ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И АВТОМАТИЗАЦИЯ

UDC 004.4-004.9

EVALUATING 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 *L*0, *L*1, *L*2 and *L*3, 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:

$$PgrmWLevel(p,c) = weight(c) \times PgrmLevel(p,c) \qquad (1)$$

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*.

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

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

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

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.

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	platforms with professional

a file	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

- *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 a value in interval [0, 1]:

1) weighted average competency over all team members and topics

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

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

2) weighted best-representative competency over all topics

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

3) integrated competency of a team

$$TeamIntCompet(t) = \lambda \times TeamWAvrComp(t) + (1-\lambda) \times TeamWBestComp(t)$$

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:

1) $\exists c \left(AvrTeamComp(t,c) < TACConstr(c) \right)$
2) $\exists c \left(BestTeamComp(t,c) < TBCConstr(c) \right)$
3) $TeamIntComp(t) < TIConstr$.

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:

1) the all teams competency

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

2) the average competency of a team

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

3) 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 [3-5]. 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. The value of |*T*| and *OverallComp*(*T*) depend on *TIConstr*. The value of *OverallComp*(*T*) decreases from 6.06 to 0.83, and the value of |*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 cases because the team competency is less than 1 for each team.

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: 23.02.2021.
2. Prihozhy A.A., Zhdanouski A.M. Method of qualification estimation and optimization of professional teams of programmers. *«System analysis and applied information science»*. 2018;(2):4-11. (In Russ.) https://doi.org/10.21122/2309-4923-2018-2-4-11
3. Prihozhy, A. Genetic algorithm of optimizing the size, staff and number of professional teams of programmers / A. Prihozhy, A. Zhdanouski // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2019) : материалы международной научно-технической конференции, Минск, 21 - 23 февраля 2019 г. / Белорусский государственный университет

информатики и радиоэлектроники; редкол.: В. В. Голенков (гл. ред.) [и др.]. - Минск, 2019. - С. 305 - 310.

4. Prihozhy A.A., Zhdanouski A.M. Genetic algorithm of optimizing the qualification of programmer teams. *«System analysis and applied information science»*. 2020;(4):31-38. https://doi.org/10.21122/2309-4923-2020-4-31-38

5. Прихожий, А. А. Оптимизация состава профессиональных групп программистов для работы над большими проектами / А. А. Прихожий, А. М. Ждановский // BIG DATA Advanced Analytics: collection of materials of the fourth international scientific and practical conference, Minsk, Belarus, May 3 – 4, 2018 / editorial board: M. Batura [etc.]. – Minsk, BSUIR, 2018. – P. 405 – 408.

УДК 004.415:378

ОПРЕДЕЛЕНИЕ ОБЛАСТЕЙ РАЗМЕЩЕНИЯ ЯЧЕИСТЫХ СТРУКТУР ПРИ ПРОЕКТИРОВАНИИ ЛЕГКОВЕСНЫХ ДЕТАЛЕЙ

Полозков Ю.В., Ярмошук Ю.М., Кункевич Д.П.,
Напрасников В.В.
Белорусский национальный технический университет
Минск, Беларусь

Рассматривается проблема определения наименее нагруженных и деформирующихся областей для размещения ячеистых структур при проектировании легковесных деталей. Описываются особенности реализованного алгоритма поиска таких областей на основе триангуляции Делоне. Представлены результаты работы алгоритма.

При создании легковесных деталей ключевым условием возможности размещения ячеистых структур, является то, что показатели напряженно-деформированного состояния проектируемой детали должны оставаться в пределах, обеспечивающих ее работоспособность. Поэтому эти показатели,