Using COSMIC-FFP to Quantify Functional Reuse in Software Development

Vinh T. Ho, Alain Abran, Serge Oligny

Presented at

ESCOM-SCOPE 2000, Munich, Germany

April 18-20, 2000

Software Engineering Management Research Laboratory

Agenda

- Context
- Functional reuse
- Measuring functional reuse
- Examples
- Conclusion

Context

- **Goal: Using COSMIC-FFP, explore functional reuse measurement as a quantitative basis for evaluating alternatives designs**
- **Exploratory work**
- **Reuse mostly measured at code level**
- **Reuse at earlier stages is of interests**

UQÀM

Functional reuse

Kinds of reuse:

- Reuse without modification: black-box reuse
- Reuse with modification: white-box reuse

This exploratory research concentrates on black-box reuse.

UQÀM

Functional reuse

- __Functional reuse__: identifying “avoided” functions, that is functions which do not need to be re-developed.

- __Goal:__ How much such re-development could be avoided ?

Measuring functional reuse

- COSMIC-FFP: a functional size measurement method
- Candidates for reuse
- Quantifying functional reuse

Measuring functional reuse

COSMIC-FFP:

Principle

Software functional size is **directly proportional** to the number of its data-movement sub-processes.

Base Functional Components

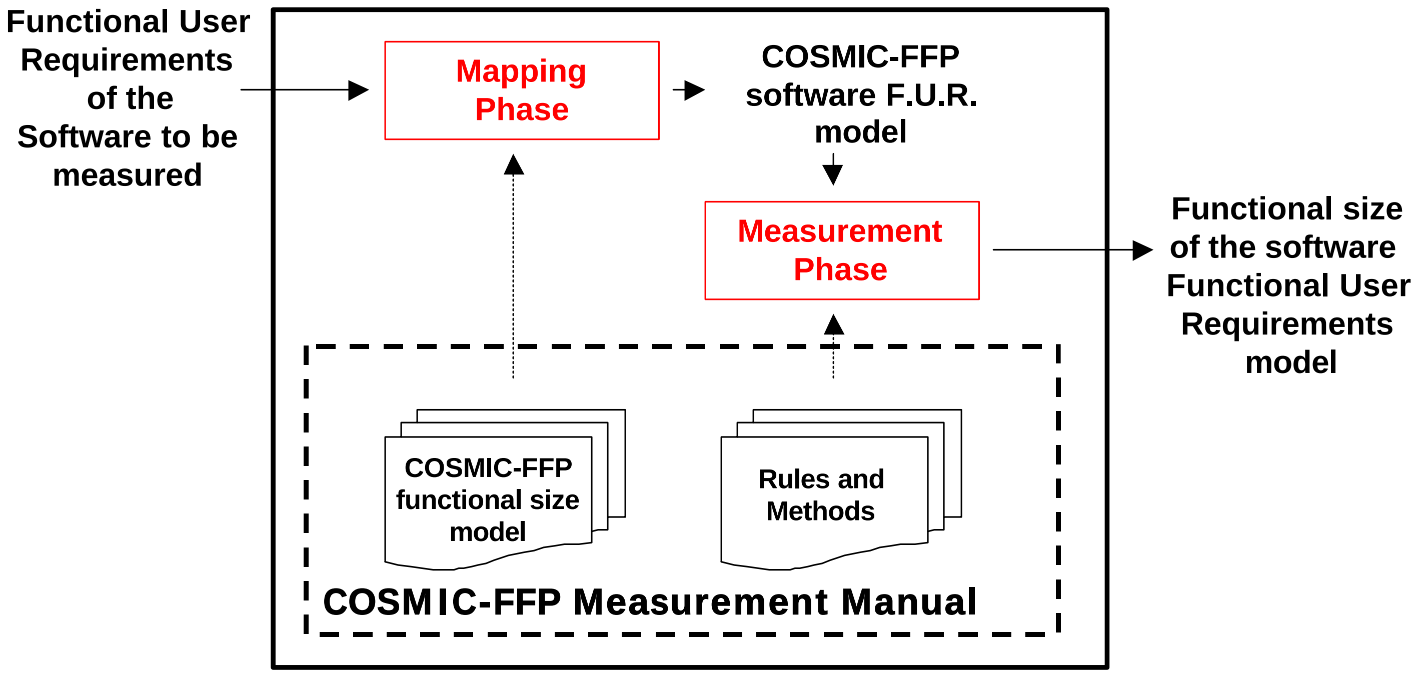
COSMIC-FFP recognize 4 types of data movements: **Entry**, **Exit**, **Reads** and **Writes**.

