Early Project Estimations

White Paper

Project:	White Paper
Document–ID:	EarlyProjectEstimations.docx
Version:	V1.0-18
Issue of:	16. April 2013
Replaces:	--
Author:	Th. Fehlmann
Confidentiality:	Publication
Distribution:	

Carbon Copy:

Table of Contents

Chapter 1: Management Summary

1.1 Why is Software Project Estimation so Difficult?

1.1.1 Software Project Cost Prediction for Software Projects

Project estimation is a difficult task. It requires predicting the future, and this is much more difficult than explaining the past.

In the past, project managers used Gantt charts to predict cost. Today's lean and agile software developers have been accustomed to use various kind of Delphi techniques, to assign Story Points to developing tasks, and never commit to anything except to the next sprint.

Nevertheless, software must be created and ICT has become essential for all but very few organizations. Thus, getting reliable predictions what software will cost at the end is mission-critical.

1.1.2 A New Tool—The Buglione-Trudel matrix

The *Buglione-Trudel* matrix combines functional and non-functional aspects of software. This allows predicting software cost well before detail requirements are known. It combines functional size of software, recorded in the lower part of the matrix, with a draft design of the expected software contributions to business drivers in the upper part. Such quality issues account for the waste majority of change requests that make software projects so difficult to predict.

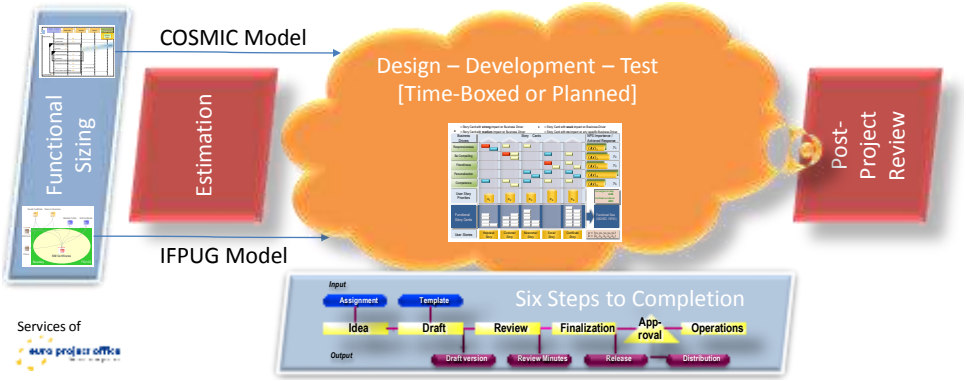

Figure 1: Project Office Services in an ICT project

Now it is possible to predict how much such requests must be expected in an ICT project, be it software development or operations.

1.1.3 The Role of the Project Office

Every complex project needs an independent 3rd party project office that does sizing and estimates, tracks progress and reviews lessons learned, to complement in-house estimates and help project teams to keep on track.

1.2 Functional Size – The Inception of an ICT Project

Functional Size is the most important cost driver, in any case, be it mobile apps or large, networked IT all cost predictions start with a functional size count. There are international standards, ISO/IEC 20926 and 19761, which define how to measure functional size.

However, although functional size is easily measurable, quality requirements and non-functional requirements are not, and they contribute quite a bit as cost drivers. Thus it is necessary to complement functional size with other project parameters when doing estimates.

1.2.1 Macro Estimations

Macro estimations try to make a comparison, based on functional size, with other, similar projects, either by benchmarking or by using parameterized cost drivers. This gives a ballpoint number for cost, however there are many unknowns with cost drivers, e.g., how many skilled people will be available for the chosen technology, with likeliness and upper and lower bound for cost.

1.2.2 Micro Estimations

Thus, macro estimations are always complemented by micro estimations based on a work breakdown structure, identifying *what* needs to be done, even while acknowledging it is yet unknown, *how*. They go with five steps:

- Voice of the Customer—understanding the priorities of business drivers for the customer
- Work breakdown based on the functional size measurement — the cellar in the Buglione-Trudel matrix
- Risk Assessment, identification of risk mitigation efforts and adding risk contingency to the budget
- Quality Planning, adding the work needed to meet business needs, the upper part of the matrix
- Test Planning and balancing quality expectations within available budget

1.2.3 Benefits

Project estimations can be done before evaluating suppliers. Thus software evaluation and acquisition are based on a realistic budget, and those that offer too high or too low are easily identified.

