UDC 004.89

A. Semchenko, Y. Oliinyk, O. Demydenko

SOFTWARE DEVELOPMENT PROCESS MONITORING METHOD BASED ON TEXTUAL DATA ANALYSIS

Abstract: A considerable number of IT projects fail due to problems with operating processes rather than a shortage of technical expertise. The publication is dedicated to the problem of adequate measurement and monitoring of the development process in software engineering. The goal is to improve monitoring of the software development process. It is proposed that the method of monitoring the software development process is based on the textual data analysis of the data emerging during the development process. The results of the experiments show that the use of the proposed method can speed up the monitoring of the software development process by 22%.

Keywords: software development metrics, development process monitoring, project management, semantic analysis, sentiment analysis.

Formulation of the problem

Flexible development methodologies based on Agile principles have become extensively spread. Feedback mechanisms are being actively introduced to adjust the development process and ameliorate understanding of what is happening in each particular iteration. In the paper, a broad scientific theoretical basis was implemented describing approaches to measuring development processes and end-products [1] [2].

Generally, specific metrics are represented to assess software processes or products, the values of which are monitored from time to time. Nonetheless, apart from metrics, there are multiple artifacts arising in the development process. Some of them contain a lot of interesting information about the software development process. For example, tasks and their discussion in the form of comments are stored in the task management system. At present, the scientific base is under-researched. There are no methods of monitoring the software development based on the analysis of the textual artifacts. This publication is devoted to this problem.

Publication analysis

Many studies and publications have been dedicated to the problem of measurement and monitoring in software engineering. The paper [3] reviews the basic principles and approaches to measurement in software engineering. This publication reconsiders the model of software measurements, describes the classification of software metrics, and proposes the principles that should be guided in carrying out measurements in software engineering.

The GQM [4] (Goal-Question-Metric) method is constructively used by NASA for creating its projects. This is a method of creating new software metrics during measuring software projects. An approach to control the process of development and analysis using software metrics is introduced in the book [5]. Publications [6-8] comprise methods description for statistical processing of the measuring results relating to the program metrics, namely aggregation, and normalization of accumulated data. Articles [9-10] provide information on the usage of semantic methods for processing text data and approaches to collecting such data (text mining).

Conversely, there are no methods allowing to monitor the software development process based on the analysis of the textual artifacts arising over the development process. As a consequence of that, there is a need to develop such type of method.

Purpose of the research

The research purpose is to increase the efficiency of monitoring the state of software development based on the analysis of text data.

Main part

To solve the problem that consists in absence of methods for monitoring the software development process based on the analysis of text data, it was decided to comply with the following tasks:

1. Review the literature, highlighting the major approaches to monitoring the software development process;
2. Develop a method for monitoring the software development process based on the analysis of the textual artifacts;
3. Conduct a study for revising the developed method reliability.

To provide a better overview of the software development process, it is suggested to use textual data volume accumulated during the development process. For example, textual data appear during the process of task discussion at Task Management Systems (TMS) [11] in the form of descriptions and comments.

The developed method of monitoring the software development process consists of the following main stages:

- Identify main discussed topics;
- Risk assessment for the task execution terms violation;
- Recognize the categories and subcategories among the terms under discussion.

The scheme of the method is shown in Fig. 1.

