Information and Software Technology 94 (2018) 111–129

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

A systematic mapping study about socio-technical congruence

José María Sierra, Aurora Vizcaíno*, Marcela Genero, Mario Piattini

Institute of Technology and Information Systems, Alarcos Research Group, Escuela Superior de Informática, University of Castilla-La Mancha, Ciudad Real, Spain

A R T I C L E I N F O

Keywords:
Socio-technical congruence
Social-technical dependencies
Coordination
Tools
Systematic Mapping Study

A B S T R A C T

Context: Lack of coordination may create significant problems between work teams, this problem is even most critical when team workers are geographically distributed as it results in cost increases and delays in the projects. There exists a technique called Socio-Technical Congruence (STC) that aims at helping to measure and control the coordination level existing in an organization at their different levels.
Objective: The objective of this paper is to carry out a systematic mapping of the field of socio-technical congruence. The aspects of particular interest for this article are: Socio-Technical Congruence definition, different ways to measure it, available tools to measure or that can help to measure it, the areas of application, its benefits and the case studies that analyze the effects Socio-Technical Congruence has on the organization as regards the products quality and the improvements in performance in the long term development, in an attempt to characterize the state of the art of this field identifying gaps and opportunities for further research. Therefore, companies could use this work as a starting point to apply STC measures in their work teams.
Results: This paper presents the results of a systematic mapping of Literature about Socio-Technical Congruence (STC) in order to investigate and classify the existing articles and conferences about the subject, as well as summarizing the most important aspects in regards to provide a general overview about the existing studies.
Conclusions: After analyzing the 40 papers found, we can conclude that there is no one standard measure of socio-technical congruence, although most take the proposal by Kwan et al. applying adaptations and improvements on it as regards the environment that it will be focused on. In general, most case studies talk about the benefits of STC control in organizations. However, only one paper focus on global software development where the problems of communication, coordination and control are an important risk. Moreover, there are only a few papers that explore the risks of excessively overloading users with coordination iterations when controlling STC. In fact, no case study to examine these risks and their effect on developers' productivity has been found. The small number of studies found on STC, together with the research gaps we have pointed out, suggest that further investigation on socio-technical congruence is required.

1. Introduction

It is becoming increasingly common for development teams of software organizations to find themselves distributed and geographically separated. Distribution and modularity of software development have as their main objective to reduce labour costs, but this does imply implicit risks connected with coordination gaps [1]. Lack of coordination may create significant problems between work teams, all of which results in increases in costs and delays in the projects, and these effects could counteract the modularity and task externalization [2]. This issue is even more problematic in both distributed development and global software development; interest has thus grown in organizations as regards measuring, assessing and correcting the level of coordination reached among their work teams [1].

There exists a technique called Socio-Technical Congruence (STC) that aims to help measure and control the level of coordination existing in an organization at its own different levels [3]. This is the concept of Socio-technical congruence, defined in [3](page 1) as follows:

> "*A technique to measure task dependencies among people, and the 'fit' between these task dependencies and the coordination activities performed by individuals*".

With the help of the right mechanisms, an organization can correct the coordination gaps early, thus avoiding possible misunderstandings caused by a lack of information that ends up delaying the work, or in the worst-case scenario, leading to the failure of the project [1].

Aware of the importance of this topic, the software factory (name is omitted by privacy) with which we are working on the LPS-BIGGER project (http://www.cienlpsbigger.es/pt4-bigdata-en-el-software.html) asked us to design and implement a tool that could be used to measure

* Corresponding author.
E-mail addresses: jose.sierra@uclm.es (J.M. Sierra), aurora.vizcaino@uclm.es (A. Vizcaíno), marcela.genero@uclm.es (M. Genero), mario.piattini@uclm.es (M. Piattini).

http://dx.doi.org/10.1016/j.infsof.2017.10.004
Received 22 October 2016; Received in revised form 11 September 2017; Accepted 6 October 2017
Available online 09 October 2017
0950-5849/ © 2017 Elsevier B.V. All rights reserved.

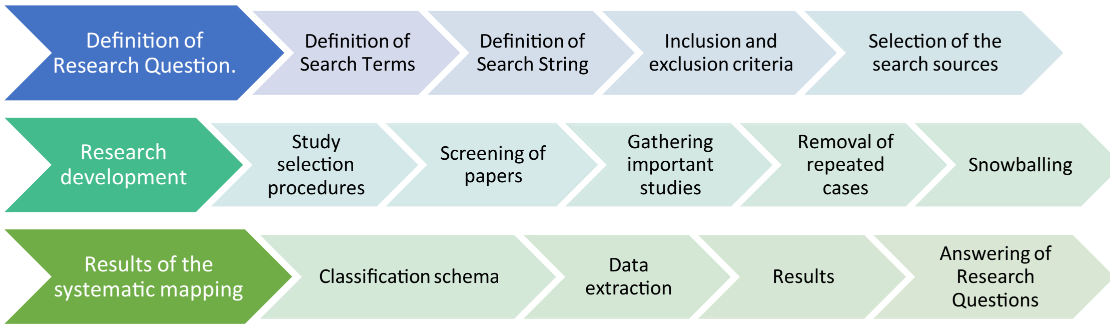

Fig. 1. Research protocol.

the STC in this company as it is very interested in monitoring the coordination among employees, projects and the different factories that this company has in order to detect a lack of coordination and prevent possible problems in advance. Before starting, we decided to analyse the related literature in order to discover which the best STC measure was and whether there was a tool that would fulfil the company's needs, in which case the company could perhaps buy it. After doing so, we realised that we had found no information about a standard measure for STC, best practices for designing tools to support STC or a tool that would fulfil our needs. In order to discover whether we had overlooked an important paper about the topic we therefore decided to perform a systematic mapping review. To the best of our knowledge, there is no previous systematic study in the field of socio-technical congruence. In our study we focus on a definition of socio-technical congruence, along with different ways in which to measure it, and discover the tools available to measure it, or that can help to do so, along with the areas of application. We also seek to look at the benefits and risks of measuring STC, along with the case studies that analyze the effects that socio-technical congruence has on the organization as regards product quality and improvements in performance in long-term development. We have thus attempted to characterise the state of the art of this field, identifying gaps and opportunities for further research.

This paper, then, presents the results of a systematic mapping of literature about socio-technical congruence, aiming to investigate and classify the existing papers and conferences about the subject, as well as to summarize the most important parts, so as to provide a general overview of the existing studies. The paper is organized as follows: Section 2 presents the related work, and Section 3 describes how the systematic mapping was planned. In Section 4 a description of the search conducted in the study is set out. The scheme used to classify the papers obtained in the search is presented in Section 5, while in Section 6 we present the results we obtained during the study. The research questions are answered in Section 7. Finally, Section 8 summarizes the conclusions of the paper and outlines challenges that may lead to future research.

2. Related work

To the best of our knowledge, in the relevant literature there are no systematic literature reviews (SLR) or systematic mapping studies (SMS) which deal with all the issues that have to do with socio-technical congruence. It is true, however, that there is a partial systematic review by Betz et al. [4] that examines empirical research related to Conway's law and its application for cross-site coordination. Conway's law is therefore the starting point of socio-technical congruence. The research presented in [4] summarizes findings from a systematic review of empirical studies. These studies validate the importance of socio-technical congruence by looking at the pros and cons of congruent and incongruent scenarios, studying the evolution of socio-technical congruence over time.

There is also a piece of research from Portillo [1] that allows the state of the art to be established, giving a brief review of the different metrics established for socio-technical congruence, as well as of their possible applications. The different metrics presented in this paper can be divided into two well-differentiated groups: metrics based on matrices and metrics based on social networks. Similarly, two main application areas for socio-technical congruence can be highlighted:

- Socio-technical congruence control oriented toward management: to help in decision-making processes.
- Socio-technical congruence control oriented toward the user: to raise awareness regarding changes in the work artefacts.

These data make it clear that a complete systematic review or a systematic mapping study are needed if we are to gain a thorough understanding of the state in which research into socio-technical congruence finds itself. Furthermore, in addition to the shortcomings that can be noted in the studies [1,4], it must be observed that they include only studies published until 2012 and 2011, respectively.

3. Search process

The purpose of this study is to determine and characterize the state of the art of socio-technical congruence in software engineering, analyzing the existing proposals and research work, thus identifying potential gaps, risks and opportunities for future research.

To carry out this systematic mapping, we followed the recommendations in [5,6]. In this section we present the planning of each step of the study: research questions, data sources and search strategy, along with the classification. We have based on in order to carry out the research protocol that has been produced to conduct the Systematic Mapping Study, as presented in Fig. 1.

3.1. Definition of research questions

The Systematic Mapping Study conducted helps to answer the following research question that was proposed in (RQ), and at the same time, this question has been divided into more questions (RQ1-8, see Table 1):

Table 1
Research questions.

Ref.	Questions
RQ1	How has socio-technical congruence been defined in literature?
RQ2	What are the properties of socio-technical congruence?
RQ3	What benefits and risks can socio-technical congruence provide to the organizations?
RQ4	What ways of measuring socio-technical congruence have been proposed?
RQ5	What are the pros and cons of different ways of measuring socio-technical congruence?
RQ6	In what areas can socio-technical congruence be applied?
RQ7	What kind of available tools help to measure Socio-technical congruence?
RQ8	What case studies have been published about Socio-technical congruence?

***RQ:** What has been published about socio-technical congruence in literature since 2000?*

This question was unfolded into those set out below, in order to make the investigation clearer and easier:

The reason why this research considers only publications after 2000 is because, although the first evidence of the use of the term "socio-technical congruence" was in Cataldo et al. [3] when it was defined, we decided to broaden the research range back to the year 2000 to make sure that the results are complete, and also to verify that there is no work prior to Cataldo's on this topic.

3.2. Definition of research terms

To decide the terms involved in our search, we conducted a non-systematic review of the literature dealing with socio-technical congruence. The most important terms and keywords used to refer to this topic were extracted from significant papers, along with every variation of the term "socio-technical congruence" used by authors. The candidate terms that subsequently formed the most suitable search string for the research are shown below in Table 2.

3.3. Definition of the search string

Several search strings combining the candidate terms were studied; S1 (see Table 3) was eventually chosen as the most suitable chain for the investigation. S1 was selected after verifying that the results and adaptation obtained previously included every important paper that had been studied in an earlier non-systematic review. It was also chosen after confirming that it excluded a great amount of papers that were not of interest to us, since these dealt neither with congruence nor the tools to measure it. S1 is included in each variation of the term "Socio-Technical Congruence" that has been used by several authors in papers studied previously. The term "coordination" has also been included, because it is common in every definition of socio-technical congruence found in the documents studied in the non-systematic review of literature mentioned above.

The string chosen was adapted in a specific way for each of the digital libraries because of search source limitations where the insertion of complex strings was not allowed. Adaptation of different search sources can be found in APPENDIX B.

3.4. Inclusion and exclusion criteria

The objective of this review was to discover all the papers that present any research related to definition of Socio-Technical Congruence, the advantages of using this, and the tool applied to calculate it; these papers had to be written in English and published after 2000. The documents that met the following criteria were therefore admitted:

1. Documents that answer our research questions, in other words:
   a. They describe what Socio-Technical Congruence is.
   b. They describe tools that help to measure socio-technical congruence.
   c. They describe the benefits and risks of socio-technical congruence.
   d. They describe the application areas of the socio-technical congruence.
   e. Case studies about socio-technical congruence.
2. Duplicate studies. When several papers are written by the same authors describing the same topic, the most complete one will be considered.
3. Documents published after 2000.
4. Documents written in English.

The following exclusion criteria are applicable in this review; that is, the kind of studies that are excluded are:

1. Those papers that do not focus on Socio-Technical Congruence.
2. Those documents that say that they use socio-technical congruence, but their goal is not to describe socio-technical congruence itself, nor to specify tools for measuring STC or to discuss case studies about STC.
3. Those documents where the search terms appear only in the references.
4. Repeated documents. The first instance found will be considered.

3.5. Source research election

The databases chosen were Scopus, ACM and IEEE Explore, since they include the most important conferences about the topic, such as The International Conference on Global Software Engineering (ICGSE) and the International Conference on Software Engineering (ICSE). In Table 4 these data sources are shown, together with the criteria applied to the results obtained using the search string in each of them:

4. Primary study selection process

This section describes the process that has been followed to carry out the search for primary studies used to define the state-of-the-art: the process of study selection procedures, results obtained in search source, data extraction and classification of different documents.