Chapter 2: Early Project Estimations

2.1 Estimating Software Projects

Software engineering is not civil engineering, where you first create a plan then execute the plan, and all you need to do is making sure the plan takes all eventualities into due consideration. Software has to explain complex tasks in a language simple enough such that ICT systems are able to understand and execute it correctly. It's a translation process starting with humans, some actual processes, some explicit and many more implicit requirements, involves social behaviour, organizational capability maturity, ability to communicate, to formulate in different industry-specific languages, of keeping trust and continual engagement that eventually ends in an integrated men-machine system creating value. It is complex but it can be mastered using adequate tools.

2.1.1 Software Sizing

Cost of projects depends on its cost drivers. In software projects, a prominent cost driver is size of software. Functional size is measurable by different international standards, among them ISO/IEC 20926 (IFPUG) and ISO/IEC 19761 (COSMIC). However, sizing requires a few things to know, among them

- Purpose of the project and its business goals;
- Actors, stakeholders and involved organizations and data;
- Environment and interfaces to other ICT systems.

2.1.2 Requirements

Detailed requirements are not needed for sizing. In contrary, since size drives cost, the project estimations come prior to detailed requirements; they likely will depend upon the budget available. Before doing a detailed requirement analysis, sizing the project avoids spending effort on fancy dreams.

2.1.3 Sizing Method COSMIC — the Common Software Measurement International Consortium

Sizing methods are based on a model of the intended software. While IFPUG focuses on input, output, queries, and data stores and external interfaces, COSMIC focuses on data movements. The two methods are complementary, not competitive.

Sometimes, both methods are applied, since both capture a different aspect of the intended software. Both offer easily comprehensible visualizations to software purchasers, facilitating requirements elicitation in later project steps.

