PPNOTE

Cost and effort estimation: COCOMO II

For estimating cost and effort, we use the Cocomo II method, statistical approach.

First step for arriving to an effort estimation is to understand in which case we are:

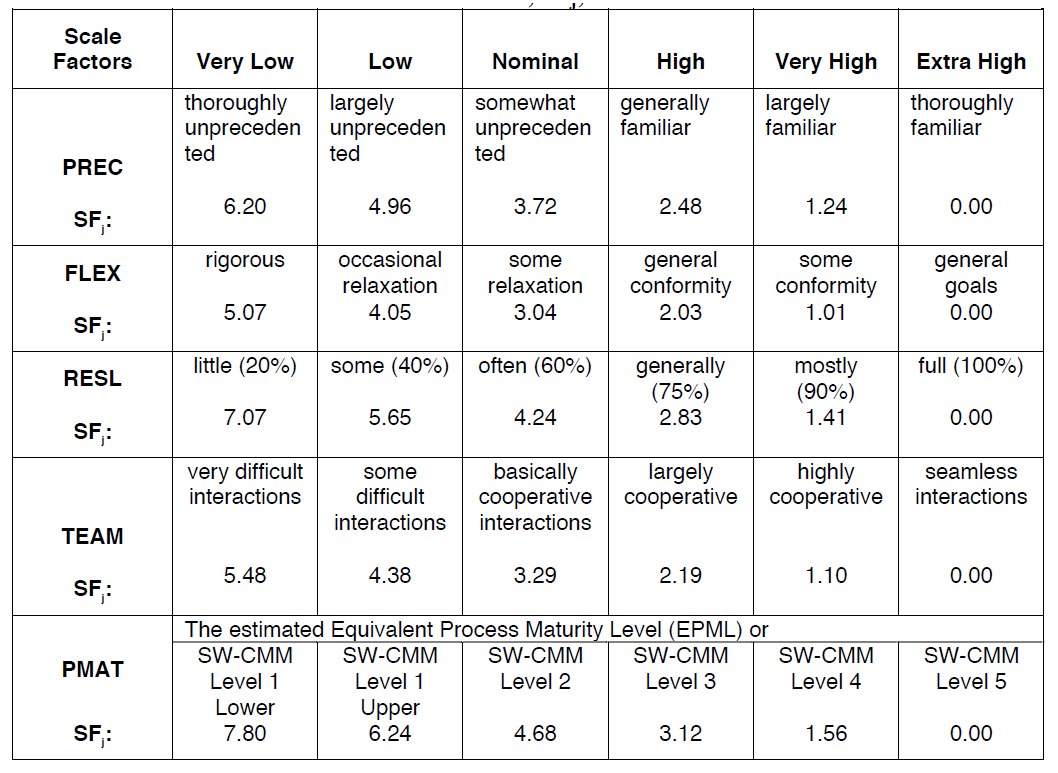
Post-Architecture: when we are extending an existing product line (we have already detailed information on cost).

Early Design: when we don’t have clear information on the architecture of the system.

We decide to follow Post-Architecture method They have an assign document with a specification of what the customer wants and this document will be redact after RASD,DD and ITPD document so we have a clearly information about the product.

SCALE FACTOR:

In order to evaluate the values of the scale drivers, we refer to the following official COCOMO II table:



Precedentedness: this product is the first one that we have projected so we don’t have experience in project development, the values will be low.

Development Flexibility: we can considerate this factor in two aspects:

* Need for software conformance with pre-established requirements: we have an assignment document with several implicit specification and from this we create a Rasd document and define explicit requirements, so we have some requirements to achieve, the value of this sub-factor will be nominal.
* Need for software conformance with external interface specifications: in the project will be necessary a communication among several different type of clients so the external interface will be well-defined for each type of client, providing specific service in base of client type, hence the value will be nominal.

The value of FLEX is based of these two parameters will be nominal.

Risk resolution: the risk plan that will be performed in the next chapter will be quite extensive and cover major aspects, the value will be high. //DA FARE DOPO RISK PLAN

Team Cohesion: the team is formed by two persons that have already worked together for three projects, so the parameters constitute this factor:

* Consistency of stakeholder objectives and cultures.
* Willingness of stakeholders to accommodate other stakeholders’ objectives: there is always a constructive discussion about main important point of the project.
* Stakeholder teambuilding to achieve shared vision and commitments.