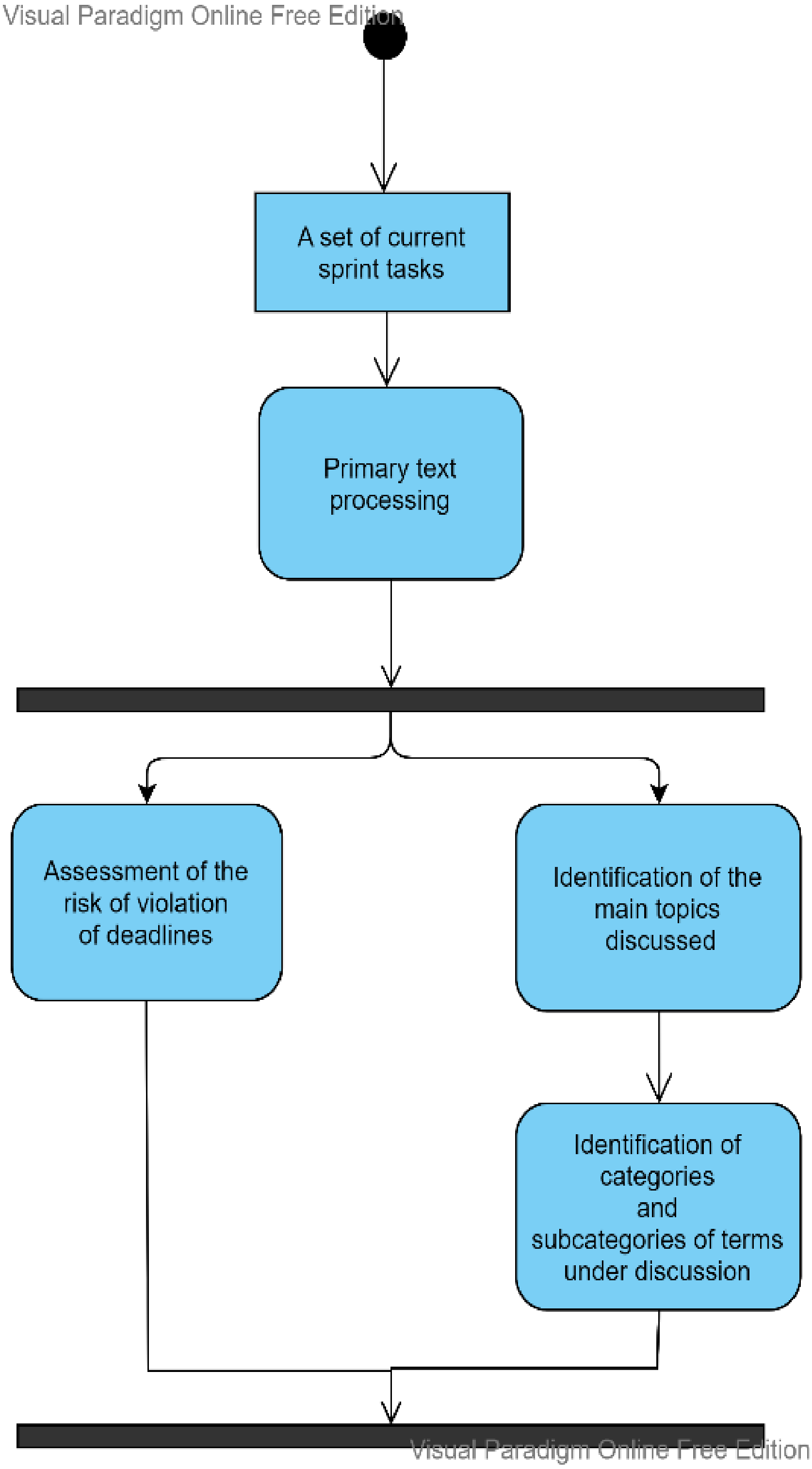

Visual Paradigm Online Free Edition
A set of current sprint tasks
Primary text processing
Assessment of the risk of violation of deadlines
Identification of the main topics discussed
Identification of categories and subcategories of terms under discussion
Visual Paradigm Online Free Edition

Figure 1. Method operational scheme

Detection of the main topics under discussion ensures to assess quickly what was discussed during the review of the current issue. To accomplish this, it is proposed to use latent-semantic analysis. Latent-semantic analysis (LSA) is a method of processing information in natural language, which allows to analyze the relationship between the document library and the found terms, as well as to identify the area-specific subject for documents. This method is quite successfully used to introduce knowledge bases [12] and to build cognitive models [13].

The method of latent-semantic analysis consists of several steps: preparation of document corpus, construction of a therm-documentary matrix as well as identification of dependencies. Preparation of document corpus is aimed at breaking each text into tokens, eliminating "noise" token conjunctions, and punctuation marks, and reduction of words to their original form (normalization). Subsequently, document-term metrics are built, the rows of which are documents and the columns are the words that are found there. The metrics element is a certain weight of the word in the corpus documents. TF-IDF metrics are usually used to calculate the weight.

$$tf - idf(r, d, D) = tf(t, d) \times idf(t, D)$$

$$tf(t, d) = \frac{n_t}{\sum_k n_k} idf(t, D) = \frac{|D|}{|\{d_i \in D | t \in d_i\}|}$$

As the next step, the singular value decomposition [14] of the obtained matrix is conducted, allowing to clarify hidden relationships between documents. This approach permits the identification of semantic relationships between terms in documents and clustering terms to facilitate clarifying the main subjects within documents.

Using this approach, it becomes possible to reveal the main subjects that were up-to-date at the current stage of the software development process, extracting this information from the text issues and comments on them.

