is the phase of testing process, an executable representation of the source code is needed.

Differently from testing, inspection is applicable to any kind of software [15, 18]. The first conventional inspection process was proposed by Fagan [15], who defined software inspection as a formal, efficient and economical method of finding errors in design and code. The process is composed of five sequential phases: overview, preparation, inspection, rework and follow up. In the overview phase the author of the software artefact describes and provides details about this artefact. The documentation of this artefact is distributed to all the members of the inspection team in the preparation phase. During the inspection phase a face-to-face meeting is carried out to identify the defects. In the rework phase, the moderator of the meeting produces a written report for the author of the artefact, where the defects to be removed are reported. Finally, in the follow-up phase the moderator checks the quality of the new version of the artefact and determines whether a re-inspection is required.

Distributed inspection methods are mostly based on Fagan's process (e.g. [3]). The main difference with respect to that process is that the interaction among the members of the inspection team and the author of the artefact is not face-to-face, but supported by synchronous and asynchronous communication tools [7]. Several tools are available in the literature and many of them are web based (e.g. [3, 16]).

Walkthrough is a form of formal software peer review [19]. To identify defects within software artefact, this method exploits five steps: (i) the author of the artefact selects the review participants, obtains their agreement and schedules a walkthrough meeting; (ii) the author distributes the artefact prior to the meeting; (iii) the author describes the artefact and leads the discussion; (iv) the reviewers present the comments, the possible defects and the improvement suggestions to the author; and (v) based on the reviewers' comment the author performs any necessary rework of the artefact. Walkthrough is generally applied on source code.

Pair inspection [17] requires only two participants: the author and an inspector who reviews the author's artefact. Pair inspection requires continuous iterations, so any strict rules can be formulated to guide the inspection. Owing to the iterative nature of the process, the recording of defects found in informal meetings is not required. A form or a template is adopted to annotate the defects of the software artefact under revision. In some cases the defects are typically marked on paper documents and corrected during the next correction cycle.

2.2 Related work

There is a growing interest towards survey investigations to study and analyse the use of processes, methods and tools within the development and maintenance of software products (e.g. [7, 20–22]). For example, Torchiano *et al.* [22] reports on a state of the practice survey conducted among 59 Italian software companies. This survey is conducted within a research project [23] and aims at analysing the state of the practice in software migration. The results of the survey indicate that about 66% of the interviewed companies have some experiences in migration tasks. The study also highlights the lack of tools for the execution of migration tasks. This however does not seem to constitute a problem for the interviewed companies. We used an approach similar to that employed in [22] (e.g. a questionnaire-based survey), but the subject of the investigation is different.

Differently, a few number of survey investigations have been conducted to study and analyse the use of processes, methods and approaches for the identification of error/defect in software artefacts [8, 9]. For example, the survey presented in [8] is focused on the current state of the practice in industry regarding the application of reviews and inspections. In particular, the major focus is on the concrete application of walkthrough, peer review and software inspection (i.e. variants of the Fagan method). The results indicate that there are still many objections against the usage of the techniques considered in the study. The main concern is that these techniques are perceived as too time consuming and then not applicable in practice. The results are in line with ours, so showing that industry is not ready yet to apply reviews and inspections approaches. Differently from us, this state of the practice surveys is only focused on walkthrough and software inspections.

Jedlitschka *et al.* [9] conduct an empirical research to understand the needed information to increase the likelihood of successful technology adoption. To this end, the authors carry out a survey to investigate the state of the practice of inspection in German software industry decision makers. They involved 92 companies and observed that information regarding the impact of technologies on product quality, cost and development time, as well as on technology cost–benefit ratio is considered highly relevant for the interviewed decision makers. Differently from us, the goal of that study is to understand the needed information about a new developed technology and how this information has to be presented to facilitate the technology transfer to the industry.

3 Definition and design

Surveys are investigations to gather data from respondents (i.e. key employees identified by the management of the involved companies), using a questionnaire composed of closed or open questions [11]. Depending on the survey purpose, it may focus on opinions or factual information [24]. Data can be collected by: face-to-face and phone interviews, mail, e-mails and web pages. E-mail surveys are both very economical and very fast. They are often preferable for sensitive items, and there is no interviewer bias. On the other hand, email surveys are limited to simple questionnaires. The data can be analysed to derive descriptive and explanatory conclusions [25] that are applicable only to the selected population.

In this paper, we have conducted a descriptive survey based on both opinions and factual information to ascertain the relevance and the typology of the software error/defect identification methods/approaches used in the practice of IT Italian industry. We also collected opinions to better plan the subsequent future empirical investigations. Thus, once the scope and the purpose of the survey have been defined, we have reviewed the available literature in the area of empirical software engineering and then we have decided to adopt questionnaire-based survey to gather the data from the respondents and to collect them by e-mail.