All of this parameters will be very high value; hence team cohesion will be very high.

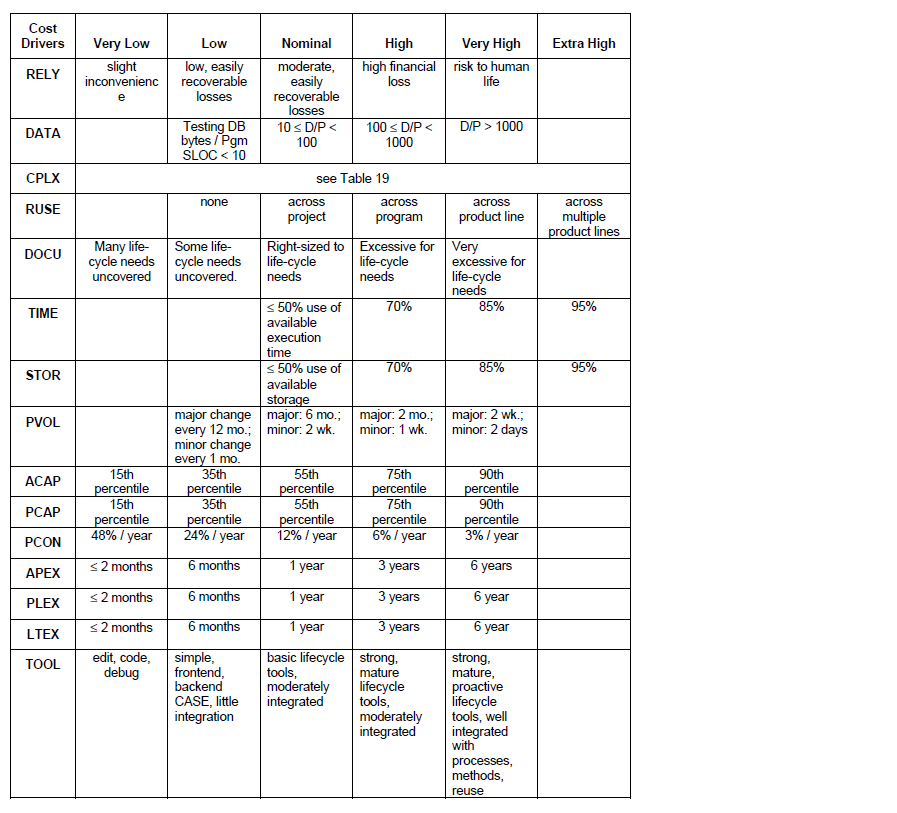
Process Maturity: for define the level of our project we use CMMI, this is the first our project so in based of our experience we think that our specification can reach level 3 of CMMI certification, this set of document (RASD, DD, ITPD and PP) have the achieve of avoiding inconsistency and incoherency among processes that provide same or different service.

The result of our evaluation is the following:

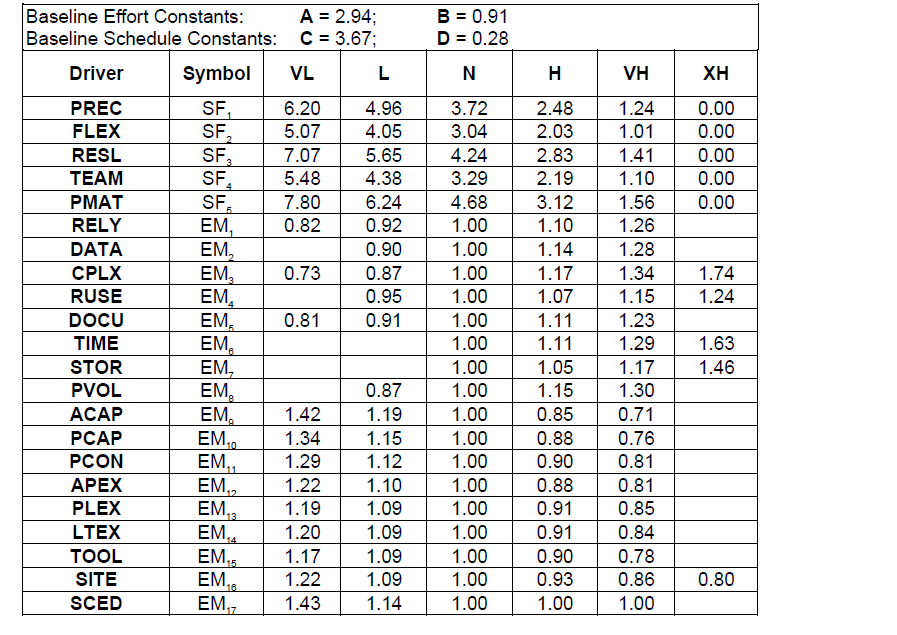
|  |  |  |
| --- | --- | --- |
| **Scale Driver** | **Factor** | **Value** |
| Precedentedness (PREC) | Low | 4.96 |
| Development flexibility(FLEX) | Nominal | 3.04 |
| Risk Resolution(RESL) | High | 2.83 |
| Team Cohesion(TEAM) | Very High | 1.10 |
| Process Maturity(PMAT) | Level 3 | 3.12 |
| **Total** |  | 15,05 |

COST DRIVERS

In order to evaluate the values of the cost drivers, we refer to the following official COCOMO II table:

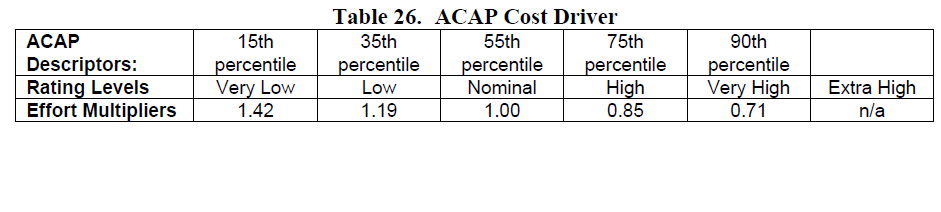


In the following table we see the effort multiplier for each cost driver:



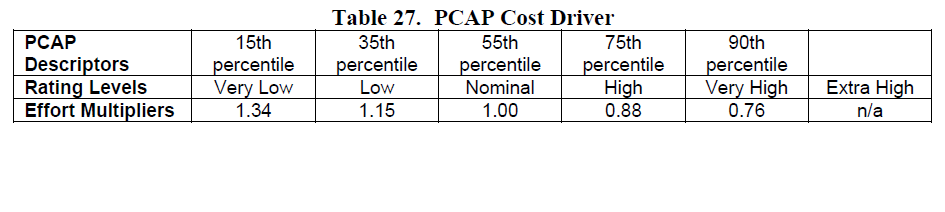
* Analyst Capability(ACAP):

The analysis of requirements that we have computed in RASD and DD document try to cover all possible aspects of the project moreover we insert “very high” in TEAM scale factor because there is a great communication among team members so the of ACAP will be high.



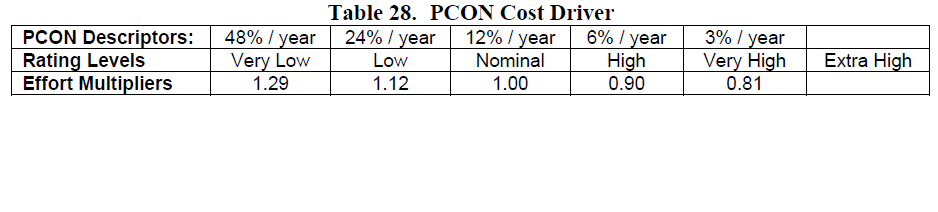
* Programmer Capability(PCAP):

This evaluation is based on the whole team capability in programming, for our team the capability is nominal because we have worked at only programming project so our experience is neither low nor high, PCAP will be nominal.