Table 5 displays the quantity of papers selected in each of the databases chosen after the search, which was conducted on February 8, 2016.

The meaning of each column of the Table 5 is explained follows:

- *Found*: papers found in each database using the search string.
- *Relevant*: The papers selected from the initial search results containing the search terms in the abstract, the introduction, keywords or title.
- *Not repeated*: Papers considered relevant that had not yet been found

Table 2
Research terms.

Ref.	Term	Ref.	Term
T1	*Socio-Technical Congruence*	**T2**	*Socio Technical Congruence*
T3	*SocioTechnical Congruence*	**T4**	*Technical dependencies*
T5	*Social dependencies*	**T6**	*Socio technical relationship*
T7	*Software dependencies*	**T8**	*Coordination*

Table 3
Research string.

Ref.	String
S1	*("technical dependencies"* ***or*** *"technical dependency"* ***or*** *"social dependencies"* ***or*** *"social dependency"* ***or*** *"software dependencies"* ***or*** *"software dependency"* ***or*** *"socio technical relationship"* ***or*** *"socio technical congruence"* ***or*** *"socio-technical congruence"* ***or*** *"sociotechnical congruence"* ***and*** *"coordination")*

Table 4
Database and search criteria for the result screening.

Digital Library	Options
IEEE Explore	2000–2016, Conference Publication, Journal & Magazines
ACM	2000–2016, Proceeding, Journal & Magazines
Scopus	2000–2016, Computer Science, Conference Papers, Papers, English

Table 5
Databases and the papers selected for each stage.

Digital Library	Found	Relevant	Not repeated	Primary
IEEE Explore	238	26	26	16
ACM	38	17	9	8
Scopus	287	57	29	5
Total:	**563**	**100**	**64**	**29**

in other databases.

- *Primary:* After reading the full paper we realise that these papers were suitable for our research as they fulfilled the inclusion criteria.

4.1. Snowballing

We should point out that the search for primary studies was recursive (Snowballing). That is, once a primary study had been identified in one of the data sources, the references in that primary study were explored recursively by following the same search criteria. The 'Snowballing' method was conducted to find more documents, following the model described by Wohlin in [7], since a low number of primary studies had already been obtained. After this new search, 11 extra documents were collected, leaving a total of 40 primary studies.

5. Classification schema

After reading the papers, we realised that they could be classified into seven categories according to the research question that the paper answered. This classification of the papers helps us to process their information. These categories describe, for instance, the existing STC measures; it was detected that within this category there were two main sub-categories: those based on matrices and those that were based on social networks. Apart from those, we found another group of papers describing case studies. These can be divided into sub-categories of case studies carried out in an academic environment, others performed in a business setting, and those conducted in open source settings. The studies have thus been classified into the different types represented in Fig. 2:

It was considered that to justify forming a new category there had to be two or more papers on the same subject. This principle led us to establish the "others" category, which includes documents that are not sufficient in number for them to warrant their having their own group.

6. Results of the mapping

This section shows the statistics of the results obtained, such as papers found, publication forums, annual evolution of publications, the countries that contribute the most, research environments, etc.

Having finished the search, a classification of the papers was conducted once the whole content had been read, taking into consideration the classification schema defined in Fig. 2 and the classification form shown in APPENDIX C. Some of the papers were included in two different categories, as they contained information that was clearly differentiated and that could be studied separately; an example would be those papers presenting measures which also had study cases. The diagram in Fig. 3 presents the number of papers that were found in each category. APPENDIX A specifies the categories into which they were classified. The papers classified into two different categories have been counted separately for Fig. 3. The total amount in the diagram is therefore higher (49) than the number of primary studies (40).

Based on the results shown in the chart in Fig. 3, it can be stated that there is a great amount of publications about tools. However, it is important to consider that the vast majority of the papers of this kind deal with tools whose purpose is to help extract information so that socio-technical congruence can be measured, rather than to actually carry out that measurement themselves. The case study category, which has been conducted mainly in the business environment, is particularly worth highlighting. It can be seen that there is a lack of studies in open source (only 6 papers) and academic environments (only 2 papers), where the results of the few studies that do exist are very dissimilar. We would have liked there to have been a few more papers that analyzed the risks of overloading users with information telling them how to increase socio-technical congruence; these risks are only briefly explained by Valetto et al. in [8].

Fig. 4 illustrates the evolution of the number of papers over time. The first publication that we are aware of about a tool [9] that could subsequently be used to help in the measurement of socio-technical congruence came on the scene in 2005, although it is true that the term was defined in 2006. In that year two publications appear: the first one addressed the issue of socio-technical congruence [3] and the other presented a new tool to help measure it [10]. In 2007 another definition of congruence and new tools emerged, now making a total of 3 papers. 2008, it must be noted, clearly stands out as the year with most publications on this topic of socio technical congruence, with a total of 7, followed by 2009, with 6 publications. Over these two years, great interest was aroused on the topic, with a total number of papers that was greater than for all the previous years combined. In 2010, there was a drop in the number of publications, but the amount then grew again in the subsequent years, reaching 5 publications in both 2012 and 2013. It can be seen that 2015 was the year with the fewest number of papers published since the term socio-technical congruence was defined (2006), as only one publication was found. This 2015 paper is a thesis compendium of publications [11], which includes several pieces of work that have been studied separately [12–14]. Although the search was conducted on February 8, 2016, no paper from this year has been included in Fig. 4, since no result was found.

Fig. 5 shows the evolution of the publications over time with respect to the categories in Fig. 2**.** This diagram allows us to observe where the focus of interest is as regards the knowledge acquired about socio-technical congruence. In Fig. 5 it is clearly observable that proposals for socio-technical congruence, or evolutions of the measures mentioned above, have been published almost annually, with respective case

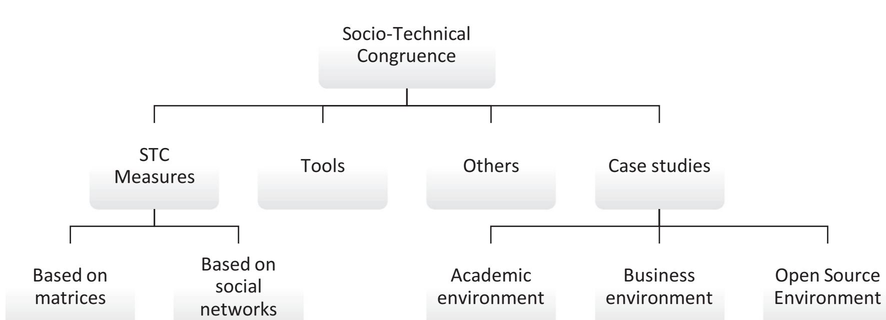

Fig. 2. Classification Schema.

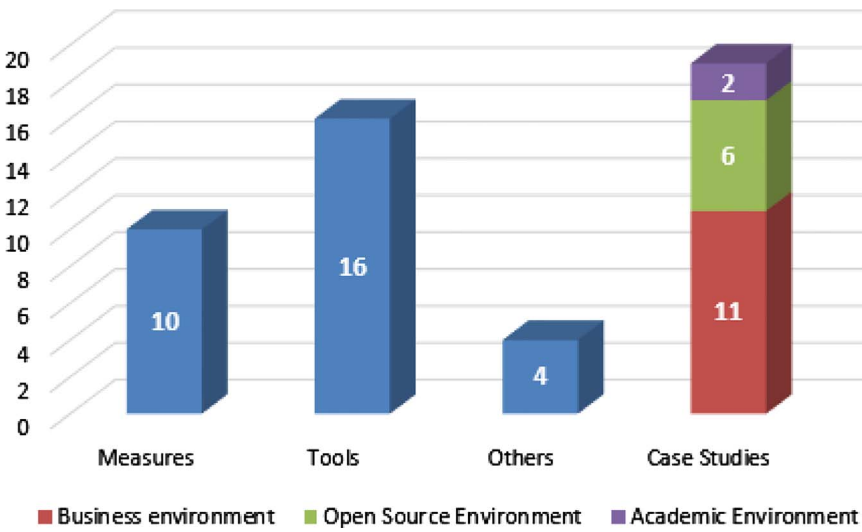

Fig. 3. Number of papers per type.

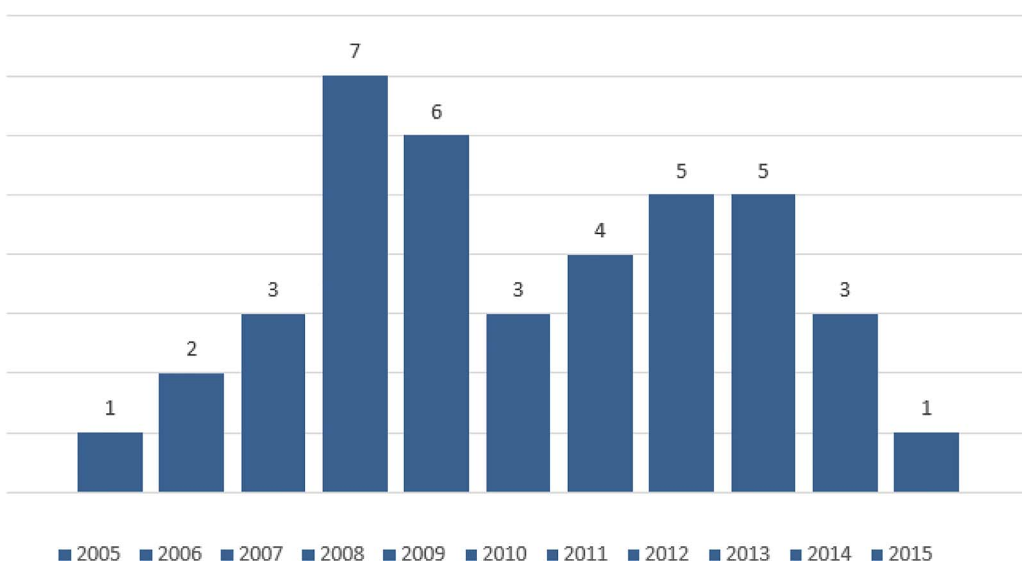

Fig. 4. Number of papers per year.

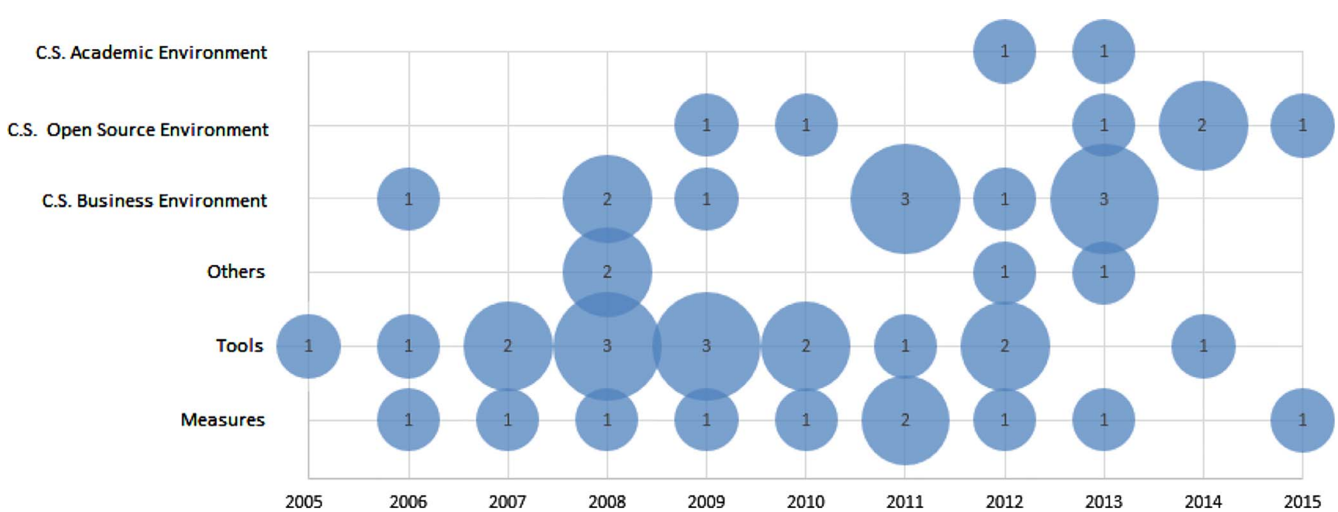

Fig. 5. Number of papers per type and year.

