

Software Metrics & Software Metrology

Alain Abran

Chapter 11 COSMIC – Design of an Initial Prototype

Part 3 of the book

- Part 3 of this book illustrates how to leverage the lessons learned from:
 - The quality criteria from the field of Metrology that should be built into the design of software measures (chapters 1 to 5)
 - The weaknesses in the designs of some of the current most popular 'software metrics' (chapters 6 to 10)
- The design of the COSMIC – ISO 19761 measurement method is taken as an example to illustrate how to build a design strong enough to get it recognized as an ISO measurement standard
- Chapter 11 presents the R&D part
- Chapter 12 present the Industrialization part

Agenda

This chapter covers:

- The industry's need for a functional size measure for real-time software
- Research project planning
- Literature review
- Design of the initial prototype
- Field tests in industry
- First-year feedback from industry

Agenda

This chapter covers:

- **The industry's need for a functional size measure for real-time software**
- Research project planning
- Literature review
- Design of the initial prototype
- Field tests in industry
- First-year feedback from industry



Introduction

■ Why improve Function Points?

- Mid-1990: a number of practitioners and organizations in various software domains, including the real-time domain, had come to the conclusion that the 1st generation of functional size measures, based on the design of Function Points, did not provide an adequate measure of the functionality of real-time and embedded software:
 - Given software to be measured and a method to measure it, a measurement method will produce a 'number' but ...
 - This number has limited practical meaning unless it is certain that it provides a meaningful quantification of the attribute being measured.
 - Said differently, applying Function Point rules to software of types other than MIS will produce such a number, but does that number represent an adequate quantification of the functional size of that software?

Introduction

■ Industry-led initiative

- Late 1990s, organizations in 3 countries (USA, Canada & Japan):
 - recognized that a good solution would not be found by luck alone
 - decided to get together to put up the financial resources required to set up a research project with:
 - Dedicated staff with expertise in software measurement
 - Adequate funding
 - A project schedule, including tests in industry.
 - These organizations commissioned a research project to be conducted by the research group headed by Dr. A. Abran, in collaboration with the following researchers and industry participants
 - Later on, other organizations in Europe & Australia provided more funding to finance a 2nd round of tests in industrial settings.

Introduction


- The industry partners recognized that:
 - R&D takes times and need resources
 - That the initial design of the measure would need to evolve
 - A prototype had to be built in a first phase to meet their own requirements within their own context

- That this R&D phase needed to be followed by a 2nd phase:
 - To get additional feedback from participants outside of the design team
 - To increase consensus on the design
 - To bring it to world standard by submitting it to the ISO approval process

- This corresponds to the Industrialization of the design of a measurement method

Agenda

This chapter covers:

- The industry's need for a functional size measure for real-time software
 - **Research project planning**
 - Literature review
 - Design of the initial prototype
 - Field tests in industry
 - First-year feedback from industry
- 

Research Project Planning

The prototype for the design of the new measurement method had to retain the key quality characteristics of FP from a measurement perspective:

- **Relevance:**
 - The measurement technique is perceived by practitioners as adequately measuring the functional size of the applications in their domain.
- **Measurement instrumentation: for repetitiveness.**
 - This means that different individuals, in different contexts, at different times, and following the same measurement procedures will obtain measurement results with the following quality criteria:
 - relatively similar.
 - obtained with minimal judgment.
 - are auditable.
- **Initial measurement instrumentation:**
 - a measurement manual to document & clarify the measurement objectives & measurement rules adopted by a user group.
- **In the public domain.**
- **Practical:**
 - based on current software development practices & on the content of the software documentation.

Research Project Planning

Project Plan: 5 steps to develop an initial prototype

1. *Literature review*

- Reported usage of FP to measure real-time software,
- Identification of extensions proposed to adapt FP to this type of software
- Analysis of these proposals to identify their strengths and weaknesses.

2. *Proposal for an extension to FP for real-time software*

- Based on the lessons learned

3. *Field tests of the design of the prototype*: measuring some real-time software applications in industry.

