

Adopting Configuration Management Principles for Managing Experiment Materials in Families of Experiments

Edison Espinosa

Departamento de Computación, Universidad de las Fuerzas Armadas ESPE
Av. General Rumiñahui s/n y Paseo Escénico Santa Clara. Quito, Ecuador
egespinosal@espe.edu.ec

Silvia Teresita Acuña

Escuela Politécnica Superior, Universidad Autónoma de Madrid
Calle Francisco Tomás y Valiente 11, 28049 Madrid, Spain
silvia.acunna@uam.es

Sira Vegas

Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid
Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain
svegas@fi.upm.es

Natalia Juristo

Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid
Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain
natalia@fi.upm.es

Abstract

Context

Replication is a key component of experimentation for verifying previous results and findings. Experiment replication requires products like documentation describing the baseline experiment and a version of the experimental material. When replicating an experiment, changes may have to be made to some of the products, leading to new or modified versions of materials. After the replication has been conducted, part of or all the materials should be added to the family history or to the baseline experiment documentation. As the number of replications increases, more versions of the materials are generated. This can lead to product management chaos in replications sharing the same protocol.

Objective

The aim of this paper is to adopt configuration management principles to manage experimental materials. We apply and validate these principles in a code inspection technique comparison experiment and a personality quasi-experiment.

Method

The study was conducted within a research group with lengthy experience in experiment replication. This research group has had trouble with the management of the materials used to run some of the experiments replicated by other colleagues. This is a suitable context for applying action research. We used action research to adopt the configuration management principles and build a materials management framework.

Result

We generated the instances of an experiment and a quasi-experiment, identifying the status and traceability of the materials. Additionally, we documented the workload required for instantiation in person-hours. We also checked the ease of use and understanding of the framework for instantiating the personality quasi-experiment configuration plan executed by researchers who did not develop the framework, as well as its usefulness for managing the experimental materials.

Conclusion

The experimental materials management framework is useful for establishing the status and traceability of the experimental materials. Additionally, it improves the storage, search, location and retrieval of the experimental material versions.

Keywords

Experimental software engineering, experiment replication, experimental material, experimental software configuration management.

1 Introduction

Software engineering experimentation is a necessary, but complex, process that relies for support on mechanisms to motivate researchers to conduct studies and integrate results. It depends, in turn, on a community of experimenters engaged in experiment replication [1]. Replication is necessary for validating the empirical results published by software engineering researchers [2]. Currently, replications usually produce different experimental results that are hard to reconcile with the baseline experiment findings, as there are a host of factors that can alter the experimental results and are hard to predict beforehand [3].

Experiment replication requires products such as documentation describing the baseline experiment and a version of the experimental materials. When replicating an experiment, changes may have to be made to some of the products. Changes are the result of the execution of the replication in a different context, defects found in the materials or the evolution of the research. The changes lead to new or modified versions of materials, and part of or all the materials should be added to the family history or baseline experiment documentation at the end of the replication. As the number of replications increases, the experimental material versions do so as well. This may lead to confusion and disorder, causing product management chaos in replications sharing the same protocol [4].

The aim of this research is to adopt the principles of software configuration management to manage experimental materials in experimentation. The study was carried out within a research group with lengthy research experience in experimental replications. This research group administers a number of experiments, from which we selected an experiment comparing code inspection techniques for a pilot study [5]. The available files experimental plans, objects, materials, experimental data, results and analyses, and publications describe the experiment and product versions. These documents are the result of experiment executions at different sites and by different researchers for a period of over ten years. This context is suitable for applying the action research method [6], because it is a real-world problem occurring within a research group that has problems with the materials that it administers and employs to execute experiments replicated by other colleagues.

In this paper, we report the process enacted to adopt software configuration management for materials management in order to improve the replication process. Software configuration is commonly used in software engineering to ascertain the status and traceability of intermediate and final products of the software development process [7]. We generated an experimental material management framework based on information from two sources: knowledge elicitation sessions that were conducted with experts in experimentation and a study of materials from experiment replications conducted from 2000 to 2014. We also established an evaluation process that aimed to measure the feasibility, usability and usefulness of the framework. Feasibility assesses how well the experiment configuration management plan can be instantiated based on existing information and material about the experiments. Usability measures the ease of use and understanding of the framework after instantiating the above plan. Usefulness evaluates researcher satisfaction with the information recorded in the configuration management plan and specifically validates whether it is possible to identify from this information the status and traceability of the different experimental materials entered in the configuration management plan. It also checks whether the information on the experimental materials can be located by searching the instruments that are part of the plan. Two types of evaluation were conducted: internal and external. The internal evaluation validated feasibility, whereas the external evaluation rated feasibility, usability and usefulness. We instantiated two experiments. First, we applied configuration management to the code inspection technique comparison experiment [4], which was used to conduct the internal evaluation. Second, we validated configuration management on a personality quasi-experiment [8] through an external evaluation of the framework. By generating these instances, we were able to identify the workload in person-hours required for instantiation. Additionally, we found that the personality quasi-experiment configuration plan could be instantiated using the framework, which was usable and useful for identifying the status and traceability of the experimental materials within the instantiated quasi-experiment configuration management plan. This study demonstrates that the information on each

instance brings to light the status and traceability of the experimental materials. This study provides the research group with a framework for managing the experimental material to clarify the evolution and status of the experiments, replications and materials.

The article is structured as follows. Section 2 describes the background. Section 3 details related work. Section 4 specifies the research method. Section 5 describes the framework for experimental material management. Section 6 specifies the procedure for evaluating the proposed framework. Section 7 shows the experiment instantiation (internal evaluation). Section 8 describes the quasi-experiment instantiation (external evaluation). Section 9 discusses the results. Finally, Section 10 outlines the conclusions of the research.

2 Background

In the life sciences, experimental processes are increasingly complex and commonly related to data and experimental results acquisition, management and analysis activities [9]. The biomedical community has detected problems with the replication of experiments in cancer biology caused by problems of access to information and materials [10]. Additionally, chemical, biological, physical and environmental scientists have come up against the problem of the reproducibility of experimental results caused by non-existent experimental designs, reagent (material) variability, raw data, etc. [11]. These are similar issues to software engineering experimentation issues, where researchers aim to improve and increase the number of experiment replications.

Several instruments for transmitting information among experimenters have been proposed in order to perform experimental software engineering replications [12]. Initially, the published experiment reports were used as information to generate the materials for replicating the experiment [13]. These materials are essential and are used to execute replications in other disciplines [14]. In order to promote replications, several researchers developed replication packages (static materials containers) [15-17]. Later, software environments were created to support experimentation. These environments range from a simple experimental materials repository to the definition and implementation of tools to be used in the experimental process activities. Examples of these environments are SESE [18] and eSEE [19, 20]. However, the evolution of these instruments designed to improve and increase replication execution led to more than one version of the materials being available at any one time, causing confusion about which version of the experimental material should be used for a new replication [21].

The experimental elements may vary in the replication depending on the goal of the replication [22]. Such variations are useful for the purposes of verifying the results of an experiment and also for learning [23]. Variations in the replications lead to changes in the experimental elements and materials, resulting in new or modified versions of the elements. This leads to the problem of there being multiple versions of experimental materials and replications of an experiment [17].

In the case of materials, there are versions (current or obsolete) created and/or modified as a result of both variations adopted in the replications with respect to the baseline experiment and the evolution of the experiment according to the experimental paradigm. The information available in replications regarding materials, raw data, data analysis, publications, etc., is usually incomplete, disperse and vague. Additionally, the different versions of materials and information on replications usually have different contents in different formats stored at different sites and are commonly used by different experimenters to execute experiments (even within the same time periods). This was discovered when we analysed the documents, materials and replications of the code inspection technique comparison experiment administered by the group conducting the research. Other software engineering experiment research groups appear to have met with similar problems in their experiments [17, 21].

The specification of experimental materials management and replications to support experimentation and replication processes will help experimenters to:

1. Ascertain the status and traceability of the materials, experimental results and replications of each experiment
2. Identify the experimental materials and results used in each replication
3. Support the search, location and retrieval of different versions of materials for executing new replications of an experiment.

If experimenters do not have efficient, adequate and timely information about the status and traceability of experiments, replications and materials, they cannot make good decisions on how to continue replicating and

evolving the experiment. This problem often leads to a wasteful expenditure of time, money and effort on planning, design, execution and analysis for executing new replications.

The management of software engineering experimental materials and replications is an open and extremely important question [4, 18]. Additionally, it is a novel and underresearched domain, as there are currently no satisfactory approaches [14]. We also believe that the results reported in this article provide support for the above three points within experimental materials management.

3 Related Work

The literature review on experimental materials management in families of experiments was based on two systematic literature searches planned and executed on the Scopus, IEEE, ACM and SpringerLink databases up to July 2014 [24] and was recently updated to March 2019. Both reviews established: 1) the inclusion and exclusion criteria, 2) the search process of the above databases for papers, and 3) the analysis of retrieved papers. In the first review on research on the management experimental materials and their use in replications, the inclusion criteria accepted studies that described version control activities on at least one experimental material, whereas the exclusion criteria rejected studies that detailed version control on products unrelated to software engineering experimentation. In the papers search process, search strings were formed and used to search the above databases. The strings were devised by experts in experimentation and the researchers who designed the framework. They used the following terms and other synonyms: experimental material, replication packages, infrastructure, repository, support and materials management. The strings were entered in the database search for field. The search returned a total of 36 papers to which the inclusion and exclusion criteria were applied, leading to the retrieval of the five papers detailed in Section 3.1.

In the second review on replication management tools and their experimental materials [25], the inclusion criteria accepted studies that provided evidence of the use of web systems to support experiment replications or demonstrated the use of offline systems to share materials under licence by electronic mail, whereas the exclusion criteria rejected studies that were merely experimental reports or contained inactive web addresses. The paper search process was started by forming the search strings using the following terms and other synonyms: software engineering, experimentation and support for replication. These search strings were entered in the database search field. The search returned a total of 24 papers to which the inclusion and exclusion criteria were applied, leading to the selection of another four studies detailed in Section 3.2. Finally, from the analysis of the papers retrieved by the two reviews, we found that all the papers aimed to improve the experiment replication process. Different authors described the importance of conducting experimental materials management for replicating experiments in software engineering [4, 17, 20]. Good materials management helps researchers decide which materials should be used for replication in a particular context.

3.1 Research on Experimental Materials Management and Use in Replications

The literature review on research into experimental materials management and use in replications uncovered evidence of research aiming to improve the experiment replication process in software engineering. Shull et al. [4] address the problem of the knowledge to be transmitted between experimenters to increase the number of, and improve, replications. They propose the use of a collaborative structure that uses laboratory packages combined with several means of communication. However, they had trouble establishing and adapting materials for conducting replications as a result of there being multiple versions of experimental materials for a particular experiment.

Mendonça et al. [21] address the problems of knowledge transmission between research groups in order to enact the experiment replication process. The solution that they propose is FIRE (Framework for Improving the Replication of Experiments). Additionally, they claim that a key question for replication is to provide a package that contains the materials. Besides, they found that materials evolve without appropriate version control.

Shull et al. [17] address the problem of tacit knowledge transfer between experimenters for improving experiment replication processes, complementary studies and conclusions. The proposed solution is a collaborative structure based on a laboratory package. However, they claim that it is troublesome to build the package because there are several versions of artefacts (experimental materials) as a result of the evolution of the experiment over time.

Do et al. [20] address the following problems: small number of experiments on testing technique effectiveness, shortage of empirical data and costs of executing controlled experiments in software engineering. Additionally, they stress that it is tough to retrieve and organize the correct versions of the artefacts (materials) to support controlled experiments. The proposed solution is an infrastructure to support controlled testing technique and regression testing experiments.

Solari et al. [26] addressed this problem with a view to discovering which information is required to successfully replicate an experiment. To do this, they examined a total of seven completed replications. Based on these replications, they identified a set of incidents, which they used to generate a set of documents, instruments and materials that were put together as a laboratory package. However, they point out that there was no version control of the instruments and materials belonging to the laboratory package.

In sum, from the analysis of the retrieved studies, we found that their authors propose different solutions to the problem of improving the experiment replication process, but all the researchers point to critical experimental material version control problems caused by the evolution of the experiment.

3.2 Research on Replication and Experimental Material Management Tools

The papers retrieved on replication and experimental material management tools aim to support the replication process by means of functionalities implemented in software systems. We categorized these papers as repositories and experimental packages based on: 1) the approach to the addressed problem type, 2) the information analysis described by the researchers, and 3) the evaluation of the functionalities of the system supporting the software engineering experiment replication process.

With regard to the repositories, we analysed two papers. Shull et al. [4] store laboratory packages, workshops, materials and associated documents to provide support for the replication process. Anyone can download the experimental materials in the package from the web. Do et al. [20] designed and built a system to provide support for controlled experiments on testing and regression testing techniques. Additionally, this system can select and organize the latest version of the objects (experimental materials) for experiment replication.