* Personnel Continuity (PCON):

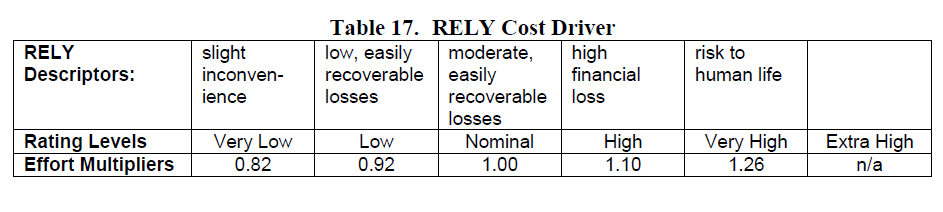
The time for this project is limited so we set the value of this parameters very low.



* Required Software Reliability(RELY):

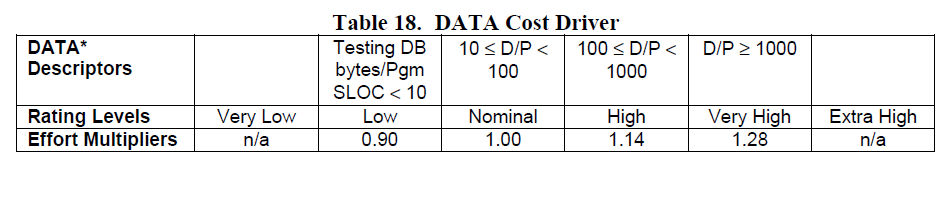
The reliability of the services is very important point for this project, a user should be performing a reservation when he wants, the major services like reservation and unlocking car must be available at any time.

If there is a software failure maybe, we can loss the client so the RELY values is set to high.



* Database size(DATA):

Database test should be quite large because our database should be memorizing all personal user data and providing the history payment data the database capture also all ride data for each user, DATA will be high.

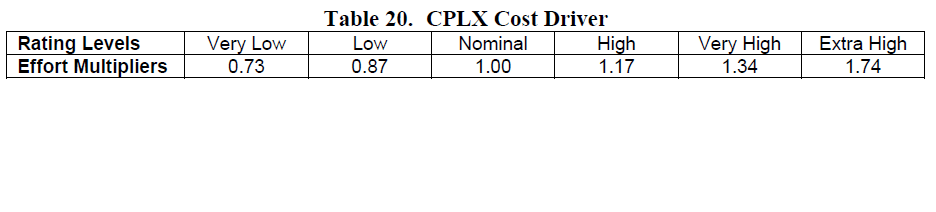


* Product Complexity(CPLX):

Complexity is divided into five points:

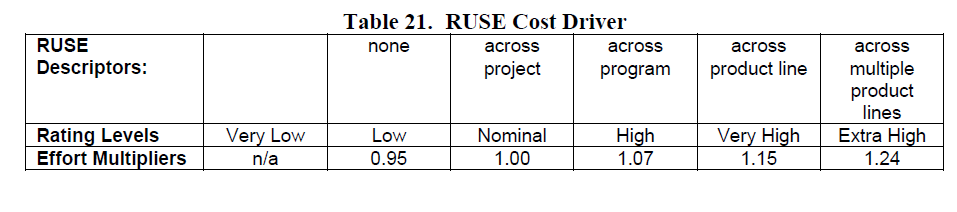
* + Control operations: the system must perform all request that clients send, only for this fact we must implement some control operation to control the incoming and outgoing stacks and the application must be supported for each type of device and system (Very-High).
  + Computational Operations: for develop the system we need to use structure like matrix and vector for controlling the queues and capture the user position for example (Nominal).
  + Device-dependent Operations: all operation I/O will be done with the simply methods GET or POST to capture the information that user sends (Low).
  + Data Management Operations: the database that we would build is a simple database containing with re-edit and change database structure is not contemplated, the dbms manager create various type of queries (Low).
  + User Interface Management Operations: The various User interface is composed by GUI for providing the services (Low).

Through this analysis the value of CPLX is Nominal.