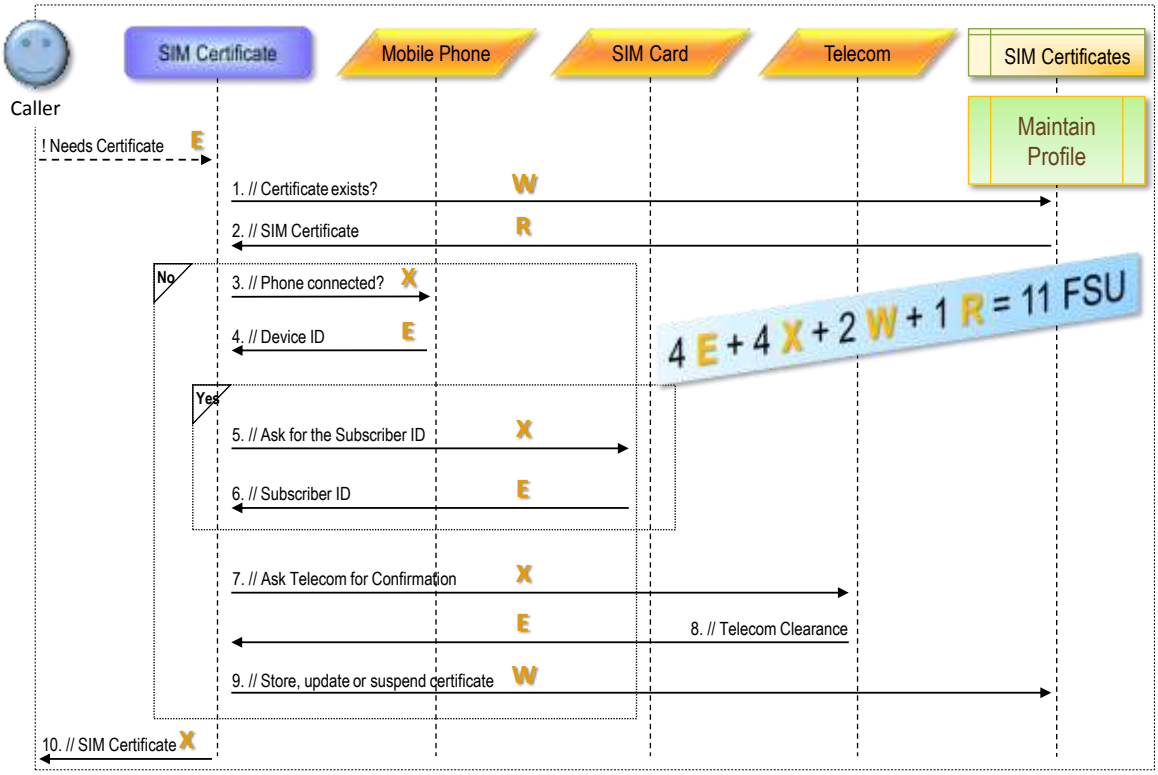

Figure 2: Sample COSMIC Count visualization for a Telecom application

COSMIC sizing relies on counting the number of data movements between objects of interest, and thus is most welcomed by developers. IFPUG rather looks at user interfaces and reports; business people find it easier as a start for requirements elicitation.

Developers find it easy to understand the architecture with COSMIC because they see the data movements. The COSMIC count is therefore the method of choice to create an engineering development plan, or identify the backlog in an Agile development environment.

Scott W. Ambler recommends modeling the flow of logic within a system in a visual manner using UML sequence diagrams, enabling both to document and validate the logic, and identifying the behavior within the system. Sequence diagramming is commonly used for both analysis and design purposes.

The COSMIC count is derived from sequence diagrams almost for free.

Figure 2 shows a sample sequence diagram for an application that creates and maintains a certificate in the user profile based on information collected from the *Subscriber Identity Module* (SIM) card in his mobile phone.

2.1.4 Sizing Method IFPUG — International Function Points User Group

The same project sized with IFPUG looks different; but sizing is equivalent. By comparing effort needed for implementing one Functional Size Unit (FSU) respectively, it takes 1 COSMIC FSU ≅ 3 IFPUG FSU, approximately.

Business analysts understand well what happens within this model of *Business Transactions*:

- EI = User or Device Input
- EO = Report, Output, Data to another application
- EQ = Data Query (not present here)
- ILF = Internal Data Store managed by application
- ELF = External Service of another application

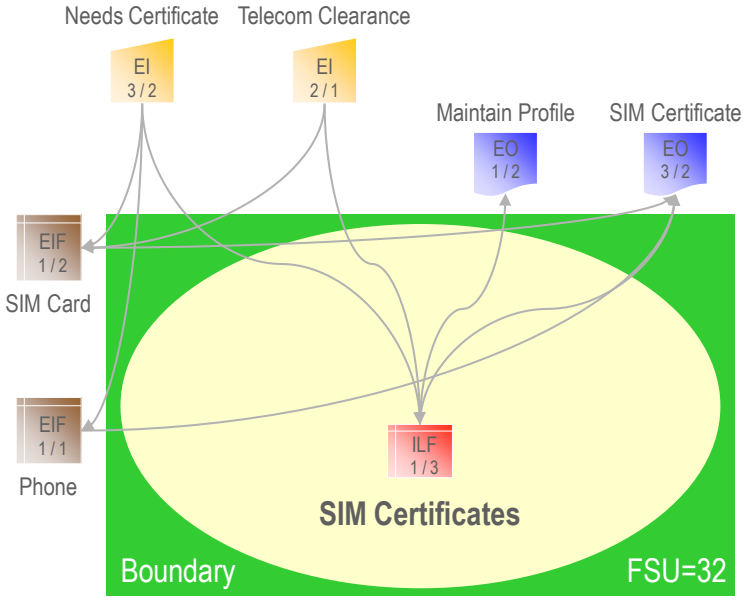

Figure 3: Sample IFPUG Count showing the same Telecom application as before

IFPUG sizing requires less design information than COSMIC but also is less useful and planning the approach with software engineers. Nevertheless, an IFPUG count also constitutes a work breakdown structure for a project.

