We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

3

A Method for Project Member Role Assignment in Open Source Software Development using Self-Organizing Maps

Shingo Kawamura, Minoru Uehara, and Hideki Mori
Department of Open Information Systems,
Graduate School of Engineering,
Toyo University,
Japan

1. Introduction

We propose the development of open source software (OSS) by a web community. At present, SourceForge.net (http://sourceforge.net/) is a typical community for OSS development, consisting of CVS/SVN repositories, mailing lists, bug tracking systems, task management systems, BBS, open discussion forums, and so on. Although many people are involved in the open discussion forums of SourceForge.net, a project leader is required to expedite and manage the process, which is a great responsibility. Membership is also restricted to expert software developers. In the method proposed in this paper, we envisage not only expert users being able to participate, but also those that use the software but do not have programming skills. Moreover, when the development groups are organized, a supervisor is no longer necessary. The community is managed automatically through the aggregation and distribution of the participating members. The basic concept is exactly the model used by existing SNSs and Wikis, and the development process is based on the spiral model. This paper aims to construct a Web2.0 environment that supports such development and enables the proposal of topics by users. Our method allows the selection of suitable project members from a human resource database (DB) using a self-organizing map (SOM), that is, reinforcement learning. In other words, we propose a web community based on the wisdom of crowds, which is distributed and aggregated. To achieve this, we propose the use of a SOM. Section 2 explains the SOM and the concept of the wisdom of the crowd. The proposed system is described in Section 3. Experiments on feature analysis of the members using a SOM are presented in Section 4, with the results and a discussion thereof given in Section 5 and our conclusions in Section 6.

2. Related work

2.1 SOM

The SOM was designed by Kohonen (1995) at Helsinki University. The neural network is modeled by the visual area in the human brain, and consists of two layers, an input layer and an output (map) layer.

$$d_1 = \sqrt{\sum_{j=1}^{n} \left(x_j - \omega_{ij} \right)^2} \tag{1}$$

x: input vector, ω: reference vector

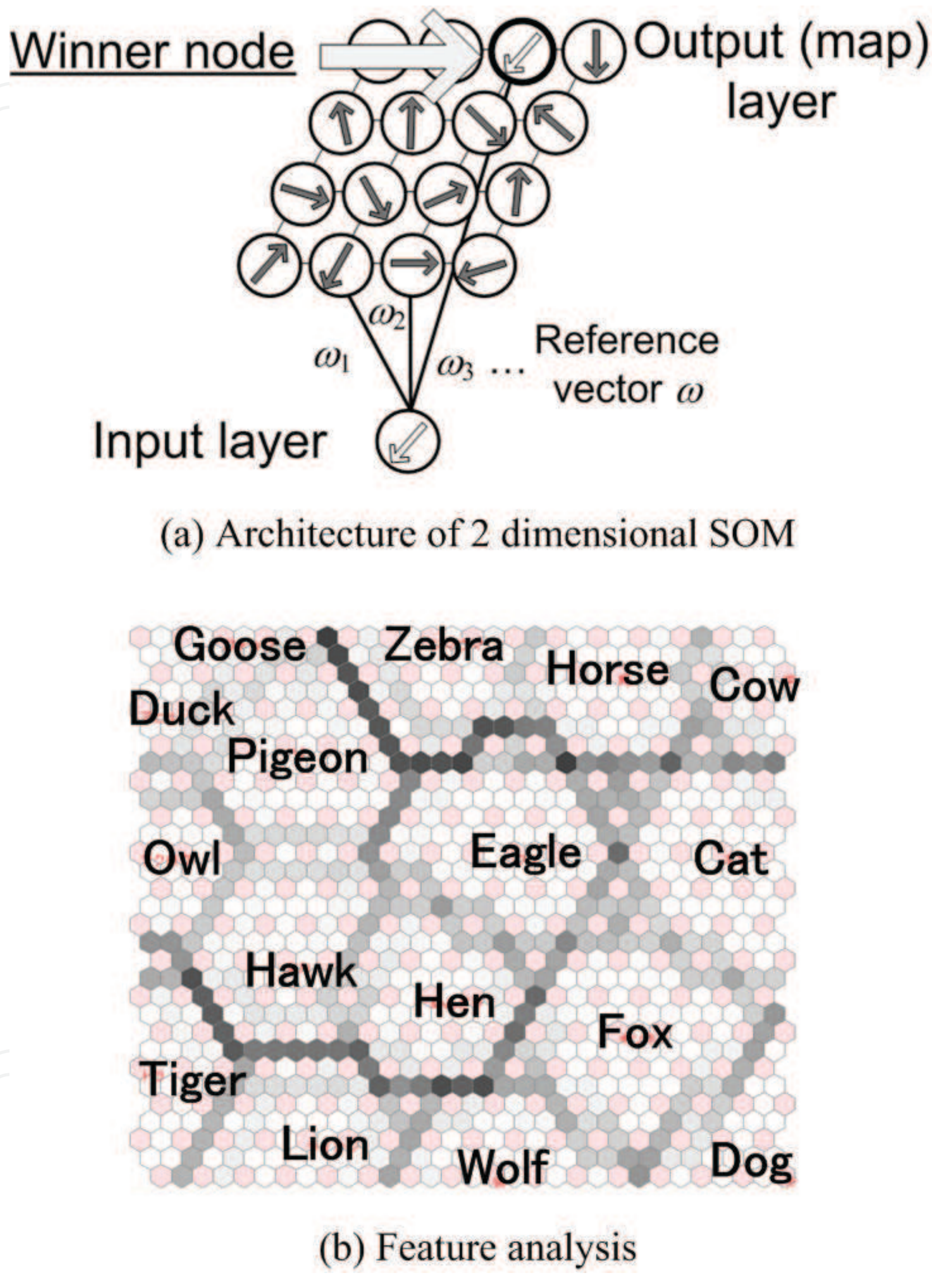


Fig. 1. Example of a SOM

In a SOM, the most common similarity measure is the Euclidean distance, which is defined as the distance of the input vector from the weight vector of a neural node. The nodes are arranged in a two dimensional layout as shown in Figure 1(a), where x is the input vector and ω the reference vector.

The weights of each node are initialized randomly in the initial state of learning. Then, the node, whose weight vector is the closest to the input vector, is selected as the winner node and the weights of the winner node and its neighbors are updated by the following neighborhood function.

$$\text{if } i \notin N_c \text{, then } h_{ci} = \alpha_0(1 - t / T) \quad (2)$$

$$\text{else } h_{ci}(t) = 0 \quad N_c(t) = N_c(0)(1 - t / T) \text{ ,} \quad (3)$$

where $\alpha(t)$ is the coefficient of learning, T is the learning time, and N_c is the neighborhood region.

