SHORT PAPER INDUSTRY FOR IWSM (AMSTERDAM, 5th November)

Practical experimentations with the COSMIC method in Automotive embedded software field.

Sophie Stern*

RENAULT, Embedded software group
TCR RUC T 65, 1 avenue du Golf, 78 288 Guyancourt cedex, France
sophie.stern@renault.com

Abstract

More and more functionalities are available in cars (to increase security, to reduce the CO2 emissions, to improve the connectivity with the outside world and so on) and the most part of these new features is realized by software. The automotive industry is used to manage very accurately the physical parts costs with its suppliers and has now to face to the software parts development costs management too. For that, it is necessary to construct effort estimation models based on a standard functional size measurement method which must be explicable, factual, simple and reproducible.
The major goal of this paper is to give a feedback on the Renault practical experimentations with the COSMIC method on the embedded software field.

Keywords

The COSMIC functional size measurement method, Effort estimation models, Software development costs estimations, Software productivity management.

1. Introduction

During the last ten years, as the functionalities available for customers are more and more numerous, the cars' complexity has evolved increasingly. This complexity is mainly supported by calculators (Airbag, ABS,…) that we call ECU for Electronic Control Units.
Historically, Renault subcontracts the development and the manufacturing of its ECU to many suppliers. An ECU presents the particularity to be part of hardware and part of embedded software. For several years, Renault has been used to pay each ECU as a whole. But with the increase of the software complexity, and on the same time the decrease of the electronic parts costs, the software development cost is less and less marginal and may be higher than the hardware one for major ECU.

If Renault, as the other cars manufacturers, is very used to estimate the development cost of physical parts such as harness or electronic components, it is quite lacking in for software development cost estimation.
Several departments but especially the Renault purchasing one asked to the Renault embedded software group to find a method to estimate the embedded software development cost.

The main goal of this short industry paper is to show how the COSMIC method has been used in the Renault experimentation and the learnt lessons.

After the presentation of the reasons why Renault chose to evaluate the COSMIC method for embedded software development effort management, I will present the goals assigned to the experimentation, first results, best practices and I will conclude on what next.

2. Why had Renault chosen to evaluate the COSMIC method for embedded software development effort management?

In 2008, the Renault purchasing department requested from the embedded software group a method to estimate the embedded software development cost in order to manage it more accurately.
The first point was to find a standard method to estimate the development effort which could be converted later in development cost with applying the suppliers' rates.
As everybody knows, it is not possible to measure a piece of software with a meter or a balance, and it appears that it is not possible anymore to measure one with the number of written lines code because it would have no sense with automatic coding.
So we decided to interest ourselves to the Functional Size Measurement (FSM) methods.

The Renault embedded software group started by a state of the art of the FSM methods, two were studied particularly at first because of their past in Renault: the IFPUG and the COCOMO methods.
The COCOMO method had been experimented a few years ago in the ECU diagnostic department, nevertheless with unsuccessful results.
The IFPUG method has been used for several years in the Renault Information System department, and no new information system can be launched without its cost has been first evaluated by the IFPUG software effort estimation cell.
Nevertheless, the possible application of IFPUG on embedded software had to be checked with an evaluation.

In order to benchmark at least two FSM methods, we chose the COSMIC method as it was announced to be well adapted to real-time software as the embedded software in ECU is.

Our first experiments started on the Engine Control Unit at mid 2008 with the IFPUG and COSMIC methods. The Engine Control Unit is modular and each of its modules is a set of specifications under the Matlab/Simulink tool with also textual requirements. The effort supplier invoice is available for each module.
Functional size measurements were realized on the same nine modules with the two studied methods, and then results were compared.
In the Renault experimentation, the IFPUG functional sizes measurements were always higher than the COSMIC ones, the COSMIC method suited well for embedded software whereas the IFPUG method appeared not pertinent especially when the functional software size increases. Furthermore, the measurements with the COSMIC method seemed to us easier than the ones with the IFPUG method.
So we decided to pursue the experimentation for embedded software with the COSMIC method in a project way.

3. The Renault COSMIC project

In 2009, we decided to pursue in a project way and to take into account another ECU, the Body Control Module (BCM). The BCM is specified by Renault with the Statemate tool and textual requirements. The BCM is specified in software functions, the supplier effort invoice is available for each of them.

