FOLCOM or the Costs of Tagging

Elena Simperl[1], Tobias Bürger[2], and Christian Hofer[3]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
elena.simperl@kit.edu
[2] Salzburg Research Forschungsgesellschaft mbH, Salzburg, Austria
tobias.buerger@salzburgresearch.at
[3] University of Innsbruck, Innsbruck, Austria
c.hofer@student.uibk.ac.at

Abstract. This paper introduces FOLCOM, a FOLksonomy Cost estimatiOn Method that uses a story-points-approach to quantitatively assess the efforts that are cumulatively associated with tagging a collection of information objects by a community of users. The method was evaluated through individual, face-to-face structured interviews with eight knowledge management experts from several large ICT enterprises interested in either adopting tagging internally as a knowledge management solution, or just in tangible evidence of its added value. As a second theme of our evaluation, we calibrated the parameters of the method based on data collected from a series of six user experiments, reaching a promising prediction accuracy within a margin of $\pm 25\%$ in 75% of the cases.

1 Motivation and Main Contributions

Capitalizing on their popularity on the public Web – through Web 2.0-style platforms such as del.icio.us, Flickr and YouTube – folksonomies gradually enter the enterprise arena with the promise to provide a lightweight, easy-to-use means to manage and share knowledge in a collaborative environment [6,14,20]. Nevertheless, to sustain this trend, and to have a strong case in favor of knowledge-based technologies, CIOs and CTOs are yet seeking for instruments to accurately analyze the costs and benefits associated with the adoption of tagging, and the creation and maintenance of folksonomies, within enterprises. A study done by McKinsey in 2008 on the usage of Web 2.0 technologies within companies confirms this state of affairs – the most important barrier impeding the mainstream adoption of tagging, wikis, social networks, to name just a few, at the corporate level lays within the fact that the benefits of these technologies are not tangible, or yet poorly investigated [13]. Furthermore, the study identifies a number of additional open issues in this regard: Web 2.0 projects often lack commitment at the management level, are rarely fully compliant with the corporate culture, and overlook the importance of setting in place the proper incentive schemes to ensure the durable involvement of a critical mass of enterprise users. Supported by these findings, we argue that instruments to analyze the real costs and benefits of tagging are a must to provide businesses with the right arguments in favor of the usage of Web 2.0 technologies, and to encourage large-scale, sustainable take-up.

P. Cimiano and H.S. Pinto (Eds.): EKAW 2010, LNAI 6317, pp. 163–177, 2010.

This paper introduces FOLCOM, which offers such an instrument. FOLCOM, which stays for **FOL**ksonomy **C**ost estimati**O**n **M**ethod, uses a story-points-approach to quantitatively assess the efforts that are cumulatively associated with tagging a collection of information objects by a community of users. We surveyed well-established approaches to cost estimation in software and knowledge engineering, in particular along the themes of agile development, and open source and community-driven development, which share many commonalities with the tagging scenario from a procedural point of view. Based on the findings of this survey, we designed a method by which the time required to annotate a collection of information objects by a community of users can be estimated in relation to the size of this collection, the complexity of the content it contains, and the expertise of the community contributing to this effort. The method was evaluated using individual, face-to-face structured interviews with eight knowledge management experts from several large ICT enterprises interested in either adopting tagging internally as a knowledge management solution, or just in tangible evidence of its added value. In addition, we calibrated the parameters of the method by collecting data from a series of six user experiments, reaching an adequate prediction accuracy within a margin of $\pm 25\%$ in 75% of the cases.

Applications of FOLCOM include planning and controlling of knowledge management projects. The results of the method can be transferred into financial outputs based on the employee-salary/time relation. Furthermore, the estimates offer a quantitative means to compare the added value of folksonomies with alternative approaches to organize and structure knowledge (e.g., ontologies) in terms of effort and costs. Finally, by monitoring the efficiency of tagging one could identify specific difficulties and challenges of the tagging process, and consider automated tool support for those aspects.

2 Folksonomies and Tagging in a Nutshell

The term "folksonomy" was first coined by Thomas Vander Wal in 2004 as the *"result of personal free tagging of information and objects (anything with a URL) for one's own retrieval."*[1] Typically, folksonomies emerge in Web-based, collaborative environments in which users produce, consume and share information. They are lightweight forms of knowledge management, unconstrained in the choice of the keywords they include, openly structured, and inclusive.[2]

The process of creating a folksonomy is conceived and understood as a continuous, iterative effort in which a loosely defined community of users describe or annotate information objects through tags, according to their knowledge management needs. The operations which can be executed in the course of this process can be divided into two distinct categories: (i) *add*, through which a user assigns a tag to an object; and (ii) *remove*, through which the user deletes a tag previously assigned to an object. Changes, such as modifications of the keywords used within a tag, can be modeled as sequences of add and remove operations [8].

