



Mètriques de productivitat de programari per a la gestió de projectes

Alumne

Jaime Sánchez Sanz
jsanchezsanz@uoc.edu

Consultora

Ana Cristina Domingo Troncho

Asignatura

TFC-Gestió de projectes
Segon semestre 2010/2011

INDEX DE CONTINGUT

| | |
|--|----|
| INDEX DE FIGURES | 5 |
| INDEX DE TAULES | 5 |
| 1 INTRODUCCIÓ | 8 |
| 1.1 Justificació del TFC..... | 8 |
| 1.2 Objectius del TFC | 8 |
| 1.3 Enfocament i mètode seguit | 9 |
| 1.4 Planificació el Projecte..... | 9 |
| 1.5 Productes obtinguts | 11 |
| 1.6 Breu descripció dels capítols de la memòria | 11 |
| 2 CONCEPTES BÀSICS..... | 13 |
| 2.1 Mesures | 13 |
| 2.2 Mètriques | 13 |
| 2.3 Indicadors | 13 |
| 2.4 Atributs interns i atributs externs..... | 13 |
| 2.5 Factors a tenir en compte | 14 |
| 3 MÈTRIQUES | 15 |
| 3.1 Mètriques del Procés | 15 |
| 3.2 Mètriques del Projecte | 16 |
| 3.3 Mètriques del Programari..... | 17 |
| 3.4 Mètriques de Productivitat | 18 |
| 4 MÈTRIQUES DE PRODUCTIVITAT | 18 |
| 4.1 Introducció | 18 |
| 4.2 Raons per mesurar un producte | 19 |
| 4.3 Necessitat de les mètriques de productivitat | 19 |
| 4.4 Mètriques aplicades a la gestió de projectes | 20 |
| 4.4.1. Requisits | 20 |

| | | |
|---------|---|----|
| 4.4.1.1 | Validació de REQ (VR) | 20 |
| 4.4.2. | Mida de la aplicació..... | 21 |
| 4.4.2.1 | Classes Clau (CC) | 21 |
| 4.4.2.2 | Classes Suport (CS) | 21 |
| 4.4.2.3 | Classes Totals (CT) | 22 |
| 4.4.2.4 | Centenars de Instruccions Fonts (CIF) | 22 |
| 4.4.3. | Estimació d'esforç | 23 |
| 4.4.3.1 | Esforç del projecte | 23 |
| 4.4.3.2 | Esforç amb reutilització | 24 |
| 4.4.4. | Personal | 24 |
| 4.4.4.1 | Mida del Equip de Treball (E.T.)..... | 24 |
| 4.4.4.2 | Experts per l'Àrea..... | 25 |
| 4.4.4.3 | Experiència com E.T. | 26 |
| 4.4.5. | Mètodes i eines | 27 |
| 4.4.5.1 | Mètodes i eines | 27 |
| 4.4.6. | Temps | 28 |
| 4.4.6.1 | Persones-Dia-Classe | 28 |
| 4.4.7. | Grau del rigor del projecte..... | 28 |
| 4.4.7.1 | Rigor del projecte | 28 |
| 4.4.8. | Control de canvis | 29 |
| 4.4.8.1 | Impacte de Risc | 29 |
| 4.4.9. | Conclusions..... | 30 |
| 5 | PROGRAMARI PER MÈTRIQUES | 30 |
| 5.1 | COCOMO 81 | 30 |
| 5.1.1. | Descripció del model COCOMO | 30 |
| 5.1.2. | Avantatges i desavantatges | 34 |
| 5.2 | ADA COCOMO | 35 |
| 5.3 | COCOMO INCREMENTAL..... | 37 |

| | | |
|---------|---|----|
| 5.4 | COCOMO II | 37 |
| 5.4.1. | Descripció del model COCOMO II | 37 |
| 5.4.1.1 | Model de Composició d'Aplicacions..... | 39 |
| 5.4.1.2 | Model de Disseny Anticipat..... | 40 |
| 5.4.1.3 | Model Post-Arquitectura..... | 41 |
| 5.4.1.4 | Punts Funció | 47 |
| 5.4.1.5 | Línies de Codi Font | 50 |
| 5.4.1.6 | Conversió de Punts Funció (UFP) a Línies de Codi Font (SLOC) | 50 |
| 5.5 | Costar | 51 |
| 5.5.1. | La pantalla principal de Costar..... | 51 |
| 5.5.2. | Introduir les línies de codi font | 52 |
| 5.5.3. | El editor de controlador de costos..... | 53 |
| 5.5.4. | Ajuda del editor de controlador de costos..... | 53 |
| 5.5.5. | Actualització d'un Factor de Cost..... | 54 |
| 5.5.6. | Factors d'escala de COCOMO II | 54 |
| 5.5.7. | Actualitzar els costos | 55 |
| 5.5.8. | El Informe Detallat..... | 56 |
| 5.5.9. | El Informe Programat..... | 56 |
| 5.5.10. | Gràfic de Costos vs Temps..... | 57 |
| 6 | CONCLUSIONS | 57 |
| 7 | GLOSSARI | 58 |
| 8 | BIBLIOGRAFIA..... | 59 |

INDEX DE FIGURES

| | |
|---|----|
| Figura 1: Calendari de projecte | 10 |
| Figura 2: Diagrama de Gantt del projecte..... | 11 |
| Figura 3: Mètriques de projecte..... | 17 |
| Figura 4: Checklist per la definició d'una Línea de Codi Font | 50 |
| Figura 5: Pantalla principal de Costar..... | 52 |
| Figura 6: Introduir les línies de codi font..... | 52 |
| Figura 7: Editor de controlador de costos..... | 53 |
| Figura 8: Ajuda del editor de controlador de costos | 53 |
| Figura 9: Actualització d'un Factor de Cost | 54 |
| Figura 10: Factors d'escala de COCOMO II..... | 55 |
| Figura 11: Actualitzar els costos..... | 55 |
| Figura 12: El Informe Detallat | 56 |
| Figura 13: El Informe Programat | 57 |
| Figura 14: Gràfic de Costos vs Temps | 57 |

INDEX DE TAULES

| | |
|--|----|
| Taula 1: Fites de control del projecte..... | 11 |
| Taula 2: Atributs Interns i Externs..... | 14 |
| Taula 3: Validació REQ | 21 |
| Taula 4: Funcions de cada participant en el projecte | 25 |
| Taula 5: Taula de mètriques | 27 |
| Taula 6: Descripció de les variables de projecte | 27 |
| Taula 7: Relació VALOR/ESCALA | 27 |
| Taula 8: Escala de les eines i mètodes utilitzats | 28 |

| | |
|--|----|
| Taula 9: Rigor del projecte..... | 29 |
| Taula 10: Riscos d'un projecte | 30 |
| Taula 11: Coeficients pels models de desenvolupament | 32 |
| Taula 12: Factors de Cost. Model COCOMO 81 | 33 |
| Taula 13: Abreviació dels Factors de Cost. Model COCOMO 81..... | 34 |
| Taula 14: Factors de Cost. Model ADA COCOMO..... | 36 |
| Taula 15: Abreviació dels Factors de Cost. Model ADA COCOMO | 37 |
| Taula 16: PROD. Model de Composició d'Aplicacions. COCOMO II | 40 |
| Taula 17: Multiplicadors i Valors. Model Disseny Anticipat. COCOMO II..... | 41 |
| Taula 18: RELY. Model Post-Arquitectura. COCOMO II | 42 |
| Taula 19: DATA. Model Post-Arquitectura. COCOMO II | 42 |
| Taula 20: CPLX. Model Post-Arquitectura. COCOMO II | 43 |
| Taula 21: RUSE. Model Post-Arquitectura. COCOMO II..... | 43 |
| Taula 22: DOCU. Model Post-Arquitectura. COCOMO II | 44 |
| Taula 23: PVOL . Model Post-Arquitectura. COCOMO II | 44 |
| Taula 24: STOR. Model Post-Arquitectura. COCOMO II..... | 44 |
| Taula 25: TIME. Model Post-Arquitectura. COCOMO II | 44 |
| Taula 26: ACAP. Model Post-Arquitectura. COCOMO II..... | 45 |
| Taula 27: PCAP. Model Post-Arquitectura. COCOMO II..... | 45 |
| Taula 28: PCON. Model Post-Arquitectura. COCOMO II | 45 |
| Taula 29: AEXP. Model Post-Arquitectura. COCOMO II | 45 |
| Taula 30: PEXP. Model Post-Arquitectura. COCOMO II | 46 |
| Taula 31: LTEX. Model Post-Arquitectura. COCOMO II..... | 46 |
| Taula 32: TOOL. Model Post-Arquitectura. COCOMO II..... | 46 |
| Taula 33: SITE Ub. Espacial. Model Post-Arquitectura. COCOMO II | 46 |
| Taula 34: SITE Comunicació. Model Post-Arquitectura. COCOMO II..... | 47 |
| Taula 35: SCED. Model Post-Arquitectura. COCOMO II..... | 47 |
| Taula 36: Calculació dels Factors d'Ajustament. Model Post-Arquitectura. COCOMO II | 47 |

| | |
|---|----|
| Taula 37: Nivells de Complexitat. Model Post-Arquitectura. COCOMO II | 48 |
| Taula 38: Pes del Factor de Complexitat. Model Post-Arquitectura. COCOMO II | 49 |
| Taula 39: SLOC/Punt Funció..... | 51 |

1 INTRODUCCIÓ

1.1 Justificació del TFC

La indústria del software, a diferència d'altres indústries, té molt poc temps d'existència. Dos punts que han cridat força l'atenció en aquest mercat han sigut la velocitat amb la que han crescut i el seu abast. Des de el seu inici van existir persones en diferents camps que van veure l'avanç que per ells representava l'ús del software especialitzat que els permetia automatitzar processos i accelerar-los.

Al haver-hi tanta demanda es van iniciar moltes investigacions a la rama de software i hardware. Amb el temps, es van reduir costos i el software es va convertir en un negoci rentable. De tot això van sorgir les primeres grans empreses de software. Tot això, va portar que hi hagués moltes companyies de diferents països i, per tant, diferents programaris amb una diversitat d'estils molt variada. Aquesta diversitat va suposar la necessitat de poder mesurar els temps de realització, costos i recursos de qualsevol projecte que es plantejava.

Moltes empreses, sobretot les empreses dedicades al desenvolupament de software, preocupades amb aquesta necessitat van desenvolupar les seves pròpies formes de obtenir dades concretes de la productivitat.

Amb la importància d'aquestes mètriques en el desenvolupament de software, van sorgir diverses organismes per donar solució a aquests problemes. Aquets organismes van ser "IEEE" al 1992 i "ISO" al 2002, que van definir els seus estàndards de mètriques de productivitat.

Aquest TFC vol ser una ajuda per explicar tots aquests estàndards de productivitat que s'han estat implantant al llarg d'aquets anys de vida del desenvolupament del software. Així, es farà una revisió de les mètriques de productivitat més utilitzades per poder fer una mesura de temps, costos i recursos molt més acurada, necessari per al desenvolupament de projectes de programari.

1.2 Objectius del TFC

Els objectius d'aquest Treball de Final de Carrera són:

- Definir els conceptes bàsics per entendre millor les mètriques, la seva utilitat i, així, poder justificar el seu us.
- Enumerar i explicar els diferents tipus de mètriques que podem utilitzar.

- Farem un aprofundiment molt exhaustiu de les mètriques que ens ocupen i preocupen en aquest TFC, es a dir, farem un estudi de les mètriques de productivitat del programari.
- Estudiarem quatre models que ajuden a planificar i controlar l'esforç i temps necessari pel desenvolupament de software.

Per tant, farem un estudi profund de les mètriques de productivitat i la seva importància per poder mesurar el procés evolutiu del software.

1.3 Enfocament i mètode seguit

Per realitzar aquest projecte s'ha optat per un desenvolupament en cascada. Seguint aquest model, s'han ordenat totes les etapes en cascada de forma que el començament d'una etapa ha d'esperar a la finalització de l'altra etapa.

Com només hi ha un recurs per a la realització d'aquest TFC s'ha optat per aquest model com la millor opció.

1.4 Planificació el Projecte

Donat la importància del TFC i el temps que se li ha de dedicar, es aconsellable la dedicació diària en la elaboració d'aquest projecte. La data de començament serà el 18 de Març del 2011 i la data de entrega del TFC serà el 06 de Juny del 2011. S'ha optat per deixar uns dies de marge al final per qualsevol imprevist que pugui sorgir en la elaboració d'aquest TFC.

Així, per tant, la descomposició en tasques d'aquest projecte serà:

| | Task Name | Duración | Comienzo | Fin | Predecesoras |
|----|--|-----------|--------------|--------------|--------------|
| 1 | Definición del alcance del proyecto. | 81 días | vie 18/03/11 | lun 06/06/11 | |
| 2 | Punt 1: Introducció | 6 días | vie 18/03/11 | mié 23/03/11 | |
| 3 | Justificació del TFC | 1 día | vie 18/03/11 | vie 18/03/11 | |
| 4 | Objectius del TFC | 1 día | sáb 19/03/11 | sáb 19/03/11 | 3 |
| 5 | Enfocament i mètode seguit | 1 día | dom 20/03/11 | dom 20/03/11 | 4 |
| 6 | Planificació del projecte | 1 día | lun 21/03/11 | lun 21/03/11 | 5 |
| 7 | Productes obtinguts | 1 día | mar 22/03/11 | mar 22/03/11 | 6 |
| 8 | Breu descripció dels capítols de la memòria | 1 día | mié 23/03/11 | mié 23/03/11 | 7 |
| 9 | Punt 2: Conceptes Bàsics | 3,13 días | jue 24/03/11 | dom 27/03/11 | |
| 10 | Mesures | 5 hrs | jue 24/03/11 | jue 24/03/11 | 8 |
| 11 | Mètriques | 5 hrs | jue 24/03/11 | vie 25/03/11 | 10 |
| 12 | Indicadors | 5 hrs | vie 25/03/11 | vie 25/03/11 | 11 |
| 13 | Atributs interns i atributs externs | 5 hrs | vie 25/03/11 | sáb 26/03/11 | 12 |
| 14 | Factors a tenir en compte | 5 hrs | sáb 26/03/11 | dom 27/03/11 | 13 |
| 15 | Punt 3: Mètriques | 2,5 días | dom 27/03/11 | mar 29/03/11 | |
| 16 | Mètriques del Procés | 4 hrs | dom 27/03/11 | dom 27/03/11 | 14 |
| 17 | Mètriques del Projecte | 4 hrs | dom 27/03/11 | lun 28/03/11 | 16 |
| 18 | Mètriques del Programari | 4 hrs | lun 28/03/11 | lun 28/03/11 | 17 |
| 19 | Mètriques de Productivitat | 1 día | lun 28/03/11 | mar 29/03/11 | 18 |
| 20 | Punt 4: Mètriques de Productivitat | 17,1 días | mar 29/03/11 | vie 15/04/11 | |
| 21 | Introducció | 1 día | mar 29/03/11 | mié 30/03/11 | 19 |
| 22 | Raons per mesurar un producte | 1 día | mié 30/03/11 | jue 31/03/11 | 21 |
| 23 | Necessitat de les mètriques de productivitat | 1 día | jue 31/03/11 | vie 01/04/11 | 22 |
| 24 | Mètriques aplicades a la gestió de projectes | 14,1 días | vie 01/04/11 | vie 15/04/11 | |
| 25 | Requisits | 1,8 días | vie 01/04/11 | dom 03/04/11 | 23 |
| 26 | Mida de la aplicació | 1,8 días | dom 03/04/11 | mar 05/04/11 | 25 |
| 27 | Estimació d'esforç | 1,8 días | mar 05/04/11 | jue 07/04/11 | 26 |
| 28 | Personal | 1,5 días | jue 07/04/11 | vie 08/04/11 | 27 |
| 29 | Mètodes i eines | 1,8 días | vie 08/04/11 | dom 10/04/11 | 28 |
| 30 | Temps | 1,8 días | dom 10/04/11 | mar 12/04/11 | 29 |
| 31 | Grau del rigor del projecte | 1,8 días | mar 12/04/11 | mié 13/04/11 | 30 |
| 32 | Control de canvis | 1,8 días | mié 13/04/11 | vie 15/04/11 | 31 |
| 33 | Punt 8: Bibliografia | 1 día | vie 15/04/11 | sáb 16/04/11 | 32 |
| 34 | Revisió PAC2 | 3 días | sáb 16/04/11 | mar 19/04/11 | 33 |
| 35 | Entrega PAC2 | 0 hrs | mar 19/04/11 | mar 19/04/11 | 34 |
| 36 | Punt 5: Programari per mètriques | 17 días | mié 20/04/11 | vie 06/05/11 | |
| 37 | COCOMO 81 | 4 días | mié 20/04/11 | sáb 23/04/11 | 35 |
| 38 | ADA COCOMO | 3 días | dom 24/04/11 | mar 26/04/11 | 37 |
| 39 | COCOMO INCREMENTAL | 3 días | mié 27/04/11 | vie 29/04/11 | 38 |
| 40 | COCOMO II | 3 días | sáb 30/04/11 | lun 02/05/11 | 39 |
| 41 | Costar | 4 días | mar 03/05/11 | vie 06/05/11 | 40 |
| 42 | Punt 6: Conclusions | 2 días | sáb 07/05/11 | dom 08/05/11 | 41 |
| 43 | Punt 7: Glossari | 2 días | lun 09/05/11 | mar 10/05/11 | 42 |
| 44 | Punt 8: Bibliografia | 1 día | mié 11/05/11 | mié 11/05/11 | 43 |
| 45 | Tancar el projecte | 2 días | jue 12/05/11 | vie 13/05/11 | 44 |
| 46 | Revisió PAC3 | 4 días | sáb 14/05/11 | mar 17/05/11 | 45 |
| 47 | Entrega PAC3 | 0 hrs | mar 17/05/11 | mar 17/05/11 | 46 |
| 48 | Revisión memoria y Presentación Proyecto. | 20 días | mié 18/05/11 | lun 06/06/11 | |
| 49 | Revisión Memoria proyecto. | 1 día | mié 18/05/11 | mié 18/05/11 | 47 |
| 50 | Preparación presentación ejecutiva. | 15 días | jue 19/05/11 | jue 02/06/11 | 49 |
| 51 | Revisió TFC | 4 días | vie 03/06/11 | lun 06/06/11 | 50 |
| 52 | Entrega TFC | 0 hrs | lun 06/06/11 | lun 06/06/11 | 51 |

Figura 1: Calendari de projecte

I es proposa seguir el següent diagrama de Gantt:

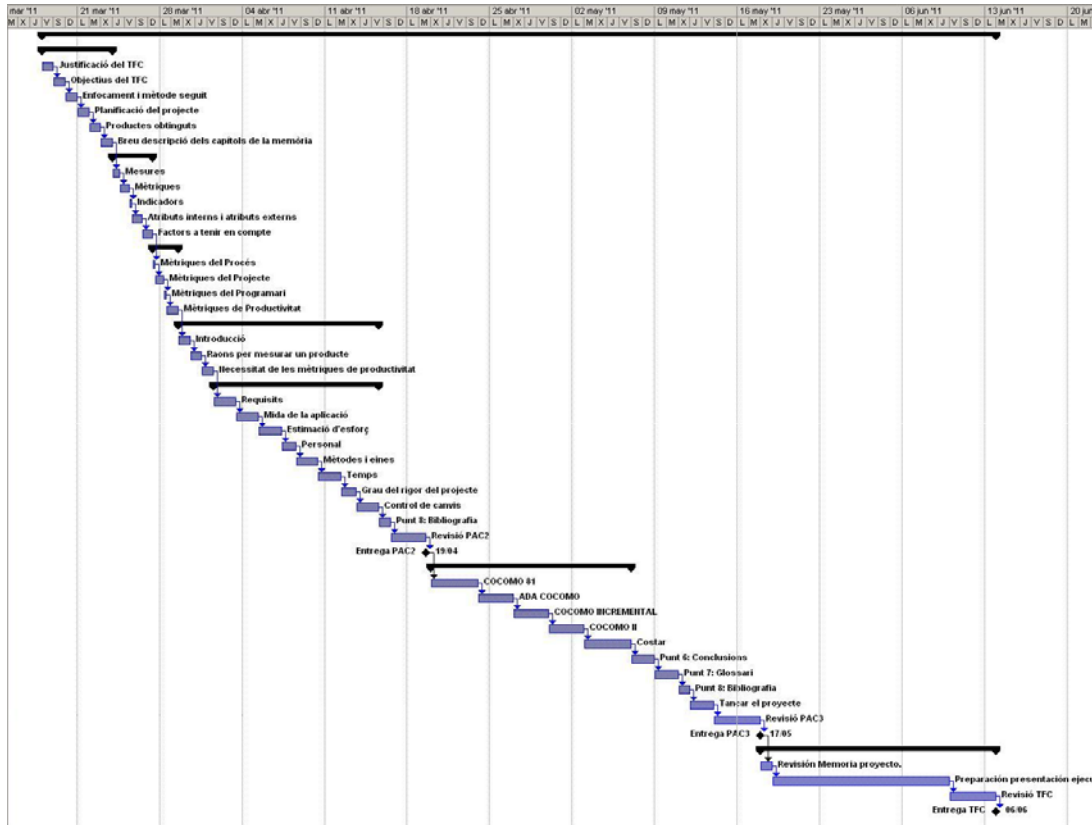


Figura 2: Diagrama de Gantt del projecte

Les fites de control del TFC proposades seran:

| Fita | Data | Descripció |
|---------------|------------|--|
| PAC1 | 22/03/2011 | Elecció del tema del TFC, el seu abast i pla de treball |
| PAC2 | 19/04/2011 | Introducció al TFC, Conceptes Bàsics i Mètriques |
| PAC3 | 17/05/2011 | Programari per mètriques, Conclusions i Tancar el projecte |
| Entrega Final | 06/06/2011 | Finalització Memòria del projecte i Presentació |

Taula 1: Fites de control del projecte

1.5 Productes obtinguts

El producte obtingut d'aquest projecte serà aquesta memòria i una presentació virtual que sintetitzi el contingut de la memòria.

1.6 Breu descripció dels capítols de la memòria

Segons el punts marcats al pla de projecte la descripció dels capítols de la memòria serà com s'explica a continuació.

Les punts del TFC s'aniran desenvolupant al llarg del semestre de forma ordenada i equitativament perquè totes les activitats es puguin complir sense problemes de temps.

El TFC estarà dividit en 8 punts:

Punt 1: Introducció. Aquest punt serà la introducció de tot el TFC. Primer farem una justificació del perquè de la realització d'aquest projecte. Posteriorment, mostrarem els objectius d'aquest treball. A continuació, enfocarem el TFC i el mètode seguit per a l'elaboració del mateix. Es mostrarà la planificació del projecte seguit per aconseguir elaborar tot el document i la presentació final i, com a últim punt, farem una breu descripció de tots els punts del TFC.

Punt 2: Conceptes bàsics. Aquest apartat té com a finalitat fer una petita introducció dels conceptes més importants per a entendre aquest projecte amb molta més claredat. Podríem pensar que es similar a un glossari, però la seva funció es una mica més extensa ja que els punts explicats són amb més profunditat que la funció pròpia d'un glossari.

Els conceptes que tractarem seran les mesures, les mètriques, els indicadors, els atributs interns i atributs externs i, estudiarem un últim punt amb factors a tenir en compte.

Punt 3: Mètriques. En el punt 3 es descriuen els tipus més importants de mètriques que actualment podem utilitzar per estudiar la productivitat de un producte, un projecte, un programari i/o de la pròpia gestió del projecte.

Punt 4: Mètriques de Productivitat. Les mètriques de productivitat són les mètriques que tenim interès estudiar en aquest TFC. Dins d'aquestes mètriques de productivitat s'estudiaran fins a catorze tipus. Aquests catorze tipus seran: Validació de Requisits, Classes Clau, Classes Suport, Classes Totals, Centenars de Instruccions Fonts, Esforç del projecte, Esforç amb reutilització, Mida del equip de treball, Experts per l'Àrea, Experiència com equip de treball, Mètodes i eines, Persones-Dia-Classe, Rigor del projecte, Impacte de Risc.

Punt 5: Programari per mètriques. En aquest apartat estudiarem quatre tipus d'estimació de costos molt importants: COCOMO 81, ADA COCOMO, COCOMO INCREMENTAL, COCOMO II i Costar.

Punt 6. Conclusions. En aquest punt farem una conclusió de tots els punts comentats anteriorment amb la nostra opinió i la situació actual que hi ha en base a la utilització de totes les mètriques comentades en la gestió de projectes.

Punt 7. Glossari. Es realitzarà un glossari de tots els termes més confusos del document.

Punt 8. Bibliografia. Es farà esmenta a tota la bibliografia utilitzada i complementaria que pugui ser d'utilitat.

2 CONCEPTES BÀSICS

2.1 Mesures

La mesura proporciona una indicació quantitativa de la extensió, quantitat, dimensió, capacitat y mida d'alguns atributs d'un procés o producte. Si ens basem en la enginyeria del software, la mesura serà la quantitat de línies de codi que té un programa.

Per exemple: un programa conté 20.000 LDC (Línies de codi).

Hi poden haver mesures directes i mesures indirectes. La mesura directa és aquella que no depèn de cap altre atribut. La mesura indirecta és aquella que depèn d'un o més atributs.

2.2 Mètriques

Una mètrica es una mesura quantitativa del grau en que un sistema, component o procés posseeix un atribut donat.

Per exemple: la productivitat d'aquest projecte va ser de 500 LDC/persona-mes.

Així com la mesura captura una característica individual, la mètrica ens permet relacionar i comparar mesures. La mètrica és el fonament dels indicadors.

2.3 Indicadors

Un indicador es una mètrica o combinació de mètriques que proporciona una visió profunda del procés del software, del projecte de software o del producte en si.

Per exemple: la productivitat mitja de la nostra empresa es de 500 LDC/persona-mes i en el últim projecte ha sigut de 250 LDC/persona-mes.

2.4 Atributs interns i atributs externs

Els atributs interns de un producte, procés o recurs són aquells que podem mesurar purament en termes del producte, procés o recurs del mateix. Poden ser mesurats directament. Per exemple: la longitud d'un programa o el temps transcorregut de qualsevol document de software.

Els atributs externs d'un producte, procés o recurs són aquells que solament poden ser mesurats amb respecte al com el producte, procés o recurs es relacionen amb el seu ambient. Aquets tendeixen a ser els que el administrador i el usuari del software solen mesurar i precedir. Per exemple: el administrador de software li agradaria saber

el cost de eficàcia de alguns processos o de la productivitat del seu personal, mentre que als usuaris li agradarien saber la utilització, fiabilitat o portabilitat d'un sistema que ells observen per comprar. Els atributs externs són els més difícils de mesurar, perquè aquets no poden ser mesurats directament.

A la taula següent es mostra alguns exemples de atributs interns i externs:

| ENTITATS | ATRIBUTS | |
|--------------------------------|---|--|
| | Interns | Externs |
| Recursos | | |
| Personal | edat, preu | productivitat, experiència, intel·ligència |
| Equip | mida, estructures | productivitat, qualitat |
| Software | preu, mida | usabilitat, seguretat |
| Hardware | preu, velocitat, mida de la memòria | seguretat |
| Oficines | mida, temperatura, llum | confort, qualitat |
| Processos | | |
| Construcció de especificacions | temps, esforç, nombre de canvis | qualitat, cost, estabilitat |
| Disseny detallat | temps, esforç | cost, cost efectiu |
| Proves | nombre d'errors de codi encontrats, temps | estabilitat, cost, cost efectiu |
| Productes | | |
| Especificacions | mida, utilització, funcionalitat | comprensibilitat, manteniment |
| Dissenys | acoblament, modularitat, mida, utilització | qualitat, complexitat, manteniment |
| Codi | funcionalitat, complexitat algorítmica, control de flux | seguretat, utilització, manteniment |
| Dades de prova | mida, nivell de protecció | qualitat |

Taula 2: Atributs Interns i Externs

2.5 Factors a tenir en compte

Existeixen altres factors a tenir en compte a l'hora de determinar la productivitat d'un programari. Aquets factors són factors del projecte, factors de l'administració i factors del producte.

- **Factors del projecte:** els factors a nivell de projecte més comuns són:
 - **Personal.** Els factors personals poden ser importants a l'hora de determinar la qualitat i cost d'un programari. Així, un desenvolupador amb un bon nivell acadèmic y/o una experiència professional extensa, per norma general, és molt més productiu que un desenvolupador novell o sense formació acadèmica.

- **Ambient de desenvolupament.** Aquest factor és important a l'hora de que tots els membres d'un projecte pugin realitzar les tasques d'una forma ordenada. Així, és molt important que la metodologia de programació sigui homogènia, la documentació estigui basada en plantilles ja definides prèviament, hagi un bon control de versions instal·lat, etc.
- **Administració:** Els factors a nivell administratiu seran factors com la estabilitat del grup de projecte, els costos fixes i variables requerits en la totalitat de l'execució del projecte, etc.
- **Producte:** A nivell dels factors del producte s'han de tenir en compte variables importants com poden ser la qualitat del propi producte, la innovació en aquest, la tecnologia utilitzada pel seu desenvolupament, la complexitat tant en la programació com en el model de dades, etc.

3 MÈTRIQUES

3.1 Mètriques del Procés

Les mètriques del procés es recopilen de tots els projectes y durant un llarg període de temps. La seva intenció és proporcionar indicadors que portin a millorar els processos de software a llarg termini. Un indicador de mètrica o una combinació de mètriques que proporcionen una visió profunda del procés de software, del projecte de software o del producte en si.

