ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.12, December 2009, pp.3137−3149
doi: 10.3724/SP.J.1001.2009.00582

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种个体软件过程能力度量方法*

张　晭[1,2+]，王永吉[1,3]，阮　利[4]

[1](中国科学院　软件研究所　互联网软件技术实验室,北京　100190)
[2](中国科学院　研究生院,北京　100049)
[3](中国科学院　软件研究所　计算机科学国家重点实验室,北京　100190)
[4](北京航天航空大学　计算机学院,北京　100191)

Personal Software Process Capability Assessment Method

ZHANG Shen[1,2+], WANG Yong-Ji[1,3], RUAN Li[4]

[1](Laboratory for Internet Software Technologies, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)
[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)
[3](State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)
[4](School of Computer Science and Engineering, BeiHang University, Beijing 100191, China)
+ Corresponding author: E-mail: zhangshen@itechs.iscas.ac.cn, http://itechs.iscas.ac.cn\

Zhang S, Wang YJ, Ruan L. Personal software process capability assessment method. *Journal of Software*, 2009,20(12):3137−3149. http://www.jos.org.cn/1000-9825/582.htm

Abstract: Personal Software Process (PSP) was introduced by Watts Humphrey in CMU/SEI. It is a measured software process aiming at individual software engineers. With the increasing industrial demand for software process improvement, PSP has become a hot topic for software organizations to achieve the goal of total (from macro to micro) quantitative process management. Since higher process capability is recognized as a determinant of better project performance, it is a critical step to assess the personal software process. However, the assessment of PSP capability exhibits Variable Return to Scale (VRS), Multi-Input-Multi-Output (MIMO) and Decision-Making preference problems, which makes existing traditional assessment methods ineffective. In this paper, a novel Personal Software Process Assessment method by synthesizing Data Envelopment Analysis (DEA) and Analytical Hierarchy Process (AHP)—PSPADA is proposed. PSPADA's hybrid model and fundamental assessment algorithms (incorporating decision-making preferences and estimating return to scale) are introduced. Experimental results show that the proposed PSPADA model would be particularly helpful in assessing the capability of personal software processes under the MIMO and VRS constraint, by incorporating Decision-Making preferences.

Key words: personal software process (PSP); data envelopment analysis (DEA); analytic hierarchy process (AHP); variable return to scale (VRS); multi-input-multi-output (MIMO)

---

* Supported by the National Natural Science Foundation of China under Grant Nos.90718042, 60873072 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2007AA010303, 2007AA01A127 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2007CB310802 (国家重点基础研究发展计划(973))

Received 2008-07-25; Revised 2009-01-14; Accepted 2009-03-31

摘 要: 个体软件过程(PSP)是由卡内基·梅隆大学软件工程研究所的 Humphrey 领导开发的.它是一种可用于控制、管理和改进个人工作方式的自我持续改进过程.随着工业界对软件过程改进需求的日益增长,PSP 成为了软件组织为达成完全(从宏观到微观)量化过程管理研究中的一个热点课题.软件过程研究表明,高水平的个体软件过程能力是软件项目成功的关键,如何进行有效的个体软件过程能力度量是 PSP 中的一个核心问题.现有方法不能同时有效处理个体软件过程能力度量中的可变规模收益、多变量输入/输出以及决策者偏好问题.提出了一种综合了数据包络分析(DEA)和层次分析法(AHP)的个体软件过程能力评价方法——PSPADA,介绍了 PSPADA 的个体软件过程能力评价模型和核心算法(集成决策者偏好和估计规模收益).实验结果显示,PSPADA 能够在考虑决策者偏好的同时,有效地进行多指标、规模收益可变的量化评估.

关键词: 个体软件过程(PSP);数据包络分析(DEA);层次分析法(AHP);可变规模收益(VRS);多变量输入/输出(MIMO)

中图法分类号: TP311 **文献标识码**: A

1 Introduction

With the increasing industrial demand for software process improvement[1,2], Personal Software Process (PSP) recently becomes a hot topic for software organizations to achieve the goal of total (from macro to micro) quantitative process management. PSP was introduced in 1995 by Watts Humphrey in CMU[2]. It is a measured software process aimed at individual software engineers. Since higher process capability is recently recognized as a determinant of better project performance, it is a critical step to assess the personal software process and then galvanize the individual to take action on needed improvements immediately following the assessment. However, there are three problems needed to be solved during the assessment.

Firstly, as proposed in Ref.[2], multiple metrics should be incorporated when performing the assessment of PSP capability. For example, as the PSP assessment team usually shows interest in the project schedule, the accuracy of schedule estimation will be used as a metric of PSP capability. Meanwhile, the defect density should be incorporated if the assessment team needs to pay particular attention to the product quality. On the other hand, as Boehm stated in Ref.[3], software is composed of a hierarchy of modules, each of which can connect its inputs to its outputs. Furthermore, a software process can be defined as a collection of process elements, which consume the inputs and produce the outputs respectively. In a word, the assessment of PSP capability is doubtless a multi-input-multi-output (MIMO) problem that must take the needed metrics into account.

Secondly, Stensrud, *et al*.[4] points out that small and large software projects exhibit Variable Return to Scale (VRS, i.e. the relationship between the input and the output is non-linear), whereas medium software projects probably exhibit Constant Return to Scale (CRS, i.e. the relationship between the input and the output is linear). Since the size of programs developed in PSP usually ranges from 50 LOC (line of code) to 5000 LOC, the PSP project exhibits VRS.

