Characterizing the Implementation of Software Reuse Processes in Brazilian Organizations

Marcelo Schots, Cláudia Werner

Systems Engineering and Computer Science Program (PESC)
Federal University of Rio de Janeiro – COPPE/UFRJ
schots@cos.ufrj.br, werner@cos.ufrj.br

Technical Report ES-749/14

August 2014

Abstract

Software reuse has come a long way in terms of academic research and industrial practice, and is present in day-to-day software development activities. However, although it is known that many software organizations still do not perform reuse in a controlled, pre-planned, strategic, or systematic way, the problems faced by many of them have not yet been properly characterized, particularly in Brazil.

In this sense, this work contributes with a characterization of the Brazilian scenario, by presenting a study conducted with implementers and assessors of MR-MPS-SW, a Brazilian maturity model that comprises software reuse processes, regarding the outcomes related to reuse activities. The main goal of this study is to get an overview of how reuse practices have been implemented and performed in software development organizations in Brazil so far.

The research findings show that most problems are recurring, especially in terms of the lack of adequate tool support and the need for more engagement in reuse initiatives. The results also indicate the need for actions from both academia and industry in terms of an effective understanding of software reuse practices and their benefits, as well as bridging the (still large) gap between research and practice.

Summary

Figures

Tables

1. Introduction

Software reuse has become a very common and widespread concept nowadays. According to [Holmes 2008], it has a well-established history in both research literature [Naur & Randell 1968] and industrial practice [Poulin et al. 1993]. It can be stated that reuse is present in the routine of software developers, yet mostly in an ad-hoc or pragmatic way.

A number of benefits can be achieved through reuse practices, such as reducing the effort and time spent on software development [Naur & Randell 1968] [Krueger 1992] [Mili et al. 1995] [Poulin et al. 1993]. Reusing assets from past projects (i.e., that have been already tested and deployed) also allows developing more reliable applications and decreasing maintenance efforts, since their quality is assured on previous experiences of use [Benedicenti et al. 1996] [Morisio et al. 2002]. Besides, the availability of reusable assets can facilitate newcomers in dealing with new technologies and domains (taking external solutions as a basis for their own development), as well as experienced developers in increasing their productivity by composing existing solutions.

After several years of software reuse research, it is important to constantly revisit and analyze how reuse practices are being implemented and performed in software development organizations in Brazil, aiming to recognize how reuse is taken into account in their day-to-day activities and propose new ways to approach new or remaining problems, if necessary.

In this sense, this report presents a study conducted for characterizing the state-of-the-practice of software reuse based on the point of view of implementers and assessors of MR-MPS-SW[1] (Brazilian Reference Model for Software Process Improvement) [Rocha et al. 2007] [SOFTEX 2012], a Brazilian maturity model that encompasses software reuse processes (among others), namely Reuse Management and Development for Reuse. The study aims at characterizing and identifying usual practices, problems, challenges, and opportunities for improvement in the implementation of software reuse processes.

The findings indicate that most problems (if not all of them) are recurring, i.e., have been identified previously. In other words, despite the extensive literature on software reuse, including several reports on how to achieve the expected benefits (and what should not be done, in order to avoid recurring failures), organizations still struggle in beginning and coping with reuse practices, as well as in selecting solutions that are suitable for the execution of organizational processes that involve activities related to reuse.

The remainder of this text is organized as follows: Section 2 provides some background on the implementation of reuse activities in software development, software reuse processes, some common issues identified in this context, and some reports on reuse implementations in the Brazilian scenario. Sections 3 and 4 describe the planning and execution of the study conducted for characterizing software reuse implementations, under the point of view of implementers and assessors of software reuse processes. The results of the study and a brief

[1] MR-MPS-SW stands for "MPS Reference Model for Software" (Modelo de Referência MPS para Software, in Portuguese).

discussion of the findings are presented in Sections 0 and 6, respectively. Finally, Section 7 shows the final remarks.

2. Background

This section presents some background on implementing software reuse processes and the issues reported during such implementations.

2.1. Reuse activities in software development

Since the idea of building new software from existing pieces of preexistent software arose [Naur & Randell 1968], it was noticed that several types of artifacts can be reused in software development, such as requirements specifications, software designs, test cases and so on [IEEE 2010]. However, some studies (including the one presented in this paper) and books point out that reuse of source code artifacts is still on the mainstream of software development, such as [Sá et al. 1997], [Haefliger et al. 2008], and [Leach 2012].

Reuse can occur within several activities in software development. A reuse-based development model (as illustrated in Figure 2.1) divides activities into two groups [Kim & Stohr 1998]: (i) *producing activities*, involving the identification, classification and cataloging of software resources, and (ii) *consuming activities*, comprising the retrieval, understanding, modification, and integration of those resources into the software product. These groups can also be referred to as *development for reuse* (i.e., build generic assets that can be reused in similar contexts) and *development with reuse* or *by reuse* (i.e., use existing assets to build [parts of the] software), respectively [Moore & Bailin 1991].

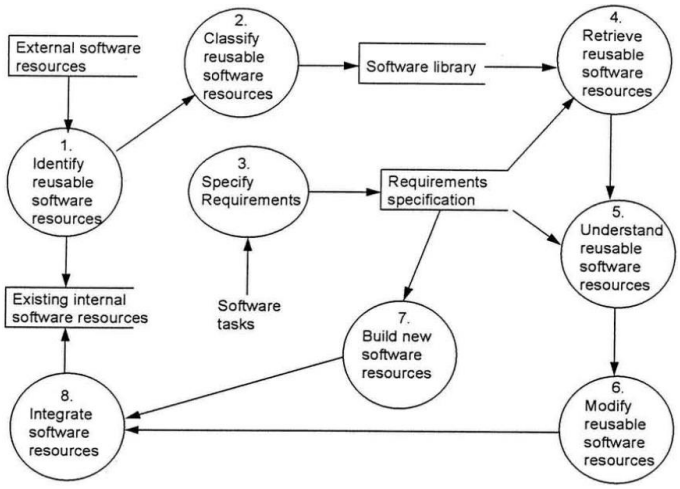

Figure 2.1 – Reuse-based development [Kim & Stohr 1998]

According to this figure, the first step (step 1) involves analyzing existing software resources (that are developed internally or externally) in order to identify potentially reusable artifacts (that may require some adjustments to this end), which must be then classified and cataloged (step 2) in a software library. These two steps have to be performed at the beginning of a reuse program and whenever a new software resource is acquired/developed [Kim & Stohr 1998]. Specifying requirements for the new system (step 3) has to be performed regardless of whether the software resource is to be developed from scratch or not.

Retrieving appropriate reusable software resources from the software library (step 4) is only necessary in a software reuse scenario [Kim & Stohr 1998]. After that, the next step (step 5) is to understand and assess the functionality of the selected resources in order to use or modify them. Modifying software resources (step 6) is necessary for adapting reusable assets to the context of a given application when the retrieved resources do not exactly match the requirements specification – which often occurs –, while building new software resources (step 7) is required when there are no similar resources in the software library for meeting some of the requirements. Finally, the last step (step 8) is the integration of both new and reusable software resources into the target software system [Kim & Stohr 1998].

The advantage obtained from a reuse-based software development scenario is to develop software assets aiming at their future reuse (if appropriate, according to the organization's goals). Developing assets without taking into account their reuse potential (and consequently without aiming their reuse beforehand) makes it hard to fit them into other contexts beyond the original ones to which they were developed. This is partially due to the lack of systematization in the construction of reusable assets [Prieto-Diaz & Arango 1991].

Before the introduction of a reuse program, some steps must be accomplished. According to [Benedicenti et al. 1996], it is necessary to perform an accurate assessment of the organization's current situation, including goals, mission and market strategies defined by top management. The authors also state that the integration of a reuse program to the current development process can only be effective "if the process itself is well defined and structured, and the software life cycle is planned and managed" by the organization. Finally, one must identify and describe the specific domains in which there are high reuse potential (e.g., through domain analysis techniques [Moore & Bailin 1991]), since reuse "can significantly increase the efficiency of development and maintenance processes for applications characterized by commonalties of aspects and targets" [Benedicenti et al. 1996].

Assuming that these steps have been properly accomplished, a general process for introducing a reuse paradigm can be modeled as follows [Benedicenti et al. 1996]:

i. identify and formalize the primary goals that the organization wants to achieve with the introduction of reuse;

ii. define the aspects and fields of the organization which will be targeted by the reuse program, choosing the application domains with greatest reuse opportunities;

iii. identify and collect commonalties inside the chosen domains;

iv. set up a group of people with the task of supporting each reuse activity, developing reuse guidelines and reuse documentation for the organization;

v. define a process to assess the evolution of the reuse program, along with a continuous process aimed to improve and specialize it; and

vi. introduce a strong education and training program to stimulate and support the transition to a reuse-based software life-cycle, fostering the integration of the reuse program in the software development process.

2.2. Implementing software reuse processes

Software development organizations need to continually seek for improvement of the quality of their products and services, in order to endure in the competitive market. Consequently, they are also aiming at improving the quality of their processes. Due to this increasing demand for software quality, a number of quality standards and maturity models – such as CMMI-DEV (Capability Maturity Model Integration for Development) [CMMI Product Team 2010] and MR-MPS-SW (Brazilian Reference Model for Software Process Improvement) [SOFTEX 2012] – have been proposed, establishing requirements for defining, evaluating, and improving software processes.

In order to promote ways towards its systematization, software reuse is covered by several quality standards (e.g., [ISO/IEC 2008], [IEEE 2010], and [ISO/IEC 2012]), which comprise activities related to the management of the reuse program, as well as the storage, retrieval, management and control of the assets, among others. Such standards also contain guidelines for integrating reuse in the primary processes of the software life cycle, along with processes for reuse across projects.

According to [ISO/IEC 2008], for instance, a successful implementation of the Reuse Program Management process should provide the following outcomes as results [ISO/IEC 2008]:

- define the organization's reuse strategy, including its purpose, scope, goals and objectives;
- identify the domains in which to investigate reuse opportunities or in which it intends to practice reuse;
- assess the organization's systematic reuse capability;
- assess each domain to determine its reuse potential;
- evaluate reuse proposals to ensure the reuse product is suitable for the proposed application;
- implement the reuse strategy in the organization;
- establish feedback, communication, and notification mechanisms that operate among reuse program administrators, asset managers, domain engineers, developers, operators, and maintainers; and
- monitor and evaluate the reuse program.

