

A Human-Centered Approach to Assessment of Team Diversity in Multi-Version Software Development

Serhii Lilikovych¹ and Volodymyr Sokol¹

National Technical University Kharkiv Polytechnic Institute, Kyrpychova str., 21,
Kharkiv, Ukraine 61002

{sergiy.lilikovych@gmail.com, vlad.sokol@gmail.com}

Abstract. The article proposes comprehensive approach to personal diversity level assessment of involved developers in order to improve efficiency of multi-version software development. The approach allows to build project teams with predefined level of diversity intentionally. A common conceptual scheme of the approach is elaborated, an appropriate criteria and metrics to estimate personal diversity of target software products diversity are given, and future research work is outlined as well.

Keywords: human-centered, diversity, multi-version software, criteria, metrics

Key Terms: ModelBasedSoftwareDevelopmentMethodology, Model, SoftwareSystem

1 Introduction: Problem Actuality and Research Aims

One of generally recognized approaches to improvement of software reliability by reducing a number of errors is usage of the diversification principle, which assumes development and integration of different versions of a target software product or its parts respectively [3]. Such project diversity is provided by usage of alternative design methods, different programming languages, tools combined to achieve difference between project versions. Such differences are called design diversity dimensions – DDD, and it is also possible to apply well-formalized criteria and mathematical methods for evaluation of the diversity level in final software products [7]. Recently it became clear that very important DDD-component is the human factor, namely: how a personal diversity, and therefore, project teams diversity, can be taken into account in multi-version software development (see e.g. in [1]). It is also worth to mention that exactly for these issues there are very few precise decision methods and support tools. That is why the actual purpose of our research is to develop a human-centered approach to assessment of personal diversity, and in this way to form diverse developer team (DDT) in multi-version software development. The rest of this paper is organized as follows: Section 2 introduces the common conceptual scheme of the proposed

approach, in Section 3 we outline first implementation issues for this one, and in Section 4 the paper concludes with a short summary, and with outlook on the future research work to be done.

2 Conceptual Scheme of Human-centered Approach to Assessment of Team Diversity in N-Version Programming

One of the proven techniques of software diversification is N-version programming (NVP). It assumes a development of $N \geq 2$ independent software versions, wherein all versions are supposed to satisfy the same initial system requirements. Taking into account one of the well-known NVP workflows [4], we propose to extend it with the DDTs forming procedure, as shown in figure 1 in the form of an UML activity diagram [5].

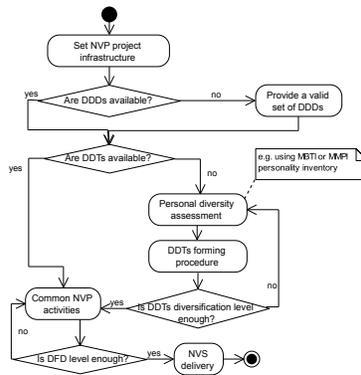


Fig. 1. NVP development with the DDTs forming procedure

The proposed approach consists of the following actions:

1. set up initial NVP projects infrastructure with selection of technological parameters for choosing appropriate DDD; the process continues until a valid set of DDDs is available (see figure 1);
2. prove the condition, whether a number of DDTs needed is available, where any DDT has to be considered as an additional dimension in DDD collection;
3. in case both DDD and DDTs are given, the common NVP-projects activities can be executed, and a Distinct Failure Diversity (DFD) of software product may be calculated using the appropriate metrics (see below);
4. if DFD value is acceptable (enough), then NVS product may be delivered, otherwise Step 3 has to be repeated;

5. If DDTs are not available by checking on Step 2, the personal diversity forming procedure has to be started in order to create new DDTs, and it requires Personal diversity assessment to be provided using appropriate methods: e.g., Minnesota Multiphasic Personality Inventory (MMPI), or Myers-Briggs Type Indicator (MBTI) [1].