With regard to experimentation packages, the system developed by Basili et al. [27] to manage their package, merely provides functionalities for gathering the materials to replicate the experiment. Shull [28] built a system that administers this package based on a web site for downloading some of the materials and data gathered from the baseline experiment.

In conclusion, we have found that the systems that administer experimental materials have limitations with respect to the following: 1) most systems do not provide the option of adding information on new or modified versions of materials, 2) most systems do not provide information for establishing the status and traceability of materials, results and replications within the research, and 3) none of the studies provide information for generating experiment variants.

Therefore, we can say that the experimental materials management problem has not yet been fully solved in experimental software engineering. The materials management framework proposed in this paper can provide change control for experimental materials within a family of experiments.

4 Research Method

We applied the action research research method. Action research is an iterative, participatory, collaborative scientific research method aimed at solving a real problem in a specific context. It is iterative because the solution is reached by means of approximations driven by the execution of cycles. It is participatory because the researcher and practitioner roles are actively involved in the search for problem-solving approaches. It is collaborative because each role performs action research activities jointly to find the solution to the problem [29, 30]. According to Koshi et al. [31], the characteristics of action research are as follows:

- It is based on a problem situation in a specific context.
- It is used to improve practice based on the application of a theory in a real situation.
- It is participatory and collaborative because it actively involves a group of people jointly performing activities in pursuit of a common aim.
- Its development requires the participants to reflect on and interpret whether the problem was solved.
- Knowledge is generated by actions performed at the site where the research is applied.
- The findings emerge with the development of the actions. However, they are neither conclusive nor absolute.

4.1 Application of Action Research

The application of action research to address the problem of experimental material and experiment replication management is supported by the following points:

- There is a group of experimenters that are very experienced in experiment replication. They use several forms and means of communication, like face-to-face meetings, electronic mail, videoconference, chat, etc., to share knowledge.
- The stated problem occurs in a research group administering a number of experiments where there are a lot of versions of documents, materials, objects, instruments and publications. These versions are the result of several replications conducted over more than a decade at different sites and in different settings.
- There is a software configuration management paradigm that can be adopted to improve the management of materials in software engineering experiment. Software configuration management can be applied to the management of intermediate and final products output by the software development process. The application of this paradigm has proved to be fairly effective within software engineering, because it can mitigate the product management chaos within the software development process [32, 33].

4.2 Cycles for Adopting the Configuration Management Paradigm in Experimentation

In order to apply action research in our research, we planned and executed activities that required one or more cycles, in which we applied literature review and knowledge elicitation techniques. These activities output information and knowledge on software configuration management, as well as software development and experimentation processes and activities. Figure 1 shows the cycles used to adopt software configuration management for experimentation. The first activity targeted the adoption of software configuration management concepts in experimentation. It took one cycle, in which each software configuration management concept was singled out and studied. In this cycle, we applied the interview and literature review techniques, and the result was the experiment configuration management (ECM) document. The second activity aimed to build the instruments for ECM. This activity took two cycles, during which the instruments to support the materials management activities were built. In these cycles, the interview technique was applied to validate the instruments. This resulted in two versions of instruments for supporting the materials entry, monitoring and control activities. The third activity was designed to adopt concepts and instruments in the experiment configuration management plan (ECMP). This took two cycles during which the concepts and instruments were added to the different sections of the plan proposed by [7] and yielded the ECMP. Finally, the ECMP instantiation activity was developed by enacting two cycles. In this activity, information on the experiment, replications and experimental materials was entered in the different instruments in order to generate the instance of the code inspection technique comparison experiment management plan. Each of the activities is described in more detail in the following.

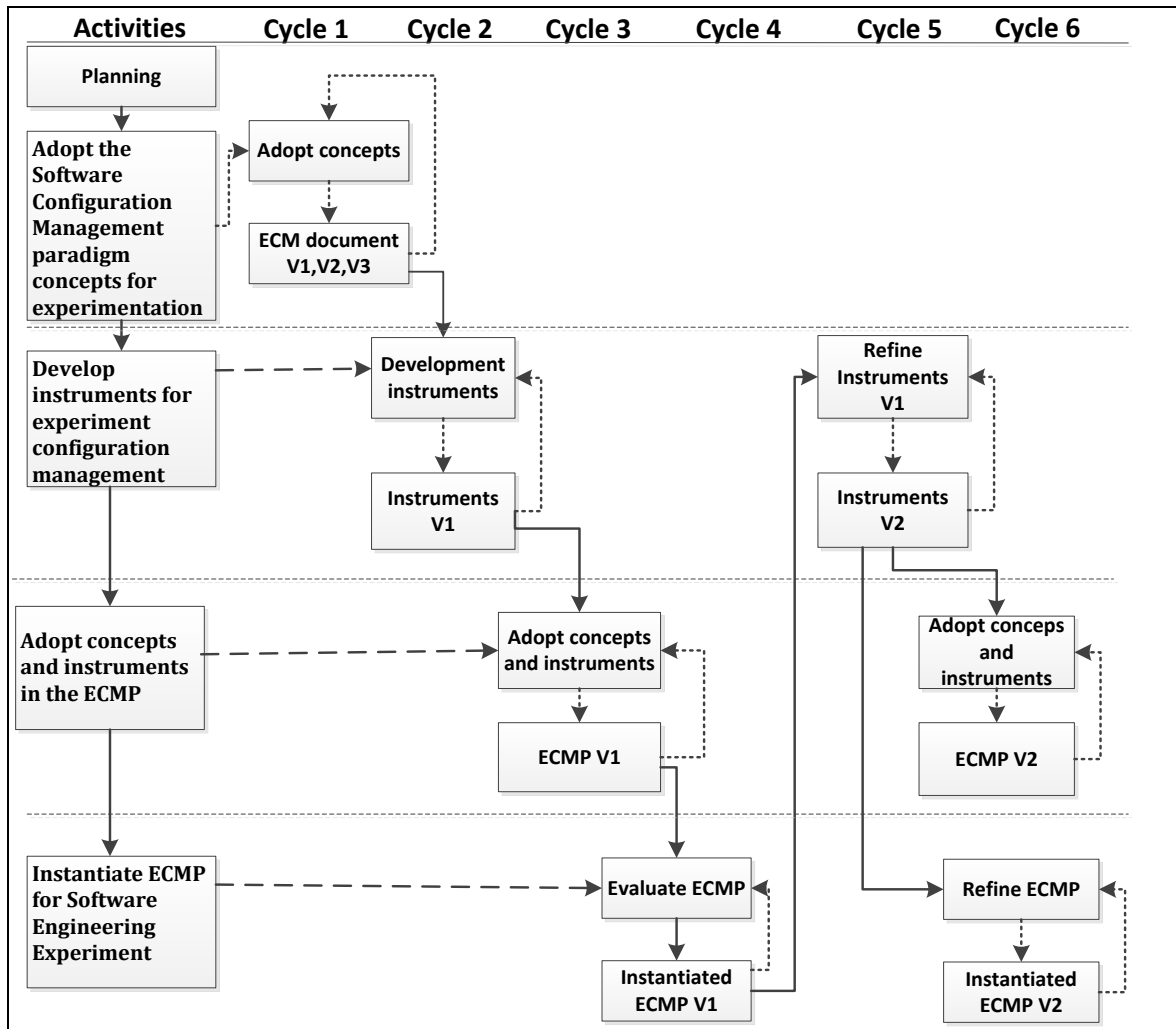


Figure 1. Activities and cycles for adopting software configuration management in experimentation

In the **adopt the software configuration management paradigm concepts for experimentation** activity, we analysed the process of experimentation as a transformation of products. We started by studying the software development and experimentation processes to identify their respective activities, products and intermediate and final results. We then developed an analogy between the concepts, singling out and studying the software configuration management concepts applicable for adoption in experimentation.

ISO/IEC/IEEE 12207: 2017 [34] establishes a common framework for software lifecycle processes containing processes, activities and tasks. This standard is composed of primary, support and organizational processes. The software development process is one of the primary processes. The software development process is composed of the following activities and their specific tasks: system requirements analysis, system design, software requirements analysis, software architecture design, detailed software design, software coding and testing, software integration, etc. These activities and tasks are carried out to create products that are part of the software system [35]. These products are documents, designs, diagrams, source code, object code, databases, etc. They evolve over time due to the changes that take place as a result of applying inspection or evaluation activities. The software configuration management process is a support process established to control the changes to software products in the software life cycle [32].

Within the experimental software engineering process, a set of activities are, according to Wohlin et al. [36], likewise, often carried out in an established order. According to Juristo and Moreno [37], the experimentation activities are: definition of the experiment objectives, experiment design, experiment execution and results analysis. As part of each of these activities, several tasks are performed and a number of products and results are output. Some publications of experiments, like [38, 39], specify that experimental materials are handed out to each experimental subject in the experiment execution sessions. Some of these materials are created and/or modified in response to changes from one experiment to another [22]. Such changes lead to an increased number of versions of materials, causing problems with regard to replication materials management

processes for researchers. These problems have a direct impact on the time, costs and effort required to perform the experimental replication process [1].

Based on the software development approach in which the process transforms a product, we analysed and described the concepts that are commonly used in the software development life cycle to search for a definition or, failing this, approximate how they could be applied in the experimental research cycle. In the following, we use the example of the process concept that is applied in the software development life cycle as part of the analysis of the analogy that we developed for the adoption of ECM.

In the software development life cycle, the process concept defines a set of activities to be executed by the developer team to create the different products (requirements specification, databases, mock-ups, source code, target code, tests, etc.) of a software project [35]. In the experimental research cycle, on the other hand, the experimental process establishes a set of activities that produce certain (intermediate or final) products or results necessary for the experiment [36, 37]. A similar reasoning is applied to all the concepts that are shown in Table 1 detailing the analogy between concepts of the software development and experimental research cycle.

Table 1. Software development cycle and experimental research cycle concepts analogy

Concept	Software Development Cycle	Experimental Research Cycle
Process	A set of activities to be executed by the developer team to create the different products	A set of activities that produce certain products necessary for the experiment
Phase	A period of time in the software development cycle during which user needs are described and evaluated in documents	Undefined, but applicable considering that the experimentation process has a set of phases
Product	The result of the application of methods, techniques and tools in a specific development process phase	Undefined, but applicable considering that several tasks are carried out and several experimental process products are produced in each phase
Change	The evolution over time of the intermediate and final products resulting from the application of the software process as a result of their use or maintenance through user interaction	Undefined, but applicable considering that the experimental materials and results evolve over time as changes are made to experiment to validate new knowledge
Relations	Connection between the intermediate and final products resulting from software development process application	Undefined, but applicable considering that the experimental materials and results are not isolated elements

The above description demonstrates that there is a process-product duality and that the management of intermediate and final product versions is troublesome in both cases. Software configuration management is used in the software development process for product version change control, whereas, in experimentation, there is no process for controlling the status and evolution of materials.

Software configuration management is based on the following concepts: configuration, configuration element, software configuration, software configuration element, configuration management, software configuration management, baseline, version, revision, release, library [35, 40].

We analysed the definition of these concepts and detailed the arguments that justify their adoption or adaptation in ECM. We proceeded based on the documents output by the expert elicitation process and experience gathered from participating in the execution of an experiment replication [41]. Finally, we defined the ECM concept. In the following, we give an example to illustrate the development of the concept analogy.

In software configuration management, a software configuration element is an aggregation of software selected and identified, processed as a single entity in the configuration management process [35]. A software configuration element is an intermediate or final product resulting from the application of the software development activities. Software configuration elements are clearly identified for adoption in the software configuration management process.

The arguments in favour of the adoption of the software configuration element concept as defined in international standards [7, 34] for use in materials management in experimentation are as follows:

- Software development process products are related to the experimental materials used in the experimentation processes.
- The experimental materials like instruments, forms, experimental objects, evolve over time due to the changes that take place at different levels of an experiment [22].

- The materials that are used to execute an experiment commonly form composition relations.

Therefore, the software configuration element concept is adopted for use in experimentation changing the name and definition accordingly: software configuration element is renamed experimental configuration element. The experimental configuration elements of an experiment are the clearly identified, reusable materials that require monitoring and control as part of the experimentation process. An example of an experimental configuration element is the `cmdline.doc` code shown in Figure 2, detailing its identification and versions.

A similar rationale applies for the other software configuration management concepts that are detailed in Table 2. Table 2 illustrates the concepts used in the software configuration management process and their relation to concepts adopted for use in ECM.