2.2 Macro and Micro Estimation

2.2.1 The Role of Functional Sizing

With the functional size (FSU), efforts can be predicted based on the ISBSG benchmarking database, collected from over 6000 ICT projects worldwide. This database is available as a web service through SwissICT, member of the International Software Benchmarking Standards Group (ISBSG). Macro estimates are

comparisons with other ICT projects that typically were done following best-in-class development and effort recording processes.

Micro estimation is different. It involves breaking down the project into work items, adding provisions for risks, bug fixes, and change requests. This involves many guesses, e.g., how many bug fixes and change requests must be expected.

Both macro and micro estimation are based in functional size; however, the functionality doesn't tell how much must be done to deliver the quality expected by the customer. Thus, micro estimation is a more complicated process. In either case, estimation should be done by a trusted third party.

2.2.2 Parameterization

Both macro and micro estimation needs significant skills investments. However, there is an easier way of determining the impact of cost factors. It is called *Parameterization*.

Euro Project Office offers tools that do macro estimations, taking other cost drivers into account based on domain knowledge base parameters, namely the Galorath SEER suite of estimation tools. These tools also use ISBSG's benchmarking database, but they offer additionally a series of parameters allowing to gauge cost drivers as needed.

Galorath SEER places macro estimations and benchmarks at everyone's fingertips; needing an investment in tool training only. Organizations that often evaluate software acquisition benefit most from Galorath SEER before committing to the supplier's solution.

2.3 Micro Estimation Steps

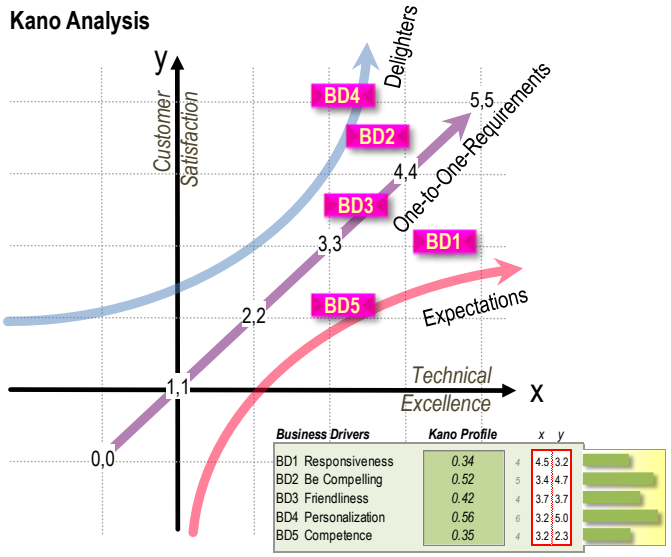

Figure 4: Kano Analysis of Business Drivers for sample Telecom application

2.3.1 Step 1: Voice of the Customer for profiling Business Drivers

The most prominent cost drivers are the customer's values and business needs. For estimating projects, It is necessary to identify the business drivers that create value for the customer.

The simplest method available is Kano analysis. Business drivers are classified into delighters, one-to-one requirements, and expectations. It allows getting an importance profile of the business drivers to identify quality requirements and non-functionality that later will pop up during the project.

An even better method is to derive such profiles from Net Promoter® Surveys.

2.3.2 Step 2: Plan the Deliverables

Deliverables are identified by functional sizing and the business needs profile. Some of them have impact on business drivers, some have no specific impact because they provide must-be functionality. These deliverables are written into the Buglione-Trudel matrix, a data structure visualization combining functional and non-functional work items, see Figure 5.

