WORK BEHAVIOR PREDICTION DURING SOFTWARE PROJECTS DEVELOPMENT

Ciprian-Leontin Stanciu
"Politehnica" University of Timisoara, Vasile Parvan Avenue, No. 2, 300223 Timisoara, Romania

Keywords: Software project management, Forecasting, Prediction model, Work progress metrics, Work behavior.

Abstract: Software projects are well known for their high overruns in terms of budget and time. As they grow larger, project monitoring becomes harder. In such a context, forecasting becomes a critical "ability", helping the project manager to understand where the project is heading, in terms of required budged and implementation time. In this paper, we present a forecasting method that we developed, which makes use of a distinct representation of the observed behavior of project team members towards work that is Work Behavior. Moreover, we present the first results from experiments on real-world software project development data that show that our method is more accurate than a very popular prediction method, implemented by most ALM tools.

1 INTRODUCTION AND BACKGROUND

Many popular ALM (Application Lifecycle Management) tool providers consider that, in certain application conditions, the forecasting capability is a must for such tools. Many ALM tools offer this capability, despite critics like those in (DeMarco, 2009) and (Wysocki, 2010).

Considering the existing project management prediction methods, we can state there are many methods destined to forecast the resources required by a project, also known as estimation methods (Project Management Institute, 2008), which are used at the beginning of the project, and which resulted from documented research. An example is the COCOMO suite of models (Boehm et al., 2005).

In the same time, there are just a few prediction methods that can be used during project development to support decision making. One is the Velocity Trend prediction which is part of the popular Scrum Agile framework (Deemer and Benefield, 2007), and which is offered in most ALM tools, such as CollabNet Team Forge (CollabNet, n.d.) and IBM Rational Team Concert (IBM, n.d.).

In this context, we are developing a monitoring framework for large-scale software projects, the Behavioral Monitoring Framework that has a prediction dedicated component model. We portrayed the monitoring framework and a part of its component models in (Stanciu et al., 2009), (Stanciu et al., 2010), and (Stanciu et al., 2010). A very interesting approach to historical information characterization that inspired us in the development of the Work Behavior Prediction method is presented in (Gîrba, 2005).

In this paper, we present the Work Behavior Prediction method, which is part of our Behavioral Monitoring Framework. The paper is organized as follows: section 2 describes the prediction method that we propose; section 3 presents the methodology used in the evaluation of our forecasting method; in section 4, the evaluation results are showed and discussed; finally, section 5 presents the conclusions of this paper.

2 WORK BEHAVIOR PREDICTION

The Work Behavior Prediction is a forecasting methodology that uses historical information in order to predict, at any time during project development, the remaining effort for a task at a chosen time in the future.

2.1 Definitions

Definition 1 (Remaining Effort). Remaining Effort**.**

Stanciu C..
WORK BEHAVIOR PREDICTION DURING SOFTWARE PROJECTS DEVELOPMENT.
DOI: 10.5220/0003505000470052
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 47-52
ISBN: 978-989-8425-76-8

for a task is an amount of work considered necessary to be spent for completing that task.

Definition 2 (History)**.** History (H) for a task is an chronologically ordered set of Remaining Effort reports, covering a subinterval of or the entire period of time between task start and task completion.

Definition 3 (Stagnation)**.** Stagnation (ST) is the probability that, given a History for a task, two consecutive History elements show the same Remaining Effort.

Equation (1) shows the Stagnation computed on a History H.

$$ST = P_{H(i)=H(i+1)} \tag{1}$$

Definition 4 (Diversification)**.** Diversification (DV) is the probability that, given a History for a task, exactly two of three consecutive History elements show the same Remaining Effort.

Equation (2) shows the Diversification computed on a History H.

$$DV = P_{H(i)=H(i+1)\neq H(i+2)} + P_{H(i)\neq H(i+1)=H(i+2).} \tag{2}$$

Definition 5 (Velocity)**.** Given a History for a task, Velocity (VL) is the mean difference between consecutive History elements' Remaining Effort.

Equation (3) shows the Velocity computed on a History H.

$$VL = \overline{H(\iota) - H(\iota+1)}. \tag{3}$$

Definition 6 (Work Behavior)**.** Given a History for a task, Work Behavior (WB) is a triplet of Stagnation, Diversification, and Velocity values computed for the given History.

Equation (4) shows the meaning of Work Behavior.

$$WB = (ST, DV, VL). \tag{4}$$

Definition 7 (Implementation Moment)**.** Given a History for an in-work task, the Implementation Moment (IM) is the number of History elements divided by the first History element's Remaining Effort.