The first step was to measure functional sizes on BCM functions and on Engine Control Unit modules and to try to find linear regressions with the supplier effort invoices. For the Engine Control Module, it was on software evolutions, for the BCM, it was on software evolutions and on new developments.
In each case we found one or several linear regressions. At this moment, as the Renault implicated actors and managers were numerous, from different departments and with different interests in the COSMIC method, it appeared to me that it was necessary to define very clearly the goals of the COSMIC method application evaluation on Renault ECU.
We defined with purchasing departments, departments in charge of specifications, ECU project management, four major goals.

3.1. The four major goals of the COSMIC method application evaluation on Renault ECU

- Have indicators to manage the supplier productivity year after year.
- Predict the cost of the software functions development in order to negotiate with suppliers the right price.*
- Be able to estimate a function software development cost as soon as its specification is written to decide to implement or not the function.
- Benchmark the productivity of the different suppliers.

* This objective is a first milestone. The final goal is to be able to contract the price of a functional size in CFP (COSMIC Function Points) with suppliers. Notice that this process is already applied since several years for information systems.

These major goals are then declined in smaller goals for each.

I put in place steering committees for each ECU development sector to check regularly the goals and of course to check the advancement on work packages and usual indicators on projects: problems resolution, risks and planning.

3.2. A cell of measurers, COSMIC measurement textual guides.

As the software functional size measurements are performed by several measurers, it is mandatory to define very clearly and without any ambiguity, the way of measuring functional sizes. We wrote COSMIC measurements textual guides. Several functional size measurements were performed independently by different persons, these experiments showed that even manual measurements are very reproducible with at the most 1% of difference.

3.3. The way of constructing COSMIC models.

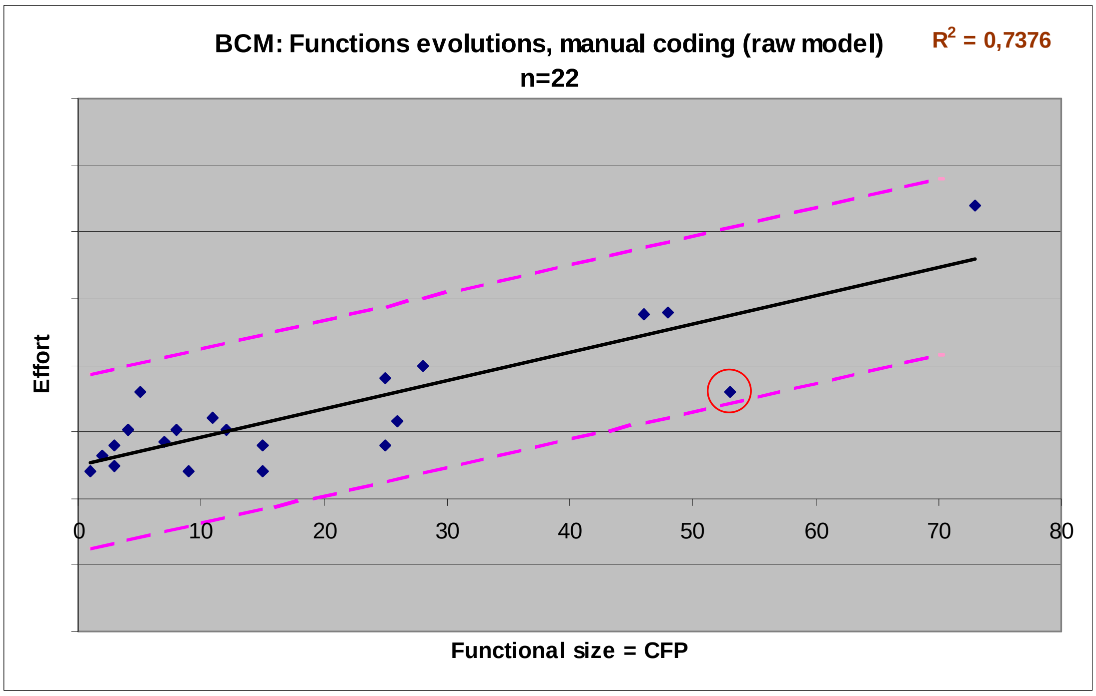
We made the choice to have one different COSMIC model for each ECU. We also constructed different models for each supplier. This is interesting to have standards of comparison and to benchmark suppliers.
Until know, we made the choice to split models when influent factors are different, but I intend to pursue our statistics studies with multi-factors approaches.

