

PSP 1 (English)



Español

Entries

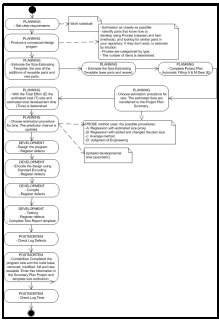
- Summary Form of PSP1 Project Plan
- Requirements and correct description of the problem
- Historical data of times and sizes (estimated and actual)
- Template Estimated Size
- Test Report Template
- Time and default logs record
- Standard of types of defects
- PIP Form
- Chronometer

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
- Estimating the size template completed.
- Test report template completed.

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 1 Class

Related Patterns

- [PSP 0 \(English\)](#)
- [PSP 0.1 \(English\)](#)
- [CRC Cards](#)
- [Sequence Diagram](#)

Quality Controllers

- None

Templates

 [Logs of PSP 1](#)  [Script Template PSP 1](#)



Examples



Ejemplo Plantilla Estimación Tamaño

Ejemplo estimación de tamaño.



Support Tools

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



Initial Context

This product is used when we want to have an accurate, precise and efficient estimation of the size and time, incorporating discipline and measure your process and serve to have a standard of comparison with real data so that in the end the best final results are generated. It help the measurement of time spent in each phase and defect detection. Forms are used to guide the process and a tool is used (PROBE) to make these estimations of time and size.



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 proposal better process form(PIP) completed. Keeps a running log of each of the tests are performed and the results of each of these tests (Test Report).



Problem

It is desired to use an appropriate structure to carry out tasks of small scale and size measurement framework that occupy these tasks and 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).



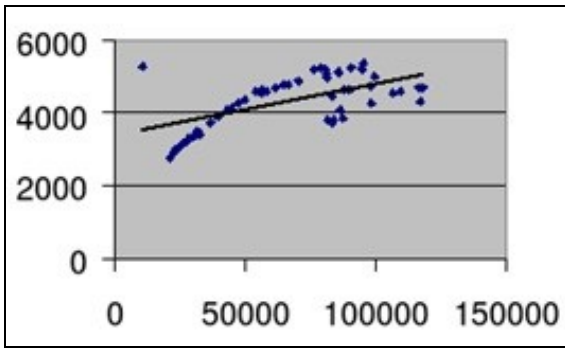
Roles

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



Lessons Learned

- A good design will make the program pass the test phase more easily.
- Using Scripts. Strive to use this paradigm until it becomes a habit.
- For the estimation of initial size and time is as nearly as possible you need a good conceptual design. To do this you must refine the conceptual design to identify parts that you are able to identify and develop parts that you know.
- Estimated size = item size * number of items.
- The Summary Plan Project PSP1 has a section for measuring productivity (number of units of size added and modified by time).
- Different size categories LOC (lines of code) are considered
 - ◆ Base: size of the original unmodified version of the product.
 - ◆ Added: writing code for new program or added to an existing one.
 - ◆ Modified: Modified code of an existing program.
 - ◆ Deleted: Code removed.
 - ◆ Reused: Code taken from a library.
- PROBE: Method used for estimating the size and time. Adjust the points by a regression line.



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.
-