Table 2. ECM concepts

Software Development Cycle		Experimental Research Cycle	
Concept	Definition	Concept	Definition
Configuration	Set of related functional or physical components for outputting a specific version of a product	Configuration	Set of related Experimental Configuration Elements for outputting a specific version of an experiment
Configuration Element	An aggregation of hardware, software or both, which is selected and identified for the configuration management process and is processed as a single configuration management process entity	Configuration Element	Entity within a configuration that fulfils a function that can be clearly identified at a given reference point within the experimentation process
Software Configuration	Set of related SCEs like documents, programs, databases, tools, used to build a software system	Experiment Configuration	Set (collection) of experimental configuration elements instantiated in an experiment or at least a replication
Configuration Management	Discipline applying technical project control management to identify and document the functional and physical characteristics of a configuration element	Configuration Management	Control discipline whose aim is to maintain component integrity in order to evaluate and control the changes and facilitate product visibility
Software Configuration Management	Discipline for managing the changes in the SCEs built at different stages of the software development process and guaranteeing traceability, integrity, reliability and visibility	Experiment Configuration Management	Control discipline whose aim is to maintain experiment experimental configuration elements integrity to evaluate and control the changes and facilitate experimental configuration element visibility within the experimentation process

In order to **develop instruments for experiment configuration management**, the adopted material management concepts had to be provided. They are supported by instruments for recording the information on experiments, replications and experimental materials. We adapted the technical and administrative documents and developed a set of instruments based on the recommendations provided by software configuration management in order to carry out ECM activities. In the following, we describe the instrument that was built to enter the succession relation used to log experimental configuration element changes.

The schema of the developed instrument includes the version, type, replication, description, date and location fields. The version field should record a meaningful sequential integer. The type field specifies the experimental configuration element class (hardcopy, digital). The experiment field identifies the experiment in which the respective version of the experimental configuration element was used. The description field is used to detail significant changes made to the experimental configuration element. The date field is used to record the date on which the change to the experimental configuration element was entered by the person responsible for the change. Finally, the location field is useful for ascertaining where to get a specific version of the experimental configuration element. There should be one instrument to record each experimental

configuration element adopted in the configuration management process. Figure 2 shows an instance of the experimental configuration element versions of the cmdline.doc code document. The instrument details the specific information on each version. The example that we use is version 02 of the experimental configuration element, which is a digital document used in Experiment III. This version was entered on 13 December 2002 and is saved in a directory called C:\Experimento\Testing\Experimento III Ingles\Material Por Fases \Diseño\Particiones de Equivalencia \Objetos\cmdline. A similar rationale was applied to build the other instruments detailed in Table 3 showing the concepts and instruments built to support the ECM.

No. 04	Fields	Data			
Experimental configuration element number or code		OPE-CMD-05			
Experimental configuration element name		cmdline.doc code			
Experimental configuration element description		Source code to be printed and handed out to experimental subjects			
Baseline of which it is part		Operationalization phase			
Experimental configuration element author/s		To be defined			
Date created		30/11/2000			
Replication identification		Replication comparing testing techniques			
Experimental configuration element type (document, spreadsheet, program)		Document (Word)			
Location		C:\Replication\Testing\Replication I\Material Por Fases\Diseño\Cobertura de Sentencias\Objetos\cmdline			
Version number		01,02			
Version date		13/12/2002			
Name of Experimental Configuration Element : cmdline.doc code					
Code : OPE-CMD-05					
Research Group : GRISE					
Date : 27/10/2011					
Ver.	Type	Rep.	Description	Date	Location
01	Digital (Doc)	I, II	Source code to be printed by experimenters	30/11/2000	C:\Experimento\Testing\Experimento I\Material Por Fases \Diseño\ Cobertura de Sentencias \Objetos \cmdline
02	Digital (Doc)	III	Source code to be printed by experimenters	13/12/2002	C:\Experimento\Testing\Experimento III\Material Por Fases\ Diseño\Cobertura de Sentencias\ Objeto\cmdline

Figure 2. cmdline.doc code experimental configuration element version instrument

Table 3. List of concepts and instruments for ECM

Concept	Instrument
Experiment configuration element	Instrument for identifying the experimental configuration element
Baseline, relation	Instruments for identifying the baseline Instruments for identifying composition relations between experimental configuration elements Instruments for identifying derivation relations between experimental configuration elements
Version, release and instance, revision, relation	Instruments for identifying succession relations between experimental configuration elements (experiment, replications and experimental materials)
Library	Instruments for identifying equivalence relations between experimental configuration elements

In the **adopt experiment configuration management plan concepts and instruments** activity, the concepts adopted and instruments developed in the above activities were added to the different sections of the software configuration management proposed by IEEE 828 [7]. Finally, the aim of the **instantiation of the ECMP for an experiment** was to generate an instance of the ECMP based on the available information about the experiment. The result of this activity was the code inspection technique comparison experiment ECMP.

5 Proposed Framework for Managing Experimental Materials

The experimental materials management framework is composed of a procedure to help instantiate the experiment configuration management plan. The procedure for instantiating the ECMP is applicable in two scenarios:

- Scenario 1: a new experiment for which different materials are to be designed and built. In this scenario, the information and materials are entered and stored in the ECMP as they are released for use in the experimentation process.
- Scenario 2: an existing experiment for which the existing information and experimental materials should be recorded and migrated to the ECMP.

The activities within this procedure are detailed below:

- Analyse the physical or digital storage structure to find out how the materials supplied by the researchers are organized. In particular, we are looking for a common directory and subdirectory creation pattern.
- Study the documents constituting the experimental materials used in the experiment executions. In particular, the aim is to provide evidence of whether files with a similar name and type stored in different directories were used to execute replications of an experiment at the same or different points in time.
- Generate the uninstantiated experiment configuration management plan.
- Instantiate the experiment configuration management plan by entering the information about experiments, replications and materials specified in the different experiment configuration management plan sections and instruments.
- Create the directory structure proposed in the experiment configuration management plan in a local or remote storage device (hard drive).
- Migrate the digital archives (information describing the experiment, experimental materials) to the new directory structure.
- Implement a version control software tool, such as Git, Rational ClearCase, Subversion, TortoiseSVN, on the directory structure created in the experiment configuration management plan. These tools merely manage the status and traceability of the files as of the date on which they are entered into the automated configuration management process. Accordingly, the above activities are perfectly applicable when the procedure is implemented on the directory structure of a new experiment. In this case, the tool manages the file versions as changes are made. The directory structure of an existing experiment contains the migrated files (experimental objects, experimental materials, instruments) with similar content that have different names and dates of creation and/or modification, such as, for example, the change caused by the translation of an experimental material into another language (Spanish to English). In this case, the tool manages the file versioning independently without considering its existing relationship. In both scenarios, new or existing experiment, the instantiated experiment configuration management plan provides the researchers with support for experimental materials management, which is rounded out with the features offered by the automated software configuration management tool.

Table 4 shows the activities to be performed to instantiate the framework for a new or existing experiment. Note that activities whose performance is mandatory in each case are marked with Yes.

Table 4. Experiment Configuration Management Plan instantiation activities

No.	Activity	New Experiment	Existing Experiment
1	Analyse the physical or digital storage structure	No	Yes
2	Study the documentation on the experimental materials	No	Yes
3	Generate the Experiment Configuration Management Plan	Yes	Yes
4	Instantiate the Experiment Configuration Management Plan	Yes	Yes
5	Create the directory structure	Yes	Yes
6	Migrate the experimental materials	No	Yes
7	Implement the software configuration management tool	Yes	Yes

The experiment configuration management plan is composed of five sections. They establish the guidelines and provide the instruments for applying materials management. The experiment configuration management plan sections are scope, definitions, acronyms and abbreviations, references, organization and responsibilities, and experiment configuration management plan implementation. In the following, we detail each section.

5.1 Scope

This section should describe the reasons why it is necessary to implement experiment materials management. First, this section should detail that the application of the plan aims to generate an instance of the ECMP based on the existing information about the materials and replications. Second, it should specify that the instance will be used to ascertain the status and traceability of the experimental configuration elements for the purpose of their location, retrieval and use in replication processes. Third, it needs to state the primary sources for gathering information. Finally, it must identify the beneficiaries of plan instantiation.

5.2 Definitions, Acronyms and Abbreviations

This section includes the definitions of software configuration management terms, acronyms and abbreviations adopted for experimentation.

5.3 References

This section should include the list of documents that are referenced in the ECMP.

5.4 Organization and Responsibilities

The names of the researchers responsible for performing the ECM activities detailed in Table 5 should be entered in this section.

Table 5. ECM roles and responsibilities

Member/Role	Responsibilities	Surname and Name
ECM Manager	Role responsible for planning, designing, managing and coordinating the experiment ECM activities.	
Experiment Manager	Role responsible for supervising the research team using the library (repository) during the experimental research life cycle. Additionally, this role will liaise with the ECM manager to create and set up versioning.	
Research Team Member	This role interacts with the repository performing operations on the ECM generated during the project and is the principal producer/consumer of the information under version control.	

5.5 Implementation of the Experiment Configuration Management Plan

In this section, the information about the experiments, replications and materials should be entered in the respective instruments. The activities should be performed in the following order:

- Select experimental configuration elements
- Identify and enter baselines
- Define repository structure
- Enter experimental configuration elements composition and derivation relations
- Enter experiment and replication succession relations
- Identify and enter experimental configuration elements and their versions.

5.5.1 Select Experimental Configuration Elements

This section should list the experimental configuration elements to be controlled by ECM. To do this, researchers should consider:

- Experimental configuration elements that depend on or derive from other experimental configuration elements.
- Experimental configuration elements that are built based on the combination of a set of contextual variables.

- Experimental configuration elements that can change over time due to the changes with respect to the baseline experiment introduced in replications.

5.5.2 Identify and Enter Baselines

This section should record the milestones or checkpoints for controlling when a set of experimental configuration elements that are part of the baseline are approved. Additionally, the above milestones or checkpoints can mark the transition between one activity and another. The names of the experimentation activities —definition of experiment objectives, experiment design, experiment execution and analysis of results— must be assigned as baselines. This constitutes the roadmap for researchers to identify the materials used to execute the experiment and its replications. Figure 3 shows the instrument.

This instrument is composed of a header and body. The header should contain the information describing the baseline experiment. The body should account for the new and existing experiment scenarios with respect to information entry. The information that should be entered for both scenarios is code, name of the baseline experiment and the experimental configuration elements making up the baseline experiment. For a new experiment, the information recorded in the release date and replication name fields is entered after the material has been released for use in experimentation or replication processes. For an existing experiment, on the other hand, the information is gathered by analysing the directories storing the experiment and replication information and materials. To be precise, the file name and date of creation and/or modification is identified and recorded in the body of the instrument.

Name of the Organization					
Name of the Unit or Department					
Name of the Experiment		: _____			
Research Group		: _____			
Date		: _____			
No.		: _____			
Code	Baseline Name	Baseline Experimental Configuration Elements	Entry Date	Experiment Replications	
				Replication Name	Replication Name

Figure 3. Instrument for identifying baselines

5.5.3 Define Repository Structure

This section establishes the repository structure to store the different experimental materials and results developed in the experimentation activities. This should be implemented in local or remote storage devices.

5.5.4 Enter Experimental Configuration Elements Composition and Derivation Relations

This section should first record the composition relation, which describes that an experimental configuration element is composed of two or more experimental configuration element, whereas derivation describes the relation where one experimental configuration element originates from another experimental configuration element. Any experiment configuration management plan must necessarily include these relations because they are used to: 1) check whether all the experimental configuration elements of which a specific experimental configuration element is composed have been created, and 2) establish the dependency relation between experimental configuration elements.

The composition relation (Figure 4) and the derivation relation (Figure 5) instruments have a header and body. The body contains fields where the information on the experimental configuration elements should be entered. The header fields of both instruments should contain unique experiment identifier information. The experimental configuration elements composed of more than one experimental configuration element should be entered one by one in the fields of the composition instrument body, whereas the experimental configuration elements on which other experimental configuration elements are based should be entered in the body of the derivation relations instrument. Finally, the experimental configuration element type (document, graph, program, form, instrument, etc.) should be entered in both instruments.

Name of the Organization		
Name of the Unit or Department		
Name of the Experiment	:	_____
Research Group	:	_____
Date	:	_____
No.	:	_____
Experimental Configuration Element Name	Experimental Configuration Element Component Name	Type

Figure 4. Instrument for identifying composition relations

Name of the Organization		
Name of the Unit or Department		
Name of the Experiment	:	_____
Research Group	:	_____
Date	:	_____
No.	:	_____
Source Experimental Configuration Element Name	Derived Experimental Configuration Element Name	Type

Figure 5. Instrument for identifying derivation relations

5.5.5 Enter Experiment and Replication Succession Relations

The instruments in this section should record the information on the experiments and replications used to ascertain the evolution of the experiment and the status of planned and/or executed replications. It is mandatory to enter the date in the header and the information for identifying and locating the experiments executed by a research group in the body of the experiment succession relations instrument (Figure 6). On the other hand, the header of the replication succession instrument (Figure 7) details the information on the baseline experiment, and the fields of the body describe each executed replication of an experiment.

Name of the Organization							
Name of the Unit or Department							
Research Group							
Date : _____							
No.	Name	Code	Domain/Subject Area	Description	Date	Experimenters	Location
1			Requirements				
2			Tests				
3							

Figure 6. Instrument for the experiment succession relations

Name of the Organization						
Name of the Unit or Department						
Name of the Experiment	:	_____				
Research Group	:	_____				
Date	:	_____				
No.	:	_____				
No.	Replication Name	Type	Date	Venue	Experimenters	Location
1						
2						

Figure 7. Instrument for the replication succession relations

identified from the ECMP and also measures whether the information on the experimental materials can be located by searching the ECMP instruments.

