Investigating the Use of Tags in Collaborative Development Environments: A Replicated Study

Fabio Calefato
University of Bari
Dipartimento di Informatica
Via E. Orabona, 4 - 70125 Bari
+390805442286
calefato@di.uniba.it

Domenico Gendarmi
University of Bari
Dipartimento di Informatica
Via E. Orabona, 4 - 70125 Bari
+390805442286
gendarmi@di.uniba.it

Filippo Lanubile
University of Bari
Dipartimento di Informatica
Via E. Orabona, 4 - 70125 Bari
+390805443261
lanubile@di.uniba.it

ABSTRACT

Modern collaborative development environments have recently introduced tagging as a new feature in order to let developers annotate software artifacts with free keywords. Since tagging has the potential to have an impact on task management in software development processes, there is a need to understand how developers use tagging in projects supported by collaborative development environments and how developers' behavior differ from collaborative tagging in the Social Web.

We have conducted an independent replication of an empirical study, which first investigated how tags are used in a large software project. In our replication, we have analyzed two further projects coordinated through two different collaborative development environments, Jazz and Trac. The findings from our replicated study extend the initial contribution of the original study by (1) showing evidence of differences in tag usage between the two collaborative development environments, and (2) providing a clear understanding that tags used in such environments significantly differs from those used in traditional collaborative tagging systems.

Categories and Subject Descriptors

D.2.6 [**Programming Environments**]: *Integrated Environments*

General Terms

Human Factors

Keywords

Collaborative Development Environments, Tagging, Empirical Study, Replication.

1. INTRODUCTION

A collaborative development environment (CDE) is commonly regarded as a development platform, which provides a project workspace with a set of collaborative features available as components that extend core applications (e.g., the IDE), thus increasing the users' comfort and productivity of a global software team. Nowadays CDEs have become mainstream technologies for distributed teams as they provide a considerable help to software engineering activities by preventing developers from wasting effort in switching back and forth between different applications to communicate and work together [4].

Recently, besides classic groupware tools (e.g., email, shared calendars), other collaborative applications, known as social software have shown an appealing ability to overcome typical issues of remote group interaction, by making easier to communicate, collaborate, and share knowledge [1]. Among these applications, collaborative tagging has proved to be one of the most prominent features of the Social Web [5]. Collaborative tagging systems let individuals organize information they found on the Web using freely chosen keywords, commonly known as *tags*. The combined result is a collection of annotations, also called a *folksonomy* [10]. So far, Delicious[1] has led the way to social bookmarking, making tags popular to store, organize and share bookmarks for web resources. Nowadays, tagging has found its way also into CDEs with the purpose of providing developers with the ability to organize software artifacts (e.g., documents, source files, issue tracker entries) according to a personal perspective [14].

The aim of the research presented here is to analyze software artifact tagging in collaborative development environments, from the perspective of developers, and evaluate differences with tagging web content in collaborative tagging systems. A similar research that investigated how tags are used in a large software project during a period of two years has been recently conducted by Treude & Storey [16]. The project involved 175 developers who use the IBM Jazz platform to collaborate during the development of the platform itself. Because it is not possible to draw conclusions from a single study, we conducted an independent replication of the original research. Therefore, the main contribution of this work is that the findings from our replication increase the generalizability of the initial results by providing: (1) evidence of differences in tag usage between a popular open source CDE and an advanced commercial CDE; (2) a clear understanding that tags used in CDEs, and not only in Jazz, significantly differ from those used in traditional collaborative tagging systems.

The remainder of the paper is organized as follows. Section 2 presents an overview of related research on tagging that is relevant to our work. The replicated experiment, including research questions, settings, and data collection methods are reported in Section 3. Section 4 presents the results of our analysis, which are discussed in Section 5 compares these results to the original study and discusses similarities and differences between the two experiments. Section 6 discusses the threats to validity identified in our study. Finally, conclusions and further research activities are presented in Section 7.

[1] http://delicious.com

2. RELATED WORK

Most of existing research efforts on collaborative tagging comes from the Social Web domain [5]. In this section, we first review relevant studies that investigate usage dynamics in popular collaborative tagging systems on the Web. Then, we present existing works that have previously investigated how tags are used in software development. Finally, we discuss what a CDE is and how tagging software artifacts in such environment works.

2.1 Tagging in Social Web

The earliest study that investigated how tags are used over time by users was performed on Delicious by Golder and Humerman [8]. Building on Golder and Humerman's work, Sen et al. [13] defined three general tag categories (see Table 1), describing user tasks that can be supported by tagging in MovieLens[2]. In particular, they identify *Factual tags*, used to describe and find related movies, *Subjective tags*, used as a form of evaluation and recommendation, and *Personal tags*, applied for organizing a movie collection.

Overall, results from existing literature on collaborative tagging in the social web [3],[8],[9],[12] provide evidence that the types of tags used strongly depend on the types of resources being tagged by system users (e.g., music, movies, pictures, or web pages).

