



**Universitat Autònoma
de Barcelona**

MAQU

**Metodologia
d'Assegurament de la
Qualitat**

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Gestió

realitzat per
Vanessa Gómez Monfort
i dirigit per
Xavier Verge Mestre

Escola Universitària d'Informàtica
Sabadell, Juny de 2009

El sotasignant, Xavier Verge Mestre,
professor de l'Escola Universitària d'Informàtica de la U.A.B,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per na Vanessa Gómez Monfort.

I per a que consti firma la present.
Sabadell, Juny de 2009

Signat: Xavier Verge Mestre

La sotasignant, Raquel López Núñez,
d' International Softmachine System s.l.,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva supervisió
per na Vanessa Gómez Monfort.

I per a que consti firma la present.
Sabadell, Juny de 2009

Signat: Raquel López Núñez

Resum introductori

La dificultat del desenvolupament de software i el seu impacte en el negoci, fa necessari aplicar una metodologia als projectes que es vulguin dur a terme amb èxit, per obtenir resultats predicibles que permetin alliberar un producte final d'alta qualitat que satisfaci les necessitats i expectatives dels clients.

El cicle de vida d'un software involucra una sèrie de processos: el procés de desenvolupament descriu un producte des del seu inici, construcció, alliberament i manteniment del producte i el procés de proves descriu la forma en que un producte ha de ser provat per obtenir els resultats esperats i desitjats garantint l'assegurament de la qualitat.

Les proves son uns dels aspectes més descuidats en el desenvolupament software encara que sigui el procés pel qual s'assegura la qualitat del producte, complir amb els requisits dels usuaris finals en la majoria dels casos no és suficient, s'ha de definir un procés de proves efectiu i estable per tots els projectes, és a dir, un mètode estàndard que inclogui les millors pràctiques de proves sota uns models que assegurin la millora continua.

Les proves de software és el punt més fort a analitzar en aquest projecte, s'executen amb la intenció de trobar els errors o defectes permetent validar, verificar i revelar la qualitat d'un producte obtenint el comportament pactat i desitjat de l'usuari final.

Les organitzacions tenen objectius per aconseguir beneficis, bons resultats, clients satisfets, ... i una de les causes importants que provoca no arribar als objectius son: les deficiències en les proves del software.

En el present projecte es realitza un estudi, anàlisi i disseny de 'MAQU' la metodologia aplicada a l'etapa de proves d'un producte software basada en els models de qualitat. Per arribar a establir una metodologia en el procés de proves es necessari estudiar els processos que intervenen en el cicle de vida del software, els models estàndards, models de qualitat i proves disponibles al mercat més apropiats i adequats a les necessitats.

A nivell mundial s'estan generant un conjunt de models per mesurar la qualitat dels productes desenvolupats, com el model CMMI, els models de qualitat i proves TMAP, TMMI, TPI, etc., aquests models representen les característiques perquè els productes o serveis de qualitat puguin competir en un mercat obert i global. Moltes vegades els defectes de les aplicacions no son provocades per falta de personal a les empreses o per falta de temps en els projectes, sinó per no tenir una metodologia que garanteixi la qualitat dels productes amb una prèvia planificació, definició de responsabilitats, activitats i rols definits dins dels diferents processos d'un projecte.

La motivació del projecte sorgeix de la necessitat de definir una metodologia de treball a utilitzar en el procés de proves d'un producte software considerant la participació dels recursos implicats en el desenvolupament del producte, a l'empresa International Softmachine System s.l. dedicada a dissenyar, construir i instal·lar aplicacions software de 'Control Horari i d'Accés' per empreses de tots els sectors.