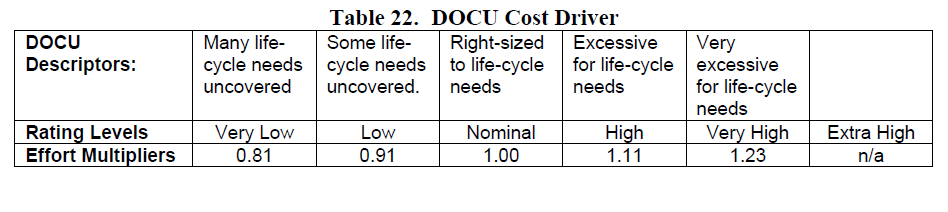
* Developed for Reusability(RUSE):

In the DD document we declare to use JEE for supporting our component architecture, one of the most advantage of using JEE is the code reusability because it’s organize in “Beans” and if the customer wants another service is enough to add a new beans and create an interaction among the other ones (Very High).



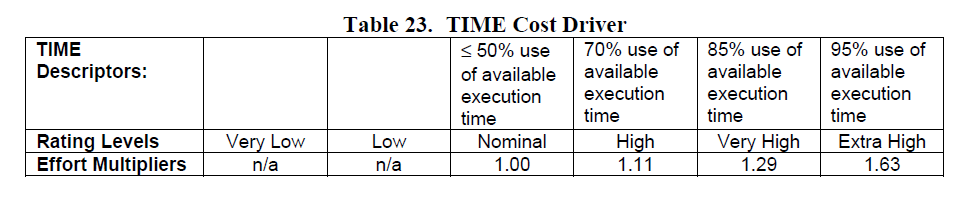
* Documentation match to life-cycle needs(DOCU):

The documentation covers all parts of the product life-cycle (Nominal) but in RUSE we set very high so the documentation will be excessive for life-cycle needs in this way we have a fully reusability(High).



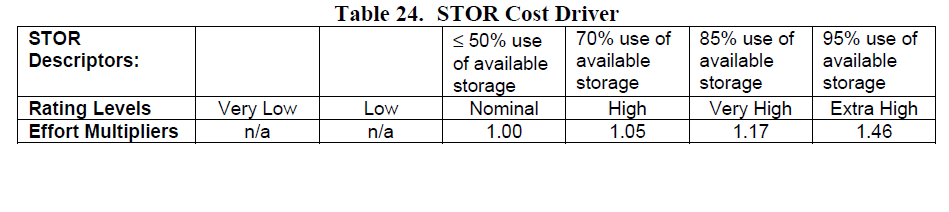
* Execution Time Constraint (TIME):

When the software system runs on server doesn’t occupy much space because there isn’t heavy process, the system just send/receive message (Nominal).



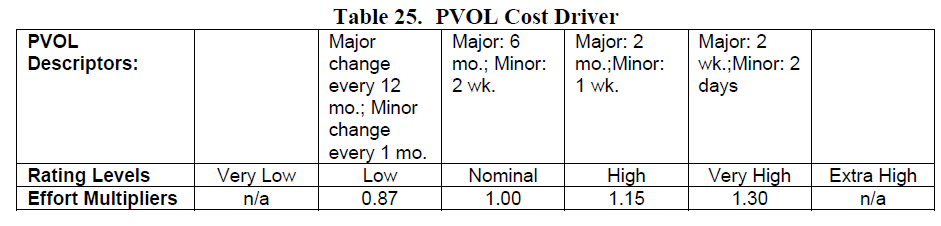
* Main storage constraint (STOR):

We think that the amount of storage usage is enough respect the availability of the hardware because when the software run occupy at most 50$ of available storage (Nominal).



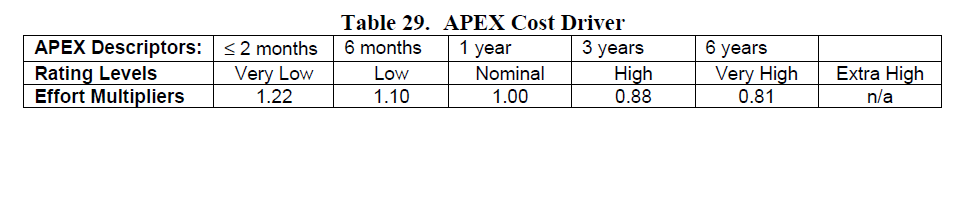
* Platform volatility (PVOL):