Equation (5) shows the Implementation Moment computed on a History H. Please note that a first History element, H(0), of value 0 (meaning an initial Remaining Effort of 0 effort units) makes no sense.

$$IM = \frac{n}{H(0)}, n - size\ of\ H. \tag{5}$$

Definition 8 (Virtual Present)**.** Given a History for a completed task and an Implementation Moment of an in-work task, Virtual Present (VP) is the first History element's Remaining Effort multiplied by the given Implementation Moment.

In other words, Virtual Present is the position of a given in-work task's present in the History of a completed task. Equation (6) shows a Virtual Present computed on a History H of a completed task and for a given Implementation Moment IM of an in-work task.

$$VP = H(0) \times IM. \tag{6}$$

2.2 Methodology

The Work Behavior Prediction methodology is presented in Figure 1 and described next.

The prediction process starts with the selection of a project task to be the subject of prediction. This is the target task in Figure 1.

The tasks are represented as Histories, as defined in the previous subsection. This is why a time axis is shown for each task in Figure 1.

The target task has a History, named Known history in Figure 1. Based on this History, the target Work Behavior (WB_{target}) is computed.

As shown in Figure 1, completed tasks are used in the prediction process. These tasks actually represent a selection of completed tasks that have their assignee in common with the target task. Their histories characterize the behavior towards work of their assignee, this being a good reason for using their histories in the forecasting process.

For the target task, the Implementation Moment IM is computed. By using IM, the Virtual Present is computed for all the completed tasks selected for prediction (by using the definition of Virtual Present from the previous subsection, a VP is computed for each completed task in Figure 1). This way, the Histories of the completed tasks are split into two parts, so that the History for a task contains a History before the Virtual Present of that task, and a History after this Virtual Present. In case the History after the Virtual Present for a task contains no element (this is a possibility), that task is ignored in the prediction process.

For each History before VP in Figure 1, a Work Behavior is computed resulting a WB_{before}. In the same time, for each History after VP in Figure 1, a Work Behavior is computed resulting a WB_{after}.

The WB_{before} elements are than compared with WB_{target} producing a weight for each WB_{after} element, which will be further used in the prediction process.

The Work Behavior elements are compared by using Euclidean distance, considering the Work Behavior component metrics (ST, DV, and VL)

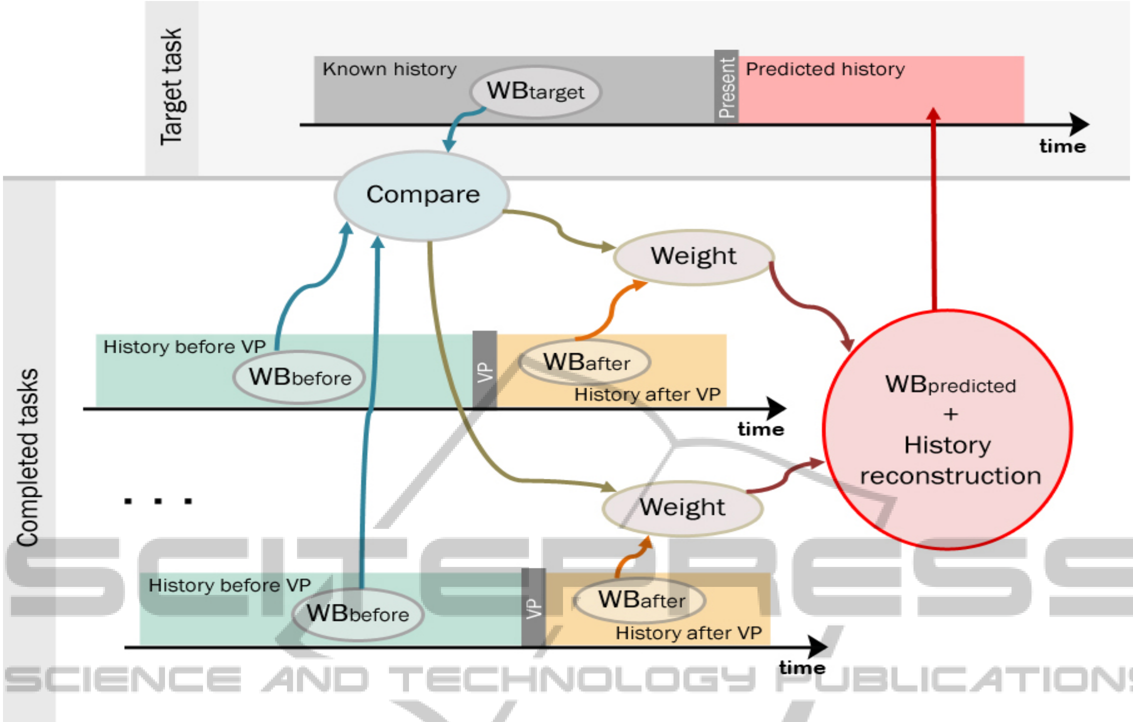

Figure 1: Work Behavior Prediction methodology.

as dimensions. The closest WB_{before} to WB_{target} produces the biggest weight for its twin, WB_{after}.