3.4. Obtained COSMIC results.

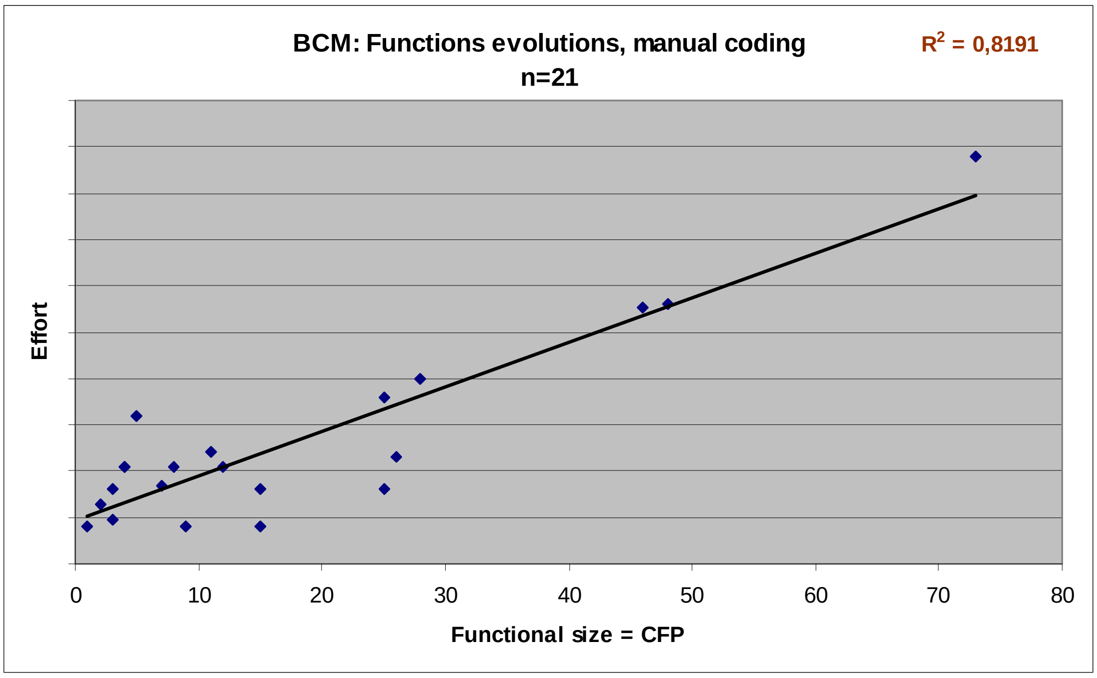
We followed the COSMIC method and we constructed regression models with only one variable: the functional size in CFP.
After our experimentations, my opinion is that it is necessary to find the balance between too many models and models easily understood by many actors as the ECU development project, the purchasing, suppliers. The split of one COSMIC model in several models must always have a meaning.

BCM COSMIC models

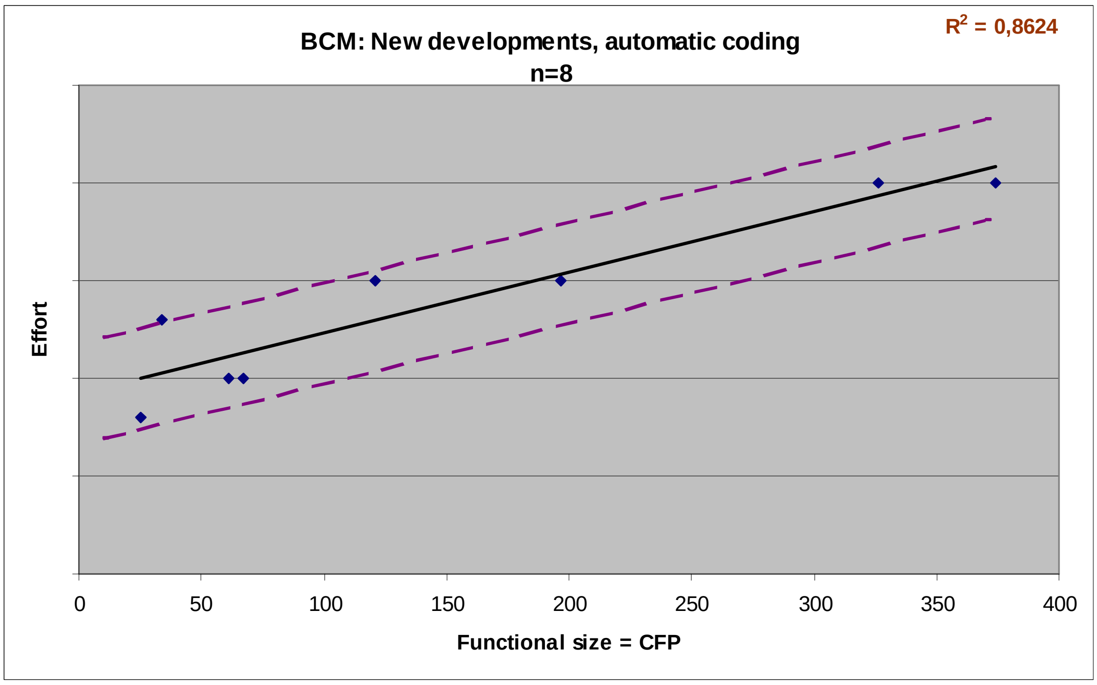
The BCM COSMIC model was first split between: new developments and functions evolutions. Then theses two BCM models were split again between: manual developments and automatic ones. Four reference COSMIC models are defined at the moment for the Renault BCM.