4. *Analysis of the measurement results*: against criteria such as:

- Perception of the application specialists with respect to the coherence between the measurement results & size of the applications measured.
- Time required to identify & measure the function types.
- Difficulty in learning & applying the basic concepts, as well as the measurement procedures & rules of the proposed prototype.
- Comparison between the measurement procedures & results obtained with the prototype & Function Points.

Research Project Planning

5. *Public release:*

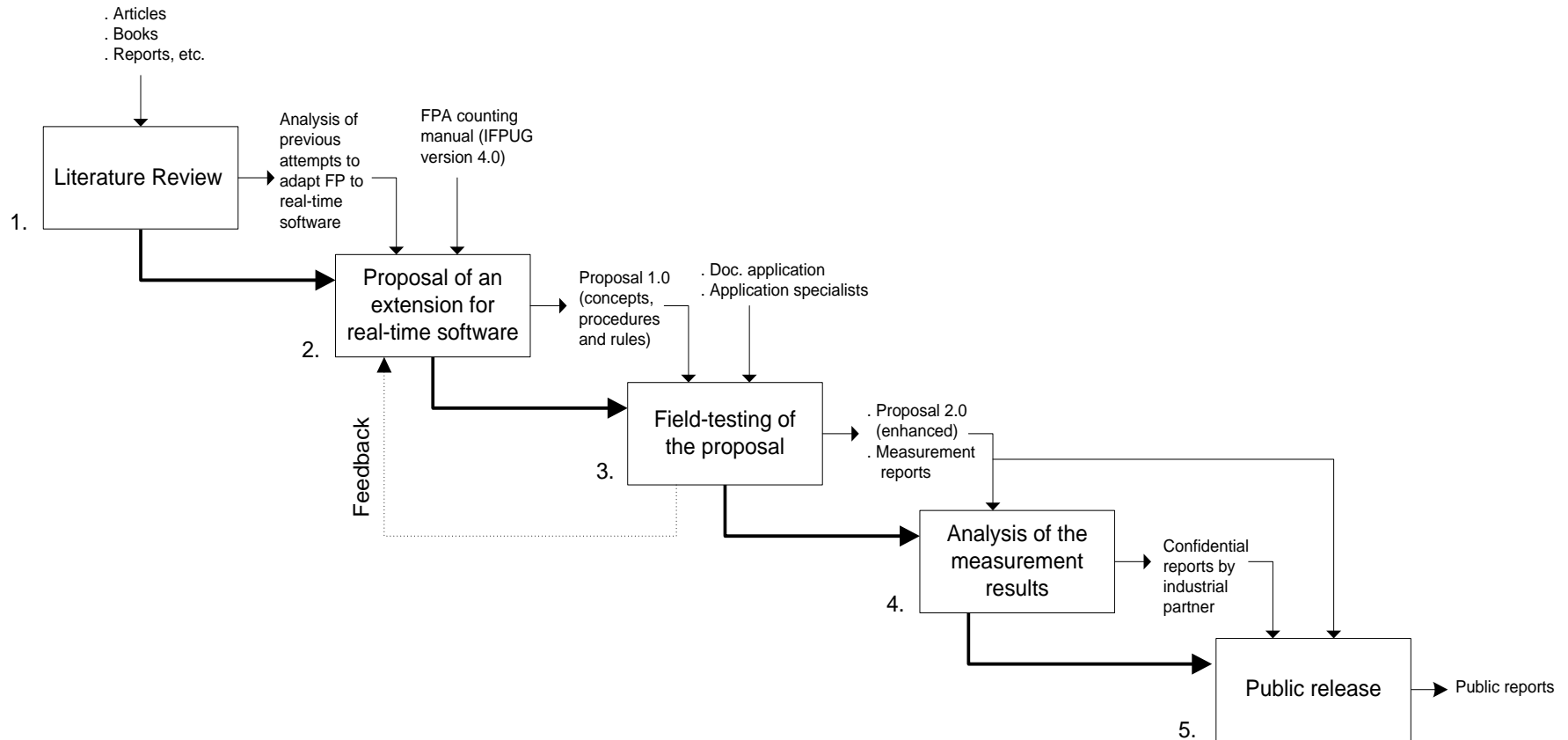
- Description of the extension proposed,
- Measurement procedures and rules,
- Overview of the field tests, and
- Comments on the measurement results from a measurement instrumentation perspective.

■ See the figure in next slide (with the project steps, their inputs, and their deliverables).

- Steps 2 and 3: strongly related and interactive:
 - That is, as the field tests were progressing, the initial prototype was improved based on the feedback obtained from the industry participants in the measurement sessions.


Research Project Planning

Figure: Project Phases for the Initial Prototype



Agenda

This chapter covers:

- The industry's need for a functional size measure for real-time software
- Research project planning
- **Literature review** 
- Design of the initial prototype
- Field tests in industry
- First-year feedback from industry

Literature Review

■ Findings and lessons learned

- Most of the publications on FP up to the time of that literature review in 1997 did not deal with its application to real-time software, and the datasets used in most publications originated from MIS software applications
 - this is still the situation in 2010+
- Several authors [Jones 1991; Reifer 1990; Whitmire 1992, 1995; Galea 1995] agreed that the FP structure was not adequate for measuring processor-intensive software with a high number of control functions and internal calculations.

Literature Review

- Attempts to adapt FP to real-time software identified at that time:
 1. Status quo: IFPUG Case Study 4 on real time [IFPUG 1997].
 2. Application Features [Mukhopadhyay and Kerke, 1992].
 3. Feature Points [Jones 1988].
 4. Asset-R [Reifer 1990].
 5. 3D Function Points [Whitmire 1992, 1995].
- However, none of these approaches had succeeded by the late 1990s in gaining any significant market acceptance in the real-time software market.

Literature Review

1. Status quo: IFPUG

- The status quo approach assumes that Albrecht's FP method is applicable as is to non MIS software.
 - Approach adopted by IFPUG.
 - However, many practitioners and researchers have indicated that the status quo method applied as is to non MIS software does not adequately capture the functional size of this software.

2. Application Features

- 'Application Features' were proposed as a measurement technique by [Mukhopadhyay & Kerke 1992]. These researchers were not directly interested in redefining FP as a measurement method, but were rather looking for a way to estimate functional size, in FP, earlier in the development life cycle.

Literature Review

- Their goal was to apply this measurement technique to process control software in the manufacturing domain (a category of real-time software).
 - This is interesting, because it identifies specific characteristics of a certain category of real-time software which contribute to functional size and which are known very early in the development life cycle.
- However, these characteristics (physical positioning and movement) were specific to a specialized application domain and could not be generalized across all types of real-time software.
 - Detailed procedures and rules, to ensure accuracy and repeatability over time in the measurement of the three application features identified, were not provided.
 - The authors focused on the presentation and analysis of an estimation model, the measurement process itself was not covered.

Literature Review

3. Feature Points

- Feature Points were developed by [Jones 1988].
 - According to Jones, real-time software is high in algorithmic complexity, but sparse in inputs and outputs, and, when the FP method is applied to such software, the results generated appear to be misleading.
 - However, it was pointed out by [Whitmire 1992] that the definition of algorithms and the rules proposed were not sufficient for measurement purposes:
 - *“No guidance is provided to identify algorithms at the desired level of abstraction.”*

Literature Review

4. Asset-R

- 'Asset-R' was proposed by [Reifer, 1990]. According to [Reifer, 1990], Albrecht's method failed to successfully handle four important characteristics associated with real-time and scientific software: parallelism, synchronization, concurrency, and intensive mathematical calculation.
- New function types were therefore introduced, according to the type of software measured, while the weighting factors and the adjustment factors proposed in Albrecht's method were eliminated.
- However, there was a lack of detailed definitions, measurement rules, and examples for the new function types (modes, stimulus/response relationships, rendezvous), and it was not clear how to measure them.