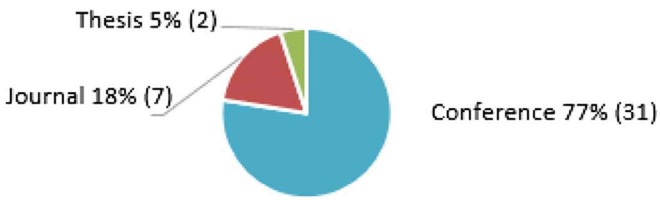

Fig. 6. Number of papers per type and year.

studies to prove their effectiveness. Research on socio-technical congruence has shown increasing interest in open source environments. In fact, the great majority of studies found over the last three years have focused on those kinds of environments. The use of tools is another area where there were more publications than in the rest of the topics. On the other hand, of those published, there were only a few papers describing case studies applied in academic environments.

We should point out that most of the papers found were published in conferences, making up 77% of the total, followed by publications in journals (18%) and theses (5%), as shown in Fig. 6.

As regards the forum where papers were published, we found that the conferences where most papers have been published on this topic are: the International Conference on Software Engineering (ICSE), with a total of 11 papers; it is indeed the most distinguished forum, both globally and annually. The forum with the next-highest number of papers was the Conference on Computer-Supported cooperative Work (CSCW), with three documents. The other publications can be seen in Table 6, where the first column shows the names of all publications in which documents have been found; the second column shows the number of papers in each of those publications.

We also analyzed the countries where the authors work. It was found that 49% of authors made contributions from organizations and universities in the United States, followed by Sweden, with 13%. The origins of the remaining 38% were very widely-dispersed, as shown in Table 7.

As regards the environment where the studies were conducted, it can be found that 73% of these belong to the academic field, compared to 27% of studies that are developed by the industry, as can be seen in Fig. 7.

Table 6
Number of papers per publication.

Publication	Number of papers
SEAFOOD, Academy of Management Proceedings, CHASE, CIRCUS, CollaborateCom, CSEE&T, ESEC/FSE, ESEM, GROUP, ICGSE, ICSSP, IEEE Software, Journal of Software, OOPSLA, OpenSym, OSS, PROFES, RESER, SIGSOFT, Tampere University of Technology, University of Castilla-La Mancha, VL/HCC	1
IEEE Transactions on Software Engineering	2
Information Sciences	2
CSCW	3
ICSE	11

Table 7
Number of contributions per country.

Country	Number of papers
Australia, Finland, Germany, Ireland, Italy, Latvia, Pakistan, United Arab Emirates	1
Denmark	2
Spain	3
Brazil	4
Canada	5
China	5
Sweden	9
USA	35

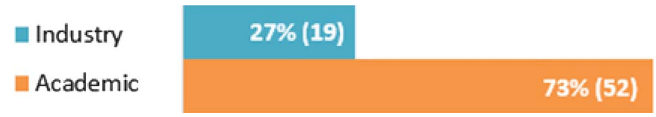

Fig. 7. Contributions according to the field of development.

7. Research questions

In this section we present an analysis of the primary studies obtained; this was performed following the classification criteria and research questions that have been outlined previously. On the basis of the analysis carried out on the primary studies selected, the answers to the stated research questions are as follows:

7.1. RQ1. How has socio-technical congruence been defined in literature?

In this section the different definitions found are presented, several of which belong to the same authors. However, they improved or made them more specific in later papers. In 1968 Melvin Conway [15](Pg. 31) suggested that "*Organizations which design systems … are constrained to produce designs which are copies of the communication structures of these organizations*" ; this is known as "*Conways' law.*" In 2006, an investigation branch dealing with 'Socio-Technical Congruence' appeared—in which Conway's study [15] was taken as a base—. This term was coined by Cataldo and his colleagues in [3] (page 1), where it is defined as follows: "*A technique to measure task dependencies among people, and the 'fit' between these task dependencies and the coordination activities performed by individuals*" . Similarly, in [16] (page 4), these same authors redefined the term as "*The match between the coordination requirements established by the dependencies among tasks and the actual coordination activities carried out by the engineers.*" Finally, in [17] (page 345), they concluded that the definition is "*The degree of matching between two elements, coordination requirements and actual coordination activities.*" This definition is more general than previous ones, as in the first definitions only people are considered. However, this one considers other entities that need coordination, such as factories, projects, teams, etc.

In a similar vein, according to Wan et al. [2] (page 1), socio-technical congruence is referred to as "*The fit between an organization's coordination requirements and an organization's social interactions*" . In this case the term socio-technical congruence is applied to the organization's general coordination needs. However, in the following definition, found in the same paper, the focus is much more specific as they talk about software development projects: "*An intuitive way to compare required coordination effort within a software development project with the actual ongoing coordination*". In [18] (page 1), meanwhile, they omitted the word coordination, and use the concept of relationships instead, defining the term of socio-technical congruence as "*The alignment between a team's interpersonal relationships and work-derived technical relationships.*" The same authors put their definition into context in [19] (page 1), and therefore state; "*An approach that measures coordination by examining the alignment between the technical dependencies and the social coordination in the project*".

Other authors, such as Wagstrom et al. in [20] (page 1), simplify the definition, "*Socio-Technical Congruence relates the actual communication of an organization to the coordination requirements of that organization as derived from the dependencies among the tasks performed.*" Sarma et al. also proposed another definition in [21] (page 1). The definition proposed by Cataldo is therefore paraphrased as: "*The state in which a software development organization harbors sufficient coordination capabilities to meet the coordination demands of the technical products under development*". In this case, the concept is not focused on personal coordination but is a little more general since it discusses a software development organization.

7.2. RQ2.What are the properties of socio-technical congruence?

Only in the paper by Sarma et al. [21] did we find an answer to this question. The authors, taking into consideration the definitions set out above, list different properties of socio-technical congruence:

1. *STC is a state* that represents the fit between the Coordination Requirements, —necessary to carry out a given development situation—and Actual Coordination Actions conducted.
2. *STC is descriptive*: Congruence describes the state in which an organization finds itself at any moment in time. Nevertheless, this does not guarantee that the following state will be optimal as regards congruence.
3. *STC is dynamic*: STC measures the state of an organization over a particular period. In other words, an organization may be congruent at a certain time, but that does not guarantee that this same organization will be congruent at a later moment in time. This is because there may be changes in the coordination needs, dependencies or social structures.
4. *STC is multi-dimensional*: there are as many dimensions in congruence as there are different ways to coordinate work. Congruence is based not only on the communication between developers; there are also other important aspects such as experience, comments on tools, work patterns, resources, etc.
5. *STC can be defined at multiple levels*: whether an organization is congruent or not depends on how congruent individuals and teams are. Congruence must therefore be tackled at multiple levels, from individuals—which would be the lowest level—all the way up to the organization itself, or the organization system wherever there are several of them cooperating in a common project.
6. *STC involves both advantages and disadvantages*: It is certainly possible for an organization to achieve congruence at a particular level or dimension that is beneficial for workers, but which at the same time can be counterproductive at another stage. For example, it might happen that the work pattern that is introduced leads to incongruence at a certain more local stage, but that it is at the same time optimal at a global level. This being so, congruence must encompass all levels and seek global benefit.

7.3. RQ3.What benefits and risks can socio-technical congruence provide to the organizations?

To respond to the RQ3 research question, we have analyzed several studies that suggest that if an organization reaches a high level of socio-technical congruence, the resolution time for the modifications requested in the code is reduced. At the same time, the modification requests and errors decrease [3,16,17,22]. This means that when there is high socio-technical congruence, teams are more efficient.

If the level of socio-technical congruence is high, one of its main benefits is that we can ensure that the coordination needs of an organization are achieved. If the level of congruence has dropped in an organization, but possesses adequate tools for detecting congruence deficit, these would make it easier to detect and react to communication problems such as those reported in Šmite and Galviņa [23], where the lack of communication among some teams of developers leads to delays. In addition, in a case study described in [23] a lack of understanding in individuals was observed as regards their responsibilities; this in turn meant that problems arose with respect to coordination in solving problems. It should be pointed out that in [23] part of the organizational structure was hidden, which made a large amount of the necessary coordination impossible; this in turn led the project to fail.

In Li et al. [24] the authors propose a multidimensional approach to congruence. It involves increasing awareness about the use of resources on the part of each employee, as well as raising cognizance of the specific abilities and knowledge that colleagues possess, seeing which are actually necessary for the development of each task.

To sum up, it can be stated that a high level of congruence should lead the organization to reach a better level of coordination, where stakeholders are acutely aware of the communication that must be carried out, and know precisely what resources they have at their disposal and exactly where to act so that the coordination gaps observed may be closed as soon as possible [2,3,19,25]. All of this entails improvement in performance and software quality, as well as a reduction in costs.

It is important to get across the idea that coordination and communication in the organization implies some cost, since the burden of making sure all this happens is placed on the personnel. This might lead to a loss of productivity at an individual level, as referred to in [8]. This is therefore an additional risk that has to be taken into consideration.

7.4. RQ4.What ways of measuring socio-technical congruence have been proposed?

As STC is related to communication, measuring congruence is also a means of controlling and improving work team communication at both a local and distributed level. For this reason, when measuring STC it must be taken into account how individuals communicate among themselves within the organization. This implies that in addition to measuring STC itself, the measurements can be a means of detecting and solving possible communication problems that may arise from either too much, or too little, communication. Betz et al. [4] studied empirical research related to Conway's law and obtained that changes in communication affect the design structure of the software product. Moreover, in [1] Portillo-Rodríguez (page 34), it is claimed that "*measuring STC can result in a better level of coordination*".

Different authors have conducted studies with measures for calculating socio-technical congruence; these help us to answer our research question RQ4, as presented below. This section describes the most important details of the metrics proposed in the literature for measuring and controlling socio-technical congruence

7.4.1. Cataldo et al. proposal [3,16]

As reported in [16], calculation of socio-technical congruence is based on the existing relationship between two sets: one set for individuals working on tasks, and the other set for the tasks themselves and the relationship between them. These sets are represented in two matrices: "Task Assignment matrix" (T_A) and "Task Dependency matrix" (T_D). Each cell *ij* of the Task Dependency matrix represents that task *i* has dependencies on task *j* (Table 8 (a)). In relation to this, each cell *ij* of the Task Assignment matrix indicates that a worker *i* is

Table 8
(a) Task Dependency matrix (TD). (b) Task Assignment matrix (TA).

(a)

T_D	Task 1	Task 2	Task 3	Task 4
Task 1	–		1	1
Task 2		–		
Task 3			–	1
Task 4				–

(b)

T_A	Task 1	Task 2	Task 3	Task 4
Person 1	1			
Person 2		1		
Person 3			1	
Person 4				1

Table 9
Coordination Requirements matrix (CR).

C_R	Person 1	Person 2	Person 3	Person 4
Person 1	–		1	1
Person 2		–		
Person 3			–	1
Person 4				–

assigned to a particular task *j* (Table 8 (b)).

The individuals that must coordinate their activities have to be identified, taking into account these sets represented in their corresponding matrices. Firstly, by multiplying the T_A and T_D matrices we obtain a people by task matrix which represents the extent to which a worker will be aware of the interdependent tasks he/she is responsible for. The next step is to multiply the result of T_D*T_A by $T_A{}^T$ (the transpose matrix of T_A) so that a new people by task matrix is obtained, which is called 'Coordination Requirements matrix' (C_R) Table 9. This shows which workers must coordinate between themselves.

Matrix CR is represented in [16] as the formula below:

$$C_R = T_A * T_D * T_A^T \tag{1}$$

Where, T_A is the Task Assignment matrix, T_D is the Task Dependencies matrix and $T_A{}^T$ is the transpose of the T_A matrix.