Índex

1	Motivacions del projecte	1
1.1	Objectius i estructura de la memòria	2
1.2	Planificació del projecte.....	3
1.3	Projecte a International Softmachine System s.l.....	4
1.3.1	Departament I+D	5
2	Introducció a la Metodologia	6
2.1	Construir una metodologia	7
2.2	Establir una metodologia	8
3	Qualitat.....	10
3.1	Evolució històrica de la qualitat	10
3.2	Assegurament de la qualitat	17
3.3	Objectius d'un projecte de qualitat	20
4	Processos i models de qualitat software	22
4.1	Procés i cicle de vida del Software	22
4.1.1	Etapes i elements del cicle de vida del software.....	24
4.1.2	Processos del cicle de vida del software	26
4.1.3	Models del cicle de vida del software	30
4.1.4	Estàndards del Cicle de vida dels processos	32
4.2	Models de procés, qualitat i millora.....	34
4.2.1	Models estàndards	34
4.2.2	CMMI (<i>Capability Maturity Model Integration</i>).....	39
4.2.3	PSP Personal Software Process	40
4.3	Models de proves de software	40
4.3.1	TMap® (<i>Test Management Approach</i>)	41
4.3.2	TMM (<i>Test Maturity Model</i>)	43
4.3.3	TPI® (<i>Test Process Improvement</i>).....	47
5	Proves de software	53
5.1	Verificació i Validació	55
5.2	Proves	56
5.2.1	Mètodes dinàmics	58
5.2.2	Requeriments del software	58
5.2.3	Objectiu del procés de proves	59
5.2.4	Tècniques i eines de les proves	60
5.2.5	Mètriques de les proves.....	64
6	Metodologia d'Assegurament de la Qualitat.....	65
6.1	Objectius en el procés de proves	67
6.2	Relació entre el procés de proves i desenvolupament.....	68
6.3	Rols de QA	70
6.4	Desenvolupament	73
6.5	Estudi preliminar	73
6.6	Planificació de proves	76
6.7	Pla de proves	78
6.7.1	Tests i proves a dissenyar.....	81
6.7.2	Especificacions del disseny de proves	81
6.7.3	Test d'exploració	82
6.7.4	Test de Proves específiques (funcional)	83
6.7.5	Test de Proves de combinacions	84
6.8	Cicle de proves	85
6.8.1	Configuració de l'entorn.....	86
6.8.2	Execució de les proves	89
6.8.3	Verificació i validació del software.....	89
6.8.4	Report dels resultats	91

6.8.5	Avaluació de les proves	92
6.9	Avaluació i millores.....	94
7	Cas pràctic aplicant MAQU	96
7.1	Aplicació Tempo Millenium.....	97
7.1.1	Mòdul Personal Extern.....	98
7.2	Estudi preliminar	99
7.3	Planificació de proves	101
7.4	Pla de proves.....	103
7.5	Cicle de proves	106
7.6	Avaluació i millores.....	112
8	Conclusions del projecte	113
8.1	Seguiment del projecte	115
8.2	Línies futures.....	116
9	Bibliografia	117

1 Motivacions del projecte

Al iniciar un nou projecte un aspecte que resulta determinant per trobar una forma de treball sana és una sòlida planificació del mateix ja que no hi ha una forma màgica que pugui aplicar-se a qualsevol projecte. És interessant veure com diferents persones es plantegen un projecte de diverses formes, en molts casos certes etapes son iguals i en altres les etapes tenen subtils diferències però molt importants al resultat final.

International Softmachine Systems s.l. és una empresa dedicada a desenvolupar i implantar solucions de Control Horari i d'Accés per les empreses. Per exemple, un empleat que va a treballar accedeix a la empresa per un terminal, en el cas de no arribar a temps a la seva jornada laboral se li conta un retràs depenent de la hora del fitxatge, al arribar al seu lloc de treball pot tenir un altre tipus de control per accedir-hi, quan va a dinar pot tenir un menjador on hagi de justificar el temps invertit, etc... Recursos humans necessita llistats d'assistència, càlcul d'hores, vacances dels empleats, ... Softmachine disposa d'aplicacions que donen solució a les necessitats dels clients.

Des del punt de vista del client i dels usuaris, la qualitat d'un producte software és principalment percebuda pels errors que es troben en el producte i per la gravetat que aquestes falles provoquen sobre el negoci del client. Per ser una empresa competitiva, les empreses desenvolupadores de software necessiten assegurar-se de la qualitat dels seus productes previ a la instal·lació al usuari final.

