A new method for measuring software functional size

Luigi Buglione, Ph.D.
Process Improvement & Measurement Specialist
Industry Business Unit
Engineering.IT

www.eng.it
The first Italian ICT player
- more than 730 M/€ revenues
- 1000 clients
- 6,300 IT specialists
**COSMIC**

Goals of the presentation

- **G1.** Present the COSMIC FSM method and principles
- **G2.** Discuss main changes/improvements against the so-called 1°-generation FSM methods
- **G3.** Show possible scenarios for applicability of COSMIC Function Points (CFP) in the Automotive sector
COSMIC

**Introduction**
- How requirements can be perceived
- A bit of humour...

**Functional Size Measurement (FSM) Methods**
- History: the first 30 years
- The ‘productivity paradox’
- 1st generation FSM methods

**COSMIC**
- Origins & Evolution
- Reference documents
- Scope of Application
- General Concepts
- Counting Principles
- Case Study: the Rice Cooker
- Benchmarking data
- Conversions with other FSMM

**Conclusions & Prospects**

**Q & A**
Top Metrics for SPICE-compliant projects

Luigi Buglione
Engineering.it
luigi.buglione@eng.it
One year ago...

Possible solution

Determine the possible best combination of measures

Project

Resources

Product

Processes

COSMIC
**Possible Solution**

2. Determine the list of most representative measures (1)

<table>
<thead>
<tr>
<th>Entity</th>
<th>Attribute</th>
<th>Measure</th>
<th>Threshold</th>
<th>A-SPICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>Planning compliance</td>
<td>Effort (man/hrs) per SLC phase, per iteration (abs, %)</td>
<td>(profiles on hist.data)</td>
<td>MAN.3</td>
</tr>
<tr>
<td>Resource</td>
<td>Time</td>
<td>% of open complaints / notes for delaying in providing the agreed furnished (tracked) per contract</td>
<td>≤10%</td>
<td>ACQ.4</td>
</tr>
<tr>
<td>Process*</td>
<td>Time performance</td>
<td>SPI (Schedule Performance Index)</td>
<td>ongoing</td>
<td>MAN.3</td>
</tr>
<tr>
<td>Process*</td>
<td>Cost performance</td>
<td>CPI (Cost Performance Index)</td>
<td>ongoing</td>
<td>MAN.3</td>
</tr>
<tr>
<td>Process</td>
<td>QA performance</td>
<td>% of non-conformances still open</td>
<td>≤15%</td>
<td>SUP.1</td>
</tr>
<tr>
<td>Process*</td>
<td>Maturity</td>
<td>Problem Reports (PR) by status (open, closed)</td>
<td>(profiles on hist.data)</td>
<td>SUP.9</td>
</tr>
<tr>
<td>Process</td>
<td>Changeability</td>
<td>Avg Change Requests (CR) working time by status</td>
<td>(profiles on hist.data)</td>
<td>SUP.8 - SUP.10</td>
</tr>
<tr>
<td>Process*</td>
<td>Planning reliability</td>
<td>Requirements Volatility of 'Scope Creep' Index (# of modified/new UR not formally traced / tct. # UR) by iteration</td>
<td>≤10%</td>
<td>ENG.4</td>
</tr>
<tr>
<td>Product*</td>
<td>Code Length</td>
<td>Kilo Lines of Code (kLOC) [system, function, module] c/a $functions per module</td>
<td>(abs, 100-150, 700-1000)</td>
<td>ENG.4</td>
</tr>
<tr>
<td>Product*</td>
<td>Functional Size</td>
<td>Functional Size (fsu) [system]</td>
<td>(abs)</td>
<td>ENG.4</td>
</tr>
</tbody>
</table>
Introduction

How requirements can be perceived...

- How the customer explained it
- How the project leader understood it
- How the engineer designed it
- How the programmer wrote it
- How the sales executive described it
- How the project was documented
- What operations installed
- How the customer was billed
- How the helpdesk supported it
- What the customer really needed
Introduction

Dilbert on Requirements...

I’LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.

FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?

I’M TRYING TO MAKE YOU DESIGN MY SOFTWARE.

I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?

I WON’T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.

TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!

CAN YOU DESIGN IT TO TELL ME MY REQUIREMENTS?

