THE VALUE OF COSMIC SOFTWARE SIZE MEASUREMENT FOR PROJECT ESTIMATING AND CONTROL

Polish Software Measurement Association Conference
June 2016, Warsaw

Charles Symons
Objectives of software size measurement:

‘Functional Size Measurement’ (FSM) methods

- The COSMIC method - key features
- Evidence that the COSMIC method has achieved its goals
- Conclusions
Objective 1: Software managers want to control and compare performance of project activities

**Delivery to budget & time:**
- Actual vs. Estimated Cost
- Actual vs. Estimated Duration

**Project speed**
- **Size / Duration**

**Project productivity**
- **Size / Effort**

**Product scope and quality**
- **Size**
- Functional (e.g. business needs)
- Technical (e.g. maintainability, post-delivery defects density)

Also consider the performance of on-going maintenance and enhancement activities

Copyright: COSMIC 2017. All rights reserved
Objective 2: ....and to use performance data for estimating future projects

- Establish benchmarks
- Measure and track software requirements size
- Measure project performance
- Estimating budgeting, etc.

Copyright: COSMIC 2017
A simple approach to effort estimation - for a project, iteration, etc

Productivity = \frac{\text{Software size}}{\text{Project effort}} \quad (Establish \ benchmarks \ from \ data \ for \ several \ projects)

New project ‘typical’ effort = \frac{\text{Estimated software size}}{\text{Benchmark project productivity}}

New project estimated effort = \left\{ \frac{\text{Estimated software size}}{\text{Benchmark project productivity}} \right\} \times \left\{ \text{Adjustments for project-specific factors} \right\}

Copyright: COSMIC 2017. All rights reserved
COSMIC (α IFPUG) define three types of requirements for a software project, or iteration, etc.

**Functional** User Requirements (FUR)
- what the software must do

**Non-Functional** Requirements (NFR)
- quality, technical and environmental constraints, etc.

**Project** Requirements and Constraints (PRC)
- targets, processes & tools, resources, dependencies, etc.
## Functional Size Measurement (FSM) methods can help achieve both objectives

### Sizing method options:

**Counts of lines of code:**
- X Cannot be estimated until software designed
- X Technology-dependent, no standards
- ✓ Accounts for all requirements that are delivered

**Other sizing methods:**
- X No reliable standards; only local benchmark data possible
- X What about ‘Non-Functional’ Requirements?
- X Early total effort estimation?

**Functional size:**
- ✓ International standard methods
- ✓ Technology-independent
- ▪ What about ‘Non-Functional’ Requirements?

---

Copyright: COSMIC 2017. All rights reserved
However, there are several methods of Functional Size Measurement.
‘First Generation’ FSM Methods share several weaknesses

- Designed to measure ‘whole’ business applications
- Difficult to use with modern software requirements engineering and development methods (e.g. Agile)
- Sizes
  - No well-defined unit of measure
  - Mostly limited, closed size ranges
Traditional FP vs. COSMIC FP measurement scale - a key difference

Function Points (FP) vs. COSMIC Function Points - CFP

- No arbitrary max
- A single CFP exists and is well defined

Copyright: COSMIC 2017. All rights reserved
Agenda

- Objectives of software size measurement: FSM methods
- The COSMIC method - key features
  - Evidence that the COSMIC method has achieved its goals
- Conclusions
Goals of the COSMIC method for sizing software

A size measure of Functional User Requirements
- based on fundamental software engineering principles
- for business, real-time and infrastructure software
- independent of technology or processes used for the software or project
- usable for both our software measurement objectives
- ‘open’, freely available (via www.cosmic-sizing.org)
The COSMIC method has a well-defined 3-stage Measurement Process

Input from measurement sponsor → Measurement Strategy
Software Context Model → Measurement Strategy
FUR → Measurement Strategy

Definition of each piece of software to be measured and of the required measurement → Mapping Phase
FUR → Mapping Phase
Generic Software Model → Mapping Phase

FUR in the form of the Generic Software Model → Measurement Phase

Functional size of the software in units of CFP → Measurement Phase

Copyright: COSMIC 2017. All rights reserved
In the Strategy phase, we must define the software to be measured, its users, the layers, etc.

Functional Users?
- Humans
- Hardware devices
- Other software

Persistent Storage

Software being measured

Layers?