Next, the WB_{after} elements are weighted and combined for computing the predicted Work Behavior ($WB_{predicted}$ in Figure 1). A weighted mean is used in this process.

The Known history in Figure 1 is used along with $WB_{predicted}$ to build a History structure that corresponds to the predicted progress for the target task (Predicted history in Figure 1).

Ideally, the Histories used in the forecasting process should contain elements for equally distanced moments in time. If this is not the case, an extrapolation method on the existing data is used beforehand.

3 FORECASTS EVALUATION METHODOLOGY

This section presents the results of applying the Work Behavior Prediction method on real-world commercial software projects development data.

3.1 Competing Prediction Method

In this evaluation, the Work Behavior Prediction competes with Velocity Trend prediction, which is part of the Scrum management framework (Scrum Alliance, 2007).

3.2 Data used in Evaluation

In the forecasts evaluation, data retrieved from the development of two real-world software projects are used. These projects are not related and were developed by two different software development companies.

We refer to these projects as project X and project Y. Project X was developed by a project team of 23 members, while project Y (smaller than X), by a team of only 6 members. Further information cannot be provided at the moment due to confidentiality.

3.3 Metrics and Tools

We use several error metrics to assess the prediction quality. These metrics, along with their strengths and weaknesses are presented in (Zivelin, n.d.). In the following equations, D represents an observation, F is a forecast, and n is the number of (D, F) pairs.

The simplest metric is MFE (Mean Forecasting Error). Equation (6) shows how this metric is computed.

$$MFE = \frac{\sum_i (Di - Fi)}{n}. \quad (6)$$

Another metric used in this evaluation is MAD (Mean Absolute Deviation). Equation (7) shows how this metric is computed.

$$MAD = \frac{\sum_i |Di - Fi|}{n}. \quad (7)$$

The third metric used in this evaluation is MAPE (Mean Absolute Percentage Error). Equation (8) shows how this metric is computed. Although MAPE, also known as MMRE, is the most common measurement of forecast accuracy, it has an important weakness that is shown in (Foss, Stensrud, Kitchenham, and Myrtveit, 2002).

$$MAPE = \frac{\sum_i \left|\frac{Di - Fi}{Di}\right|}{n} \times 100. \quad (8)$$

The last metric used in this evaluation is WMAPE (Weighted Mean Absolute Percentage Error). Equation (9) shows how WMAPE is computed.

$$WMAPE = \frac{\sum_i \left|\frac{Di - Fi}{Di}\right| \times Di}{\sum_i Di}. \quad (9)$$

For analyzing the available project data, we developed a software prototype of our Behavioral Project Monitoring Framework. This software prototype has a forecasting module, implementing the Work Behavior Prediction method.

The project data is provided in the form of Microsoft Project Plan files which are available on a monthly basis for several months in the case of project X, and on a weekly basis for several weeks in the case of project Y.

The software prototype automatically computes the four metrics for all the project tasks for which data is available, so that an index i of D and F from equations (6), (7), (8), and (9) refers to one task.

4 RESULTS AND DISCUSSION

The forecasts evaluation results are presented in Table 1, for project X, and Table 2, for project Y. Table 1 and Table 2 show the prediction time span, which is measured in months, in the case of project X, and weeks in the case of the smaller project Y. The main reason for making predictions on such time spans was that project development data is available on a monthly-basis, in the case of project X, and on a weekly-basis, in the case of project Y. Consequently, forecasts at the end of the prediction time span can be compared to existing information regarding project progress.

The four metrics used in evaluation that were presented in the previous section, are computed for Velocity Trend prediction (VPT in Table 1 and Table 2) and for our prediction method, Work Behavior Prediction (WBP in Table 1 and Table 2).

A prediction method is considered better than the other for a case if at least half plus one of the available metric values are lower for the first method (considering, of course, the metrics that are used for this evaluation for which lower means better).

In Table 1 and Table 2, the cases in which our prediction method (WBP) is better than Velocity Trend prediction (VTP) are shaded.