Finally, as one of the basic principles stated in CMMI, the capability assessment of software process should be consistent with the organizational objectives and managerial strategies[1], CMMI assure the consistency of the assessing results through interviews, workforce discussion and questionnaires. Similarly, when PSP is applied in industry, there generally exists Decision Makers (DM) with preferences as to which of the input/output metrics they consider to be “more important”, “equally important” or “less important” metrics. For example, because of higher expected estimation accuracy in schedule and effort, some DM may rank CPI more beneficial than defect ratio and process yield (CPI, defect ratio and process yield are PSP recommended metrics in Table 4). To sum up, these preferences should be incorporated within the assessment in order to bring the results closer to the improvement goals.

In this paper, we propose a novel Personal Software Process Assessment method by incorporating Data Envelopment Analysis (DEA) and Analytical Hierarchy Process (AHP)—PSPADA, which can deal with the multivariate input/output, VRS and managerial preference problems simultaneously. The PSPADA method can be regarded as an straightforward extension of our previous work in Ref.[5] by introducing mechanism of incorporating Decision-Making preferences, and it also builds on our previous work[6] by scaling the DEA-based projects assessment method down to fine-grained personal software processes.

This paper is outlined as follows. Section 2 discusses the state of the art. Section 3 first presents PSPADA's hybrid assessment model based on DEA and AHP, then detailed PSPADA's fundamental assessment algorithms of incorporating Decision-Making preference and estimating return to scale. To verify the proposed method, an experiment is demonstrated and its results are analyzed in Section 4. Section 5 closes with a conclusion.

2 Related Work

The most recently widely-used improvement model and methods CMMI/TSP/PSP classify software process improvement into three levels: organization level, team level and individual level. Moreover, since CMMI aims to support software process improvement at the organizational-level, organizational software process is the key capability assessment component to measure the organizational capability[1]. While PSP focuses on the software process improvement at the individual-level, personal software process is the key assessment component to reflect personal capability[2].

To date, in software process improvement field, the most recent research mainly focuses on projects[4,7,8] so as to realize the organizational process improvement. On the other hand, due to research of Personal Software Process (PSP), the current trend of software process improvement is "scaled down" to the level of individual developers. Personal software process capability assessment is vital for any developer seeking to continuously improve the current personal practices. However, although there have been deep research on assessing organizational software process to achieve quantitative macro process management, the assessment of fine-grained personal software processes, which is a key step to achieve quantitative micro personal software process management, is simply ignored. Moreover, existing literature has proposed few assessment methods that explicitly consider their multivariate input/output and VRS constraints by incorporating Decision-Making preferences. Statistical methods[9] propose to compare the process capability with some theoretical optimal ones (e.g. theoretical baselines)[7]. However, as Ref.[4] recently reports, in software engineering, it seems more sensible to compare the capability with the best practice rather than with some theoretical optimal (and probably non-attainable) performance. We have presented a method to evaluate the software project quality by mining the bug reports from bug tracking systems in Ref.[6]. It focuses on macro view of software process which is insufficient to reflect the capability at the individual-level. We have also presented our experiences on mining libre software repositories for PSP metrics[5]. However, the impact of preference factors hasn't been taken into consideration in our previous work. To sum up, existing assessment researches in software process field can't deal with the multivariate input/output, VRS and Decision-Making preference problems simultaneously.

Data Envelopment Analysis (DEA) developed by Charnes and Cooper[10] in 1978 is a non-parametric programming-based performance assessment model. It can be used to analyze the relative capability of a number of units, which can be viewed as a multi-input-multi-output system consuming inputs to produce outputs. Recently, DEA is gaining increasing interests in software process field[4−8] after Stensrud, *et al*. first introduced it into software project assessment in 1999[7]. DEA gains interests mainly because it provides a powerful unique advantage of MIMO and VRS. Ref.[4] especially points out that "DEA is the only method complying with the two requirements (multivariate inputs/outputs and VRS) that we consider crucial to perform correct performance assessment in software engineering".

The objective non-parametric methods used in DEA have also been criticized for not being able to reflect the managerial preference[11]. Traditional DEA models by incorporating preference may lead to a more reasonable result, which is expected to be consistent with the organizational objectives and managerial strategies. This brought forward the cone ratio DEA model (C^2WH) which provides a framework to incorporate the Decision-Making preferences[12]. However, with a general preference selected in practice, we can only ambiguously recognize that one metric is more or less important than another in the decision maker's preference[11]. Therefore, an improved assessment model is needed to clarify the precise meaning of fuzzy perception concept like "more important" or "equally important" and to impose preference restrictions for the input or output metrics. In this paper, we introduce analytic hierarchy process method (AHP) into DEA to solve the capability assessment problem involving multi-input-multi-output, VRS and Decision-Making preferences. The AHP allows decision makers to specify their preferences using a verbal scale[13]. This verbal scale will be very useful in helping a group or an individual to make a fuzzy decision. Therefore, in this paper, AHP is used to introduce preference information in DEA calculations. Preference information is introduced in the form of subjective pairwise comparison matrix generated using AHP.

3 PSPADA's Capability Assessment Model Description

In this section, we will formulate and discuss our PSPADA's capability assessment model based on DEA and AHP. Besides, PSPADA's three fundamental assessment algorithms (incorporating decision-making preferences and estimating return to scale) will be introduced in the following subsections.

Let us assume that there are n PSPs to be evaluated. Each PSP is a multi-input-multi-output (MIMO) process, which consumes varying amounts of m different inputs to produce s different outputs.