The result of applying the Eq. (1) on the matrices on Table 8 is as follows:

The C_R matrix needs to be compared with the Actual Coordination Matrix (C_A) in order to calculate the level of Congruence achieved in the Coordination Requirements matrix (C_R). In other words, Congruence will define the proportion of activities that have really taken place (C_A Matrix), in relation to the entire number of activities that should have occurred (C_R Matrix). For example, if C_R matrix shows that 8 pairs of software developers should coordinate with each other, and 4 of these show coordination activities in the C_A matrix, then the congruence obtained is 0.5. The definition of congruence can be expressed in [16] as follows:

$$\begin{aligned} Diff\,(C_R, C_A) &= card\{diff_{ij}\,|cr_{ij} > 0\,\&\, ca_{ij} > 0\} \\ |C_R| &= card\{cr_{ij} > 0\} \end{aligned}$$

The values of *i,j* should be different, as the matrix represents the same people in the rows as in the columns. The cells in the main diagonal should, therefore, be ignored as they represent the same person.

Where the *i* represents each column of the matrices, while the *j* represents each row. Together, *ij* represent each of the cells in the matrix. We thus have:

$$Congruence(C_R, C_A) = Diff\,(C_R, C_A)/|C_R| \tag{2}$$

The range of values of Congruence belongs to the [0.1] interval, which represents the portion of the requirements that have been satisfied through some type of coordination activity.

Given the Coordination Requirements matrix (C_R) in Table 10 (a) and the Actual Coordination matrix (C_A) of Table 10 (b) we have:

$$Congruence = \frac{2}{3} = 0.66$$

Table 11
Matrices with weights concerning task dependencies.

T_A		T_D		$T_A{}^T$		C_R
		0 0 3 0 0		0 0 1		
0 1 0 0 1	*	0 0 0 1 1	*	1 0 1	=	– 3 0
0 0 1 1 0		1 0 0 0 4		0 1 0		6 – 3
1 1 0 0 0		0 2 1 0 0		0 1 0		1 4 –
		0 0 0 2 0		1 0 0		

The previous examples seen in Portillo-Rodriguez [1], do not use weights, but in Cataldo et al. [16] whole numbers are used to represent weights in relation to the number of dependencies amongst tasks, see Table 11:

The Actual Coordination Matrix (C_A) has not been discussed at length until now. It is built with 4 different measures in [16], given that software engineers can coordinate and exchange information through several means of communication. The coordination activity is therefore represented by means of these matrices; this is conducted in different ways throughout the work associated with a modification request.

1. The first measure represents the *structural coordination*, which captures the potential communication and coordination activities that individuals belonging to the same working team may have through different means, such as meetings, interviews and similar mechanisms. The Actual Coordination Matrix is built to include a coordination activity, only if the *i* and *j* developers belong to the same team.
2. The second measure broadens the previous one and represents the *geographical coordination,* assuming that every software developer is in the same location in a developer centre and has the chance to communicate with others there, thanks to their physical proximity.
3. Communication through the Modification Request (MR): there is considered to be only one interaction between developers *i* and *j* when both of them comment explicitly in the modification request report.
4. Lastly, the IRC communication is calculated by considering IRC registers of interactions between developers. This creates a problem, since the resolution of an MR could take days to months. The IRC messages must be examined, so as to identify the interactions between developers that are relevant to the MR. Moreover, developers do not always make reference to the MR code.

In (1) and (2) the expected coordination is represented within the teams and locations. In (3) and (4) the communication and direct coordination are represented.

7.4.2. Kwan et al. proposal [2,19]

In Cataldo et al. [3], every single coordination need was considered to be equally important; weights were not used, thus producing a coarse-grained measure for socio-technical congruence. Taking that into account, in [2] it is understood that the measure proposed in [3] has limitations. The former authors suggest a different model for measuring congruence, though their model is also based on matrices [3]. It has resulted in a weighted measure which is fine-grained; it

Table 10
(a) Coordination Requirements Matrix (CR). (b). Actual Coordination Matrix (CA).

(a)

C_R	Person1	Person2	Person3	Person4
Person1	–		1	1
Person2		–		
Person3			–	1
Person4				–

(b)

C_A	Person1	Person2	Person3	Person4
Person1	–		1	1
Person2		–		
Person3			–	
Person4				–

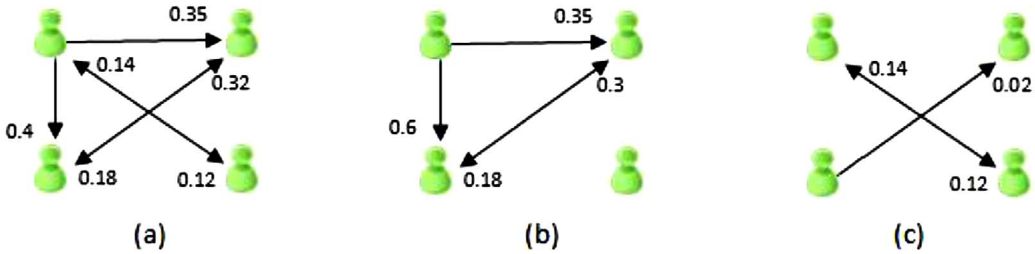

Fig. 8. According to Kwan et al. [2] (a) Coordination requirement matrix. (b) Actual coordination Matrix. (c) lack-of-coordination matrix.

allows gaps in coordination to be detected, locating them exactly, one by one, and prioritising them. The improved calculation of Congruence in [2,19] models the connections between people-tasks, tasks-tasks and people-people, applying weights to each relationship. This congruence measurement thus results in a measure that shows the strengths that each relationship has. Consequently, social interactions and organizational techniques are shown more accurately. It is worth underlining that this idea seeks to step away from underlying assumptions such as: that every dependency is equally important, or that any type of coordination satisfies coordination gaps.

In the measure proposed in [2,19] weighted edges between 0 and 1 are used in the Task Assignment Matrix (T_A), the Task Dependency Matrix (T_D) and the Real Coordination or the Coordination Requirements Matrix (C_R).

In the case of [2,19], just as in [16], the *Task Assignment matrix* (T_A) is a matrix *mxn*, where *m* represents the number of the selected people and *n* represents the selected tasks. It needs to be clarified that in [2,19], it is a "Weighted Task matrix", and every entry in the matrix describes the strength of the connection between the person *i* and the task *j*. In the case of the T_A matrix, for instance, every entry can show how many hours one person is expected to spend on a particular task. In order to set this value in the range [1–0], the number of hours expected for the task needs to be divided by the total number of hours that a person spends on that task. In [2,19]the importance of taking into account the experience a person has in a particular task is also mentioned, so that the amount of knowledge a person can bring to a task can be measured and shared.

The *Weighted Task Dependency matrix* (T_D) in [2] is equivalent to T_D in [16], but in this case, as in the previous one, every entry defines the strength of the relationships between the two tasks (it is a *n* x *n* matrix, where *n* and *n* are the selected tasks). Each matrix cell shows the dependency relationship that a task has in relation to another one, as regards the total number of dependencies that task has over the rest of them.

To calculate the Coordination Requirements Matrix (C_R), in [2] the calculation proposed by Cataldo et al. [16] is followed (Eq. (1)). Consequently, the calculation obtained from the people by people matrix shows how strong the relationships between the different developers are supposed to be.

In addition, the "*weighted actual coordination matrix*" (C_A) is required in order to obtain the final congruence coefficient, since the C_R matrix needs to be compared to the C_A matrix. To calculate the C_A weighted congruence, there are two proposals in [2]:

1. With respect to *communication*: a communication network can be weighted taking into account the ongoing communication. For example, let's contemplate dependencies between person A and person B:
   - If A communicates with B about their task dependencies, for each of the tasks they discuss the communication weight increases, "a" in "a*1/*n*", where "*n*" is the number of dependencies between A and B.
   - Given a three point scale, let's consider: ["Occasionally", "Frequently", "very frequently"], so that if A communicates with B about a task dependency "very frequently", then "A" is set to 1.

   For another task that is "frequently" discussed, "a" would be set to 0.66 and if o "occasionally" is the case, "a" would be 0.33.
   To sum up, if A has two interdependent tasks with B and they communicate "very frequently" in one task, and "occasionally" in the other, then the effective communication weight between A and B is 1.0*(1/2) + + 0.3*(1/2) = 0.66
2. With respect to the *distance*: we can measure the weight of the relationship between A and B as regards:
   - Distance, in terms of physical distance.
   - Difference of time between their time zones.
   - The number of organization units they belong to.

The values obtained must normalize in order to fit in within interval [0.1]. This results in a symmetric matrix, meaning that the distance between A and B is exactly the same as between B and A.

After obtaining $C_{R,}$ and once the T_A, T_D, and C_A matrices have been weighted, we can compare C_R with C_A to obtain the lack-of-coordination matrix, according to [19]. It is at this point where we find one of the most noticeable differences with the measure proposed in [16]. The following steps are carried out in [2] to compare C_R with C_A:

1. Create a people by people matrix (C_R) using the weighted method mentioned above.
2. Subtract the actual coordination matrix (C_a) from the coordination requirement matrix (C_R).
3. Remove values that are less than zero.

In the lack-of-coordination matrix Fig. 8(c) not only the gaps themselves are represented; their size can also be registered. The larger the value, the more severe the lack of coordination is between person *i* and person *j*.

In [19] the equation used to obtain the lack-of-coordination matrix is defined as follows:

$$G_{ij}(C_R, C_A) = C_{R\,ij} - C_{Aij} \tag{3}$$

Where G_{ij} is the size of the gap between the *ij* pair.

Unlike [19], (Fig. 9) we have observed in [2] (Fig. 8) that the resulting values in the lack-of-coordination matrix of less than 0 are taken into consideration, while before this the negative values were removed. This allows us to illustrate both the lack of coordination (positive values) and the excesses (negative values), considering excess as a greater quantity of coordination in relation to the coordination requirement matrix (C_R); this should not be a problem.

To calculate the weight index of socio-technical congruence by means of this matrix, all the cell weights must be added, and this must afterwards be divided by the total weight obtained after adding every weight of the Coordination Requirements matrix (C_R).The result must be subtracted from 1; thus the total congruence is obtained. This is formally defined in [19] as follows:

$$Congruence = 1 - \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} Gij}{\sum_{i=1}^{m}\sum_{j=1}^{m} CRij} \tag{4}$$

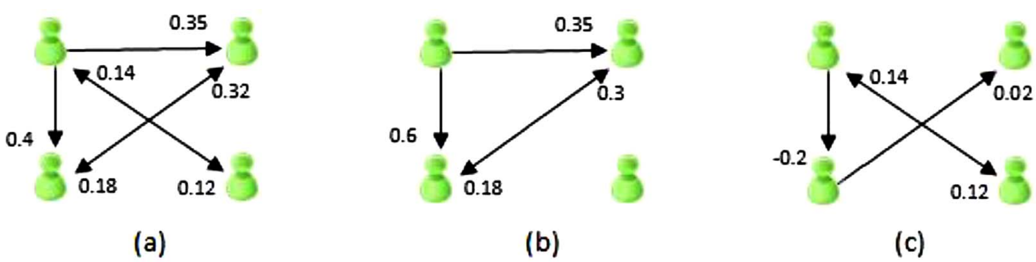

Fig. 9. According to Kwan et al. [19] (a) Coordination requirement matrix. (b) Actual coordination Matrix. (c) lack-of-coordination matrix.

7.4.3. Proposal for individualized STC by Wagstrom et al. [20]

Taking into consideration the studies by Cataldo et al. [3,16], in [20] and in relation to the contribution of the congruence of each person, the ISTC metric was created to obtain an individual measure of socio-technical congruence. To calculate ISTC, the difference between the actual coordination regarding the coordination requirements of an individual, *i,* is divided by the coordination requirements. This can be defined as follows:

$$ISTC_i = \frac{\sum (C_R[i,]^\wedge C_A[i,]) + \sum (C_R[,i]^\wedge C_A[,i])}{\sum C_R[i,] + \sum C_R[,i]} \quad (5)$$

The numerator of the Eq. (5) calculates the number of coordination requirements that individual *i* has accomplished, considering only the rows and columns of the matrix corresponding to *i.* The denominator sums up the coordination requirements that must be achieved by *i.*

The original STC formula [3,16] is also separated into 3 elements in [20] *matchComm (*it differentiates between actual coordination and coordination requirements), *coordReq (*coordination requirements), and *extraComm* (extra communication not reflected in the coordination requirements). These three elements can be individualized/personalized and are formally defined as seen below:

$$\begin{aligned} matchComm_i &= \sum (C_R[i,]^\wedge C_A[i,]) + \sum (C_R[,i]^\wedge C_A[,i]) \\ coordReq_i &= \sum C_R[i,] + \sum + C_R[,i] \\ extraComm_i &= \sum C_A[i,] + \sum C_A[,i] - matchComm_i \end{aligned}$$

This is intended to address possible large-scale project problems caused by participation in several projects —which can influence the coordination requirements obtained. It also makes it possible to distinguish between simple, normal communication within a team ($extraComm + + marchComm$gathers all the information) and the communication that is carried specifically out to resolve dependencies.