© Scott Adams, Inc./Dist. by UFS, Inc.
FSM Methods

History: First 30 years

- **IFPUG FPA** (ISO/IEC 20926:2009): v4.3 (2009), it explicitly excludes VAF
- **NESMA FPA** (ISO/IEC 24570:2005): update of the Dutch v2.0 (1998) up to v2.1, mostly devoted to enhancement projects
- **FISMA FPA** (ISO/IEC 29881:2008): the v1.1 Finnish method, including some BFC (Base Functional Components) different than other FSM methods

Each method has its own **CPM** (*Counting Practice Manual*) or **MM** (*Measurement Manual*) with details about “how” counting the *points*. 
## FSM Methods

The ‘productivity paradox’: LOC

<table>
<thead>
<tr>
<th>Source code size</th>
<th>Assembler version</th>
<th>Ada version</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source code size</td>
<td>100.000</td>
<td>25.000</td>
<td>-75.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity in PM:</th>
<th>Assembler version</th>
<th>Ada version</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>• requirements</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>• design</td>
<td>25</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>• coding</td>
<td>100</td>
<td>20</td>
<td>-80</td>
</tr>
<tr>
<td>• documentation</td>
<td>15</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>• integration &amp; test</td>
<td>25</td>
<td>15</td>
<td>-10</td>
</tr>
<tr>
<td>• management</td>
<td>25</td>
<td>15</td>
<td>-10</td>
</tr>
<tr>
<td><strong>Total Effort</strong></td>
<td><strong>200</strong></td>
<td><strong>100</strong></td>
<td><strong>-100</strong></td>
</tr>
</tbody>
</table>

| Total Cost       | $1.000.000        | $500.000    | -$500.000  |

| Cost per line    | $10               | $20         | +10$       |
| Lines per month  | 500               | 250         | -250       |

### FSM Methods

#### The ‘productivity paradox’: FP

<table>
<thead>
<tr>
<th></th>
<th>Assembler version</th>
<th>Ada version</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>300</td>
<td>300</td>
<td>0</td>
</tr>
<tr>
<td>Activity in PM:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• requirements</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>• design</td>
<td>25</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>• coding</td>
<td>100</td>
<td>20</td>
<td>-80</td>
</tr>
<tr>
<td>• documentation</td>
<td>15</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>• integration &amp; test</td>
<td>25</td>
<td>15</td>
<td>-10</td>
</tr>
<tr>
<td>• management</td>
<td>25</td>
<td>15</td>
<td>-10</td>
</tr>
<tr>
<td><strong>Total Effort</strong></td>
<td>200</td>
<td>100</td>
<td>-100</td>
</tr>
<tr>
<td><strong>Total Cost</strong></td>
<td>$1,000,000</td>
<td>$500,000</td>
<td>-$500,000</td>
</tr>
<tr>
<td>Cost per FP</td>
<td>$3.333</td>
<td>$1.666</td>
<td>-$1.667</td>
</tr>
<tr>
<td>FP per month</td>
<td>1.5</td>
<td>3.0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

**Source:** C. Jones, *What are Function Points?*, SPR website, URL: http://web.archive.org/web/19990421055424/www.spr.com/library/0funcmet.htm
The International association managing since 1986 updates of Albrecht's FPA counting rules is the International Function Point Users Group (IFPUG), composed worldwide from 700+ members – www.ifpug.org

Some active SMAs:

- GUFPI-ISMA (Gruppo Utenti Function Point Italia – Italian Software Metrics Association)
- UKSMA (UK Software Metrics Association)
- NESMA (Netherlands Software Metrics Association)
- FISMA (Finnish Software Measurement Association)
- DASMA (Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandschätzung)
- QESP (Quantitative Enterprise Software Performance, ex ASMA)

• ...
• **Which ones?**
  - FSM methods having in their counting scope the solely application layer

• **Limits and Scope of Applicability**
  - **Application domains:** IFPUG FPA not particularly feasible to non-MIS software (e.g. real-time apps, embedded software, etc.)
  - **Scope:** Software system seen only at the *application layer* level
  - **Viewpoint:** The *end user* one
  - **Weighting system:** 1°gen- FSM methods use a weighting system for their BFCs, based on a project sample (typically not so huge)
  - **Possible consequences:** not performant usage of fsu for building effort and cost estimation models

**Possible solutions**
- Create a new FSMM, aligned with ISO/IEC 14143-x series rules and criteria, with the goal to overcome the above mentioned limits
Origins & Evolutions

- **Origins**
  
  ✓ **Main Evolutions:**
  - Full Function Points (FFP, UQAM, 1997) → v1.0
  - COSMIC-FFP (1999) → v2.0
  - COSMIC (2007) → v3.0