Les proves d'un producte abans de ser entregat al client, implica provar el comportament de totes les combinacions vàlides i no vàlides de les entrades en l'aplicació, sota tot estat possible del sistema. Provar totes les combinacions és quasi inviable però les empreses han de fer entrega de les aplicacions amb el menor número de defectes possibles. Per això, es defineix quan entregar el producte al usuari final en funció del desenvolupament i de la realització de les proves.

L'objectiu de les empreses és alliberar els productes amb el menor impacte negatiu pel negoci del client i per aconseguir això existeixen tècniques i estratègies per definir, dissenyar i executar les proves des de l'equip de encarregat del procés de proves també anomenat testing.

International Softmachine Systems s.l. disposa de l'equip QA (Quality Assurance) dintre del departament d'Investigació i Desenvolupament i realitza tot el procés de definició, disseny i aplicació de proves del software. Per realitzar el procés de proves del software de forma independent al departament de desenvolupament, el procés a definir per QA no pot dependre íntegrament de la metodologia de treball utilitzada pel desenvolupament del producte ja que les proves han de ser funcionals i no enfocades al codi font dels desenvolupadors. Així l'àrea QA no necessita ni disposa del codi font de les aplicacions per realitzar les proves per obtenir un software de qualitat.

La motivació d'aquest projecte neix amb l'objectiu d'establir una metodologia de treball a utilitzar en el procés de proves del software dels projectes a International

Softmachine Systems s.l. i així entendre i implantar la metodologia seguida a la fase de proves de l'aplicació basada en un model de qualitat software. El seguiment de proves del software és important durant qualsevol fase de desenvolupament del producte, alhora les activitats de verificació i validació dutes a terme durant tot el cicle de vida del software ofereixen una major confiança i qualitat en el producte software desenvolupat i faciliten el manteniment del mateix.

1.1 Objectius i estructura de la memòria

El projecte s'inicia des de la primera idea de millorar el mètode de treball en el procés de proves de l'àrea encarregada de les proves software. L'objectiu d'invertir recursos en aquest projecte és obtenir la metodologia a seguir al llarg dels presents i futurs projectes fet indispensable per assegurar bons resultats finals.

Les proves de software permeten verificar la qualitat d'un producte i s'integren dins de les diferents fases del 'Cicle de vida del software' dins de l'Enginyeria de Software.

Així mitjançant les tècniques adequades es tracta d'analitzar i executar les aplicacions construïdes, i descobrir els errors o funcionalitats no adequades que contenen abans de alliberar al client. Les aplicacions han de complir amb les especificacions i requeriments sol·licitats i pactats amb el client, i la fase de proves ha de ser present al llarg del desenvolupament de l'aplicació per evitar arribar a l'alliberament del producte sense complir amb les especificacions inicials del sistema. Les proves de software són imprescindibles per comprovar aquest grau de compliment i és el procés que identifica les possibles falles d'implementació, qualitat i estabilitat d'un programa.

Amb aquest estudi s'espera contribuir a establir una metodologia de treball a l'àrea QA intentant establir un estàndard per propers projectes i així partint d'adequades planificacions dels projectes donant una empenta a la qualitat del software. Això es podrà fer possible utilitzant les diferents eines i tècniques per respondre'ns a dubtes alhora de fer el seguiment del projecte per obtenir una millor qualitat. Cadascuna de les tasques d'un projecte en cadascuna de les seves fases pot ser resolta de múltiples maneres.. però un bon mètode de treball sempre ens portarà a resoldre qüestions respecte a les tasques del projecte. És necessari saber quan podem donar per finalitzada una tasca, qui realitza les tasques, quines tasques són predecessores i quines antecessores respecte altres tasques, quina documentació hem d'utilitzar per cada tasca, ...

En aquest sentit, aquest projecte espera establir clarament una metodologia de treball utilitzant les tecnologies de la informació, així com models de gestió de qualitat com eines imprescindibles per millorar en la qualitat del software.

El projecte s'estructura de la següent manera:

En el primer apartat del projecte s'exposa una breu introducció i els objectius del projecte. Alhora es descriu l'empresa on es realitza el projecte i abastament del projecte.

El segon apartat reflexa la importància d'establir una metodologia com a principal objectiu del projecte.

El tercer apartat presenta l'estudi de Qualitat. És interessant conèixer l'entorn de la qualitat per poder arribar a entendre la metodologia seguida per la gestió d'un software de qualitat.