7.4.4. Proposal for three-dimensional STC measurement by Li et al. [24]

On the other hand, in [24] the authors argue that Coordination Needs of teams are based not only on task dependencies, but also on sharing information and knowledge, abilities, resources, etc. The concept of socio-technical congruence is thereby broadened, and two additional dimensions are incorporated to the measure proposed by [3,16]: Knowledge-dependent congruence and resource-dependent congruence. Congruence must be taken into account as follows:

1. Communication requires a sharing of abilities and knowledge among software developers.
2. Communication requires a sharing of resources, organization of meetings, and any other kind of relevant management.

Taking the previous aspects into consideration, the following coordination requirements, in addition to those given by Cataldo et al. [16], are proposed:

- *Knowledge-dependent congruence*: this can be calculated by using the "knowledge-dependent" matrix (CR_K). This matrix can be defined in the following equation:

$$CR_K = TA^*(KT)^{T*}(AK)^T \quad (6)$$

- Resource-dependent congruence: this can be calculated by using the "resource-dependent" matrix. This matrix can be defined in the following equation:

$$CR_R = TA^*(RT)^{T*}(AR)^T \quad (7)$$

The two new levels of congruence introduced in [24] that take the previous matrices as a basis are defined as follows:

$$Congruence(CR_K, C_A) = 1 - \frac{diff(CR_K, C_A)}{|CR_K|} \quad (8)$$

$$Congruence(CR_R, C_A) = 1 - \frac{diff(CR_R, C_A)}{|CR_R|} \quad (9)$$

Furthermore, in [24] the previous equations are reformulated so as to include a time parameter (t), since software development is a changing process. The parameter t can represent days, weeks, months, or any other time unit that is appropriate in the context of measurement.

The meaning of symbolism is presented as follows:

- ***TA:*** Task Assignment matrix (people-task).
- ***CR_K:*** People by people matrix indicates the degree of communication that there must be between each pair of CR_{ij}, according to the *knowledge required* to complete the assigned tasks. (The subscript K indicates knowledge).
- ***CR_R:*** People by people matrix indicates the degree of communication that there must be between each pair of $CR_{ij,}$ according to *shared resources required* to complete the assigned tasks. (The subscript R indicates resources).
- ***KT:*** Knowledge Requirement Matrix: contains information about the knowledge/skills required to complete a task. Knowledge-Task.
- ***$(KT)^T$:*** is the transpose of *KT.*
- ***AK:*** Knowledge Matrix: contains information about who knows what.
- ***$(AK)^T$:*** is the transpose of *AK.*
- ***RT:*** Resource Requirement task matrix: contains information about what resources are required to complete a specific task. Resource-Task.
- ***$(RT)^T$:*** is the transpose of *RT.*
- ***AR:*** Resource Use matrix: contains information about who uses what resources.
- ***$(AR)^T$:*** is the transpose of *AR.*
- ***C_A:*** Actual Coordination matrix. Real coordination of the organization.

7.4.5. Portillo-Rodríguez proposal aimed at global software engineering [1]

Portillo-Rodríguez [1] proposed a multi-agent architecture designed to manage socio-technical congruence (measure it, control it and maintain it), which is targeted at Global Software Development (GSD). According to Portillo, the congruence measurement needs to be adapted to this environment needs. In this case, an adaptation of the measure proposed by Kwan et al. [2] was developed, consisting in the inclusion of a range of factors that affect the way in which globally distributed team members perform coordination. These factors have been included as weights in order to determine the STC of each user.

Unlike in Kwan's paper, in this piece of work weights are used as tasks in the following factors:

- *Socio-Cultural Distance (SCD)*: This factor is calculated based on the native countries of the development team members, also using a table, as presented in [26], in which 50 countries are rated in dimensions that take into consideration socio-cultural factors demonstrated by the particular values of each country. Interactions are considered to be most important when they are performed by members coming from countries with a low SCD. Congruence will thus be increased if interactions are performed by this type of users.
- *Temporal Distance (TD)*: this factor influences temporal distance; in other words, it is measured as the difference in time between locations of the working team. In this case, it is assumed that team members who are separated in time must communicate more if they are to reach a good level of coordination. Hence, interactions performed by users with a higher TD are weightier in the congruence measure.
- *Geographical Distance (GD)*: This last factor takes into account the physical distance, and it is used in a similar way to the previous factors.

It was reported that although these factors are used *in a similar way*, they need to be taken as a whole, since a high SCD does not necessarily mean a high TD or GD. For instance, Spain and Argentina are separated by a high TD and GD, but the SCD is low, due to the fact that they share the language. In addition, other factors that take into consideration the priority of tasks have been incorporated. It is considered that urgent tasks must be tackled with a higher interaction.

Portillo-Rodríguez extended the scope of the congruence measurement to embrace not only the development stage, but every stage of a project. This highlights another clear difference when we compare it to Kwan's approach.

7.4.6. Valetto et al. proposal. [27]

In [27] a quantitative measure of socio-technical congruence was proposed. It uses social networks made up of the information extracted from software repositories. In particular, they used a combination of three types of information repositories:

1. *Communication/collaboration interactions among stakeholders*: these can be defined as undirected arcs between nodes lying on plane P. Set P contains the stakeholders (see Fig. 10).
2. *Inter-relationships between artefacts*: this can be defined as the directed arcs between artefacts with interdependency, which is plane S. Set S contains software artefacts.
3. *Work relationships*: this can be defined as the directed arcs from plane P until plane S. This indicates work performed by stakeholders in the software artefacts. This arc set will be called J.

In [27] it is indicated that a social network, like the one in Fig. 11, can be built using the information contained in the software repositories. Both nodes of planes P and S and their connections can easily be extracted with metadata information, as can be done with stakeholders' IDs,— those who conduct the commit—the modified artefacts, date and time of commit, etc. The dependency extraction between software artefacts can be obtained by means of processing source code files in order to get function invocation, inheritance, etc.

The possibility of adding weight to the arcs was also proposed (e.g. the weight could be the number of changes in the artefacts by the same stakeholder, or the number of dependencies of a source file on another). The measure proposed in [27] is developed as follows:

The set $G_P = (P, E_P)$ denotes the digraph of people and their relationships, where P is the node set and E_P is the arc set (each arc represents a relationship between pairs of developers in set P). For example, the arc (i,j) belonging to E_P may mean that developers i and j have communication, where $i, j \in P$. In this case, undirected edges are used, because we assume that the communication relationships between people are reciprocal.

On the other hand, set $G_S = (S, A_S)$ denotes the digraph representing software artefacts (S) and their relationships (A_S). The arc's direction indicates the dependency between artefacts.

By means of J, connections between developers of set P and software artefacts of set S are denoted, with a directed arc.

There are two aspects used to reflect the congruence proportion. The first aspect is obtained by dividing the number of arcs actually mirrored by the number of arcs in A_S that could be mirrored in E_P (Fig. 10(a)). Another aspect of congruence can be referred to as "node tie", that is, when two distinct developers have some relationship with the same artefact (Fig. 10(b)).

Relationships between software artefacts (Arcs in A_S), which are reflected in E_P must be taken into account, in order to obtain the proportion of arc mirroring. This facilitates the display of the mirror by the Union of set J. For instance, if (i,l) is an arc belonging to the A_S set, and (k,i) and (h,l) are arcs of the set J, then the arc (i,l) is an "arc mirrored" if the following conditions are true:

1. k and h are distinct nodes and belong to P
2. Either (k,h) or (h,k) or both, belong to E_P

Congruence based on arc mirroring [27] between graphs $G_P = (P, E_P)$ and $G_s = (S, A_S)$ joined to set J, is given by:

$$C(G_p, G_s, J) = k/\gamma \text{ when } \gamma > 0$$

Where k and γ are computed by the following algorithm, which detects and counts arc mirror patterns:

In [27] an example for the mirroring arcs proportion for Algorithm 1 by 20% is shown. This is obtained by counting six arcs

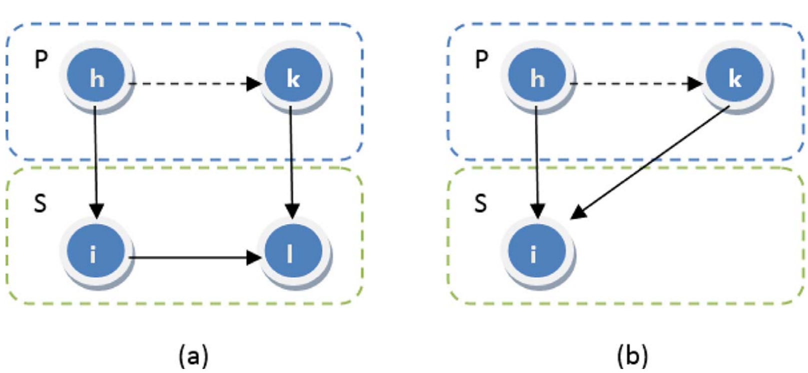

Fig. 10. (a) Arc mirroring. (b) Node tie.

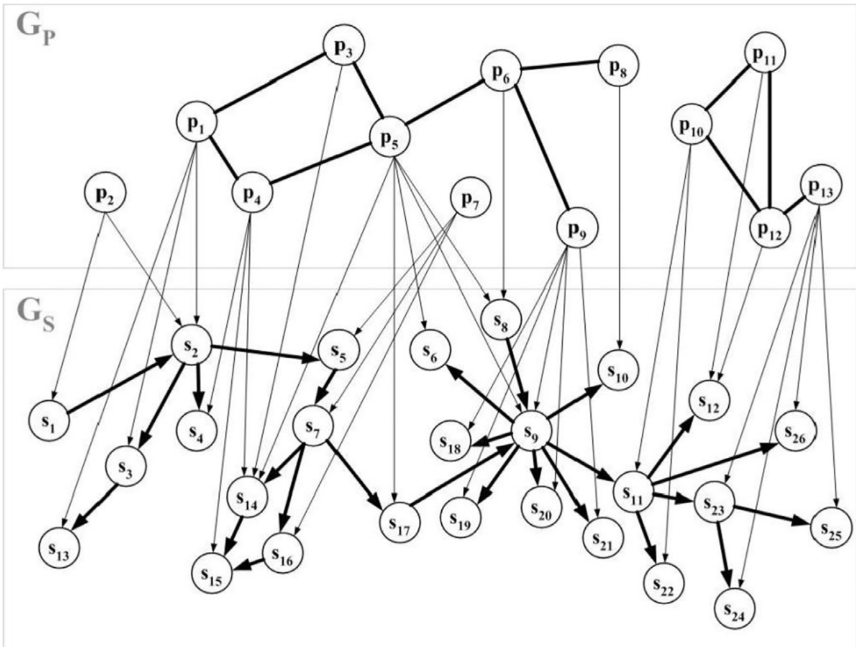

Fig. 11. Socio-Technical software network [27].

Algorithm 1
Calculation of k and γ.

```
step 1:
      set k, γ = 0
step 2:
      For each arc (i,l) in A_S:
   step 2a:
         let J_i = all arcs in J incident on node i;
         let J_j = all arcs in J incident on node j.
   step 2b:
         For each (a,i) in J_i and (b,l) in J_j:
            if a ≠ b, then γ = γ + 1.
            if (a, b) ∈ E_P, then k = k + 1.
```

reflected in A_S:

(s2, s4) by (p1, p4); (s8, s9) by (p5, p6), (s8, s9) by (p6, p9);
(s11, s12) by (p10, p11); (s11, s12) by (p10, p12);
(s14, s15) by (p4, p5)

There are 24 cases where there is a mirroring missing. For example, for arc (s2, s5) there is no mirroring in (p1, p7). Then we have:

$k = 6,\ \gamma = 30;\ C(G_p, G_s, J) = 6/30 = 0.2$

As regards the "node tie" proportion, in Fig. 11, we have 57%. Each node s8, s12 and s14 have developers which are related to each other, while nodes s2, s9 and s14 have at least one pair of developers which are not.