2.2 Tagging in Software Engineering

Tags have been already used in version control systems and issue tracking systems. However, developers were constrained to tag only check-ins and bug reports. Instead, the adoption of arbitrary keywords for the annotation of different kind of assets during the software life cycle has emerged only recently. Main research efforts on this direction focused on the collaborative annotation of source code.

The earliest research work on source code annotation was a tool named TagSEA (Tags for Software Engineering Activities), which provides tagging features to support programmers in defining semantically rich annotations to source code comments [15]. Other works in the same direction are represented by research prototypes that extend an IDE with additional tagging features.

Table 1. Tag categories by Sen et al. [13].

Tag Category	Description / Original categories
Factual tags	Tags that most people would agree apply to a given movie as they identify "facts" about a movie such as people, places, or concepts. They help to describe movies and to find related movies.
Subjective tags	Tags that express user opinions related to a movie. They can be used to help evaluate a movie recommendation.
Personal tags	Tags that have an intended audience of the tag applier themselves. They are most often used to organize a user's movies.

[2] http://movielens.umn.edu

Forward et al [7] proposed an Eclipse plugin named CodeSnippets, which uses tags to support software developers in the retrieval of code fragments from a software repository. Dekel & Herbsleb [6] presented a group memory-aid for software developers, named eMoose (Episodic Memory Of Open Source Efforts). eMoose offers a community-generated knowledge space consisting of annotations to functions in the source code and an IDE integration that presents such annotations when an artifact is viewed. Finally, Ossher *et al.* [11] built a prototype named BITKit (Business Insight Toolkit), which uses tags for identifying and organizing emerging concerns throughout the business analysis phase.

2.3 Tagging in CDEs: The Cases of Jazz and Trac

Jazz is an advanced commercial CDE built by IBM. It leverages the Eclipse notion of plug-ins to build a CDE as an extensible platform. The tool supports the management of *work items*, where a work item is a generalization of development tasks assigned to developers. Work items can be classified using predefined categories, and may be associated with other work items. In addition, Jazz supports discussion threads inside work items and a lightweight classification mechanism based on tags. Using this built-in feature, developers can freely associate user-defined keywords with work items.

Trac is a popular open source CDE. Compared to Jazz, Trac is more lightweight, as it integrates a wiki, an issue tracking system, and a front-end interface to version control systems. Project overview and progress tracking is allowed by setting a roadmap of milestones, which include a set of so-called *tickets* (i.e., tasks, feature requests, bug reports and support issues), and by viewing the timeline of changes. Trac also allows team members to use tags for annotating both tickets and wiki pages through the installation of a specific plug-in to enable tagging.

While Jazz work items are analogous to Trac tickets, there are some noticeable differences, concerning the set of features available in these two CDEs. Jazz is a full-featured collaborative development environment that can be tailored to accommodate a custom development process, thus appealing to larger software companies. As such, tagging is a built-in feature, available out of the box. Conversely, Trac represents a more lightweight alternative with a feature set kept at a minimum, typically intended for the development of free software in a collaborative endeavor. Thus, in Trac tagging is only available through a specific plugin that requires to be explicitly installed.

Finally, in order to highlight further differences between tagging in CDEs and social web tagging systems, we classify Jazz and Trac according to the taxonomy defined by Marlow et al. [9], which provides seven key dimensions in tagging systems' design.

As shown in Table 2, in general a CDE is a tagging system that can only have *User-contributed*, *Textual* resources. Despite the fact that each resource has an owner, both Trac and Jazz are *Free-for-all* tagging systems, as developers are allowed to collectively tag any item, with the only limitation of using unique keywords (*Set-model*). Both CDEs present a very primitive form of suggestive tagging (*Viewable tagging*) by providing users with all the tags already used in the whole project when they try to apply a new tag to a resource. Instead, the two environments are quite different with respect to the dimensions of *resource connectivity*

Table 2. Classification of Trac and Jazz as tagging systems.

Dimension	Category value
Tagging Rights	Free for all
Tagging support	Viewable tagging (primitive suggestive tagging)
Aggregation Model	Set-model
Object Type	Textual
Source of material	User-contributed
Resource connectivity	Links & Groups (only Jazz)
Social Connectivity	Groups (only Jazz)

and social connectivity, since Jazz incorporates features to directly link developers and resources to each other, whereas Trac is completely missing such features.

Finally, another important difference, not captured by any dimension of the classification in [9], is the lack in the user interface of CDEs of any personal tagging space where users can view and manage their personal vocabulary of tags. Thus, unlike typical collaborative tagging systems, CDEs do not allow to observe either all the tags (with their occurrences) used by a specific user or all the users who have used a specific tag. This lack prevents the generation of a network effect that fosters the discovery of relationships between users, tags, and resources, typical of folksonomies [10].

3. THE REPLICATED STUDY

We performed an independent replication of the study originally presented in [16] by Treude & Storey. The software project taken into account in the original study included 175 contributors who used the IBM Jazz platform to collaborate during the development of the platform itself. The researchers used a mix of quantitative and qualitative analysis performed on the data gathered from the project repository, between May 2006 and April 2008, and semi-structured interviews, conducted with four developers.

With respect to the original study, we did not find the need for further interviews to developers and then we focused on the research questions that required quantitative analysis of the CDE repository. Second, we extended the scope of the original study by investigating tag usage in two different projects and CDEs, rather than only in one project and environment, as in the case of Treude & Storey's research. In our replication, in fact, we gathered the data available from the public repositories of the Jazz Foundation[3] and WebLion[4] projects.

The first project, Jazz Foundation, is being currently developed using the IBM Jazz CDE, and refers to the development of the Jazz Team Server. The official team consists of 41 team areas with some teams acting as sub-teams of larger teams. The total number of contributors is approximately of 477 individuals, although this is not the number of distinct contributors as some people are assigned to multiple teams. The second project, WebLion, is under active development using the Trac environment. It refers to the development of a variety of add-on products for the Plone Content Management System[5] and Zope application framework[6], used for consulting, training, and community services at Penn State. The official WebLion team consists of 32 contributors organized in five different groups. However, the open-source nature of the project fosters external contributors to commit new add-ons.

The choice of these two specific projects was taken according to factors such as the number of contributors and the overall number of tags.

3.1 Research Questions & Data Collection

The overall goal of this replication is to (1) analyze how developers tag software artifacts in CDEs, and (2) compare tagging software artifacts in CDEs to tagging web content in collaborative tagging systems. Thus, the two research questions retained from the original study are the following:

RQ1: *How is the tagging feature used by developers to annotate artifacts in collaborative development environments?*

In order to answer RQ1, we measured how many tags the developers used in both projects.

RQ2: *How does tagging software artifacts in collaborative development environments compare to tagging web content in collaborative tagging systems?*

In order to answer RQ2 we classified and measured what categories of tags are used more frequently. More specifically, we evaluated how tags are adapted to meet the needs of software developers and whether tagging *per se* is affected by the characteristics of the two different collaborative development environments.

Unlike the original study, we did not have full access to project repositories. Hence, we were constrained to collect the data on tag usage and frequency through the web site of each project, due to the lack of an API access to the repositories programmatically. In particular, to gather data from Jazz Foundation we registered into the Jazz.net community Web site and built queries for existing work items into the project. However, with a limited access to the project repository we could only retrieve data on tag usage from November 1, 2008 to September 18, 2009. The retrieval of data from the WebLion project did not require any registration as the Trac environment set up for the project allows anyone to perform read-only queries concerning all the tickets created since its origin. Hence, data retrieved for the analysis of tag adoption in WebLion refers to the period from the beginning of the project in June 2006 to September 18, 2009. All the data collected about tags, work items, and tickets of the two projects are summarized in Table 3.

Finally, in order to observe possible differences between tagging in CDEs and collaborative tagging systems, we performed an analysis on the *type* of tags used by the developers. We took into account the whole dataset for the WebLion project. Instead, for Jazz Foundation given the large size of the project, we had to

[3] https://jazz.net/projects/jazz-foundation

[4] http://weblion.psu.edu

[5] http://plone.org

[6] http://www.zope.org

Table 3. Data retrieved from the two projects.

	Jazz	Trac
Total n° of work items/ tickets	12,535	1,405
Total n° of tag applications	8,719	298

narrow down the retrieved dataset to the work items created in the last four weeks. This time interval was chosen in order to obtain two datasets comparable in size. Thus, the narrowed dataset for the Jazz Foundation project included 355 work items and 98 distinct tag terms, whereas 177 distinct tags were extracted from the whole dataset of WebLion.

4. RESULTS

This section presents the findings from our replication. Results are presented according to the two research questions defined earlier.

4.1 RQ1 – Adoption of Tagging in CDEs

In order to answer our first research question, we analyzed how existing work items in Jazz Foundation, and tickets in WebLion, have been tagged in the respective environments. In the previous study, Treude & Storey observed how tag adoption in Jazz evolved over time. Unfortunately, this could not be replicated in our study, because the queries through the Web interface do not allow this kind of analysis.

Figure 1 and Figure 2 show the distribution of tags per work items and per tickets, respectively in Jazz Foundation and WebLion. The percentage of tagged work items in the Jazz environment, where the tagging feature is built-in, is higher than the percentage of tagged tickets in the Trac environment. Specifically, within the Jazz Foundation project the 44% of work items have been tagged at least with one tag, whereas only the 13% of all the tickets in the WebLion have at least one tag. Nevertheless, both work items in Jazz and tickets in Trac, whether tagged, usually have one or two tags, and rarely three or more.

From the analysis on the type of work items within the Jazz Foundation project, we found that the 95% of all work items were defects (49%), tasks (23%), or enhancements (23%). In the WebLion project, tickets can only be of these three types. Of the 1,405 tickets retrieved, 44% were defects, 30% were tasks, and the remaining 26% were enhancements.

Afterwards, we also analyzed tag adoption according to the specific type of work item/ticket in both projects. With respect to the Jazz Foundation project (see Figure 3), such analysis revealed that, although in absolute most of the tagged work items belong to the *defect* category (2747 out of 6124, ~45%), the number of tagged *enhancements* is proportionally higher, with over a half of its items (1663 out of 2787, ~60%) having at least one tag. Also the distribution of tags per type of work item, shown in Figure 5, confirms this general trend.

As regards the WebLion project, the analysis of tag adoption did not show any large difference between the three different types of tickets (see Figure 4). Finally, although the differences are not large, the analysis of tag distribution per type of ticket revealed that *defect* is the most tagged category within the Trac project.

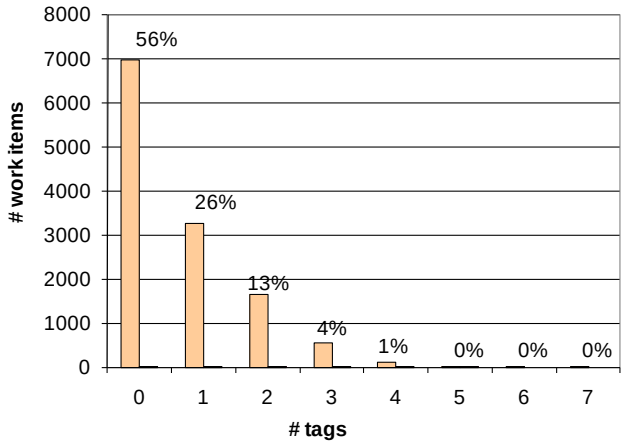

Figure 1. Distribution of tags in Jazz Foundation.

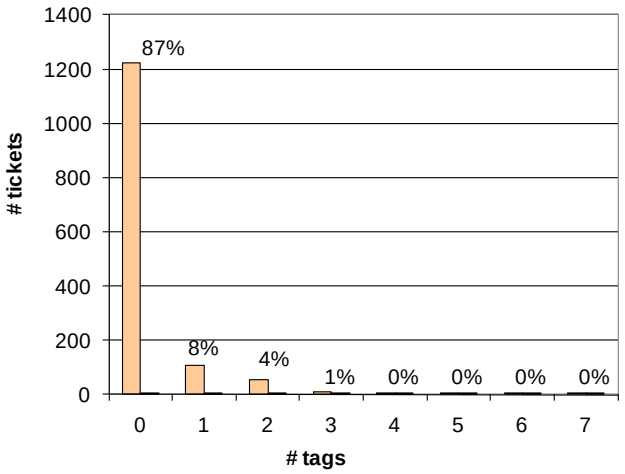

Figure 2. Distribution of tags in WebLion.

4.2 RQ2 – Tagging in CDEs vs. Collaborative Tagging Systems

With respect to RQ2, we are interested in understanding if tags used to annotate items during software development differ from tags adopted by users in collaborative tagging systems on the web. However, differences in tagging systems have a significant impact on how users utilize these systems, and thus on the usefulness of resultant tags. Therefore, for each of the two projects, we selected a dataset of the tags applied and then we classified such tags according to a taxonomy proposed by Treude & Storey in the original study [16]. The full list of tag categories with a brief description is reported in Table 4.

During the classification of tags from the Trac-based WebLion project, we felt the need to add two new tag categories: *CDE-specific* and *Divorced*.

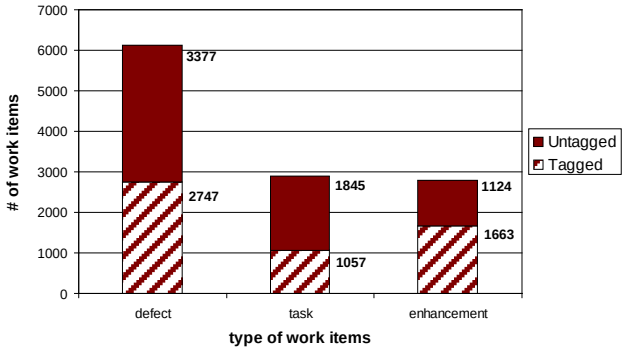

Figure 3. Percentage of tagged work items per type (Jazz).

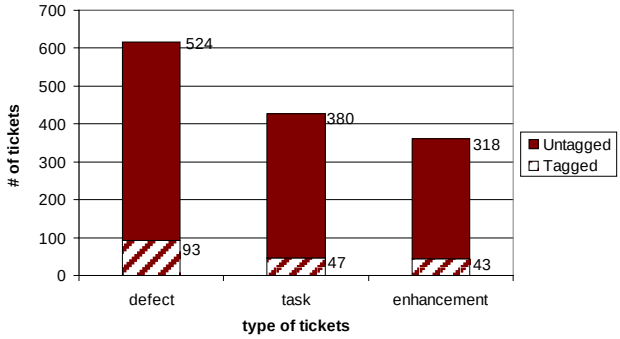

Figure 4. Percentage of tagged tickets per type (Trac).

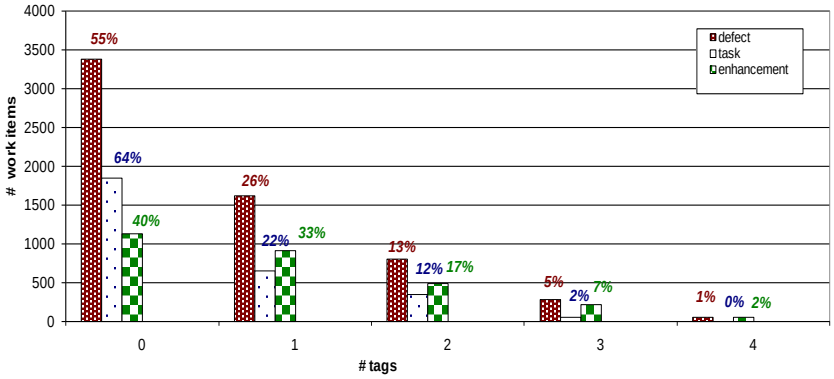

Figure 5. Distribution of tags per type of work item (Jazz).

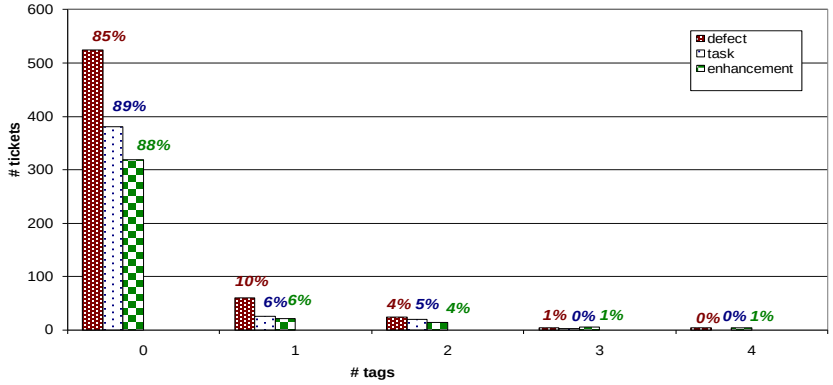

Figure 6. Distribution of tags per type of ticket (Trac).

The *CDE-specific* tag category was added because in the WebLion project we observed a particular way of using tags as a form of traceability link. For example, some tags were named with the number of another ticket referred in the description of the tagged ticket. Other times, the name of the tag matched the name of a specific Wiki page.

The other category, *Divorced tags*, was added because we realized that a number of tags were actually part of compound names. A typical example is a ticket with the summary *"Front Page - Rotating Pictures"* that was tagged with two different tags "Front" and "Page". In such a case, we classified both tags as *Divorced* tags as they only make sense if combined.

Due to its large size, the original dataset of the Jazz Foundation Project was narrowed down considering only the last four weeks of the project. Thus, from the narrowed dataset of 355 work items, we extracted 98 distinct tag terms. Then, two persons separately performed the classification of tags according to the taxonomy in Table 4.

Table 4. Taxonomy of tags for a CDE (adapted from [16]).

Category name	Category description
1- Lifecycle related	Tags related to a release or milestone in the development process, usually they include the name of the release or milestone in the tag name. These tags are transient as they are used mostly in the period of time related to specific the milestone or release
2- CDE-specific*	Tags used to indicate explicitly a relationship with some specific CDE-managed resource, such as a ticket or wiki page, which is referenced by name/identifier.
3- Component-specific	Tags used to refine or replace the resource categories that the CDE already provides. Their use depends largely on the presence of a categorization mechanism for software components.
4- Cross-cutting	Tags which capture aspects of resources that cross-cut the hierarchy of categories for software components. They are frequently used to denote non-functional requirements such as usability or performance.
5- Divorced*	Tags that do not stand alone but only make sense when combined to another tag of the resource. They are the result of a human error in correctly binding a compound name.
6- Idiosyncratic	Tags that do not belong to any of the previous categories. They are generally used to support some individual and collaborative processes for which there is no formal tool support, such as reminders or to-do actions, or just as a way to informally adding metadata to resources which are neither related to a milestone nor to specific components or cross-cutting concerns.

***** categories added to original taxonomy

A valuable support for a correct classification was the description of the tagged work items, whose reading often helped to understand the intended meaning of tags. The classification in fact was done based on tag terms, but it was always verified through the analysis of the reason why the tagged item was created. Finally, the two people compared their respective choices and converged on a shared classification. The same procedure was applied to the whole dataset retrieved from the WebLion project. In this case, from the 1,405 tickets we extracted 177 distinct tags. Two people again first classified all the tags separately and then discussed until they converged to a shared classification. Table 5 shows the results of the classifications for Jazz and Trac.

In our sampling of tags retrieved from the Jazz Foundation project, we found that most of the distinct tag terms were *Component-specific* (39%) or *Idiosyncratic* (39%). Only a small part of distinct tags were *Cross-cutting* (17%) and very few distinct tags were *Lifecycle related* (5%). We did not find any tag classifiable as *CDE-specific* or *Divorced*. However, observing all the instances of these tags in the whole project dataset we found that tags classified as *Cross-cutting* are the most frequently used to tag work items (37%), followed by *Component-specific* (35%), *Idiosyncratic* (25%), and *Lifecycle related* tags (3%).