We defined the survey to address the following two goals:

Primary goal: Comprehending the relevance of different software error/defect identification methods/approaches used in the Italian industry.

Secondary goal: Identifying the main problems and the actual needs (methods, techniques and tools) of Italian industry.

According to the primary goal, we have defined and investigated two research questions:

RQ1: What is the relevance of error/defect identification in the Italian industry?
RQ2: What are the most popular and widely used approaches/methods in the Italian industry?

With respect to the secondary goal, we have defined and investigated the following research question:
RQ3: What are the main problems related to the use of approaches/methods for software error/defect identification?

We have defined a further research question to investigate whether the Italian software industry is ready or interested in applying the practices for software error/defect identification considered in this survey (i.e. testing, inspection, distributed inspection, pair inspection and walkthrough). The research question is the following:
RQ4: Is the Italian software industry interested in using and adopting new practices for software error/defect identification?

3.1 Conceptual model

We identified three areas of interest to collect the data:

Demographic information: we collected demographic information about the respondents and their company/organisation ('company' in the following).

Relevance and typology: we collected information about the relevance and quantity of projects in which the company used the software error/defect identification methods considered in the survey (simply 'methods', 'approaches', or 'methods/approachs' in the following).

Main problems and needs: we collected information about the issues that limit the adoption of approaches and methods for error/defect identification.

Regarding the demographic information, for each company we collected: business domain; size of the company and respondent's group/business unit; typical duration of a project and its kind; average experience and skill of group members. In addition, we collected information about the respondents, such as: age, gender, educational qualification and role (e.g. IT manager, project manager).

Each respondent was also asked to specify whether his/her company had used software error/defect identification methods, approaches and tools. If the company never used approaches and methods, we asked the respondent to indicate whether and which practices they have planned to use. We also asked whether they could be interested in experimenting unknown practices, for example, in terms of special conceived empirical investigations (e.g. case studies). According to the respondents' answers, the companies were grouped in four groups, which were established before collecting and analysing the data from the respondents. The four groups are the following:

group 1: Used at least one of the methods/approaches (i.e., testing, inspection, distributed inspection, pair inspection, and walkthrough) and planned to adopt new methods/approaches;
group 2: used at least one of the methods/approaches and not interested in using new ones;
group 3: never used the methods/approaches and planned to employ at least one of them in future projects;
group 4: never used the methods/approaches and not interested in using them.

We asked the companies in the groups 1 and 2 the percentage of projects in which they adopted the considered methods/approaches. Questions on how often they used these methods/approaches in the past projects were also asked. We asked some information about future projects to the companies in the groups 1 and 2. The respondents within the group 3 were also asked questions on managerial problems regarding the reluctance to use testing, inspection, distributed inspection, pair inspection and walkthrough in the past. For the respondents of the group 4 the questionnaires finished after collecting demographic information. It is worth mentioning that the companies within two different groups had to answer a different number of questions.

3.2 Identification of the target population and of the sample

The target population consisted of IT organisations that develop and sell software as a main part of their business (e.g. software house) or develop software as an integral part of their products or services (e.g. healthcare domain).

The selection of the companies ('sampling') has been conducted using the network contacts (for convenience and/or opportunity) of the authors' research groups. The contact network was created from previous research projects (e.g. [23, 26]). The contact network also included companies that: (i) host students from the Universities of Basilicata, Molise and Salerno for external stages or (ii) employ people who received a Master's or a Bachelor's degree at one of these universities.

doi: 10.1049/iet-sen.2011.0170

The companies of the contact network were invited by phone to obtain their availability to participate to the survey. An e-mail followed providing detailed instructions about the survey and the questionnaire.

3.3 Questionnaire design and data collection

We developed the questionnaire [The questionnaire (Italian version) is available at www.scienzemfn.unisa.it/scanniello/questionnaire.pdf] following the standard schema proposed in [8]. Fig. 1*a* shows the structure of the questionnaire. It contains both open (some requiring just to fillin in a comment or text) and multiple choice questions. According to the conceptual model, the questionnaire consists of four different paths (one for each group in Section 3.1) and three sections.

