

The Software Process - PSP Intro

Charles Carter

October 17, 2017

Table of Contents

1 PSP Estimation

2 PSP Defect Procedure

PSP Estimation

What is Estimation?

How do we determine how “big” a software project will be, i some sense of the word “big?”

Comparison

Estimation by comparing similar objects is a reasonable approach to judging size.

This requires data on the planned job as well as previous jobs

The object in comparison is to have something to compare to.

Comparison

Comparison requires data on previous jobs
Comparison requires data on the planned job.

Erge Jobs

Decomposition of large jobs, previous and planned

Generic Steps

- ① estimate size
- ② estimate time
- ③ obtain comparison data
- ④ add comparison data

Proxy Estimation

- ① proxy size must be relative
- ② proxy must be countable
- ③ proxy must be easy to visualize
- ④ proxy must be customizable
- ⑤ proxy must be sensitive to implementation details

Proxy Sizes

- very small
- small
- medium
- large
- very large

Proxy Types

- calculation
- data
- input/output
- logic
- setup/configuration
- Deployment
- text/string manipulation

analysis

compare estimated metrics to actual metrics
create data file of differences
calculate mean, standard deviation, etc.
error metric is standard deviation

Tradeoffs

Variance versus Precision

variance is the “spread” of data

precision is the amount of accuracy

many times, precision and variance are inversely proportional

PSP Steps

- ① complete a conceptual design, decompose parts
- ② estimate part size
- ③ estimate reused parts size
- ④ estimate project size
- ⑤ estimate time
- ⑥ calculate prediction interval

PSP Defects

Defect Type by Phase

- requirements
- design
- coding
- documentation
- defective fixes

Defect Type by Kind

Syntax Spelling, punctuation, typos, instruction formats

Syntax Declaration, duplicate names, scope, data range, initialization of data.

iAlgorithm Errors in algorithm design; logic, pointers, loops, recursion, computation

Interface Errors in module interface design: procedure calls and references, parameter lists.

Architecture Errors in architectural design; modularization, structure, coupling, cohesion.

Data Errors in data design: structure, content

Defect Type by Kind, Continued

Checking Failure to properly validate data values before used;
error messages, asserts.

Documentation Source code comments, messages. Also external
documentation.

Build Package change management, library, version control,
makefile error, etc.

Environment CASE tool, compiler, test, or other support system
problems.

system Hardware and platform configuration, real-time resources, shared
memory.

Defect Injection Phase

When was the original defect injected?
Why is this important?

Defect Discovery Phase

When was the original defect discovered?
Why is this important?

Defect Fix Time

How long did it take to fix the defect?
Why is this important?

Injection of Defects

Were additional defects injected by the fix?
Why is this important?

Injected Defect Discovery

When were the injected defects discovered?
Why is this important?

Injected Defect Fix Time

How long did it take to fix the injected defects?
Why is this important?

Defect Database

Keep a database of the metrics discussed above.

How would you analyze such data?

What benefits would result from the analysis of defects?