  ✓ **COSMICON**: Common Software Measurement Initiative Consortium
  - Voluntary initiative born in 1998
  - Documents and rules always in the publicly available
  - COSMIC (v2.2) become an ISO standard yet in 2003 (ISO/IEC 19761)
  - Currently it has Board Members from Europe, North America, Asia and Australia
  - URL: [www.cosmicon.com](http://www.cosmicon.com)
COSMIC

Evolutions: FFP (1997)

- **v1.0 – FPA Extension for R/T software**
  - Modify IFPUG counting rules, distinguishing:
    - Management Types → to be counted according IFPUG rules
    - Control Types → to be counted according new FFP rules and to be added to the other points
  - **BFC:**
    - Control Data Types (**UCG** – Updated Control Group; **RCG** – Read-only Control Group)
    - Control Transactional Data Types (ECE – External Control Entry, **ECX** – External Control eXit, **ICR** – Internal Control Read, **ICW** – Internal Control Write)
  - **Weighting System:**
    - A weighting system was in place, as for IFPUG FPA, taking care of DET but not RET/FTR

COSMIC

Evolutions: COSMIC-FFP (1999-2001)

- v2.0 – method ‘per-se’
  - Main changes from FFP v1.0:
    - No more differences between Management and Control Types, → ‘data movement’
    - Two macro-phases: Mapping and Measurement
    - Introduced the ‘layer’ and ‘viewpoint concepts’
    - Convertibility from/to other FSMM (included FFP v1.0 and IFPUG FPA v4.1)

- BFC:
  - Possible ‘data movements’: E – Entry, X - eXit, R – Read, W –Write

- Weighting System:
  - Deleted the weighting system
  - Each ‘movement’ counts 1 cfsu (COSMIC functional size unit)

COSMIC

Evolutions: COSMIC (2007)

v3.0 – Refining the method

- Main changes from COSMIC-FFP:
  - Three macro-phases: Measurement Strategy, Mapping and Measurement
  - Granularity level in the counting
  - Clear distinction between 'principles' and 'rules'
  - Maggiore e migliore strutturazione del dataset di documenti e guide
  - Series of Guidelines (GL) per application domain
  - Possibility of Local extensions

- BFC:
  - E – Entry, X - eXit, R – Read, W –Write

- Size Unit:
  - Change of name from cfsu to CFP (COSMIC Function Point)

COSMIC

Scope of Application

• Applicability
  ✔ COSMIC can be adopted for calculating the functional size of a software
  ✔ It is valid both for the “business” than the “real-time” domains
    ❖ Nei sistemi business si considerano in più quegli aspetti
    ❖ Nel real-time si catturano quindi anche tutti gli eventi di controllo/attivazione
  ✔ Available guidelines (GL) for sizing...
    ❖ Business Application Software (BAS), v1.1 (2008)
    ❖ Datawarehouse (2009)
    ❖ Real-time (2010)
    ❖ SOA (2010)

• Non-Applicability
  ✔ COSMIC is not viable for other types of software...
    ❖ ...mathematical-algorithmic
    ❖ ...managing business rules
    ❖ ...expert and forecasting systems (e.g.weather report systems)
    ❖ ...computer games
COSMIC is a 2\textsuperscript{nd} generation FSM method

- 4 BFC, no weighting system, each data movement counts as 1 CFP
- 3 macro-phases: Measurement Strategy, Mapping and Measurement
- Software Models:
  - Software Context Model $\rightarrow$ representation of the software system to measure
  - Generic Software Model $\rightarrow$ generic representation with the 4 data movement (E, X, R, W)
- User: not only end-users, but also different viewpoints (with different views on software)
- Layer: porzione risultante dalla divisione funzionale dell’architettura software (hw/sw)
- Granularity level: each functional process can be decomposed into sub-processes
COSMIC

General Concepts: Layers

Software Layers

- Application Layer
  - App 1
  - App 2
  - App 'n'
  - Middleware Layer (Utilities, etc)
- Database Management System Layer
  - DBMS 1
  - DBMS 2
- Operating System Layer
  - Keyboard Driver
  - Screen Driver
  - Print Driver
  - Disk Driver
  - Keyboard
  - VDU Screen
  - Printer
  - Hard Disk Drive
  - Central Processor

Hardware

- Keyboard
- Screen
- Print
- Hard Disk
- Central Processor

Architettura sistema business/MIS

Architettura sistema R/T embedded

Embedded Application Layer

Operating System Layer

- Sensor Driver
- CV Driver
- Display Driver
- Mem. Chip Driver
- Sensor(s)
- Control Valve(s)
- Display
- Memory Chip
- Central Processor