En el quart apartat es realitza l'anàlisi dels factors que intervenen en el procés de desenvolupament software contemplant l'estudi de models i teories que proporcionen la base per la metodologia estudiada en aquest projecte.

En el cinquè apartat es descriuen i analitzen les proves de software, validació i etapes de test en un projecte.

En el sisè apartat es presenta 'MAQU' la proposta de metodologia basada en els models de qualitat i en models de millora de proves. Es desenvolupa el projecte, així com dissenyar, aplicar i provar la metodologia presentada.

Al setè apartat es documenta un cas pràctic aplicant MAQU amb la descripció dels resultats i conclusions obtingudes. L'estudi es basa en el procés de proves d'un nou mòdul en una aplicació d' International Softmachine System s.l. donant força al producte amb noves funcionalitats que el departament d'Investigació i Desenvolupament junt amb els responsables de prendre decisions de negoci a l'empresa han trobat rentable llençar al mercat.

La resta d'apartats exposen les conclusions i bibliografia utilitzades al llarg del projecte que han fet possible l'enteniment i estructuració de l'anàlisi, disseny i pràctica de la metodologia.

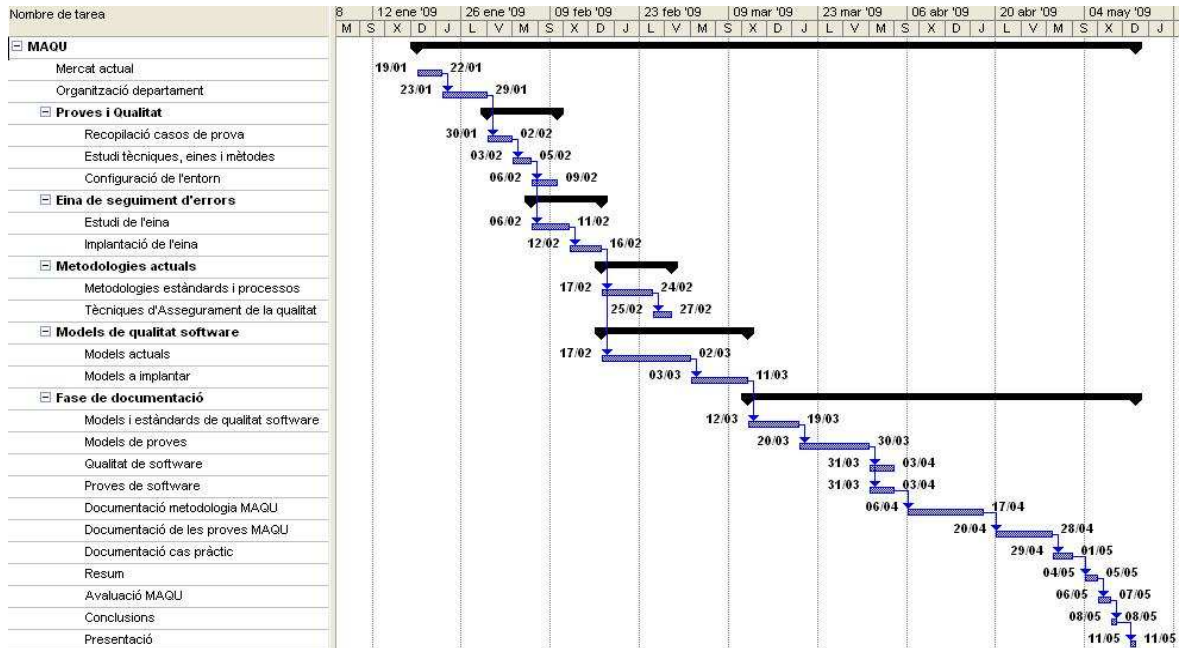
1.2 Planificació del projecte

Inicialment es planteja una planificació del projecte amb MS Project dividida en les diferents seccions d'estudi del projecte resultant un total de 250h.

Els primers punts contemplen l'anàlisi, estudi, documentació i preparació continua d'adaptació a la nova metodologia de treball i darrerament el punt 'Fase de Documentació' contempla els resultats obtinguts a la memòria del projecte i altre documentació necessària en el desenvolupament del projecte.