For the whole dataset of the WebLion project in Trac, most of the 177 distinct tag terms are *Cross-cutting* (30%) but there is also a considerable amount of distinct *Divorced* tags (23%), *Idiosyncratic* tags (19%) and *Component-specific* tags (15%). *Lifecycle related* (7%) and *CDE-specific* (6%) are the two categories with the fewest amount of distinct tag terms. Observing how these tags had then been used, we found again that the most used tags are *Cross-cutting* (31%), while the usage of *Divorced* tags, *Idiosyncratic* tags and *Component-specific* tags ranges from 16% to 19%. The 11% of all the used tags were classified as *Lifecycle related* tags while *CDE-specific* tags resulted as the least used (5%).

For the Jazz Foundation project, the sum of *Component-specific* and *Cross-cutting* tags accounts for more than 50% of all tags in terms of both tag instances and different tag names. Also for the WebLion project in Trac the tags that belong to these two categories are the 50% of all the tag instances. Conversely, categories like *Lifecycle related* and *CDE-specific* capture particular needs for grouping together all the items related to a specific milestone or to implement a traceability mechanism that the environment does not provide, as in the case of Trac. *Idiosyncratic* is the only category that could be considered as a form of personal tags, similar to those used in collaborative tagging systems.

5. COMPARISON OF REPLICATIONS

Table 6 compares our replicated study to the original research by Treude & Storey [16]. While the original study collected data for only one project developed using the Jazz CDE, our replication took into account two different projects, Jazz Foundation and WebLion, developed respectively using Jazz and Trac as CDE.

Jazz Foundation and WebLion are representatives of two extremes of the software development spectrum, as the former is an effort led by a large company (IBM), whereas the latter is a smaller,

Table 5. Classification of tags in Jazz and Trac.

	Jazz		Trac	
Category	***tag terms***	***tag instances***	***tag terms***	***tag instances***
1- Lifecycle related	5 (5%)	194 (3%)	13 (7%)	32 (11%)
2- CDE-specific	0	0	10 (6%)	14 (5%)
3- Component-specific	38 (39%)	1955 (35%)	27 (15%)	57 (19%)
4- Cross-cutting	17 (17%)	2089 (37%)	54 (30%)	94 (31%)
5- Divorced	0	0	40 (23%)	48 (16%)
6- Idiosyncratic	38 (39%)	1411 (25%)	33 (19%)	53 (18%)
Total	98	5649	177	298

open-source project involving volunteers. Hence, one important contribution of our work is that, by replicating the study on different conditions, we are better able to generalize and interpret with more confidence the results on the use of tags in software development. In the following, we provide an answer to the two research questions in our replication.

5.1 RQ1 –Tagging Adoption influenced by CDE Specific Support

The findings from our replication are consistent with those reported in the original study only with respect to the Jazz CDE. Instead, comparing the two projects we observed a more widespread adoption of tags in Jazz rather than in Trac.

In the original study, the 25% of the whole dataset of work items (37,590) was tagged at least once. We observed important differences in tag adoption between Jazz and Trac, for which the percentage of tagged items and the distribution of tags are quite divergent. In Jazz Foundation, almost half of all the work items (44% of 12,535 work items) have been tagged with at least one tag and a considerable percentage of work items have also two or more tags. The same large adoption of tags was not found into the WebLion project, for which only the 13% of 1,405 total tickets was tagged at least once.

We did not have access to raw data necessary to compute the percentage of developers using the tagging feature, as in the original study. However, unlike the former study, we were able to analyze the distribution of tags per type of work item or ticket. This further analysis revealed that in Jazz Foundation *enhancement* is proportionally the type of work items with more tags (60%). In WebLion instead, we did not find any remarkable difference concerning tag adoption for different type of tickets.

Therefore, to answer RQ1 we can say that tags are largely adopted in a full-featured environment where tagging is a built-in feature, as in the case of Jazz. Instead, tags did not play such an important role in a lightweight environment where tagging is considered an add-on feature, as in Trac.

5.2 RQ2 – Tags as a Supplement of Existing CDE Categories

One of the most significant results from the original study was the definition of four categories of development-specific tags in Jazz. In particular, Treude & Storey found that most of distinct keywords used for tagging fall in the *Component-specific* categories (61%), whereas *Cross-cutting* tags are the most frequently used to tag work items (37%). In our replication, we analyzed the dataset of tags from the Jazz Foundation project to categorize them according to the proposed taxonomy. In this case, such taxonomy was exhaustive, as we were able to assign all tags to one of the four categories. In addition, consistently with previous results, we found that also in the Jazz foundation project *Component-specific* and *Cross-cutting* were, respectively, the categories with the higher distinct keywords (39%) and the higher frequency of use (37%). Both, categories capture the intention of using tags as a supplement to existing categories in the CDE (see Table 4).

Our findings for the Trac environments were different. In fact, we discovered a couple of particular tagging behaviors through the analysis of tags used by the developers of WebLion. Therefore, we had to extend the taxonomy of tags for a CDE proposed in the original study. In particular, we identified the use of *CDE-specific* tags in Trac as a form of traceability mechanism between tickets, or between tickets and other artifacts, such as wiki pages, which is unsupported by the CDE. This behavior is not observable in Jazz as the platform already provides a mechanism to explicitly link resources within the environment, thus making pointless to use tags for that purpose. An implication of this finding is the confirmation of a hypothesis, raised by Treude & Storey [16], but left unverified: developers use tags as an informal and flexible mechanism to add some missing functionality in managing and organizing artifacts within an environment.