[1] `http://vanderwal.net/folksonomy.html`

[2] Folksonomies are inclusive in the sense that tags assigned to knowledge resources and objects do not exclude each other.

Vander Wal differentiates between two styles of folksonomy creation:[3] *collective* and *collaborative*. In the collective case the folksonomy reflects the individual perspectives of the user community with respect to the objects being described or annotated. In other words, the folksonomy is merely the collection of tags contributed by the users throughout the tagging process. In contrast, in the collaborative case the tags are agreed within the community, and the resulting folksonomy represents the consensual view of the contributors with respect to the vocabulary that should be used for tagging. Another distinction is made between *broad* and *narrow* folksonomies.[4] A broad folksonomy is typically created by many users freely assigning tags to information objects. The same tag can be used multiple times by different users to describe the same object. This type of folksonomy is delivered, for instance, by the del.icio.us platform. In del.icio.us a large user community tags bookmarks based on their own vocabulary, while network effects are crucially reinforced by automatically suggesting popular tags. The emerging folksonomy is acknowledged to be a useful means to build a shared vocabulary, and to select the preferred terms to describe specific content. In narrow folksonomies objects are tagged by a comparatively lower number of users. Users can not re-use externally contributed tags of the same information object – though they can, of course, use the same keywords to describe or annotate them. The resulting, much more focused folksonomy is useful for information retrieval, in particular for types of content that are not easily findable using traditional (e.g., full-text-based) search technology. A prominent example thereof is Flickr. In Flickr each information object is associated with a low number of tags, contributed mainly by the author, and by other users who are in possession of adequate rights. The author can add, remove and change the tags related to the content she uploads, and can grant access rights to other users to do so. The resulting folksonomy is an effective means to find Flickr photos.

In the next sections we will explain how FOLCOM can be used to accurately predict the efforts associated to creating such folksonomy structures within a community of users. First, we introduce the story-points method, the cost estimation approach which is at the core of FOLCOM, and then FOLCOM itself.

3 The Story-Points Method

The story-points method has its origins in agile software development [3,4].

3.1 Why Story Points?

We selected it after conducting a comprehensive literature survey of some of the most important cost estimation methods in software and knowledge engineering published in the last two decades – software engineering as an archetypal area in which cost estimation has a long-standing tradition both among researchers and industry; and knowledge engineering as it bears many similarities in the type of artifacts produced, which are in both cases knowledge models. We examined these approaches with respect to their applicability to the folksonomy creation process. In this paper we can only sketch the

[3] http://www.personalinfocloud.com/2008/03/getting-to-know.html
[4] http://www.personalinfocloud.com/2005/02/explaining_and_.html

main rationales for choosing this particular method due to space limitations, but a full account of the findings is available in [1].

In brief, from a procedural point of view folksonomy creation exhibits a number of features which make the application of well-established cost estimation methodologies, methods and techniques from classical software engineering unfeasible, but there are some parallels to agile and open-source software development [11]. Among these we highlight the open, evolving nature of the overall process, the lack of a clearly defined process model – including phases, activities and tasks, as well as roles, skills and expertise associated with them. The unavailability of empirical data from historical projects introduces additional constraints, as many approaches in cost estimation heavily rely on it to calibrate the underlying prediction model.

In agile software engineering, requirements, technology and team capabilities evolve in the course of a project. The development is highly iterative and incremental, and new features are continuously released. There are several proposals on how to tackle cost estimation for this particular type of projects [5,12,15,18,19], and story points are one of the most popular approaches among them. We selected it because it offers a number of key advantages: it produces continually-updated estimates throughout the entire development life cycle, it does not make assumptions on a particular work breakdown structure, involves the entire development team, and relies on prior known information acquired from previous iterations of the same project.

In the knowledge engineering area, cost estimation has received comparatively less attention. In our previous work we have introduced ONTOCOM, which estimates the costs of developing ontologies [17,16]. ONTOCOM is based on similar premises as the software-engineering approaches just mentioned, thus not addressing highly evolving, open development scenarios which are specific to folksonomy creation. Other proposals have emerged in the context of ontology reuse [2], semantic wikis [21], Semantic Web Services [22], and knowledge-based systems [7], providing either quantitative methods which typically require calibration based on historical data, or analytical considerations which have not been proven empirically. Furthermore, the procedural models they assume (implicitly or explicitly) are not compatible to folksonomy creation, which, as already mentioned, shows similarities rather with agile software engineering.

3.2 Basic Idea and Assumptions

The story-points approach is based on two core parameters: (i) the future workload (the so-called "user stories") expressed in imaginable units of complexity (termed "story points"), and (ii) the skills of the development team (termed "velocity"). The number of story points are estimated collaboratively within the development team; the velocity is measured in the course of a controlled experiment once the total number of story points is determined.

The estimated effort delivered by the story-points method is given in "ideal time". The ideal time denotes the amount of time that something takes when all peripheral activities are stripped off [4]. Additional costs – for instance related to technical infrastructure, system administration staff, and training – are not taken into account. Schedules can be derived from the effort estimates, provided information about the team productivity is available.