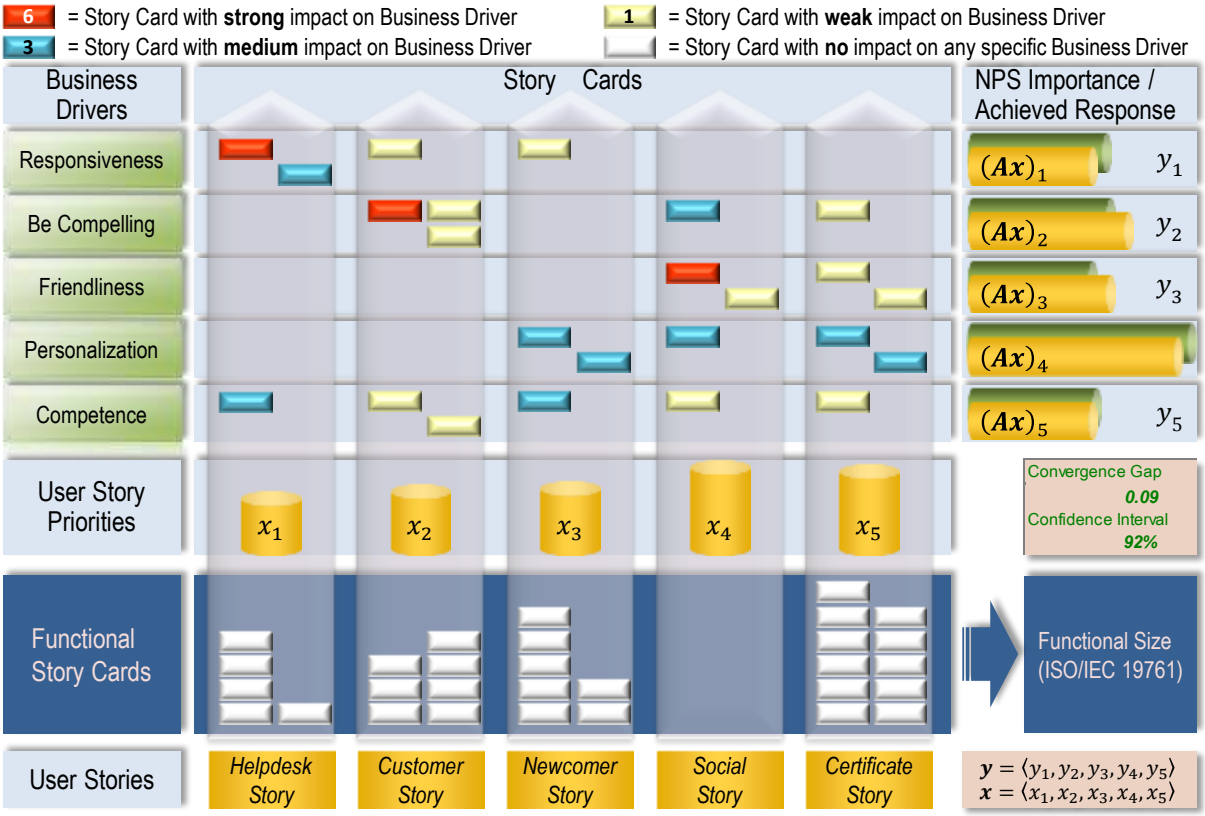

Figure 5: This picture shows a completed Buglione-Trudel Matrix for an Agile project

The functional work items are recorded in the cellar of the matrix, the non-functional items that specifically impact the business drivers of the customer are entered on the respective row of the matrix.

Thus it becomes easily and visually perceivable, which business drivers are supported and which need additional effort. Again, it is not necessary to know exactly what needs to be done, only how much effort is likely to be expected.

2.3.3 Step 3: Risk Assessment

Now identify the risks and unknowns that must be addressed. Some risks such as using unknown technology can be addressed and mitigated; this adds work items both to cellar and rows of the matrix.

A risk assessment is part of any estimation.

2.3.4 Step 4: Plan for Quality

There are still gaps to meet customer's business driver profile. During the project these gaps will be filled by change requests. For the estimation, we fill them out now, indicating what needs to be addressed.

The convergence gap — the gap between business driver profile and planned project achievements — is an indicator telling how far the project plan is from meeting customer's needs.

2.3.5 Step 5: Add Test Plan; Complete the Estimate

There remains test effort, project administration, and project management to be added to the estimate, plus other tasks such as progress tracking, internal communication and project marketing, as needed.