Level of decomposition?
Measurement Phase: software FUR define a number of ‘Functional Processes’

Theory:

Functional User Requirements

Functional Processes

Sub-processes

Data Movement

Data Manipulation

In practice:

Functional User Requirements

Functional Processes

Data Movements (account for associated data manipulation)
A software functional process must respond to an event detected or generated by a functional user.
A functional process can have four types of ‘Data Movement’ sub-processes:

Functional Users
- Hardware devices,
- Other software or
- Humans

Boundary

<table>
<thead>
<tr>
<th>Entries</th>
<th>Software being measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exits</td>
<td></td>
</tr>
<tr>
<td>Reads</td>
<td></td>
</tr>
<tr>
<td>Writes</td>
<td></td>
</tr>
</tbody>
</table>

Persistent storage

The ‘Data Movement’ is the unit of measure: 1 CFP (COSMIC Function Point)

Copyright: COSMIC 2017. All rights reserved
The Measurement Phase: Size = the total of data movements of all functional processes

- A functional process must have at least 2 CFP
  - A triggering Entry
  - An ‘outcome’ – i.e. a Write or an Exit
- Largest observed functional processes?
  - In banking ~ 65 CFP
  - In avionics >100 CFP
What about non-functional requirements (NFR)?

A joint COSMIC/IFPUG study defined NFR and developed a Glossary of NFR and Project Requirement terms.

A COSMIC Guideline advises how to deal with NFR in projects.

Glossary of terms for Non-Functional Requirements and Project Requirements used in software project performance measurement, benchmarking and estimating

Version 1.0
September 2015

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline on Non-Functional & Project Requirements

How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating

Version 1.
November 2015
Studies show that system NFR may evolve into software FUR, that COSMIC can measure.

Outline

Functional
Requirements

Outline
(System) NFR

Project
Requirements
& Constraints

Architecture

Requirements
Analysis

Approximate
FUR

Detailed
FUR

Detailed
NFR

Build, Test
& Implement

Implemented
software system
or software product

Copyright: COSMIC 2017. All rights reserved
Agile: sizes can be measured at all levels and aggregated up to the system size

COSMIC sizes may be used for:
- early total software sizing and effort estimation;
- US, Sprint, etc, sizing and estimation;
- progress control at any level.
Agenda

- Objectives of software size measurement: FSM methods
- The COSMIC method - key features
- Evidence that the COSMIC method has achieved its goals
- Conclusions
Case: Renault 1) achieves cost estimation accuracy from its ECU designs (automatically!)

Cost vs size (CFP)

Memory size vs software size (CFP)
Case: Italian web software supplier - effort estimation is more accurate with CFP than with FP 2)

25 industrial Web applications

Conclusions:
‘The results of the ... study revealed that COSMIC outperformed Function Points as indicator of development effort by providing significantly better estimations’
A customer request for new or changed function is called a ‘task’

The supplier uses the Scrum method; iterations last 3 – 6 weeks

Teams estimate tasks in USP by a ‘planning poker’ process within each iteration, and convert directly to effort in work-hours

The study involved measurements on 24 tasks from nine iterations for which estimated and actual effort were available

Each task was re-measured in CFP
The actual Effort vs USP size graph (24 tasks) would be poor for effort estimation

\[ \text{Effort} = 0.47 \times \text{Story Points} + 17.6 \text{ hours} \quad \text{and} \quad R^2 = 0.33 \]

Notice the wide spread and the 17.6 hours ‘overhead’
The final actual Effort vs CFP size graph is much better for estimating.

Resulting from CFP measurement, two tasks were identified with very low effort/CFP. These were found to involve significant software re-use, so were considered separately.
“The reality of achieving predictable project performance has driven me to investigate many methods of prediction. COSMIC is the method that lets me sleep at night.”

Denis Krizanovic, Aon Australia, August 2014
Agenda

- Objectives of software size measurement: FSM methods
- The COSMIC method - key features
- Evidence that the COSMIC method has achieved its goals

Conclusions
Conclusion: COSMIC has achieved all its design goals: a major advance in FSM

- The method is stable and ‘future-proof’.
- Proven wide scope of applicability
- CFP sizes correlate very well with effort and code size
- Used around the world:
  - Measurement Manual available in 11 languages
  - 50% of known users are software houses
  - > 30,000 downloads of research & conference papers
- ISO standard. Endorsed by US Agencies GAO, NIST.

Copyright: COSMIC 2017. All rights reserved
Thank you for your attention

(www.cosmic-sizing.org)

Charles Symons cr.symons@btinternet.com


4. Comment on Linkedin discussion, September 4th, 2014

Copyright: COSMIC 2017. All rights reserved