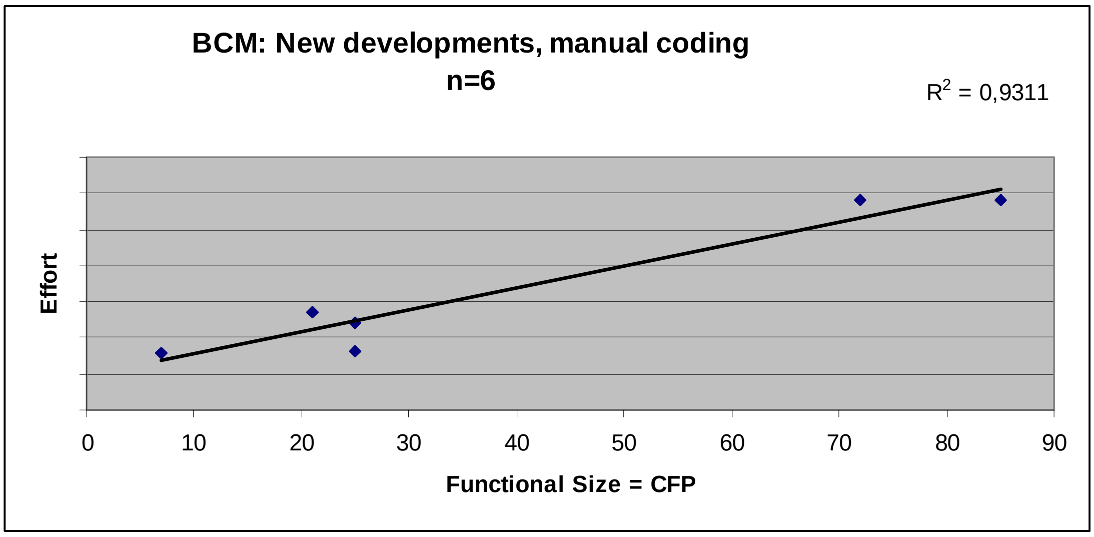


On this raw model, there is one point very close to the confidence interval limit. This BCM function was developed by an expert in the supplier team so we decided to remove this point from the regression curve and to capitalize in a checklist of particular functions.
After this correction, we have the following reference model.



The number of points for the two BCM COSMIC models for new developments were low but the coefficients of determination are correct and these two models have been very helpful for us to realize predictions during a request for quotation with suppliers for a new BCM development.