The analysis of tags from WebLion also revealed that, in contrast with the Jazz Foundation project, developers often make errors when they tag tickets, as shown by the number of tags in the *Divorced* category (i.e. tags that only make sense when combined to others). This finding suggests that developers of Jazz Foundation can be considered more expert taggers than WebLion developers. Unlike Jazz, where tagging is a built-in feature, the tagging mechanism in Trac is a very recent and additional feature that could be unknown or considered as unnecessary. Furthermore, in Jazz Foundation the development of the product is the contribution of full-time IBM developers who master the complex environment that is used on a daily basis. On the contrary, WebLion is the result of the efforts from volunteers who only use Trac as a means to develop software in a collaborative way.

Finally, we found that for WebLion, *Cross-cutting* was both the tag category containing the largest number of distinct keywords (30%) and the one most frequently used to tag tickets (31%). This result again suggests the intention of using tags as a supplement to existing categories in a CDE.

Overall, the comparison between the use of tags in CDEs and in collaborative tagging systems suggest that tags are mainly regarded as an enhancement of the existing categories already available in an environment, so as to accommodate developers' perspectives in managing software artifacts. However, this is not how tags are typically used in collaborative tagging systems. As inferred from the tag taxonomy defined by Sen et al. [13] (see Section 2), in collaborative tagging systems the primary function of tags is that of descriptive metadata used in the personal organization of resources (*personal tags*), often enriched with known facts and opinions (*factual tags* and *subjective tags*). Besides, unlike CDEs, collaborative tagging systems do not generally provide pre-existing categories to users.

Therefore, to answer RQ2 we can say that tagging software artifacts is quite different from tagging web content due to the different support to resource organization built in CDEs and collaborative tagging systems, respectively.

6. THREATS TO VALIDITY

In this section, we discuss the threats to validity of our findings. The two individuals who classified tags were neither owners nor users of the tags and, as such, they could have not correctly captured the true original meaning of the tag name. This threat was mitigated by asking to discuss the results of the initial independent classification in order to converge on a shared meaning and mutually agree on the resultant categorization.

Furthermore, it might also be argued whether the results from this study can be generalized to the general software engineering industry. In our study, this concern is alleviated to some extent by the replication of an existing study on two distinct development environments and software projects. Yet, it remains difficult to draw general conclusions as any software engineering process depends on a potentially large number of relevant context variables [2].

7. CONCLUSIONS

Collaborative tagging for software development represents quite an exploratory topic in literature. This work has reported about the replication of an empirical investigation, originally conducted by Treude & Storey [16], on how tags are used by software developers. More specifically, this paper has focused on: (1) analyzing how software developers use tags for task management into CDEs, in which development tasks are usually assigned to developers and classified according to predefined categories; (2) indentifying any difference between tagging software development artifacts and web resources. While, the former study took into account data from only one project and one collaborative environment, we analyzed tagging behaviors in two different projects (Jazz Foundation and WebLion), developed using two distinct collaborative development environments (Jazz and Trac, respectively). Overall, two main results can be drawn from our replication.

First, tagging behavior varies on the collaborative development environment. Our study showed that tags are largely adopted in a environment where tagging is a built-in feature, as in the case of Jazz. We could also observe different tagging behaviors according to the type of work item being tagged only in Jazz, but not in Trac.

Second, software developers use tags in a quite different way compared to how tags are used in collaborative tagging systems on the Web. In fact, tags are seen by developers mostly as a supplement to existing predefined categories available in a CDE.

Table 6. Comparison of the two replications.

		Original Study	**Replicated Study**	
Context	***CDE***	Jazz	Jazz,	Trac
	Project	?	Jazz Foundation	WebLion
	Data	Full access to project repository (May '06 – Apr. '08), interviews w/ developers	Limited access to project repository via Web (Nov. '08 – Sep. '09)	Limited access to project repository via Web (Jun. '06 – Sep '09)
	Analysis	Both quantitative and qualitative	Quantitative	
		Original Study	**Replicated Study**	
Results	***RQ# description***	***Findings***	***Findings CDE / Project***	
			Jazz / Jazz Foundation	***Trac / WebLion***
	RQ1 ***# of different tags used by developers to annotate artifacts in CDEs and their frequency***	25% of total work items (37.590) tagged at least once	44% of total work items (12.535) tagged at least once	13% of total tickets (1.405) tagged at least once
		63% of total project contributors (178) applied at least one tag	-	-
		-	* 45% of total defect work items (6.124) tagged at least once * 36% of total task work items (2.902) tagged at least once * 60% of total enhancement work items (2.787) tagged at least once	* 15% of total defect tickets (617) tagged at least once * 11% of total task tickets (427) tagged at least once * 12% of total enhancement tickets (361) tagged at least once
	RQ2 ***differences between tags used in CDEs and in collaborative tagging systems***	Work item properties change over time	-	-
		Work item have limited lifetime	-	-
		Identified 4 categories of development-specific tags in Jazz: * *Lifecycle* * *Component-specific* * *Cross-cutting* * *Idiosyncratic*	Same tag categories identified	Two extra tag categories identified: * *CDE-specific* * *Divorced*
		* Most of distinct keywords used for tagging are component-specific (61%) * Cross-cutting tags frequently used to tag work items (37%)	* Most of distinct keywords used for tagging are component-specific (39%) * Cross-cutting tags frequently used to tag work items (37%)	* Most of distinct keywords used for tagging are cross-cutting (30%) * Cross-cutting tags frequently used to tag tickets (31%)