The framework was evaluated internally and externally. The internal evaluation was run by the researchers that developed the framework, and it assessed the feasibility of instantiating the ECMP based on the experiment used to generate the framework. On the other hand, the external evaluation was conducted to ensure that the proposal was not only tailored to the specific needs of this experiment but also flexible enough to account for other cases. Therefore, the external evaluation evaluated feasibility, usability and usefulness. It was conducted by two external researchers who instantiated the ECMP for the personality quasi-experiment and then took three questionnaires. The first was the SUS (system usability scale) questionnaire [42] (see Appendix B), which measures two usability attributes: framework ease of use and ease of understanding. The ease of use attribute assesses the use of the ECMP content for the purpose of instantiation and the ease of understanding attribute assesses how easy it is to understand the ECMP content and instruments. The second was an ad hoc questionnaire designed to gather information on the current status of the administration of the experimental materials and replications of the experiment. There are two versions: one for baseline experimenters and one for experiment replicators. Both versions of this questionnaire are shown in Appendix C. The third questionnaire was designed and administered to establish the usefulness of the instantiated ECMP for administering the experimental materials after the surveyed researchers had run the test cases (see Appendix D).

With regard to the internal evaluation measuring feasibility, we used the instance generation, instantiation workload and acceptance attributes. Each attribute was evaluated after the ECMP instantiation process. The possible binary values of the evaluation of the instance generation attribute are: instantiated and not instantiated. Instantiation workload is the attribute for evaluating the resources required to instantiate the ECMP. This attribute is evaluated by counting the number of hours of work spent on instantiating the ECMP. For this purpose, the researcher is asked to update this information periodically. This procedure minimizes the risk of information loss over prolonged periods. Finally, acceptance is the attribute that measures whether the ECMP instances output are accepted by the experimenters. This attribute value is binary: accepted and not accepted. Additionally, the original authors of the experiment evaluated this attribute. They analysed whether the status and traceability of the experimental material could be identified from the information recorded in the different ECMP instruments and a value could be assigned to the attribute.

With regard to the external evaluation, feasibility was assessed by instantiating the personality quasi-experiment plan, considering the same attributes as described above for the internal evaluation. To measure usability (ease of use and ease of understanding), the external researchers are asked to use the framework to instantiate the quasi-experiment configuration management plan and then to complete the SUS questionnaire shown in Appendix B. The SUS questionnaire contains 10 items and is used in usability studies in research and the software industry. Questionnaire items 1, 2, 3, 5, 6, 7, 8 and 9 are used to measure the ease of use attribute, whereas questions 4 and 10 are used to evaluate the ease of understanding attribute. The questionnaire information is processed by automatically calculating the score applied by SUS [42]. According to the total score, the usability values are acceptable, marginal and unacceptable. The acceptable value is assigned when the calculation is greater than 70. The marginal value is assigned when the result of the calculation is from 50 to 70, whereas the value of unacceptable is assigned when the calculation is less than 50. Finally, to evaluate usefulness, researchers are required to complete two ad hoc questionnaires and run test cases to check that the information is useful for identifying the status and traceability of the materials and helpful in the materials search and retrieval processes. In particular, we used two ad hoc questionnaires to measure usefulness. The first questionnaire determined the current state of the administration of the experiment materials and replications by both the experimenters and replicators (Appendix C). The second questionnaire tested the usefulness of the framework on a Likert scale (1=totally disagree, 2=disagree, 3=neither disagree nor agree, 4=agree and 5=totally agree) after performing a set of case studies (Appendix D). According to the mean responses, the possible levels are average, acceptable and satisfactory. Table 6 shows the characteristics, attributes and attributes levels for both the internal and external evaluations.

Table 6. Framework instantiation evaluation attributes and levels

Evaluation	Characteristic	Attribute	Levels
Internal External	Feasibility	ECMP instantiation	Achieved Not achieved
		Instantiation effort	Person-hours
		ECMP instance acceptance	Accepted Not accepted
External	Usability	Ease of use	Acceptable Marginal Unacceptable
		Ease of understanding	Acceptable Marginal Unacceptable
	Usefulness	Researcher satisfaction	Satisfactory Acceptable Average

7 Internal Evaluation: Instantiation of the Experiment Configuration Management Plan for the Code Inspection Technique Comparison Experiment

This section describes the activities that were carried out to evaluate feasibility by instantiating the code inspection technique comparison experiment ECMP. They were carried out according to the procedure established in the framework proposed as part of this material management research. Below, we detail each of the activities that were carried out by the researcher that instantiated the proposal.

In order to **analyse the physical or digital storage structure**, we gathered a set of files that are organized and saved in a directory structure (Figure 10) containing the materials of the experiment under study. As we navigated through the directory structure, we found that it is mostly composed of four levels. Level 1 is the highest level of the structure and contains the versions of the experiments, for example, Experiment I and Experiment II. We found that Level 2 is related to the year in which the experiment was executed, for example, Universidad Politécnica de Madrid (UPM) UPM 2001, UPM 2002. Level 3 shows that most directories contain subdirectories for the analysis phase, analysed techniques and experimental objects per technique. Finally, Level 4, at the bottom of the structure, includes files like documents describing the experiment, materials, experimental objects, raw data, experimental results, etc. They were used and generated to execute the experiments carried out from 2000 to 2012.

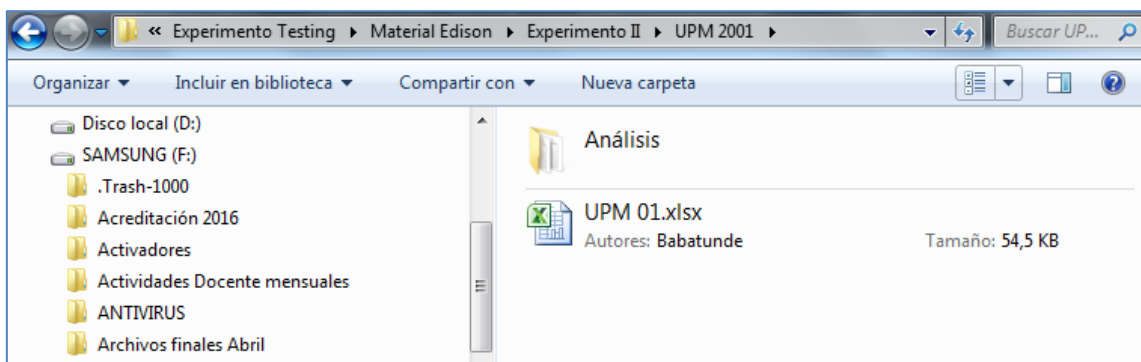


Figure 10. Experimental materials directory structure

To study the documentation on the information and materials, we navigated through the directory structure and carried out the following activities: 1) extract information (name, type and date of creation) of each of the files saved in the directory structure, and 2) analyse the information in the files. To extract the information, we navigated through the different levels of the directory structure to locate each of the files. The information extracted from each field was clustered by experiment and recorded in a set of tables. For example, Table 7 shows an excerpt containing Experiment I directory fields. Note that the Level 1, Level 2

and Level 3 fields refer to the directory names, whereas Level 4 contains the file names and types. Finally, the Date field records the file release date.

Table 7. Experiment I directory files

Level 1	Level 2	Level 3	Level 4	Date
Experiment I	First academic year 99-00	Analysis	Degree program data 99-00.xls	19/04/2005
		cmdline program	informacion cmdline.doc cmdline.c cmdline_c.doc	26/04/2000 20/03/2000 26/04/2000
		Nametbl program	informacion nametbl.doc nametbl.c nametbl_c.doc	26/04/2000 10/04/2000 10/04/2000

The **study the documentation on experimental materials** activity aims to identify whether or not the directory structure saves different versions of the experiment files. To be precise, in this activity, we retrieved and read the content, type and date of creation of each experiment file to identify different versions. The information located on the experiment is generally related to scientific articles, experiment planning, operational material, raw data and analysis, as well as the results of the questionnaires administered as part of the experiment execution. Additionally, the different field types include the following extensions: pdf (Portable Document Format), doc (Microsoft Word), xls (Microsoft Excel), txt (plain text), c, java, zip and rar. These files are used as a source of information for instantiating the ECMP.

The **Code Inspection Technique Comparison Experiment ECMP** was generated and instantiated according to the proposed framework guidelines. This instance contains information on the different instruments of the plan, which was migrated from the tables containing the experiment documents (Table 7). Finally, Table 8 shows the activities performed to instantiate the ECMP and the approximate instantiation workload which totalled about 104:38 person-hours.

Table 8. Code inspection technique comparison experiment ECMP Instantiation activity workload

Activity	Period	Approximate Workload
Analyse the storage structure	June-July 2011	32:00 person-hours
Study the documentation on the experimental materials		
Generate and instantiate the Experiment Configuration Management Plan	August-November 2011	58:38 person-hours
Implement tools	December 2011-January 2012	14:00 person-hours
Total	June 2011-January 2012	104:38 person-hours

In the following, we describe the sections of the experiment comparing code inspection techniques, the ECMP containing excerpts of information from three experiments, 15 replications and 102 experimental configuration elements entered in the instruments. The fully instantiated plan comparing the code inspection techniques is available at <https://gesproexp.herokuapp.com>.

7.1 Scope

The aim behind instantiating the experiment comparing code inspection techniques plan is to mitigate the chaos surrounding the management of experimental materials. Specifically, we aim to identify the status and traceability of the experimental configuration elements that are used in the experiment replication. The information to be entered in the plan instruments was gathered from two primary sources. First, we elicited information from experts in experiment replication. Second, we were involved in planning, executing and evaluating two experiments carried out in the years 2010 and 2012. The beneficiaries of plan instantiation will be the research group and colleagues interested in replicating the experiment.

7.2 Definitions, Acronyms and Abbreviations

Definitions:

Experimental Product. – Documents, materials used by the experimenters and experimental subjects in the experimental research cycle processes.

Experimental Configuration Element – Clearly identified reusable experimental products that require follow-up and control within the experimental research cycle.

Version. – Instance of an Experimental Configuration Element used as part of at least one configuration of one experiment or more than one replication.

Relation. - Connection or link between the Experimental Configuration Element in the experimental research cycle.

Baseline (Milestone). - Reference points in the experimental research cycle where Experimental Configuration Elements are formally approved. Required changes to the Experimental Configuration Elements are subject to formal processes.

Abbreviations:

ECMP: Experiment Configuration Management Plan

ECM: Experiment Configuration Management

References:

Standards ISO/IEC/IEEE 12207 2017 [29], IEEE [7].

Basics of Software Engineering Experimentation [37].

7.3 Organization and Responsibilities

The people responsible for performing the different roles in the ECMP are shown in Table 9.

Table 9. ECM roles

Member/Role	Responsibilities	Surname and Name
ECM Manager	Role responsible for planning, designing, managing and coordinating the experiment ECM activities.	Vegas, Sira
Librarian	Role responsible for supervising the research team using the library (repository) during the experimental research life cycle. Additionally, this role will liaise with the ECM manager to create and set up versioning.	Espinosa, Edison
Research team member	This role interacts with the repository performing operations on the ECM generated during the project and is the principal producer/consumer of the information under version control.	Research group

7.4 Management Plan Implementation

7.4.1 Identify and Enter Baselines

Experimentation activities were taken into account to establish the ECMP baselines. Figure 11 shows an excerpt from the information on the baselines defined in the experiment comparing code inspection techniques ECMP.

Universidad Politécnica de Madrid

Department of Computer Languages and Systems and Software Engineering

Experiment Name : Code Inspection Technique Comparison Experiment

Research Group : GRISE

Date : 27/10/2011

Code	Name	Experimental Configuration Element	Experiment Replications	
LB01	Planning phase	Experiment domain selection document Experiment topic selection document	Replication name	Replication name
LB02	Operationalization phase	Experiment design template Experiment adaptation document Training document Software evaluation techniques presentation (for academic year 05/06, Dynamics-2, techniques)		

Figure 11. Code inspection technique comparison experiment baseline inventory instrument

7.4.2 Define Repository Structure

Figure 12 shows the repository directory structure that was implemented on the secondary storage device to save the different Experimental Configuration Elements adopted in the ECMP.