Unit

The standard unit of measurement (1 C_{fsu}) is defined by convention as equivalent to one single data movement at the sub-process level.

Measuring functional reuse

COSMIC-FFP:


Functional User Requirements of the Software to be measured
Mapping Phase
COSMIC-FFP software F.U.R. model
Measurement Phase
Functional size of the software Functional User Requirements model
COSMIC-FFP functional size model
Rules and Methods
COSMIC-FFP Measurement Manual

Measurement process

© 2000, V. T. Ho, A. Abran, S. Oligny
Software Engineering Management Research Laboratory

Measuring functional reuse

Candidates for reuse:

- ⊙ COSMIC-FFP layer differentiates F.U.R. allocated at different levels of functional abstraction
- ⊙ Each layer encapsulates functionality useful to other layers using its services
- ⊙ Identifying layers facilitate identification of reused functionality.

UQÀM

Measuring functional reuse

Quantifying functional reuse:

Principle

Amount of functional reuse is proportional to the product of the size of the functional processes reused and the number of functional processes using their services.

$$F_{RU} = \overset{n}{\underset{i = 1}{S}} \text{Size(Func. Process}_i\text{)} * U_i$$

Where: F_{RU} represents the total amount of functional reuse, in C_{fsu},

i represents the total number of reused functional process identified,

U_i represents the number of functional processes using functional process i

Measuring functional reuse

Quantifying functional reuse:

Reuse index The relative amount of reused functionality within a piece of software.

$$R_I = \frac{F_{RU} * 100}{\text{Size(Software)}}$$

Where: R_I represents the reuse index, in %,
F_{RU} represents the total amount of reuse within a piece of software, in C_{fsu}
Size(Software) represents the size, in C_{fsu}, of the piece of software without reuse