Definition 3.1. The Personal Software Process Set (P): The process set is defined as $P=(P_1,P_2,...,P_n)$. The basic requirement is that the n processes are homogeneous which can be efficiently assessed on their relative capability.

Definition 3.2. The Input Metrics (I): It denotes the m input metrics of P. $I=(I_1,I_2,\ldots,I_m)$. Input metrics can be any factors used as a resource by the PSP for producing something of value. It may also be any environmental factor that has a strong effect on how resources are consumed.

Definition 3.3. The Output Metrics (O). It denotes the s output metrics of P. $O=(O_1,O_2,\ldots,O_s)$. Output metrics are the amounts of code lines, documents or other outcomes obtained by processing resources or, also, any factor that describes the qualitative nature of such an outcome.

Definition 3.4. The Personal Software Process (P_j). Each process P_j ($P_j\in P$) is defined as: $P_j=(X_j,Y_j)$, where X_j denotes the m inputs of P and Y_j denotes the s outputs of P_j.

Definition 3.5. The Input Set (X_j) Each process P_j's input is defined as $X_j=(x_{1j},x_{2j},\ldots,x_{mj})^T>0$, $j=1,\ldots,n$, where x_{bj} denotes the amount of the b th input metric (I_b) consumed by P_j and $x_{bj}>0$.

Definition 3.6. The Output Set (Y_j). Each process P_j's output is defined as $Y_j=(y_{1j},y_{2j},\ldots,y_{sj})^T>0$, $j=1,\ldots,n$, where y_{kj} denotes the amount of the kth output metric (O_k) produced by P_j and $y_{kj}>0$.

It should be noted that there are several principles about the selection of input/output metrics in our study. These principles are concluded from Ref.[14].

Firstly, since the highest capability score is assigned to the PSP which has the maximal ratio (weighted sum of outputs/weighted sum of inputs), we prefer the smaller values of input metrics and bigger values of output metrics.

Secondly, we must consider the relationship of the input and output metrics. Because the PSPs' input and output metrics are not isolated, the metrics which have been regarded as input or output can influence the cognizance of other metrics. For example, we should discard a metric if the information of it has been covered by other several metrics or has strong relationship with some other input/output metrics.

Thirdly, there should be no more than ten input and output metrics for every assessment process, the reason is that employing too many input and output metrics will tend to overestimate capability.

Fourthly, we have to get all the values of input and output metrics for all the PSPs. Note that they must be all positive values.

Fifthly, different input or output metrics can have different measurement units, such as the man-month, the function-point, the KLOC, etc.

We first establish the personal process capability assessment models (see Table 1) by synthesizing the cone ratio DEA model (C^2WH)[15] and AHP. The PSPADA assessment model can be expressed in linear program (LP) form (1) and dual form (2) as shown in Table 1.

Table 1 PSPADA— $P_{\mathrm{C^2WH-AHP}}$ and $D_{\mathrm{C^2WH-AHP}}$ model

$(P_{\mathrm{C^2WH-AHP}}) = \begin{cases} \theta_u = \max(\mu^T y_u) \\ \omega^T x_j - \mu^T y_j \ge 0, j = 1,\dots,n \\ \omega^T x_u = 1 \\ \omega \in V_{AHP}, \mu \in U_{AHP} \end{cases}$	(1)
$(D_{\mathrm{C^2WH-AHP}}) = \begin{cases} \min(\theta_u) \\ \sum_{j=1}^{n} x_j \lambda_j - \theta x_u \in V_{AHP}^* \\ -\sum_{j=1}^{n} y_j \lambda_j + y_u \in U_{AHP}^* \\ \lambda_j \ge 0,\ j = 1,2,\dots,n \end{cases}$	(2)

In our PSPADA's capability assessment model, the Decision-Making preference is introduced in the form of V_{AHP}, V_{AHP}^*, U_{AHP} and U_{AHP}^* generated using AHP at first, and then they are incorporated into DEA calculations for capability assessment.

The scalar variable θ in (1) and (2) represents the nonnegative capability score of each PSP, and it ranges from 0 to 1. If P_u ($P_u \in P$) receives the optimal value θ_u=1, then it is of relative high capability, but if θ_u<1, it is of relative low capability.

Furthermore, since the value of θ_u means that the P_u can still achieve a minimal decrease of θ_u times in its inputs without decreasing the production for any outputs, the capability of P_u is relatively low when the θ_u is relatively small.

It should be noted that PSPADA is a relative method and compares each PSP's capability with all other PSPs in the same process set P, so PSPADA can only be used for the assessment of relative capability, not absolute capability. Here, the *relative* means that the capability of P_u is a comparative measure based on the process set P used in these models (see Table 1).

Besides the capability score θ (see model (1) in Table 1), we also calculate another two variables: ω and μ. In the following definitions, we give an explanation of the practical meanings of ω and μ.

Definition 3.7. The preference weights of input metrics (ω):$\omega=(\omega_1,\omega_2,\dots,\omega_m)^T$, where ω_b reflects the relative importance of the input metric (I_b) in Decision-Making preferences.

Definition 3.8. The preference weights of output metrics (μ):$\mu=(\mu_1,\mu_2,\dots,\mu_s)^T$, where μ_k reflects the relative importance of the input metric (O_k) in Decision-Making preferences.

Based on the PSPADA assessment models (see Table 1), the following subsections will detail PSPADA's fundamental assessment algorithms of incorporating Decision-Making preferences and estimating return to scale.

3.1 Incorporating decision-making preferences

In order to bring the capability assessment results closer to the improvement goals of management, in this subsection we will introduce how we incorporate Decision-Making preferences into PSPADA's assessment models. The preferences are aggregated into "preference cones", which are in the form of V_{AHP} and U_{AHP} (see (1) in Table 1).