The convergence gap keeps telling you whether your planned effort is realistic. Add work items only where necessary; remove all those over-performing.

2.4 The Details behind the Matrix

2.4.1 Functional Work Items

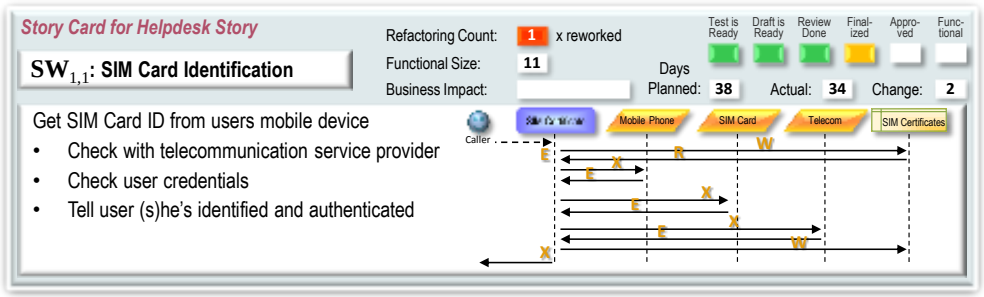

Figure 6: Sample Functional Work Item

A *Functional Work Item* describes technical features that have no specific impact on any of the business drivers. It must work as specified; it might include

a number of difficult interfaces. That in turn might require difficult meetings with stakeholders, and thus induce cost. In Kano, it refers to an expectation.

The signals on top right indicate the status of the deliverable according the Six Steps to Completion method, for tracking the status of the project. Thus, the Buglione-Trudel matrix serves also as a project or sprint backlog. Days planned might stem from an ISBSG benchmark.

The sample work item shown above (Figure 6) corresponds to Figure 2 and will be one of the work items in the cellar of the first pillar in Figure 5.

2.4.2 Non-Functional Work Items

A Non-*Functional Work Item* describes features that have specific impact on one or more of the business drivers. It is crucial for quality but might include no or only little functionality. In the Kano diagram, it typically refers to a delighter.

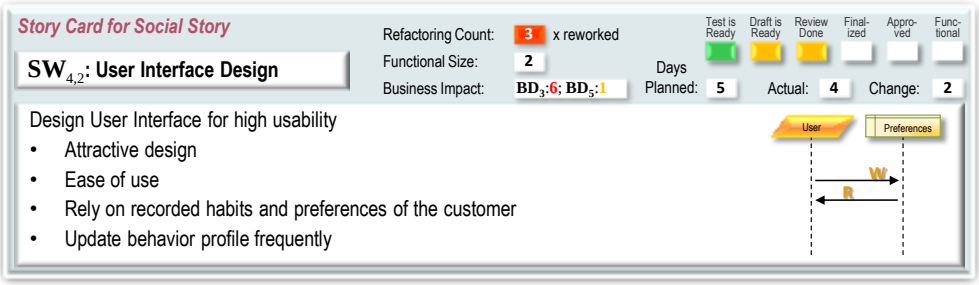

Figure 7: Sample Functional Work Item

The work needed is related to ease of use and has no particular functionality, except the ability to remember user preferences. Days planned are expert estimates depending from the amount of impact needed on the business driver.

The sample work items shown above (Figure 6) corresponds to Figure 2 and will be one of the (red) work items in the third upper row in fourth pillar in Figure 5.

2.5 Tracking Progress

It is necessary to track progress while executing an ICT project, because that data is needed to later adjust further estimations, especially to get more reliable macro estimations.

The method of choice is Six Steps to Completion. All relevant deliverables – all work items in the matrix – are classified according the following schema:

- Test is Ready stage: Is problem understood and does a unit test exist?
- Draft is Ready stage: does a full draft exist? i.e., is code completed?
- Review Done stage: has the code been quality checked?
- Finalized stage: usually, something is left for improvement.
- Approved stage: team and sponsor agree that the work is done.
- Ready for Use stage: other programmers or users rely on the work result.