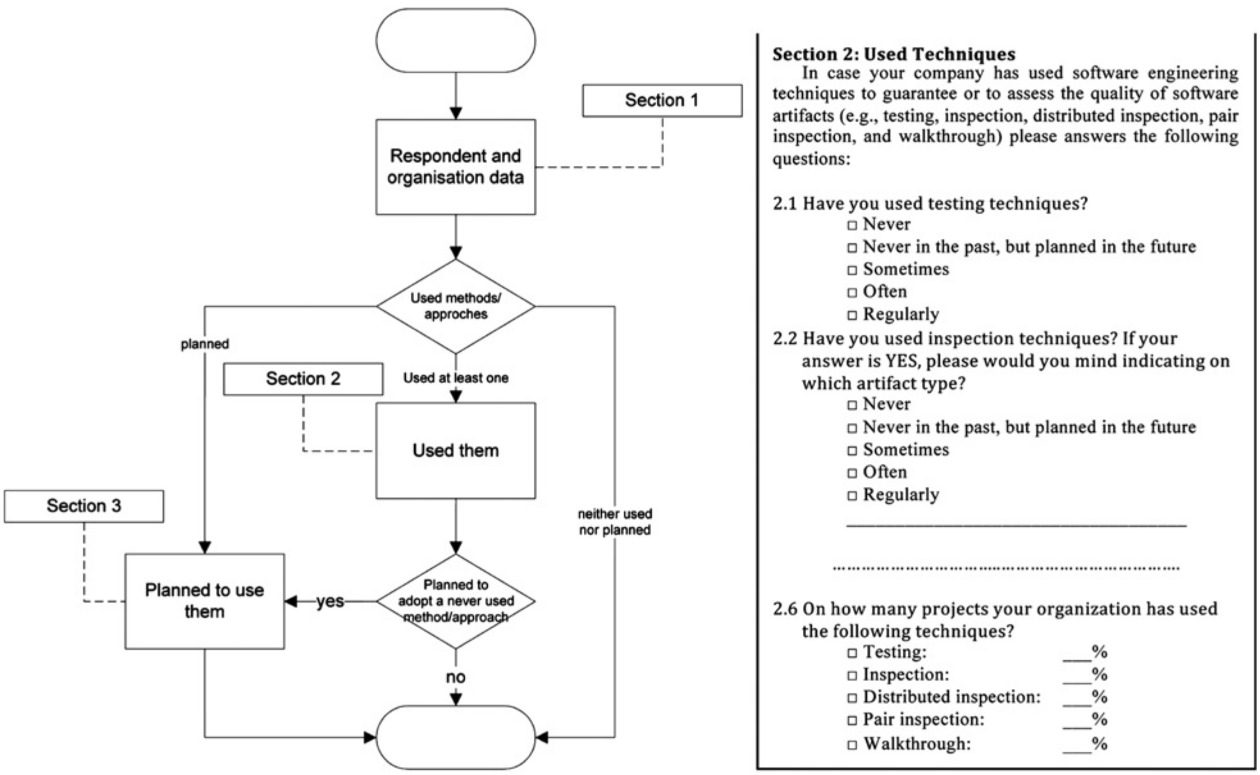
The first part (Section 1), common to all paths, contained ten questions used to obtain information on the respondents and their companies. Section 2 included eight questions (some questions are shown in Fig. 1*b*), whereas eight questions were in Section 3. The last question of Section 3 was introduced to obtain the availability of the respondents (and then of the companies) to use the methods and approaches studied in the survey.

The questionnaire was structured in such a way that the total number of questions depended on whether the respondent's company had already used or had planned to apply the methods/approaches studied in the survey. Overall, we deliberately restricted the number of questions to the essential to avoid burdening the respondents and to augment the response rate to the survey.

The questionnaire was introduced with a brief motivation sketching the general problem to be investigated. The questionnaire also included an introductory section in which we have clarified the meaning of the relevant terms (e.g. project and inspection) and describes all the entities of interest for the survey (see the appendix for details). The importance of this study and our objectives were inserted in an accompanying letter attached to the questionnaire. Great care was given to ensure ethical requirements and privacy rules imposed by the Italian regulations. We also clarified that all the information was considered confidential and that the data were used only for research purposes and revealed only in aggregated form.

We collected the answers to the questionnaire via e-mail. This decision was taken to reduce the costs. Collecting the answeres to the questionnaire by e-mail also reduced as much as possible interviewer bias [27]. However, the use of e-mail introduces the drawback that there is no control on respondents. For example, we do not have information concerning the time employed to fill the questionnaire in.

All the respondents interested in using at least one of the methods/approaches have been successively contacted by phone to obtain further information on the company. An unstructured interview was executed to obtain information on the used Software Configuration Management (SCM) [Software Configuration Management [28] encompasses the disciplines and techniques of initiating, evaluating and controlling change to software products during and after the software engineering process.] system. In case the respondents also used a distributed inspection process in the past, we asked information about the problems encountered. Otherwise, we asked them information on why distributed inspection processes has been never used. The rationale for deepening this aspect relies on the fact that this survey is a part of a research plan that is taking place over the years to



Section 2: Used Techniques

In case your company has used software engineering techniques to guarantee or to assess the quality of software artifacts (e.g., testing, inspection, distributed inspection, pair inspection, and walkthrough) please answers the following questions:

2.1 Have you used testing techniques?
- □ Never
- □ Never in the past, but planned in the future
- □ Sometimes
- □ Often
- □ Regularly

2.2 Have you used inspection techniques? If your answer is YES, please would you mind indicating on which artifact type?
- □ Never
- □ Never in the past, but planned in the future
- □ Sometimes
- □ Often
- □ Regularly

____

……………………………………………………

2.6 On how many projects your organization has used the following techniques?
- □ Testing: ___%
- □ Inspection: ___%
- □ Distributed inspection: ___%
- □ Pair inspection: ___%
- □ Walkthrough: ___%

Fig. 1 *Questionnaire design*

a Structure of the questionnaire
b Excerpt of the questionnaire

investigate the possibility of transferring the method proposed in [2, 14, 16] to the IT industry.