Each weight of a node represents a feature of the object. As an example, animal feature data analyzed by a SOM are shown in Figure 1(b). SOMs are also using in data mining and cluster analysis. In this paper, we use a SOM to ensure diversity of users in a web community. We analyze the characteristics of the available human resources using a SOM. Since the required skill set differs at each stage in a project, we envisage using not only skill, but also personality and aptitude in our analysis, as there may not be any experts participating and aggregation is incorporated in the method.

2.2 Wisdom of crowds

It is said that the service known as the Web2.0 has the characteristics of the "wisdom of crowds", that is, diversity of opinion, independence, decentralization, and aggregation (Fig. 2). Recently, much attention has been focused on crowd sourcing, which is a technique for acquiring productive capacity at an appropriate level in subcontracting development to many unspecified members. Previously, development of OSS was done in a similar manner by many volunteers. In addition, not only OSS development, but also knowledge sharing has been performed spontaneously by networks of individuals with the necessary knowledge (wikis) and experiences (blogs) in recent years. These have the characteristics of the "wisdom of crowds". An advantage of this is that diverse opinions can be reflected, although, on the other hand noisy information tends to be exaggerated.

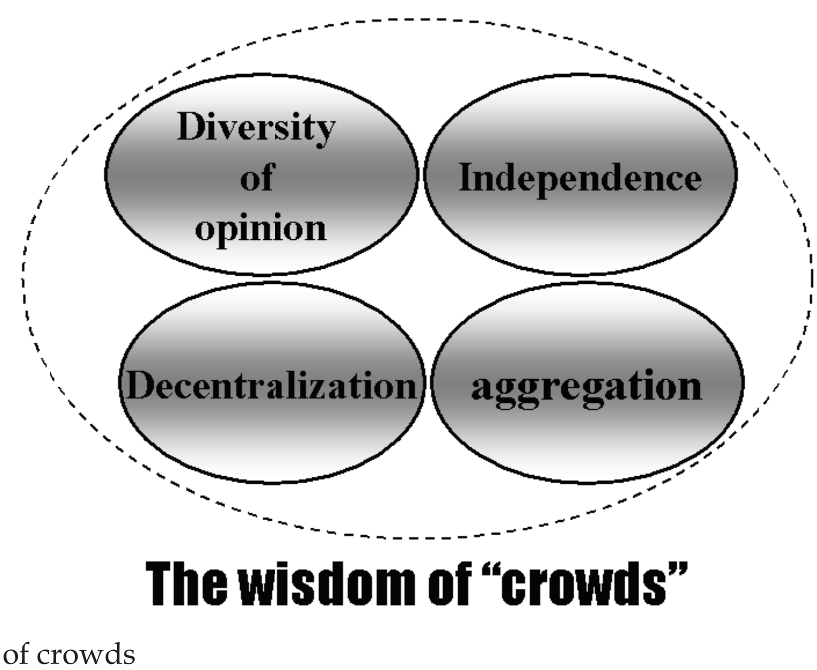


Fig. 2. Wisdom of crowds

3. OSS project

3.1 About C-DOS

OSS development often uses the agile development model that embodies short-term iteration. We propose a web community for OSS development using a SOM implementing the agile model for analyzing developers and their automatic assignment.

C-DOS (Community to Develop OSS using the Spiral model) is the name of the web community dedicated to the development of OSS. C-DOS actually consists of two communities; one to propose OSS projects and the other to promote these projects. Both communities are implemented using OpenPNE (http://docs.openpne.jp), which is an open source SNS environment.

3.2 User participation

Anyone can register and participate in the C-DOS community. When users register, they themselves must evaluate their IT development skills using SWEBOK (SoftWare Engineering Body Of Knowledge) (http://www.swebok.org/) and provide personal preferences to the community.

3.3 Proposal of a new OSS project

Once user registration has been completed, the user has the right to join the community, and is registered in the human resources DB. If users have suggestions for new software development or improvements to existing software, they can provide the requirements of the project. Alternatively, they may put in a request to be included in the development team. All requests are sent to the human resources DB. When the number of applicants exceeds the pre-defined minimum, the suggestion is approved by the community and the project starts. It should be noted that a "group leader" has not traditionally been an integral part of this type of network of members ("suggests" included). Figures 3 and 4 illustrate the process flows from registering a new user to starting a new project.