Cases with node (4): s8 (p5,p6); s12 (p11,p12); s14(p4,p5), s14(p3,p5)

Node cases with more than one developer (7): s2, s8, s9, s12, s14(x3)

$C = 4/7 = 0.57$

It is important to mention that both ways of measuring congruence can be used on a global scale (over the whole network), or on a local scale. Similarly, both measurements may be used in combination or separately.

7.4.7. Comparing the different measures

In order to the way in which the differences among the measures previously described are shown we have created Table 12. The features considered to compare them are: the approach used to measure the socio-technical congruence, such as the use of matrices, whether or not the measure is weighted, or whether the measure can be used in global software development settings. If the measure has been empirically validated, then its main features and main challenges are also summarised. Some authors such as Cataldo and Kwan propose two versions of the same measure, both of which have been included in the same row.

It will be noted that only the proposals by Portillo and Kwan et al are appropriate for use in a Global Software Development setting. However, the first uses some surveys to obtain the information, and this could be disturbing. On the other hand, only Valeto et al use Social networks, while all the previous one are based on matrices. Six out of eight of the measures are weighted.

7.5. RQ5. What are the pros and cons of different ways of measuring socio-technical congruence?

To respond to our research question RQ5, the measures set out have been analyzed, in pursuit of the most noticeable advantages and drawbacks. In Cataldo et al. [3,16] the first measure that allows us to observe the total congruence of an organization was proposed. However, it has deficiencies when defining priority, location and strength of coordination gaps, all of which results in a coarse grain perception that is rather imprecise. In this measure it is implied that any type of communication is enough to resolve a coordination problem, and also that any incidence has the same priority.

The weighted measure proposed in Kwan et al. [2,19] allows many of the defects mentioned above to be corrected. This is a more fine-grained measure that permits the relationships between people-tasks, tasks-tasks and people-people to be investigated in a much more detailed way. In [2], it is pointed out that, thanks to the weighted congruence measurement presented in their paper, the communication gaps can be located individually, thus allowing better action to be taken on these gaps. The priority of different tasks, along with coordination dependencies, is also assessed. This all means that we can detect lack of discussion between two people working on related tasks; we can also find tasks must be addressed first according to the strength of the relationships.

Table 12
Summary of STC measurements.

Measure	Approach	Weighted	Suitable for Global software development?	Empirical Validation	Main feature	Limitation
Cataldo et al. [3,16] (2006–2008)	Matrix	No [3] Yes [16]	None of them	Yes [3] Yes [16]	This is the first measure proposed	Low accuracy
Kwan et al. [2,19] **(2009–2011)**	Matrix	Yes both of them	No [2] Yes [19]	No [2] Yes [19]	Introduces fine granularity to detect coordination gaps in a more precise manner	Cultural Distance is not taken into account
Wagstrom [20] **(2010)**	Matrix	Yes	No	Yes	Proposal for individualized STC	It calculates the STC of one person but not the global STC
Li [24] **(2012)**	Matrix	No	No	Yes [24]	It provided two new STC levels: Knowledge and resource dependent congruence	The validation was only conducted with students
Portillo-Rodríguez [1] **(2013)**	Matrix	Yes	Yes	Yes [1]	It measure was defined for GSE	Tedious for the user owing to the use of surveys.
Valetto et al. [27] **(2007)**	Social networks	Yes	No	Yes [25]	It uses repositories to obtain the information	Few information sources are considered in this measure

Li et al. [24] take Cataldo et al's proposal as a starting point, without including what was proposed by Kwan et al., so it has the same deficiencies as Cataldo's proposal has. However, Li et al. propose two extra dimensions for measuring congruence, based on the awareness of who has what resources, knowledge and skills needed to complete tasks.

The measure proposed by Valetto et al. in [27] uses only the information gathered in the repositories, so that a great amount of data that influences consistency is left without being assessed. On the other hand, Ehrlich et al., in [25] propose to obtain more information (for instance about the different communication means that they used, meetings that they had, etc) by using surveys.

Blincoe et al. state in [28] that these measures have significant problems in detecting the coordination requirements because they do not carry out the measurement in real time, since this data is usually available at the end of each task. They therefore propose an algorithm called "proximity", to calculate the coordination requirements "live", in an attempt to mitigate this problem.

We can conclude that in both Kwan et al. [19] and [1] Portillo-Rodríguez the measurement of STC improves as a consequence of including physical distance, time distance and socio-cultural distance when measuring STC.

To sumarize we can state that the more information from different sources that is used in order to calculate the STC, the more exact the result will be. However, obtaining information often implies a cost. The ideal situation would be to extract the information automatically without bothering the users. However, if some particular information has to be obtained by hand by, for instance, using surveys, it would perhaps be better to consider whether this is worthwhile, as the profit could be less than the employees' extra workload.

It is surprising that the empirical studies do not mention anything about the effort/cost of obtaining the information. In fact, this is a gap that we have detected, as we shall discuss later.

7.6. RQ6. In what areas can socio-technical congruence be applied?

The measures mentioned above can be used in different tasks in the quest to raise awareness about the proper functioning of the projects as regards coordination and communication. Not all of the measures described in Section 7.4 are suitable for the tasks; for instance: one interesting use would be to observe what the most relevant communication gaps are as regards the need to take action in due time, by modifying responsibilities or distributing tasks. Some measures, such as those proposed by Kwan et al. [2], do detect this gaps on an individual level; in the Catalado et al. [3] proposal, nevertheless, they do not.

It could be worthwhile for a manager to observe the alignment existing between the project team coordination and the technical dependencies [2], as well as to visualize the time evolution of this alignment. Similarly, it can be useful to observe the temporal evolution of socio-technical congruence with a view to conducting audits in the organization.

In studies such as [3,17] it has been confirmed that individuals with a higher socio-technical congruence level generally reach a higher performance. Consequently, another use of congruence measurement could be to detect performance deficit, in order to act on this early on.

In Valeto et al. [27] the authors argue that these measures can be made available to customers so that they can manage the software development process better.

Certainly, one of the most important measures of socio-technical congruence is to raise awareness of developers, ideally in real time, about the teammates with whom they must communicate as regards coordinating tasks.

7.7. RQ7. What kind of available tools help to measure socio-technical congruence?

A summary of the uses and functions of the tools found during our mapping study is provided below:

- *Ariadne [9,29,30]*: this tool is an Eclipse plugin that serves to support the development of distributed software by allowing the socio-technical interdependences between developers and software artefacts to be visualized. Ariadne obtains the dependencies by analyzing the source code (Java) using the configuration management repositories associated with the projects. Ariadne boosts communication, since it enables the visualization of social dependencies among developers. Project managers can also use it as a control tool with which to distribute responsibilities and detect whether certain project members are overloaded with tasks. Ariadne shows by default a directed graph in which the nodes are the authors, and the edges are the dependencies among them.
- *CodeBook [31,32]*: Codebook is a tool that discovers the transitive relationships among people, code test cases, specifications and any type of artefact that is related to a repository and can be extracted by means of data mining. It makes it possible to increase awareness of the changes undergone by artefacts over time and to discover who uses the code, and whether there is already a similar project in one's organization, or to investigate the root of a problem to be resolved.
- *Fonseca* et al. *[10]:* the objective of this tool is to automatically identify the faults between dependency networks and communication networks. This is done by exploring the relationship between the software dependencies and the coordination among developers. With regard to the software dependencies, it uses Ariadne to automatically identify dependencies among code artefacts and their authors, thus making it possible to create a social network of developers in which the dependencies among them is shown. As far as communication is concerned, the tool analyzes the flow of e-mails between developers.
- *Palantir [33,34]*: this supervises the artefacts that are being modified by the developers and reports on the state of said artefacts, with the aim of avoiding conflicts resulting from simultaneous changes and a lack of coordination. It thus allows the early detection and reduction of conflicts.
- *ProxiScientia [35]:* this compiles data from the developers' work environment, in order to calculate coordination requirements using 'proximity' [28]. It presents each developer with visualizations appropriate to his or her coordination needs in order to increase awareness of other developers with whom it is necessary to communicate, or of tasks that must be carried out.
- *RaisAware [36]*: this identifies those files that have most dependencies in a project by means of analyzing a matrix containing information about the dependencies in the files. The aim of this characteristic is to provide awareness of the extent to which changes in these files may affect other developers' work. This tool also enables the user to visualize the files that will be affected by the changes made. RaisAware displays the names of the developers in place of the file names, such that it is possible to visualize which developers are affected and with which of them it will probably be necessary to coordinate.
- *Tesseract [37]***:** this tool makes it possible to explore the socio-technical relationships between project artefacts and developers, along with the interactive exploration of these relationships over time. Tesseract analyzes the different entities of a project in order to determine its socio-technical relationships and show these, thus allowing a simple exploration of its data. This tool also calculates socio-technical congruence using a method proposed by Cataldo et al.
- *Travis [38]*: This is a tool that provides visualization and analysis for traceability networks (networks whose entities are related to users, tasks and artefacts), created from the dependencies between artefacts and users. Travis is able to use these networks to show the traceability of connections originating from artefacts, tasks and individual users, with the aim of increasing the awareness of the developers working on the project.
- *WorldView [39]***:** this tool is oriented toward supporting distributed software development. It provides a visualization of the structure of distributed teams by means of a central repository. The tool makes it possible to observe the dependencies among the project components, which are represented by coloured lines of varying thicknesses according to the quantity of dependencies or artefacts shared, along with the developers' roles and knowledge. It also makes it possible to discover the geographical location and availability (bank holidays, time zones, etc.) of the various components of the team by means of a world map metaphor.
- *STCM (Socio-Technical Congruence Manager) [40]*: this makes it possible to measure and observe the evolution of socio-technical congruence over time at both a sprint and a global level. The application also reports on the unresolved deficiencies detected and informs the user about them, providing advice as to how to act in order to close possible gaps, improve times at which to communicate with colleagues, the preferred means of communication, etc. STCM is based on a multi-agent architecture that autonomously measures and detects coordination problems and informs the person responsible for the project about them. It calculates congruence by employing a modification of the means proposed by Kwan et al. [2].
- *STCML [41]*: this is not a tool, but we wished to include it because it is an XML-based language that can assist in the construction of future tools. It serves to support the modelling of socio-technical aspects such that it is possible to specify the relationships between architectonic objects and social processes. The basic types of language are 'agents', 'resources', 'tasks', 'teams', 'graphs' and 'networks', which provide an idea of its possible application.

Table 13 provides a summary of the main features of the tools. We therefore indicate which tools take the social and technical dependencies into account, whether they are able to detect gap or coordination, whether the tool calculates the STC and the use of the tool.

Only three tools detect the coordination gaps and only two the STC measure. STCM *[40]* is still a prototype that has to date been used in only two case studies, but it has a measure that is adapted solely to GSD. Furthermore, Tesseract has could possibly be used to measure the STC, but only by using Cataldo's measure which is the first measure to have recently been improved in other proposals.

7.8. RQ8. What case studies have been published about socio-technical congruence?

In order to respond to our research question RQ8, we have analyzed all the case studies found, seeking to establish 3 groups according to the environment in which they were carried out: business environment, university environment and open source environment.

- *Effects on the quality of STC software*: Studies such as Šmite and Galviņa [23] refer to coordination problems that may occur in organizations as a result of the absence of socio-technical congruence. Among the most serious problems are delays in presenting the project because of components that have yet to be developed and which are discovered late, or the bad integration among the parts already developed by different teams, thus supposing an additional cost in terms of time and money. This is confirmed by Ehrlichet al. in [25], in which a comparison is made between the quantity of gaps in coordination and output, and which verifies that there is a correlation between these two factors: the more gaps, the more the number of increases in code changes. In [3,16,17] it is noted that when the highest values of socio-technical congruence are obtained,

Table 13
Summary about tools.