Les mètriques permeten tenir una visió profunda del procés de software que ajudarà a tomar decisions més fonamentals, ajuden a analitzar el treball desenvolupat, conèixer si se ha millorat o no respecte a projectes anteriors, ajuden a detectar àrees amb problemes per poder pal·liar-los a temps i a realitzar millors estimacions.

Per millorar el procés s'ha de mesurar els atributs del mateix, desenvolupar mètriques d'acord amb aquests atributs i utilitzar-les per proporcionar indicadors que portin a millorar el procés. Els errors detectats abans de la entrega del software, la productivitat, els recursos i temps consumit i l'ajust amb la planificació són alguns dels resultats que poden mesurar-se en el procés.

Les mètriques del procés es caracteritzen per:

- El control i execució del projecte.
- Mesura de temps de anàlisi, disseny, implementació i post implantació.
- Mesura de les proves
- Mesura en la transformació i evolució del producte.

El fet de que ens trobem amb una gran quantitat d'errors durant les proves indica que s'han comès molts errors durant el desenvolupament, i, si no es fa un esforç suplementari en la fase de proves, part d'aquests errors es portaran fins que el programari estigui en producció.

Durant el procés de construcció d'un programari, una possible forma de mesurar la facilitat amb la qual pot ser modificat per corregir errors, millorar el seu funcionament o fer altres canvis, és mesurar la freqüència d'errors deguts a efectes laterals produïts després d'una modificació (X):

$$X = \frac{1 - A}{B}$$

A = nombre de errades deguts a efectes laterals detectats i corregits.

B = nombre total d'errors corregits.

Quan més gran sigui X és pot preveure que serà més difícil de mantenir en el futur un software.

3.2 Mètriques del Projecte

Les mètriques del projecte són les mesures de alguna propietat de un entregable d'un projecte o del procés de administració de projectes, efectuada per conèixer la situació i les desviacions respecte al pla original.

Les mètriques del projecte poden ser utilitzades per mesurar l'estat, efectivitat o progrés de les activitats d'un projecte i, així contribuir a prendre decisions estratègiques envers els desviaments, incidències o diferents problemes que apareixen durant l'execució.

Les mètriques del projecte tenen una doble finalitat:

- Minimitzar el temps de desenvolupament:
- Valorar la qualitat del producte. I s'estableixen, posteriorment, les possibles millores.



Figura 3: Mètriques de projecte

Es aconsellable que les mètriques del projecte es revisin periòdicament per poder determinar, valorar i actuar sobre possibles desviacions que es puguin produir durant tota la vida d'un projecte.

3.3 Mètriques del Programari

Les mètriques del programari fan referència al conjunt de mètriques del producte i del procés.

Els gestors de projecte dedicats al desenvolupament de software poden establir objectius significatius de millora del procés d'enginyeria del programari. El programari és un punt de administració estratègic per a moltes companyes, per tant, si el procés de desenvolupament de programari pot millorar, es produirà una millora a la companyia amb seguretat.

Tècnicament, les mètriques del programari, quan s'apliquen al producte, subministren beneficis immediats. Quan s'ha terminat el disseny del software, la majoria de les persones que desenvolupen poden estar amb ganes de trobar preguntes com:

- Que requisits del usuari són més susceptibles al canvi?
- Que mòduls del sistema són més propicis a error?
- Cóm s'ha de planificar la prova per a cada mòdul?
- Quants errors poden haver quan les proves comencin?

Tot aquestes respostes les poden trobar amb més exactitud si fem tres punts importants:

- **Recopilació de dades:** Recopilarem dades en base a projectes anteriors per reconstruir tot el necessari.
- **Càlcul de mètriques:** Depenent de l'amplitud de les mides recopilades, les mètriques poden ser molt amples, tals com: LDC y PF, així com mètriques de qualitat i orientades objectes.

- **Avaluació de mètriques:** La avaluació d'aquestes mètriques ens farà portar un control del projecte i millorar els punts on els indicadors ens diuen que poden fer una millor tasca.

3.4 Mètriques de Productivitat

Les mètriques de productivitat es centren en el rendiment del procés en la enginyeria del software, es a dir, avalua quan de productiu va a ser un software que es vol desenvolupar.

Les mètriques de productivitat, en aquest document, es un tema molt important a tractar en aquest TFC, per tant, el desenvoluparem amb profunditat en el següent punt.

4 MÈTRIQES DE PRODUCTIVITAT

4.1 Introducció

Com ja s'ha comentat en el punt anterior, les mètriques de productivitat es centren en el rendiment del procés en la enginyeria del programari.

Al planificar un projecte s'han de tenir en compte estimacions de temps, cost i d'esforç humà requerit. Així, mitjançant les mides de software, recollirem les dades qualitatives del programari desenvolupat i els seus processos, d'aquesta manera, una vegada analitzades les dades, podem disminuir el temps, els costos i els esforços humans.

En un principi podria semblar que la necessitat de la medició es evident, després de tot, és el que permet quantificar i, amb conseqüència, gestionar de forma més efectiva. Però la realitat es molt més diferent. Freqüentment la medició porta a una gran controvèrsia i discussió. Moltes vegades ens sorgeixen dubtes com:

- Quines són les mètriques apropiades per al procés i per el producte?
- Com s'ha d'utilitzar les dades que es recopilen?
- S'ha d'utilitzar mides per comparar persones, processos i productes?

El mesurament és molt comú en el món de la enginyeria, però el mesurament resta molt llunyana en el món de la enginyeria del programari. Trobem dificultats en posar-nos d'acord sobre què mesurar i com avaluar les mides.

Els pròxims punts farem una referència a una gran quantitats de factors que ens permeten mesurar la productivitat d'un programari, i així, determinar la forma de millorar el seu desenvolupament en temps, costos, recursos i altres factors que ens

permetin treure al mercat un producte amb garanties suficients per aconseguir una satisfacció d'usuari molt ampla.

4.2 Raons per mesurar un producte

Una mètrica de un projecte és la mida d'alguna propietat d'una entrega del projecte o procés efectuada per a conèixer l'avanç o el desviament al pla original.

Les mètriques poden ser utilitzades per a mesurar l'efectivitat, l'estat o progrés de les activitats d'un projecte i, així, poder prendre decisions estratègiques davant possibles incidents, desviaments o problemes que poden sorgir durant la vida d'un projecte.

Així podem enumerar varies raons per mesurar un producte:

- Per establir una línia base per a la estimació.
- Per avaluar la productivitat del equip que desenvolupa el producte.
- Per avaluar els beneficis parlant de productivitat i de qualitat del producte, a conseqüència de utilitzar nous mètodes i eines de la enginyeria del programari.
- Per a indicar la qualitat del producte.
- Per justificar l'ús de noves eines.
- Per realitzar formació addicional necessària per un bon desenvolupament de l'equip de treball.

4.3 Necessitat de les mètriques de productivitat

Les mètriques de productivitat són molt importants per diversos factors que podem veure a continuació:

- Pot servir com un vocabulari comú entre les persones que participen en l'equip de desenvolupament i els patrocinadors o stakeholders.
- Pot ajudar a identificar tendències i successos importants en els projectes i, així, permetre la organització de informació, molt útil per prendre decisions durant el transcurs del projecte.
- Pot ajudar a motivar el grup de treball ja que el esforç personal dels membres es pot relacionar amb els resultats del projecte.

Hi han quatre importants variables que podem augmentar o disminuir en funció de les necessitats per gestionar un projecte:

- **Abast del producte.** El nombre total de funcionalitats a desenvolupar.
- **Qualitat.** El nombre de errors totals del software desenvolupat. Es una dada fiable.
- **Cost.** La mesura més utilitzada per determinar el cost és a partir del nombre de recursos (persona/hora).
- **Temps.** Terminis i dates estimades.

4.4 Mètriques aplicades a la gestió de projectes

A continuació farem una mostra de les diferents mètriques que es poden utilitzar per mesurar un projecte. Aquestes mètriques les dividirem en varis tipus:

- Requisits: Mètriques en base a un requisits previs.
- Mida de la aplicació: Mesurem la aplicació per determinar aquesta mètrica.
- Estimació d'esforç: Les mètriques d'esforç es basen en les mesures persona/mes.
- Personal: Es mesuren factors com la quantitat de persones del projecte, la qualitat de les persones i la experiència de les mateixes.
- Mètodes i eines: Mètriques per determinar si les eines i la metodologia són les correctes.
- Temps: Aquesta es una mètrica molt exacta basada en el temps estimat.
- Grau de rigor del projecte: Determinar si el rigor del projecte es correcte.
- Control de canvis: Es fa la mida del risc del projecte.

A continuació farem una descripció més detallada de tots aquetes mètriques.

4.4.1. Requisits

4.4.1.1 Validació de REQ (VR)

La validació de requisits es una de les mètriques més utilitzades actualment, ja que es una forma de mesura molt exacta.

L'objectiu bàsic de la validació de requisits es la de examinar i assegurar que els requisits proposats pel client han estat establerts sense ambigüitat, sense inconsistències i sense omissions.

Es molt utilitzat aquest tipus de mètrica en base a un diagrama de classes prèviament establert amb el client. Així, les dos dades requerides prèviament per a poder utilitzar aquesta mètrica són els requeriments del client i el diagrama de classes. Aquestes dos dades es poden creuar en una taula, tal com es mostra a continuació, i d'aquesta forma es pot veure si el requeriment ha complert amb les expectatives:

| | CLASSES | | | |
|-----------|----------|----------|-----|----------|
| REQUISITS | CLASSE 1 | CLASSE 2 | ... | CLASSE n |
| R 1 | ✓ | | | |
| R 2 | | ✓ | | |
| ... | | | ✓ | |
| R n | | | | ✓ |

Taula 3: Validació REQ

4.4.2. Mida de la aplicació

4.4.2.1 Classes Clau (CC)

Les mètriques a partir de les Classes Clau es fan en base a la quantitat de classe principals que conté la aplicació, es a dir, es farà un recompte, a nivell del diagrama de classes, de les classes principals. La quantitat de classes principals que conté el programari desenvolupat determinarà la grandària del projecte.

El nombre de Classes Clau és directament proporcional al nombre de classes identificades i considerades de vital importància per al client. Per aquest motiu, en el moment de determinar si una classe és clau o no, es serà de utilitat realitzar diverses preguntes com:

- Es pot desenvolupar la aplicació sense aquesta classe?
- El client pot considerar aquest objecte, element o funcionalitat, com a important?
- El diagrama de classes inclou aquesta classe?

Si hi ha cap pregunta d'aquesta que ens indiqui que la classe a valorar és prescindible llavors aquesta classe no es considerarà com a Classe Clau.

Segons estudis realitzats

4.4.2.2 Classes Suport (CS)

Les classes suport són classes que són necessàries per a implementar la totalitat del sistema, però no estan directament relacionades amb el domini del problema.

Les Classes Suport s'identifiquen com les classes que no són indispensables pel domini del problema, però proporcionen funcionalitats valuoses a les Classes Clau i les complementen.

Com a norma general, el nombre de Classes Clau són conegudes en les primeres etapes del projecte, però el nombre de Classes Suport es defineixen al llarg d'aquest.

4.4.2.3 *Classes Totals (CT)*

Les Classes Totals seran la suma de les Classes Clau i les Classes Suport. Aquesta mètrica s'utilitza, al inici del projecte, per obtenir una dada quantitativa i exacta de la mida total del projecte.

Com hem dit, les dades requerides per obtenir aquesta mètrica són: Classes Clau i Classes Suport:

$$CT = CC + CS$$

4.4.2.4 *Centenars de Instruccions Fonts (CIF)*

Les mètriques de Centenars de Instruccions Fonts es basen en mètriques que transformen el nombre total de classes en un nombre de Instruccions Fonts. Així calcularem la mida total del projecte en base a Instruccions Fonts.

Com la transformació a Instruccions Fonts es basa en les Classes Totals, aquest valor, es una dada bàsica per poder utilitzar aquesta mètrica. La transformació estarà basada en les lliçons apreses, es a dir, en la experiència d'altres projectes per poder saber, en base a la dificultat de la classe, quantes instruccions fonts necessitarem.

En aquesta mètrica, la experiència és un valor important per poder assolir una mida exacta, per tant, la persona o persones responsables de assolir aquests valors tindran una experiència prèvia en aquest tipus de projectes.

Per obtenir el valor de Centenars de Instruccions Fots (CIF) utilitzarem la quantitat de Classes Totals dividit per 100, ja que volem obtenir el valor en centenars:

$$CIF = CT / 100$$

4.4.3. Estimació d'esforç

4.4.3.1 Esforç del projecte

Quan parlem d'esforç del projecte ens referim a la suma dels temps que es dediquen els diferents recursos a una determinada activitat o a la totalitat del projecte. La seva unitat de mida es persona/hora, persona/mes,...

Per obtenir l'esforç del projecte és indiferent el treball que es realitzi de forma seqüencial per un sol recurs o en paral·lel per diferents persones, es fa la suma dels temps de cadascun per a obtenir l'esforç total.

Així, la tasca per saber l'esforç d'un determinat projecte és examinar els factors d'aquest projecte per quantificar l'esforç empleat pel desenvolupament. D'aquesta manera obtindrem un valor numèric persones/mes necessari per poder mesurar un projecte.

Per calcular l'esforç real (EsfReal) d'un projecte utilitzarem la següent equació:

$$\text{EsfReal (persona/mes)} = \text{EsfNom} * \text{FEC}$$

- EsfNom és l'Esforç Nominal es calcula de la següent forma:

$$\text{EsfNom} = 2.94 * \text{CIF}^B$$

- 2.94: És una constant.
- CIF: És el valor sortint del punt anterior de Centenars d'Instruccions Fonts.
- B: Per calcular aquest valor utilitzarem la següent equació:

$$B = 1.01 + 0.01 * \sum W_i$$

- W_i : És el sumatori de diferents valors utilitzats al sistema COCOMO 2, punt que veurem més endavant. Aquests valors són:

$$W_i = \text{PREC} + \text{FLEX} + \text{RESL} + \text{TEAM}$$

- El valor de FEC és la multiplicació de valors utilitzats al sistema COCOMO 2, que com hem dit abans, veurem amb més amplitud més endavant:

$$FEC = PERS * RUSE * PDIF * PREX * SCED * FCIL$$

4.4.3.2 Esforç amb reutilització

La reutilització de software es defineix com l'ús de qualsevol tipus de artefacte (o actiu), o part del mateix, creat amb anterioritat, en un nou projecte.

La mètrica de esforç amb reutilització té com a finalitat reduir la estimació de l'esforç d'un projecte utilitzant la reutilització de classes ja elaborades i provades prèviament en el mateix o un altre projecte.