3.4 Survey preparation, execution and analysis procedure

To conduct the survey, we performed the following activities:

1. *Preparation and design of the questionnaire:* we used similar questionnaires to identify a set of questions.
2. *Invitation to participate:* the companies of our contact network were contacted by phone to obtain their availability.
3. *Data collection:* we sent an e-mail to the companies that positively answered to the invitation. This mail enclosed the questionnaire and provided detailed instructions on how to fill it in as well as information on the goals of the survey. One of the authors collected the answers to the questionnaire that the respondents returned by e-mail.
4. *Data analysis:* the answers to the questionnaire were analysed with respect to the goals identified and discussed above. Owing to the nature of the study, the data analysis is focused on descriptive statistics of the responses to the questionnaire.

Among the 70 invited companies, 48 furnished their availability to participate to the survey, 32 filled in the questionnaire and 2 of these companies wrongly filled in the questionnaire. Therefore the response rate was 43%.

4 Analysis

4.1 Respondents' background and companies' characteristics

The interviewed companies were 60% independent and 40% subsidiaries (i.e. controlled by a larger and more powerful company). Among these companies, 67% were private companies and 33% were quoted on the stock exchange.

Regarding the size, 53% of them were either small or medium-sized enterprises (i.e. <250 employees), whereas 47% were larger ones. All the large companies within the sample are in the area of software consultancy or IT services. These companies are organised into smaller business units, each of them allocated on a specific project/customer. In this setting, each business unit is independent from the other units of the same organisation, has a separate managerial structure, and is usually located on separate sites. It is worth mentioning that small companies might have only one business unit.

The size of the units were distributed as follows: 13% were micro (employees <10), 67% were small (employees between 10 and 50) and 20% were medium (employees between 50 and 250). There were no units with a number of employees larger than 250. The companies come from different industrial domains. Most of them worked in the area of software consultancy (40%). Conversely, 33 and 27% of the companies developed software and provided IT services to third parts, respectively. None of the respondents were from companies where Telecommunication was the main part of the business. The results are visually summarised in Fig. 2*b*. The distribution of the sample across the Italian territory is visually summarised in Fig. 2*c*. South and Centre are over-represented with respect to North. This aspect should be not a threat for the observed results because geographical and cultural differences among the companies are negligible. However, special conceived studies are needed to better investigate this point.

The typical size of the software systems handled/developed by the companies was: from 10 to 100 thousand lines of code (KLOCs) (7%), from 100 to 500 KLOCs (80%) and more than 500 KLOCs (13%).

All the companies that never used any of the considered methods/approaches planned to adopt at least one of them. These were 20% of the participating companies. On the other hand, 40% of the companies regularly employed one of the practices, whereas 13% used testing, inspection, distributed inspection and walkthrough. Finally, 27% occasionally used one of the methods/approaches object of the study.

As far as the respondents are concerned, their age ranged from 24 to 50 years old, with an average of 35 years. Only two respondents were female. Regarding the role of the respondents in the companies (see the pie chart of Fig. 2*a*