D'una altre banda, existeixen forces que poden afectar a la planificació com per exemple accions inesperats, resistència al canvi, insuficient informació, falta d'habilitat de mètodes entre d'altres.

A continuació es mostra el **Diagrama de Gantt** que planifica el temps a invertir en el projecte d'acord amb les hores establertes i pactades destinades al projecte.



1.3 Projecte a International Softmachine System s.l.

International Softmachine System s.l. va ser fundada al 1989, s'especialitza des del inici en el desenvolupament i implantació de sistemes de gestió horària, control de presència, control d'accessos i captura de dades en planta.

Durant aquests 20 anys de trajectòria s'ha pogut ajudar a persones de diferents països que han pogut beneficiar-se de les aplicacions i solucions per a resoldre les seves necessitats especials.

Conta amb més de 2000 clients a Espanya i a Amèrica Llatina la qual cosa empeny a posar tota la obstinació en una constant innovació tecnològica en els productes de desenvolupament propi.

Per a poder oferir una solució global, s'ha establert acords estratègics amb els fabricants de captura de dades amb el màxim reconeixement internacional, el que garanteix als clients una solució global, integral a través d'un únic interlocutor per a tota la seva problemàtica de gestió de presència, accessos i horaris.

International Softmachine System s.l. disposa d'un equip d'especialistes que posa tota l'energia en resoldre necessitats concretes i fer fàcil i àgil el treball diari. Des de l'àrea comercial s'assessora per ajudar a triar la millor solució i les persones de l'equip tècnic i de manteniment, d'assistència telefònica i de formació donen suport per a garantir el màxim rendiment de les aplicacions.

El principal objectiu de Softmachine és créixer al costat dels clients, fent que les seves inversions s'amortitzin en el menor temps possible.

1.3.1 Departament I+D

El departament d' I+D de Softmachine, és el departament encarregat de fomentar la investigació i desenvolupament dels productes per millorar les aplicacions ja existents, crear i desenvolupar de noves en funció de les necessitats del mercat i incorporar els últims avenços que es produeixen en el camp del software. Les aplicacions cada dia han de ser més professionals, complertes, àgils i integrades.

El departament consta de 8 figures dintre del desenvolupament d'un projecte a I+D:

- Responsable d'àrea: Màxim responsable del departament que realitza les funcions de gestió de projectes, gestió d'incidències i gestió del departament.
- Cap de projecte: És el responsable d'alliberar les entregues dels projectes i temps finals de desenvolupament amb qualitat i la documentació necessària. Informa, consulta i aporta plans d'actuació sobre els projectes i realitza funcions de gestió, incidències.
- Sistemes: És el responsable de solucionar les incidències de sistemes amb els clients, participa al disseny de la base de dades i realitzar l'instal·lador de les aplicacions.
- Test o Quality Assurance: Encarregat de les Proves del software per verificar i validar els productes software. Les funcions realitzades s'incorporen des del primer moment en el desenvolupament del software fins a l'alliberament a l'usuari final. Planteja millores de qualitat, realitza el disseny, revisions, execució de proves, plans de proves i realitza els manuals interns i externs dels productes software.
- Analista programador: És el responsable de realitzar l'anàlisi de desenvolupament de codi amb la qualitat òptima en el mínim temps possible.

Hi ha dos equips d'analistes programadors per cadascuna de les tecnologies utilitzades a l'organització (Delphi i Asp.net).

- Programadors: A partir d'unes especificacions donades es el responsable de desenvolupar el codi amb la qualitat òptima en el mínim temps possible.

En el present projecte es diferencia i desenvolupa entorn a l'àrea QA i l'àrea de desenvolupament del producte. En tot moment al referir àrea de desenvolupament s'està incloent els analistes programadors, programadors i sistemes i en el cas de referir a l'àrea QA es fa referència a l'equip encarregat de les proves de software.

2 Introducció a la Metodologia

La metodologia és un concepte que pot dur a confusió, no hi ha una metodologia clarament definida per cada projecte, és trobar el mètode més adequat per guiar un projecte d'inici a fi i que arribi a respondre als problemes i necessitats que sorgeixin al llarg d'un projecte.

En el món de la informàtica la constant innovació tecnològica fa que cada cop sigui més necessària l'aplicació de nous mètodes i no continuar amb vells costums sense adaptar les empreses als nous temps. És necessari seguir unes pautes predefinides en el desenvolupament del software de qualitat sota un comportament metòdic: seguir una metodologia.

No cal fixar-nos en estudis o estadístiques que demostrin que no seguir una metodologia garanteix amb seguretat una absència de qualitat, només cal veure resultats d'algunes empreses que no segueixin cap model de gestió en el desenvolupament dels seus projectes.

Un estudi realitzat per The Standish Group 'Chaos Summary 2008' exposa xifres respecte l'èxit i fracàs dels projectes en els darrers anys i els factors de l'èxit dintre de les lleis del CAOS.¹

El punt més baix en els cinc anys analitzats, del 1998 al 2006, va ser 1998 amb tan sols el 26% dels projectes es van dur a terme amb èxit i el 28% van fracassar.

A l'any 2006 el 35% dels projectes van aconseguir l'èxit, el 46% van ser impugnats (amb retràs, més pressupost, amb menys funcions o menys característiques) i el 19% van fracassar.

L'estudi CAOS determina que els projectes per aconseguir l'èxit han de ser gestionats amb un gerent de projectes. El 93% de l'èxit dels projectes son gestionats per un gestor de projectes professional, mentre que el 19% dels projectes no tenen cap gerent al projecte.

L'elecció de bones eines i apropiades estratègies duen una empresa a l'èxit dels seus projectes si l'habilitat amb la qual s'utilitzen és l'apropiada, si no s'utilitzen correctament els projectes s'enfoquen cap al fracàs. El temps és l'enemic de tots els projectes per això des del principi els projectes han de partir d'una bona planificació i coordinació per arribar a l'èxit en el temps desitjat.

A finals del 2001 es va realitzar un estudi per esbrinar els motius pels quals els projectes originaven fracassos i es va arribar a les següents interpretacions:²

- El 21% per canvis en els objectius definits a nivell estratègic.
- El 31% per la no utilització o dolenta utilització de les metodologies de treball.

¹ [CHSUM] The Standish Group, Chaos Summary 2008

² [GERPRO] Daniel Piorun, Gerencia de proyectos, 20.01.2003

- El 48% per problemes humans, de conducció, comunicació i conflictes entre la gent.

A moltes organitzacions petites no existeix el concepte de metodologia i en moltes empreses grans sí existeix però moltes vegades per falta de temps o dolentes planificacions acaba sent utilitzada com una màscara formal pel compliment de normes i no com eix del projecte utilitzant el contingut de la metodologia.

Un dels punts en els quals és més dèbil la utilització de metodologies, és en el disseny de la estructura d'un projecte i en l'estimació d'esforços i temps.

Es fonamental que el treball es mesuri utilitzant mètriques i disposar de mètodes provats a cada organització que mesurin la mida del projecte, així com tenir els estàndards propis sobre els esforços necessaris.

2.1 Construir una metodologia

L'estudi que es presenta en el present projecte pretén establir una metodologia basada en un model de qualitat. A mida que vagi avançant el projecte s'anirà posant en pràctica la metodologia ideada i així es podrà veure si en els resultats obtinguts son positius i es pot fer com estàndard pels següents projectes. L'objectiu és que pugui realitzar-se però amb aquest projecte es decidirà si l'àrea QA a partir d'ara s'acollirà a aquesta metodologia per millorar la planificació, organització i qualitat del software.

Una metodologia ha de considerar els requisits desitjables i criteris d'avaluació d'aquests requisits. Al tractar-se d'un projecte informàtic hi ha un equip de desenvolupament i aquest ha de seguir un criteri comú per dur a terme el projecte.

El desenvolupament del projecte passa per una etapa de presa de requeriments del client, reunions, anàlisi, planificació del projecte, redactar informes ... totes aquestes tasques semblen evidents però al iniciar un projecte sorgeixen mil detalls:



Qui va a les reunions?

Quin format han de tenir els informes?

Quines son les funcionalitats del projecte?