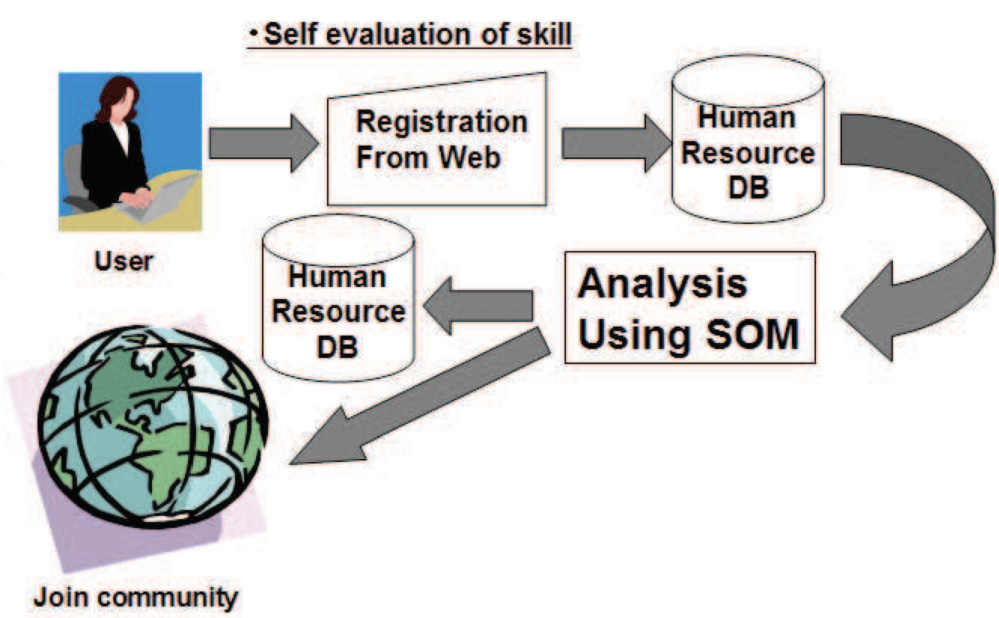


Fig. 3. Process flow for user registration

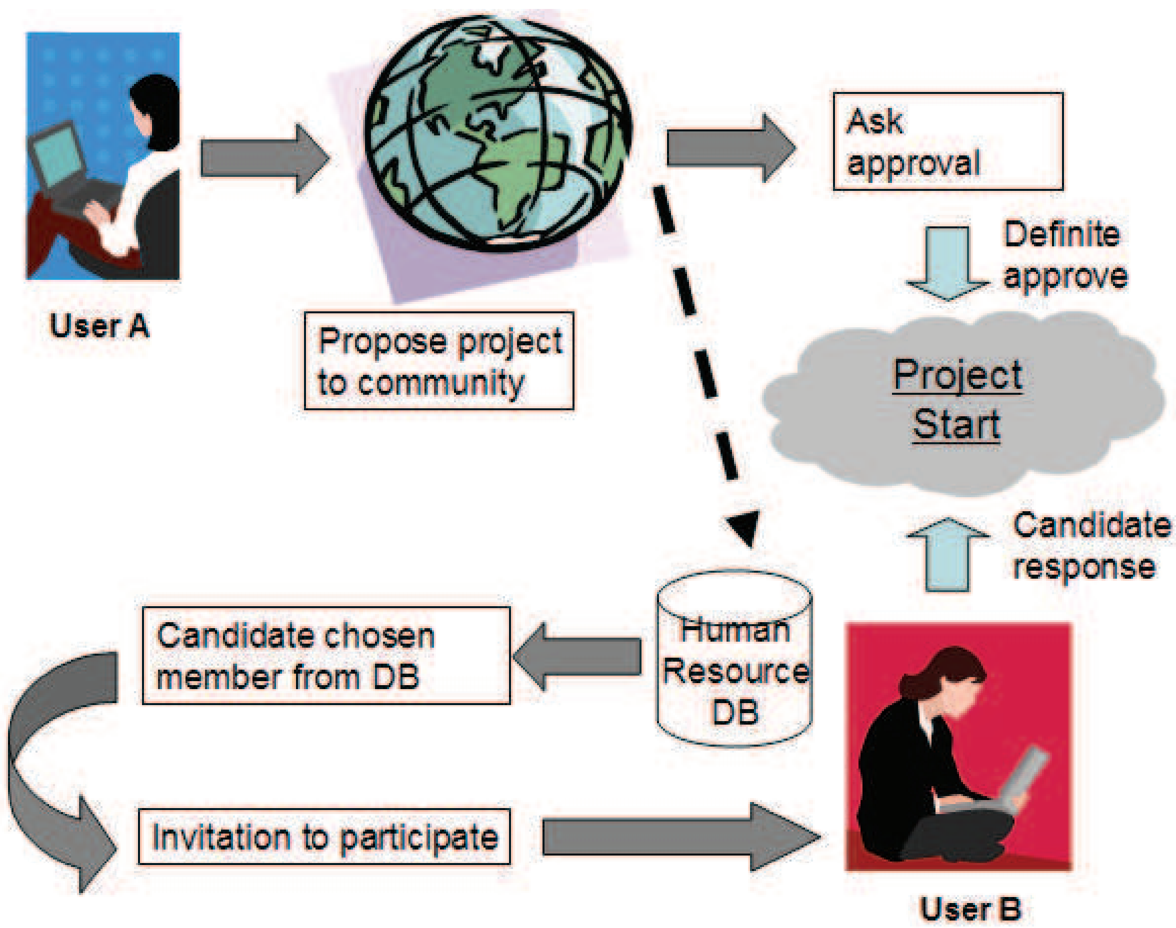


Fig. 4. Process flow for project proposal

3.4 Model for development process

We propose a new development model based on the spiral model (Boehm, 1988). OSS often uses the agile development model, which is flexible enough to be able to change OSS definitions and requirements. But, it is limited to use by expert programmers only. Thus, we have designed a development model for C-DOS that corresponds to the development process. An outline of this model is shown in Figure 5.

Each project is broken down into 4 stages or layers covering the entire process, that is, requirements, designing, building, and testing. After a project has been approved, the project members are assigned to a particular layer by the SOM. Members assigned to each layer can read the working data of the other layers, but cannot alter this data.

When members finish a series of tasks in the current layer, the next group of members begin working on the corresponding next layer.

3.5 Development of an OSS project

Included in the community developing an OSS project are a Wiki and BBS. Members engage in online discussions using the BBS and the results of each layer are written to the Wiki, which can be updated by any of the project members. Updates are finalized when the members in the layer approve the content. Moreover, updated results in the content of the

Wiki are evaluated by all the members of other layers. If agreement is reached through decisions of the majority of members, the task in the layer is considered completed. Importantly, we have designed "bridge layers" between the main layers, where members in a bridge layer have the right to participate in neighboring layers. The bridge layer has the effect of design consistency. Members in the bridge layer are also selected by the SOM, with the number of members being flexible.

