UDC 004.4-004.9

TECHNOLOGY OF EVALUATING THE PROFESSIONAL COMPETENCY OF PROGRAMMER TEAMS

Prihozhy A.A.
Belarusian National Technical University
Minsk, Belarus

Let $C = \{c_1,\ldots,c_m\}$ be a set of topics (Table) Joseph Sijin proposes in work [1] to estimate the competency of tentative participants of an IT project. Let $P = \{p_1,\ldots p_n\}$ be a set of programmers who have filled in a questionnaire and have indicated his proficiency level on each of the competency topic. Work [1] describes requirements to the programmer competency level on each of the topics. It introduces a metric of four predefined values $L0$, $L1$, $L2$ and $L3$, which we replace with the numerical values 0, 1/3, 2/3 and 1. As a result, a variable $PgrmLevel(p, c)$ describes the proficiency level of programmer p on competency c. Additionally, we introduce a $weight(c)$ of each competency topic $c \in C$ and estimate the weighted competency level of programmer p as:

$$PgrmWLevel(p,c) = weight(c) \times PgrmLevel(p,c) \tag{1}$$

Note that such a proficiency estimation technology extends the model proposed in [2]. We consider a subset $t = \{p_1,\ldots p_k\}$, $t \subseteq P$ of programmers as a team. The number $|t|$ of programmers in team t is the team size. To recognize workable and unworkable teams, we evaluate with (2) the team t average competency $AvrTeamComp(t, c)$ regarding topic c.

Table. Topics of the programmer competency matrix

Computer Science	**Software Engineering**
data structures	source code version control
algorithms	build automation
systems programming	automated testing
Programming	
problem decomposition	
systems decomposition	**Experience**
communication	languages with professional experience
code organization within a file	platforms with professional experience
code organization across files	years of professional experience
source tree organization	domain knowledge
code readability	
defensive coding	**Knowledge**
error handling	tool knowledge
IDE	languages exposed to
API	codebase knowledge
frameworks	knowledge of upcoming technologies
requirements	platform internals
scripting	books
database	blogs

$$AvrTeamComp(t,c) = \sum_{p \in t} PgrmLevel\ (p,c) \Big/ |t| \tag{2}$$

We also evaluate with (3) the best-representative team competency *BestTeamComp*(*t*, *c*).

$$BestTeamComp(t,c) = \max_{p \in t} PgrmLevel(p,c) \tag{3}$$

Every IT project formulates requirements to the competency level of a programmer and of a team with respect to each topic of the competency table. We model the requirements with three constraints:

- *TACConstr*(*c*) is a threshold value of the average competency level of a team programmer in topic *c*
- *TBCConstr*(*c*) is a threshold value of the team best-representative competency in topic *c*
- *TIConstr* is a threshold value of the integrated competency of a team.

We associate these three constraints with three team competency weighted parameters, which take value in interval [0, 1]:

1) weighted average competency over all team members and topics

$$TeamWAvrComp(t) = \sum_{c \in C} weight(c) \times AvrTeamComp(t,c) \Big/ MaxAllWComp \tag{4}$$

where *MaxAllWComp* is the sum of weights over all competency topics.

2) weighted best-representative competency over all topics

$$TeamWBestComp(t) = \sum_{c \in C} weight(c) \times BestTeamComp(t,c) \Big/ MaxAllWComp \tag{5}$$

3) integrated competency of a team

$$TeamIntCompet(t) = \lambda \times TeamWAvrComp(t) + (1-\lambda) \times TeamWBestComp(t) \tag{6}$$

where $0 \le \lambda \le 1$ describes the importance of average and best-representative team competency.

Each of the three parameters takes value 0, if inequalities as follows hold: $\exists c\ \big(AvrTeamComp(t,c) < TACConstr(c)\big)$, $\exists c\ \big(BestTeamComp(t,c) < TBCConstr(c)\big)$ and $TeamIntComp(t) < TIConstr$. Zero value means that team *t* is unworkable in the project; nonzero value means that the team is workable.