Qui redacta els informes?

Quins desenvolupadors s'encarreguen del projecte?

Qui realitza el test?

Quan es comença a desenvolupar? I quan s'inicia el test?

Quan es considerarà acceptable el projecte per entregar al client?

Quines son les dates de finalització de les fases del projecte?

...

Una persona és capaç de respondre'ns aquestes preguntes però llavors s'estan fent les coses a la manera d'un mateix sense participar en equip i sense seguir unes pautes predefinides, sense metodologia... seria possible en el cas d'organitzar un projecte molt petit però en conclusió massa càrrega per una persona sola, la desorganització i entrebancs son casi inevitables, per un mateix a lo millor no es veu tant difícil però la resta de l'equip que participa al projecte no pot saber el que la persona organitzadora pensa o vol fer, difícilment es podrà seguir unes pautes i constantment es necessita la participació de la persona que organitza i planifica. Aquesta manera de fer les coses només porta problemes perquè llavors tothom faria les coses a la seva manera.

Lamentablement, això passa als projectes reals tots els dies... generant projectes fracassats, recursos perduts i professionals frustrats. I segurament no ha sigut pels recursos empleats o per la feina mal feta dels professionals sinó per la dolenta planificació i organització del projecte.

Què s'ha de saber per construir o escollir una metodologia?

Hi ha metodologies que es coneixen a l'àrea informàtica com metodologies tradicionals. La majoria d'aquestes van ser proposades abans dels anys 90 i per molt que es modernitzin i vulguin adaptar-se per satisfer les necessitats de les empreses actuals, la seva funcionalitat sol estar limitada a projectes no gaire innovadors. Son metodologies enfocades a negocis dels anys 90 i anteriors i no solen tenir en compte conceptes com la qualitat, la satisfacció, la competitivitat, els beneficis, ... la majoria de metodologies tradicionals ja no funcionen.

Les metodologies actuals responen a una altre mentalitat totalment diferent: han de ser àgils, amb pautes comuns, adaptables a diferents projectes i fàcil d'utilitzar i alhora completa per ser utilitzable per tot l'equip de treball.

2.2 Establir una metodologia

Què comporta establir una metodologia?

Existeixen molts estàndards i models en els quals es pot basar una empresa com ITIL, ISO 9000, CMMI, etc. Els estàndards constitueixen una base poder imprescindible per començar l'anàlisi i disseny d'una metodologia però hi ha altres aspectes importants a analitzar alhora d'establir una metodologia en qualsevol organització. Al implantar una metodologia s'ha de tenir en compte que hi haurà un canvi en la gestió dels projectes i necessita l'acceptació de tot recurs que estigui implicat en els projectes, els aspectes a destacar son els següents:

- *Claus per gestionar el canvi adequadament*

Aplicar una metodologia i haver de modificar la forma habitual de treball no és feina fàcil. L'experiència durant els anys porten a adoptar una forma de treballar que poder no és l'adequada però és la que s'acostuma a utilitzar i la que tenim a l'abast per realitzar més ràpid i aparentment més ordenada. Pretendre establir una metodologia suposa una gestió en el canvi fonamental per l'acceptació dels nous mètodes de treball i els següents aspectes resulten necessaris per a realitzar el canvi:

- Es imprescindible el suport de direcció per gestionar una metodologia.
 - Preparar-se per la resistència al canvi. Les causes de la resistència s'han d'analitzar i gestionar com part del projecte.
 - Ha de ser un canvi progressiu. Establir una metodologia i realitzar els canvis massa ràpid pot dur a terme la confusió i inestabilitat per això és necessari realitzar la implantació poc a poc i respectant el temps per cadascú dels integrants afectats pel mètode de treball.
- *Utilitzar l'històric dels projectes*

Els projectes ajuden a establir la metodologia. Alhora de definir el mètode de treball i definir els processos o pràctiques a implantar amb la metodologia és important agafar les mides dels projectes reals de l'empresa, així es podrà identificar:

- Les activitats en les que l'equip presenta més dificultats.
 - Les carències de la metodologia ja implantada.
 - Les particularitats no contemplades inicialment per la metodologia i que s'han de tenir en compte per la completa implementació.
- *Enteniment de la metodologia per part de l'equip*

La metodologia s'ha d'entendre tant a nivell individual com en equip. Si no s'aprecia el valor de la metodologia per part de l'equip difícilment serà aplicat i respectat per la resta de la organització. És igual d'important transmetre la metodologia als integrants a l'àrea QA de desenvolupament que a la pròpia àrea encarregada de les proves on s'establirà la metodologia ja que la comunicació entre ambdues s'estableix a través dels mètodes establerts.

