A CRITICAL ANALYSIS OF PERSONAL SOFTWARE PROCESS (PSP) & PSP TOOLS SUPPORT

1 ASAD ABBAS, 2 MUHAMMAD FEZAN AFZAL, 3 SHAHBAZ AHMED
1,2 Research Scholar, Department of Computer Science and Software Engineering, International Islamic University Islamabad, PAKISTAN
3 Asst. Professor, Department of Computer Science and Software Engineering, International Islamic University Islamabad, PAKISTAN
E-mail: asad.msse330@iiu.edu.pk, ranafezan.afzal@gmail.com, shahbaz.ahmed@iiu.edu.pk

ABSTRACT

Development proof has been exposed that software developers present very good when they subsequent defined, reusable process such as the Personal Software Process (PSP). Unreliable and qualitative proofs from production specify that there two or more than two developers doing their work side-by-side at same system, work together on the one propose, code, test or algorithm, achieve significantly very well than the two developers perform alone i.e. couple software developing. The time and cost of Software development product are decreased in HPSP when evaluated with PSP software development. This review paper describe about PSP and its evaluation, cost and benefits of PSP in software development, B-PSP and comparative evaluation between automated PSP tools.


1. INTRODUCTION:

Before few years, companies have problem of refining the best quality of their processes which they are use, as a plan to stay secure and spirited [1]. Many organizations are work hard to automate, recover and get better the method they execute the company. In bad feeling of all these hard work, performance showed that large number of companies is contracting with software evolution still undeveloped [1]. The resulting of quality from industry sequence: enhancement in quality guide to lessening to modify, schedules and costs that guide to enhance potential, which guide to lesser cost and high marketplace share, that all guide to improved earnings and production permanence. Software development process enhancement is encouraged by this sequence response and focal point on apply disciplined software development processes [2].

In this review paper section 2 describes PSP and PSP evaluation, section 3 describes cost and benefits of PSP in software development, section 4 describes B-PSP: Adapting the Personal Software Process to B and section 5 describes comparative evaluation between automated tools of PSP and section 6 describes the conclusion.

2. WHAT IS THE PSP?

The PSP is a preparation activity. Basically this is designed for educate the basic principles to software engineers in software project and quality management [3]. The aim of Personal Software Process is to modify the action and educate to calculate software engineer’s work as well as educate them to evaluate and attain performance objectives [3].

PSP Evaluation:

2.1 PSP0:
PSP0 contains very basic process which consists of few essential evaluation techniques and exposure formats. The baseline presents a reliable foundation for calculate, development and identifies base on which progress attained [4].

2.2 PSP1:
PSP1 include some steps of planning in PSP0 like analysis statement, estimation of resources and size, planning of task schedule. It provides a framework to software developer to realize the relation between the size and time which is spent on the program which is to be developed [4].
2.3 PSP2:
PSP2 accomplish one of Personal Software Process objectives to aid software engineers study untimely to deal pragmatically and impartially with the program errors which consequence from engineers’ errors. PSP2 contain analysis methods to PSP1 help to find out the errors as soon as possible when they are low cost to correct them [4].

2.4 PSP3:
Repeated PSP3 process well balance up to huge programs support on consecutive augmentation of PSP2-sized portion divided always earlier. For evolution of a bigger program, engineers must use the concept standard alive in PSP3, as an alternative of PSP2 [4, 5].

3. COST AND BENEFIT FROM THE PSP:

3.1. Cost Model:
The objective of Benefit model is to find the cost for obtaining the Personal Software Process and measure with sensitive dimension component. Improvement of processes, this model consists of two costs measurement, preparation cost and accomplishment cost.

Preparation cost is the whole essential cost used up in refining software engineers. Accomplishment cost is the transparency cost which concur to the latest information and organization ability conveying by the Personal Software Process preparation and steady as familiar performance in future software evolution projects [5, 6].

Costs = Training Cost + Implementation Cost

The method of assessment the cost extremely depends on organization or company; hence the cost in the document is consideration as instance importance instead of financial importance. We can measure accomplishment cost by the prospect cost.

