Com S 362
Object-Oriented Analysis & Design

Spring 2018 – Friday Apr 13, 2018

Simplified Function Point Counting

A'la Robertson & Robertson

and Capers Jones

FP Tables from "Mastering The Requirements Process, 3rd. Robertson and Robertson.
Power coefficients from: "Software Systems Failure and Success," Capers Jones, p. 172

Counting Function Points (not really)

Use your feature scenarios to guide the work.

Use the tables to look up function points based on the following counts:

For the responsibilities of your feature *count*:

- non-interactive input data (classes and attributes)
- output data (classes and attributes)
- interactive data (classes and attributes)
- internally stored data (classes and attributes)
- externally stored data (records/classes and fields/attributes)

based on Robinson and Robinson

Tables

Input Flow

Figure C.4

The function point counts for an *input* business use case. The correct function point counting terminology for these is *external inputs*.

Data attributes of input flow

Classes referenced	1-4	5-15	16+
<2	3	3	4
2	3	4	6
>2	4	6	6

Output Flow

Figure C.6

The function points for *output* business use cases.

Data Elements

Classes Referenced	1-5	6-19	20+
<2	4	4	5
2-3	4	5	7
>3	5	7	7

From; Robinson and Robinson

Tables

Inquires

Figure C.8

The function points for *inquiries* or time-triggered business use cases.

Classes Referenced	Data Elements 1-5	6-19	20+
1	3	3	4
2-3	3	4	6
>3	4	6	6

Internally Stored Data

Figure C.10

The function point counts for *internally stored data*, or *internal logical files* as they are known in function point terminology.

Record Elements	Attributes 1-19	20-50	51+
<2	7	7	10
2-5	7	10	15
>5	10	15	15

From; Robinson and Robinson

Tables

Externally Stored Data

Figure C.12

The function point counts for *external interface files*, or data stored held by a cooperative adjacent system.

Record Elements	Attributes: 1-19	20-50	51+
<2	5	5	7
2-5	5	7	10
>5	7	10	10

Allowing For Uncertainty

Figure C.13

This table, courtesy of Capers Jones, shows the uncertainty ranges based on the number of parameters you use in your function point counting. This information comes from the study of many software projects.

Number of parameters used	Range of uncertainty
1	+ or - 40%
2	+ or - 20%
3	+ or - 15%
4	+ or - 10%
5	+ or - 5%

From; Robinson and Robinson

Weight points for effort

- Effort in staff months =

  (function points/150) * function points^{SD}

SD is a coefficient from this table. You can use 0.43

Table 4-12. Nominal Power Level for Schedule Duration Applied to Function Point Totals

Project Class/Type	Best-in-Class	Average	Worst-in-Class
End-user software	0.25	0.33	0.37
Commercial software	0.39	0.41	0.45
Outsource/contract software	0.40	0.42	0.46
MIS software	0.41	0.43	0.47
Systems software	0.43	0.45	0.48
Military software	0.45	0.47	0.50

Power coefficients from: “Software Systems Failure and Success,“ Capers Jones, p. 172

Observations
On Structural Patterns

Adapter vs. Facade

- Facade
  - A class that creates a single interface which unifies, simplifies, or scopes access to existing class APIs.
    - Pretends to have logic that actually exists elsewhere.
- Adapter
  - A class that *converts* the interface of an existing class into the interface clients need, often by implementing modified or new functionality.
    - Reshapes existing logic to make it more palatable.

Adapter vs. Facade

- Facade
  - Typically little, if any, significant logic ... a “top 10” selection. The included methods already exist elsewhere.
  - May hide references to multiple subsystems
  - Sometimes used to define a boundary between layers.
- Adapter
  - Typically involves more logic than Facade.
  - Often necessary to make libraries (for which you have no source) fit better in your application.
  - In Java we would use multiple interfaces and composition instead of the multiple inheritance suggested in the text.

These Structural Patterns Can Change Existing Code!

- Facade
- Adapter
- Decorator
- Proxy
- Composite