During the process of performing tasks, an employee may face difficulties regarding the impossibility of successful task accomplishment or violating task deadlines. Or a conflict situation may arise between employees, which creates a risk of violating task deadlines. In both cases, problem discussion is characterized by abnormal emotional indicators.

An intervention of the project manager in the process of performing such a task saves time and reduces the risks related to the delays in the assignment execution. That's why there is a need to evaluate the risk of violating deadlines. It is proposed to use sentiment analysis to detect such types of problems.

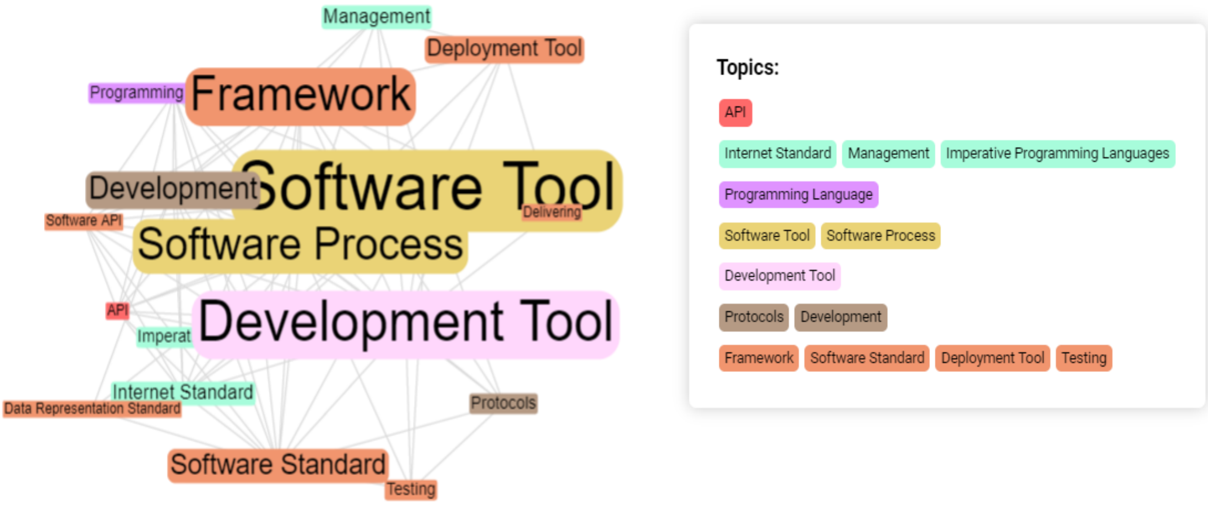


Figure 2. Identified subjects by LSA method

The tone of the comments may change depending on the task state in the process of task discussion. Furthermore, the last comments are more relevant for risk assessment than the first ones. For example, employees faced difficulties at the start of task execution. That's why the text of the first comments had a clear negative emotional value at the beginning of the task. However, after some time the problem was solved and the tone of the comments changed to neutral-positive. This aspect should be considered during assessing the risks of late performance.

Therefore, it is proposed to calculate a weighted integral tone criterion of each textual document related to the assessment to evaluate the risk of the late task.

$$= \frac{2}{N * (N+1)} \sum_{i=1}^{N} i * c_i,$$

where c_i – tone indicator of the document number i , N – number of documents, R – integrated indicator

Effectiveness evaluation

The following metrics were developed to evaluate the effectiveness

- speed of problem detection;
- speed of the interconnected tasks detection.