3.2. Benefit Model:
The objective of Benefit model is to discover the benefit features in accepting the Personal Software Process and enumerate them. The big advantage along with them is the Personal Software Process can facilitate software engineers to identify defects, eliminate and as soon as possible inserted them in their software evolution methods, and finally create a fault free manufactured goods with elevated production. There are three main attributes to measure such the benefit features are established as given below [6].

3.2.1 Productivity:
This is simply calculated by the idea of Lines of Code (LOC) as conventional method in software development. The Personal Software Process gives high-quality patterns to document and measure software size for a product in software evolution. Production in the Personal Software Process is present to the lines of code (LOC) in time spent in software evolution as a there is no errors in the project [5, 6].

Productivity = Lines of Code / Hour

3.2.2 Quality:
Low errors are inserted then the best quality is definite. Since identifying and eliminating errors ahead of the primary assemble and test is valuable and essential to decrease considerably the whole amount of possible errors, the excellence management in the Personal Software Process want to facilitate engineers identifies and eliminates a number of errors inserted as in the
initial time of software development. Mostly, quality is defined by errors concentration, means that the amount of errors inserted in one line of code (LOC) [5,6].

Quality=(1–Defect Density *100%  
Defect Density=Defects/Lines of Code

3.2.3 Size Estimate:  
How much defects in evaluation software product volume can be utilized as a pointer of budgeting in development (e.g. human resource allocation). The minor defect rate of size evaluation future to zero, the saving of cost is more expected in locating budget. The Personal Software Process present software engineers with valuable and efficient methods to evaluate the size of software (LOC) that are based on solid assumption [5, 6].

Defect Rate of Size Estimate=(LOC estimated/Real LOC developed)/Real LOC developed

If the defect rate becomes harmful, it means under evaluation but over assessment is advantage symbol. The defect evaluation zero means wonderful estimation but this is not possible, and the defect evaluation one means over assessment becomes two times higher than the original line of code (LOC) software evolution [5, 6].

4. B-PSP: ADAPTING THE PERSONAL SOFTWARE

Process to B:  
How obtain an objective of creating good quality, consistent software on given time and on particular resources? The prescribed technique recommended altering the conservative technique to software evolution to initiate proper requirement at very initial phase, after that source code is developed which is almost certainly accurate with respect to requirements [5]. The Personal Software Process technique prescribe with conservative maturity method, and as a substitute encourage the thought of personage software developers evaluating and reviewing their individual errors construction and on the whole production, excited to measure the plan of learning from experience [7]. It looks like levelheaded to us to attempt to attain more quality improvement. We consequently support the mixture of two techniques: the utilization of proper techniques joint with the quantity and evaluation order of Personal Software Process. Sake of concentration, we have learned to take focus on the B method. In contrast of several extra formal methods, the advantage of B-PSP is that it covers approximately every software life cycle, from requirement to code generation. We use the contraction B-PSP to prescribe our edition of the B technique to PSP [7, 8].

4.1. Focus of B-PSP  
Initially we focus our concentration on the requirement stage; there are a satisfactory amount of actions in this stage only to permit testing with PSP-based information gathering and evaluation. To decide which effects to evaluate to conduct a numeral of little sample evolution is totally based on exercises which are held in class rooms. After a large amount of cyclic iterations we are in a situation to suggest the B-PSP actions [8].

4.2. Data Collection in B-PSP:  
With PSP, in B evolution surroundings, the primary assignment of B-PSP is to count time as well as effort of development. The data for development which prefer to reproduces B MSS (Machine Schematic Structure) and so it is simple to gather. It was initial essential principle. In actual software evolution atmosphere tools and techniques to help in documentation; mostly the time and data size would be documented by incredible the B-Toolkit, since the arrangement of the calculated information reveal the arrangement of a B evolution. Such automatic classification tools and techniques are in boundary obstruction with a software developer’s flow of awareness. Initially we obtain “Time Recording Log” (TRL) for PSP and refine this into the “B-Time and Complexity Recording Log” and then continue to product reviewing for B-PSP. This deals with time and complexity in B evolution development situation [9].