Tool	Social dependencies	Technical dependencies	Gap detection	Is the STC measured?	Utility
Ariadne *[9,29,30]*	Yes	Yes	No	No	It enables the visualization of social dependencies among developers
CodeBook *[31,32]*	Yes	Yes	No	No	It increases awareness of the changes undergone by artefacts
Fonseca **et al.** *[10]*	Yes	Yes	Yes	No	It identifies the faults between dependency networks and communication networks
Palantir *[33,34]*	No	No	No	No	It allows the detection and reduction of conflicts in artifacts
ProxiScientia *[35]*	Yes	Yes	Yes	No	It increases awareness of other developers with whom it is necessary to communicate
RaisAware *[36]*	Yes	Yes	No	No	It enables the user to visualize the files that will be affected by the changes made.
Tesseract *[37]*	Yes	Yes	No	Yes	It analyzes the different entities and shows its socio-technical relationships. it also calculates socio-technical congruence
Travis *[38]*	Yes	Yes	No	No	It increases awareness about artefacts, tasks and individual users
WorldView *[39]*	Yes	Yes	No	No	It supports the development of distributed software
STCM *[40]*	Yes	Yes	Yes	Yes	It makes it possible to measure and observe the evolution of socio-technical congruence over time

it is possible to reduce the number of software failures. However, Marczak et al. [42] detect gaps in communication and low levels of STC that do not affect the project, which is presented on time and correctly. In Kwan et al. [19] a high level of socio-technical congruence is related to a greater probability of success as regards the results of the project.

- *Effects on STC software development time:* Cataldo et al. [3,17] provide analyses of the relationship between congruence and the speed at which tasks are developed. The results show that when communication among people is aligned with coordination requirements, the development time improves. Or, in other words, socio-technical congruence helps reduce the time needed to complete the tasks. It was also observed that those workers with the most congruence were also the most productive. However, the author of [19] discovered cases in which a high congruence in projects that are formed of too many teams may be counter-productive, owing to the overload of coordination recommendations received by the developers.
- *Effects of STC on distributed development*: Ehrlich et al. in [25] verify that there is a considerable increase in the quantity of gaps in coordination in distributed pairs, and that socio-technical congruence therefore becomes of particular interest in this type of development environments. However, Bird et al. in [43] verify that the good structure of the organization in which the study was carried out allows the detection of 6% in the gaps in coordination of distributed teams when compared to local teams. Cataldo et al. in [3,16,17] noted that congruence also has positive effects on the time needed to resolve tasks in distributed environments.
- *Mitigation of the lack of coordination using Brokers*: Kwan and Damian, and Ehrlich and colleagues in [18,25] also show interest in the figure of the broker as regards reducing gaps. A broker is a person who acts as an intermediary and who boosts the improvement in communication and collaboration. It is in the context of distributed development that brokers are most important since, as is demonstrated in [25], they can mitigate the effects of the increase in gaps between distributed pairs by acting as intermediaries between teams in order to distribute information. Those pairs that include a broker are also more unlikely to suffer gaps in coordination. It was observed, moreover, that brokers tend to be people with roles that require a broad perspective (project leaders, technical leaders, etc.).
- *Error prediction:* Bird et al. in [43] contains a study on how human factors affect software development. This is done using a socio-technical network that, by means of dependencies among modules, allows us to predict failures that may occur in those modules. The human factors studied (number of developers working on a file, quantity of those developers who left the company before the project was completed, etc.) were linked to the software quality (e.g. the loss of engineers working on a module leads to a loss of knowledge and thus a loss in the quality of that module) and it was discovered that these attributes mean that a better predictor of the failures is obtained than when analyzing the software attributes.
- *Improvements to STC using tools*: Portillo-Rodríugez. [1] analyzes the use of a tool, STCM, employed in two distributed development projects, which allows socio-technical congruence to be analyzed. Two studies were carried out during which the members of the projects being studied were interviewed in order to discover their opinions on the use of the tool. The developers revealed that the tool speeds up communication and coordination in distributed environments and is a good source of centralized data that makes it possible to detect which users are having coordination problems. Those interviewed also stated that the tool is useful, particularly when they begin working on a task, A, that does not require coordination and then with the passage of time another task, B, must be carried out, and is one which has to be coordinated with task A. The person responsible for A is not, however, informed of the need to coordinate with B, thus leading to gaps in coordination and delays in the project. STCM manages to solve this problem by informing those responsible for A and B of the need for coordination. Another person interviewed commented that the tool helps avoid problems of duplicated work, since it immediately gets two people in touch with each other who have started work on the same task without realizing it. They stated that the tool speeds up coordination work for the project leader, and that the graphics permit communication requirements to be detected efficiently so that they can be resolved. The use of this type of tool to control congruence was generally validated.
- *Socio-technical congruence in academic environments*: It is common for universities to set up large software development projects to study the behaviour of students and to teach them to coordinate themselves in groups. MacKellar [44] measured the temporal evolution of the individual socio-technical congruence of 25 students during a semester. They followed their progress for 6 weeks and determined that the students' congruence did not improve, unlike what occurred in [3] with professional teams. The most reasonable explanation for this is that the students did not know how to use the version control systems correctly, which limited the way in which they calculated

task dependencies and led to a loss of fundamental information. In Li et al. [24], however, the authors analyzed the behaviour of 7 teams of students during a semester, and did observe an increase in congruence. They also observed that the teams with high levels of congruence had better-organized processes and a significant increase in quality with regard to the rest.

- *Socio-technical congruence in open source environments*: it may be interesting to observe the behaviour of socio-technical congruence in completely different environments, such as open source environments, in which almost all communication is carried out by means of repositories, which implies that no information is lost, i.e., in meetings, as may occur in the environment of an organization. In [11,12,14]the evolution of socio-technical congruence of open source operative system development communities is analyzed, after which results were obtained over time that demonstrated a high socio-technical congruence of between 70% and 80% according to the stable version; this level grew gradually in the later version. Wagstrom et al. in [20] noted similarities between the benefits provided by STC in organizations and free software communities. In Syeed et al. [13], meanwhile, an analysis of the behaviour of congruence in an open source software ecosystem (a set of projects that evolve together, but for which, in order to measure congruence the projects must also have technical dependencies between each other) was carried out, and it was observed that there was not generally a high level of congruence. In Bolici et al. [22] evaluated STC, after which they discovered that development work in free software teams is generally highly individual, which means that there is a great lack of discursive communication (communication usually implicitly takes place by means of repositories). This variety of results in the open source environment allows us to conclude that more studies on STC in the area are required, along with more refined metrics for this type of development environment.

8. Conclusions

The objective of this research was to obtain all the existing material in the literature on socio-technical congruence as we needed to develop a tool to calculate STC in a software factory. When starting this task we realised that we had found very little information about it and that found was not very complete.

After analysing the data, we can conclude that:

- There is no standard measure of socio-technical congruence, although most of the measures that exist consider the proposal by Cataldo et al. [3,16] and make adaptations and improvements to it as regards the environment on which it will be focused.
- Only one author describes the STC properties in detail, so more research is needed in this topic in order to obtain feedback from more researchers.
- The proposals with which to measure STC in GSD settings could be improved as the Portillo et al. proposal needs survey information, which could overload employees, and one of the Kwan et al. measures did not consider the socio-cultural distance, which is quite an important factor in GSD conflicts as literature on the topic show. This is, therefore, another aspect that could be researched in more detail.
- There are also gaps in open source environment studies, since there is very few research on this topic and, surprisingly, the results found are very different in each of them. More investigation on the topic would help to evaluate the advantages or disadvantages of using STC in this environment.
- Moreover, only a few papers mention the risks of excessively overloading users with coordination iterations when controlling STC. In fact, no case study examining these risks and their effect on developers' productivity has been found. This is also an important topic to explore since it would be interesting to know under what circumstances (for instance number of employees, sites, technical dependencies, etc.) it is convenient to calculate the STC regardless the cost of obtaining the information.
- There is no range established to indicate when socio-technical congruence could be considered high or low, or how it affects the working model of each organization. This might be an important aspect to describe when case studies are conducted in the future.
- With regard to tools, only STCM and Tesseract measure socio-technical congruence. Of these, STCM focuses on managing congruence itself in an effort to proactively assist in the detection of gaps in coordination, and to propose solutions as regards maintaining the levels of optimised congruence. Tesseract uses the Cataldos et al. measure, which could be poor accuracy. The other tools generally serve to detect dependencies, gaps in communication and relationships that should be managed by users manually. There are also some tools that can provide information about awareness of the actions performed by the user. However, there is no tool with which to calculate STC and trigger a warning when a gap in communication is detected, attempt to guess the coordination problem that could is occurring at that moment and propose possible solutions. A tool with this feature would be an important contribution to this field, and we are currently working on this objective.

Nevertheless, it is important to highlight the benefits of having studied the existing tools, given that this endeavour has allowed us to obtain ideas that can be borne in mind during the design of tools, thus providing possible options that may make those tools easier to use.

The small number of studies found on STC, together with the research gaps we have pointed out, suggest that further research into socio-technical congruence is required. We thus believe that this paper may be an important contribution for researchers who have begun to work on this issue, since data on the forums in which papers have been published are provided, along with a classification of the types of studies that can be found and the countries in which STC is most frequently studied, in addition to an overview of the subject and a comparison of measures and tools.

We therefore consider that this paper is also of interest to companies, as it presents not only different ways of measuring socio-technical congruence, but also tools and prototypes that an help to measure it and a comparison of them.

Acknowledgments

The authors would like to thank Félix Oscar Garcia Rubio, Lourdes Sánchez Sevillano and Damiano Torre for their collaboration in this paper.

This work has been funded by the project LPS-BIGGER: Línea de productos Software para BiG Data a partir de aplicaciones innovadores en entornos reales (Ref.: UCTR150175), is framed under the Strategic Program CIEN, and itisco-funded by Centro para el Desarrollo Tecnológico Industrial (CDTI), and “Fondo Europeo de Desarrollo Regional (FEDER). Moreover, this work is also funded by the TEAM SPACE (Desarrollo y Automatización de la Plataforma Tecnológica de Gestión de Equipos Globales basada en Cloud Computing), company STEELMOOD, supported by CDTI (Centro para el Desarrollo Tecnológico del Ministerio de Economía y Competitividad and by the GINSENG project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2015-70259-C2-1-R). It is additionally supported by GLOBALIA (PEII11-0291-5274) and G3Soft, Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha and Fondo Europeo de Desarrollo Regional FEDER. GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01).

Appendix A. Complete list of all primary studies included in the study.

Ref.	Source	Type	Year	Publication
[9]	ACM	Tool	2005	OOPSLA
[3]	ACM	C.S. Business environment and STC Measure based on matrices	2006	CSCW
[10]	IEEE	Tool	2006	ICGSE
[30]	Scopus	Tool	2007	GROUP
[38]	Scopus	Tool	2007	SEAFOOD
[27]	IEEE	STC Measure based on social network	2007	ICSE
[16]	ACM	C.S. Business environment and STC Measure based on matrices	2008	ESEM
[25]	SB	C.S. Business environment	2008	ICSE
[39]	ACM	Tool	2008	ICSE
[29]	IEEE	Tool	2008	VL/HCC
[45]	SB	Tool	2008	SIGSOFT
[8]	SB	Others	2008	ICSE
[21]	SB	Others	2008	ICSE
[42]	IEEE	C.S. Business environment	2009	CIRCUS
[22]	SB	C.S. Open Source environment	2009	ICSE
[31]	IEEE	Tool	2009	ICSE
[37]	IEEE	Tool	2009	ICSE
[34]	IEEE	Tool	2009	IEEE Software
[2]	SB	STC Measure based on matrices	2009	ICSE
[20]	SB	C.S. Open Source environment and STC Measure based on matrices	2010	Academy of Management Proceedings
[32]	IEEE	Tool	2010	ICSE
[36]	IEEE	Tool	2010	CollaborateCom
[43]	ACM	C.S. Business environment	2011	CSCW
[18]	ACM	C.S. Business environment and STC Measure based on matrices	2011	ESEC/FSE
[19]	IEEE	C.S. Business environment and STC Measure based on matrices	2011	IEEE Transactions on Software Engineering
[41]	ACM	Tool	2011	ICSE
[23]	Scopus	C.S. Business environment	2012	PROFES
[24]	IEEE	C.S. Academic environment and STC Measure based on matrices	2012	ICSSP
[33]	IEEE	Tool	2012	IEEE Transactions on Software Engineering
[35]	IEEE	Tool	2012	CHASE
[28]	Scopus	Others	2012	CSCW
[17]	IEEE	C.S. Business environment	2013	IEEE Transactions on Software Engineering
[1]	SB	C.S. Business environment(x2) and STC Measure based on matrices	2013	University of Castilla-La Mancha
[44]	IEEE	C.S. Academic environment	2013	CSEE&T
[4]	IEEE	Others	2013	RESER
[12]	SB	C.S. Open Source environment	2013	OSS
[14]	SB	C.S. Open Source environment	2014	Journal of Software
[13]	ACM	C.S. Open Source environment	2014	OpenSym
[40]	Scopus	Tool	2014	Information Sciences
[11]	SB	C.S. Open Source environment and STC Measure based on matrices	2015	Tampere University of Technology

**SB = Snowballing

Appendix B. Search strings.