La reutilització per tant ens ajuda a:

- **Reduir els costos del desenvolupament:** Només s'ha de realitzar una vegada una part de codi determinada.
- **Augmentar la qualitat dels productes:** Com és codi ja utilitzat també es codi ja provat, per tant, la qualitat augmenta.
- **Augmentar la productivitat:** Mitjançant la millora de temps en la que es desenvolupen els projectes informàtics.
- **Millorar el manteniment i suport de la aplicació:** No hi haurà tants problemes a poder realitzar canvis, ja que és codi conegut.
- **Evitar desviaments de projecte:** Millora el control i la planificació, per tant, evitem els desviaments de projecte.

El esforç amb reutilització el podem calcular amb la següent formula:

$$\text{Esforç amb Reutilització} = (\text{EsfReal} * (100 - \%r)) / 100$$

- EsfReal és l'esforç real vist en el punt anterior.
- %r: És el percentatge de reutilització que tenim en el projecte a mesurar. Una forma d'aconseguir aquest valor és mesurant la quantitat de classes reutilitzades en base a la totalitat de classes del projecte.

4.4.4. Personal

4.4.4.1 Mida del Equip de Treball (E.T.)

La mida de l'Equip de Treball té com a objectiu predir el nombre d'elements necessaris pel desenvolupament del projecte, és a dir, la quantitat d'homes necessaris per a la realització d'un projecte.

La fórmula per elaborar aquesta mida és:

$$\text{Mida Equip de Treball (Quantitat persones)} = \text{Total Classes} / \text{Mitjana de Classes per persona}$$

4.4.4.2 Experts per l'Àrea

Aquesta mida facilitarà la tasca de saber si una persona indicada al projecte es la indicada per l'àrea. Amb els càlculs mostrats a continuació s'aconsegueix determinar el nivell d'eficiència per cada integrant i el nivell d'eficiència d'equip de treball.

Per determinar la experiència en l'àrea es realitzarà una taula com es mostra a continuació amb les diferents funcions de cada participant en el projecte i les seves activitats.

| PARTICIPANT | ACTIVITAT | PES | GRAU | NEFIC |
|------------------------|--|-----|------|-------|
| Cap de Projecte | Motivació per a l'equip tècnic | | | |
| | Habilitat per adaptar processos existents | | | |
| | Resolució de problemes | | | |
| | Control de gestió | | | |
| | Control mitjançant mètriques OO | | | |
| | Proporciona incentius pel reutilitzament | | | |
| | Planificació | | | |
| | Subtotal | | | |
| Analista - Programador | Interès pel ordre | | | |
| | Constància | | | |
| | Capacitat d'atenció i concentració | | | |
| | Innovació | | | |
| | Coneixements UML | | | |
| | Capacitat per la captura de requeriments | | | |
| | Tècniques de qualitat del software | | | |
| | Comunicació interpersonal | | | |
| Subtotal | | | | |
| Programador | Interès pel ordre | | | |
| | Constància | | | |
| | Capacitat d'atenció i concentració | | | |
| | Coneixements en llenguatges de programació | | | |
| | Subtotal | | | |

| | |
|--|--|
| Nivell d'eficiència de l'equip de treball | |
|--|--|

Taula 4: Funcions de cada participant en el projecte

La columna PES indica la importància que té la realització correcta d'aquesta tasca per al participant en qüestió. El pes es calcula com:

$$PES = \text{Importància en el participant} / 100$$

La Importància en el participant serà un valor del 1 al 100 i la suma de totes les importàncies d'un participant ha de sumar 100.

La columna GRAU s'omplirà en base al valor que aporta el participant en cada activitat:

- Molt baix = 0 Sense aptitud
- Baix = 2 Una mica de relació
- Normal = 5 Capacitat d'aplicar-lo, però no emfatitza
- Alt = 8 Altament cooperatiu
- Molt Alt = 10 Totalment aplicable (coneixements)

A la columna NEFIC s'utilitza la següent fórmula:

$$\text{Nivell d'Eficiència Activitat} = (PES * GRAU) / 10$$

El nivell d'Eficiència d'un participant serà:

$$\text{Nivell Eficiència Participant} = \sum \text{Nivell d'Eficiència Totes Activitats}$$

El nivell d'Eficiència de l'equip de treball serà:

$$\text{Nivell Eficiència E.T.} = \sum N.Ef.Participant / \text{Nombre de participants}$$

4.4.4.3 Experiència com E.T.

La experiència en el Equip de Treball ajuda a saber si els integrants del grup de treball elaboren amb eficiència les tasques a realitzar, tant de manera individual com de manera col·lectiva.

Moltes vegades les estimacions es fan valer de la experiència ja produïda com a única guia. Si un projecte es similar en mida és probable que el nou projecte calgui la mateixa quantitat d'esforç, però pot succeir que això no sigui així, o que el projecte tingui una mida diferent.

Així es farà servir una taula de mètriques com es mostra a continuació:

| D1 | D2 | VALOR | ESCALA | COMENTARI |
|-------|-------|-------|--------|-----------|
| P.Rea | P.Con | | | |
| P.Aca | P.Com | | | |
| P.Aca | P.Tie | | | |
| P.Com | P.Req | | | |
| P.Req | P.Qua | | | |

| | |
|--------------|--------------|
| P.Aca | P.Qua |
| ET | ES |
| PI | PD |
| PC | PsC |

Taula 5: Taula de mètriques

La columna D1 i D2 és la descripció de les següents variables de projecte:

| VARIABLE | DESCRIPCIÓ |
|--------------|--|
| P.Rea | Projectes Realitzats |
| P.Aca | Projectes Acabats |
| P.Com | Projectes que es van començar |
| P.Tie | Projectes acabats a temps |
| P.Req | Projectes que han complert amb els requeriments |
| P.Qua | Projectes que es van tenir en compte els factors de qualitat |
| EE | Errors trobats |
| ES | Errors solucionats |
| PI | Programadors amb iniciativa |
| PD | Programadors amb dependència del superior |
| PC | Persones al projecte amb confiança |
| PsC | Persones al projecte sense confiança |

Taula 6: Descripció de les variables de projecte

A la columna VALOR s'utilitzarà la següent formula:

$$\text{Valor} = D1 - D2$$

La columna ESCALA serà la dependència i el resultat que es busca en funció de la columna VALOR, així:

| VALOR | ESCALA |
|-------------------------|------------|
| 0 | Excel·lent |
| VALOR < D1/D2 | Bona |
| VALOR = D1/D2 | Mitja |
| VALOR > D1/D2 | Malament |

Taula 7: Relació VALOR/ESCALA

A la columna DESCRIPCIÓ s'expliquen els valors introduïts.

4.4.5. Mètodes i eines

4.4.5.1 Mètodes i eines

Una mètrica important de cara al bon funcionament d'un projecte és la de saber si els mètodes i les eines utilitzades són les adequades. Així, en base a una escala preestablerta, servirà per determinar la importància d'aquestes i si són indispensables pel el treball diari del projecte.

La taula següent ajuda a saber en quina escala esta les eines i els mètodes utilitzats per un projecte en base a què tan indispensable són aquestes:

| ESCALA | DESCRIPCIÓ |
|--------|---|
| 0 | No s'utilitzen eines |
| 1 | Ajuden en el 20% de la documentació |
| 2 | Per documentar menys del 50% del disseny de alt nivell |
| 3 | Per documentar al menys el 50% del disseny de alt nivell i disseny detallat |
| 4 | Per dissenyar i generar el codi de menys del 50% del sistema |
| 5 | Pel disseny i la generació del codi de al menys el 90% del sistema |

Taula 8: Escala de les eines i mètodes utilitzats

4.4.6. Temps

4.4.6.1 Persones-Dia-Classe

La mètrica Persones-Dia-Classe és, com el propi nom indica, una mètrica per determinar el nombre mig de dies d'esforç necessari per a una classe, i d'aquesta forma obtenir una dada estimada del temps de desenvolupament d'un projecte.

És una mètrica força utilitzada però que no sempre pot ser exacta ja que el temps en desenvolupar una classe determinada no ha de ser el mateix per desenvolupar una altra.

Per obtenir una mètrica més exacta de la duració del projecte es pot servir la formula següent:

$$\text{Temps Desenvolupament (dies)} = (\text{CT} * \text{Dia-Classe}) / \text{EsfReal}$$

CT és la mètrica vista anteriorment que indica les Classes Totals del projecte.

Dia-Classe indica el temps necessari per la realització d'una Classe.

EsfReal és la mètrica vista amb anterioritat que indicava l'esforç del projecte.

4.4.7. Grau del rigor del projecte

4.4.7.1 Rigor del projecte

El rigor del projecte ajuda a establir el nivell d'exigència amb el que serà tractat el procés de desenvolupament del projecte. Així, hi ha criteris de adaptació recomanables per la programació orientada a objectes.

La idea d'aquesta mètrica es poder determinar quan d'exigent és el projecte. Per això farem servir una formula que ens pot ajudar:

$$\text{Producte} = \text{Grau} * \text{Pes}$$

Com s'ha mencionat en un punt anterior el pes es calcula:

$$\text{PES} = \text{Importància en el participant} / 100$$

I, també com s'ha dit, el Grau es calcula en base al valor que aporta el participant en cada activitat:

- Molt baix = 0 Sense aptitud
- Baix = 2 Una mica de relació
- Normal = 5 Capacitat d'aplicar-lo, però no emfatitza
- Alt = 8 Altament cooperatiu
- Molt Alt = 10 Totalment aplicable (coneixements)

$$\text{Rigor} = \sum \text{productes} / \text{N}^{\circ} \text{ de criteris}$$

El nombre de criteris vindrà donat pels criteris d'adaptació per participant.

El resultat obtingut del Rigor es contrasta amb la taula següent per saber el rigor del projecte:

| VALOR RIGOR | RIGOR DEL PROJECTE |
|---------------------|--------------------|
| Rigor < 1.2 | Casual |
| 1.2 <= Rigor <= 2.4 | Estructurat |
| Rigor > 2.4 | Estricte |

Taula 9: Rigor del projecte

4.4.8. Control de canvis

4.4.8.1 Impacte de Risc

La mètrica sobre riscos és una mètrica força simple i que ajuda a corregir possibles desviaments que es poden tenir en el projecte, o si es dona el cas que ja ha acabat, pot ajudar a aprendre per no cometre les mateixes errades en els pròxims projectes.

Per determinar els riscos d'un projecte es farà una llista amb tots els riscos possibles en una taula com es mostra a continuació:

| RISCOS | PROBABILITAT | IMPACTE |
|------------------------------------|--------------|---------|
| Van haver canvis en les prioritats | | |
| Van sortir treballs no programats | | |
| Els proveïdors es van retardar | | |

| |
|--|
| Lentitud en les decisions |
| Els requeriments no eren prou clars |
| El equip de treball no tenia experiència |

Taula 10: Riscos d'un projecte

La columna PROBABILITAT s'omplirà amb un valor entre 0 y 100%.

La columna IMPACTE es classificarà en funció dels següents valors:

- 1 = Crític
- 2 = Alt
- 3 = Mig
- 4 = Baix

La suma de tots els impactes dels riscos serà la totalitat del risc del projecte. Quan més alt sigui la suma menys risc al projecte hi haurà.

4.4.9. Conclusions

Totes les mètriques vistes anteriorment ens poden ajudar a elaborar quantificar un projecte en temps, costos i recursos, però no són mètriques exactes. En tot el cicle de vida d'un projecte hi contribueixen molts factors que no sempre es poden quantificar, per aquest motiu, aquestes mètriques ens poden ajudar però no són mètriques perfectes.

Moltes vegades, en base a moltes variables que poden sorgir en el projecte, pot interessar tenir mètriques pròpies segons la necessitat de cada organització. Per aquest motiu hi ha vegades que sorgeixen nous processos de medicació de software aplicables a qualsevol activitat dins de la Enginyeria del Software.

En els pròxims punts veurem una sèrie de models matemàtics que utilitzen diverses mètriques que ens poden ajudar també a fer una estimació del cicle de vida de un projecte amb molta precisió.

5 PROGRAMARI PER MÈTRIQUES

5.1 COCOMO 81

5.1.1. Descripció del model COCOMO

COCOMO (COConstructive Cost Model) és el model de construcció de costos més conegut i utilitzat dels models algorítmics compostos que es basen en les dades estadístiques. També es basa en equacions analítiques i en un ajustament en base a la opinió dels experts. COCOMO va ser desenvolupat per Barry M. Boehm l'any 1981 i es va convertir en el model més conegut i referenciat, a més a més del més documentat de tots els models d'estimació.

El defineixen tres models de desenvolupament en funció dels tipus de projectes:

- **Orgànic:** Projectes senzills. Menors de 50 milers de línies de codi (KLOC), on es té una experiència de projectes similar i es troba en entorns estables.
- **Semi-acoplat:** Projectes mitjans en complexitat i mida (menors de 300 KLOC), on la experiència en aquest tipus de projectes es variable i les restriccions ni molt altes ni molt baixes.
- **Empotrat:** Projectes molt complexos, on no hi ha gaire experiència i són desenvolupats amb una innovació tècnica important. Els requisits són restrictius i molt volàtils.

Una altra forma de classificar COCOMO és mitjançant tres models diferents segons el seu grau de complexitat:

- **COCOMO Bàsic:** És un model estàtic vàlid per obtenir una estimació ràpida del esforç (persona-mes) en funció del milers de línies de codi (KLOC) al inici del cicle de vida.
- **COCOMO Intermedi:** Afegeix al càlcul del esforç en funció de la mida el efecte d'uns atributs que tenen relació amb el cost (CDA). Amb això, es vol tenir en compte el tipus de aplicació i tecnologia, les qualificacions i la experiència del personal, el entorn de disseny i programació i les eines de les que es disposen.
- **COCOMO Avançat:** El model avançat incorpora totes les característiques de la versió del model Intermedi però, en lloc de avaluar els CDA amb un únic valor per tot el cicle de vida del projecte, té en compte diferents CDA per cada fase del desenvolupament de software.

El model d'esforç general aplicable a tots els nivells de aplicació i modes esta donat per la formula:

$$E = a (EDSI)^b * (EAF)$$

Els valors de la forma els expliquen a continuació:

- **E:** És el resultat de l'esforç calculat expressat en persona-mes.
- **EDSI:** És el nombre estimat de milers de línies de codi.
- **a, b:** Són constants que depenen del tipus de projecte que estem fent la calculació d'esforç.

- EAF: És el factor de ajustament d'esforç i dependrà de si utilitzem el model Bàsic, Intermedi o Avançat.
 - EAF = 1 serà pel model Bàsic de COCOMO.
 - EAF = producte de 15 factors de cost. Es utilitza pel model Intermedi i Avançat.

Els coeficients pels models de desenvolupament són els següents:

| MOEL DE DESENVOLUPAMENT | a | b | c | d |
|-------------------------|-----|------|-----|------|
| Orgànic | 3,2 | 1,05 | 2,5 | 0,38 |
| Semi-aclopat | 3,0 | 1,12 | 2,5 | 0,35 |
| Empotrat | 2,8 | 1,20 | 2,5 | 0,32 |

Taula 11: Coeficients pels models de desenvolupament

El temps de desenvolupament requerit pel projecte en mesos és:

$$TDEV = c E^d$$

Els valors de la fórmula anterior són:

- E: És el esforç calculat a la fórmula anterior.
- c, d: Són constants que depenen del tipus de projecte mostrat a la taula anterior.