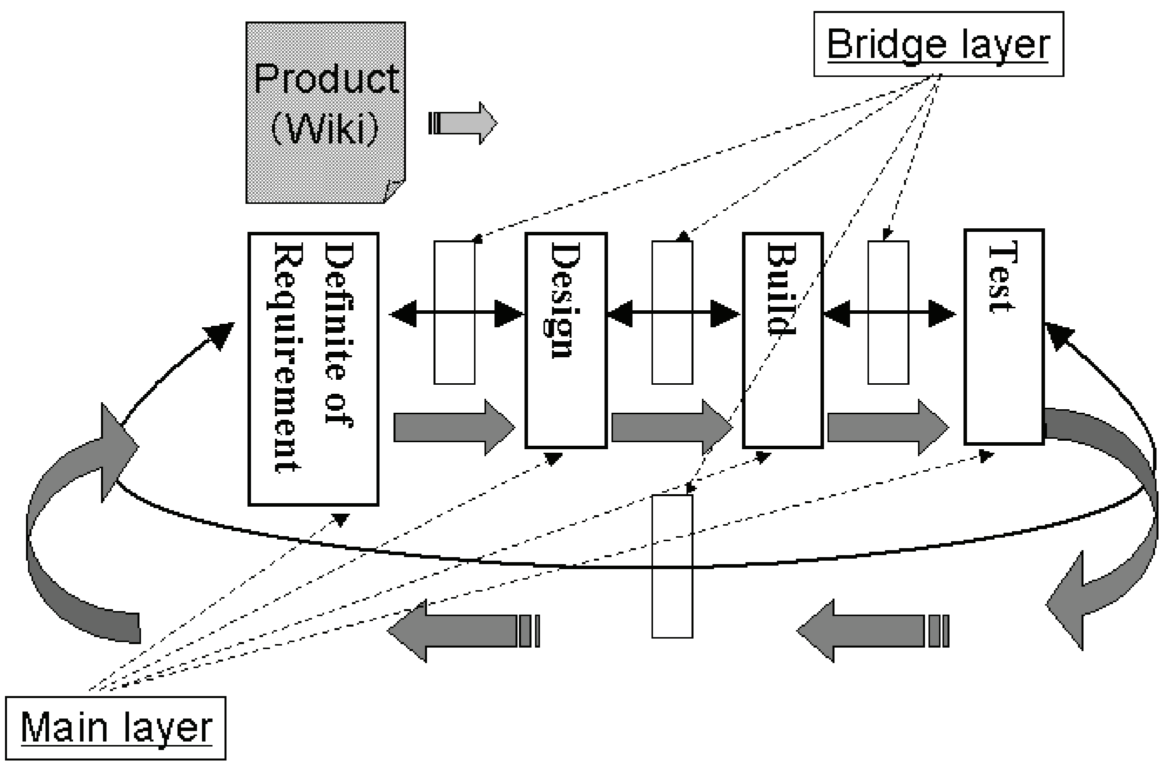


Fig. 5. Model for development of C-DOS

4. Experiment

Processing to assign members to each layer is necessary in C-DOS. This can be achieved by applying the cluster analysis described in this section, in which we conduct a SOM analysis on the information provided on the registration forms and observe the results.

4.1 Skills evaluation

Member self evaluate their knowledge and skills for software development. The resulting input data is based on a five-level rating scale using SWEBOK.

4.2 Analysis of candidate developers using SOM

In this section, we perform a SOM analysis, integrating our knowledge of candidates' interests, hobbies, and their software development skills in relation to the community. Using SWEBOK, it is possible to evaluate a member's knowledge about software development.

SWEBOK, established by a group led by the IEEE in 2001, is a system containing the minimum knowledge that a professional should have acquired after working for four years after graduation from university, The SOM analysis uses SOM_PAK, which is a tool provided by Prof. Kohonen's lab (Kate, 2002).
We show the detailed experimental conditions in Tables 1 and 2. In both evaluations, we follow the taxonomy of Bloom (1956), and users self evaluate their skills in ten phases.

Input Data	21 dimension
Map phase	12 x 8 hexagonal
Learning times	1000

Table 1. Outline of the analysis using a SOM

I.	**Definition of requirement**
i)	Basis of software requirement
ii)	Process of requirement
iii)	Extract of require
iv)	Analyses of requirement
v)	Require to specification
vi)	Confirm appropriateness of requirement
vii)	Consideration matters in practice
II.	**Designing skill**
i)	Basis of software design
ii)	Main problems of software design
iii)	Structure and architecture
iv)	Analyses and evaluation of software designing quality
v)	A notational system of software design
vi)	Tactics and method
III.	**Programming skill**
i)	Basis of building software
ii)	Management of building software
iii)	Consideration matters in practice
IV.	**Testing skill**
i)	Basis of software testing
ii)	Test level
iii)	Techniques of test
iv)	Measure of test
v)	Test process

Table 2. Input data format (knowledge and skill for software development)

Name	I							II						III			IV				
	i	ii	iii	iv	v	vi	vii	i	ii	iii	iv	v	vi	i	ii	iii	i	ii	iii	iv	v
A	10.0	9.0	8.0	8.0	8.0	8.0	8.0	9.0	8.0	9.0	7.0	8.0	8.0	10.0	8.0	8.0	9.0	8.0	8.0	7.0	8.0
B	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	9.0	9.0	9.0	9.0	9.0
C	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	9.0	9.0	9.0	8.0	8.0	8.0	8.0	8.0
D	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0
E	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0
F	6.0	6.0	6.0	6.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
G	6.0	6.0	6.0	6.0	6.0	6.0	6.0	7.0	7.0	7.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
H	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
I	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	5.0	5.0	5.0	6.0	6.0	6.0	6.0	6.0
J	7.0	7.0	7.0	7.0	6.0	7.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
K	7.0	6.0	6.0	6.0	6.0	7.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
L	9.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
M	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	9.0	9.0	9.0	7.0	7.0	7.0	7.0	7.0
N	8.3	6.2	6.1	8.9	6.0	6.7	6.8	2.4	3.1	3.7	3.0	2.5	3.1	2.3	3.0	2.9	2.6	2.4	2.4	3.1	3.4
O	7.4	8.8	6.9	8.1	6.7	9.2	7.8	3.6	2.7	3.0	2.5	3.5	2.3	4.0	3.8	3.0	2.0	3.8	3.4	3.4	3.6
P	3.8	3.1	3.1	2.7	3.3	3.1	2.1	7.0	7.8	9.4	8.3	9.9	6.7	3.7	3.0	2.5	4.0	3.5	3.8	3.1	2.7
Q	2.9	3.4	3.7	3.5	3.7	3.7	2.4	8.9	8.6	8.6	8.5	9.7	8.4	2.7	3.1	3.4	2.6	3.9	2.7	3.4	3.8
R	3.3	4.0	2.7	3.8	3.9	2.1	3.5	3.2	3.2	2.8	2.4	3.9	3.4	7.5	8.9	9.6	2.0	3.8	3.5	2.5	2.1
S	3.8	3.8	2.7	3.1	2.6	2.4	2.7	3.1	2.2	3.6	2.1	3.4	3.6	7.6	6.2	7.5	3.5	3.8	2.3	3.8	2.1
T	2.3	3.3	3.6	3.2	2.9	2.2	3.8	2.6	3.4	3.1	3.6	3.7	3.3	2.3	3.6	2.1	7.4	9.9	7.7	9.0	6.4
U	2.9	2.8	3.8	3.5	2.3	3.3	3.9	3.6	3.4	2.2	3.1	3.6	3.9	3.6	3.5	2.8	7.6	8.4	7.4	8.3	9.6
V	2.9	3.8	2.9	3.8	2.2	3.6	2.1	3.2	2.8	3.1	2.1	2.1	3.2	2.9	2.6	3.7	2.7	3.2	2.2	2.5	2.3
W	9.1	9.8	7.1	7.5	7.3	8.1	8.4	6.5	9.9	8.9	7.2	6.6	9.5	6.1	9.9	6.5	6.4	7.1	9.8	9.9	6.9

Table 3. Experimental input data

5. Results

We defined three reference models, namely, a specialist model, in which project members have excellent scores for a specific skill phase, a generalist model, in which members have good scores, above average in all phases, and an ordinary model, in which members generally have immature skill scores. Scores in the 3 reference models are shown as (N-U), W, and V, respectively.