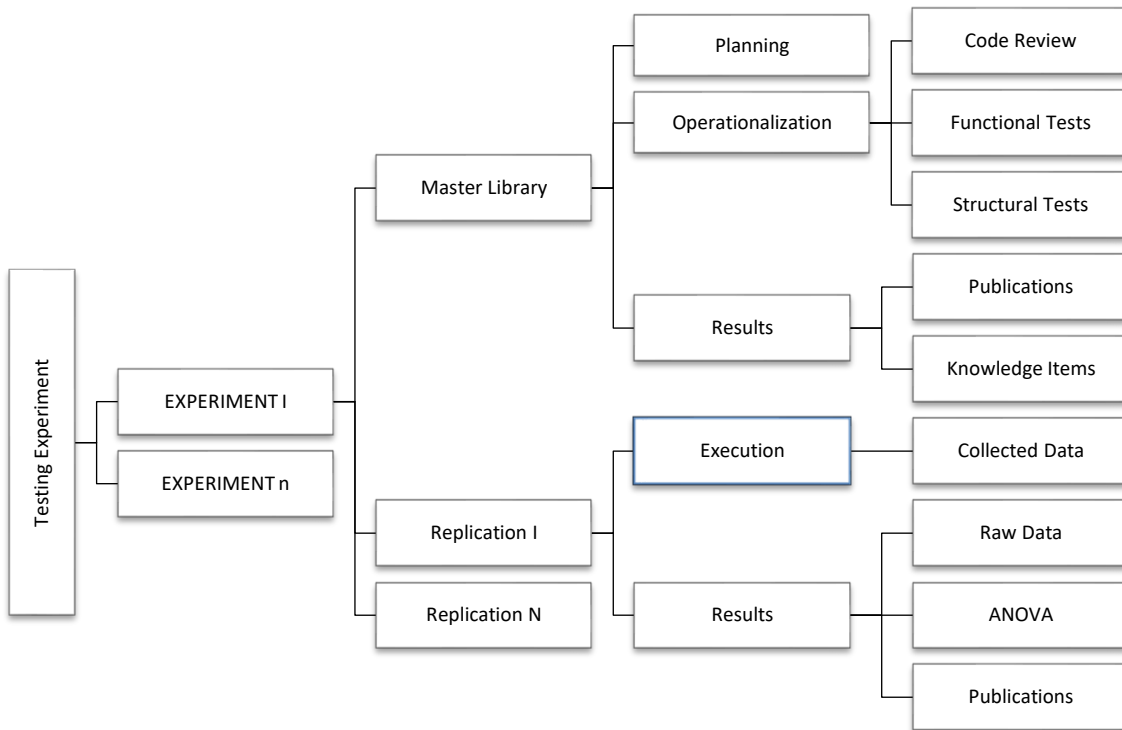


Figure 12. Code inspection technique comparison experiment repository directory structure

7.4.3 Enter Composition and Derivation Relations between Experimental Configuration Elements

The composition relations were entered based on the testing technique applied in the experiment comparing code inspection techniques. The testing technique is related to the materials, objects and instruments that are required for its execution. Figure 13 shows an excerpt of the information on the experiment comparing code inspection techniques ECMP composition relations.

Universidad Politécnica de Madrid
Department of Computer Languages and Systems and Software Engineering

Experiment Name : Code Inspection Technique Comparison Experiment
Research Group : GRISE
Date : 27/10/2011

Experimental Configuration Element Component	Experimental Configuration Element Components	Type
Code review	1.- Code review. Pre-questionnaire (Data collection form E00) Post-questionnaire (Data collection form E05) Code review abstractions form (E12) 2.- Inconsistency identification. Code specifications (Program specifications E01)	Doc Doc Doc Doc
Structural testing (Structural test forms file)	1.- Structural testing test case generation Instructions (Instructions for applying the structural testing technique E20). Supplementary sheet (Supplementary information on cmdline, nametbl, ntree E04) Pre-questionnaire (Data collection form E00) Post-questionnaire (Data collection form E05) Source code (bad cmdline code, bad nametbl code, bad ntree code)	Doc Doc Doc Doc Doc

Figure 13. Code inspection technique comparison experiment composition relations instrument

The derivation relations were established to identify the experimental configuration elements that are used to generate other experimental configuration elements. In Figure 14, we show an excerpt of the information entered about the derivation relations of the experiment comparing code inspection techniques ECMP.

Universidad Politécnica de Madrid
Department of Computer Languages and Systems and Software Engineering

Experiment Name : Code Inspection Technique Comparison Experiment
Research Group : GRISE
Date : 27/10/2011

Source Experimental Configuration Element Name	Derived Experimental Configuration Element Name	Type
Printed source code document (cmdline, nametbl, ntree) (structural technique)	Test case design (collected data)	Doc
Program specifications (functional technique)	Test case design (collected data)	Doc
Program specifications (code review). Printed source code document (cmdline, nametbl, ntree)	Program abstraction	Doc

Figure 14. Code inspection technique comparison experiment derivation relations instrument

7.4.4 Enter Experiment and Replication Succession Relations

We found three documented versions of experiments administered by the research group in the testing domain. Figure 15 shows an excerpt of the details for Experiment I comparing code inspection techniques.

Universidad Politécnica de Madrid

Department of Computer Languages and Systems and Software Engineering

Experiment Name : Code Inspection Technique Comparison Experiment

Research Group : GRISE

Date : 27/10/2011

No	Name	Code	Topic Domain	Description	Date	Experimenters	Location
1	Experiment I (Experiment comparing code inspection techniques)	001	Testing	The purpose of the experiment is to study whether there is any relationship between technique and fault in terms of effectiveness	1999-2000	Natalia Juristo Sira Vegas	

Figure 15. Code inspection technique comparison experiment version inventory instrument°

The number of internal and external experiment replications totalled 15 up to 2014, of which 11 corresponded to Experiment I and four to Experiment II. Figure 16 shows an excerpt of the information on three versions of Experiment I replications comparing code inspection techniques.

Universidad Politécnica de Madrid

Department of Computer Languages and Systems and Software Engineering

Experiment Name : Code Inspection Technique Comparison Experiment

Research Group : GRISE

Date : 27/10/2011

No. : 01

No.	Replication Name	Type	Date	Site	Experimenters	Location
01	UPM 2001	Internal, dependent, similar	2001	Universidad Politécnica de Madrid	Natalia Juristo Sira Vegas	
02	UPM 2002	Internal, dependent, similar	2002	Universidad Politécnica de Madrid	Natalia Juristo Sira Vegas	
03	UPM 2003	Internal, dependent, similar	2003	Universidad Politécnica de Madrid	Natalia Juristo Sira Vegas	

Figure 16. Code inspection technique comparison experiment replication version inventory instrument

7.4.5 Identify and Enter Experimental Configuration Elements and Versions

Information on 102 experimental configuration elements and their versions was entered in the code inspection technique experiment ECMP. Figure 17 shows the information entered in the cmdline experimental configuration element. This material is a document coded OPE.ICM.01, containing the specifications for the code inspection technique, functional tests and structural tests. It was authored by Baumgärtner, Claßen, Gieseke, Lott, released on 26 April 2000, is part of the code inspection technique comparison experiment, and there are two versions of this document. Figure 18 shows the information on the two versions of cmdline executed in the Universidad Politécnica de Madrid replications: UPM 2001 and UPM 2004. The first version is digital, was released on 26 April 2001 and is located at C:\Experimento\Testing\Experimento I\Material Por Fases \Diseño \Particiones de Equivalencia \Objetos \cmdline. The second version was executed in the UPM 2004 replication, released on 13 December 2003 and is located at C:\Experimento\Testing\Experimento III\Material Por Fases \Diseño\Revision deCodigo \Objetos\cmdline.

Universidad Politécnica de Madrid

Department of Computer Languages and Systems and Software Engineering

Experiment Name : Code Inspection Technique Comparison Experiment

Research Group : GRISE

Date : 27/10/2011

No 002	Fields	Data
Experimental Configuration Element code number		OPE.ICM.01
Experimental Configuration Element name		Informacion cmdline
Experimental Configuration Element description		Document containing: Code inspection technique specifications E02. Functional testing technique specifications E03. Structural testing technique specifications E04
Baseline of which it is part		Operationalization phase
Experimental Configuration Element author(s)		Baumgärtner, Claßen, Gieseke, Lott
Date of creation		26/04/2000
Identification of the replication project of which it is part		Experiment comparing code inspection techniques
Experimental Configuration Element type (document, spreadsheet, program)		Document (Word)
Location		C:\Experimento\Testing\Experimento I\Material Por Fases \Diseño \Particiones de Equivalencia \Objetos\cmdline
Version number		01.02
Version date		13/12/2003

Figure 17. Instrument identifying the informacion cmdline experimental configuration element

Experimental Configuration Element Name: informacion cmdline.doc

Code : 002

Research Group : GRISE

Date : 27/10/2011

Ver	Type	Replication	Description	Date	Location
01	Digital (Doc)	UPM 2001	Document containing: Program specifications E01 Code inspection technique specifications E02 Functional testing technique specifications E03 Specifications for submission together with code E03 part 2 Structural testing technique specifications E04	26/04/2001	C:\Experimento\Testing\Experimento I\Material Por Fases \Diseño \Particiones de Equivalencia \Objetos\cmdline
	Hardcopy				

Ver	Type	Replication	Description	Date	Location
02	Digital (Doc)	UOM2004	Document containing: Code inspection technique specifications E02 Functional testing technique specifications E03 Structural testing technique specifications E04	13/12/2003	C:\Experimento\Testing\Experimento III\Material Por Fases \Diseño\Revision de Codigo \Objetos\cmdline
	Hardcopy				

Figure 18. Informacion cmdline.doc experimental configuration element version inventory instrument

8 External Evaluation: Instantiation of the Personality Quasi-Experiment Configuration Management Plan

In this section, we describe the evaluation of the feasibility, usability and usefulness of the proposed framework. As already mentioned, feasibility assesses whether the ECMP can be instantiated. Usability measures the ease of use and understanding of the framework after instantiating the ECMP. Usefulness measures researcher satisfaction with the information recorded in the different ECMP sections and instruments, which they use to search and locate the experimental materials after identifying their status and traceability.

The feasibility evaluation was carried out by instantiating the personality quasi-experiment. This personality quasi-experiment studies the influence of different personality factors and teamwork issues on software production [8]. This is a complex quasi-experimental study including several replications, a great many materials and a large number of hypotheses, measurement instruments, and instructions for results analysis and interpretation, etc. The instantiation was carried out by the researchers responsible for the experiment at different time intervals. Different means of communication, such as face-to-face meetings, electronic mail and videoconference, were used for interaction at different times. The researchers performed several activities to instantiate the plan. The workload required to perform these activities was measured as the time taken to perform each activity. Activity time was accurately recorded in the instruments created for the purpose.

The iteration was composed of four primary milestones. They were: kick-off meeting, ECMP instantiation, final review of plan and software tool implementation meeting. In the following, we detail each of these milestones:

- The **kick-off meeting** was a face-to-face event used to coordinate the activities that were to be carried out to instantiate the experiment configuration management plan.
- The first step in **experiment configuration management plan instantiation** was to email the experiment configuration management plan to the researchers. They emailed the instantiated experiment configuration management plan back together with a series of questions related to the instantiation of the plan. Several of the questions were solved through an email exchange, whereas a face-to-face meeting had to be arranged to address other, conceptual issues.
- The **final review of the plan** identified whether the experimenters had entered all the required information in the different sections and instruments within the experiment configuration management plan.
- The **implementation meeting** was a face-to-face meeting and addressed the implementation of the proposed directory structure and Git tool.

Table 10 shows the schedule of the activities carried out by the experimenters together with the means of communication that was used and the approximate workload for each activity performed to instantiate the ECMP. The total workload for ECMP instantiation was approximately 23:03 person-hours.

Table 10. Personality quasi-experiment ECMP instantiation workload

Activity	Means of Communication	Date	Workload (person-hours)
Kick-off meeting	Face-to-face meeting	04/09/2013	1:30
Instantiate experiment configuration management plan (Version 1)	Email Face-to-face meeting Videoconference	08/10/2013	1:20
Review 1	Email Face-to-face meeting	08/01/2013	2:30
List problems	Face-to-face meeting	13/01/2014	3:08
Email experiment configuration management plan and solutions to problems	Email	13/01/2014	0:05
Meet to solve (conceptual) ECMP instantiation problems	Email Face-to-face meeting Videoconference	14/01/2012	1:05
Instantiate experiment configuration management plan (Version 2)	Email Face-to-face meeting Videoconference	17/01/2014	4:30
Review 2	Email Personal Videoconference	20/01/2014	1:20
Meet to generate directory structure	Face-to-face meeting	31/01/2014	1:10
Create directory structure	Face-to-face meeting	03/02/2014	2:10
Implement Git tool on the directory structure	Face-to-face meeting	04/02/2014	4:15
		Total workload	23:03

The ECMP was instantiated based on the information from the files saved in the directory structure. Table 10 shows that two reviews were conducted to instantiate the personality quasi-experiment. Review 1 of the ECMP identified problems that were reported by means of a document containing the list of errors identified in the plan. This list was drawn up during a face-to-face meeting between the group that designed the proposal and the researchers that instantiated the first version of the ECMP. The problems were divided into two classes. The first class was caused by the fact that the researchers did not know which information to enter in the baseline and release experiment configuration element fields of the baseline identification and the experiment and replication succession relation instruments. To be precise, they left some fields blank and entered incorrect data in others. The second class of problems were related to the fact that data for entry in the replication type, date of creation and location fields of the replication succession relations instrument were missing. These problems were solved through email queries, face-to-face meetings and videoconferences. Review 2 of the first ECMP version again identified empty fields due, in this case, to missing data. Finally, a face-to-face meeting was held to create the directory structure and implement the Git tool for automatic version management. Appendix A illustrates the instruments with excerpts of the personality quasi-experiment information for two replications and one experimental material, whereas the fully instantiated plan for all six replications and seven experimental materials is available at <https://gesproexp.herokuapp.com>.