D'aquesta manera es pot calcular, ara, el nombre de programadors necessaris per la realització del projecte:

$$PG = E / TDEV$$

A les fórmules anteriors s'ha vist que per calcular l'esforç, tant en el model Intermedi com en el model Avançat, era necessari un factor d'ajustament basat en el producte de 15 paràmetres de cost. Aquests paràmetres són els següents:

| FACTORS DE COST | VALORACIÓ | | | | | |
|---|-----------|------|--------|------|----------|---------------|
| | Molt Baix | Baix | Normal | Alt | Molt Alt | Extraordinari |
| Factors de Software | | | | | | |
| Fiabilitat requerida del software | 0,75 | 0,88 | 1 | 1,15 | 1,4 | - |
| Mida de la base de dades | - | 0,94 | 1 | 1,08 | 1,16 | - |
| Complexitat del producte | 0,7 | 0,85 | 1 | 1,15 | 1,3 | 1,65 |
| Factors de hardware | | | | | | |
| Restriccions del temps d'execució | - | - | 1 | 1,11 | 1,3 | 1,66 |
| Restriccions de l'emmagatzematge principal | - | - | 1 | 1,06 | 1,21 | 1,56 |
| Volatilitat de la màquina virtual | - | 0,87 | 1 | 1,15 | 1,3 | - |
| Temps de resposta de l'ordinador | - | 0,87 | 1 | 1,07 | 1,15 | - |
| Factors de personal | | | | | | |
| Capacitat de l'analista | 1,46 | 1,19 | 1 | 0,86 | 0,71 | - |
| Experiència en l'aplicació | 1,29 | 1,13 | 1 | 0,91 | 0,82 | - |
| Capacitat dels programadors | 1,42 | 1,17 | 1 | 0,86 | 0,7 | - |
| Experiència en el Sistema Operatiu utilitzat | 1,21 | 1,1 | 1 | 0,9 | - | - |
| Experiència en el llenguatge de programació | 1,14 | 1,07 | 1 | 0,95 | - | - |
| Factors de projecte | | | | | | |
| Pràctiques de programació modernes | 1,24 | 1,1 | 1 | 0,91 | 0,82 | - |
| Utilització de eines de software | 1,24 | 1,1 | 1 | 0,91 | 0,83 | - |
| Limitació de planificació del projecte | 1,23 | 1,08 | 1 | 1,04 | 1,1 | - |

Taula 12: Factors de Cost. Model COCOMO 81

Els factors de cost s'han de elegir entre una de les sis possibilitats: Molt baix (MB), Baix (B), Normal (N), Alt (A), Molt Alt (MA), Extraordinari (XA). No tots els factors són vàlids per a totes les possibilitats, per això, alguns valors estan amb guió (-).

A moltes explicacions del model COMOMO ens podem trobar amb la abreviació dels Factors de Cost. A continuació es llistarà aquesta abreviació per poder tenir-la present:

| FACTORS DE COST | ABREVIACIÓ |
|--|------------|
| Fiabilitat requerida del software | RELY |
| Mida de la base de dades | DATA |
| Complexitat del producte | CPLX |
| Restriccions del temps d'execució | TIME |
| Restriccions de l'emmagatzematge principal | STOR |
| Volatilitat de la màquina virtual | VIRT |
| Temps de resposta de l'ordinador | TURN |
| Capacitat de l'analista | ACAP |
| Experiència en l'aplicació | AEXP |
| Capacitat dels programadors | PCAP |
| Experiència en el Sistema Operatiu utilitzat | VEXP |
| Experiència en el llenguatge de programació | LEXP |
| Pràctiques de programació modernes | MODP |
| Utilització de eines de software | TOOL |
| Limitació de planificació del projecte | SCED |

Taula 13: Abreviació dels Factors de Cost. Model COCOMO 81

Així el factor d'ajustament d'esforç (EAF) serà el producte de tots els factors seleccionats en aquesta taula.

5.1.2. Avantatges i desavantatges

Les principals avantatges del model COCOMO són els següents:

- COMOCO és un model transparent, es a dir, es pot comparar amb altres models com SLIM (Software Life Cycle Management).
- Els diferents factors de cost poden ajudar al estimador a determinar el impacte de diferents factors que afecten al cost del projecte.

Les desavantatges del model COCOMO són varies:

- Una bona estimació depèn de la adaptació del model a les necessitats de la organització, utilitzant dades històriques, que no sempre estan disponibles.
- El valor EDSI es difícil de estimar amb precisió sobre un projecte.
- EDSI, no es una mida de la grandària, si no una mida de la longitud.

5.2 ADA COCOMO

Al model d'estimació de costos COCOMO 81, va continuar la actualització per al llenguatge de programació Ada en 1987, aquesta actualització va rebre el nom de ADA COCOMO.

Els principals objectius de ADA COCOMO van ser:

- Determinar l'efecte del llenguatge de programació ADA en els costos i temps de desenvolupament. Es va comprovar que els Factors de Cost quan s'utilitzava aquest llenguatge de programació eren diferents que quan s'utilitzava altres tipus de llenguatges.
- Determinar l'efecte del model ADA en els costos i temps de desenvolupament. Així es plasmava el model ADA en el model ADA COCOMO.
- Incorporar les millores del model COCOMO realitzades posteriorment a la creació del model COCOMO 81.

Les diferències entre els Factors de Cost de ADA COCOMO i els Factors de Cost del COCOMO 81 són:

- Dos noves variables s'han afegit: Reutilització Requerida (RUSE) i Aplicació de Seguretat Classificada (SECU).
- La variable Volatilitat de la Màquina Virtual (VIRT) s'ha dividit en Volatilitat de la Màquina Virtual – Principal (VMVH) i Volatilitat de la Màquina Virtual – Destí (VMVT).
- Diverses variables han canviat el seus valors.
- Canvis complementaris incorporats en el model ADA COCOMO.

La taula de Factors de Cost adaptada al nou sistema ADA COCOMO és la següent:

| FACTORS DE COST | VALORACIÓ | | | | | | |
|---|-----------|------|--------|------|----------|---------------|------------------|
| | Molt Baix | Baix | Normal | Alt | Molt Alt | Extraordinari | XX Extraordinari |
| Factors de Software | | | | | | | |
| Fiabilitat requerida del software | 0,75 | 0,88 | 0,96 | 1,07 | 1,24 | - | - |
| Mida de la base de dades | - | 0,94 | 1 | 1,08 | 1,16 | - | - |
| Complexitat del producte | 0,73 | 0,85 | 0,97 | 1,08 | 1,22 | 1,43 | - |
| Reutilització Requerida | - | - | 1 | 1,1 | 1,3 | 1,5 | - |
| Factors de hardware | | | | | | | |
| Restriccions del temps d'execució | - | - | 1 | 1,11 | 1,3 | 1,66 | - |
| Restriccions de l'emmagatzematge principal | - | - | 1 | 1,06 | 1,21 | 1,56 | - |
| Volatilitat de la màquina virtual - Principal | - | 0,92 | 1 | 1,09 | 1,17 | - | - |
| Volatilitat de la màquina virtual - Destí | - | 0,93 | 1 | 1,07 | 1,16 | - | - |
| Temps de resposta de l'ordinador | 0,79 | 0,87 | 1 | 1,07 | 1,15 | - | - |
| Factors de personal | | | | | | | |
| Capacitat de l'analista | 1,57 | 1,29 | 1 | 0,8 | 0,61 | - | - |
| Experiència en l'aplicació | 1,29 | 1,13 | 1 | 0,91 | 0,82 | - | - |
| Capacitat dels programadors | 1,3 | 1,12 | 1 | 0,89 | 0,8 | - | - |
| Experiència en el Sistema Operatiu utilitzat | 1,21 | 1,1 | 1 | 0,9 | - | - | - |
| Experiència en el llenguatge de programació | 1,26 | 1,14 | 1,04 | 0,95 | 0,86 | - | 0,62 |
| Factors de projecte | | | | | | | |
| Pràctiques de programació modernes | 1,24 | 1,1 | 0,98 | 0,86 | 0,78 | - | - |
| Utilització de eines de software | 1,24 | 1,1 | 1 | 0,91 | 0,83 | 0,73 | - |
| Limitació de planificació del projecte | 1,23 | 1,08 | 1 | 1 | 1 | - | - |
| Aplicació de Seguretat Classificada | - | - | 1 | 1,1 | - | - | - |

Taula 14: Factors de Cost. Model ADA COCOMO

Les abreviacions d'aquests factors són:

| FACTORS DE COST | ABREVIACIÓ |
|---|------------|
| Fiabilitat requerida del software | RELY |
| Mida de la base de dades | DATA |
| Complexitat del producte | CPLX |
| Reutilització Requerida | RUSE |
| Restriccions del temps d'execució | TIME |
| Restriccions de l'emmagatzematge principal | STOR |
| Volatilitat de la màquina virtual - Principal | VMVH |
| Volatilitat de la màquina virtual - Destí | VMVT |
| Temps de resposta de l'ordinador | TURN |
| Capacitat de l'analista | ACAP |
| Experiència en l'aplicació | AEXP |
| Capacitat dels programadors | PCAP |
| Experiència en el Sistema Operatiu utilitzat | VEXP |
| Experiència en el llenguatge de programació | LEXP |
| Pràctiques de programació modernes | MODP |
| Utilització de eines de software | TOOL |
| Limitació de planificació del projecte | SECD |
| Aplicació de Seguretat Classificada | SECU |

Taula 15: Abreviació dels Factors de Cost. Model ADA COCOMO

5.3 COCOMO INCREMENTAL

El model COCOMO INCREMENTAL va ser definit, pràcticament al mateix temps que el model ADA COCOMO.

El model COCOMO INCREMENTAL és una moderna alternativa pel tradicional model en cascada del desenvolupament de processos del programari. Aquest model, en comptes de modelar el software com un esforç simple per obtenir un producte simple, permet desenvolupar una sèrie de projectes de software concurrent i produir un producte intermedi.

Aquest model ens permet reduir el risc i permet entregar un producte inicial més fàcilment al client.

5.4 COCOMO II

5.4.1. Descripció del model COCOMO II

El model COCOMO II va ser incorporat l'any 1990. L'objectiu principal va ser adaptar el model COCOMO al les noves practiques de desenvolupament dels anys 90.

Així, es va aprofitar aquest model per realitzar altres mancances del anterior:

- Construir una base de dades de projectes de software que permeti l'ajustament continuat del model per poder augmentar la precisió d'aquest.
- Implementar una eina que utilitzi el model COCOMO II.
- Proveir un marc analític quantitatiu i un conjunt d'eines i tècniques que avaluaran el impacte de les millores tecnològiques de programari sobre els costos i temps en les diferents etapes del cicle de vida d'un desenvolupament.

El model COCOMO II presenta tres models: el model de Composició d'Aplicacions, el model de Disseny Anticipat i el model Post-Arquitectura.

Aquests models sorgeixen de la diversitat del mercat i els seus sectors. Aquesta diversitat pot representar-se de la següent forma:

| | | |
|--|-----------------------------------|---------------------------|
| Aplicacions desenvolupades per usuaris finals | | |
| Generadors d'Aplicacions | Aplicacions amb Components | Sistemes Integrats |
| Infraestructura | | |

- **Aplicacions desenvolupades per usuaris finals:** Són les aplicacions de processament de informació generades directament per usuaris finals. Aquestes aplicacions moltes vegades es generen amb plantilles de càlcul, sistemes de consultes, etc.
- **Generadors d'Aplicacions:** Són mòduls prèviament empaquetats per utilitzar els usuaris finals i programadors.
- **Aplicacions amb Components:** Són aplicacions preparades per ser resoltes per solucions prèviament empaquetades. Aquestes aplicacions són generades per un equip reduït de persones, en poques setmanes o mesos.
- **Sistemes Integrats:** Són sistemes de gran mida, amb una forta integració entre components i sense antecedents en el mercat per tenir com a base.
- **Infraestructura:** Engloba el desenvolupament de sistemes operatius, protocols de xarxes, sistemes de base de dades, etc.

Els tres models de COCOMO II permeten adaptar-se als diferents sectors descrits anteriorment i al tipus i quantitat d'informació disponible en cada etapa de cicle de desenvolupament.

El model de Composició d'Aplicacions es utilitza en el desenvolupament de software a partir de components prèviament empaquetats. En aquest cas, s'utilitzen Punts Objecte per la estimació de programari.

Per a la resta de sectors del mercat s'utilitza un model mixta que combina els tres models.

El model Composició d'Aplicacions s'utilitza pel desenvolupament de programari durant la etapa de prototipació.

El model Disseny Anticipat s'utilitza en les primeres etapes del desenvolupament en les qual es té poca informació. Ens és d'utilitat l'ús de Punts Funció, que veurem més endavant.

El mode Post-Arquitectura és més utilitzat en la etapa de desenvolupament i en la etapa de manteniment. Aquest model utilitza Punts Funció, 17 atributs anomenats Factors de Cost i 5 factors que determinen un exponent.

El esforç de qualsevol dels tres models explicats a continuació s'expressarà en persona-mes (PM).

5.4.1.1 Model de Composició d'Aplicacions

La formula proposada en aquest mòdul és:

$$PM = NOP / PROD$$

Els valors de la formula són:

- **NOP**: Nous Punts Objecte. És la mida del software a desenvolupar expressat en Punts Objecte.

Aquest concepte es calcula de mitjançant la següent formula:

$$NOP = OP * (100 - \%reutilització) / 100$$

Els valors d'aquesta nova formula són:

- **OP**: Punts Objecte: Mida del programari a desenvolupar expressat en Punts Objecte.
- **%reutilització**: Percentatge de reutilització que s'estima en un projecte.
- **PROD**: Significa la productivitat mitja en base al anàlisi de dades de projectes mostrada en la taula següent:

| Experiència i capacitat dels desenvolupadors | Molt baix | Baix | Normal | Alt | Molt Alt |
|--|-----------|------|--------|-----|----------|
| Maduresa i capacitat del ICASE | Molt baix | Baix | Normal | Alt | Molt Alt |
| PROD | 4 | 7 | 13 | 25 | 50 |

Taula 16: PROD. Model de Composició d'Aplicacions. COCOMO II

5.4.1.2 Model de Disseny Anticipat

El model de disseny anticipat s'utilitza principalment en etapes inicials d'un projecte, quan es coneix molt poc la mida del producte a desenvolupar. Si tenim en compte els diferents sectors, aquest model es pot utilitzar en productes desenvolupats en els sectors de Generadors d'Aplicacions, Sistemes Integrats i/o Infraestructura.