Let us assume that we have a partition of the set *P* of programmers into a set $T = \{t_1,\ldots,t_s\}$ of teams, and the team set cardinality is $|T|$. For each team $t \in T$ we have evaluated the competency *TeamIntComp*(*t*). We consider three ways to evaluate how perfect is the partitioning *T*, i.e. on the number of workable teams, all teams' competency, and average competency of a workable team in the partitioning. We maximize the value of three functions: the all teams competency

$$OverallComp(T) = \sum_{t \in T} TeamComp\ (t) \tag{7}$$

and the average competency of a team

$$AverageComp(T) = {OverallComp(T)} \big/ {WorkableTN(T)} \quad (8)$$

and the number $|T|$ of workable teams.

The maximization of each of the functions is a hard combinatorial problem, for which no algorithm of polynomial computational complexity known. That is why we have developed a genetic algorithm, which is a good heuristic for finding an acceptable suboptimal solution [3]. In this paper, we report results obtained for a set P of 33 programmers, for all of 32 competency topics, and for given constraints on an IT project. The genetic algorithm has generated various partitioning T of set P for various value of the constraints. Figure shows the dependence of $|T|$ and *OverallComp*(T) on *TIConstr*. *OverallComp*(T) decreases from 6.06 to 0.83, and $|T|$ decreases from 9 to 1 with increasing the value of *TIConstr* from 0.3 to 0.82. The value of $|T|$ is larger than *OverallComp*(T) in all points, as the team competency is less than 1 for each team.

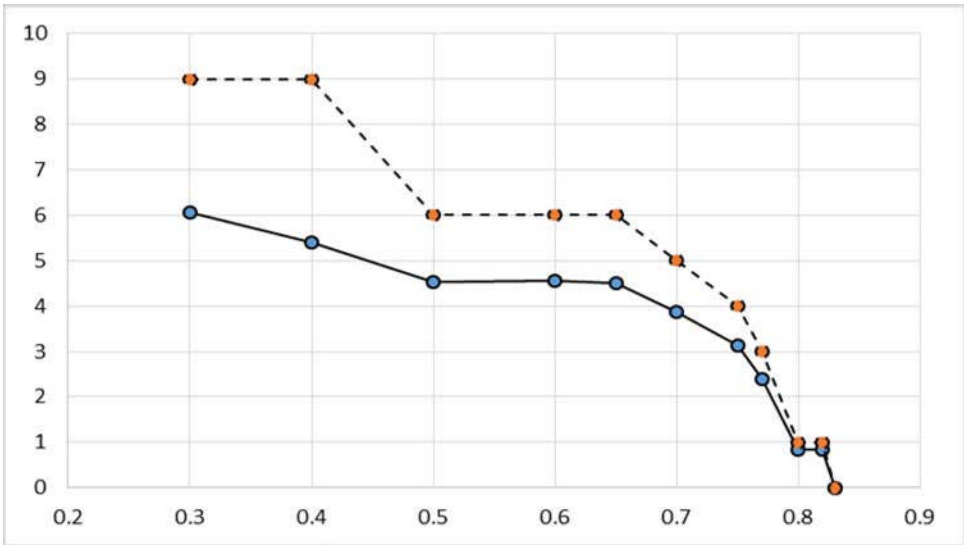

Figure – All teams competency (solid) and teams count (dash) vs. team competency constraint

References

1. Sijin, J. Perspectives on Software, Technology and Business: Programmer Competency Matrix / J. Sijin // [Electronic resource]. –Mode of access: https://sijinjoseph.com/programmer-competency-matrix/. – Date of access: 10.01.2020.
2. Прихожий, А.А. Метод оценки квалификации и оптимизация состава профессиональных групп программистов / А.А. Прихожий, А.М. Ждановский // Системный анализ и прикладная информатика. – 2018, No. 2, с. 4-12.
3. Prihozhy, A. Genetic algorithm of optimizing the size, staff and number of professional teams of programmers / A. Prihozhy, A. Zhdanouski // Open Semantic Technologies for Intelligent Systems 2019: Proceedings of International Conference. – Minsk, 2019, pp. 305–310.