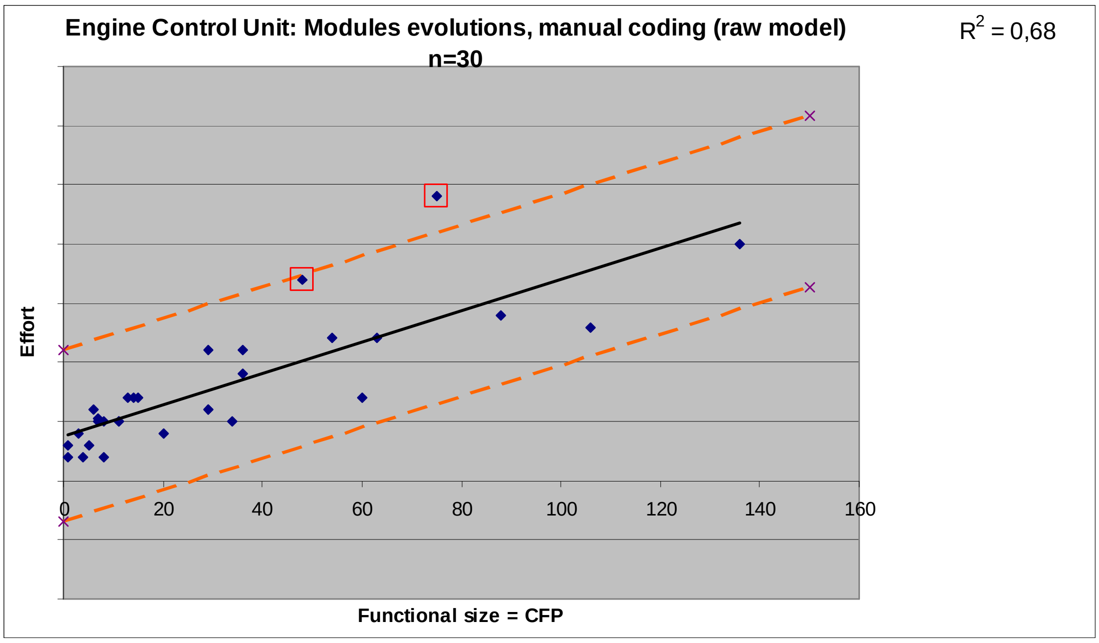




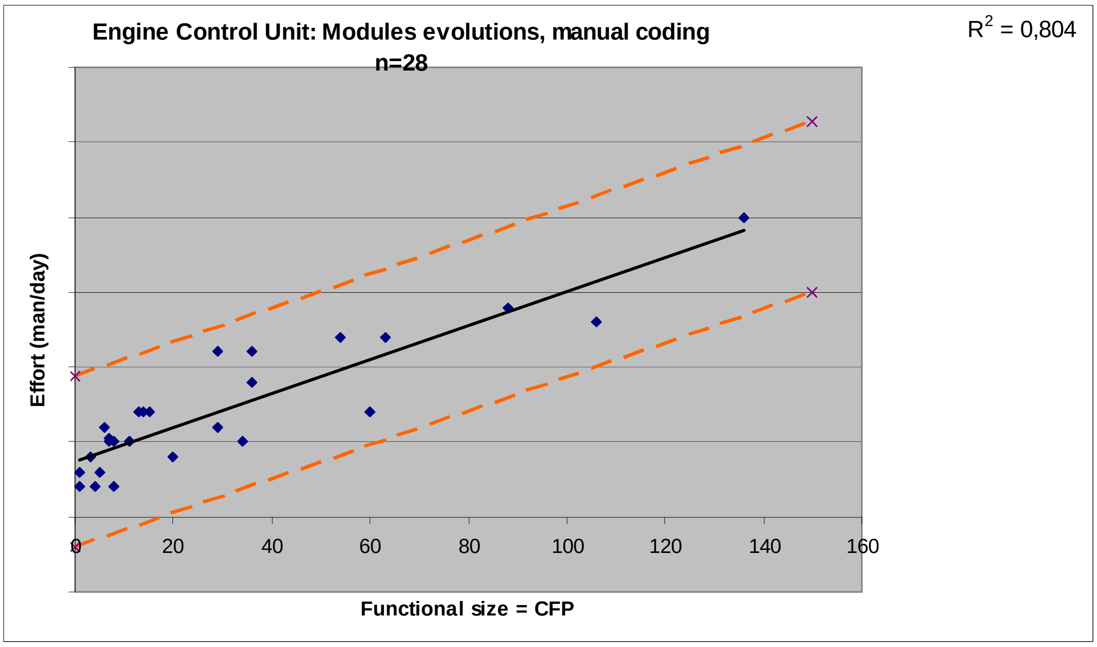
An important point will be to complete the reference COSMIC models with new data when available.

Engine Control Unit COSMIC models

For the Engine Control Unit, the COSMIC model was only split between the different suppliers, all considered developments are modules evolutions. One COSMIC model for one supplier is shown below.



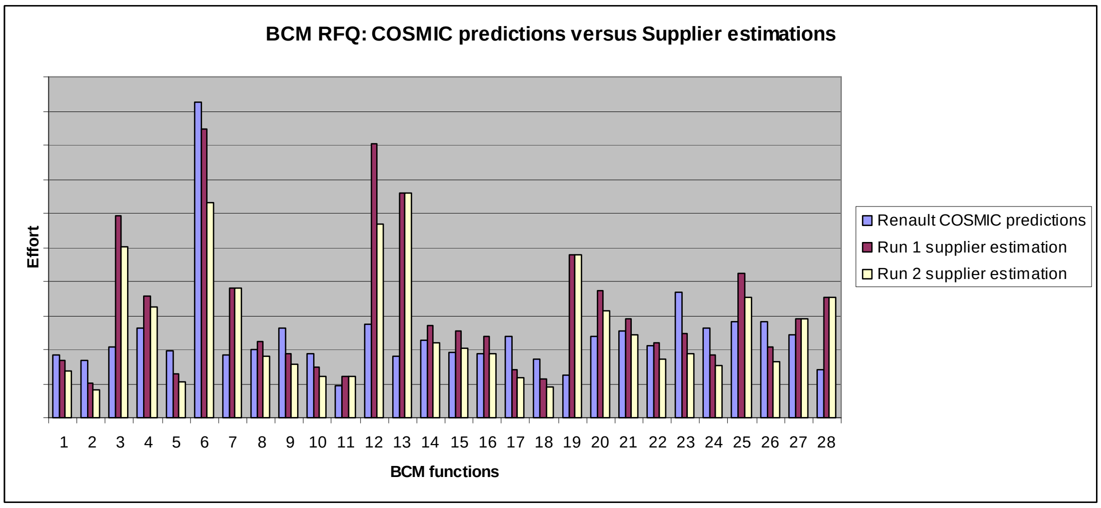
The two points in squares correspond to modules very different by their nature to the other ones. They have been removed from the COSMIC model and capitalized in a checklist until we can construct a COSMIC model for this kind of modules.
After this correction, we have the following reference model.



A real experimentation on BCM: COSMIC predictions in a RFQ phase.

We applied our BCM COSMIC new developments models during a Request For Quotation with suppliers for a new BCM in order to have a target software development cost on the applicative software. As soon as the specifications were written, we predicted the development effort for each BCM function within an prediction interval , we also predicted the development effort and the associated prediction interval for the whole applicative software. Then we negotiated with the supplier its estimations.

This is the comparison between our COSMIC predictions and the supplier estimations after the first round and after the second round with the supplier.



The negotiation is not finished but this example shows that factual measures realized with a standard method is a good lever to negotiate.

Predictions for the Engine Control Module.

For the Engine Control Module, as there are regular software development invoices for modules evolutions, we did predictions on two modules. Each supplier invoice was in our prediction interval.
We have to pursue such experiments to make everybody, inside and outside the company, confident in the predictability of the COSMIC method.

3.5. The success keys for applying the COSMIC models

➢ *Capitalization.*

As we are not fortune-tellers, it is not possible to predict software development effort or cost without capitalization on ECU projects from the past. The storage must be precise and complete, in our case we need to store software specifications and the development cost for each specification. For some ECU, the first step will be to change the process of costs negotiation with the supplier to obtain the detailed software costs.

- *The implication of the ECU purchasers.*

The goals of the ECU purchasers are in line with the COSMIC method possibilities. It is important to explain them the method and to show some successful experiments in order to have their support because they may be a very good lever of change inside the company and outside with suppliers.

- *The help of the ECU project team.*

The COSMIC measurers are often embarrassed to explain the points outside the COSMIC regression curves, the help of the ECU project team is mandatory to interpret the modules intrinsic particularities or the ECU development process specificities.
Furthermore, the implication of the ECU project team is very important to deploy the method with the supplier.

- *Meeting points with the supplier.*

The final goal of using the COSMIC method is often to share it with suppliers and to contract agreements on the basis of factual software functional measures.
It is important to explain step by step the COSMIC method to suppliers, to make experiments with them to show the accuracy of the method.

- *Multidisciplinary team.*

The COSMIC core team needs people with knowledge and capabilities in computer science and in statistics. As it is very difficult to find people with these two competencies, the best way is first to mix people with different profiles in the same team and to trainee them.

- *Process change.*

To improve the way to buy software to suppliers, it is necessary to conduct a process change in the organization with a roadmap.

4. And now, what next?

As the Renault experimentations with COSMIC are very encouraging for embedded software cost management, we intend to continue with the major following actions:

- ✓ Pursue COSMIC experimentations on other ECU.

- ✓ Experiment statistical multi-factors models.

- ✓ Find one method to work upstream when simulation models are not available.

- ✓ Estimate other software development efforts than applicative: basic software development effort, ECU functional validation effort,…

- ✓ Automate the whole measurement process. At last but not least.