La forma per calcular aquest esforç es la següent:

$$PM_{\text{estimat}} = PM_{\text{nominal}} \times \prod_{i=1}^7 EM_i$$

$$PM_{\text{nominal}} = A \times (KSLOC)^B$$

Les variables de les formules són:

- PM_{estimat} : És l'esforç nominal ajustat per 7 factors.
- $KSLOC$: La mida del projecte a desenvolupar expressat en milers de línies de codi.
- A : És una constant amb un valor de 2,94
- B : És un factor que té a veure amb característiques econòmiques d'un projecte. Per calcular aquest valor s'utilitza la següent formula:

$$B = 1.01 + 0.01 \times \sum_{j=1}^5 W_j$$

- EM_i : Són els factors de cost que tenen efecte sobre l'esforç, també se denominen Multiplicadors d'Esforç (Effort Multipliers). Aquests multiplicadors són els 7 mostrats a continuació:
 - Del producte
 - RCPX: Confiabilitat i Complexitat del producte
 - RUSE: Reutilització Requerida
 - De la plataforma
 - PDIF: Dificultat de la Plataforma
 - Del personal
 - PERS: Aptitud del personal

- PREX: Experiència del personal.
- Del projecte
 - FCIL: Facilitats
 - SCED: Cronograma de Desenvolupament Requerit

Cada multiplicador es pot classificar en sis nivells diferents segons el seu impacte, des de “Extra Baix” fins “Extra Alt”. El pes de cada multiplicador nominal es 1.0. Si el factor provoca un efecte negatiu sobre l’esforç, el valor serà major que 1.0, en cas contrari serà inferior a 1.0. La taula següent mostra tots els multiplicadors i el valor corresponent en funció del efecte que provoca:

| | Extra Baix | Molt Baix | Baix | Nominal | Alt | Molt Alt | Extra Alt |
|------|------------|-----------|------|---------|------|----------|-----------|
| RCPX | 0,73 | 0,81 | 0,98 | 1 | 1,3 | 1,74 | 2,38 |
| RUSE | - | - | 0,95 | 1 | 1,07 | 1,15 | 1,24 |
| PDIF | - | - | 0,87 | 1 | 1,29 | 1,81 | 2,61 |
| PERS | 2,12 | 1,62 | 1,26 | 1 | 0,83 | 0,63 | 0,5 |
| PREX | 1,59 | 1,33 | 1,12 | 1 | 0,87 | 0,71 | 0,62 |
| FCIL | 1,43 | 1,3 | 1,1 | 1 | 0,87 | 0,73 | 0,62 |
| SCED | - | 1,43 | 1,14 | 1 | 1 | 1 | - |

Taula 17: Multiplicadors i Valors. Model Disseny Anticipat. COCOMO II

5.4.1.3 Model Post-Arquitectura

El model Post-Arquitectura és utilitzat quan la arquitectura del projecte esta completament definida. Aquest model és utilitzat durant el desenvolupament i manteniment de programari als sectors de Sistemes Integrats, Infraestructura i Generadors de Aplicacions.

La formula utilitzada es la mostrada a continuació i es basa en 17 factors multiplicadors:

$$PM_{estimado} = PM_{nominal} \times \prod_{i=1}^{17} EM_i$$

Els 17 factors multiplicadors s’expliquen a continuació:

- RELY: Confiabilitat requerida. Calcula que el producte compleixi satisfactòriament amb la funció que ha de realitzar i es respecti el temps execució del projecte.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|--------|--|---|--|--|-----------------------|-------|
| RELY | Inconvenients insignificants que només afecten als desenvolupadors | Mínimes pèrdues a l'usuari fàcilment recuperables | Pèrdues moderades al usuari recuperables sense grans inconvenients | Pèrdua financera elevada o inconvenient massiu a nivell de recursos humans | Vida humana en perill | |

Taula 18: RELY. Model Post-Arquitectura. COCOMO II

- DATA: Mida de la base de dades. L'esforç requerit per desenvolupar un programari esta relacionat amb la mida de la base de dades.

El valor de data es calcula en funció de la següent formula:

$$D/P = \text{Mida_BaseDades (Bytes)} / \text{Mida_Programa (SLOC)}$$

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|--------|-----------|---------------------------|-------------|----------------|----------|-------|
| DATA | | DB bytes/Pgm SLOC < 10 | 10<=D/P<100 | 100<=D/P/<1000 | D/P>1000 | |

Taula 19: DATA. Model Post-Arquitectura. COCOMO II

- CPLX: Complexitat del producte. El nivell que s'obté es la mitja de cadascuna de les 5 àrees o tipus d'operacions implicades i que es mostren a continuació:

| | Operacions de Control | Operacions computacionals | Operacions dependents dels dispositius | Operacions de administració de dades | Operacions de administració de interfases d'usuari |
|------------------|---|--|--|---|---|
| Molt Baix | Poques estructures sense encadenament: DO, CASE, IF_THEN_ELSE | Avaluació d'una expressió simple Exemple: A=B+C*(D-E) | Sentències de lectura/escriptura amb formats simples | Ajustaments simples de memòria principal | Generadors de reports. Formularis entrada simples. |
| Baix | Estructures encadenades senzilles | Avaluació d'expressions de complexitat moderada | Cap necessitat de dispositius especials pel processament I/O | Arxiu sense canvis d'estructures de dades | Ús de generadors de interfícies de usuari gràfiques simples |

| | | | | | |
|------------------|---|---|--|---|---|
| Normal | Ús majoritari de encadenament senzills | Ús de rutines estàndards de matemàtica i estadística | Processament d'I/O que inclou la selecció de dispositiu, processament d'errors i comprovació d'estat | Varis arxius d'entrada i només un arxiu de sortida. | Ús simple d'alguns dispositius |
| Alt | Programació estructurada amb alt grau de encadenament. | Anàlisi numèric bàsic: Interpolació, equacions diferencials,... | Operacions d'I/O a nivell físic | Disparadors simples activats per fluxos de dades | Ús de un conjunt de dispositiu Multimèdia, I/O de processament de veu |
| Molt Alt | Codificació recursiva. Control d'interrupcions amb prioritat fixa | Anàlisi numèric estructurat: Matrius de equacions. | Rutines pel control de interrupcions, emmascarament | Coordinació de base de dades distribuïdes. Disparadors complexos. | Multimèdia, gràfics dinàmics, gràfics en 2D i 3D. |
| Extra Alt | Planificació múltiple de recursos amb canvi dinàmic de prioritats | Anàlisi numèric no estructurat: Dades estocàstiques. | Codificació de dispositius dependents del temps. | Alt grau de acoblament, relacions dinàmiques, estructura de objecte,... | Multimèdia complexa. Realitat virtual |

Taula 20: CPLX. Model Post-Arquitectura. COCOMO II

- RUSE: Requeriments de reutilització. Aquest paràmetre és l'esforç addicional necessari per construir components que poden ser utilitzats més endavant en futurs desenvolupaments o en el mateix projecte.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|-----------|-----------------------------|--|---------------------------------------|---|---|
| RUSE | | Cap component reutilitzable | Reutilització dins del mateix projecte | Reutilització dins un mateix programa | Reutilització dins una mateixa línia de productes | Reutilització en múltiples línies de producte |

Taula 21: RUSE. Model Post-Arquitectura. COCOMO II

- DOCU: Documentació en les diferents etapes del cicle de vida. Els possibles valors de DOCU van des de Molt Baix (documentació que no cobreix algunes necessitats fins Molt Alt (documentació excessiva).

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|--------|-----------|------|--------|-----|----------|-------|
|--------|-----------|------|--------|-----|----------|-------|

| | | | | | |
|-------------|---|--|--|---|--|
| DOCU | Moltes necessitats del cicle de vida sense cobrir | Algunes necessitats del cicle de vida sense cobrir | Necessitats del cicle de vida cobertes justament | Necessitats del cicle de vida cobertes àmpliament | Necessitats del cicle de vida cobertes excessivament |
|-------------|---|--|--|---|--|

Taula 22: DOCU. Model Post-Arquitectura. COCOMO II

- PVOL: Volatilitat de la plataforma. Aquest paràmetre es utilitza per representar la freqüència dels canvis en la plataforma subjacent.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|-----------|---|---|--|---|-------|
| PVOL | | Un canvi principal cada 12 mesos. Un canvi menor tots els mesos | Canvi principal cada 6 mesos. Canvi menor cada 2 setmanes | Canvi principal cada 2 mesos. Canvi menor cada setmana | Canvi principal cada 2 setmanes. Canvi menor cada 2 dies. | |

Taula 23: PVOL . Model Post-Arquitectura. COCOMO II

- STOR: Restricció d'emmagatzematge principal. Aquest factor és una funció que representa el grau de restricció de emmagatzematge.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|-----------|------|--|-----|----------|-------|
| STOR | | | Ús de <=50% del percentatge total d'emmagatzematge | 70% | 85% | 95% |

Taula 24: STOR. Model Post-Arquitectura. COCOMO II

- TIME: Restricció del temps d'execució. Representa la restricció de temps d'execució aplicada sobre el programari.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|-----------|------|--|-----|----------|-------|
| TIME | | | Us de <=50% del temps de execució disponible | 70% | 85% | 95% |

Taula 25: TIME. Model Post-Arquitectura. COCOMO II

- ACAP: Capacitat del analista. Els atributs a considerar són la habilitat pel disseny, l'anàlisi, la correcta comunicació i la cooperació interpersonal.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|--------|-----------|------|--------|-----|----------|-------|
|--------|-----------|------|--------|-----|----------|-------|

| | | | | | |
|-------------|--------------|--------------|--------------|--------------|--------------|
| ACAP | 15 Percentil | 35 percentil | 55 percentil | 75 percentil | 90 percentil |
|-------------|--------------|--------------|--------------|--------------|--------------|

Taula 26: ACAP. Model Post-Arquitectura. COCOMO II

- PCAP: Capacitat del programador. És la habilitat del programador en la utilització d'eines actuals i la capacitat pel treball en grup i la comunicació.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|--------------|--------------|--------------|--------------|--------------|-------|
| PCAP | 15 Percentil | 35 percentil | 55 percentil | 75 percentil | 90 percentil | |

Taula 27: PCAP. Model Post-Arquitectura. COCOMO II

- PCON. Continuïtat del personal. S'avalua el grau de permanència anual del personal afectat en un projecte de software.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|-------------|-------------|-------------|------------|------------|-------|
| PCON | 48% per any | 24% per any | 12% per any | 6% per any | 3% per any | |

Taula 28: PCON. Model Post-Arquitectura. COCOMO II

- AEXP. Experiència a la aplicació. El Grau va des de Molt baix, representat amb una experiència menor a 2 mesos, fins a Molt Alt, amb una experiència de 6 o més anys.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|------------|-----------|--------|--------|----------|-------|
| AEXP | <= 2 mesos | <=6 mesos | 1 any | 3 anys | 6 anys | |

Taula 29: AEXP. Model Post-Arquitectura. COCOMO II

- PEXP: Experiència en la plataforma. És important conèixer la plataforma amb la qual es treballa i noves plataformes que poden anar sorgint perquè pot ajudar a millorar la feina diària.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|------------|-----------|--------|--------|----------|-------|
| PEXP | <= 2 mesos | <=6 mesos | 1 any | 3 anys | 6 anys | |

Taula 30: PEXP. Model Post-Arquitectura. COCOMO II

- LTEX: Experiència en el llenguatge i les eines. Aquest paràmetre fa la mida del nivell d'experiència del equip de treball en l'ús de llenguatge i eines a utilitzar.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|------------|-----------|--------|--------|----------|-------|
| LTEX | <= 2 mesos | <=6 mesos | 1 any | 3 anys | 6 anys | |

Taula 31: LTEX. Model Post-Arquitectura. COCOMO II

- TOOL: Ús d'eines de programari. Les eines de software poden anar des de l'ús d'eines només per la codificació, edició i depuració, fins eines molt potents amb la possibilitat, també, de realitzar anàlisi, documentació, etc.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|--|--|--|--|---|-------|
| TOOL | Eines que permeten editar, codificar i depurar | Eines simples amb poca integració amb el desenvolupament | Eines bàsiques integrades moderadament | Eines robustes integrades moderadament | Eines altament integrades amb els processos i mètodes | |

Taula 32: TOOL. Model Post-Arquitectura. COCOMO II

- SITE: Desenvolupament multi-ús. Aquest paràmetre identifica la disposició de l'equip de treball i el suport de comunicació.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------------------------|---------------|--------------------------------|-------------------------------------|----------------|--------------|---------------------------|
| SITE Ubicació Espacial | Internacional | Multi-ciutat i multi-companyia | Mateixa ciutat o àrea metropolitana | Mateix edifici | Centralitzat | Completament Centralitzat |

Taula 33: SITE Ub. Espacial. Model Post-Arquitectura. COCOMO II

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|--------|-----------|------|--------|-----|----------|-------|
|--------|-----------|------|--------|-----|----------|-------|

| | | | | | | |
|-------------------------|------------------------|---------------------------|-------|--|--|------------|
| SITE Comunicació | Alguns telèfons, mails | Telèfons individuals, FAX | Email | Comunicacions electròniques de banda ampla | Comunicacions electròniques de banda ampla, ocasionalment videoconferència | Multimèdia |
|-------------------------|------------------------|---------------------------|-------|--|--|------------|

Taula 34: SITE Comunicació. Model Post-Arquitectura. COCOMO II

- SCED: Cronograma requerit pel desenvolupament. Aquest factor fa la mida de la restricció en fases de temps imposta pel equip de treball.

| Factor | Molt Baix | Baix | Normal | Alt | Molt Alt | Extra |
|-------------|----------------|----------------|-----------------|-----------------|-----------------|-------|
| SCED | 75% del normal | 85% del normal | 100% del normal | 130% del normal | 160% del normal | |

Taula 35. SCED. Model Post-Arquitectura. COCOMO II

Així els factors d'ajustament, segons les opcions triades anteriorment es calculen amb la següent taula:

| | Molt Baix | Baix | Nominal | Alt | Molt Alt | Extra Alt |
|-------------|-----------|------|---------|------|----------|-----------|
| RELY | 0,82 | 0,92 | 1 | 1,1 | 1,26 | - |
| DATA | - | 0,9 | 1 | 1,14 | 1,28 | - |
| CPLX | 0,73 | 0,87 | 1 | 1,17 | 1,34 | 1,74 |
| RUSE | - | 0,95 | 1 | 1,07 | 1,15 | 1,24 |
| DOCU | 0,81 | 0,91 | 1 | 1,11 | 1,23 | - |
| PVOL | - | 0,87 | 1 | 1,15 | 1,3 | - |
| STOR | - | - | 1 | 1,05 | 1,17 | 1,46 |
| TIME | - | - | 1 | 1,11 | 1,29 | 1,46 |
| ACAP | 1,42 | 1,19 | 1 | 0,85 | 0,71 | - |
| PCAP | 1,34 | 1,15 | 1 | 0,88 | 0,76 | - |
| PCON | 1,29 | 1,12 | 1 | 0,9 | 0,81 | - |
| AEXP | 1,22 | 1,1 | 1 | 0,88 | 0,81 | - |
| PEXP | 1,19 | 1,09 | 1 | 0,91 | 0,85 | - |
| LTEX | 1,2 | 1,09 | 1 | 0,91 | 0,84 | - |
| TOOL | 1,17 | 1,09 | 1 | 0,9 | 0,78 | - |
| SITE | 1,22 | 1,09 | 1 | 0,93 | 0,86 | 0,8 |
| SCED | 1,43 | 1,14 | 1 | 1 | 1 | - |

Taula 36: Calculació dels Factors d'Ajustament. Model Post-Arquitectura. COCOMO II

5.4.1.4 Punts Funció

COCOMO II utilitza els Punts Funció i/o Línies de Codi Font (SLOC) com a base per a mesurar els models d'estimació de Disseny Anticipat i Post-Arquitectura.