Examples

- ISDN Loopback tester, detailed example
- Summary of three other software

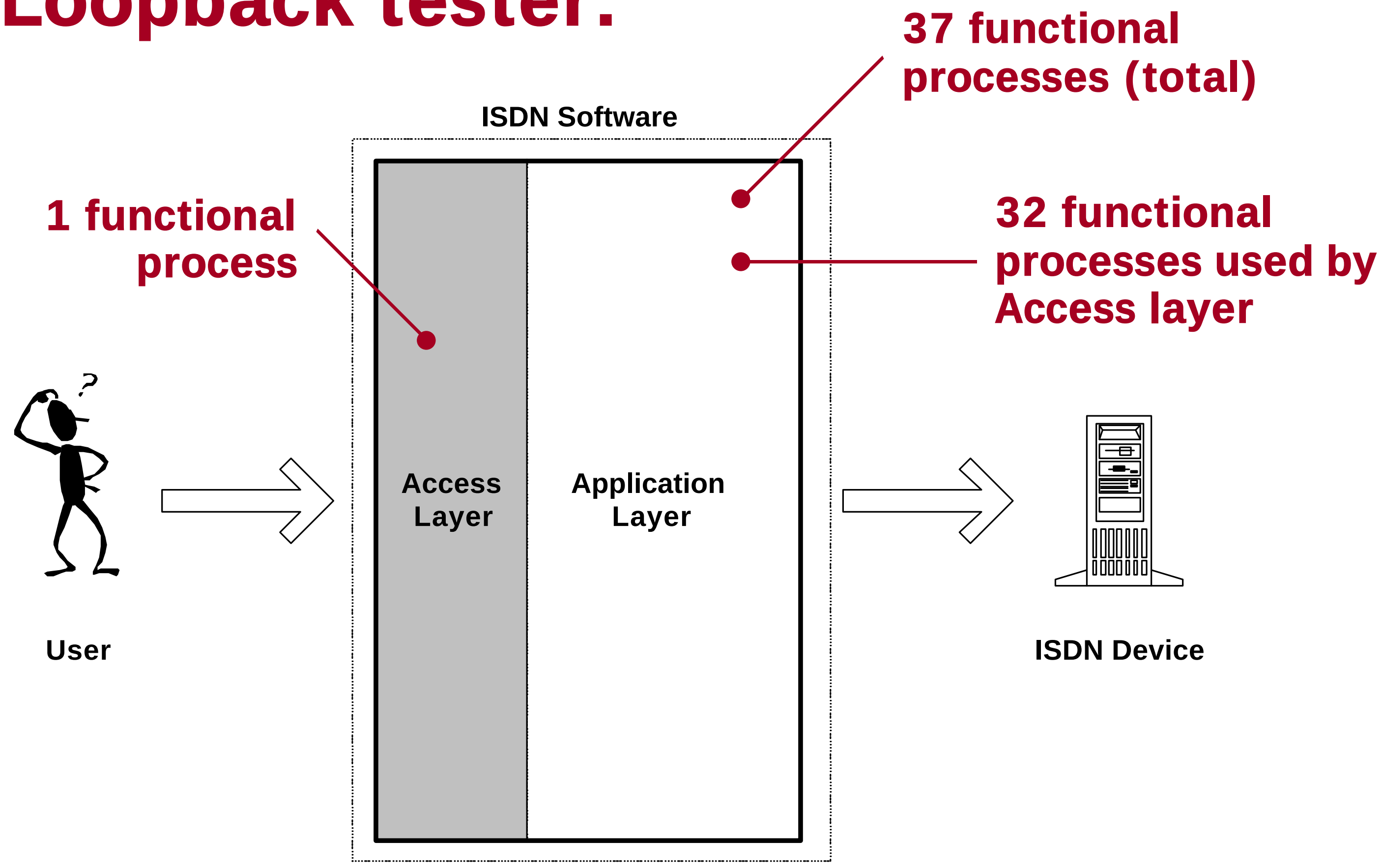
Examples

ISDN Loopback tester:

- Device used to test the integrity of four wires ISDN circuits at a remote location
- Users dial in to connect to ISDN device and performs testing and maintenance programming functions

Examples

ISDN Loopback tester:



37 functional processes (total)

1 functional process

32 functional processes used by Access layer

UQÀM

© 2000, V. T. Ho, A. Abran, S. Oligny
Software Engineering Management Research Laboratory

Examples

ISDN Loopback tester:

Scenario 1 - Assuming Access functionality is reused whenever possible

$i = 1$ (1 reused functional process - Access)

$U = 32$ (32 functional processes using Access)

Size (Func. process_1) = 10 C_{fsu}

$$F_{RU} = \sum_{i=1}^{1} 10\ C_{fsu} * 32$$

$$F_{RU} = 320\ C_{fsu}$$

Size of software with functional reuse: 136 C_{fsu}

Examples

ISDN Loopback tester:

Scenario 2 - Assuming Access functionality is not reused at all

Size of Access layer: 10 C_{fsu}

Size of Application layer: 126 C_{fsu}

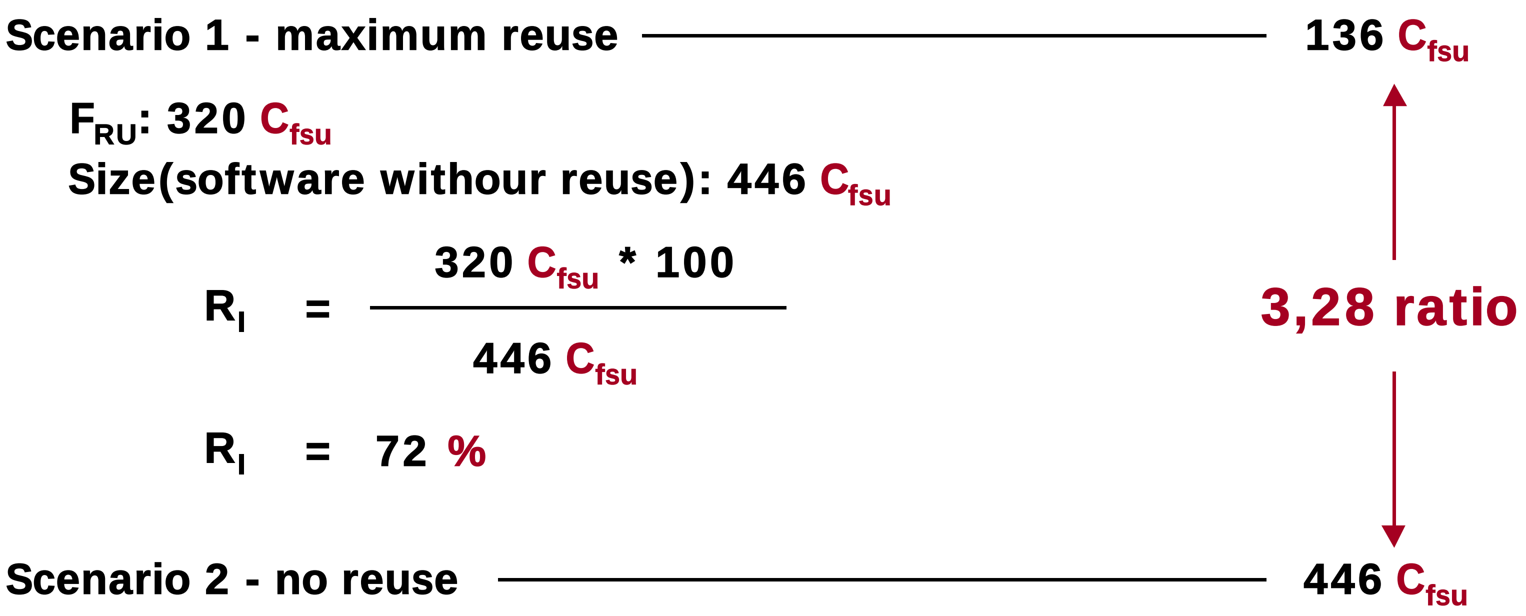
Size of duplicated access functionality (31* 10 C_{fsu}): 310 C_{fsu}

Total size of software: 446 C_{fsu}

Examples

ISDN Loopback tester:

Comparing scenario 1 and 2



Scenario 1 - maximum reuse —— 136 C_{fsu}

F_{RU}: 320 C_{fsu}

Size(software withour reuse): 446 C_{fsu}

$$R_I = \frac{320\ C_{fsu} * 100}{446\ C_{fsu}}$$

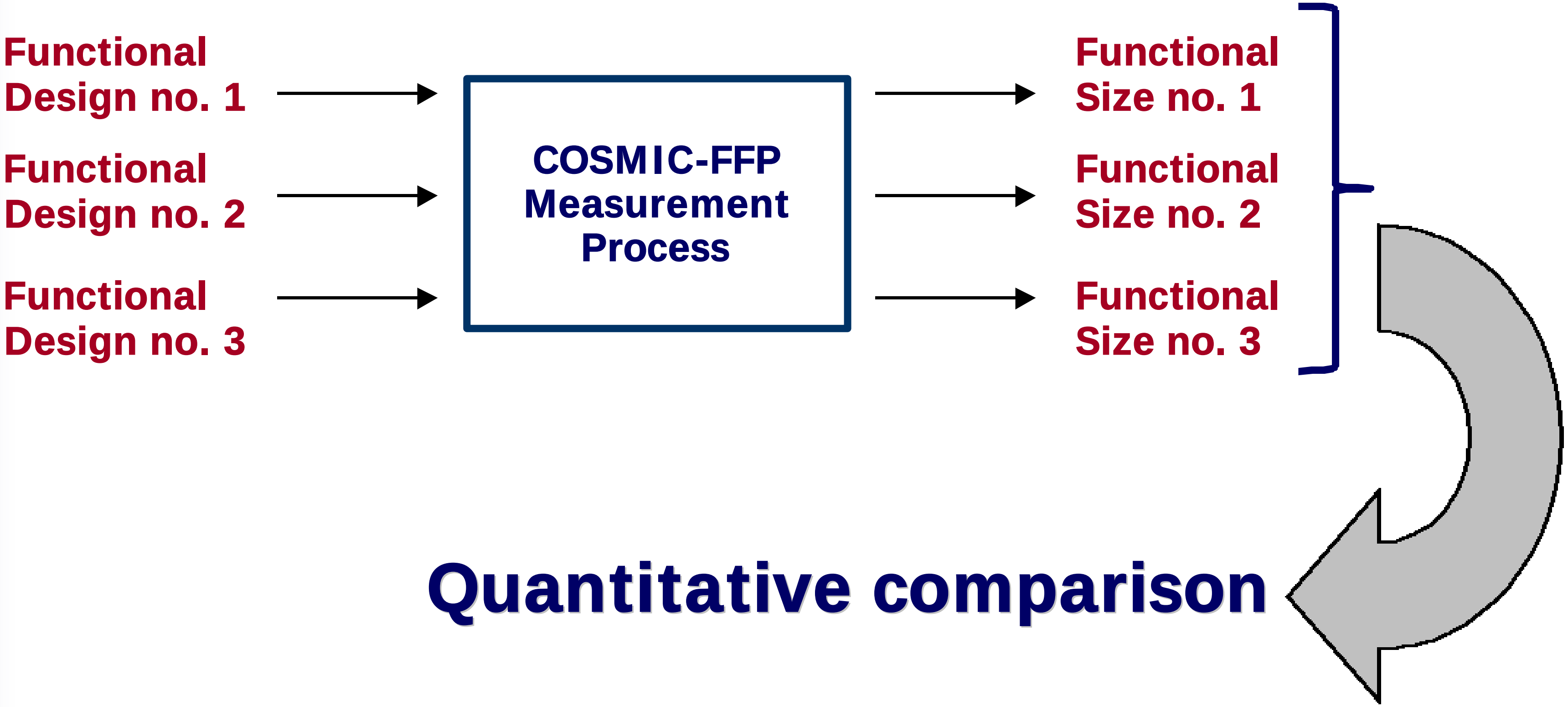
$$R_I = 72\ \%$$

3,28 ratio

Scenario 2 - no reuse —— 446 C_{fsu}

Examples

What have we done ?



Functional Design no. 1

Functional Design no. 2

Functional Design no. 3

COSMIC-FFP Measurement Process

Functional Size no. 1

Functional Size no. 2

Functional Size no. 3

Quantitative comparison

UQÀM

Examples

Three other software:

	Size(Fp_1)	U	F_{RU}	Size A[1]	Size B[2]	Ratio
Control software A	50 C_{fsu}	9	450 C_{fsu}	807 C_{fsu}	357 C_{fsu}	2,26
Control software B	25 C_{fsu}	8	200 C_{fsu}	359 C_{fsu}	159 C_{fsu}	2,26
Surveillance sub-system	29 C_{fsu}	3	87 C_{fsu}	131 C_{fsu}	44 C_{fsu}	2,98

1: size of software without functional reuse

2: size of software with complete functional reuse

UQÀM

Conclusion

- Important variance in size with and without functional reuse
  - ✓ must be taken into account in productivity and cost analysis
  - ✓ could be used as index to evaluate quality of design
  - ✓ could be used as index to evaluate integration of application portfolio

- Reuse must be recorded in data collection for benchmarking

Conclusion

- COSMIC-FFP layer concept useful to identify functional reuse
- Measuring reuse provides a quantitative basis for evaluating alternatives designs
- Enlarge scope from black-box/complete reuse to white-box/partial reuse

Further readings

- Bootsma F., “*Applying Full Function Points to drive strategic business improvement within the real-time software environment*”, Annual IFPUG Conference, New-Orleans, 1999
- Abran A., Desharnais J.M. “*Measurement of functional reuse in Maintenance*”, Software Maintenance: Research and Practice, 1995

Downloadable at www.lrgl.uqam.ca/ffp.html

Acknowledgments

- The Software Engineering Management Research Laboratory of the Université du Québec à Montréal is supported through a partnership with Bell Canada.

- Additional funding is provided by the Government of Canada.