Each task in the project is assigned a number of story points, accounting for the impact of the task – or specific features thereof – on the overall development effort. Examples of such features are the size, the complexity, and the risk of the task. Story points are *relative* measures, in the sense that a ten-point story should be twice as large, complex or risky as a five-point story, and half as large, complex or risky as a twenty-point story. They provide a consistent variable that, together with the "velocity" of the team, provides a projection of when the target functionality will be delivered – or what functionality will be completed at a specific deadline. There are various guidelines and best practices on how to optimally assign story points to stories in an agile project [5,18]. Most of the them involve the entire development team, and some Delphi-like methodology to foster effective consensus-finding [10].[5] Empirical findings recommend the usage of the Fibonacci sequence $(1, 2, 3, 5, 8, 13, \cdots)$ or powers of 2 $(1, 2, 4, 8, 16, \cdots)$ in order to facilitate effective and consistent estimations.

The velocity is determined through average productivity measurements within a "project iteration". The effort estimated for the remainder of the project can then be computed based on the total number of story points divided by the velocity. The theoretic principle underlying this formula is that the sum of the independent samples from any distribution converges towards a normal distribution. Thus, the velocity measurements from one iteration form an adequate basis for predicting the velocity of future iterations [3]. The method can be applied at various stages of the project, once its two core parameters are determined.

3.3 Example

We will illustrate the usage of the story-points method through a simple example. Assuming we would like to estimate how much time it will take to clean our apartment. The apartment consists of a living room, a bedroom, a bathroom, and a kitchen, whereas the size of the bedroom and of the kitchen are 60% the size of the living room, and the bathroom is half the size of the kitchen. According to the story-points method we first have to assign each individual room a number of story points, reflecting the relative "complexity" of the cleaning job. The dimensions of the rooms are likely to be an important relevant in this context, the furnishing as well. Based on such considerations, we come up with the following estimates for the four rooms previously mentioned: living room (5), kitchen (4), bedroom (3), bathroom (2). The kitchen story points are arguably more than 60% of the story points assigned to the living room, as the size of the room is not the only factor to take into account here; kitchens tend to be more complex on average to clean due to the high number of appliances and alike. The total number of story points is thus 14.

To determine the value of the velocity parameter one would have to measure the average time spent in cleaning, for instance, the bathroom. If cleaning the bathroom (accounting for 2 story points) takes one hour, we can estimate that the rest of the apartment will be finished after 6 more hours of work (for $14 - 2 = 12$ story points). Of

[5] See, for instance, `http://kanemar.com/2006/01/28/story-points-as-spicy-ness-using-rsp-to-estimate-story-points/` and `http://www.planningpoker.com/`

course, we can improve the accuracy of this projection by performing further measurements later in the process. If we see that, for instance, cleaning the bedroom took two hours, we can adjust our average velocity parameter based on this new evidence, and obtain a better time prediction for the bedroom and the living room.

We now turn to applying the story-points method to folksonomy creation.

4 FOLCOM: Applying Story Points to Folksonomy Creation

Our aim is to design a method that predicts the time that is cumulatively invested by a community of users in tagging a collection of information objects. Taken into account the folksonomy creation aspects discussed in Section 2, it is expected that this effort will depend on (i) the characteristics of the collection of objects to be tagged, such as the number of objects in the collection, and the complexity of the tagging task for particular types of objects; (ii) the number of tags assigned to each object by each user (single- vs multi-tagging); (iii) the degree to which the tags are assumed to be consensual, thus implying additional overhead (collaborative tagging); and (iv) the size and dynamicity of the tagging community.

The scenario investigated in our work can be summarized as *"tagging a collection of information objects"*. This scenario is certainly simple. Still, it is representative for a wide range of Web 2.0-based knowledge management applications, and allows us to design a baseline cost estimation approach for folksonomies, which will be adjusted and extended to more advanced tagging scenarios as part of our future work. Examples of such advanced scenarios include collaborative tagging, collections of information objects of various modalities and complexity, or folksonomy maintenance in terms as, for instance, tag mapping and tag cleansing activities.

The estimates, just as for the original story-points method, are in terms of "ideal time". It is assumed that the time spent for activities immediately associated with tagging can be monitored. A GOMS[6]-like analysis of folksonomy creation, in which tagging is subdivided into interaction costs, such as mouse clicks, button presses, and typing, and attention switching costs – moving attention from one window to another – can be applied for this purpose [9]. It is also assumed that a tagging tool providing users with an interface to assign tags to information objects is available. This tool should be used by a representative sample of the folksonomy contributors in a project iteration in order to determine the tagging velocity. Ideally, it should include functionality for logging the tagging time; alternatively, one could use a stopwatch to measure it.[7]

4.1 Algorithm

FOLCOM consists of three steps that are executed in sequential order: (i) story-points estimation; (ii) velocity measurement; and (iii) effort estimation.

[6] `http://en.wikipedia.org/wiki/GOMS`

[7] If the technical support changes – for instance, new features are added to the tagging interface – the velocity parameter needs to be re-estimated. The total number of story-points stays the same.

Story-points estimation. First one estimates the total number of story points associated to creating a folksonomy describing and annotating a collection of information objects. Each object in the $Collection := \{o_1, o_2, \cdots, o_n\}$ represents a tagging "story" and the number of story points of the collection is calculated cumulatively. To estimate these values effectively, one typically builds groups of similar objects according to their types and characteristics. One dimension is certainly the modality of the content (textual documents, images, videos), a second, orthogonal dimension is the size of the information object (expressed in modality-specific metrics such as number of words in a document, length of a video). Other aspects which could be taken into account are, for instance, multi-linguality or familiarity with the content. Independently of these considerations, it is important to understand story points as relative measures of complexity. They stand for challenges associated to accessing, reading, viewing, browsing and comprehending the content of an information object, and identifying tags that meaningfully reflect it. In the following, $complexity(o)$, denotes the function which returns the complexity value of object o assigned by the estimator in this first step.

As soon as each object has got its size/complexity value, the story points for the whole object collection sp_{col} can be computed as the sum of the complexity values of all the objects in the collection.

$$sp_{col} := \sum_{i=1}^{n} complexity(o_i) \tag{1}$$

where $n := |Collection|$ is the number of objects in the collection, and $o_i \in Collection$. In case objects are grouped in $Groups := \{g_1, g_2, \cdots, g_n\}$, the computation can be simplified by multiplying the complexity values of each group with the number of objects in the group and then adding up these values.

$$sp_{col} := \sum_{i=1}^{n} (complexity(g_i) * |g_i|) \tag{2}$$

where $n := |Groups|$ is the number of groups, $g_i \in Groups$, $complexity(g_i)$ returns the complexity value of group g_i, and $|g_i|$ returns the number of objects in group g_i.

Velocity measurement. Velocity relates time information to story points (e.g., 2 minutes per story point), therefore allowing to map the project size expressed in story points to effort. Typically not all members of the community contributing to a folksonomy are known in advance; for estimation purposes, however, one has to select a representative share of this community, for instance based on the types of skills and expertise which are beneficial (or expected to be available) for each group of information objects.

During a project iteration the time invested by all users in tagging-related activities, in other words in adding, removing and changing tags, is measured. As discussed earlier in the paper, peripheral activities are not taken into account. The $Samples_{it} := \{s_1, s_2, \cdots, s_n\}$ gathered during this iteration are triples of the type $sample := (o, taggingTime, user)$, where o denotes an object in the collection which was tagged during the iteration, $user$ is the user who tagged o and $taggingTime$ is the time $user$ needed for tagging o.

The total tagging effort $totalEffort_{it}$ is computed by adding up the individual tagging times for all samples:

$$totalEffort_{it} := \sum_{i=1}^{n} (taggingTime_i) \tag{3}$$

where $n = |Samples_{it}|$ is the number of samples, and $taggingTime_i \in TaggingTimes_{it}$. Here $TaggedObjects_{it}$ represents all information objects tagged during the iteration, and the multi-set $TaggingTimes_{it}$ represents the tagging times measured for each object, tag and user.

In a folksonomy where each object is tagged exactly by one user (i.e., single-tagging), the calculation of the completed story points value sp_{it} is done via the following formula:

$$sp_{it_{single}} := \sum_{i=1}^{n} (complexity(o_i)) \tag{4}$$

where $n = |Samples_{it}|$ is the number of samples and $o_i \in TaggedObjects_{it}$. For multi-tagging the formula considers how many times each object has been tagged:

$$sp_{it_{multi}} := \sum_{i=1}^{n} (complexity(o_i) * times\_tagged(o_i)) \tag{5}$$

where $n = |Samples_{it}|$ is the number of samples, $o_i \in TaggedObjects_{it}$ is an information object tagged during the iteration, and $times\_tagged(o_i)$ returns the number of users tagged the object o_i during the iteration.

The velocity is then calculated as the total effort spent per iteration divided by the total number of story points.

$$velocity := totalEffort_{it}/sp_{it} \tag{6}$$

where sp_{it} is $sp_{it_{single}}$ for single-tagging or $sp_{it_{multi}}$ for multi-tagging.

Additionally, a factor $multiTagFactor$ must be computed, which captures the increase in value of one story point due to the possibility that a single object can be tagged by multiple users:

$$multiTagFactor := sp_{it}/sp_{it_{single}} \tag{7}$$

Alternatively multi-tagging could be modeled as the average number of tags assigned to an information object as in the formula 8 This, however, does not consider the different levels of complexity of specific groups of information objects.

$$multiTagFactor := |Samples_{it}|/|TaggedObjects_{it}| \tag{8}$$

where $|Samples_{it}|$ is the number of samples gathered during the iteration (each sample corresponds to one user which tagged an object) and $|TaggedObjects_{it}|$ is the number of objects tagged in the iteration.

Effort estimation. To estimate the effort to be invested to complete the project, one first determines the remaining number of story points using formula 9.

$$sp_{rem} := sp_{col} - sp_{it} \tag{9}$$

The effort estimate is then calculated as the number of story points multiplied by the velocity measured in the previous step.

$$effortEstimation_{rem} := multiTagFactor * sp_{rem} * velocity \tag{10}$$

For single-tagging, the $multiTagFactor$ in formula 10 is equal to 1. For multi-tagging scenarios one uses formula 7.

As the community who creates the folksonomy evolves over time, both the multi-tagging factor and the velocity are likely to change, as contributors will improve their tagging skills. The second step of the method should be repeated at regular intervals to compensate for these changes. The story-points estimation needs to be revisited only if the collection of information objects radically changes – for instance, by adding new types of content or information objects which significantly vary in their tagging-related complexity.

4.2 Experimental Evaluation

FOLCOM was evaluated on a slightly adapted version of the ONTOCOM evaluation framework [17] as listed in Table 1.

The evaluation of the non-calibrated method met was performed by conducting face-to-face structured interviews with eight knowledge management experts from three large-scale corporations in the sectors telecommunications and operators, ICT consultancy, and software development. Three of the participants of business managers with an extensive background in enterprise knowledge management; the other participants were technical consultants and IT practitioners who have been actively developing

Table 1. The FOLCOM evaluation framework

No	Criterion	Description
1	Definition	- clear definition of the estimated and the excluded costs - clear definition of the decision criteria used to specify the cost factors - intuitive and non-ambiguous terms to denominate the cost factors
2	Objectivity	- objectivity of the cost factors and their decision criteria
3	Constructiveness	- human understandability of the predictions
4	Detail	- refers to the work breakdown structure used by the method, not applicable
5	Scope	- usability for a wide class of tagging scenarios
6	Ease of use	- easily understandable inputs and options - easily assessable ratings based on the decision criteria
7	Prospectiveness	- applicability early in the project
8	Stability	- small differences in inputs produce small differences in outputs
9	Parsimony	- lack of highly redundant cost factors - irrelevant factors
10	Fidelity	- reliability of predictions

knowledge management solutions. Participants were given a one hour overview of the FOLCOM approach, followed by the individual interviews covering the quality criteria of the framework previously mentioned. This part of the evaluation resulted largely in positive qualitative feedback, and we summarize the most important findings in the following:

Definition. One expert remarked that it is not totally clear how specific characteristics of the tagging scenario, be that with respect to the artifacts being tagged or the community of users, are influencing the parameters of the method. In particular, the issue of tag quality was identified as particularly important and will be taken into account in future versions of the model in the velocity determination formulas. More extensive experiments covering larger, more heterogeneous collections of information objects will lead to a refinement of the story-points-estimation guidelines summarized in Section 4.1, which have been created in response to these comments.

Objectivity. Experts requested additional clarification about the rationales to use the Fibonacci sequence or powers of 2 as story points scales. The scales should provide a framework for effective and consistent estimation; based on empirical findings in agile software development, confirmed by our own user experiments, a higher level of precision is typically neither possible, nor required to deliver accurate predictions.

Constructiveness. The experts agreed that the predictions of the method can be understood and reproduced by its users.

Scope. The applicability to arbitrary tagging scenarios is one of the main advantages of our method. The method does neither depend on the object domain, nor on the tagging interface. These aspects were appreciated by the evaluators.

Ease of use. Inputs and options of the method were easily comprehended by all experts, though concerns were raised with respect to estimating story points for heterogeneous collections of objects. We have as a result extended our method to cover groups of objects reflecting various modalities, however more user experiments would be needed to obtain a better understanding of tagging challenges in general, and to compare the complexity of this task for text, audio, images and video. This could be achieved, for instance, through an analysis of the data collected in approaches such as Games with a Purpose.[8]

Prospectiveness. There were no special concerns regarding prospectiveness as the description of our method clearly states that FOLCOM can be applied throughout a project once data from a project iteration is available.

Stability. There were no concerns regarding the stability criterion from the expert team. As shown in the experimental evaluation, the quality of the predictions improves with larger samples of tagging time data.

Parsimony. No redundant cost factors were identified.

Fidelity. This aspect was evaluated during a series of six user experiments, which are discussed in the following.

[8] http://www.gwap.com/

Experimental setup. Our experiments were based on the same collection of 200 images collected through Web crawls tagged in single-tagging mode. Tags were assigned to images with the help of self-developed folksonomy tool, which included time logging and auto-completion features. Each experiment involved at least 30 participants, who were asked to perform tagging tasks randomly assigned to them – we assigned images to participants until every image in the object collection of an experiment was tagged successfully.

Table 2 lists the six user experiments including the experiment id, the tagging interface used, the number of images in the experiment object collection, and the maximum number of recommended tags per object.[9]

Table 2. Experimental setup

Experiment	Tagging interface	Number of images	Maximum number of tag recommendations
beta	t2t	100	0
gs1	t2t	200	0
oc1	t2t+ac	200	0
mm1	t2t+ac+tr	200	20
gs2	t2t+ac+tr	200	20
pw1	t2t+tr	200	20

The outcomes of the experiments. To measure the accuracy of FOLCOM, we compared the actual effort value, directly measured by our folksonomy tool during the experiment, with the estimates delivered by our method based on a number of 30 tagging samples measured automatically during the experiments. The estimation error is defined as the difference between the actual and the estimated effort values.

Table 3. Experimental results

Experiment	Actual total effort	Estimate after 30 samples
beta	39.05 minutes	38.28 minutes
gs1	71.27 minutes	71.33 minutes
oc1	60.77 minutes	70.11 minutes
mm1	63.33 minutes	92.11 minutes
gs2	52.22 minutes	57.00 minutes
pw1	40.92 minutes	48.00 minutes

Figure 1 presents the estimation errors of our method (in percent) for each experiment for the first 30 tagging-time long entries. The only estimates within an error margin larger than 25% were observed in experiment *mm1*. This behavior can be traced back to an above average tagging time for images 15 to 50 in this particular experiment.

[9] The abbreviations used for the tagging interfaces are: t2t (type to tag, manual tagging) tr (tag recommendation, users can accept a recommendation by typing in the corresponding tag), ac (tag auto-completion). For the most complex interface covering all three features we performed two experiments with different sets of users in order to increase the accuracy of the observations.

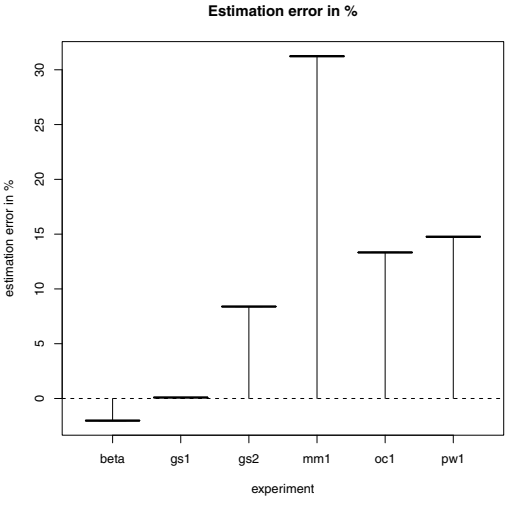

Fig. 1. FOLCOM's accuracy based on the first 30 tagging time samples

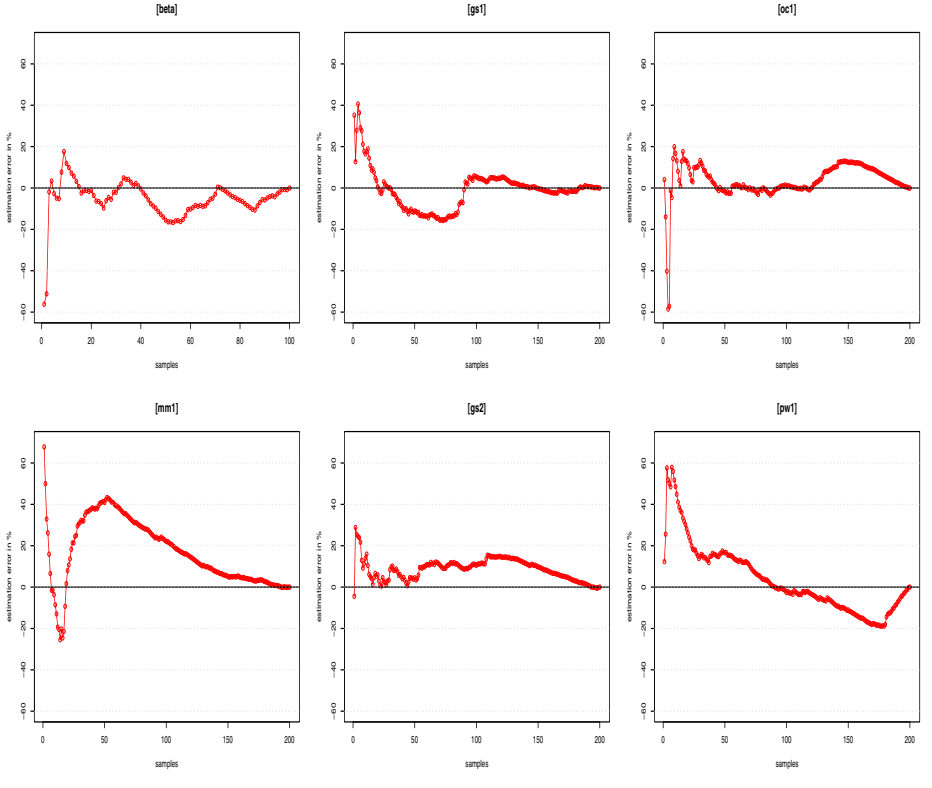

Fig. 2. Method prediction accuracy

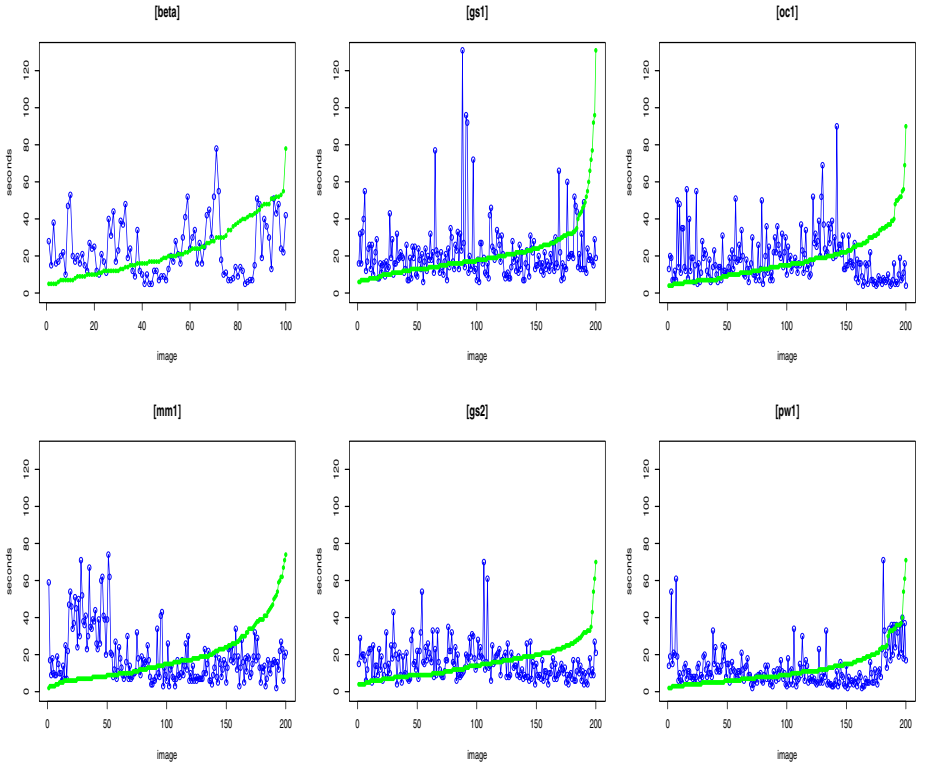

Fig. 3. Tagging times

It is furthermore interesting to see how the estimation accuracy of the method varies in relation to the number of samples taken as input. Figure 2 illustrates this relation: the x-axis displays the number of samples used to compute the estimates, and the y-axis shows the corresponding estimation error in %.

The fluctuations in Figure 2 can be explained by analyzing the tagging times displayed in Figure 3, where the x-axis of denotes the number of images and the y-axis the corresponding tagging times. The blue line corresponds to the tagging times as they were chronologically measured during the experiments. The green line plots the same tagging times, but this time in ascending order. The sum of tagging times gives the actual total effort required to tag the entire collection of images within an experiment. Since our method derives the estimates from the average of the tagging time samples given as input, a significant deviation of the average of the given samples from the average of the tagging times of the overall experiment leads to a bad estimate. This also explains the rather slow adaptation of the method's estimates to the sample fluctuations. A good example is experiment *mm1*. As illustrated in Figure 3 the tagging times for images 15 to 50 are relatively high; leading to a higher prediction error as demonstrated in Figure 2.

The experiments reveal an adequate prediction accuracy within a margin of $\pm 25\%$ in 75% of the cases. This is an indicator that the method could be reliably applied in productive environments of larger scale and diversity, though a more in-depth study of the specificities of enterprise tagging is surely needed in order to substantiate these preliminary positive results. Aspects which are likely to be of relevance include information objects such as Word documents (of tens to hundreds of pages), slides, tables and databases, but also the influence of classification practices based on controlled vocabularies and taxonomies, and in relation to incentives, the quality of the contributions.

5 Conclusions

The sustainable adoption of Web 2.0 technologies by the industry depends on the availability of reliably instruments to predict and analyze their costs and benefits, as well as on a critical level of commitment at the management level, a compatible corporate culture, and appropriate incentive schemes supporting enterprise-wide user involvement. In this paper we presented FOLCOM, a story-points-based method to predict the costs of tagging. To the best of our knowledge, this is the first folksonomy cost estimation method available so far.