To evaluate framework usability, the two external researchers that instantiated the personality quasi-experiment ECMP took the SUS questionnaire (Appendix B). The final usability result suggested that the framework was easy to use and reasonably easy to understand (72.5% for both attributes). Looking at the SUS questionnaire item ratings, we found that the ECMP was self-contained thanks to the details concerning concepts, sections and instruments. On the other hand, researchers noted that they required further information and support from the researchers that generated the proposal to enter the data especially in the baseline identification and experiment and replication succession instruments.

To evaluate usefulness, we applied a set of instruments composed of: 1) two ad hoc questionnaires, one for experimenters and another for replicators, to establish the status, traceability and location of the available material before using the framework; 2) test cases for evaluating the usefulness of the instantiated framework, and 3) a questionnaire for measuring usefulness after running the test cases. Using the ad hoc questionnaire for researchers and replicators (Appendix C), we were able to identify the constraints that researchers currently face with respect to the administration of experimental materials. To be precise, we found that information on materials with different contents was saved at different sites and in different formats. Additionally, it was incomplete, disperse and vague. Finally, it was impossible to ascertain status and traceability or identify changes. To apply the questionnaire, measuring the usefulness of the ECMP (Appendix D), it was necessary to carry out the activities detailed below:

- Design cases to evaluate the usefulness of the ECMP proposal (Appendix D).
- Instantiate the ECMP for the experiment.
- Implement the configuration management tool on the directory structure provided by the ECMP.
- Execute the cases designed to evaluate the usefulness together with the researchers.
- Analyse the results of the executed cases together with the researchers.

After running the test cases in Appendix D, we found that ECMP provides a better organization of the personality quasi-experiment experimental materials, and the usefulness questionnaire scores are acceptable (four points on the Likert scale), as it was indeed possible to identify the status, traceability and the location of the baseline quasi-experiment, its replications and the respective experimental material.

Table 11 shows the values for the different attributes of the personality quasi-experiment ECMP characteristics of feasibility, usability and usefulness.

Table 11. Attribute assessment for the external evaluation of the ECMP.

Characteristic	Attribute	Value
Feasibility	ECMP instantiation	Instantiated
	Instantiation effort	23.03 person-hours
	ECMP instance acceptance	Accepted
Usability	Ease of use	Accepted (72.5%)
	Ease of understanding	Accepted (72.5%)
Usefulness	Researcher satisfaction	Accepted (4)

9 Discussion of Results

The instances of the code inspection technique comparison experiment and personality quasi-experiment experiment configuration management plan released for verification were inspected according the evaluation process detailed in Section 6. This process validated the feasibility, usability and usefulness of the framework. This evaluation was carried out by the code inspection technique experiment and personality quasi-experiment administrators together with the researchers that created the framework. Besides, the instances were validated independently and at different time intervals. The final version of the code inspection technique experiment ECMP that was submitted for evaluation was instantiated over a nine-month period during which knowledge elicitation meetings were held, the ECMP was inspected and the code inspection technique experiment plan was instantiated. Over this time period, three versions of the ECM document, two versions of the instruments and two versions of the code inspection technique experiment ECMP were created. They were the result of changes suggested by research group members. Note that the framework was developed in parallel to the instantiation of the code inspection technique experiment ECMP applying action research by means of which its feasibility was evaluated internally, whereas the personality quasi-experiment ECMP was used to externally evaluate feasibility, usability and usefulness. The personality quasi-experiment ECMP was released for inspection within a month, during which two versions of the personality quasi-experiment plan were generated. They were the result of support provided by the framework creators to the personality quasi-experiment research group with respect to conceptual issues regarding the plan.

With regard to the evaluation of the feasibility, instances were generated for both the experiment comparing code inspection techniques and the personality quasi-experiment. The inspection consisted of identifying whether all the information on the code inspection technique experiment and personality quasi-experiment

was entered into the different ECMP sections and instruments. The instantiation workload measured in person-hours was 104:38 and 23:00, respectively. These values summarize the total workload input by the researchers to instantiate each plan and was sourced from Table 8 and

Table 10, specifying the activities carried out and the times taken. The difference between the values assigned to this attribute is due to the fact that the experiment comparing code inspection techniques was used as a pilot study for adopting concepts and developing instruments to manage experimental materials. As already mentioned, the adoption process took nine months during which several versions of the ECM document and the ECMP instruments were output as a result of reviews conducted by research group members and the framework creator. Besides, the fact that the framework creator had to discover knowledge about the experiment also added to the total workload. The knowledge was discovered by means of planned elicitation meetings with the research group and participation in the execution of an experiment replication. In sum, we identified three versions of the experiment, which was replicated 15 times at different sites, whereas the instantiated code inspection technique experiment ECMP adopted 102 experimental configuration elements. On the other hand, the workload required to instantiate the personality quasi-experiment ECMP was smaller because meetings were held with the personality quasi-experiment team. Additionally, plan instantiation was coordinated via means of communication like email, videoconference, etc., which helped to solve conceptual ECMP questions. In sum, the personality quasi-experiment ECMP adopted information taken from six replications and seven experimental configuration elements. Finally, the value that was assigned to the acceptance attribute for both instances was accepted. This value is the result of the analysis of the information entered into the different plan instruments that was conducted by the research group members that conducted each experiment. Specifically, the *accepted* value is assigned if the materials management plan proves to be applicable. In both cases, it was possible to identify the status and traceability of the experiment, replications and materials. Additionally, the plan facilitated the search, location and retrieval of all the information and materials saved in the directory structure for use in the execution of new replications. Finally, as mentioned in Section 8 (External Evaluation), two review rounds were required to instantiate the personality quasi-experiment ECMP. These reviews revealed that the fields of some instruments were either left empty or contained errors because researchers did not know which information to enter and data were missing. These problems were solved by means of email queries, face-to-face meetings and videoconferences. For future ECMP instantiation processes, we believe that it would be helpful to schedule and hold meetings between the ECMP instantiation stakeholders and the researchers that created the framework. Additionally, the stakeholders should examine the information described in Section 5 (Proposed Framework for Managing Experimental Materials) detailing the concepts and the use of the ECMP instantiation instruments. This information is also available at <https://gesproexp.herokuapp.com>.

The code inspection comparison experiment was not evaluated for usability and usefulness because the framework developers instantiated the ECMP. It was, however, possible to evaluate these characteristics for the quasi-experiment because the ECMP was instantiated by external researchers. At 72.5%, the rating for usability was *acceptable*. Looking at the the SUS questionnaire item ratings, we found that the details on the ECMP concepts, sections and instruments were helpful for instantiating the plan. On the other hand, the researchers stated that information and support from framework developers was required to improve the plan instantiation process. As regards the usefulness characteristic, a value of four on the Likert scale denotes that usefulness was *acceptable*. This value was assigned by the external researchers after executing the test cases and finding that the material was better organized than it was originally, as the plan identified the status, traceability, and location of the personality quasi-experiment, its replications and experimental material.

Finally, the framework was developed using information and materials in the more taxing context of the code inspection technique comparison experiment. This experiment provided the researchers with information on experiments, replications and materials executed over more than a decade that had to be saved in a field containing files. In this field, we found that the contents of the information on the experiment and material available at different sites (United States, Spain, Uruguay, Ecuador, etc.) varied and was stored in different formats. Besides, the information was incomplete, disperse and vague. The personality quasi-experiment provided a better context because researchers had access to semi-organized information and materials. Additionally, it was possible in both ECMP instances to check experiment evolution and discover the status and traceability of replications and materials as opposed to the original materials management chaos.

10 Conclusions

The research problem addressed in this paper targets the chaos surrounding the administration of experimental materials in software engineering experimentation. Different authors describe the importance of

managing experimental materials for experiment replication in software engineering [4, 17, 20]. From the literature analysis, we found that papers pinpoint similar shortcomings with respect to materials version control and experiment replications.

To solve the problem of experimental material and experiment replication administration in software engineering, this paper details how we applied the action research method to carry out activities in order to iteratively adopt the software engineering paradigm for application in experimental software engineering. For adoption, we carried out the following activities: 1) adopt software engineering concepts for experimentation, 2) develop instruments for configuration management, 3) adopt concepts and instruments in the ECMP, and 4) instantiate the software engineering experiment plan.

The concept adoption activity generated a software engineering concept analogy for application in materials management. This analogy was based on the study of the software development process, software engineering and the experimental process as a transformation of products.

The develop instruments activity built instruments to support experimental material change control in experimental software engineering. In fact, the instruments developed include:

- Instrument identifying the experimental configuration element and baselines that are used to clearly identify the experimental configuration elements and the software engineering experiment baselines.
- Version control instrument (succession relations between experimental configuration elements) for experimental configuration elements, experiments and replications, which are used to ascertain the traceability of the experimental configuration elements, experiments and replications in software engineering experiment.
- Instrument for identifying composition relations between experimental configuration elements, which are used to identify that an experimental configuration element is composed of more than one experimental configuration element in software engineering experiment.
- Instrument for identifying derivation relations between experimental configuration elements, which are used to identify that an experimental configuration element is an originator of more than one experimental configuration element in the experiments in software engineering.
- Instrument for identifying equivalence relations between experimental configuration elements, which is used to identify experimental configuration element types that correspond to one and the same experimental configuration element in experimentation.
- Library structure for saving and retrieving the experimental configuration element adopted in the experimental materials management process in experimental software engineering.

The ECMP was output by entering the concepts and instruments. This ECMP was instantiated for the code inspection comparison techniques experiment and a personality quasi-experiment. These instances were evaluated by researchers of each experiment and the framework developers. The evaluation aimed to identify the feasibility, usability and usefulness. To do this, the instantiated ECMPs were evaluated internally and externally. The internal evaluation validated feasibility and showed that the ECMP can be instantiated for the experiment comparing code inspection techniques. On the other hand, the external evaluation used the personality quasi-experiment to validate feasibility, as well as usability and usefulness. Usability confirmed that the quasi-experiment ECMP could be instantiated based on the use of the framework, and usefulness revealed that the information and materials were better organized as the framework identified the status and traceability of quasi-experiment, replications and materials. Further ECMP instantiations in the field of software engineering experimentation are required to confirm the results of this validation. Finally, we found that, in both cases, the storage, search, location and retrieval of the materials for use in experiment replications in different contexts is improved. Failure to manage the materials usually leads to a wasteful expenditure of the time, money and resources required to execute new replications of experiments.

Acknowledgements

This research was funded by a grant from the Universidad de las Fuerzas Armadas ESPE, National Secretariat of Higher Education, Science, Technology and Innovation of the Ecuadorian Government (SENESCYT) and partly funded by Spanish Ministry of Economy and Competitiveness TIN2014-52129-R and TIN2014-60490-P projects and the R&D program of the Madrid Region (P2018/TCS-4314).

References

1. V. R. Basili, F. Shull and F. Lanubile, Building knowledge through families of experiments, IEEE Transactions on Software Engineering, 1999, vol. 25, no 4, pp. 456-473.

2. J. W. Daly, A. Brooks, J. Miller, M. Roper and M. Wood, verification of results in software maintenance through external replication. In Proceedings of the IEEE International Conference on Software Maintenance, 1994, pp. 50-57.
3. S. Vegas, N. Juristo, A. Moreno, M. Solari and P. Letelier, Analysis of the influence of communication between researchers on experiment replication. In Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, 2006, pp. 28-37.
4. F. Shull, M. G. Mendonça, V. Basili, J. Carver, J. C. Maldonado, S. Fabbri, G. H. Travassos and M. C. Ferreira, Knowledge-sharing issues in experimental software engineering, Empirical Software Engineering, 2004, vol. 9, no 1, pp. 111-137.
5. N. Juristo, S. Vegas, M. Solari, S. Abrahao and I. Ramos, Comparing the effectiveness of equivalence partitioning, branch testing and code reading by stepwise abstraction applied by subjects. In Proceedings of the fifth international conference on software testing, verification and validation, 2012, pp. 330-339.
6. K. Lewin, Action research and minority problems, Journal of Social Issues, 1946, vol. 2, no 4, pp. 34-46.
7. T. View. IEEE Standard for Software Configuration Management Plans. IEEE Std. 828-2005 (Revision of IEEE Std. 828-1998), 2005, pp. 1-19.
8. S. T. Acuña, M. Gómez and N. Juristo, Towards understanding the relationship between team climate and software quality-A quasi-experimental study, Empirical Software Engineering, 2008, vol.13, no 4, pp. 401-434.
9. N. W. Paton, Managing and sharing experimental data: Standards, tools and pitfalls, Biochem. Soc. Trans., 2008, vol. 36, pp. 33-36.
10. J. Kaiser, Rigorous replication effort succeeds for just two of five cancer papers, Science, January 2017, vol. 18, doi: 10.1126/science.aal0628, Retrieved from <http://www.sciencemag.org/news/2017/01/rigorous-replication-effort-succeeds-just-two-five-cancer-papers> [Accessed 04/01/2019].
11. M. Baker, 1,500 scientists lift the lid on reproducibility. Nature News, 2016, vol. 533, no 7604, pp. 452.
12. E. Espinosa Gallardo, Using configuration management and product line software paradigms to support the experimentation process in software engineering. In Proceedings of the Research Challenges in Information Science, 2012, pp.1-6.
13. E. Kamsties and C. M. Lott, An empirical evaluation of three defect-detection techniques, In Proceedings of the 5th European Software Engineering Conference, 1995, pp. 362-383.
14. A. R. Jones and A. L. Lister, Managing experimental data using FuGE, Proteome Bioinformatics, 2010, pp. 333-343.
15. V. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard and M. Zelkowitz, Packaging researcher experience to assist replication of experiments. In proceedings of the ISERN Meeting, 1996, pp. 3-6.
16. R. Conradi, V. R. Basili, J. Carver, F. Shull and G. H. Travassos, Pragmatic Documents Standard for an Experience Library: Roles, Documents, Contents and Structure Computer Science Department Technical Report: University of Maryland, 2001.
17. F. Shull, V. Basili, J. Carver, J. C. Maldonado, G. H. Travassos, M. Mendonça and S. Fabbri, Replicating software engineering experiments: Addressing the tacit knowledge problem. In Proceedings of the international symposium empirical software engineering, 2002. pp. 7-16.
18. E. Arisholm, D. I. Sjøberg, G. J. Carelius and Y. Lindsjörn, SESE an experiment support environment for evaluating software engineering technologies: In Proceedings of the tenth Nordic workshop on programming and software development tools and techniques, 2002, pp. 81-98.
19. V. P. Lopes and G. H. Travassos, Knowledge repository structure of an experimental software engineering environment. In Proceedings of the XXIII Brazilian symposium on software engineering, 2009, pp. 32-42.
20. H. Do, S. Elbaum and G. Rothermel, Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact, Empirical Software Engineering, 2005, vol. 10, no 4, pp. 405-435.
21. M. Mendonça, J. Maldonado, M. de Oliveira, J. Carver, S. Fabbri, F. Shull, G. Travassos, E. Höhn, and V. Basili, A framework for software engineering experimental replications. In Proceedings of the 13th IEEE international conference on engineering of complex computer systems, 2008, pp. 203-212.
22. O. S. Gómez, N. Juristo and S. Vegas, Understanding replication of experiments in software engineering: A classification, Information and Software Technology, 2014, vol. 56, no 8, pp. 1033-1048.
23. R. M. Lindsay and A. S. C. Ehrenberg, The design of replicated studies, The American Statistician, 1993, vol. 47, no 3, pp. 217-228.

24. E. Espinosa, Configuration Management and Product Lines for Improving the SE Experimentation Process, 2014, PhD Thesis. Universidad Politécnica de Madrid. Escuela Técnica Superior de Ingenieros Informáticos.
25. E. Espinosa, J. M. Ferreira and H. Chanatasig, Using experimental material management tools in experimental replication: A systematic mapping study. In Proceedings of the International Conference on Information Theoretic Security. Springer, Cham, 2018. pp. 252-263.
26. M. Solari, S. Vegas, N. Juristo, Content and structure of laboratory packages for software engineering experiments. Information and Software Technology, 2018, vol. 97, p. 64-79.
27. V. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sørungård and M. Zelkowitz, Lab package for the empirical investigation of perspective-based reading, Available at <http://www.cs.umd.edu/projects/SoftEng/ESEG>. Accessed 2017, 1998.
28. V. Basili, F. Shull and F. Lanubile, Using experiments to build a body of knowledge. In Proceedings of the International Andrei Ershov Memorial Conference on Perspectives of System Informatics Perspectives of System Informatics, 1999, pp. 265-282.
29. J. Meyer, Using qualitative methods in health-related action research, British Medical Journal, 2000, vol. 320, no 7228, pp. 178-181.
30. R. McTaggart, Principles for participatory action research, Adult Education Quarterly, 1991, 41, pp. 168-187.
31. E. Koshy, V. Koshy and H. Waterman. Action Research in Healthcare. London: Sage, 2010.
32. E. Bersoff, V.D. Henderson and S. G. Siegel, Software Configuration Management. An investment in product integrity. Englewood Cliffs. Nueva Jersey: Prentice Hall, 1980.
33. W. A. Babich, Software Configuration Management, Coordination for Team Productivity. New York: Addison Wesley, 1986.
34. ISO/IEC/IEEE Standard for Systems and Software Engineering - Software Life Cycle Processes, IEEE Std. 12207-2017, 2017.
35. IEEE, Standard glossary of software engineering terminology. IEEE Standard 610.12-1990, 1990.
36. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, Experimentation in Software Engineering. Berlin Heidelberg: Springer Verlag, 2012.
37. N. Juristo and A. M. Moreno, Basics of Software Engineering Experimentation. New York: Springer Science & Business Media, 2013.
38. A. Stefik, S. Siebert, S. Meliss and K. Slatter, An empirical comparison of the accuracy rates of novices using the quorum, perl, and random programming languages. In Proceedings of the 3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools, 2011, pp. 3-8.
39. A. Porter and L. Votta, Comparing detection methods for software requirements inspections: A replication using professional subjects, Empirical Software Engineering, 1998, vol. 3, no 4, pp. 355-379.
40. M. B. Chrissis, M. Konrad and S. Shrum, CMMI Guidelines for Process Integration and Product Improvement. New York: Addison Wesley, 2003.
41. C. Apa and O. Dieste, Effectiveness for detecting faults within and outside the scope of testing techniques: An independent replication, Empirical Software Engineering, 2014, vol. 19, no 2, pp. 378-417.
42. Lewis, James R., and Jeff Sauro, The factor structure of the system usability scale, International Conference on Human Centered Design. Springer, Berlin, Heidelberg, 2009, pp. 94-103.

Appendices

Appendix A:

Instantiation of the Personality Quasi-Experiment Configuration Management Plan

A.1 Scope

The aim of instantiating the personality quasi-experiment plan is to mitigate the chaos surrounding the management of experimental materials. Specifically, we aim to identify the status and traceability of the experimental configuration elements that are used in the quasi-experiment replication. The information entered in the plan instruments was elicited from the personality quasi-experiment principal researcher. The beneficiaries of plan instantiation will be the research group and colleagues interested in replicating the experiment.

A.2 Definitions, Acronyms and Abbreviations

Definitions:

Experimental Product. – Documents, materials used by the experimenters and experimental subjects in the experimental research cycle processes.

Experimental Configuration Element – Clearly identified reusable experimental products that require follow-up and control within the experimental research cycle.

Version. – Instance of an Experimental Configuration Element used as part of at least one configuration of one experiment or more than one replication.

Relation. – Connection or link between the Experimental Configuration Element in the experimental research cycle.

Baseline (Milestone). – Reference points in the experimental research cycle where Experimental Configuration Elements are formally approved. Required changes to the Experimental Configuration Elements are subject to formal processes.

Abbreviations:

ECMP: Experiment Configuration Management Plan

ECM: Experiment Configuration Management

References:

Standards ISO/IEC/IEEE 12207 2017 [30], IEEE [7].

Basics of Software Engineering Experimentation [37].

A.3 Organization and Responsibilities

Table 12 details the people responsible for carrying out the different roles in ECMP.

Table 12. ECM roles

Member/Role	Responsibilities	Surname and Name
ECM Manager	Role responsible for planning, designing, managing and coordinating the experiment ECM activities.	Acuña, Silvia Teresita
Librarian	Role responsible for supervising the research team using the library (repository) during the experimental research life cycle. Additionally, this role will liaise with the ECM manager to create and set up versioning.	Gómez, Marta
Research Team Member	This role interacts with the repository performing operations on the ECM generated during the project and is the principal producer/consumer of the information under version control.	Research group

A.4 Management Plan Implementation

A.4.1 Identify and Enter Baselines

Experimentation activities were taken into account to establish the ECMP baselines. Figure 19 shows an excerpt from the information on the baselines defined in the personality quasi-experiment ECMP.

UNIVERSIDAD AUTÓNOMA DE MADRID (UAM)

School of Engineering – Telecommunications Engineering

Experiment Name: Study of the influence of personality and team climate on software development and team member satisfaction.

Research Group : GRISE

Date : 04/02/2014

Code	Name	Experimental Configuration Element	Date of delivery	Replications of the Experiment			
				UPM 05-06	UAM 05-06	OSLO 05-06	UCML
LB01	Planning phase	Problem statement		X	X	X	
LB02	Operationalization phase	<ul style="list-style-type: none"> - Test NEO-FFI (Costa y McCrae, 2002). - Gross Cohesion Questionnaire (Stokes, 1983). 	2004/2005	X	X	X	

Figure 19. Personality quasi-experiment baseline inventory instrument

A.4.2 Define Repository Structure

Figure 20 shows the repository directory structure that was implemented on the secondary storage device to save the different experimental configuration elements adopted in the ECMP.

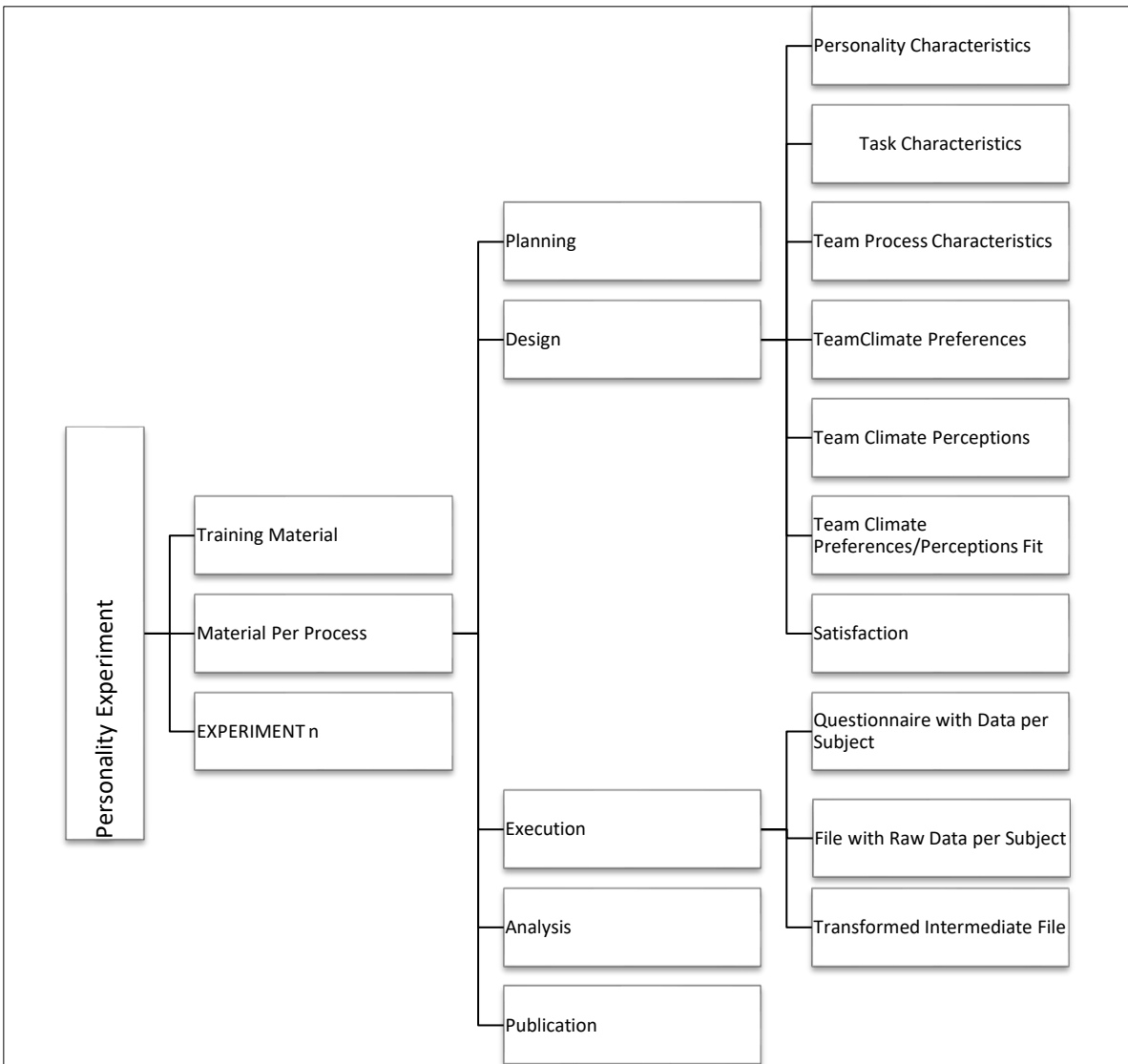


Figure 20. Personality quasi-experiment directory structure

A.4.3 Enter Composition and Derivation Relations between Experimental Configuration Elements

The composition relations were entered by identifying the relation between the tests applied with the material required to execute the test. Figure 21 shows an excerpt of the personality quasi-experiment ECMP information.