Definition 3.9. Input Preference Cone (V_{AHP}): $(V_{AHP} \subset E_m^+, IntV_{AHP} \neq \varnothing)$ is a closed convex cone that can be used to reflect the relative importance of each input metrics with respect to Decision-Making preferences.

Definition 3.10. Output Preference Cone (U_{AHP}): $(U_{AHP} \subset E_s^+, IntU_{AHP} \neq \varnothing)$ is a closed convex cone that can be used to reflect the relative importance of each output metrics with respect to Decision-Making preferences.

It should be noted that V_{AHP}^* and U_{AHP}^* (see (2) in Table 1) are the negative polar cones of V_{AHP} and U_{AHP}. They are also defined as the preference cones in our study. For a detailed explanation of the closed convex cone and the negative polar cone, the reader may refer to Ref.[16].

Definition 3.11. Input Decision-Making Matrix (A_m): $A_m=(a_{ij})_{m\times m}$. Pair-wise comparisons among $I=(I_1,I_2,\ldots,I_m)$ lead to an approximation value of a_{ij}, a_{ij}'s value denotes the ratio of the relative importance of I_i to I_j. Each entry $a_{ij}\in A_m$ is governed by three rules: 1) $a_{ij}>0$; 2) $a_{ij}=1/a_{ji}$; 3) $a_{ii}=1$.

Definition 3.12. Output Decision-Making Matrix (B_s): $B_s=(b_{ij})_{s\times s}$. Pair-wise comparisons among $O=(O_1, O_2,\ldots,O_s)$ lead to an approximation value of b_{ij}. b_{ij}'s value denotes the ratio of the relative importance of O_i to O_j. Each entry $b_{ij}\in B_s$ are governed by three rules: 1) $b_{ij}>0$; 2) $b_{ij}=1/b_{ji}$; 3) $b_{ii}=1$.

In Definitions 3.11 and 3.12, the scale of relative importance, which precisely measures the Decision-Making preferences over input/output metrics, is defined in Table 2 according to Satty 1~9 scale[13] for pair-wise comparison.

Table 2 Scale of importance

Intensity of importance	Definition	Intensity of importance	Definition
1	Equal importance	7	Very strong or demonstrated importance
3	Moderate importance	9	Extreme importance
5	Strong importance	2, 4, 6, 8	For compromise between the above values

In our PSPADA method, a consistency check is required to identify inconsistent matrix (with unacceptable deviations). The value of consistency ratio (CR) reflects the level of inconsistency of a Decision-Making Matrix C. if $CR\le 0.1$, the matrix C is considered to be consistent; if $CR>0.1$, the matrix C is considered to be inconsistent.

In our experiment, CR is computed for each Decision-Making matrix by using the eigenvector method (EVM).

Based on the above definitions, PSPADA's algorithm to incorporate the Decision-Making preferences into the assessment process is presented in Algorithm 1. In Algorithm 1, we collect the Decision-Making preferences from the organizational decision makers by using 1~9 scale in Table 2, compute each entry $a_{ij}\in A_m$ and $b_{ij}\in B_s$ based on AHP group decision making theory, and then establish the Decision-Making matrix A_m and B_s. Furthermore, if CR for the Decision-Making matrix is more than 0.1, we must recollect the preferences and repeat the steps above.

Algorithm 1. Incorporating decision-making preferences.

Input: Input Metrics $I=(I_1,I_2,\ldots,I_m)$. Output Metrics $O=(O_1,O_2,\ldots,O_s)$. Decision-Making preferences DMP, Input Decision-Making Matrix A_m, Output Decision-Making Matrix B_s;

Output: Input Preference Cone V_{AHP} and V_{AHP}^*, Output Preference Cone: U_{AHP} and U_{AHP}^*.

While (TRUE) do

For all $a_{ij}\in A_m$ $(i,j\in[1,m])$ do
  $a_{ij}=PairWise(I_i,I_j,DMP)$ //compute each entry a_{ij} based on AHP group decision making theory
End for
$CR_A=ClaculateCR(A_m)$
If $(CR_A\le 0.1)$ then Break
  End if
End while
$\gamma_A=ClaculateMaxEigenValue(A_m)$ //calculate the max eigenvalue γ_A of A_m
$\bar{A}=A_m-\gamma_A E_m$ //the E_m is an identity matrix
$V_{AHP}=\{\omega \mid \bar{A}\omega\ge 0,\ \omega\ge 0\}$
$V_{AHP}^{*}=\{\bar{A}^{T}\omega \mid \omega\le 0\}$
While (TRUE) do
  For all $b_{ij}\in B_s$ $(i,j\in[1,s])$ do
    $b_{ij}=PairWise(O_i,O_j,DMP)$
  End for
  $CR_B=ClaculateCR(B_s)$
  If $(CR_B\le 0.1)$ then
    Break
End while
$\gamma_B=ClaculateMaxEigenValue(B_s)$ //calculate the max eigenvalue γ_B of B_s
$\bar{B}=B_s-\gamma_B E_s$ //the E_s is an identity matrix
$U_{AHP}=\{\mu \mid \bar{B}\mu\ge 0,\ \mu\ge 0\}$
$U_{AHP}^{*}=\{\bar{B}^{T}\mu \mid \mu\le 0\}$
Return V_{AHP} and U_{AHP}.