Literature Review

5. 3D Function Points:

- Developed by [Whitmire 1992, 1995] as a result of a study investigating the use of FP in the measurement of scientific and real-time software.
 - According to this study, FP does not accurately measure this type of software because it does not factor in all the characteristics that contribute to the size of the software.
 - His criticism is based on the premise that all software has 3 dimensions: data (stored data and user interfaces), functions (internal processing), and control (dynamic behavior).
 - According to this scheme, FP measures only one dimension: the data dimension.
 - Whitmire therefore proposed a set of function types for each of the dimensions identified and a mechanism to combine the points assigned into a single “3D FP index”.

Literature Review

■ Pre-tests

- Of the 5 candidates identified in the literature review, the 3D Function Points method looked to be the most relevant at that time.
 - It was selected for a pre-test on some industrial applications in order to obtain empirical observations, and, perhaps, for use as the starting point for the project.
 - 2 different software applications from 2 different domains (power supply and telecommunications) were measured.
 - Key problems encountered:
 - The concepts and the procedures and rules not detailed enough to allow the identification, without ambiguity, of the new function types proposed (Transformations, States, and Transitions).


Literature Review

- For the identification of the proposed States and Transitions function types, documentation for the finite state machines (FSM) was a prerequisite.
 - However, at the industrial sites where this proposal was tested, this type of documentation was not in use.
 - Furthermore, feedback from many real-time designers in other organizations also indicated that, even though most practitioners knew about this type of notation system, they did not document functional requirements using state diagrams.

- The feedback from these pre-tests in industry and from other contacts in industry gave an indication that industry might be very reluctant to endorse a measurement method based on such a state diagram-type of notation of requirements when they were not using it even for the design of their own software.

Agenda

This chapter covers:

- The industry's need for a functional size measure for real-time software
 - Research project planning
 - Literature review
 - **Design of the initial prototype**
 - Field tests in industry
 - First-year feedback from industry
- 

Design of the Initial Prototype

■ The mismatch between the perceived size of functionality in real-time software and its size in FP

- The transactional function types in FP (External Inputs, External Outputs, and External Inquiries) are based on the concept of the elementary process.
 - An elementary process was defined in IFPUG, 1994: “The smallest unit of activity that is meaningful to the end user of the business.”
 - This elementary process must be self-contained and leave the business of counting the application in a consistent state.
- However, the number and nature of the steps or *sub processes* required to execute the elementary process are not taken into account.
 - Based on empirical observations, MIS processes of the same type have a relatively stable number of sub processes.
 - However, in real time software processes, by contrast, the number of sub processes varies substantially – see the control process in examples 1 & 2.

Design of the Initial Prototype

Example 1 – An engine temperature control process

Main purpose of the process: to turn on the engine's cooling system when necessary.

- A sensor enters the temperature into the application (sub process 1).
- The temperature is compared to the overheating threshold temperature (sub process 2).
- A Turn On message can be sent to the cooling system if needed (sub process 3).

The application is not in a consistent state until all the sub processes of the temperature control process are completed.

The temperature control process has, therefore, *3 sub processes*.

According to standard FP rules, only one transactional function would be identified, because there is only *1 elementary process*.

Design of the Initial Prototype

Example 2 – An engine diagnostic process

Main purpose of the process: to turn on an engine alarm when necessary.

- 15 engine sensors (all different) send data to the diagnostic process (15 sub processes, 1 unique sub process for each kind of sensor).
- For each sensor, the set of external data received is compared to threshold values read from an internal file, 1 unique file for each kind of sensor (15 more sub processes, 1 unique sub process for each kind of sensor).
- Depending on a number of conditions, an alarm on the dashboard may be turned on (1 sub process).

In this example, the engine diagnostic process consists of *31 sub-processes*.

The application is not in a consistent state until all the sub processes of the diagnostic process are completed.

According to standard FP rules, only 1 transactional function would be identified, because there is only *1 elementary process*.

Design of the Initial Prototype

- Therefore, when the Function Points rules are used,
 - example 1 with 3 sub processes and example 2 with 31 sub processes would have approximately the same number of points related to transactions
 - even though the real-time community would disagree that these 2 functional processes have the same functional size.
 - There is a strong case to be made that an adequate functional measure of real-time software should take into account the fact that some processes have only a few sub processes, while others have a large number of sub processes.

Design of the Initial Prototype

- ***The initial solution: A real-time extension to FP***
 - To measure these functional characteristics of real-time software adequately (i.e. to avoid this mismatch between the numbers obtained and the practitioners' perception of corresponding functional sizes), it is necessary to consider the sub processes executed by a control process.
 - The extension to FP proposed in 1997, referred to as Full FP (FFP), introduced new transactional function types to measure these characteristics:
 - External Control Entry – ECE,
 - External Control Exit – ECX,
 - Internal Control Read – ICR, and
 - Internal Control Write – ICW
 - In this extension, the new function types were only used to measure real-time control data and processes.
 - See the Advances Readings section for a discussion on the key differences between FP and the extension proposed.


Field Tests of the Prototype

■ Field test planning

- Objective of the field tests: to measure some real-time software applications to verify in an empirical way the applicability of the concepts proposed:
 1. Do the users of the measurement technique agree that the proposed measure has met its stated objective of measuring the functional size of real-time software?
 2. Is the proposed measure 'good enough' from the users' own perspective:
 - e.g. is it achieving the right balance of multiple simultaneous goals, even though any one goal might not have been achieved optimally [Bach, 1997]?
 3. Do the users agree that the extension proposed has been designed objectively, precisely, and in an auditable manner, and that someone else with the same set of rules would come up with the same results (relatively), and again, results that are 'good enough' ?
 4. Do the users agree that the proposed measurement procedures are based on current practices of what is effectively documented at the present time?

Agenda

This chapter covers:

- The industry's need for a functional size measure for real-time software
- Research project planning
- Literature review
- Design of the initial prototype
- **Field tests in industry** 
- First-year feedback from industry

Field Tests of the Prototype

- The software applications were measured with the extension proposed, as well as with IFPUG version 4.0, so that comparisons could be made.
- 3 field tests were conducted over a 4-month period.
- For the measurement of these applications, at least 3 people participated in each measurement session:
 - at least one application specialist, and
 - 2 certified FP specialists who were members of the FFP design team.

Field Tests of the Prototype

- FP and FFP measurement results for the transactional function types – table below.
 - It can be observed that, in the presence of multiple sub processes of a single process, FP produced fewer points than FFP.
 - Indeed, FP has been criticized for generating small FP sizes in a real-time environment [Jones 1991; Galea 1995] which seem unrelated to the perceived larger functional size of the software measured [Galea 1995].

Function Type	Application A		Application B		Application C	
	Occurrences	Points (Size)	Occurrences	Points (Size)	Occurrence	Points (Size)
FP (IFPUG 4.0)						
- External Input (EI)	40	202	6	21	15	50
- External Output (EO)	2	14	2	11	17	73
- External Inquiry (EQ)	12	40	1	6	0	0
Total Size	54	256 FP	13	69 FP	32	123 FP
Full FP (FFP)						
- External Control Entry (ECE)	123	123	10	10	67	69
- External Control Exit (ECX)	93	97	8	10	136	139
- Internal Control Read (ICR)	395	403	14	18	100	103
- Internal Control Write (ICW)	142	154	8	8	165	168
Total Size	753	777 FFP	45	83 FFP	468	479 FFP

Field Tests of the Prototype

- Since the FFP design takes into account the sub processes integrated within a single control process by identifying the different groups of data received, sent, read, and written, FFP will generate more points than the standard FP method – see table.
 - The real-time specialists consulted strongly concurred that a measure not taking into account user requested sub processes, like FP, could not be a ‘good enough’ functional size measure of their applications.
- Therefore, FFP results represented for them a more relevant measure of the functional size of their applications:
 - it had a greater ability to discriminate software functional size in accordance with their own intuitive perception of differences in functional size.