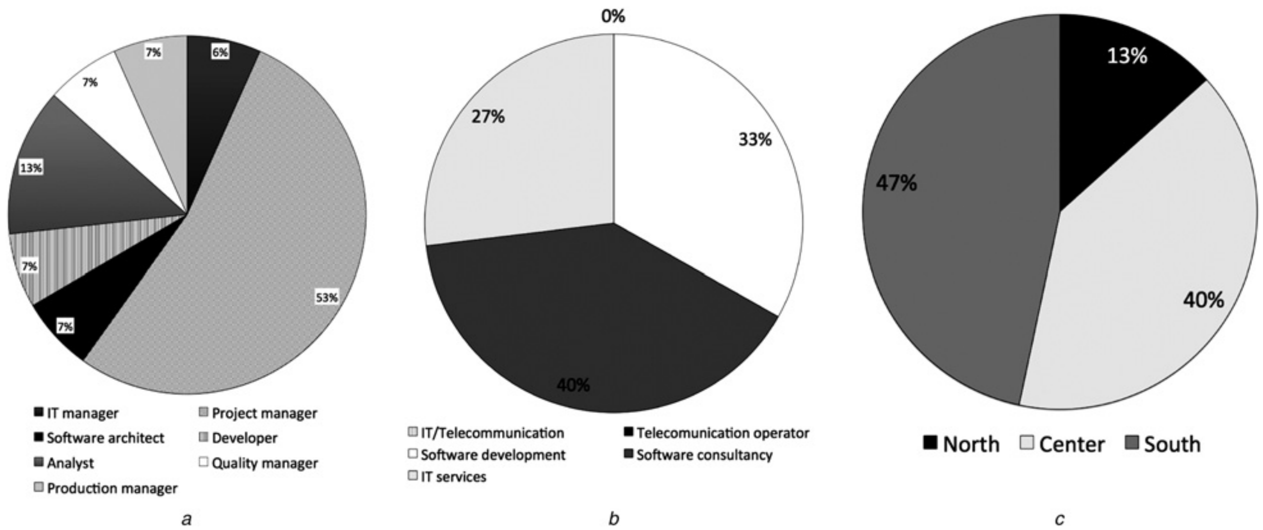


Fig. 2 *Visual summary of the respondents' background and companies' characteristics*

a Role of the respondents
b Industrial domain
c Distribution across the territory

for details), 73% of the respondents had management roles (i.e. projects manager, IT manager, quality manager or production manager) while 27% were developers or software architects. Only 7% of the respondents were quality managers. The reader may object to the fact that this result could affect the validity of the survey results because the primary person qualified to answer our questionnaire is an employee with the quality manager role. A plausible motivation for this mostly unexpected result is that in small and medium-sized software enterprises very often the role of quality manager is played by either IT managers, project managers or production managers.

As for the educational qualification of the respondents, 93% had a master degree, 7% had a bachelor degree. None had a PhD. Eighty percentage had a specific IT degree, whereas 20% had a non-specific IT degree.

4.2 Relevance of the used practices and influential factors

Among the companies that have used at least one of the methods/approaches (i.e. 24 out of 30), most of them (i.e. 14 out of 24) regularly used testing techniques, while eight companies often used them in the past. Only two respondents indicated that their company occasionally used testing. For these two companies, testing was however the most used technique.

Inspection methods (i.e., variations to the Fagan's process) were already used in two companies, whereas 10 out of 24 companies often used them. All these 12 companies often or regularly used testing as well. Inspection methods were occasionally employed within six companies. These company often or regularly used testing. On the other hand, four companies never used inspection methods even if they were interested in. Two of these companies used testing. Only two companies were not interested in using inspection methods in the future, but exploited testing.

The companies that rarely used distributed inspection were six. Moreover, six respondents stated that his/her company has never used distributed inspection, but it will be used in the future. Finally, 12 out of 24 companies never used distributed inspection and were not interested in using it.

Pair inspection was regularly used in 6 out of 24 companies, whereas two companies often used this technique. Pair inspection was occasionally used within eight companies. The respondents of eight companies stated that this technique was never used. Among them, six were not interested in using pair inspection.

Walkthrough was regularly used in four companies, while six companies often employed it. Walkthrough was occasionally used within ten companies (i.e. 42% of the cases). Finally, four companies never used this practice and were not interested in using it.

We also asked the respondents to indicate the approach/method considered: simplest, most effective, less expensive and with the best cost–benefit ratio. They recognised pair inspection as the simplest and less expensive method to identify defects within software artefacts (i.e. 12 out of 24). Testing was considered the most effective and with the best cost–benefit ratio. Table 1 summarises the responses provided.

Most of the respondents (16 out of 24) stated that the methodological aspect is the predominant factor to effectively identify defects and improve the quality of software artefacts. The human factor is the secondary concern (8 out of 24).

4.3 Main problems and needs

We first asked the respondents to indicate the practices they were interested in using among the never used ones [Respondents could indicate more than a method/approach.]. Sixteen respondents indicated distributed inspection. The respondents also manifested interest in pair inspection and in variants of the Fagan's inspection process (14 and 10, respectively). There were no respondents interested in testing and walkthrough. Regarding testing, this was due to the fact that it was already used. This was not applicable for walkthrough. The companies that did not use this method were not interested in adopting it in the future because they believed that it is not effective as compared with other inspection, pair inspection.

With regard to the inspection process proposed by Fagan, it was well known and many companies adopted it. The companies that did not adopt software inspection declared that the main problems (in increasing priority order) encountered were: lack of specialised employees, technique not properly known, technique too expensive and lack of time. Similarly, the main problems to employ a distributed inspection process were (in increasing priority order): technique not properly known, short time to market, technique too expensive and trust in the technique. Anyway, many of the companies (12 out of 24) were interested in this method.

Pair inspection was never used in eight companies because they did not know it before the survey. The other companies used this approach but did not consider it effective. However, the respondents of these companies believed that pair inspection was easy and cheap to use.

