

PSP 0.1 (English)



Español



Entries

- Summary Form of PSP0.1 Project Plan
- Time and default logs record
- Standard of types of defects
- Description of problem
- PIP Form
- Chronometer



Exit

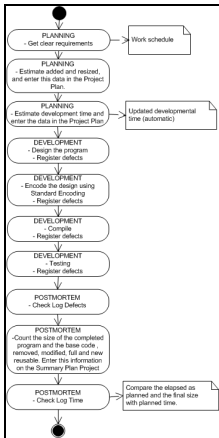
- A carefully tested program
- The Form Project Plan Summary completed with the estimated and actual data.
- The Logs Record Time and Defect completed
- The PIP complemented form



Solution



Process



Development time

- ◆ To acquire the necessary knowledge to develop the software product: 6 hours.
- ◆ To create the Product Pattern: 2 days.
- ◆ To apply the Product Pattern: 3 hours.



Explanatory Video

[Psp 0.1 Class](#)



Related Patterns

- [PSP 0 \(English\)](#)



Quality Controllers

- None



Templates



[Logs of PSP 0.1](#)



[Script Template PSP 0.1](#)



Examples

 Example of count for PSP 0.1 .

Support Tools

- Word processor [[Microsoft Word](#)], [[OpenOffice Writer](#)]
- Programming language [[Java](#)]
- Chronometer

Initial Context

This product is used when we want to have a precise and accurate estimate of the size of a program, incorporating discipline and measure your process. Forms are used to guide the process.

Result Context

The system user will get the statement of documented requirements, form Project Plan Summary completed with estimated time development data, the planned size, and registration time log and process improvement proposal form(PIP) completed. The% completed to a date it is added.

Problem

It desired to use an appropriate structure to carry out tasks of small scale and size measurement framework that occupy these tasks and the time spent for such tasks. In turn would be suitable to have a well-defined basis for improvement and process definition. Control and correction of defects is intended. Different parts within the software are determined.

Restricciones (*Forces*)

- **System Type:** It applies to all types of systems.
- **Characteristics of organizations:** This pattern can be used in existing projects in any company.
- **System Type to develop:** This product can be used in projects in which user requirements are changing. A Conceptual Design is necessary. Not useful for small programs.
- **Type of customer:** It must exist or be achieved, the target area development business being involved in achieving it.
- **Programming Paradigm:** OO (Object Oriented).

Roles

- Analyst
- Customer
- Project Manager
- System Users
- Developers

Lessons Learned

- Large programs or those that are not well understood may require an iterative approach. With small programs or well understood, you can run the steps in order.
- Using Scripts (scripts). Strive to use this paradigm until it becomes a habit.
- A good design will make the program pass the test phase more easily.

Capability Level

- Capability Level 5 CMMI..

Basic Knowledge and Skills

Knowledge

- Definition of software requirements (functional and non-functional).
- Knowing how to obtain data on how it really works.
- Accustomed to using a particular paradigm.
- Take registration experience as defects are found and corrected.

Abilities

- Capacity of Abstraction.
- Capacity of Analysis.

Information Resources

- Watts S. Humphrey. Introducción al proceso software personal. Addison Wesley. 2001.
 - Watts S. Humphrey. Introduction to the personal software process. Addison Wesley. 1997.
 - Watts S. Humphrey. PSP: a self-improvement process for software engineers. Addison Wesley. 2005.
-