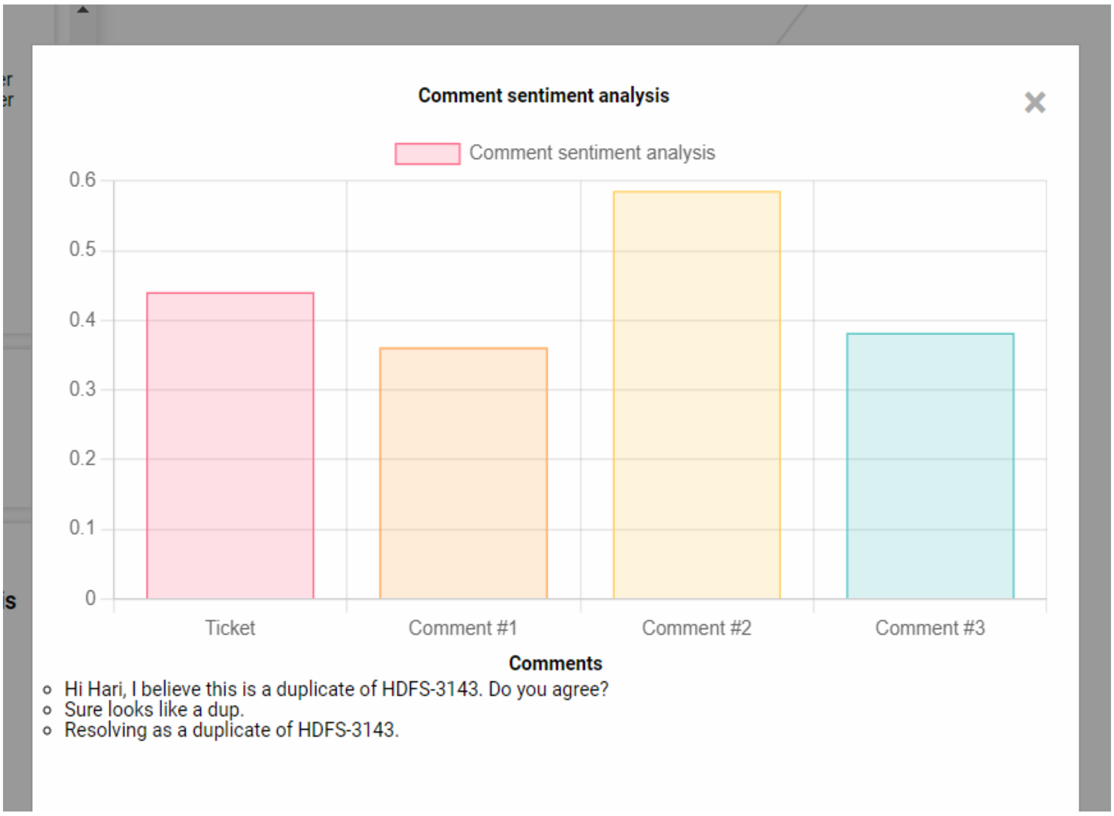
To assess the current state of project development, two experts working in the field of project management at the “Netcracker” company, who have long-term (7 years or more) project delegating experience were invited.

Table 1.

The quantitative scale of risk

Risk level	**Description**	**The value of the integrated indicator**
Probable	Failure or violation of deadlines for the specific task occurs in most cases	[-1; 0)
Usual	Violation of deadlines does not usually occur	[0; 0.35]
Low	Failure or violation of deadlines almost does not occur	(0.35; 1]

Figure 3 shows an example of sentiment analysis used to assess the risk of failure.



Comments

- Hi Hari, I believe this is a duplicate of HDFS-3143. Do you agree?
- Sure looks like a dup.
- Resolving as a duplicate of HDFS-3143.

Figure 3. Window with a diagram of the task sentiment analysis

A series of two experiments were conducted, in the framework of which an expert in the role of project manager will try to assess the current state of project development, using:

– ordinary Kanban board with a list of tasks;

– developed method for monitoring the software development process based on the analysis of the textual artifacts.

In the next step, we compare both methods according to the above-mentioned evaluation criteria.

Table 2.

The results of the experiment with identifying problems

Experiment	Expert	Method of examination	Time spent on examination
Project 1	Expert 1	Kanban-board	145 minutes
		Developed method	53 minutes
	Expert 2	Kanban-board	163 minutes
		Developed method	68 minutes
Project 2	Expert 1	Kanban-board	138 minutes
		Developed method	49 minutes
	Expert 2	Kanban-board	155 minutes
		Developed method	69 minutes

For the first experiment, 63 tasks were taken from the projects [15-16], comprising 1245 comments. The average number of comments on the task is 19.7. The experiment results indicate that the usage of the developed method accelerated the risk assessment of failure by 2.7 times for the first expert and by 2.3 times for the second expert.

For the second experiment, 78 tasks were taken from the projects [15-16], which contains 1064 comments. Thus, the average number of comments over the task is 13.6. The experiment results show that the usage of the developed method advanced the task risk assessment by 2.8 times for the first expert and by 2.2 times for the second expert.

Then a similar experiment to measure the time required by a project manager to identify related tasks (those that have similar content or that discuss the same topic) was conducted.

In the framework of this experiment the expert being in the role of project manager reviews the list of tasks and their content (text of the description and comments) to reveal interrelated tasks (issues). The ultimate purpose of the project manager is to understand the structure of the relationships between the tasks and the topic list to which each task belongs. Firstly, an experiment identifying interrelated tasks is performed with the help of a classic Kanban board. Then, a similar experiment was performed using the developed method.

Table 3.