SOM mapping is performed using the sampled score questionnaire from our OSS project members including the reference models' scores. Then, the distance is determined between the sampled location and the reference model locations on the SOM map.

The data used for mapping is shown in Table 3. For instance, N and O specialize in a phase I skill. A two-dimensional SOM is applied to 12x8 arrays. The SOM results are illustrated in Figure 6, while the reference vector after the experiment is given in Table 4. Mappings of the specialist reference model are located on the right of the map and form their own clusters. Member G, because the skill value in phase II is the highest, is mapped close to cluster (P, Q), which belongs to phase II of the specialist reference model. Member H, on the other hand, despite the phase II skill value being as high as that of G, is mapped far from P and Q, because, contrary to the previous case, the other skills are also high. In the generalist case with a high skill value, the mapping is located on the upper left of the map. Ordinary cases with no particular features are scattered around the map and do not belong to any cluster.

6. Conclusion

We proposed a method using a SOM for the assignment of project members in the development of OSS. In the method, member skills are expressed numerically for four software development phases. Then a SOM analysis is carried out on the data, resulting in developers being mapped and located on the map according to their skills. In the experiment, we introduced reference models featuring varying skill levels for particular phases of the development. Using a sample of selected members, we conducted a SOM analysis on the data from the developer questionnaires, and then performed a comparison of the distances between the reference models and the sample. For samples with a high speciality in a particular skill, the mapping shows their location on the map close to the corresponding reference models. According to the results, generalist members with overall good skills in many development phases, are concentrated on the upper left of the map. Members with average skills for each phase, on the other hand, are located randomly on the map. In future research, using the data from previous successful OSS projects, we intend to analyze the relationship between the proposed reference models and actual skills data in order to refine the reference models.

7. Acknowledgment

We would like to thank Mr. N. Oguchi, S. Touji, and K. Matsumoto for their helpful advice and encouragement during our research.