La formula per calcular els Punts Funció és:

$$FP = UFP * TCF$$

On UFP: Punts Funció no Ajustats

TCF: Factor de Complexitat Tècnica.

Per calcular UFP, primer s'ha de tenir en compte i comptabilitzar els següents ítems:

- **Entrades Externes (Imputs):** Entrada de dades d'usuari o de control.
- **Sortides Externes (Outputs):** Sortida de dades d'usuari o de control.
- **Arxiu Lògic Intern (Arxius):** Engloba cada grup lògic de dades que es creat i utilitzat pel software.
- **Arxius Externs de Interfase:** Arxius compartits entre sistemes de software.
- **Sol·licituds Externes (Queries):** Combinació d'entrada i sortida on una entrada causa immediatament una sortida.

Una vegada s'han identificat i comptabilitzat tots els ítems es classifiquen segons el seu grau de complexitat en: Baix, Promig, Alt. Se li assigna un pes específic a cada ítem segons el tipus i el grau de complexitat i s'utilitza la següent formula per calcular l'UPF:

$$UFP = \sum_{i=1}^{15} (Quantitat_Items_Tipus_i) * (Pes_i)$$

La següent taula serveix per determinar els nivells de complexitat de cada tipus de ítem en funció del nombre i del tipus d'elements de dades i arxius involucrats:

| Arxius Lògics i Arxius Externs de Interfase | | | | Sortides i Consultes Externes | | | | Entrades Externes | | | |
|---|-------------------|--------|--------|-------------------------------|-------------------|--------|--------|-------------------|-------------------|--------|--------|
| Elements de Risc | Elements de dades | | | Tipus d'arxius | Elements de dades | | | Tipus d'arxius | Elements de dades | | |
| | 10-19 | 20-50 | 51+ | | 1-5 | 6-19 | 20+ | | 1-4 | 5-15 | 16+ |
| 1 | Baix | Baix | Promig | 0-1 | Baix | Baix | Promig | 0-1 | Baix | Baix | Promig |
| 2-5 | Baix | Promig | Alt | 2-3 | Baix | Promig | Alt | 2-3 | Baix | Promig | Alt |
| 6+ | Promig | Alt | Alt | 4+ | Promig | Alt | Alt | 3+ | Promig | Alt | Alt |

Taula 37: Nivells de Complexitat. Model Post-Arquitectura. COCOMO II

La següent taula mostra el pes del factor de complexitat:

| Tipus de funció | Pes del Factor de Complexitat |
|-----------------|-------------------------------|
|-----------------|-------------------------------|

| | Baix | Promig | Alt |
|--|------|--------|-----|
| Entrades Externes (Inputs) | 3 | 4 | 6 |
| Sortides Externes (Outputs) | 4 | 5 | 7 |
| Arxius Lògics Interns (Arxius) | 7 | 10 | 15 |
| Arxius Externs de Interfase (Interfases) | 5 | 7 | 10 |
| Consultes Externes (Queries) | 3 | 4 | 6 |

Taula 38: Pes del Factor de Complexitat. Model Post-Arquitectura. COCOMO II

Per calcular el factor de complexitat tècniques'utilitza la següent formula:

$$TCF = 0.65 + 0.01 \times \sum_{i=1}^{14} F_i$$

On els factors de F_i corresponen als següents factors:

- F1: Mecanismes de recuperació i backup.
- F2: Comunicació de dades.
- F3: Funcions de Processament Distribuït.
- F4: Performance.
- F5: Configuració utilitzada rigorosament.
- F6: Entrada de dades on-line.
- F7: Factibilitat Operativa.
- F8: Actualització d'arxius on-line.
- F9: Interfases Complexes.
- F10: Processament Intern Compleix.
- F11: Reutilització.
- F12: Facilitat en la instal·lació.
- F13: Suport de múltiples instal·lacions.
- F14: Facilitat de canvis.

Els pesos es consideren com es mostra a continuació:

- 0: Sense influencia.
- 1: Incidental.
- 2: Moderat.
- 3: Mig.
- 4: Significatiu.

- 5: Essencial.

5.4.1.5 Línies de Codi Font

Definir una Línia de Codi Font és complexa degut a les diferències conceptuals entre els diferents llenguatges de programació. Per pal·liar aquests problemes, el més comú es utilitzar un checklist mostrat a continuació que ens ajuda a unificar criteris en la definició de una Línia de Codi Font:

Definition Checklist for Source Statements Counts

Definition name: Logical Source Statements Date: _____
 (basic definition) Originator: COCOMO II

| Measurement unit: | Physical source lines | | | |
|---|--|-------------------------------------|---|-------------------------------------|
| | Logical source statements | <input checked="" type="checkbox"/> | | |
| Statement type | Definition <input checked="" type="checkbox"/> | Data Array <input type="checkbox"/> | | Includes Excludes |
| <i>When a line or statement contains more than one type, classify it as the type with the highest precedence.</i> | | | | |
| 1 Executable | Order of precedence → | | 1 | <input checked="" type="checkbox"/> |
| 2 Nonexecutable | | | | |
| 3 Declarations | | | 2 | <input checked="" type="checkbox"/> |
| 4 Compiler directives | | | 3 | <input checked="" type="checkbox"/> |
| 5 Comments | | | | |
| 6 On their own lines | | | 4 | <input type="checkbox"/> |
| 7 On lines with source code | | | 5 | <input type="checkbox"/> |
| 8 Banners and non-blank spacers | | | 6 | <input checked="" type="checkbox"/> |
| 9 Blank (empty) comments | | | 7 | <input checked="" type="checkbox"/> |
| 10 Blank lines | | | 8 | <input checked="" type="checkbox"/> |
| 11 | | | | |
| 12 | | | | |
| How produced | Definition <input checked="" type="checkbox"/> | Data array <input type="checkbox"/> | | Includes Excludes |
| 1 Programmed | | | | <input checked="" type="checkbox"/> |
| 2 Generated with source code generators | | | | <input type="checkbox"/> |
| 3 Converted with automated translators | | | | <input checked="" type="checkbox"/> |
| 4 Copied or reused without change | | | | <input checked="" type="checkbox"/> |
| 5 Modified | | | | <input checked="" type="checkbox"/> |
| 6 Removed | | | | <input type="checkbox"/> |
| 7 | | | | |
| 8 | | | | |
| Origin | Definition <input checked="" type="checkbox"/> | Data array <input type="checkbox"/> | | Includes Excludes |
| 1 New work: no prior existence | | | | <input checked="" type="checkbox"/> |
| 2 Prior work: taken or adapted from | | | | |
| 3 A previous version, build, or release | | | | <input checked="" type="checkbox"/> |
| 4 Commercial, off-the-shelf software (COTS), other than libraries | | | | <input type="checkbox"/> |
| 5 Government furnished software (GFS), other than reuse libraries | | | | <input type="checkbox"/> |
| 6 Another product | | | | <input checked="" type="checkbox"/> |
| 7 A vendor-supplied language support library (unmodified) | | | | <input type="checkbox"/> |
| 8 A vendor-supplied operating system or utility (unmodified) | | | | <input type="checkbox"/> |
| 9 A local or modified language support library or operating system | | | | <input checked="" type="checkbox"/> |
| 10 Other commercial library | | | | <input type="checkbox"/> |
| 11 A reuse library (software designed for reuse) | | | | <input checked="" type="checkbox"/> |
| 12 Other software component or library | | | | <input checked="" type="checkbox"/> |
| 13 | | | | |
| 14 | | | | |

Figura 4: Checklist per la definició d'una Línia de Codi Font

5.4.1.6 Conversió de Punts Funció (UFP) a Línies de Codi Font (SLOC)

Per determinar l'esforç nominal en el model COCOMO II s'ha de realitzar la conversió de Punts Funció (UFP) a Línies de Codi Font (SLOC). Això es realitza mitjançant la taula següent en funció del llenguatge de programació que s'utilitzi:

| Llenguatge | SLOC/Punt Funció |
|------------------------|------------------|
| Ada | 71 |
| AI Shell | 49 |
| APL | 32 |
| Assembler | 320 |
| Assembler (macro) | 213 |
| ANSI/Quick/Turbo Basic | 64 |
| Basic - Compilat | 91 |
| Basic - Interpretat | 128 |
| C | 128 |
| C++ | 29 |
| ANSI Cobol 85 | 91 |
| Fortran 77 | 105 |
| Forth | 64 |
| Jovial | 105 |
| Lisp | 64 |
| Modula 2 | 80 |
| Pascal | 91 |
| Prolog | 64 |
| Generador de Reports | 80 |
| Planilla de Càlcul | 6 |

Taula 39: SLOC/Punt Funció

5.5 Costar

Costar es una eina de software de estimació de costos basada en el model COCOMO II. Els caps de projecte i totes les persones que ho necessitin poden utilitzar Costar per produir les estimacions de projecte de duració, quantitat de recursos, esforç i cost.

A continuació es mostrarà una guia rapida de utilització d'aquest software mitjançant un exemple d'utilització.

5.5.1. La pantalla principal de Costar

La pantalla principal del software es la que se mostra a continuació. La pantalla principal mostra totes les dades relatives a l'estimació actual.

Costar crea automàticament una estimació anomenada "Estimate1" amb un sol component anomenat "Component". Aquesta estimació pren valors des del model per defecte COMOMO II.

Les característiques més importants d'aquesta finestra són: 1) els menús, 2) la barra d'eines de càlcul, 3) la barra de components, 4) El panell d'estimació automàtica, 5) Les fitxes.

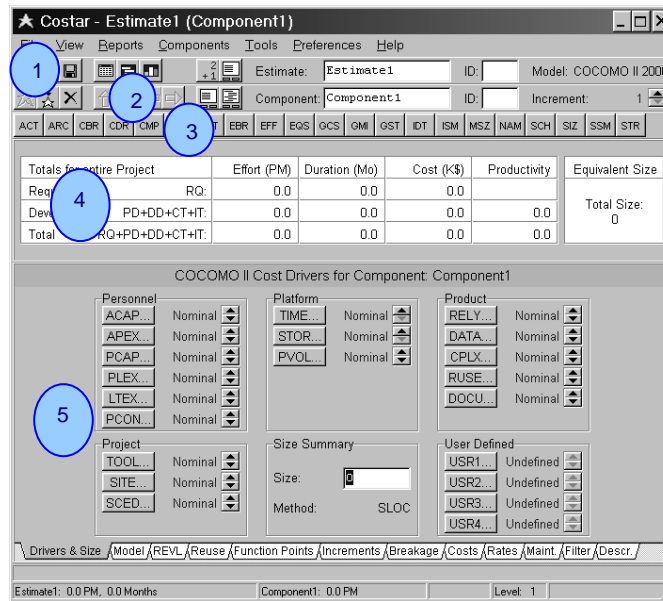


Figura 5: Pantalla principal de Costar

5.5.2. Introduir les línies de codi font

Costar ofereix diverses formes per especificar la mida de cada component de programari. Per aquest exemple, s'utilitzarà la part de "Size Summary", seleccionant l'opció SLOC (Method) amb una mida (SIZE) de 3000.

Una vegada introduïts els valors, el panell automàtic mostrarà un resum dels càlculs realitzats. Aquets càlculs els mostrarà al panell "Totals for entire Project". Aquests valors s'actualitzaran immediatament al canviar els paràmetres, modificar factors de cost, etc.

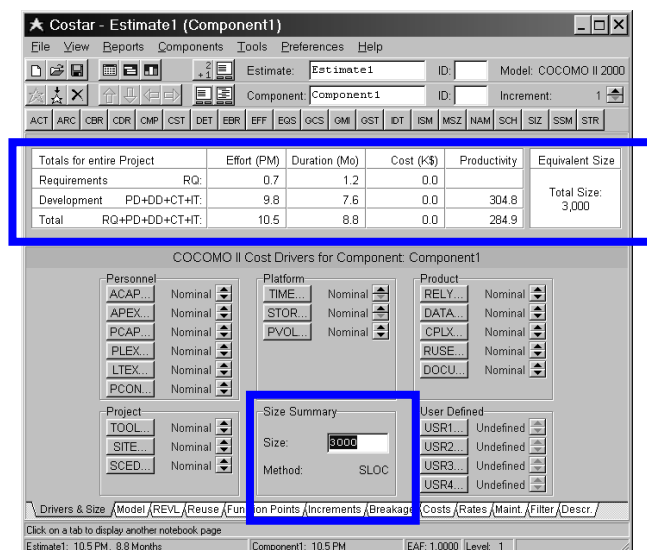


Figura 6: Introduir les línies de codi font

5.5.3. El editor de controlador de costos

La pantalla de l'Editor de Factor de Costos (Cost Driver Editor) permet modificar una de les qualificacions COMOCO estàndard. Els multiplicadors d'esforç de les qualificacions estàndard es mostren al costat de cada qualificació.

En el cas, per exemple, del multiplicador RELY la pantalla que es mostra serà com es veu a continuació:

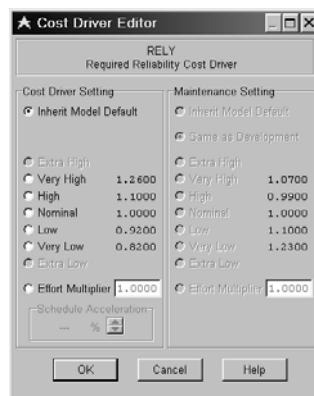


Figura 7: Editor de controlador de costos

5.5.4. Ajuda del editor de controlador de costos

Prengui F1 o faci clic al botó Ajuda (Help) de la finestra del factor de costos perquè aparegui el missatge d'ajuda de RELY.