Results of the experiment of identifying interrelated tasks

Experiment	**Expert**	**Method of examination**	**Time spent on examination**
Project 1	Expert 1	Kanban-board	179 minutes
		Developed method	136 minutes
	Expert 2	Kanban-board	190 minutes
		Developed method	153 minutes
Project 2	Expert 1	Kanban-board	213 minutes
		Developed method	180 minutes
	Expert 2	Kanban-board	243 minutes
		Developed method	209 minutes

The first experiment indicates that the usage of the developed method accelerated the process of reviewing and identifying interrelated problems and their subjects by 1.3 times for the first expert and by 1.24 times for the second expert.

Conclusions

The method of monitoring the software development process based on the analysis of the textual artifacts is represented in the paper. It is proposed the application of mathematical methods for text processing in natural language ameliorates the monitoring of the software development process. The application of linguistic methods to process natural texts enables to recognition of tendencies (topics) appearing in textual artifacts which arise in the development process (issues-tickets and comments to them). The usage of sentiment analysis enables us to evaluate the risk of violating task deadlines. The application of such methods and tools authorizes to recollect and analyze data on the software development process, contributing to further research in the area of development process management. The results of the experiments show that the developed method usage speeds up the monitoring development process by an average of 22% compared to the usage of a conventional Kanban board.

REFERENCES

1. Simon Alexandre. Software Metrics An Overview / Simon Alexandre // University of Namur Software Quality Lab Belgium. – 2002. – URL: https://www.cetic.be/IMG/pdf/Software_Metrics_Overview.pdf.

2. Software Measurement: A Necessary Scientific Basis // IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. – 1994. – URL: https://www.ipd.kit.edu/mitarbeiter/padberg/lehre/sqs07/FentonTSE1994.pdf.

3. UNIQUE FUNDAMENTALS OF SOFTWARE MEASUREMENT AND SOFTWARE METRICS IN SOFTWARE ENGINEERING // International Journal of Computer Science & Information Technology. – 2015. – URL: http://www.airccse.org/journal/jcsit/7415ijcsit03.pdf.

4. Victor R. Basili. THE GOAL QUESTION METRIC APPROACH / Victor R. Basili, Gianluigi Caldiera, H. Dieter Rombach // FB Informatik Universität Kaiserslautern Kaiserslautern, Germany – URL: https://www.cs.umd.edu/users/mvz/handouts/gqm.pdf.

5. Alan J. Perlis. Software Metrics: An Analysis and Evaluation / Alan J. Perlis, Frederick Sayward, Mary Shaw., 1983.

6. A Study of the Effect of Data Normalization on Software and Information Quality Assessment / Morgan Ericsson, Welf Löwe, Tobias Olsson та ін.] – URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.497.3097&rep=rep1&type=pdf.

7. Normalization // Wikipedia. – 2022. – URL: https://en.wikipedia.org/wiki/Normalization_(statistics).

8. Karine Mordal. Software quality metrics aggregation in industry / Karine Mordal, Nicolas Anquetil, Jannik Laval // JOURNAL OF SOFTWARE: EVOLUTION AND PROCESS. – 2012. – URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.700.305&rep=rep1&type=pdf.

9. Thomas K Landauer. An Introduction to Latent Semantic Analysis / Thomas K Landauer, Peter W. Foltz, Darrell Laham. – 1998. – URL: http://lsa.colorado.edu/papers/dp1.LSAintro.pdf.

10. Yu. Oliynik. Review and analysis of algorithms TEXT MINING / O. Gavrilenko, Yu. Oliynik, H. Hanko. // Project management, systems analysis and logistics. – K .: NTU, 2017. - Vol., pp32-41

11. Mostafa Taha. Task Management System (TMS) / Mostafa Taha. – 2016. – URL: https://repository.najah.edu/handle/20.500.11888/12232.

12. Thomas K. Landauer. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge / Thomas K. Landauer, Susan T. Dumais – URL: http://www.welchco.com/02/14/01/60/96/02/2901.HTM.

13. B. Lemaire, G. Denhière. Cognitive Models based on Latent Semantic Analysis (неопр.) // Tutorial given at the 5th International Conference on Cognitive Modeling (ICCM'2003), Bamberg, Germany, April 9 2003.. — 2003

14. Singular Value Decomposition (SVD) tutorial – URL: https://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm.

15. Apache HADOOP issues – 2022. – URL: https://issues.apache.org/jira/projects/HADOOP/issues

16. WOOKIE Jira Tickets – 2022. – URL: https://issues.apache.org/jira/projects/WOOKIE/issues/.