Field Tests of the Prototype

- The results of the field tests also confirmed that, in those tests, the number of sub processes in a real-time process varies substantially. For example:
 - there were processes with only 3 sub processes,
 - while others had over 50 sub processes.
- One of the project's industrial partners conducted a 4th field test without the assistance of the researchers. The feedback from this industrial partner is summarized as follows:
 - Concepts and measurement procedures in the FFP measurement manual were relatively clear and easy to understand.
 - It was not difficult to measure without the assistance of an FFP specialist.

Field Tests of the Prototype

- Feedback from the field tests about the quality of the measurement prototype designed:

Quality 1: Ease of understanding

- One criticism often made of FP is that it constitutes a set of complex definitions and time-consuming procedures, and that it takes an FP expert to produce an accurate FP size.
- The new function types were designed to be simpler to learn and master.
 - This simplicity was confirmed during the field test measurement sessions.
 - Once the application specialists understood the definitions of the new function types, they had no problem identifying them.
 - After a full day of FFP measurement, they were able to measure with limited assistance.

Field Tests of the Prototype

- According to application specialists participating in the measurement sessions, this is mostly due to the fact that:
 - it is much easier to identify function types that only refer to one type of action (for example, receiving data) as in FPP,
 - than it is to identify function types that potentially refer to more than one action, as in FP (some mandatory and some optional, and multiple possible combinations, some of which qualify and others that do not).
- In addition, application specialists believed that the definitions, measurement procedures, and rules were clear and detailed enough that different individuals would be able to come up with relatively similar results.
 - They also believed that the proposed concepts, measurement procedures, and rules were based on current practices of what was documented at the time.

Field Tests of the Prototype

Quality 2: Measurement effort

- FFP and FP measurement efforts were found to be similar.
 - Even though more function types had to be measured with FFP, they were easily identified and therefore the effort required to measure them was no greater.
 - The application specialists seem to require less assistance from FP experts when measuring with FFP than with FP, so the identification of more function types did not increase the measurement effort.

Field Tests of the Prototype

To summarize: the results of the field tests were positive:

- According to the real-time specialists present at the field tests, FFP met its stated objective of measuring the user's functional requirements adequately:
 - They believe that this was done objectively, precisely, and in an auditable manner, and that someone else with the same set of rules would come up with more or less the same results.
 - They also believe that FFP measurement procedures and rules are based on current practices of what is really being documented at the present time.

Field Tests of the Prototype

■ *Limitations of the initial prototype*

- The designers of this prototype were very involved, not only throughout the design process, but also in its evaluation.
 - They could not, therefore, be totally objective in their evaluation of the prototype.
- The field tests were carried out in a small number of organizations, and without a sufficient number of field tests to make them significant for statistical purposes.
 - The results could be considered only as anecdotal, and without generalization power per se.
- It was recognized that more people would need to be brought on board to strengthen the prototype of the proposed measurement method to ensure that the prototype would reflect a broader consensual basis and find a broader application domain.

Agenda

This chapter covers:

- The industry's need for a functional size measure for real-time software
- Research project planning
- Literature review
- Design of the initial prototype
- Field tests in industry
- **First-year feedback from industry**

First-year Feedback from Industry

Feedback

- Strong interest was voiced from organizations that used and tested FFP.
- Within a year, a new international users group was set up to foster the scaling up and deployment of the prototype:
 - The 'Common Software Measurement International Consortium' - COSMIC was founded in 1998 as a voluntary initiative of software measurement experts, which included practitioners and academics from the Asia/Pacific region, Europe, and North America.
- The aim of the COSMIC group is to design and bring to market a new generation of functional size measurement methods for software.
 - The work done by the COSMIC group to scale up the COSMIC method is presented in chapter 12.
 - In 2010, the COSMIC group includes representatives from at least 13 countries.

Summary

This chapter has covered:

- The industry's need for a functional size measure for real-time software
- Research project planning
 - How to organize this type of R&D project for the design of a measurement method
 - The literature review and the selection for a pre-test
 - The design of the initial prototype to address the industry objectives
 - The results of the field tests in industry to verify that the design objectives had been met
- First-year feedback from industry and the set up of the COSMIC group as a user group