Analyzing the results presented in Table 1 and considering all the available 24 presented cases, our prediction method (WBP) proves to be systematically better than Scrum's Velocity Trend prediction (VTP). The 1 month prediction time span shows the lowest differences between the two prediction methods. Even so, in 7 of the 8 cases our prediction method has a lower MFE, meaning that is more "on target" than the competing Velocity Trend method. The 2 month prediction time span shows better results for our prediction method in 6 of the 9 cases. For 3 month time span prediction, according to the metrics values, our prediction method is better in 6 of the 7 cases. The results presented in Table 1 suggest that, for long term prediction, considering the available information, our method is more appropriate to be used for decision support than the popular Velocity Trend prediction. For example, for case 17 (Table 1), using Work Behavior Prediction, the project manager knows two months ahead of time where project tasks will be in terms of work progress with an average absolute prediction error per task of only 10 working days (see MAD for case 17 in Table 1) meaning 2 calendar weeks. Applying Velocity Trend Prediction on the same data and for the same time span, the average absolute error per task is 35 working days, meaning one calendar month and a half, which almost equals the prediction time span.

Analyzing the results shown in Table 2 and considering all the available 10 cases, we conclude than our prediction method is better than Velocity Trend prediction for project Y also. For 1 week prediction time span, our method shows better results in 3 of the 4 cases. For the other prediction time spans (2, 3, and 4 weeks), our prediction method is better in all the cases.

Just like for project X, the results for project Y, which are presented in Table 2, suggest that, for long

Table 1: Evaluation results for project X (VTP – Velocity Trend Prediction; WBP – Work Behaviour Prediction).

Prediction time span	Case no.	WMAPE		MAPE		MAD [days]		MFE [days]	
		VTP	WBP	VTP	WBP	VTP	WBP	VTP	WBP
1 month	1	0.579	1.149	3.003	25.000	8.530	16.936	7.579	-6.302
	2	0.673	0.583	123.150	44.741	8.480	7.352	7.887	-3.229
	3	0.665	0.276	44.117	40.149	5.425	2.249	1.312	-0.796
	4	0.458	0.769	5.228	14.662	3.043	5.108	1.736	0.990
	5	0.577	0.616	20.224	21.754	13.170	14.073	4.409	-5.245
	6	0.683	0.692	27.281	25.658	11.560	11.711	8.635	-3.166
	7	0.501	0.305	40.469	7.843	10.199	6.212	5.614	0.040
	8	1.094	0.919	40.014	29.534	16.322	13.709	13.647	0.777
2 months	9	0.822	1.402	21.579	30.357	8.315	14.180	7.897	-1.670
	10	3.350	1.146	94.713	97.422	12.462	4.264	12.026	2.087
	11	1.669	1.026	10.881	7.491	6.964	4.282	3.180	1.079
	12	1.512	1.180	10.281	6.799	6.864	5.358	2.064	2.242
	13	0.752	1.047	25.561	26.972	13.892	19.357	7.562	-7.988
	14	1.481	1.079	65.402	17.527	16.919	12.246	15.022	-1.330
	15	2.345	1.563	7.584	9.833	19.276	12.845	18.667	7.138
	16	0.673	0.873	42.902	56.699	24.956	32.393	-8.456	-30.890
	17	10.932	2.928	521.888	170.305	34.393	9.211	34.393	7.843
3 months	18	9.714	5.027	84.660	25.755	12.993	6.723	12.993	6.323
	19	-	-	-	-	9.995	4.533	9.995	4.533
	20	2.122	1.358	13.808	9.539	8.169	5.229	2.229	0.008
	21	1.527	1.366	11.404	4.649	5.561	4.973	2.594	2.885
	22	1.014	1.103	66.473	30.956	16.589	18.048	10.192	-7.120
	23	4.971	2.231	6.492	5.106	23.575	10.583	23.575	3.268
	24	0.903	0.800	14.429	15.622	17.103	15.152	7.837	-7.315

Table 2: Evaluation results for project Y (VTP – Velocity Trend Prediction; WBP – Work Behavior Prediction).