Key:
- Superior Layer
- Subordinate Layer

www.eng.it
COSMIC

General Concepts: Level of Granularity

Level 1

Level 2

Level 3

Level 4
COSMIC

Counting Principles

- Counting principles
  - Each data movement (E, X, R, W) creates 1 CFP
  - Functional size of a functional process = sum of frequencies of its BFCs
  - COSMIC Function Points (CFP) own the additive property
    - Functional size of a software = sum of the size of its functional processes derived from FURs, according to what established in the Measurement Strategy phase
    - Functional size of a modified software system = sum of the sizes for added, changed and deleted data movements

- Reporting of Results
<table>
<thead>
<tr>
<th>No.</th>
<th>Funct. Process</th>
<th>Triggering Event</th>
<th>Data Movement Description</th>
<th>Data Group</th>
<th>DM Type</th>
<th>CFP</th>
<th>CFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select target temperature</td>
<td>30sec. Timer Signal</td>
<td>Receive Triggering event Read Cooking mode RAM Receive Elapsed Time Read Temperature relationship from ROM Write target temperature</td>
<td>30sec. Timer Signal Cooking mode Elapsed time Signal Temperature relationships Target temperature</td>
<td>E</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Control Heater</td>
<td>5sec. Timer Signal</td>
<td>Receive Triggering event Receive Actual Temp. from Sensor Read Target temperature Fissa il comando Calore (On/Off)</td>
<td>5sec. Timer Signal Actual Temperature Target Temperature Heater Turn On/off command</td>
<td>E</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Control Cooking Lamp</td>
<td>Elapsed Time Signal</td>
<td>Receive Triggering event Send Cooking Lamp Command</td>
<td>Elapsed time Signal Cooking lamp turn on/off command</td>
<td>E</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Tot 11
## COSMIC Benchmarking Initiative

- **Proposal COSMIC w/ISBSG (2006-)**
  - **Goal:** spreading the number of project data sized with COSMIC in the current ISBSG repository
  - **News:**
    - Simplified questionnaire than current standard version
    - Add the number of layers taken into account

## ISBSG r11 (May 2009)

- 5052 projects, measured with different methods:
  - **345** projects measured with COSMIC (+203% than in r10)

### Table 1: COSMIC Sizing Results

<table>
<thead>
<tr>
<th></th>
<th>CFP</th>
<th>Eff m/d</th>
<th>Prod</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max</strong></td>
<td>2003</td>
<td>5936.6</td>
<td>5.48</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>170.1</td>
<td>548.9</td>
<td>0.72</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>95.0</td>
<td>319.3</td>
<td>0.30</td>
</tr>
<tr>
<td><strong>Min</strong></td>
<td>3</td>
<td>2</td>
<td>0.03</td>
</tr>
</tbody>
</table>

### Table 2: NewDev Sizing Results

<table>
<thead>
<tr>
<th></th>
<th>CFP</th>
<th>Eff m/d</th>
<th>Prod</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max</strong></td>
<td>2090</td>
<td>5907</td>
<td>40.87</td>
</tr>
<tr>
<td><strong>Avg</strong></td>
<td>309.2</td>
<td>716.6</td>
<td>1.54</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>182.0</td>
<td>350.7</td>
<td>0.45</td>
</tr>
<tr>
<td><strong>Min</strong></td>
<td>8</td>
<td>1</td>
<td>0.02</td>
</tr>
</tbody>
</table>
COSMIC

Conversions with other FSMM

• Motivations for conversions
  ✓ “Save” the value from application portfolio counts yet done with another Fsm method (e.g. IFPUG)

• Suggestions / Studies
  ✓ COSMIC “Advanced & Related Topics” v3.0, Chapter 3
    ▶ Multiple 'native' count against \( n \) projects
    ▶ Derive regression straight line/ curve from projects historical functional size data
    ▶ Avoid to apply ‘as-is’ formulas