Reuse practices are also integrated into models that aim to measure the maturity level of organizations that produce software, such as MR-MPS-SW [Rocha et al. 2007] [SOFTEX 2012], a program for software process improvement coordinated by the Association for Promoting the Brazilian Software Excellence (SOFTEX). This program aims to define and enhance a model for improvement and assessment of software processes focusing on micro, small, and medium enterprises (MSMEs). The MPS-SW model complies with ISO/IEC 12207 and 15504, is compatible with CMMI-DEV, adopts software engineering best practices, and is appropriate (both from the technical point of view as to costs) to the reality of Brazilian organizations [SOFTEX 2012].

MR-MPS-SW is divided into 7 maturity levels, from level G (lowest maturity level) to level A (highest maturity level), in ascending order. Since its version 1.2 (released in 2007), this

model indicates reuse as one of the goals to be accomplished by organizations in order to evolve their maturity levels. In this model, reuse is expressed in two processes: Reuse Management – GRU[2], required since the intermediary maturity stages (starting from level E), and Development for Reuse – DRU[3], in more advanced stages (from level C onwards).

The purpose of the Reuse Management process is to manage the life cycle of reusable assets [SOFTEX 2012]. To make this possible, the process defines that the organizations must have a documented strategy for asset management, including criteria that govern their life cycle (i.e., criteria for acceptance, certification, classification, discontinuity and evaluation of assets) (GRU 1). In addition, there must be a mechanism for the storage and retrieval of assets (GRU 2). Modifications on these assets must be controlled throughout the life cycle (GRU 4), and usage data shall be recorded (GRU 3), so as to notify users about potential problems detected, modifications carried out, new versions available and discontinued assets (GRU 5) [SOFTEX 2012].

The purpose of the Development for Reuse process, in turn, is to identify opportunities for systematic reuse of assets in the organization and, if possible, establish a reuse program for developing assets from the engineering of application domains [SOFTEX 2012]. This process starts with the identification of the reuse potential (DRU 1) and the reuse capabilities (DRU 2) of the organization. The ensuing steps are the planning (DRU 3), implementation, monitoring and evaluation (DRU 4) of a reuse program, which comprise the evaluation of proposals for reuse (DRU 5), the development of domain models and domain architectures (DRU 6, DRU 7 and DRU 8), and the specification, development (or acquisition) and maintenance of domain assets (DRU 9) [SOFTEX 2012].

2.3. Issues on software reuse implementations

Achieving effective software reuse is a difficult problem in itself, one that requires proper support in a number of facets, such as managerial aspects [Griss et al. 1994], the aid of tools [Marshall et al. 2003], and adequate mechanisms for retrieval of reusable assets [Braga et al. 2006], among others. In order to be acquainted with the barriers related to effective reuse, it is important to recognize some usual concerns and issues associated to software reuse initiatives.

A number of studies and reports on the implementation of reuse processes in organizations are presented in the literature (e.g., [Kim & Stohr 1998], [Morisio et al. 2002] and [Sherif & Vinze 2003], among others). Some of the frequent issues and challenges pointed out regarding the establishment of a reuse program include the following:

- the difficulty in understanding software reuse concepts and how to effectively apply them [Mili et al. 1995] [Morisio et al. 2002];
- the lack of acceptance of reuse practices by the development team and top management in software organizations [Mili et al. 1995] [Benedicenti et al. 1996] [Sherif & Vinze 2003];

[2] Acronym for “Gerência de Reutilização”, in Portuguese.
[3] Acronym for “Desenvolvimento para Reutilização”, in Portuguese.

- the lack of knowledge and experience for the creation and management of reuse repositories [Morisio et al. 2002] and the definition, identification and evaluation of reusable assets, as well as making such assets available and findable [Frakes & Kang 2005];
- a long learning curve of understanding a software asset, i.e., its structure, behavior and functionality [Ye & Fischer 2002] [Marshall et al. 2003] [Frakes & Kang 2005];
- the lack of proper tool support for performing software reuse tasks [Mili et al. 1995] [Benedicenti et al. 1996] [Morisio et al. 2002];
- the absence of a culture of development for reuse in development teams and the lack of systematization for the construction of reusable assets [Sherif & Vinze 2003]; and
- the "Not-Invented-Here" (NIH) syndrome [Sherif & Vinze 2003], i.e., the difficulty of accepting and trusting third-party developed assets, resulting in a tendency towards "reinventing the wheel" (recreating something from scratch instead of reusing) based on the belief that in-house developments are inherently better than existing implementations.

Many reuse-related issues can be associated to technical aspects, such as the lack of tools and techniques for effectively supporting software reuse, as pointed out by [Kim & Stohr 1998], [Lucrédio et al. 2008] and other works. Particularly, wrong technology choices may considerably hamper the execution of reuse processes [Lucrédio et al. 2008]. However, it is important to emphasize that solving these aspects is not enough for the success of a reuse program. According to [Card & Comer 1994] and [Morisio et al. 2002], a misconception of the reuse needs may lead to the probability of neglecting the importance of assessing the reuse potential at the organizational level and addressing other barriers, treating reuse as a matter of technology acquisition. Thus, as with any other software process, a crucial concern that must be taken into account is the envisioning of non-technical aspects [Kim & Stohr 1998]. Sherif and Vinze (2003) highlight that reuse provides better results when all stakeholders are committed to it.

Because software development processes are performed by people, attempts to introduce a software reuse program may also fail because of human issues, such as: (i) lack of management commitment, (ii) lack of understanding of reusable assets, (iii) lack of engagement of team members, (iv) absence of incentives, and (v) cognitive overload [Kim & Stohr 1998]. According to [Schmidt 1999], some of the non-technical impediments to successful reuse commonly include the following:

- *Organizational impediments*: Systematically developing, deploying, and supporting reusable software assets require a deep understanding of application developers' needs and business requirements. As the number of developers and projects employing reusable assets increases, it becomes hard to structure an organization to provide effective feedback loops between these constituencies.
- *Administrative impediments*: It is hard to catalog, archive, and retrieve reusable assets across multiple business units within large organizations. Although it is common to opportunistically scavenge small classes or functions from existing programs, developers often find it hard to locate suitable reusable assets outside of their immediate workgroups.
- *Psychological impediments*: Application developers may also perceive "top down" reuse efforts as an indication that management lacks confidence in their technical abilities. In

addition, the “Not Invented Here” (NIH) syndrome [Sametinger 1997] [Sherif & Vinze 2003] is ubiquitous in many organizations, particularly among highly talented programmers.

Regarding the latter impediment, another study [Frakes & Fox 1995] pointed out that the NIH syndrome has become a minor obstacle, and has included reuse education and the perceived economic feasibility (among others) as factors that affect reuse, in accordance with [Card & Comer 1994]. However, although this syndrome has been alleviated over time, much of the phenomenon is caused by the cognitive difficulties that are inherent in the reuse process [Ye & Fischer 2000].

2.4. The software development scenario in Brazil

According to [ABES 2014], in 2013, the software industry in Brazil had an increase of 13.5% on the investments compared to 2012. Overall, software and services grew by 10.1%, above the great majority of other sectors of the Brazilian economy. The use of computer programs developed in Brazil (standard and custom) increased 15.3%, higher than the 12.9% growth identified in the use of computer programs developed abroad, reinforcing the trend of growth that comes been appointed since 2004 [ABES 2014].

The domestic market is operated by approximately 11,230 companies, dedicated to the development, production, and distribution of software and services. Finance, Services and Telecom accounted for almost 51% of the user market, followed by Industry, Government, and Commerce. Considering only the size of companies engaged in the development and production of software (around 2,700 at the date of the report), these can be divided as follows: micro (43.9%), small (49.6%), medium (5.2%) and large (1.3%) [ABES 2014].

The Brazilian scenario is very competitive (considering both nationwide and worldwide settings), and software reuse processes play an important role in this regard, due to its well-known benefits. Thus, it becomes important to characterize this scenario and obtain more information on the implementation of reuse processes in Brazilian software organizations, i.e., how they consider reuse practices.

2.5. Related work

After performing the literature review on common issues in implementing software reuse processes, a search was performed for identifying reports concerning Brazilian organizations. The identified reports are listed as follows.

Sá et al. (1997) report the experience of introducing software reuse in an organization, by measuring aspects related to reuse before and after the implementation [Sá et al. 1997]. The authors mention technical and cultural obstacles identified during the process, such as: (i) reuse was only understood as code reuse; (ii) there was no technical or managerial commitment to produce reusable assets; (iii) the development of most systems was going straight to the implementation phase, because stakeholders did not believe in Software Engineering as presented in the literature; and (iv) the view of profits was immediate (short-sighted) regarding the production of reusable assets [Sá et al. 1997].

Lucrédio et al. (2008) present a survey carried out with industry professionals, involving Brazilian organizations, aiming to relate organizational characteristics with the successful adoption of reuse, not taking into account the reasons why some organizations were not

successful. The survey comprised several factors divided into four perspectives: organizational factors, business factors, technological factors, and processes factors. From the 200 contacted organizations, 57 answered the survey. The main influence factors identified include the development team, the use of tools and quality models, the prior development of reusable assets, the type of these assets, and the existence of a systematic reuse process. The difficulties encountered are also related to these factors (e.g., an inadequate tool support and the lack of systematization of reuse represent negative influence factors) [Lucrédio et al. 2008].

Silva Filho et al. (2008) describe the implementation of the MR-MPS-SW Reuse Management (GRU) process at the Software Engineering Laboratory of an academic institution. Any software artifact (process asset, source code, or executable) could be considered as reusable assets; they were suggested by the team and evaluated against their quality and reuse potential. Notifications related to the assets' status were made manually by e-mail. The main difficulties mentioned were the definition of a non-intrusive strategy (i.e., which would not impact the usual activities of the organizational unit) and the choice of useful metrics to monitor and control the process. As to technical aspects, the identification of reusable assets was considered the most critical activity regarding the level of intrusion, cost, and effort. Some lessons learned include: (i) the definition of a reuse management focus (such as the minimization of projects' cost and effort) can guide to the prioritization of software process improvements, and (ii) the more mature the reuse management process is, the clearer the perception on how it can be automated [Silva Filho et al. 2008].

Santos et al. (2009) describe the experience on implementing MR-MPS-SW Reuse Management (GRU) and Development for Reuse (DRU) processes in a medium-sized, geographically distributed organization. The defined process for GRU is triggered either from the need to assess candidate assets or for implementing enhancements in a particular asset (based on problems or opportunities for improvement identified over time). Regardless of how the process starts, it ends with the notification of interested parties regarding the availability, evolution or discontinuation of reusable assets. A research is performed for identifying people potentially interested in a given reusable asset, as well as for defining the role responsible for maintaining such asset. An assessment of the reusable assets base is periodically made for identifying assets that can be improved or be subject to discontinuation (e.g., criticized by users or less used). In an initial search effort on the organization's legacy systems, 4 potential reusable assets were identified, being 3 approved on the acceptance and certification criteria, becoming part of the reusable assets library. The authors underline the low number of identified reusable assets. Besides, the tools used for supporting the reuse program were too general, such as text editors and spreadsheets. Communications related to the reuse processes are made by e-mails that are sent manually [Santos et al. 2009].

Regarding DRU, an initial list of nine areas of expertise was identified, corresponding to the business processes supported by the systems developed by the organization. From these, only three were rated as having some potential for systematic reuse and, therefore, were analyzed in more detail (according to the authors, the other processes did not follow a formal line of development compatible with reuse principles). The assessment of the reuse capabilities of the organization showed that there were limited resources for the establishment of an appropriate reuse program, but a plan was drawn up to overcome this limitation. Nevertheless, DRU was considered out of scope during the final assessment of the implementation, not being fully

implemented due to the lack of both data and results on the development of reusable assets [Santos et al. 2009].

Although the literature reports on the implementation of reuse processes in the Brazilian scenario present some problems in common, they usually describe isolated cases, and do not aim at comprehensively characterizing usual problems identified during the implementation and assessment of reuse processes. The most comprehensive one is the work of Lucrédio et al. (2008), but it is not based on a widely used quality standard or maturity model, i.e., it cannot be ensured that all the analyzed organizations perform a set of reuse tasks in common. This is one of the main motivations for conducting the study presented as follows.

3. Study Planning

3.1. Goal

The study goals are described in the Goal-Question-Metric (GQM) format [Basili et al. 1994] as follows:

Analyze software reuse implementations

For the purpose of characterizing

With respect to usual practices, problems, challenges, and opportunities for improvement

Under the point of view of MR-MPS-SW implementers and assessors

In the context of Brazilian software development organizations

3.2. Population

Since this study aims at characterizing software reuse in Brazilian organizations based on a set of outcomes in common, MPS.BR implementers and assessors[4] compose the population of this study. The choice for this population is due to the fact that there is a representative number of MPS.BR assessments on level E (60 out of the 488 organizations successfully assessed in MPS.BR[5] are in level E or above, including 38 in level C or above[6]), covering a considerable portion of the nationwide scenario[7].

[4] MPS implementers are affiliated to Implementing Institutions (II) accredited to render consulting services regarding the implementation of the MR-MPS-SW and MR-MPS-SV reference models, while MPS assessors are affiliated to Assessment Institutions (AI) accredited to render assessment services based on the MA-MPS Assessment Method. According to the MPS organizational structure, the MPS Accreditation Forum is responsible for accrediting such institutions [SOFTEX 2013].

[5] MPS-SW Published Assessments (data from August 23, 2013), extracted from http://www.softex.br/wp-content/uploads/2013/07/Avalia%C3%A7%C3%B5es-MPS-SW.pdf.

[6] It is noteworthy that the DRU process allows the exclusion of most outcomes from an assessment if the organization does not have opportunity and/or ability to perform development for reuse. Thus, one cannot state that all these organizations perform DRU.

[7] Please refer to http://www.softex.br/mpsbr/avaliacoes/mps-sw/mpsbr-ma-mps/ for an overview.

3.3. Methodology

In order to obtain more information on the implementation of processes related to reuse in software organizations in Brazil, semi-structured interviews were conducted with the participants of the study. Semi-structured interviews represent a viable alternative when conducted to obtain or confirm information about a predetermined topic. They "are designed to elicit not only the information foreseen, but also unexpected types of information" [Seaman 1999], which meets our expectations with this study.

For conducting the interviews, some advice from [Seaman 1999] and [Hove & Anda 2005] was used in order to ensure good interaction between the interviewer and interviewees. For analyzing the collected data, the open coding technique [Seaman 1999] is used, by marking and categorizing snippets of interviews, relating them to questions (categories) initially defined.

3.4. Study design decisions

During the planning of this study, some decisions that were taken may influence or lead to limitations on the findings. We list the main decisions and discuss our rationale for each of them.

- *Regarding the population of the study*: There were two main concerns: (i) the representatives of participants and (ii) their capability to provide answers to the stated questions. Regarding the former, we decided to send invitation e-mails based on the list of authorized Implementing Institutions (IIs) and Assessment Institutions (AIs) available on the SOFTEX website (as discussed in Section 0), so that every implementer and assessor who was willing to contribute with the study could be able to do it. Regarding the latter, we depended on the participants' profile and experience, which had to be analyzed afterwards (as discussed in Section 0).
- *Regarding the choice of MR-MPS-SW*: One of the advantages of the MR-MPS-SW is that it provides a set of outcomes that must be implemented by all the organizations attempting to achieve a given level of maturity. This allows ensuring that they all had to show evidence of accomplishment of the same set of goals in such level, providing a sort of uniformity among them in this regard. Besides, the outcomes of the GRU and DRU processes from MR-MPS-SW were used as input for designing the questions of the study.
- *Regarding the method used*: A downside of performing semi-structured interviews is the fact that interviews are a resource-demanding data collection method, because activities such as planning, conducting, and analyzing are time-consuming by nature [Hove & Anda 2005]. Due to the effort of performing several interviews and analyzing a huge amount of data from the answers, one could resort to surveys, which considerably increases scalability of participations. However, response rates in surveys are usually low [Kitchenham & Pfleeger 2008], and through this kind of study we would not be able to collect detailed qualitative information. On the other hand, semi-structured interviews allow more flexibility to deepen in a particular subject matter that deserves more attention. Besides, any response that is not well understood by the interviewer can be clarified immediately. Finally, due to the exploratory nature of this study, we decided that the cost-benefit would be worth the effort. This is why we opted out to keep the semi-structured interview approach, in spite of the underlying limitations.

3.5. Interview Questions

The interview questions were designed to obtain both technical (regarding the decisions for implementing the processes, based on the outcomes) and non-technical information (involving implementers' opinions concerning the assessed organizations, as well as difficulties and frequent problems) with respect to reuse processes. Some of the questions were directly derived from the MR-MPS-SW reuse-related outcomes. Table 3.1, Table 3.2 and Table 3.3 show the questions for Reuse Management (GRU), Development for Reuse (DRU) and other questions that are relevant for reuse processes in general, respectively, along with their corresponding goals.

Table 3.1 – Questions and Goals of the Study – Reuse Management (GRU)

ID	Question	Goal
Q1	Which kinds of assets have been considered as reusable by the organizations?	Identify which types of artifacts are considered as reusable by organizations in their projects/ processes. Related to GRU 1.
Q2	Where are the reusable assets usually stored?	Identify mechanisms (tools) used for storing reusable assets. Related to GRU 2.
Q3	Where/how are the reusable assets made available for reuse, i.e., where/how are the stored assets listed so that the interested parties can find them?	Identify the way organizations make their reusable assets available and the mechanism (tool) used to this end. Related to GRU 2.
Q4	How are the usage data about the assets logged?	Identify how organizations record reusable assets' usage data. Related to GRU 3.
Q5	How are interested parties informed of problems detected, modifications made, new versions released, and discontinued assets?	Identify the mechanisms used for notifying interested parties about changes in the status of assets. Related directly to GRU 5 and indirectly to GRU 4.

Table 3.2 – Questions and Goals of the Study – Development for Reuse (DRU)

ID	Question	Goal
Q6	What are the application domains of the organizations in which opportunities for reusing assets have been identified, or in which they have intended to practice reuse?	Identify relevant application domains from the viewpoint of the state-of-the-practice. Related to DRU1.
Q7	Are organizations able to plan and establish an effective reuse program?	Check if reuse programs have been properly established in organizations. Related to DRU3 and DRU4.
Q8	How are organizations monitoring the reuse program?	Identify monitoring mechanisms and strategies being used by organizations. Related to DRU4.
Q9	How are reuse proposals (requests for reusing existing domain assets or developing/acquiring new ones) made?	Identify how reuse proposals are made and which kinds of request are more frequent. Related to DRU5.

ID	Question	Goal
Q10	How are domain models and domain architectures represented in organizations?	Identify techniques being used by organizations for representing domain models and domain architectures. Related to DRU6, DRU7, and DRU8.
Q11	How are domain assets specified/acquired/developed and maintained?	Identify techniques being used by organizations for specifying, acquiring, and/or developing domain assets. Related to DRU9.

Table 3.3 – Questions and Goals of the Study – General Questions on Reuse Processes

ID	Question	Goal
Q12	Which comments are made by the organizations regarding the GRU and DRU processes?	Characterize general problems pointed out by organizations. The answers to this question may drive the remainder of the interview for more details (funnel strategy).
Q13	What is the point of view of the diverse stakeholders (developers, project managers, top management) about reuse?	Identify whether there is any cultural resistance by stakeholders and, if so, which roles have such resistance. This information is also relevant for DRU4.
Q14	Which GRU and DRU aspects are more difficult to understand by the organizations?	Obtain more information about difficulties in understanding (including processes, concepts, tasks, tools etc.) pointed out by the respondent. This question is purposely broad.
Q15	Which are the most difficult tasks (particularly, GRU and DRU tasks) for the organizations to perform?	Identify information about the most difficult tasks.
Q16	What are the problems ("required" items) usually identified on GRU and DRU during assessments?	Identify issues that organizations cannot accomplish in GRU and DRU, as well as potential difficulties in implementations.
Q17	Which aspects related to the implementations or assessments of the GRU and DRU processes would you like to add (including the moment in the MR-MPS-SW implementation when you start to implement GRU and DRU processes, and potential difficulties in implementing or evaluating these processes)?	Identify difficulties on the implementations or assessments of the GRU and DRU processes, and ultimately verify how organizations prepare themselves to assessments.
Q18	Is there anything else that has not been asked and you would like to comment on?	Obtain feedback on the process and other aspects that participants would like to add.

Other aspects related to the outcomes were not directly included in the questions, such as the control of changes in assets (related to GRU 4) and the criteria for acceptance, certification, classification, evaluation and discontinuity of assets (related to GRU 1), among others. These items are very specialized and were indirectly evaluated through the general questions and the intersection with other outcomes.

4. Execution

Invitation e-mails were sent based on the list of authorized Implementing Institutions (IIs) and Assessment Institutions (AIs) available on the SOFTEX website[8]. The response rate in terms of the IIs and AIs was 38.46%. The criterion for participation in the study was the experience in the implementation and/or assessment of the GRU and/or DRU processes. Participants were interviewed in person during the XII Brazilian Symposium on Software Quality (July 1 to 5, 2013), or remotely, via Skype (between July 6, 2013 and August 25, 2013).

In total, there were 10 respondents, all concomitantly MR-MPS-SW implementers and assessors, having carried out (or accompanied, as leader assessors) at least 1 implementation or assessment of the GRU process (in most cases, more than 3 assessments). Figure 4.1 shows the distribution of the respondents according to the MPS assessor levels (ordered from the lowest to the highest), while Figure 4.2 shows the year of authorization[9] to perform MR-MPS-SW implementations and assessments.

As it can be seen, most participants are competent/lead assessors, meaning that they received a specific training from an assessment institution and performed at least 6 assessments as provisional/assistant assessors [SOFTEX 2013]. Moreover, 2 of them are experienced competent/lead assessors, i.e., besides having competent/lead assessor's skills, they had a specific training on statistical process control and performed at least 4 assessments in levels E, D and C as competent/lead assessors [SOFTEX 2013]. Additionally, all the respondents were formed implementers before the release of version 1.2 of MR-MPS. Because a complementary training course is mandatory whenever substantial changes are made in the model, all of them were trained and are allowed to perform implementations of reuse processes from the moment such processes were incorporated into MR-MPS-SW.

Regarding the participants' background, MPS assessors must fulfill requirements such as academic qualification (post-graduation, master or doctorate degrees and knowledge on Software Engineering focused on software process) and confirmed professional experience (at least 3 years of experience on Software Engineering and 6 years of experience on software project management or experience with significant participation on deploying software process to an organization unit successfully assessed at a given maturity level) [Rocha et al. 2007] [SOFTEX 2013]. These requirements become stricter as the assessor levels increase.

[8] Available at: http://www.softex.br/mpsbr/instituicoes-autorizadas/.
[9] Based on http://www.softex.br/mpsbr/profissionais-habilitados-2/.

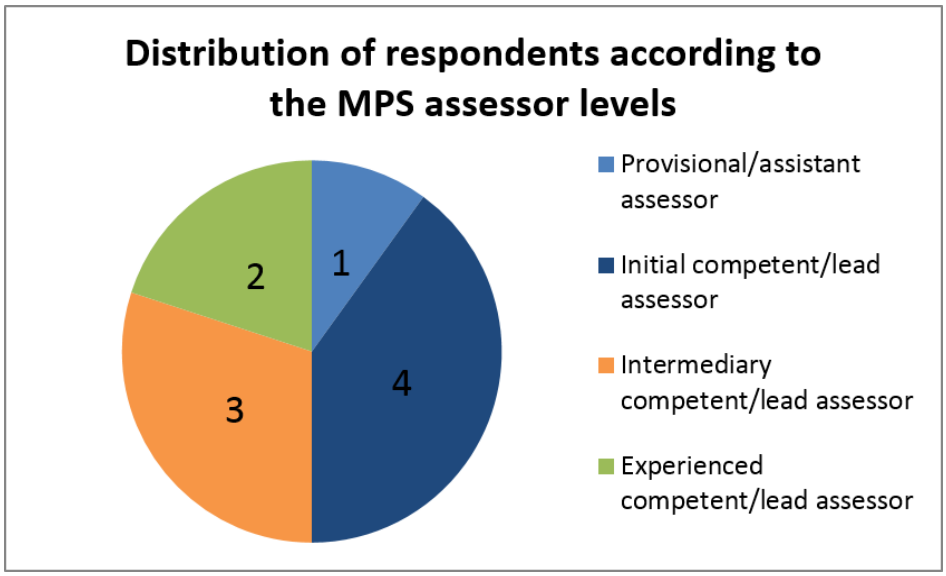

Figure 4.1 – Distribution of respondents, according to the MPS assessor levels

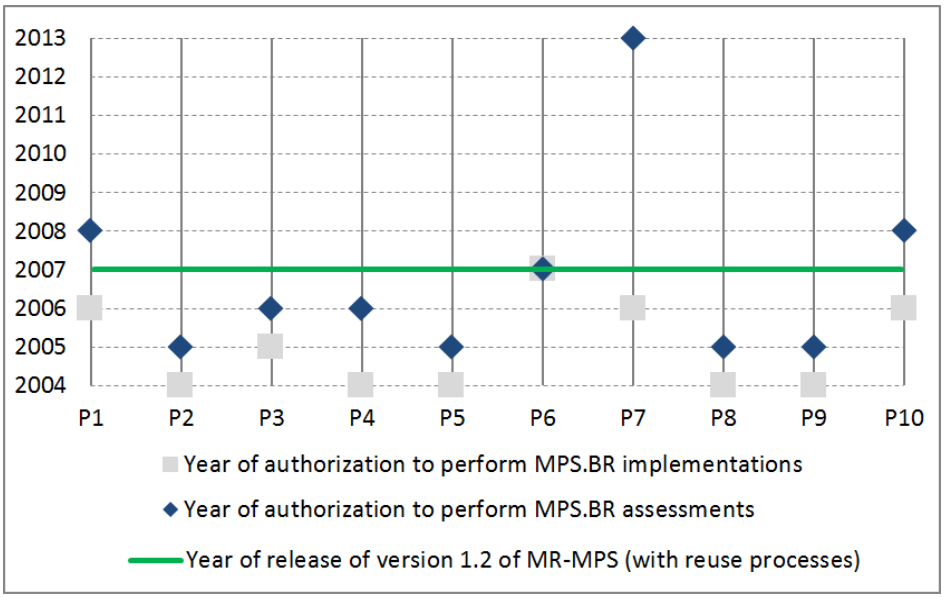

Figure 4.2 – Participants' experience (based on the year of authorization) in implementations and assessments

Since the participants were responsible for more than 60% of the implementations and assessments nationwide[10] (some of the respondents participate in the MPS.BR program since its beginning), they are considered a representative sample, given their vast experience. It must be

[10] Although the SOFTEX website does not provide the number of implementations/assessments per person as well as the list of states where each person has conducted an implementation/assessment, the list of published assessments [SOFTEX 2014] contains, for each assessed organization, the city/state of the organization and the persons responsible for performing the implementation and the assessment.

emphasized that the participants are not able to provide detailed data of the assessed organizations (due to disclosure agreements and given the number of assessments performed by them); however, [Travassos & Kalinowski 2012] and [Travassos & Kalinowski 2013] provide an overview of the Brazilian organizations that have implemented the MR-MPS-SW model.

5. Results

For a better reading of the results, the answers are categorized into topics, regardless of the order of the questions. The numbers in parentheses refer to the number of respondents who provided the corresponding statement[11]. It is worth emphasizing that the same respondent may have provided more than one answer per question. In order to ensure the confidentiality of responses, results that may allow the identification of the respondent are always reported together and in aggregated form. Furthermore, no gender distinction is made in the text.

5.1. Kinds of reusable assets and mechanisms for storage and retrieval

In response to Q1, all respondents cited source code as an asset considered by organizations to be reusable, either in the form of libraries, frameworks, individual classes (for object-oriented systems) or even code routines (for systems developed in other programming paradigms). Source code is also the asset most often identified as reusable in organizations, and was pointed out by some respondents as "most useful" and/or "most suitable". Nevertheless, respondents indicated that the number of reusable assets found in the reuse base of organizations is considerably small (2-4 assets were mentioned, in some cases).

Document templates related to processes – such as project plan templates or checklists – are also usually considered to be reusable (5). Other kinds of assets mentioned are standard processes[12] (2), test scenarios (2), test plans (1), knowledge assets (1), business rules (1), and calculation spreadsheets (1). With respect to plans and test scenarios, one respondent mentioned that these were reused without specific adjustments for each scenario, i.e., their content was entirely reused in different projects.

Some respondents stated that they do not consider appropriate to treat knowledge assets and document templates as reusable assets (GRU 1), for different reasons: either they consider the management of the GRU process as laborious/onerous (given the high number of assets and the high frequency of use), or they believe that, by using such kinds of assets, the benefits arising from reuse are not well noticeable.

Additionally, some respondents noticed that a number of implementations of GRU use standard processes and knowledge assets as reusable assets, but they disagree with this approach. According to them, these assets are already addressed in other processes of the same MR-MPS-SW maturity level – Organizational Process Definition (DFP[13]) and Human Resource

[11] The numbers are used only for accounting purposes. The contribution of studies like this is not the quantification, but the richness and variety of data, in accordance with [Seaman 1999].
[12] Considering each instantiation of a standard process as a form of reuse.
[13] Acronym for "Definição do Processo Organizacional", in Portuguese.

Management (GRH[14]), respectively –, thus not aggregating noticeable reuse results to the organizations. One can "end up improving knowledge management rather than making the implementation of a reuse process", for instance.

Regarding Q2, related to the storage of reusable assets (in GRU 2), all respondents mentioned that version control repositories are usually employed for this purpose, being SVN the most frequent implementation. Some respondents stated that the storage mechanism varies according to the type of asset. Version control repositories are most common when the reusable asset is source code (but are also used in some implementations of reuse to store other types of assets). Other forms of storage were also cited: folders / directories (usually shared on the Intranet), either with or without controlled access (6), used for storing process templates documents (2) and standard processes (1); default location (local folder, in the cloud etc.) of collaborative tools and tools with social network features: Sharepoint[15] (2), Google Docs/Drive[16] (1) – also used for knowledge assets (1) –, and Confluence[17] (1); and a tool developed by the organization (1).

The most cited form of making reusable assets available (Q3, also related to GRU 2) is by using tools for integration with repositories (7), such as TortoiseSVN[18] (4) and Maven[19] (1). Other cited forms of provision were: Wiki[20] (4); collaborative tools (aforementioned); a directory[21] informing the classification, type, and link of the reusable assets (1), and a list of assets available in the Intranet portal (2). One respondent affirmed that "it is difficult to make software development stakeholders obtain the reusable assets from a Wiki or similar mechanism, when there is a version control tool integrated to the development environment from which obtaining the assets is more practical" (in response to Q14).

A respondent stated that sometimes "the component works very well, but no one knows where it is, and the time it takes to find the component, study it and know how it works, depending on the size of the component, may end up giving rise for the developer to think that it is better to build his/her own component". In fact, the programmer can be reluctant in "exploring a huge reusable asset database for identifying which asset fits his/her needs", and "may therefore prefer to develop from scratch, which might be faster in his/her discretion".

In addition, programmers often "do not have the necessary knowledge to decide whether it is better or not to reuse a given component", according to one respondent. A huge reusable asset database to explore, the size of the asset, the time it takes to find and study it, and the lack of associated documentation are aggravating factors mentioned by respondents. When there are a

[14] Acronym for "Gerência de Recursos Humanos", in Portuguese.
[15] http://office.microsoft.com/pt-br/sharepoint/
[16] https://drive.google.com/
[17] https://www.atlassian.com/software/confluence/
[18] http://tortoisesvn.net/
[19] http://maven.apache.org/
[20] http://www.mediawiki.org/wiki/MediaWiki
[21] In this case, unlike the concept of file folder, directories are indexes of sites, usually organized into categories and subcategories, whose main purpose is to quickly find desired websites, searching by categories. An example is Yahoo! Directory (http://dir.yahoo.com/).

large number of assets, "analyzing which one best fits the needs – including items that are not well documented – and checking 'the whole stuff' is not a trivial task". One respondent pointed out that the major issues in reuse are "to quickly find the component you need", "having adequate documentation" and "having confidence that the component will work"; according to him/her, "these are the complicated things in reuse". A respondent stated that he/she has suggested organizations to "invest in a technical lead or architect to be responsible for selecting what will be reused. Such person would first make an analysis if the solution is appropriate to the context".

5.2. Logging of usage data and triggering of notifications

For collecting information about the usage data of reusable assets (Q4, concerning GRU 3), some respondents cited manual ways. In one of these, the reuse manager is responsible for capturing such information by analyzing software projects and searching for reuse occurrences, by storing results in an Excel spreadsheet (2) or a sort of list (1). Another form requires that, in case any asset is reused in a project, the reuse manager must be informed by the project manager (1) or by the development team (1). Other forms of accounting include cross-references (1), number of downloads (2), a tool developed by the organization (1), and issue trackers, like Mantis[22] (1).

The logging of usage data (GRU 3) is a recurrent complaint (mentioned by respondents in Q16, Q17 and Q18), which often becomes error-prone, since it is very dependent on people. During the implementation of GRU in an organization, one respondent identified that "the data related to assets' reuse that were reported did not correspond to what was actually done". In the accounting of reuse data, "the probability of error is enormous, because it is not known what has happened at all". According to the respondents, difficulties lie in verifying whether an asset was effectively reused ("deep down it is not known what the person has actually used"). This problem occurs frequently when the parameter for accounting is the number of downloads, which "is just an approximation, because a person may have downloaded it and not reused it in practice". In fact, "after [a reusable asset] that was in a repository is on a developer's machine, the person may use [such asset] without [formally] requesting [or communicating]".

One of the respondents stated that, in most organizations, reuse "works in a more dynamic way", so that a member can obtain an asset without having to undergo a "usage control". A solution pointed out by the respondent would be the use of a tool that is able to "scavenge" the code and check for the presence of a given asset. One concern (mentioned in Q17) is "to find an efficient way of accounting", "which is not only for meeting the model", but also one "that the organization can use for 'statistics', so that it can start evaluating cost-benefit". According to one respondent, such evaluation is the major issue, because "the organization wants to [perform] reuse knowing how much it is earning in return (e.g., in terms of reduction in rework)".

Regarding forms of notification to interested parties about changes in the status of assets (Q5, related to GRU 5), most respondents cited the use of e-mails (7). This is the only noticed

[22] http://www.mantisbt.org/

way up to that moment mentioned by some of the respondents (2), and the most frequent one to all of them. E-mails are sent manually, either to all the members of the organization (irrespective of being interested parties) (3) or from a list of interested parties, manually maintained based on the usage data (from GRU 3) (3). A respondent stated that organizations sometimes find it difficult to notify interested parties about changes in the status of assets.

There are also cases in which the Wiki or the issue tracking system contains a record of consumers (considered as interested parties) of a given reusable asset, so that only these consumers are notified by e-mail (by the system) about changes in the status of assets (2). Other forms of communication are a tool developed by the organization (1) and an adaptation of the concept of "followers" in social networks (1). In the latter, interested parties in a given asset must "follow" an asset in order to be informed of updates to the "profile" of such asset. This is similar to the Github's approach of "watching" repositories.

Concerning difficulties in evaluating reuse processes (Q17), one respondent pointed out difficulties in assessing the changes made to assets – in GRU 4 and its intersection with the Configuration Management (GCO[23]) process –, because "it is difficult to assess whether a developer has complied or not with the established level of control" on the evolution of reusable assets. One respondent cited the need for a tool to "scavenge" the repository to help assessing whether developers comply with the level of control previously established when evolving reusable assets, instead of creating blocking policies that may cause adverse effects. The respondent further stated that "the assessment of the communication between interested parties (GRU 5) is not trivial".

5.3. Reuse Management (GRU) implementations

With respect to the aspects that are more difficult to understand by the organizations (Q14), a respondent stated that "organizations find it difficult to understand what an evaluation criterion is, as well as the difference between acceptance and certification criteria" (GRU 1). An issue commented by one respondent is that organizations are "unaware" if the reuse manager role is a more managerial or a more technical assignment.

Most respondents emphasized (in Q18) that some identified difficulties related to the implementation of GRU are due to the field of activity or to the technology used by some organizations. A point raised is that reuse gets "more complicated" when developing more focused on innovation. Furthermore, when working with projects with some kind of control (e.g., confidentiality), "reusability is also limited", because one cannot "pick up a component that has been developed to a product and apply such component in another product, even if it fits perfectly, because there are confidentiality restrictions".

Two respondents reported a scenario in which the organization could not reuse code due to contractual issues with the client. It was a system for the banking sector and the organization "could not run the risk of reusing something external, because it could contain some malicious code that could pose a risk to the project". In this case, the organization had to opt for the reuse

[23] Acronym for "Gerência de Configuração", in Portuguese. This process is comprised in MR-MPS-SW level F (one level before level E) [SOFTEX 2012].

of process assets (although some respondents stated that they disagree with this approach of reuse). On the other hand, another respondent said that this situation may represent, in some cases, a "matter of convenience", as an organization "cannot reuse [an asset] in other client, but could reuse [an asset] from the same client in another project [of such client]".

Apart from the issue of the organization domain, the use of particular technologies may help or hinder the application of reuse in an organization. For example, one respondent stated that reuse "becomes easier" depending on the development paradigm used (e.g., "when it is object oriented"). Moreover, the high variation of technology in product development "can leave doubts as to earning money with reuse, which may end up postponing efforts in this respect".

Another factor mentioned by some respondents is the need for organizational culture and appropriate infrastructure to implement reuse processes. Most of the respondents cited the "lack of tool support" for the execution of these processes. Although the tools mentioned in the study partially assist GRU activities, organizations demand for applications that support the execution of these activities in an integrated way, allowing communication with other tools (e.g., configuration management tools), for more effective, visible and reliable results. For instance, as mentioned, it is difficult to make stakeholders obtain reusable assets from a specific mechanism, when the version control tool is already integrated to the development environment.

5.4. Development for Reuse (DRU) implementations

Most participants were involved in at least one assessment that included the DRU process. However, the MPS-SW model allows the exclusion of DRU 3 to DRU 9 outcomes from the scope of the assessment (when the organization shows that it does not have opportunities for reuse in its application domains (DRU 1) or does not have the required abilities to perform DRU at the moment (DRU 2) [SOFTEX 2012]). Consequently, only few assessments encompassed all the DRU outcomes. In this regard, one respondent stated that organizations often "prefer to claim that they do not have a domain of interest, in order to avoid committing to the next assessment"[24]. Another respondent affirmed that organizations frequently "state that they are not prepared" for such assessment.

Regarding the context in which DRU was implemented (Q6), organizations that implemented the entire DRU process "had the whole development structure towards reuse", according to one respondent. The core business of one of the organizations was to "sell software that serve as a basis for other developments, such as frameworks": "the organization has a product line and several different modules that support development". Another respondent stated that "the organization was a software factory, which builds component line domains", but "it was a very simple reuse, with few components in the [reuse] base [(i.e., in the repository)]".

Two respondents stated that they did not consider the planning and establishment of the reuse program as effective (Q7): one of them stated that "it stands on the threshold of what is expected in an assessment". Another respondent mentioned that "the planning and establishment of the reuse program are assessed particularly to that [assessment] context, and one cannot

[24] When reuse opportunities exist but the organization lacks reuse capacity, it must invest in qualification, and the outcomes become mandatory in the next assessment of the organization in the same level or above [SOFTEX 2012].

verified whether they are effective [along time]". Regarding the monitoring (Q8), one respondent affirmed that it happens "as a protocol formality", i.e., it is carried out only for the record; organizations "do not really keep up with it".

Regarding the way reuse proposals are made (Q9), one respondent stated that the analyst of the organization identifies a part of a project that could be reused in the future, and hence makes a request to the reuse group through a template (a Word form). According to the same respondent, the representation of domain models and domain architectures (Q10) were made in the Enterprise Architect tool[25], but members "did not use the most appropriate notation on domain engineering". Finally, although the respondent stated that domain assets were usually developed internally, no specific technique for their development (Q11) was mentioned.

Regarding the implementation of DRU (Q17), one respondent stated that "the parts related to architecture are harder to explain, understand, and define [in an organization]". Moreover, "many people cannot even understand the concepts involved in feature modeling". Another respondent said that the main challenge is the effectiveness of the implementation, so that it "does not become cosmetic", but "adds value to the organization" instead.

Similarly to GRU, the field of activity of the organization can also impact DRU implementations (mentioned in Q18), as stated by some respondents. A difficulty is related to how to implement DRU "when the organization always 'receives' [(i.e., is required to develop)] a different type of project" (i.e., projects are always different in nature or domain). According to one respondent, "DRU is difficult to equate, for there must be a change of philosophy in organizations".

A respondent emphasized that organizations in areas such as banking cannot do development for reuse because "it is the bank who makes the demand and the specification", and the organizations "only make the implementation of the project and deliver the product". In other words, they "do not have the opportunity to develop the architecture, or take architectural decisions inherent to the development for reuse".

One of the respondents stated that the need for tool support in DRU is greater than that observed in GRU, since "DRU is more complicated because it requires a more elaborated infrastructure". According to the respondent, an organization claimed that it did not implement DRU because it "still had no infrastructure for indeed developing for reuse". "They did not feel safe with infrastructure to do it systematically".

5.5. Perceptions related to organizations

Regarding the comments made by organizations concerning the GRU and DRU processes (Q12), some respondents mentioned that organizations usually do not have problems regarding GRU, and it is "well received" by them. Another respondent mentioned that organizations that use source code as a reusable asset consider the GRU process "very important", while many organizations that perform process reuse "do not even realize that they are implementing GRU", and it is "done basically 'for the record'"; thus, "it cannot properly be characterized as reuse".

[25] http://www.sparxsystems.com/products/ea/

Because the GRU process is mandatory (i.e., its outcomes are not subjected to exclusion in the context of an assessment in particular), some organizations "reuse any component just to say that they have [some] reuse". A respondent stated that organizations usually consider reuse a process "of little relevance". They get to implement the process, but do not have the "spirit of reuse".

Two respondents claimed that the most prominent shortage observed is the "short time of use" of the GRU process. Such process "lies among the latest ones to be implemented in an organization"; such neglect "results in a low number of reusable assets being managed". One of these respondents considered that the implementation of GRU in organizations has "low maturity", and both respondents stated that it is "made at the last minute before the assessment, sometimes almost on the eve".

In this sense, a respondent mentioned that GRU implementations conducted by him/her are done in parallel with the implementation of other management processes, because "the sooner it starts, the more people will experience if it is going to work in practice or not". What is noticeable is that "it is very rare for an organization to have no reuse activity", "not having some degree of reuse", because "they all have a lot of frameworks, a lot of components, web development, [...] a lot of reusable stuff; what they lack is the systematic way that is provided by the GRU process". Additionally, the respondent stated that, in his/her process implementations, he/she takes advantage of other processes that need to be (or are already) implemented – mainly Quality Assurance (GQA[26]), keeping criteria in audits for detecting any nonconformance related to GRU, in order to create and stimulate a reuse culture.

With respect to the aspects that are more difficult to understand by the organizations (Q14), one respondent affirmed that "the greatest difficulty does not lie in understanding the process itself, but in the day-to-day application of practices". Another respondent stated that he/she does not believe that the problem in GRU is in understanding, although organizations "do not understand very well how they will implement it". According to him/her, the main problem is that some organizations "lack knowledge from the software engineering point of view" and "lack understanding of the need for GRU". Organizations "have not yet realized how important it can be", otherwise "they might have more stuff" on a reuse repository.

Occasionally, organizations do not see the opportunity to implement the DRU process, which is sometimes questioned regarding its viability when the organization is still not sure whether to invest for reuse. According to one respondent, the view of organizations about DRU is that "it is something that will take a lot of work and will require a high investment", which causes them to claim that they "cannot afford to do it at that moment". Organizations also have trouble in knowing "what they are going to do with it", and in some cases, how they can "get rid" of this process. According to one respondent, DRU can be seen as an "additional expense for small enterprises that are unable to afford it". Organizations also may see DRU as "a great difficulty" and "want to try to 'go off on tangents'". Another respondent stated that organizations "do not even know what it is".

[26] Acronym for "Garantia da Qualidade", in Portuguese. This process is comprised in MR-MPS-SW level F (one level before level E) [SOFTEX 2012].

Concerning the most difficult tasks to be performed by the organizations (Q15), one respondent affirmed that he/she does not think that it is a matter of difficulty, but a motivation issue instead: organizations "do not see the importance" of reuse in day-to-day activities. A difficulty pointed out by one respondent lies in the application of evaluation criteria (GRU 1) in a systematic way. According to him/her, "defining criteria that make sense is not that difficult, but the application of such criteria is not trivial".

Many respondents stated that the number of problems ("required" items) identified during assessments (Q16) is usually low: "in organizations that do not neglect the implementation of the process, normally there are very few [reuse-related] items with problems"; "in GRU, it is not common to have many items with problems". However, one respondent stated that the way the GRU process implementation is required allows one to "meet all the GRU outcomes without showing a single project 'executing' GRU" (i.e., with an asset being effectively reused).

Required items in GRU 1 are related to the generality of the evaluation criteria ("sometimes the criteria are in a very high level or are very simple") or "the absence of defined criteria". Problems related to GRU 3 involve "the control of the accounting of reusable assets". Two respondents emphasized some required items in GRU 4 ("mainly [regarding] the control and the configuration management of assets", "[although] a modification in an asset should be demonstrated[27], such evidence does not exist") and GRU 5 ("when communication is not well defined", "problems related to notifications"). According to one of these respondents, "sometimes there is only one component which was made by the organization one week before [the assessment] and which was neither communicated [to interested parties] nor placed to use".

Regarding DRU, it is more common to have a required item due to the absence of a plan or a defined strategy for overcoming the "current lack of capacity" for executing the process (declared by organizations). Other items are not frequent because of the low number of complete DRU evaluations (i.e., including outcomes from DRU 3 onwards).

According to one respondent (in Q18), in most of the organizations whose products have more than 5 or 10 years (i.e., legacy systems), "software has a bad cohesion, high coupling etc.". He/she suggests that "the use of a technical debt tool (such as Sonar) or the use of a heat map of the product could help to identify critical points, and thus the organization could focus on the implementation of these points so that it perceives reuse payback faster".

Finally, some respondents with considerable experience in implementation and/or assessment stated (in Q18) that many organizations "have not yet understood the benefits of GRU", and sometimes implement this process more "to fulfill a requirement than for having understood that its goal is important". On the other hand, it was also stated that organizations "lack a lot of knowledge" and there is a "lack of market maturity" with respect to reuse. Additionally, some implementations seem to "follow a recipe", being necessary to invest in "training implementers and assessors" in reuse processes.

[27] GRU 4 implies that some modification must take place on some asset, so that such modifications can be communicated in GRU 5. However, this can only be possible if the asset is reused over time.

A complaint regarding the assessment of the organizations (also stated in Q18) was pointed out by another respondent: he/she posed that, during assessments, it should be required "to show at least one project with a change, the GRU process working", i.e., "to show that you reused something that is in the [reusable] asset base". According to the respondent, there should be a result, at least asking if some project used a reusable asset, in order to "show the instantiation of that outcome" and provide "a 'feeling' that people are reusing and things are working". He/she also claims that, unfortunately, there is actually room for a "makeup" in accounting for usage data, and currently "there is no resource on the day of assessment that allows assessors to request for showing the projects that used the component", because it "becomes too complicated to do it at that time [of the assessment]".

5.6. Perceptions related to stakeholders

Finally, regarding the point of view of the diverse stakeholders about reuse (Q13), some respondents affirmed that top management is acquainted to the GRU process, but in a macro level. One respondent commented that they lack a vision of how much the organization has gained with reuse – "such analysis, such indicator has not been explored the way it should be".

An example given was how much it has been saved by "hand labor" with reuse. For the business owner, the vision that there is a chance to gain money makes him/her want the organization to reuse as much as possible. However, such vision is not well explained to top management, and, according to the respondent, that might be the problem: it should be "better explained, with more emphasis on top management, 'we have chances of gains'. If this is realized, investment will be made".

According to one respondent, this vision of economy depends on the organization, and some fail to realize it (another respondent said that often, top management "does not even know what is going on"). "If a programmer must be stopped for developing a component, spending *x* more hours to have its future reuse, the interpretation of this scenario will largely depend on the organization profile". The respondent states that the problem is that "sometimes top management lacks a perception that spending 10 more hours in developing a reusable component can generate savings of *x* hours". In addition, a respondent stated that when knowledge management is more widespread, "hierarchical levels can have access to information more easily".

Regarding software developers, a respondent stated that, "when a component has no associated documentation, the reuser [or consumer] can go through a trial and error activity". However, "when using the component in the wrong way and seeing that it did not work [(which is a time-consuming task)], one can give up reusing it". Moreover, prospective consumers sometimes express misgiving in reusing something that was not made by them, reinforcing the NIH syndrome (discussed previously).

Additionally, according to one of the respondents, "programmers think that, for having an effective reuse, one needs to first refactor the code, or code snippets. But a product of 15 years is not refactored overnight". He/she stated that legacy software can be refactored for reuse while changes must be made: "typically, one starts from the defects pointed out by clients", and by addressing these points, refactoring is made "on that part of the product that the organization actually uses, and always maintains and evolves".

6. Analysis and Discussion

The kind of assets defined by the organizations as reusable (Q1) has a direct influence on the effectiveness and usefulness of reuse initiatives. For instance, one may not identify any benefit in monitoring kinds of assets that are rarely reused (because the visibility of benefits would be indeed impaired) or assets that do not have impact on the software development lifecycle. One of the reasons why source code assets are considered more often as reusable assets may be because any issue related to it (e.g., bug or missing feature) may lead to problems in software maintenance and evolution. Thus, it is important to choose assets whose information regarding reuse, evolution, and discontinuation is certainly relevant for the organization. Proper tools play a crucial role in this regard, especially in terms of collecting information about usage data (Q4) and notifying interested parties about changes in the status of assets (Q5). In fact, while the benefits of reuse are significant, there remain many technical challenges that must be addressed to fully realize this potential. The management of data related to reuse processes is a major challenge, and is one of the most easily recognized through this study.

It was noted that, in many cases, notifications about changes in the status of assets are triggered without any distinction of actual interested parties. This may compromise the effectiveness of communication (since information overloading may also adversely affect the perceived benefits of reuse), leading stakeholders to ignore important notifications. Another error-prone situation occurs when the maintenance of the list of interested parties and the sending of e-mails are performed manually (as in [Santos et al. 2009]). This can be partially due to limitations on tracking which team members reused which assets (i.e., collecting usage data), thus hindering communication.

The fact that some organizations store their reusable assets in version control repositories (Q2) has also been observed in other studies, such as [Morisio et al. 2002] and [Lucrédio et al. 2008] – in the latter study, the corresponding authors suggest that this may contribute to the failure of the efforts in fostering reuse. In this regard, it is noteworthy that each type of repository has features aimed at ensuring the better functioning for their intended purpose, and an inappropriate technology selection can hinder the adoption and implementation of reuse processes. Reuse repositories and configuration management repositories have different purposes (for instance, the former is optimized for searching operations, and should only contain releases of assets), and this must be taken into account when instantiating a repository for an organization. Moreover, the institutionalization of a reuse repository requires proper mechanisms for exploration, search, and retrieval of assets (Q3), allowing potential consumers to obtain information that can be useful for deciding whether a given asset should be reused or not.

For an appropriate awareness of the reuse scenario and communication of results in an effective way, monitoring mechanisms are crucial. Most of the identified ways for collecting and logging information about usage data (Q4) are error-prone, being very dependent on people's feedback. Besides misjudging the importance of monitoring when conducting a reuse program, the lack of availability and trustworthiness of reuse-related data may be some of the reasons why organizations do not keep up with monitoring practices. Mechanisms for data acquisition, cleaning, integration, aggregation, and representation play an essential role in this regard. In addition, due to the fact that people are the ones who make important decisions on reuse processes, such mechanisms must also account for facilitating analyses from the perspective of

humans. Research on visual analytics may provide some guidance in this topic [Thomas & Cook 2006] [Keim et al. 2008].

Concerning Development for Reuse, many organizations were not yet able to implement all of the outcomes, which makes it hard to draw any conclusions. However, we believe that more time should be spent on domain analysis and, if necessary, organizations should ask for consultancy or expert assistance aiming at a better understanding of the gains of implementing this process, as well as identifying which tools best serve this purpose. Technical difficulties or lack of knowledge on a particular notation/methodology should not make organizations avoid this process. Additionally, it seems that industry claims for more evidence (academic and especially industrial) on the benefits of adopting Development for Reuse.

For organizations that do not have the chance to realize the benefits reuse can bring to them, opportunistic reuse may seem to be enough. However, it does not seem fair to assume that the organization itself is the root cause of the problem. Some issues pointed out by this study can be due to the lack of proper preparation (of the organization members, the process implementers, or a combination of both) for the implementation of reuse processes. Neglecting the importance of such processes, putting them at the end of the list of the processes to be implemented, is also an aggravating factor (among others). This can be either an implementation strategy or an organization's decision. However, we claim this is not a good choice for the organizations: because of the delayed return on investments associated to reuse [Benedicenti et al. 1996], the benefits of reuse processes may take some time to arise and become noticeable; thus, the earlier they are implemented, the better.

The importance of finding what managers and developers need to understand about reuse is highlighted in [Kim & Stohr 1998]. Top management needs more awareness and visibility of relevant information of the reuse processes (as pointed out by [Morisio et al. 2002]), being able to measure and control the impact of a software reuse program. In other words, the value of reuse must be somehow established and communicated to managers [Kim & Stohr 1998], so that they can be aware and become committed to reuse initiatives. A vision of how much the organization has gained (or saved) in reuse is necessary, but other metrics can also be relevant for reuse initiatives. An issue pointed out in [Benedicenti et al. 1996] is that many organizations that manage to introduce reuse fail in tracking and controlling its evolution, especially if compared with organization objectives and mission.

Moreover, for a better acceptance of reuse-oriented changes in stakeholders' usual activities with less impact, suitable mechanisms must be identified and developed. Particularly, because some necessary steps for implementing reuse may be intimidating, organizations should try to accomplish them in a progressive way, in order to avoid resistance and allow for a better acceptance by the stakeholders (especially the reuse manager, who usually must have a more technical profile and may feel overburdened with organizational reuse tasks).

Finally, in order to institutionalize reuse processes, it is important to ensure that they are defined in organizations with a reasonable time frame before their assessment (preferably right after the definition of the development process). This is necessary so that there can be more chances of identifying candidate assets for reuse in the projects, as well as opportunities for evolving and identifying improvements in the reusable assets.

7. Final Remarks

This study provided some evidence for understanding the characteristics of the implementation of software reuse processes in Brazilian organizations. Since the beginning of software engineering, much research has been done in developing techniques and tools for supporting software reuse [Mili et al. 1995] [Frakes & Kang 2005]. In spite of that, many organizations still have difficulties in understanding and implementing reuse practices. As it can be noticed, many of the findings identified in the study match the literature reports both in the Brazilian and worldwide scenarios, particularly concerning the lack of adequate tool support and the need for more engagement in reuse initiatives. These are recurring problems.

Academia has been researching on software reuse for decades and proposing novel solutions for many problems (some of these problems have been pointed out by the industry, as stated in papers such as [Morisio et al. 2002] and [Poulin et al. 1993]). However, the software industry has not yet adopted many of the proposals so far. This problem is not restricted to the software reuse field – there is a late adoption of practices and tools in industry, sometimes justified by the need for evidence of effectiveness. The analysis points out that some needs from the industry that should be long gone still persist along the time.

Based on this observation, it can be perceived that a large gap remains between research and practice that needs to be bridged from academic research and industry practices. Both have advanced separately, but they do not seem to be going to the same direction, based on the reported findings. Thus, academia and industry should invest more effort in bridging the existing gap. This is one of the contributions of this work: by pointing out that some old problems still remain, we intend to make software industry invest on existing academic solutions, as well as stimulate academics to do more research on real industry problems. Motivation of organization members, cultural change, and other non-technical factors must be addressed as well.

Every organization must keep up with the evolution of assets they reuse, either developed by them or not. Therefore, it is important to improve their reuse capabilities proportionally to their current maturity stage. Otherwise, they are not expected to endure in the competitive market. We expect that, based on the study results, some organizations can perceive the need to accurately perform software reuse practices and go a step further, so that they can achieve higher maturity on software reuse practices.

A crucial concern is how to facilitate the acceptance/consciousness and adoption of reuse. Reuse stakeholders must be aware of the reuse results that are relevant to them and need awareness support for reuse tasks. Monitoring activities, for instance, allows early detection (and possibly resolution) of inconsistencies and shortcomings inside the software process, supporting and fostering the real integration of reuse paradigm into the existing software development process, encouraging continuous process improvement [Benedicenti et al. 1996]. Since there is a lot of information involved for performing reuse tasks, the lack of awareness and understanding of such information can hinder obtaining the expected results [Selby 2005] [Gill 2006].

We also believe that attempting to implement strategic reuse (and development for reuse) in any organization (regardless of its size) may be useless if their members do not actually perceive the benefits and gains of simpler reuse practices, such as reuse management and the

integration of software reuse tasks into the development process. These steps compose an assumption to enable organizations to better think about strategic reuse.

Finally, we hope that the observations of the presented study can be used as input for both research initiatives and the development of tool support for the implementation of reuse processes in software organizations. We also expect that this study can serve as a basis for other similar studies, in order to characterize software reuse in other scenarios and countries, thus contributing to a better understanding of the usual practices, problems, challenges, and opportunities for improvement on the implementation of software reuse processes worldwide.

7.1. Research limitations

During the conduction of this study, the following limitations were identified:

- the information about the organizations was obtained by implementers and assessors, and may represent a particular point of view of the respondents;
- the data analysis was performed by the interviewer himself, and there may have been problems of interpretation in the analysis of responses (however, the points on which there was uncertainty were confirmed with the respondents, both during the interviews and afterwards);
- it stands out that the results, while comprehensive, are not generalizable; and
- the whole study (including the interviews and the analysis) was conducted using the Portuguese language; thus, some English translation issues may have caused some deviations in the interpretation of results.

7.2. Open questions and directions for future research

Some suggestions for future work for overcoming some study limitations in the future include:

- For obtaining the point of view of the members of organizations regarding the stated questions, semi-structured interviews could be conducted with a limited number of organizations; one could also resort to surveys, in order to achieve a larger scope of organizations (taking into account the problems stated in Section 3.4). Another aspect that could be encompassed by a study with software organizations would be what industry requires from the research community.
- It would also be interesting to conduct studies with the research community worldwide in the software reuse field, in order to identify the academic point of view of the discussed problems and find out ways to convince industry to delve into current research literature and dedicate more time (or sufficient time) to reuse. This would help us bridge the (still large) existing gap between research and practice.

This study is part of an ongoing research that aims at supporting software reuse needs with awareness resources [Schots 2014]. Its results are being used as input for identifying desirable and necessary features for supporting reuse through visualization as a means of awareness. We believe that an appropriate awareness of the reuse scenario can help communicating results to stakeholders more effectively.