We also asked the respondents to indicate which practices (not covered in the survey) are currently used to identify defects within software artefacts. None answered this question, thus enforcing the assumption that methods/approaches investigated in our survey extensively cover the state of the practice of the companies involved in the study.

The last question of Section 3 was intended to obtain information regarding the interest in the methods/approaches considered in the survey. The same level of interest was generally manifested for all these methods/approaches. This result indicates that the interest in experimenting new methods increases when companies are aware of the benefits deriving from their use.

Table 1 Respondents' judgement on the considered approach/method

	Testing	Inspection	Distributed inspection	Pair inspection	Walkthrough
simple to use	6	0	0	12	6
effective	16	6	2	0	0
less expensive	4	4	0	12	4
best cost benefit ratio	14	4	0	2	4

5 Threats to validity

The threats to validity (i.e. internal, external and construct) that could affect the results of the survey are presented and discussed in the following subsections.

5.1 Internal validity

Internal validity threats regard external factors that may affect the observed results. In industrial surveys, it is usually impossible to know whether the respondents answer truthfully, or whether other effects bias the results. Scarce motivation to answer the questionnaire could negatively affect the results. To mitigate this threat, we properly designed the survey. For example, we chose an attractive, non-intimidating and professional layout for the questionnaire.

Another factor that may have influenced the internal validity is the number of invited companies that did not answer the questionnaire. For example, these companies could be not interested in software quality. This aspect could bias the achieved results. Even, the fact of having recruited the interviewed within our industrial contact network may influence the internal validity.

Another negative factor could be the difficulty to comprehend the questions (e.g. ambiguous, not clear, not well formulated). Accordingly, the questionnaire was designed to (i) minimise comprehension problems, (ii) reduce complexity and memory overload and (iii) increase respondent attention (e.g. reducing the number of questions to the essential). The use of unfamiliar terms in the questionnaire may also affect the obtained results. To reduce this threat, the questionnaire was provided with an introductory section in which we have defined any term/concept concerning the survey (see the appendix). This should avoid that differences in the respondents might bias the results, for example, different understanding of some aspects of software quality or the terms/concepts used on the questionnaire.

5.2 External validity

External validity threats concern the generalisation of the results. These threats are always present in case of industrial surveys. In our case, we used convenience sampling. Concerning the roles of the participants, our sample represents Italian IT companies/organisations that develop and sell software as a main part of their business or develop software as an integral part of their products or services. Therefore we cannot be sure that our sample is representative of the Italian IT industry in general and we are aware that Southern Italy is over-represented compared with Northern Italy. Replications are needed to increase our awareness on the survey results.

5.3 Construct validity

Construct validity threats concerns the metrics used in the study. In our case, the questionnaire was designed using standard ways and scales [29]. Furthermore, the questions were formulated to minimise possible ambiguities.

6 Discussion and future work

In this paper, we have presented the results of a survey to study and understand the state of the practice of software error/defect identification in the IT Italian industry. We invited 70 companies to participate and received 30 fully and correctly completed questionnaires, so obtaining 43% response rate that represents a good result for industrial survey especially in Italy (e.g. [23]).

The target population consisted of companies/organisations that develop and sell software. In particular, we considered IT Italian companies that develop and sell software as a main part of their business (e.g. software house) or develop software as an integral part of their products or services (e.g. commerce in the healthcare domain). The companies were distributed on the Italian territory as follows: 47% South, 40% Centre and 13% North.

With respect to the defined research questions, the findings of the survey can be summarised as follows:

RQ1: 'What is the relevance of error/defect identification in the Italian industry?' The data analysis showed that software error/defect identification is very relevant in the IT Italian companies of our industrial network. Most of the companies regularly use testing and have used often or sometimes diferent forms of inspection. This result suggests that software quality management is relevant in the design and development of software systems [30].

RQ2: 'What are the most popular and widely used approaches/methods (or simply practices) in the Italian industry?' Testing [1] is the most employed practice. This is because the companies considered testing effective and with the best cost benefit ratio. Future investigations are needed to better ascertain the testing methods (e.g. white- or black-box) employed in the industrial practice and in which phase of the testing process these are used (e.g. unit, integration and system levels). This further investigation makes sense also because of the results of the survey presented here. The study also revealed that most of the companies that regularly or often use testing (22 out of 30) adopt also other error/defect identification methods and they are mostly not interested in adopting new methods. This result suggests that these company believe that the used error/defect identification methods are more than sufficient to guarantee high-quality software products. In addition to testing, 12 companies regularly or often used software inspection. It is possible that variations of the Fagan's process are more used than other forms of software inspection (e.g. pair and distributed) because it is more formal and more consolidated than its variants. Note that almost all the companies that used pair inspection also used software inspection. Concluding, testing and software inspection are generally used in combination in the IT Italian industry. As inspection can be applied on any software artefact (contrasting to testing), it is likely that companies use inspection also on artefacts produced in the early phases of the development process. This conjecture needs to be evaluated. Future work will be devoted to understand why these practices are so popular and how they are used in combination.