The method has been evaluated by eight knowledge management experts experts according to several evaluation criteria with positive results. Furthermore, we conducted six different tagging experiments, in which the method was able to predict the effort with sufficient accuracy (within the 25% error margin). While more comprehensive experiments are needed to increase the reliability of the method, these first findings indicate that the approach works and is able to provide accurate estimations. The tagging experiments were also used to compare different tagging interfaces with different tagging features. The results hint at the fact that the tag recommendation feature can reduce tagging times per word in general and hence improve tag production. The autocomplete feature, however, seemed to be rejected by the users and/or did not lead to any positive tagging effects.

In the near future we will continue to evaluate FOLCOM along two dimensions. One of them is surely multi-tagging. Estimating the efforts implied by creating broad folksonomies is more complicated, since it involves a multi-tag factor. The behavior of this factor over time is largely unknown and additional empirical evidence is needed to determine it. In addition, more experiments are needed to allow for a more careful analysis of the types of automatic tag recommendation functionality and their effect on tagging costs. Finally, FOLCOM should take into account the quality of tags created by users, as as additional parameter to be taken into account when determining the velocity parameter.

Acknowledgements

The research leading to this paper was partially supported by the European Commission under the contract FP7-215040 “ACTIVE”.

References

1. Bürger, T., Popov, I., Simperl, E., Hofer, C., Imtiaz, A., Krenge, J.: Calibrated predictive model for costs and benefits. Deliverable D4.1.2, ACTIVE (February 2010)
2. Cohen, P.R., Chaudhri, V.K., Pease, A., Schrag, R.: Does prior knowledge facilitate the development of knowledge-based systems? In: AAAI/IAAI, pp. 221–226 (1999)
3. Cohn, M.: User Stories Applied For Agile Software Development. Addison-Wesley, Reading (2004)
4. Cohn, M.: Agile Estimating and Planning. Robert C. Martin Series. Prentice Hall PTR, Englewood Cliffs (November 2005)
5. Cohn, M.: Agile Estimation and Planning. Prentice-Hall, Englewood Cliffs (2005)
6. Cook, N.: Enterprise 2.0: How Social Software Will Change the Future of Work. Gower Publishing Ltd. (2008)
7. Felfernig, A.: Effort estimation for knowledge-based configuration systems. In: Proc. of the 16th Int. Conf. of Software Engineering and Knowledge Engineering SEKE 2004 (2004)
8. Heymann, P., Garcia-Molina, H.H.: Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford InfoLab (2006)
9. Hong, L., Chi, E., Budiu, R., Pirolli, P., Nelson, L.: Spartag.us: a low cost tagging system for foraging of web content, pp. 65–72. ACM, New York (2008)
10. Linstone, H.A., Turoff, M.: The Delphi Method: Techniques and Applications. Addison-Wesley Educational Publishers Inc., Reading (1975)
11. Martin, R.C.: Agile Software Development. Principles, Patterns, and Practices. Prentice-Hall, Englewood Cliffs (2002)
12. McConnell, S.: Software Estimation: Demystifying the Black Art (Best Practices (Microsoft)). Microsoft Press, Redmond (2006)
13. McKinsey. Building the web 2.0 enterprise: Mckinsey global survey results (July 2008)
14. Morrison, J.: Tagging and searching: Search retrieval effectiveness of folksonomies on the world wide web. Information Processing & Management 44(4), 1562–1579 (2008)
15. Lavanya, R., Chandrasekaran, S., Kanchana, V.: Multi-criteria approach for agile software cost estimation model. In: Proceedings of the International Conference on Global Manufacturing and Innovation in Engineering, GMICIT (2006)
16. Simperl, E., Popov, I., Bürger, T.: ONTOCOM Revisited: Towards Accurate Cost Predictions for Ontology Development Projects. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 248–262. Springer, Heidelberg (2009)
17. Simperl, E., Tempich, C., Sure, Y.: ONTOCOM: A Cost Estimation Model for Ontology Engineering. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 625–639. Springer, Heidelberg (2006)
18. Steindl, C., Krogdahl, P.: Estimation in agile projects. Presentation at IBM Academy of Technology Best Practices in Project Estimation Conference (2005)
19. Stellman, A., Greene, J.: Agile Software Project Management. O'Reilly, Sebastopol (2005)
20. Van Damme, C., Coenen, T., Vandijck, E.: Turning a corporate folksonomy into a lightweight corporate ontology. In: Proceedings of the 11th International Conference on Business Information Systems, BIS 2008 (2008)
21. Völkel, M., Abecker, A.: Cost-benefit analysis for the design of personal knowledge management systems. In: Proceedings of 10th International Conference on Enterprise Information Systems (ICEIS 2008), pp. 95–105 (2008)
22. Wolff, F., Oberle, D., Lamparter, S., Staab, S.: Economic reflections on managing web service using semantics. In: EMISA, pp. 194–207 (2005)