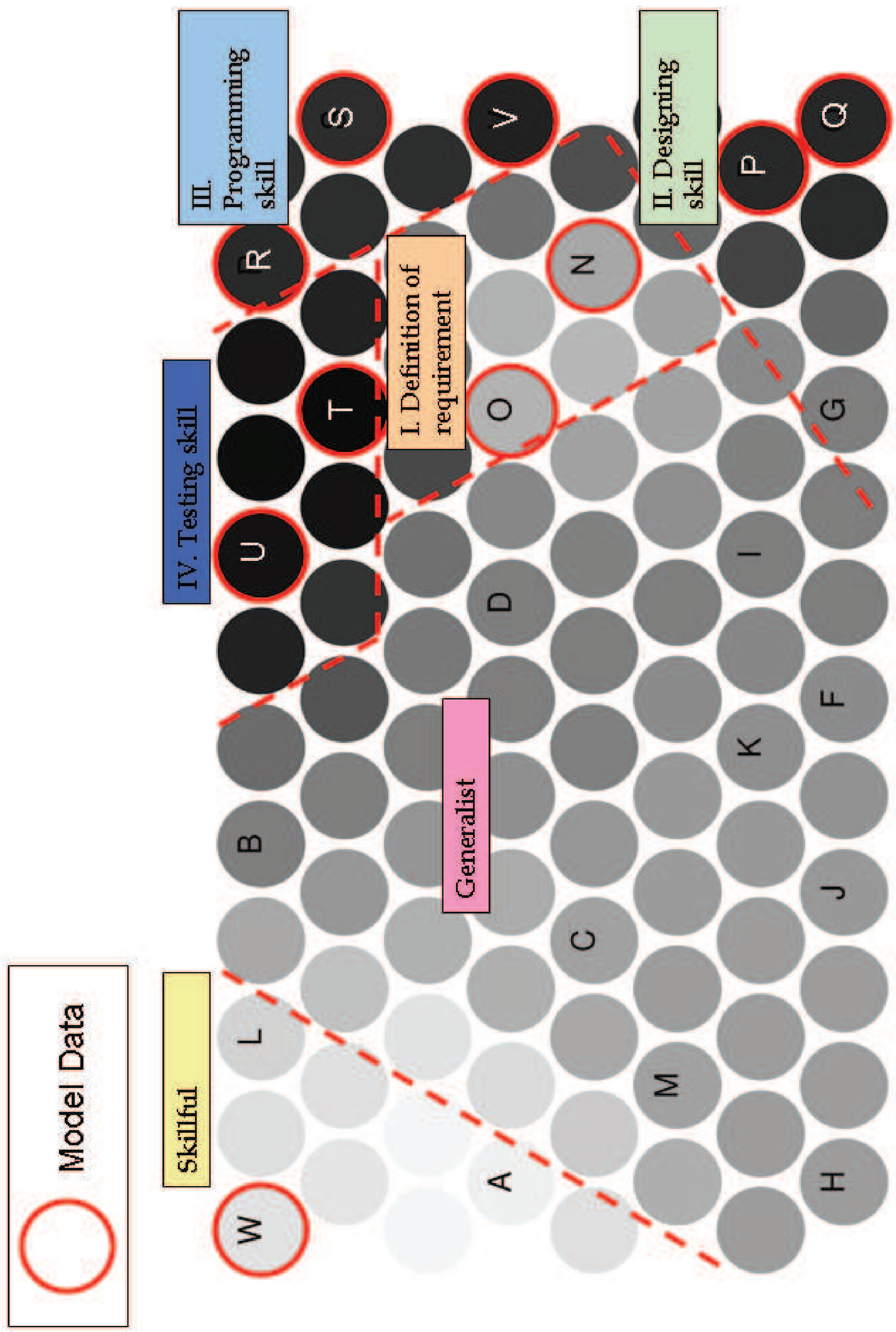


Fig. 6. Experimental results

x_{dim}	y_{dim}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	Name
1	1	9.1	9.7	7.2	7.5	7.3	8.1	8.4	6.6	9.8	8.9	7.2	6.7	9.4	6.3	9.8	6.6	6.6	7.2	9.7	9.8	7.0	W
2	1	9.1	9.0	7.5	7.7	7.6	8.1	8.2	7.2	9.0	8.5	7.6	7.2	8.8	7.9	9.9	8.1	8.1	8.4	9.9	9.9	8.3	
3	1	8.7	8.0	7.7	7.7	7.7	7.8	7.8	7.6	8.0	7.9	7.7	7.6	7.9	9.3	9.8	9.4	9.5	9.6	9.9	9.9	9.5	L
4	1	7.4	6.9	6.9	6.9	6.9	6.9	6.9	6.9	6.9	6.9	6.9	6.9	6.9	8.9	8.9	8.9	9.5	9.5	9.5	9.5	9.5	
5	1	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	9.0	9.0	9.0	9.0	9.0	B
6	1	5.5	5.5	5.6	5.6	5.4	5.6	5.7	5.6	5.6	5.4	5.5	5.6	5.7	7.3	7.3	7.1	8.8	8.9	8.7	8.9	9.1	
7	1	3.2	3.2	4.0	3.8	2.8	3.5	4.1	3.8	3.7	2.7	3.5	3.9	4.1	4.0	4.0	3.3	7.7	8.6	7.6	8.5	9.2	
8	1	2.7	2.9	3.7	3.4	2.5	3.0	3.9	3.3	3.4	2.5	3.2	3.6	3.7	3.2	3.5	2.6	7.5	8.8	7.5	8.5	8.7	U
9	1	2.6	3.0	3.7	3.4	2.6	2.8	3.8	3.1	3.4	2.6	3.3	3.6	3.6	3.0	3.6	2.5	7.5	9.1	7.5	8.6	8.0	
10	1	2.8	3.5	3.3	3.3	3.0	2.3	3.5	2.8	3.1	3.1	3.1	3.7	3.4	4.2	4.9	4.4	5.8	7.7	6.0	6.9	5.0	
11	1	3.4	3.9	2.8	3.6	3.4	2.2	3.3	3.1	2.9	3.1	2.4	3.7	3.5	7.2	7.8	8.5	2.8	4.1	3.4	3.3	2.3	R
12	1	3.5	3.9	2.7	3.5	3.2	2.3	3.1	3.2	2.7	3.2	2.2	3.6	3.5	7.5	7.5	8.5	2.8	3.8	2.9	3.1	2.1	
1	2	9.2	9.3	7.4	7.7	7.6	8.1	8.3	7.2	9.2	8.8	7.3	7.1	8.9	7.6	9.6	7.4	7.6	7.8	9.5	9.4	7.7	
2	2	9.1	8.5	7.8	7.9	7.8	8.0	8.1	7.7	8.5	8.3	7.7	7.7	8.4	9.0	9.8	9.0	9.1	9.2	9.8	9.8	9.1	
3	2	8.2	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4	7.4	9.4	9.4	9.4	9.7	9.7	9.7	9.7	9.7	
4	2	6.8	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	8.6	8.6	8.6	9.2	9.2	9.2	9.2	9.2	
5	2	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.1	8.1	8.1	8.1	8.8	8.8	8.8	8.8	8.8	
6	2	4.7	4.7	5.1	5.0	4.5	4.9	5.1	5.0	4.9	4.5	4.8	5.0	5.1	6.2	6.2	5.9	8.1	8.4	8.0	8.3	8.9	
7	2	3.7	3.8	4.4	4.2	3.4	3.9	4.5	4.1	4.1	3.4	4.0	4.3	4.4	4.7	4.8	4.2	7.4	8.2	7.3	8.0	8.4	
8	2	2.8	3.2	3.8	3.5	2.8	2.9	4.0	3.3	3.5	2.8	3.5	3.8	3.7	3.2	3.8	2.8	7.5	9.0	7.5	8.6	7.9	
9	2	2.5	3.2	3.7	3.3	2.7	2.5	3.8	2.9	3.4	2.9	3.5	3.7	3.5	2.7	3.6	2.3	7.4	9.5	7.6	8.8	7.3	T
10	2	3.2	3.7	3.4	3.7	3.3	2.5	3.7	2.8	3.1	3.2	3.0	3.6	3.4	4.6	5.3	5.0	5.1	6.8	5.3	6.1	4.4	
11	2	3.5	3.9	2.7	3.5	3.2	2.3	3.1	3.2	2.7	3.2	2.2	3.6	3.5	7.4	7.4	8.4	2.8	3.8	2.9	3.1	2.1	
12	2	3.5	3.9	2.7	3.4	3.0	2.4	2.9	3.1	2.6	3.2	2.2	3.4	3.5	7.0	6.8	7.8	2.8	3.7	2.7	3.2	2.1	S
1	3	9.7	9.1	7.8	7.9	7.8	8.0	8.1	8.3	8.5	8.9	7.1	7.7	8.4	9.0	8.6	7.8	8.4	7.9	8.6	7.9	7.9	
2	3	9.7	8.8	7.9	8.0	7.9	8.0	8.0	8.6	8.1	8.8	7.2	7.9	8.1	9.7	8.6	8.3	9.0	8.4	8.6	7.9	8.3	
3	3	9.2	8.3	7.9	7.9	7.9	7.9	7.9	8.3	7.9	8.3	7.4	7.9	7.9	9.9	9.0	9.0	9.3	8.9	8.9	8.4	8.9	
4	3	7.6	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	9.2	9.2	9.2	8.9	8.9	8.9	8.9	8.9	
5	3	6.8	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	8.6	8.6	8.6	9.0	9.0	9.0	9.0	9.0	