Source	Search string
IEEE	*((.QT.technicaldependencies.QT.* ***OR*** *.QT.technicaldependency.QT.)* ***OR*** *(.QT.socialdependencies.QT.* ***OR*** *.QT.socialdependency.QT.)* ***OR*** *(.QT.softwaredependencies.QT.* ***OR*** *.QT.softwaredependency.QT.)* ***OR*** *(.QT.socio technical relationship.QT.)* ***OR*** *(.QT.socio technical congruence.QT.* ***OR*** *.QT.socio-technical congruence.QT.* ***OR*** *.QT.sociotechnicalcongruence.QT.))* ***AND*** *(.QT.coordination.QT.)*
ACM	*"**query**": {**content.ftsec:**(+ + "coordination")* ***AND*** *("software dependencies" "software dependency" "technical dependencies" "technical dependency" "social dependencies" "social dependency" "socio technical relationship" "socio technical congruence" "socio-technical congruence" "sociotechnical congruence")}* *"**filter**": {"**publicationYear**":{"gte":2000, "lte":2016}},* *{owners.owner = ACM}*
Scopus	*ALL(("technical dependencies"* ***or*** *"technical dependency")* ***or*** *("social dependencies"* ***or*** *"social dependency")* ***or*** *("software dependencies"* ***or*** *"software dependency")* ***or*** *"socio technical relationship"* ***or*** *("socio technical congruence"* ***or*** *"socio-technical congruence"* ***or*** *"sociotechnical congruence")* ***and*** *"**coordination**")*

Appendix C. Data extraction form

Information
Id Paper
Title
Reference
Abstract
Extraction
STC Definition
Tools
Metrics
Classification
Tool
Case Study – Academic environment
Case Study – Business environment
Case Study – Open source environment
STC Measure based on matrices
STC Measure based on social network
Others
Secondary Searches (Identified References)
Ref1
Ref2
…

References

[1] J. Portillo-Rodríguez, An Agent Architecture With Which to Improve Coordination and Communication in Global Software Engineering, University of Castilla-La Mancha, Ciudad Real, Spain, 2013.
[2] I. Kwan, A. Schroter, D. Damian, A weighted congruence measure, Proceedings of the Workshop on SocioTechnical Congruence, 2009.
[3] M. Cataldo, P. Wagstrom, J. Herbsleb, K.M. Carley, Identification of coordination requirements: implications for the design of collaboration and awareness tools, Proceedings of the Twentieth Anniversary Conference on Computer supported Cooperative Work, Banff, Alberta, Canada, 2006, pp. 353–362.
[4] S. Betz, et al., An evolutionary perspective on socio-technical congruence: the rubber band effect, Proceedings of the Third International Workshop on Replication in Empirical Software Engineering Research (RESER), 2013, pp. 15–24.
[5] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, Proceedings of the Twelfth International Conference on Evaluation and Assessment in Software Engineering, Italy, 2008, pp. 68–77.
[6] B.A. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software Engineering and Systematic Reviews, Chapman and Hall, 2015.
[7] Claes Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, Proceedings of the Eighteenth International Conference on Evaluation and Assessment in Software Engineering, London, England, United Kingdom, 2014, pp. 1–10.
[8] G. Valetto, S. Chulani, C. Williams, Balancing the value and risk of socio-technical congruence, Proceedings of the International Conference on Software Engineering, ICSE pages, Leipzig, Germany, 2008.
[9] E. Trainer, S. Quirk, C. De Souza, D. Redmiles, Bridging the gap between technical and social dependencies with Ariadne, Proceedings of the OOPSLA Workshop On Eclipse Technology Exchange, San Diego, California, 2005, pp. 26–30.
[10] S.B. Fonseca, C.R.B. De Souza, D.F. Redmiles, Exploring the relationship between dependencies and coordination to support global software development projects, Proceedings of the International Conference on Global Software Engineering, ICGSE '06, 2006, p. 243.
[11] M. Syeed, On the Socio-Technical Dependencies in Free/Libre/Open Source Software Projects, Tampere University of Technology, Tampere, Finland, 2015.
[12] M. Syeed, I. Hammouda, Socio-technical Congruence in OSS Projects: exploring Conway's law in FreeBSD, Proceedings of the IFIP International Conference on Open Source Systems, OSS, 2013, pp. 109–126.
[13] M. Syeed, K.M. Hansen, I. Hammouda, K. Manikas, Socio-technical congruence in the Ruby ecosystem, Proceedings of the International Symposium on Open Collaboration, Berlin, Germany, 2014, pp. 1–9.
[14] M. Syeed, I. Hammouda, Socio-technical dependencies in forked OSS projects: evidence from the BSD family, J. Softw. 9 (2014) 2014.
[15] M. Conway, How do committees invent? Datamat. Mag. 14 (4) (1968) 28–31.
[16] M. Cataldo, J. Herbsleb, K.M. Carley, Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity, Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Kaiserslautern, Germany, 2008, pp. 2–11.
[17] M. Cataldo, J.D. Herbsleb, Coordination breakdowns and their impact on development productivity and software failures, IEEE Trans. Softw. Eng. 39 (3) (2013) 343–360 DOI:10.1109/TSE.2012.32.
[18] I. Kwan, D. Damian, Extending socio-technical congruence with awareness relationships, Proceedings of the Fourth International Workshop on Social software Engineering, Szeged, Hungary, 2011, pp. 23–30.
[19] I. Kwan, A. Schroter, D. Damian, Does socio-technical congruence have an effect on software build success? A study of coordination in a software project. IEEE Trans. Softw. Eng. 37 (3) (2011) 307–324 DOI:10.1109/TSE.2011.29.
[20] P. Wagstrom, J. Herbsleb, K.M. Carley, Communication, team performance, and the individual: bridging technical dependencies, Acad. Manag. Proc. 2010 (1) (2010) 1–7 DOI:10.5465/ambpp.2010.54500789.
[21] A. Sarma, J. Herbsleb, A. Van Der Hoek, Challenges in measuring, understanding, and achieving social-technical congruence, Technical Report, Institute for Software Research, Carnegie Mellon University, 2008.
[22] F. Bolici, J. Howison, K. Crowston, Coordination without discussion? Socio-technical congruence and stigmergy in free and open source software projects, ICSE, Vancouver, Canada, 2009, p. 2009 pages.
[23] D. Šmite, Z. Galviņa, Socio-technical congruence sabotaged by a hidden onshore outsourcing relationship: lessons learned from an empirical study, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), (2012), pp. 190–202.
[24] J. Li, K.M. Carley, A. Eberlein, Assessing team performance from a socio-technical congruence perspective, Proceedings of the International Conference on Software and System Process (ICSSP), 2012, pp. 160–169.
[25] K. Ehrlich, M. Helander, G. Valetto, S. Davies, C. Williams, An analysis of congruence gaps and their effect on distributed software development, Proceedings of the International Conference on Software Engineering, Leipzig, Germany, 2008 pages.
[26] G Hofstede, Cultures and Organizations: Software of the Mind, McGraw-Hill, 2004.
[27] G. Valetto, et al., Using software repositories to investigate socio-technical congruence in development projects, Proceedings of the Fourth International Workshop on Mining Software Repositories, ICSE Workshops MSR '07. 2007, p. 25.
[28] K. Blincoe, G. Valetto, S. Goggins, Proximity: a measure to quantify the need for developers' coordination, Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW, 2012, pp. 1351–1360.
[29] E. Trainer, S. Quirk, C. De Souza, D. Redmiles, Analyzing a socio-technical visualization tool using usability inspection methods, Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC, 2008, pp. 78–81.
[30] C.R. De Souza, S. Quirk, E. Trainer, D.F. Redmiles, Supporting collaborative software development through the visualization of socio-technical dependencies, GROUP'07 – Proceedings of the International ACM Conference on Supporting Group Work, 2007, pp. 147–156.
[31] A. Begel, R. Deline, Codebook: social networking over code, Proceedings of the Thirty-First International Conference on Software Engineering – Companion Volume, ICSE-Companion, 2009, pp. 263–266.
[32] A. Begel, K. Yit Phang, T. Zimmermann, Codebook: discovering and exploiting relationships in software repositories, Proceedings of the Thirty-Second International Conference on Software Engineering, ACM/IEEE, 2010, pp. 125–134.
[33] A. Sarma, D.F. Redmiles, A. Van Der Hoek, Palantir: early detection of development conflicts arising from parallel code changes, IEEE Trans. Softw. Eng. 38 (4) (2012)

889–908 DOI:10.1109/TSE.2011.64.
[34] B. Al-Ani, D. Redmiles, Trust in distributed teams: support through continuous coordination, IEEE Softw. 26 (6) (2009) 35–40 DOI:10.1109/MS.2009.192.
[35] A. Borici, K. Blincoe, A. Schroter, G. Valetto, D. Damian, ProxiScientia: toward real-time visualization of task and developer dependencies in collaborating software development teams, Proceedings of the Fifth International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2012, pp. 5–11.
[36] J.M.R. Costa, R.M. Feitosa, C.R.B. De Souza, Tool support for collaborative software development based on dependency analysis, Proceedings of the Sixth International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010, pp. 1–10.
[37] A. Sarma, L. Maccherone, P. Wagstrom, J. Herbsleb, Tesseract: interactive visual exploration of socio-technical relationships in software development, Proceedings of the Thirty-First IEEE International Conference on Software Engineering, ICSE 2009, 2009, pp. 23–33.
[38] C.R.B. De Souza, T. Hildenbrand, D. Redmiles, Toward visualization and analysis of traceability relationships in distributed and offshore software development projects, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), (2007), pp. 182–199.
[39] B. Al-Ani, et al., Continuous coordination within the context of cooperative and human aspects of software engineering, Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2008), Leipzig, Germany, 2008, pp. 1–4.
[40] J. Portillo-Rodríguez, A. Vizcaíno, M. Piattini, S. Beecham, Using agents to manage socio-technical congruence in a global software engineering project. Inf. Sci. 264 (2014) 230–259 DOI:10.1016/j.ins.2014.01.009.
[41] J.C. Georgas, A. Sarma, STCML: an extensible XML-based language for socio-technical modeling, Proceedings of the Fourth International Workshop on Cooperative and Human Aspects of Software Engineering, Waikiki, Honolulu, HI, USA, 2011, pp. 61–64.
[42] S. Marczak, I. Kwan, D. Damian, Investigating collaboration driven by requirements in cross-functional software teams, Proceedings of the Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills, 2009, pp. 15–22.
[43] C. Bird, B. Murphy, N. Nagappan, T. Zimmermann, Empirical software engineering at Microsoft Research, Proceedings of the ACM Conference on Computer Supported Cooperative Work, Hangzhou, China, 2011, pp. 143–150.
[44] B.K. Mackellar, Analyzing coordination among students in a software engineering project course, Proceedings of the IEEE Twenty-Sixth Conference on Software Engineering Education and Training (CSEE&T), 2013, pp. 279–283.
[45] P.F. Xiang, et al., Ensemble: a recommendation tool for promoting communication in software teams, Proceedings of the International Workshop on Recommendation Systems for Software Engineering, Atlanta, Georgia, 2008, p. 1.