RQ3: 'What are the main problems related to the use of approaches/methods for software error/defect identification?' The main concerns to introduce software inspection, pair inspection and walkthrough are: the short time to market and the availability of specialised employees. Inspection is used and is considered effective. The companies believed that pair inspection is not effective, but easy and cheap to use. Walkthrough is not adopted because it is considered non effective. Distributed inspection is not used because companies do not properly

doi: 10.1049/iet-sen.2011.0170

known it. The companies never used this approach for the following three main reasons:

(i) Employees are not geographically distributed. In case a company has more distributed business units, employees communicate using standard synchronous (e.g. instant messaging) and asynchronous communication media (e.g. email and/or forum).
(ii) Although the number of distributed inspection processes and tools (see [2]), the industrial practice is still far to adopt them. This indicates a gap between research laboratories and industrial reality that deserves a concrete cooperation between academy and industry based on technology transfer projects.
(iii) The available distributed inspection tools are not integrated with the SCM systems used in the companies. This lack was observed in [16]. We think that the integration of these tools within widely known systems for the management and version control would significantly increase their diffusion, thus improving the identification of error/defects within software artefacts developed in distributed contexts. It is also possible that the scant diffusion of these tools is because many of the companies in Italy are at best compliant with the level two of CMM and only few of them have an ISO 9000 certification. The lack of a market competition based on software quality could be the cause. These conjectures need to be investigated through special conceived industrial survey.

Despite the scant usage of distributed inspection, this method aroused great curiosity and interest. This is probably due to the fact that the respondents perceived it as less expensive than software inspection based on the Fagan's process. Furthermore, distributed inspection avoids problems related to the different time zones, when synchronous communication is not needed [2].

RQ4: 'Is the Italian software industry interested in using and adopting new practices for software error/defect identification?' Some companies were interested in experimentally using distributed inspection. It is likely that these companies were interested in distributed inspection because it is based on the process proposed by Fagan. In fact, the main difference between these two approaches is in the communication among team members. In distributed inspection, face-to-face interaction is replaced with synchronous and asynchronous communication media (e.g. [3, 16]).

Finally, the results of the unstructured interview also revealed that the interviewed companies used open source or commercial SCM systems. In particular, the largely adopted open source SCM systems are: subversion [31] and concurrent versions system (CVS) [32]. The widely used systems, among the commercial ones, are: team system and team foundation server. Owing to the interest in distributed inspection, integration of distributed inspection tools with these SCM systems could facilitate the introduction of distributed inspection in industry.