3.2 Estimating return to scale

By imposing an additional restriction $\sum\lambda=1$, the $D_{\mathrm{C^2WH-AHP}}$ model (see (2) in Table 1) can be transformed into an extended assessment model $D_{\mathrm{C^2WY-AHP}}$ (see Table 3), which is based on DEA $\mathrm{C^2WY}$ model and AHP. The DEA $\mathrm{C^2WY}$ model is developed to orient a limited number of units under assessment[11]. As mentioned in Ref.[4,7], the $\mathrm{C^2WY}$ model is especially capable of handling the VRS issue for capability assessment in software engineering.

Table 3 PSPADA— $D_{\mathrm{C^2WY-AHP}}$ model

$$(D_{\mathrm{C^2WY\text{-}AHP}})=\begin{cases}\min(\varphi_u)\\ \sum_{j=1}^{n}X_j\lambda_j-\varphi X_u\in V_{AHP}^{*}\\ -\sum_{j=1}^{n}Y_j\lambda_j+Y_u\in U_{AHP}^{*}\\ \sum_{j=1}^{n}\lambda_j=1\\ \lambda_j\ge 0,\ j=1,2,\dots,n\end{cases}\qquad(3)$$

The variable Φ in $D_{\mathrm{C^2WY-AHP}}$ has the same meaning as θ in Table 1. However, its value may be different from θ with the restriction $\sum\lambda=1$. As the kernel of PSPADA is DEA and DEA has been shown to give a very good estimation for return to scale[17]. Based on $D_{\mathrm{C^2WH-AHP}}$ and $D_{\mathrm{C^2WY-AHP}}$, we can perform return to scale analysis on each P_j's $(P_j\in P)$ to get the personal software process improvement direction, for a complete description of the theoretical foundation, interested readers may refer to Refs.[10,16].

PSPADA's algorithm to analyze the P_j's return to scale is presented in Algorithm 2. There are three types of return to scale:

1. Increasing returns to scale (IRS): an increase in the Inputs Set (X_j) results in a more than proportionate increase in the Output Set (Y_j);

2. Constant returns to scale (CRS): an increase in the Inputs Set (X_j) results in an equal proportionate increase in the Output Set (Y_j);
3. Decreasing returns to scale (DRS): an increase in the Inputs Set (X_j) results in a less than proportionate increase in the Output Set (Y_j).

Algorithm 2. Return to scale analysis.

Input: The PSP set $P=(P_1, P_2,...,P_n)$;
Output: The PSP set with P of IRS: P_{IRS}, The PSP set with P of CRS: P_{CRS}, The PSP set with P of DRS: P_{DRS}.

```
P_IRS=∅              //Initialize PSP set
P_CRS=∅
P_DRS=∅
For all P_j∈P (j∈[1,n]) do
    θ_j=CalculateCapabilityScore(P_j,P, D_{C²WH−AHP})
    Φ_j=CalculateCapabilityScore(P_j, P, D_{C²WY−AHP})
    If (θ_j=Φ_j) then
        P_CRS=P_CRS∪P_j
    Else If (θ_j<Φ_j) then
        λ=CalculateLamda(P_j,P, D_{C²WH−AHP})
        If (∑λ >1) then
            P_DRS=P_DRS∪P_j
        Else if (∑λ <1) then
            P_IRS=P_IRS∪P_j
        End if
    End if
End for
Return P_IRS, P_CRS, P_DR
```

When a personal software process P_j has increasing returns to scale ($P_j \in P_{IRS}$), the P_j itself is suggested to increase its inputs to obtain a higher proportionate increase in outputs; when P_j has decreasing returns to scale ($P_j \in P_{CRS}$), the P_j itself is suggested to slow down the resource expansion, then turns to make improvements in technique, knowledge and skills.

4 Experimental Results and Analysis

In this experiment, we present experiment results of PSPADA's mechanisms of incorporating Decision-Making preferences, establishing reference sets and estimating return to scale on a standard and representative PSP dataset selected from Putz's book[34].

All the metrics in Table 4 are derived from the Project Plan Summary of PSP[2]. Among these metrics, the "Total Schedule" can be taken as the Input Metrics (I), while the rest will be chosen as the Output Metric (O) used in our assessment models. Therefore, only the preference weights of output metrics (μ) will be gathered and used to impose preference restrictions for the output metrics (O). In our experiment, we derive six sample metrics (Table 5) from "PSP recommended metrics" in Section 3 for personal process capability assessment.

The time consumed by development processes should be taken as the input metric, while other metrics are regarded as the output metrics in our PSPADA method. Among the output metrics, since PSP regards defect reduction and the accuracy of process improvement estimation as the two primary goals of personal process improvement[2], the "Scale Estimation Accuracy" and "Time Estimation Accuracy" are chosen to describe process capability in estimation accuracy, while the "Reciprocal of Defect Density" and "Process Yield" are used to measure the improvement in defect reduction.