Various applications don’t change very often, even if the reusability of the code; maybe database maintenance every six months (Nominal)



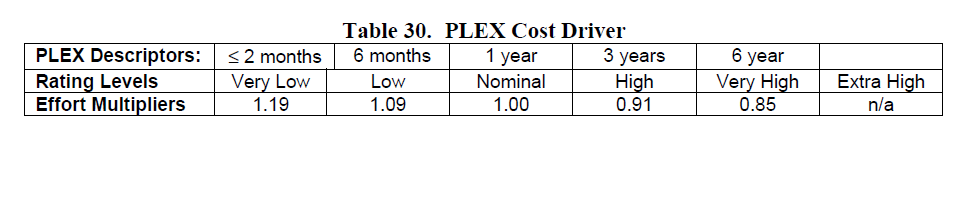
* Application Experience (APEX):

The team members work on a project only one time for six months and the focus of that project is quite different from this project so the experience value will be low.



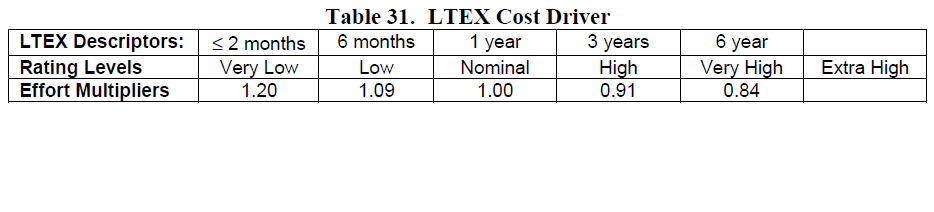
* Platform Experience (PLEX):

The team member never worked on any platform used in this project but they have a theory background on database and network platform and how these platforms are used hence the value of PLEX will be set to low.



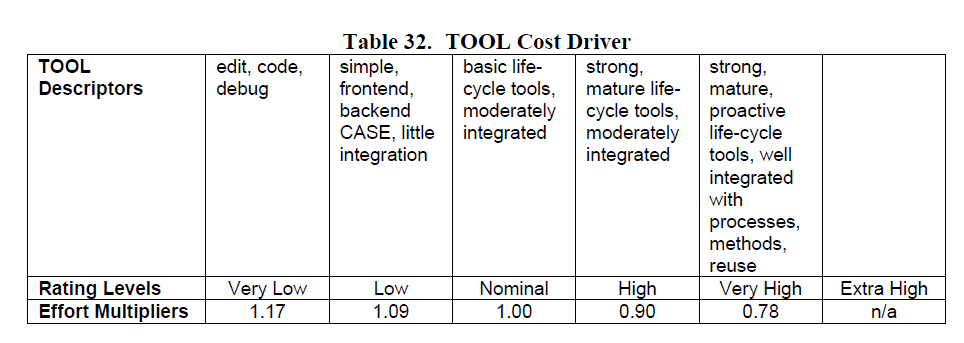
* Language and tool experience (LTEX):

The team members have a theory background on programming language, networking and database management moreover the team have just worked on a programming project the value set for LTEX will be Nominal.



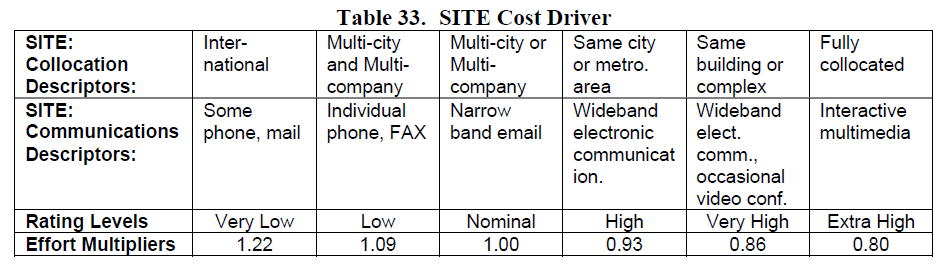
* Use of software tool (TOOL):

We use tools for the basic life-cycle of the project, this tools support the project development, the value of TOOL will be Nominal