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Data points</th>
<th>FP range</th>
<th>Formula</th>
<th>( R^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fetcke (1999)</td>
<td>4</td>
<td>40-77</td>
<td>( \text{CFP} = 1.1\times\text{FP}_{\text{IFPUG}} - 7.6 )</td>
<td>0.97</td>
</tr>
<tr>
<td>Vogelezang &amp; Lesterhuis (2003)</td>
<td>11</td>
<td>39-1424</td>
<td>( \text{CFP} = 1.2\times\text{FP}_{\text{NESMA}} - 87 )</td>
<td>0.99</td>
</tr>
<tr>
<td>Abran, Desharnais, Azziz (2005)</td>
<td>6</td>
<td>103-1146</td>
<td>( \text{CFP} = 0.84\times\text{FP}_{\text{IFPUG}} + 18 )</td>
<td>0.91</td>
</tr>
<tr>
<td>Desharnais &amp; Abran (2006)</td>
<td>14</td>
<td>111-647</td>
<td>( \text{CFP} = 1.0\times\text{FP}_{\text{IFPUG}} - 3 )</td>
<td>0.93</td>
</tr>
<tr>
<td>Van Heeringen (2007)</td>
<td>26</td>
<td>61-1422</td>
<td>( \text{CFP} = 1.22\times\text{FP}_{\text{NESMA}} - 64 )</td>
<td>0.97</td>
</tr>
</tbody>
</table>

• Counting exercises with multiple FSM methods
COSMIC

Renault France

✓ Object of interest: ECU (Electronic Control Units)
✓ Effort Estimations for sw development done in the past using IFPUG FPA method and COCOMO technique
✓ COCOMO was unsuccessful after a first experimentation
✓ IFPUG and COSMIC experimentation started in mid 2008
✓ ECU with a set of specifications under Matlab/Simulink tools with also textual requirements
✓ 9 modules under the experimentation, sized with both methods (IFPUG and COSMIC)

First results

✓ IFPUG UFP values always higher than COSMIC CFP
✓ COSMIC seemed to have a best fit with embedded software projects
✓ Easier counting rules with COSMIC
✓ As a consequence – COSMIC used for a larger experimentation for embedded software projects
✓ The BCM (Body Control Module), with Statemate tool + textual requirements

Four goals

✓ To have indicators for managing suppliers productivity during time
✓ To predict software function development costs for better negotiating with suppliers
✓ To be able to estimate a function software development cost as soon as specifications are written and choose if implement or not such function
✓ To benchmark productivity levels among different suppliers

Source: S.Stern, Practical Experimentation with the COSMIC method in Automotive Embedded Software Field, in IWSM-Mensura 2009 Proceedings, Amsterdam (Netherlands), November 2009
COSMIC

A case study in the Automotive Sector

- BCM: Functions evolutions, manual coding (raw model)
  \( R^2 = 0.7376 \)
  \( n=22 \)

- BCM: New developments, manual coding
  \( R^2 = 0.9311 \)
  \( n=6 \)

Source: S. Stern, *Practical Experimentation with the COSMIC method in Automotive Embedded Software Field*, in IWSM-Mensura 2009 Proceedings, Amsterdam (Netherlands), November 2009
COSMIC

A case study in the Automotive Sector

Source: S. Stern, *Practical Experimentation with the COSMIC method in Automotive Embedded Software Field*, in IWSM-Mensura 2009 Proceedings, Amsterdam (Netherlands), November 2009
COSMIC

Conclusions & Perspectives

- **FSM Methods**
  - Born with the goal to provide more objectivity in sizing FUR for a software system
  - The IFPUG method moved from the initial Albrecht’s study and is nowadays the most diffused one after 30+ years (from 1979, when it was firstly presented)
  - 2 generations of FSMM (with/without weighting systems; single-multi layer, etc.)
  - Nowadays 5 FSMM are ISO/IEC standards, based on ISO 14143-x family
  - Intrinsic characteristic: fsu measure the software product, not its project and only for its functional side

- **COSMIC**
  - Method born in 1997 with the goal to complement the IFPUG one for real-time projects, become a per-se method during the years, as a further option against other FSMM
  - No weighting system, it counts only the frequencies of the 4 data movements
  - Good approximation when converting with other FSMM
  - Worldwide diffusion on-going

- **Sizing & Estimation issues**
  - Each FSMM measures the *product*, not the *project* → assumptions in estimates about NFR
  - Complement the non-functional side for deployment
  - Make distinction between 'nominal' and 'functional' productivity
  - Cost/fsu: deal it with great care! → need for a clear definition of cost elements and its nature
  - Continual (technical) benchmarking, based on BFC
Thanks for your attention!
Grazie per la vostra attenzione!
Misurare il software

Quantità, qualità, standard e miglioramento di processo nell’Information & Communication Technology

Franco Angeli, 2008 – 3ª edizione
Collana: Informatica ed Organizzazioni
pp. 380 -Volume 724.20

Luigi Buglione

www.semq.eu/leng/booksms.htm

Parte dei proventi sono devoluti alla FISM (Fondazione Italiana Sclerosi Multipla)
Thanks for your Attention!

We care of your problems and we have in mind a solution

Misurare per migliorarsi