UNIVERSIDAD AUTÓNOMA DE MADRID

School of Engineering – Telecommunications Engineering

Experiment Name: Study of the influence of personality and team climate on software development and team member satisfaction.

Research Group : GRISE

Date : 04/02/2014

Experimental Configuration Element Component	Experimental Configuration Element Components	Type
Personality test (NEO-FFI)	Neuroticism Factor Extraversion Factor Openness to Experience Factor Warmth Factor Conscientiousness Factor	Doc
Teamwork Climate Preferences Questionnaire (TSI)	Participative Safety Factor Innovation Support Factor Team Vision Factor Task Orientation Factor	Doc Doc

Figure 21. Personality quasi-experiment composition relation instrument

The derivation relations were established to identify the experimental configuration elements that are used to generate other experimental configuration elements. Figure 22 shows an excerpt of the information on the derivation relations of the personality quasi-experiment ECMP.

UNIVERSIDAD AUTÓNOMA DE MADRID

School of Engineering – Telecommunications Engineering

Experiment Name: Study of the influence of personality and team climate on software development and team member satisfaction.

Research Group : GRISE

Date : 04/02/2014

Source Experimental Configuration Element Name	Derived Experimental Configuration Element Name	Type
Personality test (NEO-FFI)	Files with data from each participant questionnaire	xls
Team software development practical assignment	Assessment of practical team software development assignment	Doc

Figure 22. Personality quasi-experiment derivation relation instrument

A.4.4 Enter Experiment and Replication Succession Relations

This section records the information on the quasi-experiment administered by the research group. Figure 23 shows an excerpt of the details of the personality quasi-experiment executed at the Universidad Autónoma de Madrid.

UNIVERSIDAD AUTÓNOMA DE MADRID

School of Engineering – Telecommunications Engineering

Experiment Name: Study of the influence of personality and team climate on software development and team member satisfaction.

Research Group : GRISE

Date : 04/02/2014

No	Name	Code	Domain/Subject Matter	Description	Date	Experimenters	Location
1	Quasi-experiment UAM C0405	001	Personality Team climate Task characteristics Team processes Software quality Satisfaction	The quasi-experiment deals with the influence of personality and team climate on software development and team member satisfaction	2004/2005 academic year	Silvia Teresita Acuña Marta N. Gómez	C:\Mis documentos\CROSS\ANALISIS DATOS UAM-CEU CROSS 0405

Figure 23. Personality quasi-experiment version inventory instrument

The number of internal and external experiment replications totalled six up to 2006. Figure 24 shows an excerpt of the information about the two versions of the quasi-experiment replications, executed at the Universidad Politécnica de Madrid and the Universidad Autónoma de Madrid in 2005 and 2006.

UNIVERSIDAD AUTÓNOMA DE MADRID

School of Engineering – Telecommunications Engineering

Experiment Name: Study of the influence of personality and team climate on software development and team member satisfaction.

Research Group : GRISE

Date : 04/02/2014

No: 01

No	Replication Name	Type	Date	Site	Experimenters	Location
001	Quasi-experiment UPM 05-06	External, joint, differentiated	2005-2006	UPM	Marta Gómez Silvia Acuña Sira Vegas Angélica Antonio Andres Silva	C:\Mis documentos\CROSS OK\ANALISIS DATOS UPM CROSS 0506
002	Quasi-experiment UAM 05-06	Internal	2005-2006	UAM	Marta Gómez Silvia Acuña	C:\Mis documentos\CROSS OK\ANALISIS DATOS UAM-CEU CROSS 0506

Figure 24. Personality quasi-experiment replication version inventory instrument

A.4.5 Identify and Enter Experimental Configuration Elements and Versions

The information on seven experimental configuration elements and their versions was entered in the ECMP. Figure 25 shows the information entered in the experimental configuration element called Group processes questionnaire, which measures team cohesiveness, and a questionnaire to measure intragroup conflict (task conflict and social conflict) composed of seven and three items, respectively, whereas Figure 26 shows the information on the two versions of the experimental configuration element of the replications conducted at the Universidad Autónoma de Madrid in 2004 and 2006.

UNIVERSIDAD AUTÓNOMA DE MADRID

School of Engineering – Telecommunications Engineering

Research Group : GRISE

Date : 04/02/2014

No 002	Fields	Data
Experimental Configuration Element number or code		UAM200412
Experimental Configuration Element name		Group processes questionnaire
Experimental Configuration Element description		Questionnaire to measure team cohesiveness and questionnaire to measure intragroup conflict (task conflict and social conflict) composed of seven and three items, respectively
Baseline of which it is part		
Experimental Configuration Element author/s		Stokes (Gross Cohesiveness Questionnaire, 1983) and Jehn (Intragroup Conflict Scale, 1995).
Date created		
Replication identification		UAM 04-05
Experimental Configuration Element type (document, spreadsheet, program)		Doc
Location		C:\Mis documentos\CROSS OK\CUESTIONARIOS CROSS
Version number		01, 02
Version date		2004, 2005

Figure 25. Instrument for identifying the group processes questionnaire experimental configuration element

Experiment Name: Study of the influence of personality and team climate on software development and team member satisfaction.

Code : UAM200412

Research Group : GRISE

Date : 04/02/2014

Ver	Type	Rep.	Description	Date	Location
1	Digital	Quasi-experiment UAM C0405	A test for determining group processes: team cohesiveness and interdependency	2004/2005	C:\Cross OK\CUESTIONARIOS CROSS\Cuestionarios UAM 0405
	Hardcopy				
2	Digital	Quasi-experiment UAM C0506	A test for determining group processes: team cohesiveness and interdependency		C:\Cross OK\Cuasiexperimento OSLO
	Hardcopy				

Figure 26. Group processes questionnaire experimental configuration element version inventory instrument

Appendix B:

SUS questionnaire for measuring usability

Please check the option that denotes your immediate response to each statement. Don't think too long about each statement. Make sure you respond to every statement. If you don't know what to respond, simply check option "3".

No	Items	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	I would like to use this framework on a regular basis	1	2	3	4	5
2	I found the framework unnecessarily complex	1	2	3	4	5
3	I thought the framework was easy to use	1	2	3	4	5
4	I think that I would need the support of a specialist to be able to use this framework	1	2	3	4	5
5	I found that the range of functions in this framework were well integrated	1	2	3	4	5
6	I thought there was too much inconsistency in this framework	1	2	3	4	5
7	I would imagine that most people would learn to use this framework very quickly	1	2	3	4	5
8	I found the framework very awkward to use	1	2	3	4	5
9	I felt very confident using the framework	1	2	3	4	5
10	I needed to learn a lot of things before I could get going with this framework	1	2	3	4	5

Appendix C:

Instruments for measuring usefulness (I)

Experimenter Questionnaire for Identifying the Status of the Experimental Materials

Name :
Experiment Name :
Role :
 Experimenter []
 Replicator []
 Other []
 Please specify : _____
Date : ____/____/____/

1. Select the type of storage used to store the experimental materials:

Local hard disk []
Server disk []
Web []
Other []
Please specify: _____

2. Which of the following structures are used to organize your experimental materials?

Directory []
Database []
Others []
Please specify: _____

3. How are changes with respect to the original experimental materials controlled?

No control []
Printed list of materials with date of modification []
Digital document stored on your computer containing the list of materials []
Digital document stored on a server containing the list of materials []
Hypertext on the web containing the list of materials []
Configuration management process []
Software tool managing materials []
Others []
Please specify: _____

4. What materials are provided to carry out the experiment replication?

No materials []
Materials used to execute the experiment []
Instructions []
Exercises []
Tutorials []
Training material for experimental subjects []
Others []
Please specify: _____

5. Select from the list. Experimental material is accessible via:

Experiment manager []
Experimenters linked to the experiment []
Face-to-face meeting []
Others []
Please specify: _____

6. Select the materials available for the experiment replication processes:

- None
- Material without user instructions
- Material with user instructions
- Material with user tutorials
- Training material
- Exercises
- Others

Please specify: _____

7. What type of support is available for comprehending the materials used in the experiment?

- No support
- Material without user instructions
- Material with user instructions
- Web portal with links
- Hypertext to browse materials
- Tutorials
- Others

Please specify: _____

8. Please specify the media you usually use to distribute and locate the experimental material:

- No media
- Email
- Private portal
- Public portal
- Other media

Please specify: _____

9. Select from the list the experimental materials adapted to different contexts to execute the experiment:

- None
- Materials used to execute the experiment
- Instructions
- Tutorials
- Training material for experimental subjects
- Exercises
- Others

Please specify: _____

Replicator Questionnaire for Identifying the Status of the Experimental Materials

Name :
Experiment Name :
Role :
 Experimenter []
 Replicator []
 Other []
Please specify : _____
Date : ____/____/____/

1. Select the type of storage used to store the experimental materials:

Local hard disk []
Server disk []
Web []
Other []

Please specify: _____

2. Which of the following structures are used to organize your experimental materials?

Directory []
Database []
Others []

Please specify: _____

3. How are changes with respect to the original experimental materials controlled?

No control []
Printed list of materials with date of modification []
Digital document stored on your computer containing the list of materials []
Digital document stored on a server containing the list of materials []
Hypertext on the web containing the list of materials []
Configuration management process []
Software tool managing materials []
Others []

Please specify: _____

4. What materials are provided to carry out the experiment replication?

No materials []
Materials used to execute the experiment []
Instructions []
Exercises []
Tutorials []
Training material for experimental subjects []
Others []

Please specify: _____

5. Select from the list. Experimental material is accessible via:

Experiment manager []
Experimenters linked to the experiment []
Face-to-face meeting []
Others []

Please specify: _____

6. Select the materials available for the experiment replication processes:

- None
- Material without user instructions
- Material with user instructions
- Material with user tutorials
- Training material
- Exercises
- Others

Please specify: _____

7. What type of support is available for comprehending the materials used in the experiment?

- No support
- Material without user instructions
- Material with user instructions
- Web portal with links
- Hypertext to browse materials
- Tutorials
- Others

Please specify: _____

8. Please specify the media that the original experimenters used to supply the experimental material:

- No media
- Email
- Private portal
- Public portal
- Other media

Please specify: _____

9. Select from the list the experimental materials that you adapted to different contexts to execute the experiment:

- None
- Materials used to execute the experiment
- Instructions
- Tutorials
- Training material for experimental subjects
- Exercises
- Others

Please specify: _____

Appendix D:

Instruments for Measuring Usefulness (II)

Design of Test Cases to Measure ECMP Usefulness

Name :
Experiment Name :
Role :
 Experimenter []
 Replicator []
 Other []
 Please specify : _____
Date : ____/____/____/

Requirements: Implementation of the Personality Quasi-Experiment ECMP

Use the instance of the personality quasi-experiment configuration management plan to answer the following questions.

Question 1	Do you know which experiments the research group has executed?
Action in instantiated plan	Search the personality quasi-experiment ECMP index
Question 2	Do you know which replications the research group has executed?
Action in instantiated plan	Search the personality quasi-experiment ECMP index
Question 3	Are you familiar with the Group process questionnaire experimental element versions?
Action in instantiated plan	Search the personality quasi-experiment ECMP index
Question 4	Are you familiar with the Task characteristics questionnaire experimental element versions?
Action in instantiated plan	Search the personality quasi-experiment ECMP index
Question 5	Download the Personality test (NEO FFI) configuration element
Tool operator	Search the personality quasi-experiment ECMP index for the Personality test (NEO FFI). Download the Personality test (NEO FFI).
Question 6	Download the Group process questionnaire configuration element
Tool operator	Search the personality quasi-experiment ECMP index for the Group processes questionnaire Download the group process questionnaire

Questionnaire on ECMP Usefulness after Test Case Execution

Name : _____
Experiment Name : _____
Role : _____
 Experimenter []
 Replicator []
 Other []
Please specify : _____
Date : ____/____/____/

Please check the option that denotes your immediate response to each statement. Don't think too long about each statement. Make sure you respond to every statement. If you don't know what to respond, simply check option "N".

No.	Item	TD	D	N	A	TA
1	The use of the ECMP reduces the time taken to query the information on the experimental materials					
2	The use of the ECMP could reduce the time taken to locate the experimental materials.					
3	The tool implementation of the ECMP helped me to administer the experimental material versions					
4	The ECMP instruments implemented in my experiment helped me to discover the different versions of the experimental materials.					
4	The implemented ECMP instruments helped me to discover the relationships between experimental materials.					
5	The instantiated ECMP helped me to organize the experimental materials.					
6	I would like to use the implemented experiment ECMP to administer the experimental material					

TD: Totally disagree
D: Disagree
N: Neither agree nor disagree
A: Agree
TA: Totally agree