x_{dim}	y_{dim}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	Name
6	3	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.2	7.2	7.2	7.2	7.2	
7	3	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	7.8	7.8	7.8	7.0	7.1	7.0	7.0	7.1	
8	3	5.6	5.6	5.7	5.7	5.6	5.7	5.8	5.7	5.7	5.6	5.7	5.7	5.8	7.5	7.5	7.4	7.1	7.2	7.1	7.2	7.2	
9	3	4.4	4.9	4.8	5.0	4.4	4.4	5.1	3.6	4.0	3.9	4.0	4.2	3.9	4.0	4.7	3.8	6.2	7.8	6.5	7.3	6.0	
10	3	5.5	5.5	5.1	6.2	4.7	5.2	5.6	2.8	3.1	3.3	3.0	3.2	3.0	3.1	3.6	3.0	4.3	5.6	4.6	5.4	4.5	
11	3	5.8	5.1	4.5	6.1	4.3	4.9	4.7	2.8	2.8	3.5	2.5	2.8	3.2	4.1	4.1	4.6	2.8	3.0	2.4	3.2	2.8	
12	3	3.6	3.9	2.9	3.7	2.7	3.1	2.7	3.1	2.6	3.3	2.2	2.8	3.4	5.3	4.7	5.8	3.0	3.5	2.4	3.1	2.2	
1	4	9.6	8.6	7.9	7.9	7.9	7.9	7.9	8.7	8.0	8.7	7.3	8.0	8.0	9.9	8.5	8.5	9.1	8.4	8.4	7.6	8.4	A
2	4	9.0	8.3	7.7	7.7	7.7	7.7	7.7	8.4	7.8	8.4	7.3	7.8	7.8	9.7	8.5	8.5	8.6	8.0	8.0	7.5	8.0	
3	4	7.6	7.3	7.3	7.3	7.3	7.3	7.3	7.4	7.3	7.4	7.2	7.3	7.3	9.3	9.1	9.1	8.4	8.4	8.4	8.3	8.4	
4	4	7.5	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	9.2	9.2	9.2	8.5	8.5	8.5	8.5	8.5	
5	4	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	8.6	8.6	8.6	7.6	7.6	7.6	7.6	7.6	
6	4	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0	
7	4	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0	D
8	4	6.3	6.5	6.2	6.5	6.1	6.6	6.4	5.4	5.3	5.4	5.3	5.4	5.2	7.1	7.1	6.9	5.9	6.3	6.2	6.2	6.3	
9	4	7.7	7.6	6.5	8.3	6.4	8.0	7.3	3.3	3.1	3.5	2.9	3.3	2.9	3.6	3.7	3.3	2.6	3.4	3.2	3.5	3.7	O
10	4	7.7	7.4	6.4	8.3	6.2	7.8	7.1	3.0	2.9	3.3	2.7	3.0	2.7	3.1	3.4	3.0	2.3	3.1	2.9	3.2	3.5	
11	4	5.6	5.2	4.6	6.3	4.2	5.3	4.6	2.9	2.9	3.4	2.5	2.5	3.1	3.0	3.1	3.5	2.7	2.9	2.4	2.9	2.9	
12	4	3.1	3.9	3.0	4.0	2.4	3.7	2.3	3.2	2.8	3.1	2.1	2.1	3.2	2.9	2.7	3.7	2.7	3.2	2.2	2.5	2.3	V
1	5	9.1	8.4	7.7	7.7	7.7	7.7	7.7	8.7	8.0	8.7	7.3	8.0	8.0	9.6	8.2	8.2	8.4	7.7	7.7	7.0	7.7	
2	5	8.5	8.0	7.5	7.5	7.5	7.5	7.5	8.4	8.0	8.4	7.5	8.0	8.0	9.5	8.5	8.5	8.0	7.6	7.6	7.1	7.6	
3	5	7.4	7.2	7.1	7.1	7.1	7.1	7.1	7.7	7.6	7.7	7.5	7.6	7.6	9.1	9.0	9.0	7.8	7.7	7.7	7.6	7.7	
4	5	7.1	7.1	7.1	7.1	7.1	7.1	7.1	7.3	7.3	7.3	7.3	7.3	7.3	9.0	9.0	9.0	7.9	7.9	7.9	7.9	7.9	C
5	5	6.8	6.8	6.8	6.8	6.8	6.8	6.8	7.0	7.0	7.0	6.8	6.8	6.8	8.6	8.6	8.6	7.6	7.6	7.6	7.6	7.6	
6	5	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.2	6.2	6.2	6.1	6.1	6.1	7.9	7.9	7.9	6.9	6.9	6.9	6.9	6.9	
7	5	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0	
8	5	6.1	6.1	6.0	6.1	6.0	6.1	6.1	5.8	5.8	5.9	5.8	5.8	5.8	7.7	7.7	7.7	6.7	6.8	6.8	6.8	6.8	
9	5	7.1	7.7	6.5	7.6	6.4	8.0	7.2	4.1	3.7	3.9	3.5	4.1	3.4	4.9	4.9	4.4	3.5	4.5	4.3	4.4	4.5	
10	5	7.8	7.5	6.5	8.5	6.4	8.0	7.3	3.0	2.9	3.3	2.7	3.0	2.7	3.2	3.4	3.0	2.3	3.1	2.9	3.3	3.5	
11	5	7.3	6.9	6.0	8.0	5.8	7.3	6.6	3.0	2.9	3.4	2.7	2.8	2.8	3.0	3.3	3.0	2.4	3.0	2.8	3.1	3.3	N