4.3. Flexibility in “B-Time and Complexity Recording Log”  
The method undertakes through B evolution, we will document the evidences which are necessary to be release using the B axiom after primary four stages. This involves a major attempt on the software developer’s elements; hence we document the both quantity of evidences, as an irregular amount calculate, jointly the time use in this stage. Rather than start another feature for the number of evidences, that is stage-specific
5 PSP EVALUATION TOOLS

5.1 Sensor-based Scheduling:
A general saying that dropping to map is development to fail. For this cause, keeping a program schedule is critical to the achievement of the programs itself. It is like as it is in software evolution process, where programs frequently are unsuccessful as the programmers are capable to face neither what is at present going on nor what aim are for program evolution [11, 12]. Initially programmer accumulates their time limit for each assignment. This is all the time known as information. After that the Gantt chart facilitate to imagine the whole thing from information, which known as document [13, 14].

5.2 Visualization of Performance:
Visualization arrangement of data (performance and process) can be professionally apply with an interface mediator. As a center task, the connector mediator describes the engineer of software in PSP platform. The customer can perceive and observe the complete process and their presentation by using the connecter mediator and information based warehouse [15]. This moves toward automated assets more intellectual by combining them with mediator abilities. Therefore the proposal at the back of connector mediator is to make software systems same like a practical supporter. In several automated Personal Software Process tools, visualization is restricted to the manufacture of a design statement when the assignment is fulfilled [16].

Table 1 : Relative Evaluation between Automated Tools of PSP

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Agent</td>
<td>Not supported</td>
<td>No supported</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>Environment</td>
<td>Open source IDE</td>
<td>Just Windows</td>
<td>Unix Platform and Windows</td>
<td>Command line windows e.g. Linux, Unix etc</td>
<td>MS office, Eclipse</td>
<td>Eclipse, visual studio</td>
</tr>
<tr>
<td>Interface</td>
<td>Graphical User Interface, windows pop-up</td>
<td>easy and simple to learn</td>
<td>POP-up and Graphical User Interface</td>
<td>Graphical User Interface</td>
<td>Graphical User Interface</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>Defect Sharing</td>
<td>Yes (automatic)</td>
<td>No supported</td>
<td>Yes supported</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>Scripting</td>
<td>C language and java</td>
<td>No declared</td>
<td>PHP</td>
<td>Java</td>
<td>XML, Java</td>
<td>Java</td>
</tr>
<tr>
<td>Report</td>
<td>Diagram, PPS</td>
<td>Graphical evaluate statement</td>
<td>Diagram and information summary</td>
<td>PSP, chart and diagrams</td>
<td>Testimony description</td>
<td>Briefly describe and diagrams</td>
</tr>
<tr>
<td>LOC counter</td>
<td>Automatic (sensor-base)</td>
<td>No supported</td>
<td>Automatic (remains offline)</td>
<td>Automatic (sensor-based)</td>
<td>Automatic (sensor-based)</td>
<td>Tool container</td>
</tr>
<tr>
<td>Metaphor</td>
<td>Knowledge management</td>
<td>No declared</td>
<td>Only Web-base</td>
<td>No declared</td>
<td>No declared</td>
<td>Tool container</td>
</tr>
<tr>
<td>Planning Wizard</td>
<td>Time planning pattern</td>
<td>PSP pattern</td>
<td>Time planning pattern</td>
<td>Program plan pattern</td>
<td>Program plan pattern</td>
<td>Program Plane pattern</td>
</tr>
<tr>
<td>User Privacy</td>
<td>Yes (login and privacy task)</td>
<td>No supported</td>
<td>Yes (login)</td>
<td>Not Supported</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
<tr>
<td>User available</td>
<td>Programmer and program manager</td>
<td>Just programmer</td>
<td>Just programmer</td>
<td>Just programmer</td>
<td>Just programmer</td>
<td>Just programmer</td>
</tr>
</tbody>
</table>
6. CONCLUSION:

Starting from the engineering of software system side, quick and consistent evaluation and follow up the process will enhance an individual’s developer actions, from the program executive’s thinking; the information presented by PSP-evaluation will assist in observing and evaluating the presentation of developers as well as in conveying fresh projects to them. In short, this project will gives a sympathetic stage for both developers and program executive as in social gathering will obtain profit of integrating a software mediator into automatic PSP tools. PSP cost and benefits analysis for individual developer is very simple and easy task. By this method we easily find the individual effort and time which consuming in software development and all process are used.

References:


