Sanae Saadaoui[1], Annick Majchrowski[2], Christophe Ponsard[3]

Tips and Hints for an Effective COSMIC Learning Process Derived from Industrial Training

Abstract

As a technology transfer centre specialising in IT, we have been training companies to use the COSMIC estimation methodology for several years. This paper reports a retrospective look at a number of recurrent issues that occurred during those training sessions across a large variety of business domains, company sizes and maturity levels. Some of these issues are related to the COSMIC method itself or to the functional size measurement in general, while others have their roots in other fields like project management or requirements engineering. Failing to address these issues in an adequate way may prevent from meeting the training goals and the successful adoption of the COSMIC method. We also report how we have designed our training to handle most of these issues and to capitalize on returns from trainees. We will explain the most recurrent issues and give some hints and tips to design and conduct training.

Keywords: effort estimation, COSMIC Function Points, training, requirements engineering, project management, industrial application

1. Introduction

Estimating the effort required for software development projects remains a challenge for project managers. A survey on effort estimation reported that 60% to 80% of IT projects exceed the budget and/or time limits with an average

[1] CETIC Research Centre, rue des Frères Wright 29/3, 6041 Gosselies, Belgium, sanae.saadaoui@cetic.be

[2] CETIC Research Centre, rue des Frères Wright 29/3, 6041 Gosselies, Belgium, annick.majchrowski@cetic.be

[3] CETIC Research Centre, rue des Frères Wright 29/3, 6041 Gosselies, Belgium, christophe.ponsard@cetic.be

overrun of 30% to 40%[4]. Moreover, many projects ending within time have to compromise on quality. Project failure factors are numerous, but one of the major causes is bad estimates, mainly due to underestimations and subjective optimistic biases[5].

Several methods for estimating the effort have been defined. Model-based methods use algorithms to consolidate available data and make predictions about new projects, whereas expert-based methods rely on human expertise along with guidelines. Both approaches can combine hybrid methods[6]. However, the produced estimates are often too approximate, and unreliable. Producing accurate estimates is a delicate process that should be performed in compliance with good practices and methodological ways to be effective and reliable. This is far from an intuitive approach. A fundamental mistake is to believe that estimators can quickly calculate or even guess "unique numbers" based on "black-box" tools, or on a manual of recipes like alchemists transforming dust into gold[7].

Among the available methods, Function Size Methods (FSMs) is the only internationally recognised and ISO standardised technique to measure the size of Functional User Requirements, i.e. independently of any constraints of how the software is built. It can be applied from the requirements analysis phase and can cope with different tools, techniques and technologies developed to build software over time[8]. In the scope of this paper we will consider the FSM and more specifically COSMIC, which is one of the most recent methods of this kind[9]. As a "second generation" method, COSMIC took into account the mistakes and limitations of the first generation methods, especially to develop a new approach adapted to new software development methodologies and new types of projects.

[4] K. Molloken, M. Jorgensen, *A Review of Survey on Software Effort Estimation, Empirical Software Engineering*, ISESE 2003.

[5] K. Hemant, *Why IT Projects Really Fail: Over-optimism and Complexity Are Just Some of the Many Reasons Why IT Projects Continue to Suffer*, "CIO", December 2013.

[6] T. Menzies, Z. Chen, J. Hihn, K. Lum, *Selecting Best Practices for Effort Estimation*, "The IEEE Transactions on Software Engineering", November 2006, vol. 32, no. 11.

[7] A. Abran, *Software Estimation: Transforming Dust Into Pots of Gold?*, Proc Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM–MENSURA), Rotterdam, Netherlands, October 2014.

[8] W. Khatibi, D.N.A. Jawawi, *Software Estimation Methods. A Review*, "Journal of Emerging Trends in Computing and Information Sciences" 2010, vol. 2, no, 1, pp. 21–28; *Total Metics: Methods for Software Sizing – How to Decide Which Method to Use*, Version 1.1, August 2007.

[9] ISO: ISO/IEC 19761, Software Engineering – CFF – A Functional Size Measurement Method 2003, International Organization for Standardization, Geneva 2003.

Despite its growing popularity, applying COSMIC requires solid training. As for other FSMs, a specialised training session of two to three days is typically required. In order to become proficient, it is necessary to practise during several months by applying the technique in a variety of situations[10].

The training phase is hence quite critical and the goal of this paper is to report our experience in organizing such training since 2009. Becoming a good trainer is also a long process, and over the years, we have learnt a lot about recurring difficulties and how to best address them. They are not only related to the COSMIC method itself but also have to be considered in the larger scope of a company's IT processes. A good illustration of this is the quality of the requirements document: from poor requirements, only poor estimates can be produced. More generally, a lower maturity level in IT processes will result in less predictability of the company projects and thus reduce the benefit of producing estimates.

This paper is structured as follows: Section 2 will give some background about the COSMIC method. Section 3 will present our experience in training companies and characterise each of the training sessions while staying anonymous. Section 4 will discuss a number of highlighted issues. In section 5, we summarize some lessons learnt from those training sessions. Section 6 will formulate some recommendations. Finally, section 7 will draw some conclusions and interesting related research work to equip companies with better tools to produce COSMIC-based estimates.

2. The Background: COSMIC Function Points

The COSMIC method, standardized as ISO/IEC 19761:2011, is one of the most recent functional size measurement (FSM) methods[11]. It is becoming more and largely used and sometimes adopted as a national standard. It is adapted to modern software development projects. COSMIC is applicable to software projects from different domains: Business Application, real-time, infrastructure and some scientific/engineering software.

[10] J.-M. Desharnais, A. Abran, *Applying a Functional Measurement Method: Cognitive Issues*, 11th International Workshop on Software Measurement, Montréal, Québec 2001.

[11] ISO: ISO/IEC 19761, Software Engineering – CFF – *A Functional Size Measurement Method 2003*, International Organization for Standardization, Geneva 2003.

The COSMIC method measures the functional size of software projects. It consists in applying a set of rules and models to measure Functional User Requirements (FUR). In COSMIC, a FUR is a subset of the whole user requirements, which describes "WHAT" software should do. In general, FUR is extracted from requirement documents. There is no template on how these documents should be described. Non-functional requirements (the "HOW") like quality requirements are not addressed by the COSMIC method. However, it has been shown that most non-functional requirements evolve to functional requirements when a more detailed requirement analysis is performed[12].

The COSMIC measurement process is based on two models, the "software context model" and the "COSMIC Generic Software Model", followed by a summing process.

The software context model – several elements have to be defined:

- Identify the architectural layers of the software, if any. A layer may contain different "peer" pieces of the software. Each layer should be measured separately.
- Define the purpose and the scope of the measurement.
- Identify the functional users, which can be humans or not.
- Define the level of granularity at which the software will be measured.

The COSMIC Generic Software Model – is applied to FUR once the context of a measurement is well defined, and the reason why it is performed is clear:

- Identify the software boundaries across which functional users interact with the software.
- Identify data groups.
- Identify the functional processes and the triggering events.
- Identify the sub-functional processes, either "data manipulation" or "data movement". Only the latter are considered in the measurement. A data movement is "a base functional component which moves a single data group type". There are four types of data movement: *ENTRY*, which "moves a data group form a functional user across the boundary into the functional process where it is required"; *EXIT*, which "moves a data group from a functional process across the boundary to a functional user that requires it"; *WRITE*, which "moves a data group lying inside a functional process to a persistent

[12] *Guideline on Non-Functional & Project Requirements: How to Consider Non-functional and Project Requirements in Software Project Performance Measurement, Benchmarking and Estimating*, V1., COSMIC, November 2015, http://www.cosmic-sizing.org

storage"; and *READ*, which "moves a data group from the persistent storage into a functional process which requires it"[13].

Functional size computation. The functional size of the software is then measured as the sum of the functional processes sizes. The size of a functional process is the sum of all its data movements.

The COSMIC Measurement Manual explains in detail how COSMIC measurement should be performed. This manual has been updated several times to reflect the evolution of COSMIC[14].

3. Overview of Training Sessions

Most trained organizations were disappointed with the estimation methods and tools they had tried before and were looking for a more suitable method for their projects. Several companies were willing to understand how to use the COSMIC method because of the need to use it, typically in calls for tenders.

3.1. Training Goals

Our training sessions are designed to fit to a given company's needs and goals. But the main training objectives are to:

- Transfer the knowledge of the COSMIC philosophy,
- Understand the concepts and the methodology,
- Practise and apply the COSMIC method,
- Adapt and apply the methodology to the organization projects.

3.2. Training Content

Our training sessions are based both on theory and practice. The theory is enriched by many examples and academic exercises with increasing levels of difficulty, to illustrate the concepts of the method. To support practice, we elaborate customized case studies based on reports and documents from the company

[13] *The COSMIC Functional Size Measurement Method, Version 3.0 – Measurement Manual*, COSMICON 2015, http://www.cosmic-sizing.org; *The COSMIC functional Size Measurement Method, Version 4.0.1 – Measurement Manual*, COSMICON 2015, http://www.cosmic-sizing.org

[14] Ibidem.

of the trainees. Customized case studies have a twofold objective. They help us understand the culture of a company and ensure we speak the same language and that our training is meeting the right requirements of the companies. They are also the shortest way to convince people and encourage them to use the method directly after the training because the trainees see how they can use it in their daily work. These are some principles to motivate people and to ensure success in any project[15].

3.3. Training Audience

We have been delivering COSMIC training for years to companies from different sectors. This covers a variety of:

- Domains: banking, retirement, software companies, space companies, etc.
- Size: small, medium-sized and large companies.
- Maturity level: low, medium or high.

Motivations. Companies have different needs and interests in COSMIC training:

- Some companies are looking for a suitable estimation method and are willing to discover COSMIC and see if it suits their needs and context.
- Some are already convinced by the COSMIC method and are willing to master it and practise it on their projects.
- For others, the use of COSMIC is mandatory and they are willing to learn COSMIC and apply it directly (often in calls for tenders).

Table 1 illustrates the different profiles of the trained companies. We can see that the trained roles were typically project managers (PM), analysts (AN) and sometimes also specific developers (DEV).

The first company specializes in tax software. They mainly work with the Directorate-General (DG) of the European Commission (EC). They regularly submit proposals for calls for tenders with this DG in which they are asked to use the COSMIC method to justify their estimations. So the objectives of the trainees were to learn how to effectively practise the method so that they can have a competitive advantage over their contenders in the call. All the trainees were project managers.

[15] E. Verzuh, *The Fast Forward MBA in Project Management: A Practical Handbook and Reference*, 4th ed., John Wiley and Sons, 2012; T. Schmidt, *Strategic Project Management Made Simple: Practical Tools for Leaders and Teams*, John Wiley and Sons, 2009; P. Lencioni, *The Advantage: Why Organisational Health Trumps Everything Else in Business*, Jossy-Bass, 2012.

Table 1. Characteristics of our Industrial Training

Year	Company Domain	COSMIC Version	Comp size	Aud. size	Roles	#days	Maturity level	Training Objective
2009	Tax Admin.	V2.2	Big	12	PM	2j	Medium	Learn to use directly
2010	Retirement Pub. Sector	V3.0	Med	15	PM	1j	Medium	Learn to use directly
2010	Banking Sector	V3.0.1	Med	10	PM AN	2j	Medium	Introduction
2012	Medical Systems	V3.0.1	Small	10	PM AN, DEV	1j	Very low	Introduction
2012	Banking Sector	V3.1	Big	15	PM AN	2j	High	Learn and practise
2013	Banking/ Space/etc.	V3.1	Big	10	PM AN	2j	Low	Learn and practise
2014	IT Devpt	V4.0	Small	8	PM AN DEV	1j	Low	Learn to use directly

Source: the authors' own work.

The second company specializes in legal retirement systems and works regularly in the public sector. Their objective in attending the training was to get expertise in estimation methods to use in their proposals for projects. They used generally expert judgment estimations but began to see the limits of this kind of estimations and were willing to use a more objective method to help them compare and learn from their projects. To be sure to get the best from the COSMIC method, individual coaching sessions were organized when necessary.

The third one was interested in improving its quality process and willing to standardize the measurement process.

The fourth company specializes in delivering software for healthcare systems. Their objective was mainly to understand the basics of the COSMIC method and check whether it fits their needs.

Like the first one, the fifth company often submits proposals for calls for tenders and also works in a regular manner with the EC. Prior to the training, we had achieved several estimations on some of their critical projects. The company faced two challenges. Firstly, they have offshore departments in Europe and outside Europe. The lack of a standardized estimation tool makes it very difficult to compare projects from the different entities. This costs a lot of time and a lot of money. Secondly, the EC imposed a standardized way of estimating projects, which was not easy to use. So they decided to use internally a standardized estimation method, COSMIC. On the one hand, the standard will be the basis for projects from the different divisions to enhance communication. On the other hand, it will be used as the first level of estimation in projects with the EC before applying the EC estimation method (a non-standard one). We were asked to deliver the same training to offshore divisions, too. The company wanted to master COSMIC more thoroughly when they had decided to evolve from CMMI (Capability Maturity Model Integration) level two to three. In this context, their need was to implement a complete estimation process and we helped them elaborate a proof of the concept based on COSMIC: a repository to gather projects statistics to elaborate a historical database, an estimation tool using COSMIC and translating results in the mandated format.

The sixth company is active in different sectors and was interested in learning and testing the applicability of the COSMIC method in their complex environment.

The last company mainly works with Agile methodologies. They are aware of the advantage of using a standardized estimation method and they were searching for a method adapted to their needs.

In terms of the application domain, table 1 shows that we mainly dealt with companies from the Business Application domain. This was not a choice but just a fact. In near future, we are conducting a survey to understand the estimation culture in Belgian companies and to understand why there was no interest from real time and embedded software companies although COSMIC is known to be very effective in these fields.

Apart from the fifth company, no one gathers internal historical data and then has no view on how they are really performing.

3.4. Training Organisation

Our training sessions are systematically adapted to fit to the last version of COSMIC. Over time, we have evolved from version 2.2 to 3, and currently 4. This gives us indirectly an opportunity to effectively evaluate the improvement made from one version of COSMIC to another. Training sessions are organised first in group sessions and afterwards, in more individual sessions.

Group sessions. Training sessions are organised for a maximum of 8 people and generally during two days. The training is composed of theory and practice.

The theory addresses COSMIC concepts and is illustrated by many simple and academic examples and general examples. The goal is to understand the fundamental concepts and to get the first flavour of the method and its applications.

In the practice part, we go in depth in the application of the COSMIC method. We use real case studies extracted from the projects of the trained company. We develop these case studies based on documents and reports from the company. In some complex contexts, we might perform the first estimation of a project to understand the context before developing the case studies.

The use of real case studies has a twofold advantage. It allows us to speak the same language of the trainees and to easily communicate with them. It is also a strong motivation tool for trainees. They see clearly how they can use the method in their day-to-day work. They are convinced they are not just losing their time on another and irrelevant training session.

Individual sessions. The objective of individual sessions is to master the use of the COSMIC method. It allows going deeper than during group sessions. We coach a trainee on his or her on-going projects. He or she practises directly what he/she has learnt and we discuss and resolve issues as they come. The duration of an individual session is in general half a day. We may organise as many sessions as needed by the trainee.

4. Highlighted Issues

In this section, we detail the issues of different nature and analyse them in order to enhance our training. We have used the following techniques to identify them:

- Direct feedback from trainees during the sessions.
- Feedback collected using satisfaction forms.
- Internal retrospectives on the training.

4.1. Understanding Software Functional Size

Most of the trained companies had tried some different estimation methods or tools (often expert-based and relating to lines of code). Although they were aware they needed a different kind of measurement approach, it was surprisingly difficult to get trainees to understand why measuring functionality gives a reliable estimate of the future software size. Understanding functional size seems to be not so concrete to understand it easily, either. The tendency is to think “how big will the software be” in terms of lines of code and to say “Oh! This is not possible, my experience tells me that it should be bigger than this”. This is mainly the reaction from developers' roles. They are much focused on technical aspects and rapidly think about low levels of data, how complex the database will be and so on. The problem is also more specific to small companies where persons play different roles in the same project (the project manager and analyst and developer). In general, there is no real project management methodology and the planned tasks and mostly development tasks.

4.2. Non-Functional Requirements (NFRs)

COSMIC is about measuring the functional size of software and the method is related to functional requirements explaining what the software shall do. Non-functional requirements are technical, quality or system requirements. The importance of NFRs in some projects makes it difficult to agree that measuring only the functional size is a good estimate for a project. One way we used to resolve this problem was to show that many NFRs considered (e.g. security), can be described as functionalities and accounted for in the estimation. Other NFRs (e.g. technology constraints) can be dealt with in the later process of translating function points into man-days and do not affect the functional size.

This issue was highly discussed during the 2014 COSMIC Master Class[16]. A smart analysis should be conducted before estimation because many of the requirements considered first as NFRs will evolve to real functional requirements. This step is also a problem for lower maturity level companies and for small companies more focused on delivering software. The issue of NFRs is one of the improvements in COSMIC version 4.0[17]. A new section explains the difference between functional and non-functional requirements and how to deal with them, and currently a guideline is dedicated to this issue[18].

4.3. Level of Granularity

Defining a standard level of granularity is important to make things comparable. The level of granularity refers to the level of refinements of the requirements.

[16] A. Abran et al.: *COSMIC Master Class*, Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM–MENSURA), Rotterdam, Netherlands, October 2014.

[17] *The COSMIC Functional Size Measurement Method, Version 4.0.1 – Measurement Manual*, COSMICON 2015, http://www.cosmic-sizing.org

[18] *Guideline on Non-Functional & Project Requirements: How to Consider Non-Functional and Project Requirements in Software Project Performance Measurement, Benchmarking and Estimating, V1*, COSMIC, November 2015, http://www.cosmic-sizing.org

As projects go on, more details on the functional requirements are revealed within the same scope. For COSMIC measurement, there is only one standard level of granularity which is "the level of granularity at which individual functional processes and their data movements can be identified and defined"[19].

Identifying the right level of granularity at which measurement should be taken is also difficult to capture. It is also a requirement analysis issue and the same remarks as the two first points can be made.

4.4. Processes and Sub-processes

The basis of a measurement is the identification of functional processes in FUR to be measured. The description of the functional processes and their identification was highly improved in version 4.0 of COSMIC with many examples and illustrations to ease the understanding of this important concept.

Each functional process is composed of sub-functional processes of two types, "data manipulation" and "data movement". Only data movements are counted in the measurement process, data manipulations are considered as being counted in the data movement type with which they are associated[20]. This is possible due to the fact that for non-algorithmic software, the number of manipulation in data is very small regarding the number of data movements, so it can be in a way "neglected" and considered as being counted in the associated data movement.

It is the case especially in the banking field that we noticed the difficulty in accepting that way of measuring. This is most likely related to the large number of transactions and calculi.

4.5. Data Groups and Data Attributes

This issue is related to the above matter of processes and sub-processes. As we said before, in COSMIC, we count the number of data movements of data groups. A data group is composed of several attributes depending on the data analysis performed in a project. This way of doing is not so intuitive. Some

[19] A. Abran, *Software Estimation: Transforming Dust Into Pots of Gold?*, Proc Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM–MENSURA), Rotterdam, Netherlands, October 2014; *The COSMIC Functional Size Measurement Method, Version 4.0.1 – Measurement Manual*, COSMICON 2015, http://www.cosmic-sizing.org

[20] *The COSMIC Functional Size Measurement Method, Version 4.0.1 – Measurement Manual*, COSMICON 2015, http://www.cosmic-sizing.org

trainees find it difficult to think in terms of moving data groups instead of moving separately data attributes. Some of the recurrent remarks are "with attributes, the count should be correct, because in my experience…". Here again, we noticed some psychological resistance. This behaviour is also related to the role of trainees more present in technical jobs like developers.

4.6. Development Effort Estimations and Productivity

When measuring with COSMIC, most companies are interested in getting the development effort estimation in Man-Month (MM). COSMIC gives a measurement in the CFP (COSMIC Function Point) instead. From this number, we can use historical data or some international benchmarking repository, like the I.S.B.S.G.[21] to get the estimation in MM. But this is not so easy to obtain for different reasons and there is no standard way to do so.

The strength of COSMIC is to offer a standard way to measure and compare projects. But, once we have the CFP measure, there is no standard way to refine it into MM estimates. When a company has its own historical data, it is easy to build a measurement process. This helps compare projects effectively and develop a productivity analysis. But most companies, especially small and medium-sized ones, (SMEs) lack historical data.

This issue can be resolved by the use of an international repository like the I.S.B. S.G. This repository offers now a database of COSMIC measured projects. But still, it is difficult for some companies to structure the use of the repository.

4.7. Influence of the COSMIC Version

We noticed a big change from version 2.2 to version 3.0 and a real improvement. This is especially the case with the concept of functional user. In version 2.2, sizing was different depending if we measured from the end user's or the development user's point of view. This made comparisons harder to explain to a manager if a project was sized from these two different points of view. From version 3.0 on, the two points of view were merged and the concept became easier to learn and practise.

[21] P. Hill, *Practical Software Project Estimation – A Toolkit for Estimating Software Development Effort & Duration*, "MH Professional" 2010; R. Darimont, C. Ponsard, *Supporting Quantitative Assessment of Requirements in Goal Orientation*, 23rd IEEE International Requirements Engineering Conference (RE'15), Ottawa, Canada, August 2015.

In general, the latest versions, especially version 4.0, are much easier to understand and use. Explanations of data manipulation were very light in older COSMIC versions. In version 4, these explanations were highly improved and a new dedicated section created. A new exception in the counting was also introduced; all data manipulation should be ignored "EXCEPT if there is a FUR that must be measured for a change for data manipulation"[22]. Manuals on specific applications of COSMIC (Business Application, SOA, real-time etc.) were also improved. The guidelines[23] to use COSMIC in agile projects were especially appreciated as more and more projects use such methodologies.

4.8. Summary of the Highlighted Issues

The different highlighted issues can be summarised in several categories.

COSMIC related issues: with a focus on COSMIC concepts and the difficulty in understanding them. As we said, the definitions were highly enhanced through the different versions of COSMIC. However, the COSMIC method is still a complex method. This complexity is related to the complexity of requirement analysis tasks.

Requirements engineering issues. Many of the misunderstandings in the use of COSMIC concepts come from a bad requirements engineering task. Companies encountering this issue often lack a structured or a standard approach to analyse requirements, and have a low to very low maturity in requirements software practices in general.

Project management issues. Many of such issues are discussed in Abran's book[24]. Several identified issues relate to the lack of a structured approach to project management:

- Difficulty to link functional size estimation to productivity issues: team/project productivity.
- Difficulty in how to go from functional size to effort and duration numbers.
- The tendency to add function points because "we feel it should be like that" and to go back unconsciously to a "judgment expert estimate". Habits are hard to change!

[22] *The COSMIC Functional Size Measurement Method, Version 4.0.1 – Measurement Manual*, COSMICON 2015, http://www.cosmic-sizing.org

[23] *Guidelines to the Use of COSMIC FSM to Manage Agile Projects*, V1.0, COSMIC 2011, http://www.cosmic-sizing.org

[24] A. Abran, *Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers*, IEEE Computer Society, John Willey and Sons, 2015.

- A lack of a high level or helicopter view of the project being measured and then a difficulty in analysing it and identifying all the important elements for the COSMIC measurement.
- A poor assignment of roles to measurement tasks. Technical roles may not have enough distance to achieve effectively measurement tasks.

Psychological resistance. Resistance to change is one of the most recurrent factors on understanding and using COSMIC effectively. Trainees unconsciously have a tendency to use personal judgments to justify their results with COSMIC.

5. Lessons Learnt from COSMIC Training

We can identify a clear correlation between the highlighted issues and the maturity level of a given company. A good COSMIC measurement is highly dependent on the quality of requirements analysis. The use of a structured requirements analysis method/template and best practices is a key element to achieve a reliable measurement. This also impacts the time needed to take the measurement.

We think that project management has also a strong impact on the way estimations are understood and made. The lack of a PM methodology or at least a structured way of doing things prevents from thinking out of the box and from gathering the right elements for taking reliable measurements.

It is not surprising that most project failures are due to bad requirements and bad estimates as it was shown in many surveys[25]. It, therefore, seems natural that companies with low maturity, in particular in requirements analyses, find it more difficult to get used to implementing the method.

In trying to understand why it is so difficult to understand functional size, we discussed with the trainees requirements analysis tasks and software development methodologies. We observed that most of the time, the problem is not related to the size of the company but mostly to its maturity to deal with those processes. We noticed that more mature companies have defined processes and standardized analysis methods. They often use specification templates that help them structure their work but also their way of thinking and analysing the

[25] J.-M. Desharnais, A. Abran, *Applying a Functional Measurement Method: Cognitive Issues*, 11th International Workshop on Software Measurement, Montréal, Québec 2001; T. Menzies, Z. Chen, J. Hihn, K. Lum, *Selecting Best Practices for Effort Estimation*, “The IEEE Transactions on Software Engineering”, November 2006, vol. 32, no. 11.

requirements. Smaller companies have often more problems related to this, as confirmed by some studies on small company practices and maturity levels based on the lightweight ISO29110 software development support[26].

6. Recommendations on How to Address Issues

Training on the COSMIC method is a complex task, which is closely related to requirements engineering and project management, and has its psychological aspects. A reliable and effective training session should address all these elements. That is why we re-design our training process in the following way:

1. First, enhance the maturity level of a given company.
2. Integrate requirements engineering and project management aspects in the training.
3. Define a long term and global training process.
4. Address psychological aspects to avoid change resistance and frustration.

Based on the feedback, we have redesigned our training to adapt to each case and to prevent frustration and resistance during the training. We formulate them here as useful recommendations to follow.

Define a global and long term training process: we considered different and successive levels of training: basic, intermediate and advanced, with enough time in between (from three to six months). The rationale is twofold: firstly, it takes time to assimilate the training in the real practice and secondly, the company may also take time to adapt some of its processes to reach a suitable level of maturity (see the next point). Using coaching can help in speeding up the process.

Match each training step with the current maturity level: we have designed the content of the training to fit to the maturity levels of a given company. To quickly assess the maturity level of the company, we use a questionnaire inspired from the lightweight ISO29110 standard[27]. We distinguish the flowing levels of training:

[26] ISO: ISO/IEC: 29110:2011, *Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs)*, International Organization for Standardization, Geneva 2011; C. Ponsard, A. Majchrowski, J. Flamand, *Assessing and Driving Software Development Practices in SMEs through an Online ISO29110-based Survey*, 25th International Conference on Software and System Engineering and their Applications, Telecom Paris-Tech, Paris, May 2015.

[27] ISO: ISO/IEC: 29110:2011, *Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs)*, International Organization for Standardization, Geneva 2011.

- The *basic level* addresses two points: a) acquiring fundamentals of requirements and project management focusing on why it is so important to estimate and (b) learning the principles of the methodology illustrated by simple exercises. The objective is to raise awareness and understanding of the concepts and methodology. This mainly targets lower maturity companies lacking structured methods.
- The *intermediate level* addresses more complicated cases. It goes forward on theory to visit the concepts more in depth and to focus on special cases (e.g. more sophisticated messages). The trainees are asked to work in groups and then the results are confronted and discussed. The objective is that they are able to take measurements by themselves in most cases. This level is suitable for companies with some maturity. They are conscious about the need for standardized or structured approaches and often use some in-house methods.
- The *advanced level* is for people willing to get deep understanding of theory and practice and to be able to use effectively the method in all situations. They often use some estimation tools already and know exactly about their problems and their needs. The training focuses on real complex case studies addressing complex situations. It is mostly for more mature companies willing to go fast and forward on their knowledge.

Improve the maturity level of a given company in software development: when some lack of maturity is detected (e.g. requirements and/or project management), it is important to engage the company in software process improvement (SPI) to build on solid grounds and make sure all key concepts are well understood in the same manner by the whole audience. It is quite common to combine training/coaching sessions in requirements writing and effort estimation. We give some examples and do some exercises to be sure they get the picture. We focus on requirements analysis best practices and also on project management and the general context of estimations. To keep the SPI dimension light, we recommend using specific tools proposed by ISO29110. Because it was developed for small entities developing, it really focuses on the core practices and it also proposes a deployment toolbox including requirements and project management templates for supporting related activities. ISO29110 is also structured in successive maturity levels based on a set of profiles that range from an “entry” profile to the most “advanced” profile. The second level, called the basic profile, is a good minimal target for starting to seriously consider effort estimation.

Make sure people feel engaged in the same way. If during the COSMIC training there is some form of resistance, it is important to identify the cause (a specific role, problems with some background or other reasons) and to address it adequately, for example, by going back to some key concepts and using examples (especially company examples) that can ease doing the matching. Communicating and explaining the reason for each decision or choice, and how it relates to the company's culture, is a strong motivating factor.

7. Conclusion and Future Work

In this paper, we reported what we have learned from our industrial experience in performing COSMIC training. We were able to relate it to some extent with explanatory factors mainly linked to human factors, requirements or project management issues.

We also highlighted the valuable improvements brought by version 4 of COSMIC in describing the concepts of the COSMIC method. It was illustrated with many examples and use cases that ease the understanding of the concepts, in different fields. Based on this new version, we are developing some templates and tools for effective training in COSMIC. Those should guide and help in the rapid design of high quality training content.

In order to support the adoption of the method, it is also important to provide efficient tools, going beyond the support for counting function points. Starting from the highlighted point that there is no way for a good estimation without a good requirements analysis, we think it is a good line of research to integrate those activities more, especially using model-based and goal-oriented requirements methodologies[28]. Such methods allow an analyst to refine high level goals (or business processes) to concrete requirements and connect them with specific processes and information models. Some ideas have been prototyped with the i* method[29]. We are currently working on extending them using a quantitative reasoning framework taken from the KAOS method[30]. Jointly building the requirements and estimation analysis model will not only help automate the counting process, but it will also force the use of a standardized approach

[28] A. van Lamsweerde, *Goal-Oriented Requirements Engineering: A Guided Tour*, 5th IEEE International Symposium on Requirements Engineering 2001.

[29] G. Grau, *Adapting the COSMIC Method for Evaluating the Functional Size in PRiM*, IWSM/Mensura, Palma de Mallorca, Spain 2007.

[30] R. Darimont, C. Ponsard, *Supporting Quantitative Assessment of Requirements in Goal Orientation*, 23rd IEEE International Requirements Engineering Conference (RE'15), Ottawa, Canada, August 2015.

for requirements analyses and improve the requirements quality. This will also improve the reliability of effort estimates by reducing the uncertainty in the rest of the development process.

8. Acknowledgements

This work was partly funded by the WALLISTIC3 project from the Walloon Region (DGO6). We warmly thank the companies trained over the years for their valuable feedback about their whole training process.

References

Abran A., *Software Estimation: Transforming Dust Into Pots of Gold?*, Proc Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM–MENSURA), Rotterdam, Netherlands, October 2014.

Abran A., *Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers*, IEEE Computer Society, John Willey and Sons, 2015.

Abran A. et al.: *COSMIC Master Class*, Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM–MENSURA), Rotterdam, Netherlands, October 2014.

The COSMIC Functional Size Measurement Method, Version 3.0 – Measurement Manual, COSMICON 2015, http://www.cosmic-sizing.org

The COSMIC Functional Size Measurement Method, Version 4.0.1 – Measurement Manual, COSMICON 2015, http://www.cosmic-sizing.org

Darimont R., Ponsard C., *Supporting Quantitative Assessment of Requirements in Goal Orientation*, 23rd IEEE International Requirements Engineering Conference (RE'15), Ottawa, Canada, August 2015.

Desharnais J.-M., Abran A., *Applying a Functional Measurement Method: Cognitive Issues*, 11th International Workshop on Software Measurement, Montréal, Québec 2001.

Grau G., *Adapting the COSMIC Method for Evaluating the Functional Size in PRiM*, IWSM/Mensura, Palma de Mallorca, Spain 2007.

Guidelines to the Use of COSMIC FSM to Manage Agile Projects, V1.0, COSMIC 2011, http://www.cosmic-sizing.org

Guideline on Non-Functional & Project Requirements: How to Consider Non-Functional and Project Requirements in Software Project Performance Measurement, Benchmarking and Estimating, V1.,COSMIC, November 2015, http://www.cosmic-sizing.org

Hemant K., *Why IT Projects Really Fail: Over-optimism and Complexity Are Just Some of the Many Reasons Why IT Projects Continue to Suffer*, "CIO", December 2013.

Hill P., *Practical Software Project Estimation – A Toolkit for Estimating Software Development Effort & Duration*, "MH Professional" 2010.

The International Software Benchmarking Standards Group, Development & Enhancement Repository, release 13, ISBSG, 2015, http://www.isbsg.org

ISO: ISO/IEC 19761, *Software Engineering* – COSMIC-FFP – *A Functional Size Measurement Method*, International Organization for Standardization, Geneva 2003.

ISO: ISO/IEC 29110:2011, *Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs)*, International Organization for Standardization, Geneva 2011.

Khatibi W., Jawawi D.N.A., *Software Estimation Methods. A Review*, "Journal of Emerging Trends in Computing and Information Sciences" 2010, vol. 2, no. 1, pp. 21–28.

Lamsweerde A. van, *Goal-Oriented Requirements Engineering: A Guided Tour*, 5th IEEE International Symposium on Requirements Engineering 2001.

Lencioni P., *The Advantage: Why Organisational Health Trumps Everything Else in Business*, Jossy-Bass, 2012.

Menzies T., Chen Z., Hihn J., Lum K., *Selecting Best Practices for Effort Estimation*, "The IEEE Transactions on Software Engineering", November 2006, vol. 32, no. 11.

Molloken K., Jorgensen M., *A Review of Survey on Software Effort Estimation, Empirical Software Engineering*, ISESE 2003.

Ponsard C., Majchrowski A., Flamand J., *Assessing and Driving Software Development Practices in SMEs through an Online ISO29110-based Survey*, 25th International Conference on Software and System Engineering and their Applications, Telecom Paris-Tech, Paris, Mai 2015.

Schmidt T., *Strategic Project Management Made Simple: Practical Tools for Leaders and Teams*, John Wiley and Sons, 2009.

Total Metics: Methods for Software Sizing – How to Decide Which Method to Use, Version 1.1, August 2007.

Verzuh E., *The Fast Forward MBA in Project Management: A Practical Handbook and Reference*, 4th ed., John Wiley and Sons, 2012.