Consistently with Treude & Storey's observation, we found that sometimes tags are also used to address specific needs of software developers, such as creating a traceability mechanism missing in the environment.

As future work, we plan to provide tool support that follows some suggestions emerged from these empirical studies. For example, a tag checking mechanism might help in preventing tagging errors, such as *Divorced* tags, through a semantic analysis of the textual content of work items. Moreover, we argue that it could be more useful to provide developers with a more sophisticated suggestive tagging feature that proposes just those tags that have been already applied to work items in the same category (as an approximation of *Component-specific* tags) or only those applied to work items in the same milestone (as an approximation of *Lifecycle related* tags). Finally, as for the original classification proposed by Treude & Storey, it could potentially benefit from a further decomposition of the *Idiosyncratic* category, for instance, by turning to-dos and reminders as categories on their own.

8. ACKNOWLEDGMENTS

We wish to thank Antonio Matarrese and Adriana Napolitano for their valuable help in the gathering and analysis of data.

9. REFERENCES

[1] Abbattista, F. Calefato, F. Gendarmi, D. Lanubile, F. 2008. Incorporating social software into distributed agile development environments. In Proceedings of 23rd IEEE/ACM International Conference on Automated Software Engineering . ASE Workshops 2008, 46-51.

[2] Basili, V. R., Shull, F., and Lanubile, F. 1999. Building Knowledge through Families of Experiments. *IEEE Trans. Softw. Eng.* 25, 4 (Jul. 1999), 456-473.

[3] Bischoff, K., Firan, C. S., Nejdl, W., and Paiu, R. 2008. Can all tags be used for search?. In *Proceeding of the 17th ACM Conference on information and Knowledge Management.* CIKM '08. ACM, New York, NY, 193-202.

[4] Booch, G. and Brown, A.W., *Collaborative Development Environments*, Advances in Computers 59, 2003.

[5] Chi, E.H., The Social Web: Research and Opportunities, IEEE *Computer* , vol.41, no.9, pp.88-91, 2008.

[6] Dekel, U. and Herbsleb, J. D. 2008. Pushing relevant artifact annotations in collaborative software development. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work.* ACM, New York, NY, 1-4.

[7] Forward, A., Lethbridge, T., and Deugo, D. 2007. CodeSnippets Plug-in to Eclipse: Introducing Web 2.0 Tagging to Improve Software Developer Recall. In *Proceedings of the 5th ACIS international Conference on Software Engineering Research, Management & Applications*. SERA. IEEE Computer Society, Washington, DC, 451-460.

[8] Golder, S. A. and Huberman, B. A. 2006. Usage patterns of collaborative tagging systems. *J. Inf. Sci.* 32, 2 (Apr. 2006), 198-208.

[9] Marlow, C., Naaman, M., Boyd, D., and Davis, M. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*. HYPERTEXT '06. ACM, New York, NY, 31-40.

[10] Mathes. A. 2004. Folksonomies - Cooperative Classification and Communication Through Shared Metadata. UIC Technical Report.

[11] Ossher, H., Amid, D., Anaby-Tavor, A., Bellamy, R., Callery, M., Desmond, M., De Vries, J., Fisher, A., Krasikov, S., Simmonds, I., and Swart, C. 2009. Using tagging to identify and organize concerns during pre-requirements analysis. In *Proceedings of the 2009 ICSE Workshop on Aspect-Oriented Requirements Engineering and Architecture Design*. International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 25-30.

[12] Robu, V., Halpin, H., Shepherd, H. 2009. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Transactions on the Web* 3, 4, pp. 1-34.

[13] Sen, S., Lam, S. K., Rashid, A., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., and Riedl, J. 2006. tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work.* ACM, New York, NY, 181-190.

[14] Storey, M., Ryall, J., Bull, R., Myers, D., and Singer, J. 2008. TODO or to bug: exploring how task annotations play a role in the work practices of software developers. In *Proceedings of the 30th international Conference on Software Engineering*. ICSE '08. ACM, New York, NY, 251-260.

[15] Storey, M., Ryall, J., Singer, J., Myers, D., Cheng, L., and Muller, M. 2009. How Software Developers Use Tagging to Support Reminding and Refinding. *IEEE Trans. Softw. Eng.* 35, 4 (Jul. 2009), 470-483.

[16] Treude, C. and Storey, M. 2009. How tagging helps bridge the gap between social and technical aspects in software development. In *Proceedings of the 2009 IEEE 31st international Conference on Software Engineering*. International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 12-22.