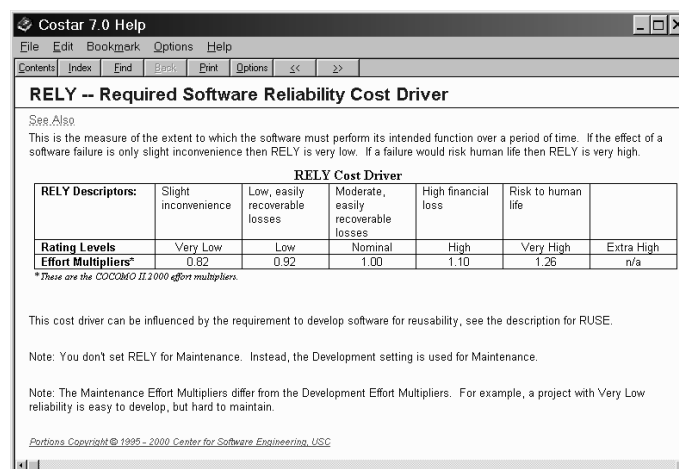


Figura 8: Ajuda del editor de controlador de costos

5.5.5. Actualització d'un Factor de Cost

Per actualitzar, per exemple, el factor PCAP pot ser de dos formes:

- Polsant en el botó del factor corresponent i actualitzant el valor a la pantalla resultant, com hem vist en punts anteriors.
- Fent clic a la fletxa cap a dalt o cap a baix just al costat del botó del factor corresponent, en aquest cas, just al costat del factor de cost PCAP.

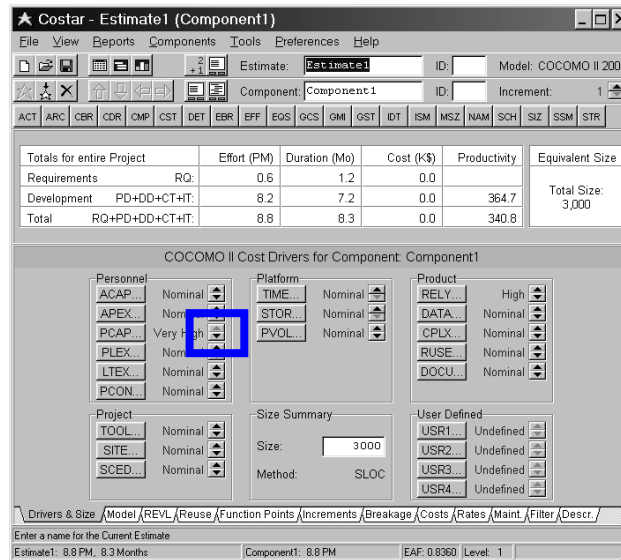


Figura 9: Actualització d'un Factor de Cost

5.5.6. Factors d'escala de COCOMO II

La fitxa de Model de Costar s'utilitza per seleccionar el model d'estimació que es vol utilitzar al projecte mitjançant el botó "Select COCOMO Model..." i per establir els factors d'escala d'aquest model.

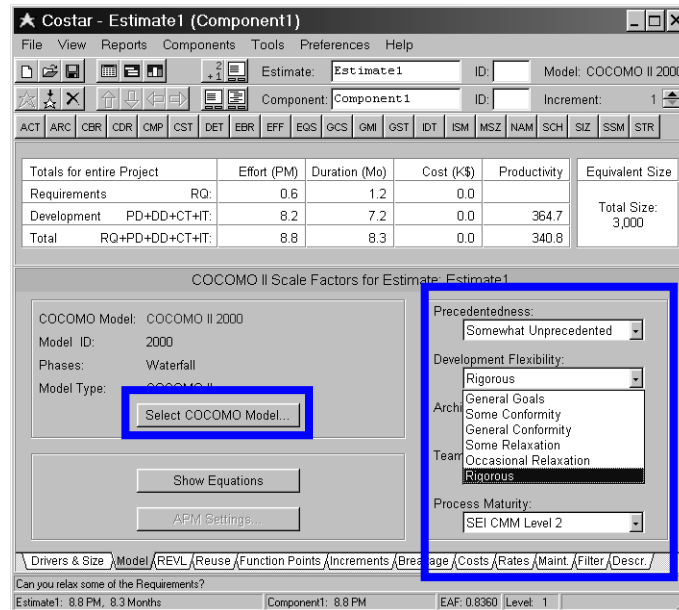


Figura 10: Factors d'escala de COCOMO II

5.5.7. Actualitzar els costos

A la fitxa de "Costs" introduïrem els costos del projecte. Cada camp indica els costos per a una persona durant un mes per a una de les fases de COCOMO.

S'ha de tenir en compte que Costar torna a calcular les estimacions cada vegada que es canvia qualsevol d'aquests valors.

Al exemple es pot veure com el projecte del programari calculat s'estima un cost de 54.000\$.

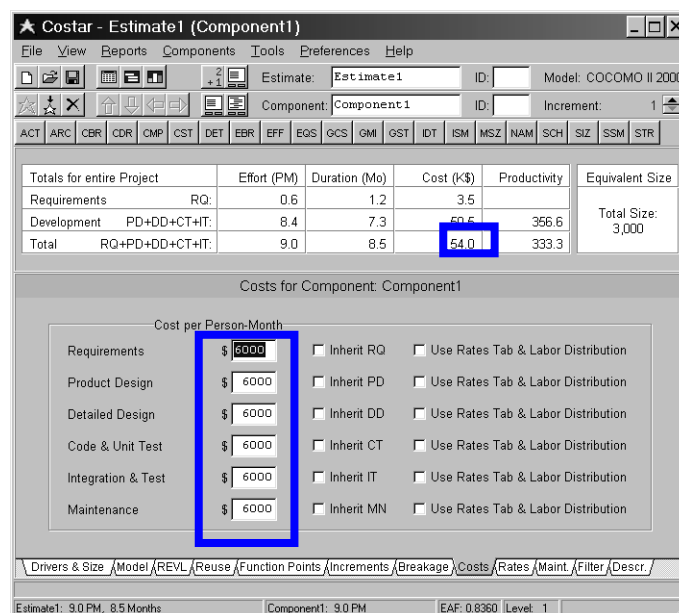


Figura 11: Actualitzar els costos

5.5.8. El Informe Detallat

Per accedir al Informe Detallat es realitza seleccionant el menú “Reports” i la opció “Detail Report” o fent clic al botó “DET” de la Barra de Informes.

El Informe Detallat mostra la estimació en recursos, cost i durada de la fase de requisits, cadascuna de les quatre fases de desenvolupament, i de la fase de manteniment.

| Phase | Effort (Person-Months) | Cost (K\$) | Duration (Months) | Staffing |
|------------------------------|---------------------------|------------|----------------------|----------|
| RQ -- Requirements | 0.6 | 3.5 | 1.2 | 0.5 |
| PD -- Product Design | 1.4 | 8.6 | 1.8 | 0.8 |
| DD -- Detailed Design | 2.3 | 13.5 | 1.7 | 1.3 |
| CT -- Code & Unit Test | 3.1 | 18.5 | 2.3 | 1.3 |
| IT -- Integration & Test | 1.6 | 9.8 | 1.5 | 1.1 |
| Development (PD+DD+CT+IT) | 8.4 | 50.5 | 7.3 | |
| Totals (RQ+PD+DD+CT+IT) | 9.0 | 54.0 | 8.5 | |
| MN -- Maintenance (per year) | 0.0 | 0.0 | | 0.0 |

Figura 12: El Informe Detallat

5.5.9. El Informe Programat

Per accedir al Informe Programat s’ha de seleccionar el menú “Reports” i la opció “Schedule Report”.

Aquest report mostra l’esforç en recursos i costos detallat mes a mes.

| Estimate1 - Schedule Report | | | | | | | | | | |
|--------------------------------|-----------------------------------|------------|-------------------|----------|-----|---------|-------------------|------------------|-----------------------|--|
| Costar 7.02 | | 01/20/2005 | | 16:18:27 | | Page: 1 | | | | |
| Estimate Name: Estimate1 | | | Estimate ID: 2000 | | | | | | | |
| Model Name: COCOMO II 2000 | | | Model ID: 2000 | | | | | | | |
| Process Model: COCOMO II Model | | | Phases: Waterfall | | | | | | | |
| Month | Effort this Month (Person-Months) | | | | | Total | Cumulative Effort | Cost | | |
| | RQ | PD | DD | CT | IT | | | (K\$) This Month | Cumulative Cost (K\$) | |
| 1 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.5 | 3.0 | 3.0 | |
| 2 | 0.1 | 0.7 | 0.0 | 0.0 | 0.0 | 0.8 | 1.2 | 4.5 | 7.5 | |
| 3 | 0.0 | 0.8 | 0.1 | 0.0 | 0.0 | 0.8 | 2.1 | 5.0 | 12.5 | |
| 4 | 0.0 | 0.0 | 1.3 | 0.0 | 0.0 | 1.3 | 3.4 | 7.9 | 20.4 | |
| 5 | 0.0 | 0.0 | 0.9 | 0.4 | 0.0 | 1.3 | 4.7 | 7.9 | 28.3 | |
| 6 | 0.0 | 0.0 | 0.0 | 1.3 | 0.0 | 1.3 | 6.0 | 8.0 | 36.3 | |
| 7 | 0.0 | 0.0 | 0.0 | 1.3 | 0.0 | 1.3 | 7.4 | 8.0 | 44.3 | |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 1.1 | 1.1 | 8.5 | 6.6 | 50.9 | |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 0.5 | 9.0 | 3.1 | 54.0 | |

Figura 13: El Informe Programat

5.5.10. Gràfic de Costos vs Temps

Per visualitzar el gràfic de Costos vs Temps (“Cost vs. Time”) s’accedeix des del menú “Reports”.

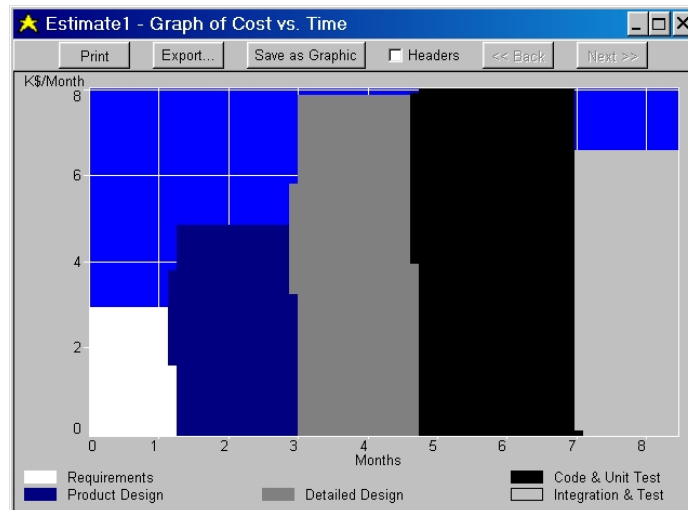


Figura 14: Gràfic de Costos vs Temps

6 CONCLUSIONS

Al realitzar aquesta memòria s’arriba a varies conclusions.

Lo que no es mesura, no es controla. Aquesta dita succeeix a moltes facetes de la vida i el treball, i el desenvolupament de programari no esta exempt d’aquesta dita. La mètrica és una combinació de indicadors que ens proporcionen una visió molt ampla del procés de software, d’aquesta forma, ens permet una millor adaptació als possibles

imprevistos i ens ajuda a practicar la reutilització del programari, molt fonamental avui dia.

És molt comú a les empreses de desenvolupament i als propis desenvolupadors de programari no utilitzar mètriques correctament o, fins i tot, no utilitzar-les. Aquest anarquisme en el procés fa que moltes vegades no hi hagi una estimació real sobre la qual partir el treball a realitzar, i consegüentment, els recursos, costos i temps són variables difícils de determinar.

La aplicació de mètriques del programari a les empreses és un punt a canviar perquè aquestes ho instaurin de manera estàndard en el treball diari. Avui dia, molts clients estan començant a demanar que els desenvolupaments que paguen es realitzin en base a mètriques estàndards. D'aquesta forma, els clients s'asseguren que el treball realitzat en un producte que ells mateixos han finançat, es realitza amb uns recursos, costos i temps òptims.

Un dels models que es basen en mètriques estàndard és COCOMO i la seva eina COSTAR. COCOMO es un model basat en les línies de codi, i no solament mesuren l'esforç com fan altres. Aquest model fa molt poderosa la estimació de costos.

COCOMO es una eina basada en les línies de codi la qual fa la estimació de costos molt poderosa i no es basa només en la mida, com si passa a altres eines.

Un factor a favor de COCOMO II, al igual que les versions anteriors, és que es de domini públic en relació als algorismes, les eines de software, estructures de dades, relacions e interfases. Per tant, la seva aplicació és molt més senzilla que altres models d'estimació privats.

A la actualitat hi han moltes avantatges i inconvenients en unes estimacions i altres, però el que si esta clar es que COCOMO II ha recorregut un camí força llarg i important, aconseguint una estabilitat i maduresa per aconseguir estimacions creïbles i amb força precisió.

7 GLOSSARI

Backup: És el procés de seguretat amb la finalitat que aquestes còpies es pugin utilitzar per restaurar el original després d'una possible pèrdua de dades.

Classe: Classe és una construcció que se utilitza com a model per crear objectes d'aquest tipus.

COCOMO: És un model matemàtic de base empírica utilitzat per a l'estimació de costos de programari.

Hardware: Hardware correspon a totes les parts tangibles de una computadora.

Indústria del software: La indústria del software és la indústria que involucra la investigació, desenvolupament, distribució i comercialització de programari.

Interfase: És la connexió entre dos ordinadors o màquines de qualsevol tipus donant lloc a una comunicació entre diferents nivells.

Mètrica: La mètrica és una metodologia de planificació, desenvolupament i manteniment de sistemes de informació.

On-line: En general es diu que un sistema esta on-line si esta connectat a una xarxa o sistema major.

Persona-Dia: És una unitat de estimació del esforç per realitzar una tasca. La seva unitat es una dia de treball d'un treballador mitjà. És força utilitzada també la terminologia "Persona-Hora".

Punts Objecte: Són el recompte de pantalles, informes i mòduls de llenguatge de tercera generació.

Requeriments: Requeriment és la necessitat documentada sobre el contingut, forma o funcionalitat d'un producte o servei.

Software o Programari: Es coneix com software o programari al equipament lògic o suport lògic d'una computadora digital.

8 BIBLIOGRAFIA

Llibres

Manual de Project Management. Método Consultores.

Ingenieria del Sotware, un enfoquepráctico. Pressman, Roger S.

Controlling Software Projects, DeMarco, T.

Internet

METRICA v 3.0: <http://www.csi.map.es/csi/metrica3/index.html>

Métricas, Estimación y Planificación de Proyectos de Software: http://www.willydev.net/descargas/willydev_planeasoftware.pdf

Métricas del Software, Por Alejandro De Coss: <http://www.gdl-mexcomp.com/Documents/metricas%20de%20software.pdf>

Asociación Española para la Gobernanza, la Gestión y la Medición de las Tecnologías de la Información: <http://www.aemes.org/>

Gestión de proyectos:

<http://www.padem.org.bo/focam2/documentos/MaterialCONSULTA/GestionProyectos.pdf>

Wikipedia: <http://www.wikipedia.com>