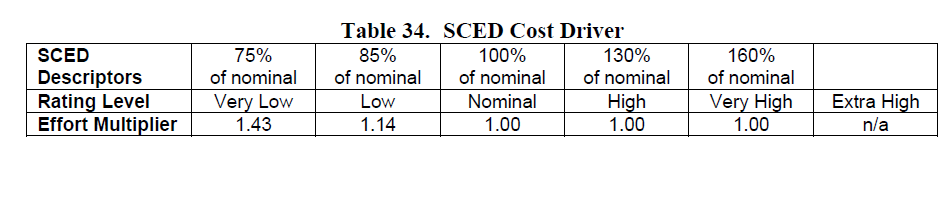
* Multisite Development (SITE):

The major part of the time the team is in the same building for discussing the project; so the value of SITE will be very high.



* Requirement Development Schedule (SCED):

The deadline for this project is fixed and the range of time for redact each document is variable, even if we are well-balanced the work the part of requirements and the component was quite long and after these parts we must accelerate to accomplish the deadline, so the value of SCED is set to high.

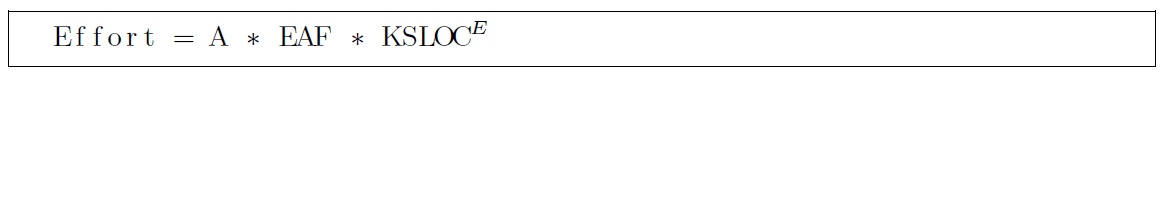


The following table resume the results of cost driver:

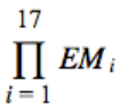
|  |  |  |
| --- | --- | --- |
| Cost Driver | Rank | Value |
| Analyst Capability(ACAP) | High | 0,85 |
| Program. Capability(PCAP) | Nominal | 1,00 |
| Person. Continuity(PCON) | Low | 1,12 |
| Required Software Reliability(RELY) | High | 1,10 |
| Database size(DATA) | High | 1,14 |
| Product Complexity(CPLX) | Nominal | 1,00 |
| Developed for Reusability(RUSE) | Very High | 1,15 |
| Documentation match to life-cycle needs(DOCU) | High | 1,11 |
| Execution Time Constraint (TIME) | Nominal | 1,00 |
| Main storage constraint (STOR) | Nominal | 1,00 |
| Platform volatility (PVOL) | Nominal | 1,00 |
| Application Experience (APEX) | Low | 1,10 |
| Platform Experience (PLEX) | Low | 1,09 |
| Language and tool experience (LTEX) | Nominal | 1,00 |
| Use of software tool (TOOL) | Nominal | 1,00 |
| Multisite Development (SITE) | Very High | 0,86 |
| Requirement Development Schedule (SCED) | High | 1,00 |
| **Total** |  | **1.5714** |

EFFORT ESTIMATION

The following formula permits us to calculate the effort estimation in terms of Person-Month:



A=2,94 in COCOMO II

Where EAF= =1.5714

E= exponent derived from the scale driver analysis = =0,91+0,01\*15,05=1,0605

B=0,91 in COCOMO II

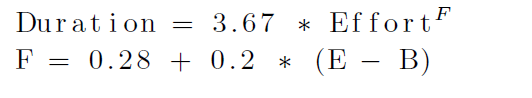
SLOC=5198=5,198KSLOC

So, with this parameter the effort estimation is:

Effort=2,94 \* 1,5714 \* 5.743KSLOC =26,532 PM=>27PM

Duration Estimation

The following formula permits us to calculate the duration estimation:



F=0.28+0.2\*(1.0605-0.91) = 0.3101

Duration= 3.67 \* 26.532 PM ^ 0.3101=10,15 months => 11months