Chapter 3: Estimation In, Out, and Cost

3.1 The Role of an Independent Project Office

It is common practice that both estimations as progress tracking are done by either the supplier, or the sponsor. Professional ICT suppliers have an in-house project office to ensure compliance to the committed plans.

Such a solution is not appropriate to complex projects. Neither estimations can be guaranteed to remain unbiased, nor is tracking free from political thoughtfulness when done in-house. However, the aim of both is getting an expert view on reality.

It is recommended best practice to rely on third party recommendations for the following aspects of an ICT project:

- Functional Sizing – an independent evaluation of project goals and requirements serves as a second opinion to the suppliers proposal
- Estimation based on benchmarking data and the Buglione-Trudel matrix
- Tracking progress with the Six Sigma method Six Steps to Completion
- Doing a Post-Project Review based on all project data collected

If both customer and supplier agree, the independent project office submits the collected data anonymously to the ISBSG benchmarking data base – increasing knowledge about project costs and cost drivers and avoiding future cost for the industry and for society.

3.2 Preconditions

For doing project estimates, the project must have stated goals, with a profile indicating relative importance between the goal topics.

Technical constraints and the technology platform (e.g., choice of smartphones) must be known and stated, and overall requirements regarding user interface, interface to other systems, existing database, external services must be known. If these are not stated, or no Kano analysis is available, we need to elaborate these requirements in a half-day workshop.

3.3 Deliverables

Results of the early estimation are a series of commented graphical overviews, either in the IFPUG or the COSMIC style that represent the application being counted. Comments include brief descriptions of the entities entered in the functional size count – objects of interest in the COSMIC case, transactions for IFPUG sizing.

Also delivered is a Kano diagram and a Buglione-Trudel matrix, identifying the presumed work items needed to meet business goals. This work item selection is not necessarily what will be implemented; they represent what effort will be needed to meet business goals.

The overall estimation takes one to five business days and Euro Project Office offers them as fixed price, depending from the size (apps, small, medium, large, extra-large).

3.4 Tool Support by Euro Project Office

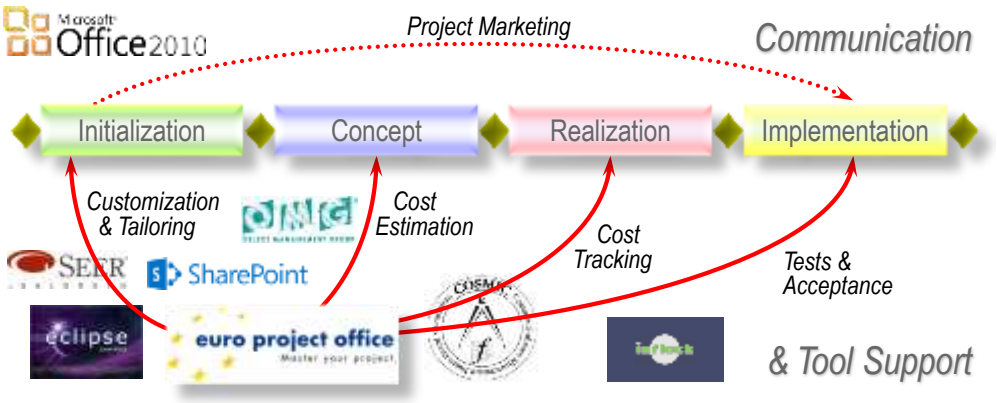

Figure 8: Full Euro Project Office Support in an ICT project

Euro Project has a long-standing collaboration with the Swiss Informatiksteuerungsorgan Bund (ISB) on distributing and supporting the project methodology HERMES, now in version 5; distribution agreements for the SEER estimation tools with Galorath, Inc., for the web eMetrics tools from IntLock, Inc., and can help with almost any relevant open and international standard in the ICT area.

Change Control

The following table documents the actual development stage of this document.
Every change made to this document requires a new issue.

Modification Notice	Author	Version	Date
New document	Th. Fehlmann	W1.0	2012-09-25
Added tool support; published	Th. Fehlmann	V1.0	2013-04-16