Table 4 PSP recommended metrics

Metric	Formula
Total schedule	The sum of planned or actual time for all phases of a project
Scale	Source line of code
Review rate	60*(New and changed code in LOC)/Review minutes
Time ratios	Design time/Coding time Design review time/Design time Code review time/Coding time
Defect ratios	Remove defects in code review/Defects when compiling Removed defects in design review/Defects when unit test
Process yield	Removed defects before compiling and unit test/Total defects
Phase yield	Defects at entry/Defects at ending
A/FR	(Design review time+Code review time)/(Compiling time+Unit test time)
LOC/Hour	Total new and changed code in LOC/Total schedule in hour
CPI	Planned time/Actual time
Reuse rate	Reused LOC/Total LOC
Increased reuse rate	The new increased reuse code LOC/New and changed code in LOC
Defects/KLOC	1000*(Defects removed in test)/Actual new and changed LOC
Defect density	1000*(Total defects removed)/Actual new and changed LOC

Table 5 Input and output assessment metrics of personal software process

Metric	Formula	Type	Meaning
Schedule (I_1)	Development time (minute)	Input	Activity input (or investment)
Scale (O_1)	Source line of code	Output	Product scale
Scale estimation accuracy (O_2)	10/(|Planned Scale−Actual Scale|/ Actual Scale)	Output	Ability of scale estimation
Time estimation accuracy (O_3)	10/(|Planned Schedule−Actual Schedule|/ Actual Schedule)	Output	Ability of schedule estimation
Reciprocal of defect density (O_4)	10^4/(Total Defects/Scale in KLOC)	Output	Product quality
Process yield (O_5)	Number of defects removed before compiling and unit test/Total defects	Output	Process performance

It should be noted that there are three metrics requiring transforming in Table 5: the "Scale Estimation Accuracy", the "Time Estimation Accuracy" and the "Defect Density". Because an increase in an input should contribute to increased output, these three metrics, which are undesirable outputs in DEA terminology, should be transformed. Therefore, the reciprocal transformation is applied to these metrics as the formulas shown in Table 5.

Based on sample metrics defined in Table 5, a dataset containing 10 PSPs is derived from the project plan summary data in Ref.[34] and shown in Table 6.

Table 6 Measures derived from project plan summaries listed in Ref.[34]

P	Schedule	Scale (LOC)	Size estimation accuracy	Schedule estimation accuracy	Reciprocal of defect density	Process yield
P_1	114	94	35	48	78	58
P_2	214	233	30	55	75	71
P_3	310	263	23	15	109	50
P_4	188	236	63	28	157	87
P_5	182	178	32	455	93	89
P_6	315	568	45	53	75	67
P_7	198	678	72	66	43	95
P_8	393	458	49	18	63	87
P_9	342	824	25	60	74	80
P_{10}	498	1 202	85	23	61	85

4.1 Incorporating decision-making preferences

After the dataset has been established, we next apply PSPADA's Algorithm 1 to the dataset to introduce Decision-Making preferences by establishing preference cones. Since there are five output metrics plus one input metric, we only consider the impact of Decision-Making preferences on the output metrics.

In our study, the Decision-Making preferences are gathered from 6 project managers in ISCAS regarding the 5 output metrics in Table 5. By referring to Algorithm 1 (see Section 3.1), these managers quantitatively mark their preferences on the “relative importance of each output metric” by considering the two primary goals of PSP—estimation accuracy improvement and defect reduction. When these managers emphasize more needs for estimation accuracy improvement than defect reduction, we obtain a Decision-Making matrix $B_{estimation}$. When these managers pay more attention to defect reduction than estimation accuracy improvement, we obtain another Decision-Making matrix $B_{quality}$. The two matrices are given in Table 7.

Table 7 Decision-Making matrices: $B_{estimation}$ and $B_{quality}$

$$B_{estimation}=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{3} & 4 & 3\\ 2 & 1 & \frac{1}{2} & 7 & 5\\ 3 & 2 & 1 & 8 & 7\\ \frac{1}{4} & \frac{1}{7} & \frac{1}{8} & 1 & \frac{1}{2}\\ \frac{1}{3} & \frac{1}{5} & \frac{1}{7} & 2 & 1\end{bmatrix},\quad B_{quality}=\begin{bmatrix}1 & 5 & 3 & \frac{1}{3} & \frac{1}{4}\\ \frac{1}{5} & 1 & \frac{1}{2} & \frac{1}{9} & \frac{1}{7}\\ \frac{1}{3} & 2 & 1 & \frac{1}{7} & \frac{1}{5}\\ 3 & 9 & 7 & 1 & \frac{1}{2}\\ 4 & 7 & 5 & 2 & 1\end{bmatrix}$$

Since the two matrices meet the consistency requirements through the consistency validation with $CR\le 0.1$, we construct the output preference cones $U_{estimation}$ and $U_{quality}$ by using Algorithm 1:

$$\bar{B}_e=B_{estimation}-\gamma_{be}E_5\ U_{estimation}=\{\mu\,|\,\bar{B}_e\mu\ge 0,\ \mu\ge 0\}\Rightarrow U^*_{estimation}=\{\bar{B}_e^T\mu\,|\,\mu\le 0\},$$
$$\bar{B}_q=B_{quality}-\gamma_{bq}E_5\ U_{quality}=\{\mu\,|\,\bar{B}_q\mu\ge 0,\ \mu\ge 0\}\Rightarrow U^*_{quality}=\{\bar{B}_q^T\mu\,|\,\mu\le 0\}.$$

By incorporating these preference cones into the model (1) in Table 1, the capability assessment results of these 10 PSPs are calculated and shown in Fig.1.

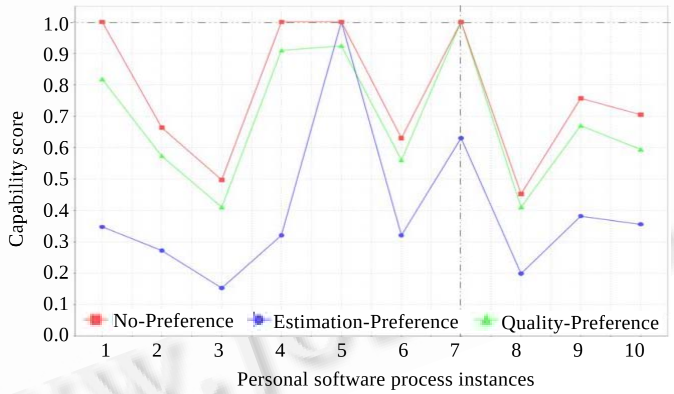


Fig.1 Capability score distribution with various output preference cones

In Fig.1, the blue and green series represent separately the capability score derived from model (1) incorporating $U_{estimation}$ and $U_{quality}$. Besides, to clarify the results of comparison study, we also calculate the capability score θ without regarding any managerial preference as depicted by the red series.