7 References

1 Pezzé, M., Young, M.: 'Software testing and analysis: process, principles, and techniques' (John Wiley & Sons, 2008)
2 De Lucia, A., Fasano, F., Scanniello, G., Tortora, G.: 'Evaluating distributed inspection through controlled experiments', *IET Softw.*, 2009, **3**, (5), pp. 381–394
3 Lanubile, F., Mallardo, T., Calefato, F.: 'Tool support for geographically dispersed inspection teams', *Softw. Process. Improv. Pract.*, 2003, **8**, (4), pp. 217–231
4 Chandola, V., Banerjee, A., Kumar, V.: 'Anomaly detection: a survey', *ACM Comput. Surv.*, 2009, **41**, pp. 15:1–15:58
5 Czaja, R.F., Blair, J.E. (Eds.): 'Designing surveys: a guide to decisions and procedures' (Pine Forge Press, 2004, 2nd edn.)
6 Kim, L.P.W., Sauer, C., Jeffery, D.R.: 'A framework for software development technical reviews'. Software Quality and Productivity: Theory, Practice and Training, London, UK, 1995, pp. 294–299
7 Laitenberger, O.: 'A Survey of Software Inspection Technologies' (World Scientific Publishing, 2002), vol. 2, pp. 517–555
8 Ciolkowski, M., Laitenberger, O., Biffl, S.: 'Software reviews: the state of the practice', *IEEE Softw.*, 2003, **20**, pp. 46–51
9 Jedlitschka, A., Ciolkowski, M., Denger, C., Freimut, B., Schlichting, A.: 'Relevant information sources for successful technology transfer: a survey using inspections as an example'. Proc. First Int. Symp. Empirical Software Engineering and Measurement, ESEM'07, Washington, DC, USA, 2007, pp. 31–40
10 Babbie, E.R.: 'Survey research methods' (Wadsworth Publishing, 1990, 2nd edn.)
11 Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: 'Experimentation in software engineering – an introduction' (Kluwer, 2000)
12 Runeson, P., Andersson, C., Thelin, T., Andrews, A., Berling, T.: 'What do we know about defect detection methods?', *IEEE Softw.*, 2006, **23**, (3), pp. 82–90
13 Dyba, T., Kitchenham, B.A., Jorgensen, M.: 'Evidence-based software engineering for practitioners', *IEEE Softw.*, 2005, **22**, (1), pp. 58–65
14 De Lucia, A., Fasano, F., Scanniello, G., Tortora, G.: 'Comparing inspection methods using controlled experiments'. Proc. 12th Int. Conf. Evaluation and Assessment in Software Engineering, Bari, Italy, 2008, pp. 1–10
15 Fagan, M.E.: 'Design and code inspections to reduce errors in program development', *IBM Syst. J.*, 1976, **15**, (3), pp. 182–211
16 Lucia, A.D., Fasano, F., Scanniello, G., Tortora, G.: 'Improving artefact quality management in advanced artefact management system with distributed inspection', *IET Softw.*, 2011, **5**, (6), pp. 510–527
17 Iisakka, J., Tervonen, I., Harjumaa, L.: 'Experiences of painless improvements in software inspection', in R. Kusters, A. Cowderoy, F. Heemstra, *et al.* (Ed.), 'Project control for software quality' (Shakes Publishing, 1999), pp. 321–327
18 Gilb, T., Graham, D., Finzi, S.: 'Software inspection' (Addison-Wesley, 1993)
19 IEEE: 'IEEE standard for software reviews'. Technical report, 1998
20 Hauge, O., Osterlie, T., Sorensen, C.-F., Gerea, M.: 'An empirical study on selection of open source software – preliminary results'. Proc. Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development, 2009, pp. 42–47
21 Li, J., Conradi, R., Slyngstad, O.P., Torchiano, M., Morisio, M., Bunse, C.: 'A state-of-the-practice survey of risk management in development with off-the-shelf software components', *IEEE Trans. Softw. Eng.*, 2008, **34**, pp. 271–286
22 Torchiano, M., Di Penta, M., Ricca, F., De Lucia, A., Lanubile, F.: 'Migration of information systems in the Italian industry: a state of the practice survey', *Inf. Softw. Technol.*, 2011, **53**, (1), pp. 71–86
23 De Lucia, A., Di Penta, M., Lanubile, F., Torchiano, M.: 'Metamorphos: methods and tools for migrating software systems towards web and service oriented architectures: experimental evaluation, usability, and technology transfer'. CSMR, 2009, pp. 301–304
24 Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., *et al.*: 'Preliminary guidelines for empirical research in software engineering', *IEEE Trans. Softw. Eng.*, 2002, **28**, pp. 721–734
25 Basili, V.R.: 'The role of experimentation in software engineering: past, current, and future'. Proc. 18th Int. Conf. Software Engineering, ICSE'96, Washington, DC, USA, 1996, pp. 442–449
26 De Lucia, A., Francese, R., Scanniello, G., Tortora, G.: 'Developing legacy system migration methods and tools for technology transfer', *Softw. Pract. Exper.*, 2008, **38**, (13), pp. 1333–1364
27 McCallon, E., McCray, E.: 'Designing and using questionnaires. School accountability series: monograph. Learning Concepts, 1975
28 Estublier, J., Leblang, D., Clemm, G., *et al.*: 'Impact of the research community on the field of software configuration management', *ACM Trans. Softw. Eng. Methodol.*, 2005, **14**, (4), pp. 1–48
29 Oppenheim, A.N.: 'Questionnaire design, interviewing and attitude measurement' (Printer Pub, 1992)
30 Basili, V.R., Musa, J.D.: 'The future engineering of software: a management perspective', *IEEE Comput.*, 1991, **24**, (9), pp. 90–96
31 Collins-Sussman, B., Fitzpatrick, B., Pilato, C.: 'Version Control with Subversion. O'Reilly, 2004. draft from http://www.svnbook.red-bean.com/
32 Berliner, B.: 'CVS II: parallelizing software development'. Proc. USENIX Winter 1990 Technical Conf., Berkeley, CA, 1990, pp. 341–352

8 Appendix

We clarify the meaning of some terms and describe all the entities of interest for the survey.

Project: a completed software development/maintenance project.

Software artefact: a tangible product created during software development. It can be created in the early and final stages of a software project.

Testing: a practice aimed at identifying defects within software [1].

Inspection: a formal method to identify defects within software artefacts [15, 18].

Distributed inspection: a method to support geographically distributed teams in the inspection of software artefacts [7]. Several approaches and tools are available in the literature that are more or less variants of a traditional inspection process (e.g. [3, 16]).

Pair inspection: an informal method that requires only two participants: the author and an inspector who reviews the authors' artefact [17].

Walkthrough: a form of software peer review to identify defects within software artefact [19]. It is generally adopted on source code and consists of manually executing it.

IET Softw., 2013, Vol. 7, Iss. 2, pp. 76–84
doi: 10.1049/iet-sen.2011.0170