Acknowledgements

The authors would like to thank:

- all the interviewees, for the readiness, availability and willingness to participate in this study,
- professor Carolyn Seaman, for the encouragement and the theoretical contributions used in the preparation of this study,
- professor Natália Schots, for kindly supporting the data verification and aggregation during the analysis,
- Maria Helena Oliveira, for partially supporting the transcription and verification of interviews, and
- CAPES, FAPERJ and CNPq, for the financial support.

References

ABES (2014). "Brazilian Software Market: Scenario and Trends". Associação Brasileira das Empresas de Software, 1st ed., São Paulo, June. ISBN 978-8586700-03-3. Available at http://www.abessoftware.com.br/dados-do-setor/.

Basili, V., Caldiera, G., Rombach, H. (1994). "Goal Question Metric Paradigm", *Encyclopedia of Software Engineering*, v. 1, edited by John J. Marciniak, John Wiley & Sons, pp. 528-532.

Benedicenti, L., Succi, G., Valerio, A., Vernazza, T. (1996). "Monitoring the efficiency of a reuse program". *SIGAPP Applied Computing Review*, v. 4, n. 2, pp. 8-14, September.

Braga, R. M. M., Werner, C. M. L., Mattoso, M. (2006). "Odyssey-Search: A Multi-Agent System for Component Information Search and Retrieval". *Journal of Systems and Software*, v. 79, n. 2, pp. 204-215, February.

Card, D., Comer, E. (1994). "Why do so many reuse programs fail?". *IEEE Software*, v. 11, n. 5, pp. 114-115, September.

CMMI Product Team (2010). "CMMI for Development, Version 1.3", Technical Report CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA. Available at http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm.

Frakes, W. B., Fox, C. J. (1995). "Sixteen questions about software reuse". *Communications of the ACM*, v. 38, n. 6, pp. 75-87, June.

Frakes, W. B., Kang, K. (2005). "Software reuse research: Status and future". *IEEE Transactions on Software Engineering*, v.31, n.7, pp. 529-536.

Gill, N. S. (2006). "Importance of Software Component Characterization for Better Software Reusability". *SIGSOFT Software Engineering Notes*, v. 31, n. 1, pp. 1-3, January.

Griss, M. L., Favaro, J., Walton, P. (1994). “Managerial and organizational issues – Starting and Running a Software Reuse Program”. In: Schäfer, W., Prieto-Diaz, R., Matsumoto, M. (eds.), *Software Reusability*, pp. 51-78, Ellis Horwood Ltd..

Haefliger, S., von Krogh, G., Spaeth, S. (2008). “Code Reuse in Open Source Software”. *Management Science*, v. 54, n. 1, pp. 180-193, January.

Holmes, R. (2008). “Pragmatic software reuse”. Ph.D. Thesis, University of Calgary, Calgary, Canada.

Hove, S. E., Anda, B. (2005). “Experiences from conducting semi-structured interviews in empirical software engineering research”. In: *11th IEEE International Software Metrics Symposium*, pp. 1-10, September.

IEEE (2010). “IEEE Std 1517-2010: IEEE Standard for Information Technology – System and Software Life Cycle Processes – Reuse Processes”, 39p, Institute of Electrical and Electronics Engineers.

ISO/IEC (2008). “ISO/IEC 12207:2008 – Systems and software engineering – Software life cycle processes”, 123p, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

ISO/IEC (2012). “ISO/IEC 15504-5:2012 – Process assessment, Part 5: An exemplar software life cycle process assessment model”, 196p, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., & Ziegler, H. (2008). “Visual analytics: Scope and challenges”. In: Simoff, S. J., Böhlen, M. H., Mazeika, A. (eds.), *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, pp. 76-90, Springer Berlin Heidelberg.

Kim, Y., Stohr, E.A. (1998). “Software reuse: survey and research directions”. *Journal of Management Information Systems*, v. 14, n. 4, pp. 113-147, March.

Kitchenham, B. A., Pfleeger, S. L. (2008). “Personal opinion surveys”. *Guide to Advanced Empirical Software Engineering*, Springer London, pp. 63-92.

Krueger, C. W. (1992). “Software reuse”. *ACM Computing Surveys*, v. 24, n. 2, pp. 131-183, June.

Leach, R. J. (2012). Software Reuse: Methods, Models, and Costs, 2nd ed., ISBN-13: 978-0-9853685-1-7, 656p.

Lucrédio, D., Brito, K. S., Alvaro, A., Garcia, V. C., Almeida, E. S., Fortes, R. P. M., Meira, S. L. (2008). “Software reuse: The Brazilian industry scenario”. *Journal of Systems and Software*, v. 81, n. 6, pp. 996-1013, June.

Marshall, S. Jackson, K., Anslow, C., Biddle, R. (2003). “Aspects to visualising reusable components”. In: *Australasian Symposium on Information Visualisation (InVis.au 2003)*, Adelaide, Australia, pp. 81-88, February.

Mili, H., Mili, F., Mili, A. (1995). “Reusing software: Issues and research directions”. *IEEE Transactions on Software Engineering*, v.21. n.6, pp. 528-562.

Moore, J. M., Bailin, S. C. (1991). “Domain Analysis: Framework for reuse”. In: Prieto-Díaz, R., Arango, G. (eds.), *Domain Analysis and Software System Modeling*, pp. 179-202, IEEE Computer Society Press, Los Alamitos, USA.

Morisio, M., Ezran, M. and Tully, C. (2002). Success and failure factors in software reuse. *IEEE Transactions on Software Engineering*, v.28, n.4, pp. 340-357.

Naur, P., Randell, B. (1968). “Software Engineering”, Scientific Affairs Division, NATO, Brussels, Garmisch, Germany, Report on a conference sponsored by the NATO Science Committee, October.

Poulin, J. S., Caruso, J. M., Hancock, D. R. (1993). “The Business Case for Software Reuse”. *IBM Systems Journal*, v. 32, n. 4, pp. 567-594, October.

Prieto-Diaz, R., Arango, G. F. (1991). *Domain Analysis and Software Systems Modeling*. IEEE Computer Society Press.

Rocha, A. R. C., Montoni, M., Weber, K. C., Araujo, E. E. R. (2007). “A Nationwide Program for Software Process Improvement in Brazil”. In: *6th International Conference on Quality of Information and Communications Technology (QUATIC 2007)*, Lisbon, Portugal, pp. 167-176, September.

Sá, M. L. B., Werner, C. M. L., Goldman, I. (1997). “Introduction of Reuse in a Brazilian Enterprise on Software Production” [Introdução da Reutilização em uma Empresa Brasileira de Produção de Software] (in Portuguese). In: *11th Brazilian Symposium on Software Engineering (SBES 1997)*, Fortaleza, Brazil, pp. 233-248, October.

Santos, G., Zanetti, D., Maciel, M., Simões, C. A., Werner, C., Rocha, A. R. (2009). “The Experience of the Implementation of Reuse Management and Development for Reuse Processes in Synapsis-Brazil” [A Experiência de Implantação dos Processos Gerência de Reutilização e Desenvolvimento para Reutilização na Synapsis-Brasil] (in Portuguese). In: *5th Annual Workshop of MPS (WAMPS)*, Campinas, Brazil, pp. 128-135, October.

Schmidt, D. C. (1999). “Why software reuse has failed and how to make it work for you”. *C++ Report*, SIGS Publications Group, January.

Schots, M. (2014). “On the Use of Visualization for Supporting Software Reuse”. In: *36th International Conference on Software Engineering (ICSE 2014), Doctoral Symposium*, Hyderabad, India, pp. 704-707, June.

Seaman, C. B. (1999). “Qualitative methods in empirical studies of software engineering”. *IEEE Transactions on Software Engineering*, v. 25, n. 4, pp. 557-572, July.

Selby, R. W. (2005). “Enabling reuse-based software development of large-scale systems”. *IEEE Transactions on Software Engineering*, v. 31, n. 6, pp. 495-510, June.

Sherif, K., Vinze, A. (2003). “Barriers to adoption of software reuse: A qualitative study”. *Information & Management*, v. 41, n. 2, pp. 159-175, December.

Silva Filho, R. C., Katsurayama, A. E., Santos, G., Murta, L., Rocha, A. R. (2008). “Deploying Software Reuse Management at COPPE/UFRJ Software Engineering Laboratory”. In: *1st Workshop on Software Reuse Efforts (WSRE)*, Recife, Brazil, pp. 1-5, November.

SOFTEX (2012). “MPS.BR – Brazilian Software Process Improvement – MPS Software General Guide” [MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral MPS de Software] (in Portuguese), August. Available at http://www.softex.br/mpsbr/guias.

SOFTEX (2013). “Assessment Guide”, August. Available at http://www.softex.br/mpsbr/english-guides/.

SOFTEX (2014). “MPS-SW Published Assessments” [Avaliações MPS-SW (Software) Publicadas] (in Portuguese), August. Available at http://www.softex.br/mpsbr/avaliacoes/mps-sw/.

Thomas, J. J., Cook, K. A. (2006) “A visual analytics agenda”. *IEEE Computer Graphics and Applications*, v. 26, n. 1, pp. 10-13.

Travassos, G. H., Kalinowski, M. (2012). “iMPS 2012: Evidence on Performance of Organizations that Adopted the MPS-SW Model since 2008”. Available at http://www.softex.br/wp-content/uploads/2013/08/iMPS-2012.pdf.

Travassos, G. H., Kalinowski, M. (2013). “iMPS 2013: Evidence on Performance of Organizations that Adopted the MPS-SW Model” [iMPS 2013: Evidências Sobre o Desempenho das Empresas que Adotaram o Modelo MPS-SW] (in Portuguese). Available at http://www.softex.br/wp-content/uploads/2013/08/Livro_iMPS-2013-PT_v3.pdf.

Ye, Y., Fischer, G., Reeves, B. (2000). “Integrating Active Information Delivery and Reuse Repository Systems”. In: *ACM SIGSOFT Symposium on Foundations on Software Engineering (FSE 2000)*, San Diego, USA, pp. 60-68, November.

Ye, Y., Fischer, G. (2002). “Supporting reuse by delivering task-relevant and personalized information”. In: *24th International Conference on Software Engineering (ICSE 2002)*, Orlando, USA, pp. 513-523, May.