As is stated in Section 3, the PSPs, whose capability scores equal 1, are identified as relatively high capability ones. However, from the above figures, it can be observed that the same processes' capability scores with different preferences may sometimes vary significantly from each other.

For example, as shown in Fig.1, P_7 has the best capability score θ=1 (U_{AHP}=*Null* or $U_{AHP}=U_{quality}$), but its capability score θ is just around 0.6 when assessment model incorporates an “estimation accuracy” preference cone. The reason for this difference can be attributed to the impact of Decision-Making preferences. To help clarify these issues, we adopt model (1) in Table 1 to calculate the preference weights of output metrics (μ) and explain the discrepancy. The weights vector μ for output metrics under different restrictions are shown below.

$$P_7=\begin{cases}\text{without output preference cone,} & \theta=1.000,\ \mu=(1.898,\ 0.295,\ 0.481,\ 2.593,\ 3.312)^T\\ \text{output preference cone}=U_{estimation}, & \theta_p=0.630,\ \mu=(0.612,\ 1.089,\ 1.625,\ 0.158,\ 0.243)^T\\ \text{output preference cone}=U_{quality}, & \theta_q=1.000,\ \mu=(0.917,\ 0.242,\ 0.397,\ 1.769,\ 2.721)^T\end{cases}.$$

It is obvious that distinct output weights have imposed preference restrictions for the output metrics. While the assessment model adopts the preference cone $U_{estimation}$, the "estimation accuracy" related metrics have the greatest impact on the final capability assessment results. Since P_5 holds a very high value for "estimation accuracy" ("Schedule Estimation Accuracy"=455) against the other 9 PSPs, the capability scores of the other 9 processes generally appear with rather low values. This can explain the reason why P_7's capability score θ_j varies significantly with different output preference cones.

In Fig.1, when the Decision-Making preferences aren't incorporated within the assessment model as depicted by the red series, our proposed assessment method will only enable a purely mathematical process for relative capability rating, which is totally dependent on the objective input and output metric data. The result is that this method may fail to ensure consistency with the managerial or economic objectives. Specifically, unreasonable low or high bounds are often placed on the input/output metrics due to the dataset characteristics in DEA model. For example, In Fig.1, the red curve follows a similar distribution as the green curve, so it is obvious that the capability assessment without output preference cone indeed leads to a quality bias, which may seriously contradict the Decision-Making preferences for estimation accuracy improvement.

4.2 Estimating return to scale

We next perform the return to scale analysis by applying Algorithm 2. The return to scale analysis will help PSP users to make a decision on an expansion or a reduction in software scale. Using Algorithm 2, PSPADA first calculates the capability score (θ_j) and (Φ_j) for each P_j ($P_j \in P$) based on $D_{\mathrm{C^2WH-AHP}}$ and $D_{\mathrm{C^2WY-AHP}}$ (see Table 1 and Table 3), then the $\sum\lambda$ are also derived from $D_{\mathrm{C^2WH-AHP}}$. The PSPs' return to scale properties, which indicate whether a process has IRS or DRS, are listed in Table 8.

Table 8 Return to scale of 10 PSPs

P	Φ_j/θ_j	$\sum\lambda$	Return to scale	$P(U_{estimation})$	Φ_j/θ_j	$\sum\lambda$	Return to scale	$P(U_{quality})$	Φ_j/θ_j	$\sum\lambda$	Return to scale
P_1	=1	=1	CRS	P_1	>1	>1	DRS	P_1	>1	>1	DRS
P_2	>1	>1	DRS	P_2	>1	>1	DRS	P_2	>1	>1	DRS
P_3	>1	<1	IRS	P_3	>1	<1	IRS	P_3	>1	<1	IRS
P_4	=1	=1	CRS	P_4	>1	>1	DRS	P_4	>1	>1	DRS
P_5	=1	=1	CRS	P_5	=1	=1	CRS	P_5	>1	>1	DRS
P_6	>1	>1	DRS	P_6	>1	>1	DRS	P_6	>1	>1	DRS
P_7	=1	=1	CRS	P_7	>1	>1	DRS	P_7	=1	=1	CRS
P_8	>1	>1	DRS	P_8	>1	>1	DRS	P_8	>1	>1	DRS
P_9	>1	>1	DRS	P_9	>1	>1	DRS	P_9	>1	>1	DRS
P_{10}	>1	>1	DRS	P_{10}	>1	>1	DRS	P_{10}	>1	>1	DRS

In Table 8, we apply the return to scale analysis under different preference constraints. In the first column, the analysis results are calculated without considering any preference, while in the other two columns, we study the return to scale by incorporating $U_{estimation}$ and $U_{quality}$ (see Section 4.1).