x_{dim}	y_{dim}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	Name
12	5	4.6	4.6	3.9	5.1	3.5	4.6	3.5	3.5	3.5	4.0	3.1	3.2	3.6	2.9	2.8	3.3	2.8	3.1	2.5	2.8	2.7	
1	6	7.6	7.4	7.2	7.2	7.2	7.2	7.2	8.2	8.0	8.2	7.8	8.0	8.0	8.8	8.4	8.4	7.2	7.0	7.0	6.8	7.0	
2	6	7.1	7.1	7.0	7.0	7.0	7.0	7.0	7.8	7.8	7.8	7.7	7.8	7.8	8.9	8.8	8.8	7.3	7.2	7.2	7.2	7.2	M
3	6	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.5	7.5	7.5	7.4	7.4	7.4	8.9	8.9	8.9	7.4	7.4	7.4	7.4	7.4	
4	6	6.9	6.8	6.8	6.8	6.6	6.9	6.6	7.0	7.0	7.0	6.7	6.7	6.7	8.3	8.3	8.3	7.3	7.3	7.3	7.3	7.3	
5	6	6.9	6.3	6.3	6.3	6.3	6.9	6.3	7.0	7.0	7.0	6.3	6.3	6.3	7.5	7.5	7.5	6.5	6.5	6.5	6.5	6.5	
6	6	6.4	6.0	6.0	6.0	6.0	6.4	6.0	6.8	6.8	6.8	6.0	6.0	6.0	7.2	7.2	7.2	6.2	6.2	6.2	6.2	6.2	
7	6	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.1	6.1	6.1	6.0	6.0	6.0	6.6	6.6	6.6	6.5	6.5	6.5	6.5	6.5	
8	6	6.5	6.6	6.2	6.8	6.1	6.8	6.5	5.1	5.0	5.1	4.9	5.1	4.9	4.7	4.7	4.5	4.8	5.2	5.1	5.2	5.2	
9	6	7.1	7.7	6.6	7.6	6.4	8.0	7.2	4.3	3.9	4.1	3.3	3.8	3.2	4.1	4.0	3.5	2.9	4.0	3.8	3.8	4.0	
10	6	7.1	7.1	6.1	7.5	6.0	7.5	6.7	3.8	3.7	4.2	3.5	4.1	3.3	3.5	3.5	3.0	2.6	3.4	3.2	3.3	3.5	
11	6	5.3	4.8	4.4	5.0	4.4	5.0	4.1	5.4	5.8	6.9	6.0	7.0	5.1	3.4	3.1	2.7	3.3	3.3	3.4	3.1	3.0	
12	6	3.5	3.3	3.2	3.1	3.2	3.3	2.2	6.6	7.0	8.1	7.2	8.4	6.4	3.4	2.9	2.9	3.5	3.5	3.3	3.0	2.8	
1	7	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	7.6	7.6	7.6	6.3	6.3	6.3	6.3	6.3	
2	7	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	6.5	6.5	6.5	6.5	6.5	
3	7	7.0	6.9	6.9	6.9	6.8	7.0	6.8	7.7	7.7	7.7	7.5	7.5	7.5	8.2	8.2	8.2	6.6	6.6	6.6	6.6	6.6	
4	7	7.0	6.9	6.9	6.9	6.3	7.0	6.3	7.1	7.1	7.1	6.4	6.4	6.4	7.5	7.5	7.5	6.4	6.4	6.4	6.4	6.4	
5	7	6.9	6.5	6.5	6.5	6.0	6.9	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	
6	7	6.6	6.1	6.1	6.1	6.0	6.6	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	K
7	7	6.3	6.0	6.0	6.0	6.0	6.3	6.0	6.7	6.7	6.7	6.0	6.0	6.0	6.3	6.3	6.3	6.0	6.0	6.0	6.0	6.0	
8	7	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.4	6.4	6.4	5.8	5.8	5.8	5.3	5.3	5.3	5.8	5.8	5.8	5.8	5.8	I
9	7	6.1	6.1	6.0	6.1	6.0	6.1	6.1	6.3	6.3	6.3	5.3	5.4	5.3	4.9	4.9	4.9	5.3	5.4	5.4	5.4	5.4	
10	7	6.4	6.5	6.2	6.6	6.1	6.6	6.4	5.9	5.8	5.9	4.6	4.7	4.6	4.6	4.7	4.5	4.5	4.8	4.7	4.8	4.8	
11	7	4.2	3.9	3.9	3.7	4.0	3.9	3.1	7.1	7.6	8.6	7.5	8.7	6.5	3.7	3.4	3.1	3.9	3.8	3.8	3.5	3.3	
12	7	3.4	3.3	3.4	3.1	3.5	3.4	2.3	7.9	8.2	9.0	8.4	9.8	7.5	3.2	3.1	3.0	3.3	3.7	3.2	3.3	3.3	P
1	8	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	7.3	7.3	7.3	6.2	6.2	6.2	6.2	6.2	H
2	8	7.0	6.9	6.9	6.9	6.9	7.0	6.9	7.9	7.9	7.9	7.8	7.8	7.8	7.5	7.5	7.5	6.2	6.2	6.2	6.2	6.2	
3	8	7.0	6.8	6.8	6.8	6.1	7.0	6.1	7.1	7.1	7.1	6.2	6.2	6.2	7.1	7.1	7.1	6.0	6.0	6.0	6.0	6.0	
4	8	6.9	6.7	6.7	6.7	6.0	6.9	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	J
5	8	6.7	6.3	6.3	6.3	6.0	6.7	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	
6	8	6.5	6.0	6.0	6.0	6.0	6.5	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.9	6.9	6.9	6.0	6.0	6.0	6.0	6.0	F
7	8	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.5	6.5	6.5	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	
8	8	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.5	6.5	6.5	5.5	5.5	5.5	5.0	5.0	5.0	5.5	5.5	5.5	5.5	5.5	
9	8	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.7	6.7	6.7	5.3	5.3	5.3	5.0	5.0	5.0	5.3	5.3	5.3	5.3	5.3	G
10	8	5.4	5.2	5.3	5.2	5.3	5.3	5.0	7.0	7.2	7.5	6.0	6.4	5.6	4.6	4.5	4.4	4.7	4.7	4.7	4.6	4.5	
11	8	3.4	3.4	3.5	3.2	3.6	3.5	2.4	7.9	8.2	8.9	8.3	9.6	7.5	3.3	3.1	3.0	3.4	3.7	3.3	3.3	3.3	
12	8	3.3	3.3	3.4	3.1	3.5	3.4	2.3	8.0	8.2	9.0	8.4	9.8	7.6	3.2	3.1	3.0	3.3	3.7	3.2	3.3	3.3	Q

Table 4. Reference vector

8. References

S. Bloom. (Ed.) (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals; *Handbook* I, Cognitive Domain, New York

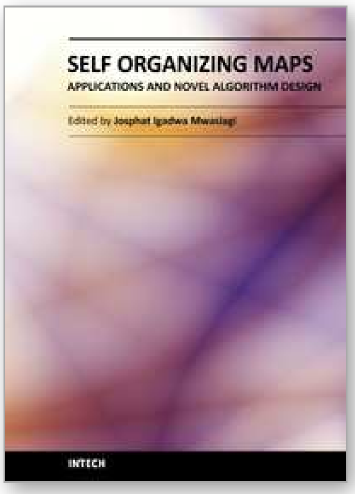
W. B. Boehm. (1988). Spiral Model of Software Development and Enhancement, *IEEE Computer* 21(5): pp. 61-72

M. Furst. & R. A. DeMillo. (2006). Creating Symphonic-Thinking Computer Science Graduates for an Increasingly Competitive Global Environment p.3. (http://www.cc.gatech.edu/images/pdfs/threads_whitepaper.pdf)

A.S. Kate (Ed.) (2002). *Neural Networks in Business: Techniques and Applications*, IGI, Hershey

T. Kohonen. (1995). *Self-Organizing Maps*, *Springer*, New York

J. Surowiecki. (2005). *The Wisdom of Crowds, Anchor*, New York



Self Organizing Maps - Applications and Novel Algorithm Design

Edited by Dr Josphat Igadwa Mwasiagi

ISBN 978-953-307-546-4
Hard cover, 702 pages
Publisher InTech
Published online 21, January, 2011
Published in print edition January, 2011

Kohonen Self Organizing Maps (SOM) has found application in practical all fields, especially those which tend to handle high dimensional data. SOM can be used for the clustering of genes in the medical field, the study of multi-media and web based contents and in the transportation industry, just to name a few. Apart from the aforementioned areas this book also covers the study of complex data found in meteorological and remotely sensed images acquired using satellite sensing. Data management and envelopment analysis has also been covered. The application of SOM in mechanical and manufacturing engineering forms another important area of this book. The final section of this book, addresses the design and application of novel variants of SOM algorithms.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821