The proposed DDT forming procedure is based on the hypothesis that certain properties of any developer (age, social status, education degree, sex, etc.) affect projects diversification level in general. Moreover, such properties have an influence on software code they create [2]. Consequently, we may identify specific developers programming styles by analyzing certain attributes of their code.

3 First Implementation Issues for the Proposed Approach

Personal properties set of each software developer can be evaluated using the MMPI method, which has 8 scales and assumes 16 different types of personality[1]. The goal is to create software development teams with diversification levels as low as possible inside the team, and as high as possible between the teams in order to maximize diversity level between NVP's versions and minimize it inside teams. The diversification level of source code may be assessed using a set of metrics for developers programming style. There are 50 such metrics presented in [2] which can be used for software authorship identification. Some of them are given in table 1.

Table 1. Some programming style code metrics

Metrics name	Description	Possible values range
STY1c	Number of opening brackets, which are last symbols in the row	$[0; M]$
STY2a	Ratio of rows with comments only to total number of rows	$[0; 1]$
STY5	Average number of spaces to the right of operators	$[0; M]$
PRO2b	Average length of method names	$[1; M]$

In table 1 the value M varies depending on the specific programming languages which are used. An effectiveness evaluation of DDT formation process may be done with the help of reliability value at the current NVS development iteration. The following formula may be used to calculate the DFD value [6]:

$$DFD = \sum_{i=1}^N \frac{N-i}{N-1} k_i \quad (1)$$

where N stands for the number of NVS versions; K means a number of unique failures of NVS runs; K_i represents a number of identical failures among

i versions, which are supposed to be identified on the current NVP projects iteration; therefore $k_i = K_i/K$. Such way we get a diversity level indicator as a normalized value between 0 and 1, namely: $DFD \in [0; 1]$.

4 Conclusions and Future Work

In this paper we have presented the human-centered approach to the assessment of team diversity in N-version programming based on the hypothesis of existence a correlation between certain number of developers personal properties and their specific programming styles. Therefore, the possibility of creation development teams with target diversification level can be provided. It affects positively a software product diversification level in general, and consequently improves its reliability. Next steps in this research will be dealing with the following theoretical and practical tasks:

1. perform a more sophisticated investigation of personal diversity assessment methods and different metrics of developers programming styles;
2. provide a motivated choice of measurement method which should be used to determine the correlation between parameters mentioned above;
3. propose a formalized DDT forming procedure with a pre-defined diversity level, e.g. using some linear programming methods;
4. design and implement the corresponding CASE-tool for automated execution of all steps in the elaborated DDTs forming procedure;
5. develop the methodology, and to perform a number of simulation experiments to check the workability and effectiveness of the proposed approach in real-life software development.

The resolution of these issues is foreseen in the framework of the PhD-research, which is planned for the period of years 2017-2020.

References

1. Capretz, L.F., Ahmed, F.: Why do we need personality diversity in software engineering? *ACM SIGSOFT Software Engineering Notes* 35(2), 1–11 (2010)
2. Ding, H., Samadzadeh, M.H.: Extraction of Java program fingerprints for software authorship identification. *Journal of Systems and Software* 72(1), 49–57 (2004)
3. Liang, T.P., Wu, J.C.H., Jiang, J.J., Klein, G.: The impact of value diversity on information system development projects. *International Journal of Project Management* 30(6), 731–739 (2012)
4. Lyu, M.R., He, Y.T.: Improving the N-version programming process through the evolution of a design paradigm. *IEEE Transactions on Reliability* 42(2), 179–189 (1993)
5. Official Website of OMG consortium. <http://www.omg.org/spec/UML/2.0/>
6. Partridge, D., Krzanowski, W.: Distinct failure diversity in multiversion software. *Res. Rep* 348, 24 (1997)
7. Schaefer, I., Rabiser, R., Clarke, D., Bettini, L., Benavides, D., Botterweck, G., Pathak, A., Trujillo, S., Vilella, K.: *Software diversity: state of the art and perspectives*. Springer (2012)