In all of the three columns, most of the PSPs exhibit DRS, while only P_2 has IRS under different preference constraints, so the personal software processes likely exhibit decreasing return to scale. Two explanations can be given for the DRS. Firstly, as Brooks stated in Ref.[19], formula (1) is concluded from a study done at System Development Corporation. This formula aims to explore some of the relationships between effort and the increment of scale, and the development effort can be expressed as an exponential function of product scale (KLOC or FP) with a fixed exponent being 1.5. Another SDC study also recommends an exponent near 1.5. But in some previous COCOMO models[20], Boehm's data doesn't at all agree with this, but varies from 1.05 to I.2. However, all of them agree that the project effort increases exponentially with scale and the exponent is larger than 1, so that with increasing product scale the overall productivity goes down.

$$Effort=(Constant)\times(Product\ Scale)^{1.5} \tag{1}$$

Obviously, our results in this study support previous conclusions as well. Besides, the second explanation for the DRS may be that scheduled overtime will eventually lead to a decrease in labor productivity, especially when no new partners are involved in PSP practices for input expansion.

To the IRS personal processes, it can be taken for granted that as far as these involved developers spend more time on code production on the current technical levels, they can get a proportionate increase in quantity and quality of outputs. To the DRS processes, there is a suggestion that these software engineers should consider of slowing down the input scale expansion, then turn to make improvements in the programming techniques and the development process efficiency.

5 Conclusion

In this paper, we propose a novel personal software process assessment method by incorporating DEA and AHP—PSPADA to support quantitative software process improvement. PSPADA's hybrid assessment model and fundamental assessment algorithms (incorporating decision-making preferences and estimating return to scale) are introduced. The PSPADA method can be regarded as an straightforward extension of our previous work in Ref.[5] by introducing mechanism of incorporating Decision-Making preferences, and it also builds on our previous work[6] by scaling the DEA-based projects method down to fine-grained personal software processes.

Experimental results show that the proposed method would be particularly helpful in assessing the capability of personal software processes under the MIMO and VRS constraint, meanwhile incorporating Decision-Making preferences can assure the assessment results to be consistent with the organizational specific objectives. Now we are experimenting on wide-scale industrial applications of incorporating PSPADA into PSP to assist in raising the effeciency of PSP assessment.

References:

[1] CMMI Product Team. Capability maturity model integration, Version 1.1. CMU/SEI-2002-TR-003, ESC-TR-2002-003, Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 2001.

[2] Humphrey W. PSP(sm): A Self-Improvement Process for Software Engineers (SEI Series in Software Engineering). Addison-Wesley Professional, 2005.

[3] Bohem B. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research. IEEE Computer Society Press, 2007.

[4] Stensrud E, Myrtveit I. Identifying high performance ERP projects. IEEE Trans. on Software Engineering, 2003,29(5):387−416.

[5] Zhang S, Wang YJ, Yang Y, Xiao JC. Capability assessment of individual software development processes using software repositories and DEA. In: Proc. of the Int'l Conf. on Software Process (ICSP 2008). Leipzig: Springer-Verlag, 2008. 147−159.

[6] Zhang S, Wang YJ, Tong J, Zhou JH, Ruan L. Evaluation of project quality: A DEA-based approach. In: Proc. of the Int'l Software Process Workshop and Int'l Workshop on Software Process Simulation and Modeling (SPW/ProSim 2006). Beijing: Springer-Verlag, 2006. 88−96.

[7] Myrtveit I, Stensrud E. Benchmarking COTS projects using data envelopment analysis. In: William A, James B, *et al*., eds. Proc. of the 6th Int'l Software Metrics Symp. Washington: IEEE Computer Society, 1999. 269−278.

[8] Farris JA, Groesbeck RL, Aken EMV, Letens G. Evaluating the relative performance of engineering design projects: A case study using data envelopment analysis. IEEE Trans. on Engineering Management, 2006,53(3):471–482.

[9] Florac WA, Carleton AD. Measuring the Software Process: Statistical Process Control for Software Process Improvement. Addison-Wesley Publishing, 1999.

[10] Charnes A, Cooper WW, Seiford LM. Data Envelopment Analysis: Theory, Methodology, and Application. 5th ed., Springer-Verlag, 1994.

[11] Charnes A, Seiford LM, Zhu J. Handbook on Data Envelopment Analysis. 2nd ed., Kluwer Academic, 2004.

[12] Ramanathan R. An Introduction to Data Envelopment Analysis. SAGE, 2003.

[13] Saaty TL, Vargas LG. Models, Methods, Concepts & Applications of the Analytic Hierarchy Process. Springer Netherlands, 2001.

[14] Cooper WW, Seiford LM, Tone K. Introduction to Data Envelopment Analysis and Its Uses: With DEA-solver Software and References. Birkhäuser, 2006.

[15] Ray SC. Data Envelopment Analysis: Theory and Techniques for Economics and Operations Research. Cambridge University Press, 2004.

[16] Mantri JK. Research Methodology on Data Envelopment Analysis (DEA). Universal-Publishers, 2008.

[17] Thanassoulis E. Introduction to the Theory and Application of Data Envelopment Analysis. Springer-Verlag, 2001.

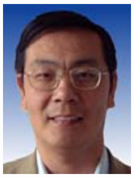
[18] Putz V. The personal software process: An independent study. http://www.nyx.net/~vputz/psp_index/book1.html

[19] Brooks FP. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley Publishing, 1995.

[20] Bohem B. Software Cost Estimation with COCOMO II. Prentice Hall, 2000.

ZHANG Shen is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. His current research areas are software measurement and data mining.

RUAN Li is a teacher at the School of Computer Science and Engineering, Beihang University. Her current research areas are high performance computing and virtualization technology.



WANG Yong-Ji is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences and a CCF senior member. His research areas are real-time systems, artificial intelligence and intelligent software engineering.