Prediction time span	Case no.	WMAPE		MAPE		MAD [days]		MFE [days]	
		VTP	WBP	VTP	WBP	VTP	WBP	VTP	WBP
1 week	1	0.333	0.083	33.333	12.500	0.750	0.188	-0.250	-0.188
	2	0.250	0.000	25.000	0.000	0.750	0.000	-0.750	0.000
	3	0.657	0.791	98.886	221.694	1.557	1.876	-1.107	-0.676
	4	0.318	0.070	72.727	27.895	0.382	0.084	-0.382	-0.084
2 weeks	5	0.500	0.083	41.667	12.500	1.125	0.188	-0.875	-0.188
	6	0.375	0.000	37.5	0.000	1.125	0.000	-1.125	0.000
	7	0.393	0.382	63.750	127.323	0.412	0.401	-0.337	0.326
3 weeks	8	1.159	0.250	262.500	137.500	1.912	0.413	-0.587	0.413
	9	0.625	0.000	62.500	0.000	1.875	0.000	-1.875	0.000
4 weeks	10	1.235	0.407	229.167	146.139	2.038	0.672	-0.962	0.153

term prediction, our method is more appropriate to be used for decision support than the popular Velocity Trend prediction. For example, for case 5 (Table 2), using Work Behavior Prediction, the project manager knows two weeks ahead of time where project tasks will be in terms of work progress with an average absolute prediction error per task of only 0.2 working days (see MAD for case 5 in Table 2) meaning less than 2 working hours, considering that a full working day consists in 8 working hours. Applying Velocity Trend Prediction on the same data and for the same time span, the average absolute error per task is 1.2 working days, meaning almost 10 working hours.

Although we evaluated our prediction method, Work Behavior Prediction, only on two real-world software project development data, we believe the results are valuable in the context in which such project data is very hard to get, considering its confidential nature. Even for those two projects, according to Table 1 and Table 2, our method shows an evident superiority to a very popular prediction method, which is implemented by most ALM tools, Velocity Trend prediction.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a prediction method that we developed, named Work Behavior

Prediction.

This method is evaluated against a set of error metrics and the results are compared to those obtained for a very popular prediction method, Velocity Trend prediction. Real-world commercial software projects development data are used in the evaluation process.

As future work, we intend to enhance our prediction method by using it for more real-world software projects and by evaluating the obtained results. Furthermore, we will analyze the applicability of our prediction method for other types of projects (e.g. construction projects).

ACKNOWLEDGEMENTS

This work was supported by QuarterMill Technologies. This work was partially supported by the strategic grant POSDRU 6/1.5/S/13, (2008) of the Ministry of Labour, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in People.

REFERENCES

Boehm, B., Valerdi, R., Lane, J., and Brown, W. (2005). COCOMO Suite Methodology and Evolution. *In: Crosstalk, Vol.18 i4, pp. 20-25.*

CollabNet (n.d). CollabNet. Retrieved January 4, 2011 from http://www.open.collab.net.

Deemer, P. and Benefield, G. (2007). An Introduction to Agile Project Management with Scrum. Retrieved November 20, 2010 from *http://www.rallydev.com/documents/scrumprimer.pdf.*

DeMarco, T. (2009). Software Engineering: An Idea Whose Time Has Come and Gone? *In IEEE Software. Viewpoints, pp 94-95.*

Foss, T., Stensrud, E., Kitchenham, B., Myrtveit, I. (2002). A Simulation Study of the Model Evaluation Criterion MMRE. Discussion paper. *Norwegian School of Management BI. ISSN: 0807-3406.*

Gîrba, T. (2005). Modeling History to Understand Software Evolution. PhD. thesis, University of Bern.

IBM (n.d). IBM Rational Team Concert. Retrieved January 4, 2011 from *http://www.ibm.com.*

Project Management Institute (2008). A Guide to theProject Management Body of Knowledge (PMBOK® Guide) - *Fourth Edition. Project Management Institute, ISBN13:9781933890517.*

Scrum Alliance (2007). Glossary of Terms. Retrieved November 30, 2010 from http://www.scrumalliance.org/articles/39-glossary-of-scrum-terms.

Stanciu, C., Tudor, D. and Crețu V.I. (2009). Towards an adaptable large scale project execution monitoring. *5th International Symposium on Applied Computational Intelligence and Informatics, pp. 503 - 508, Romania.*

Stanciu, C., Crețu, V.I. and Cireş-Marinescu, R. (2010). Monitoring Framework for Large-Scale Software Projects. *Proceeding of the IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics 2010, pp. 333-338, Timisoara, Romania.*

Stanciu, C., Tudor, D. and Crețu V.I. (2010). Towards Modeling Large Scale Project Execution Monitoring: Project Status Model. Proceedings: 5th International Conference on Software and Data Technologies, *ICSOFT 2010, Athens, Greece, ISBN 978-989-8425-22-5, Volume 1, pp. 36-41. Portugal: SciTePress.*

Wysocki, R. K. (2010). Adaptive Project Framework: Managing Complexity in the Face of Uncertainty. Addison-Wesley Professional.

Zivelin, N. (n.d). Forecast Metrics and Evaluation. Oracle. Retrieved on December 27, 2010 from *http://demantrasig.oaug.org.*