- *Pensar en tothom*

Per molt que la metodologia a seguir sembla que afecti al departament encarregat de les proves més directament que a la resta d'integrants de l'organització, la metodologia ha de ser pensada des d'un punt de vista genèric ja que la seva implantació no és possible sense l'acceptació de la resta ja que està definida per adaptar-se a tots i per poder ser seguida per tots.

- *Llenguatge i contingut dels procediments*

Els procediments s'han de descriure de manera clara i fàcil d'entendre ja que ha de ser entesa per varis perfils i per això s'agraeix simplicitat per evitar ambigüitats i diferents interpretacions.

Cal evitar continguts massa extensos i repeticions innecessàries ja que pot fer que el lector es perdi en la lectura o perdi interès dificultant l'acceptació del projecte.

3 Qualitat

En aquest apartat es realitza un breu resum de l'evolució històrica de la qualitat, seguidament en el següent punt es presenta el concepte 'Assegurament de la Qualitat' en el que s'argumenten diversos conceptes com gestió de qualitat, sistema de qualitat, qualitat d'un procés, planificació de la qualitat, etc. , en el següent punt s'expliquen els factors de la qualitat software i en el darrer punt es menciona els objectius d'un projecte de qualitat.

3.1 Evolució històrica de la qualitat

La interpretació del concepte de qualitat varia al llarg dels anys i per entendre la qualitat en els temps actuals és imprescindible fer una ullada a l'evolució històrica de la mateixa destacant alguns personatges que es preocupaven de la qualitat dels productes i d'alguns conceptes relacionats.

Finals segle XIII i segle XIX

La Revolució industrial va representar una transformació del treball manual pel treball mecanitzat. Abans d'aquesta etapa el treball era artesanal i la responsabilitat sobre la producció completa d'un producte recau sobre del treballador.

La segona Guerra Mundial va representar un procés de constant canvis, els països directament involucrats al conflicte son els que avui volen dominar els mercats mundials i son els que han generat el coneixement més important del que disposa l'administració de les organitzacions, Estats Units com el triomfador de la guerra i Japó com el perdedor on es va establir la revolució de la qualitat que ara s'escampa per la resta del planeta.

Al segle XX

Es va desenvolupar una era tecnològica que va permetre a les masses de disposar de productes que fins aleshores només hi disposava les classes privilegiades.

En aquesta època *Henry Ford* introduí 'Ford Motor Company' amb l'objectiu de separar els productes acceptables dels no acceptables: la qualitat era responsabilitat del departament de fabricació.

Més endavant es va comprovar que la prioritat del departament de producció es preocupava de complir amb les dates d'entrega fixades i no podia assumir la responsabilitat de revisar perfectament la qualitat.

Entre 1920 i 1940

Hi ha un canvi tecnològic. *Walter A. Shewhart* introdueix el concepte de 'Control estadístic de qualitat' i junt amb *George D. Edwards* esdevenen líders d'un departament d'Enginyeria d'Inspecció que es preocupava dels problemes detectats als productes i per la coordinació entre departaments.

Es prenen accions correctives sobre els processos tecnològics. Els inspectors de qualitat continuaven sent un factor clau en el resultat de l'empresa i no només inspeccionen el producte final sinó que estan distribuïts al llarg de tot el procés productiu.

Al 1935 Egon S. Pearson, entorn a la segona guerra mundial, va reforçar la tecnologia de qualitat. La necessitat de millorar la qualitat del producte va donar una empenta a l'estudi del control de qualitat.

En aquest moment es va expandir els conceptes de control de qualitat i els tècnics de qualitat es van començar a involucrar en les primeres etapes del disseny del producte.

Al 1950 Edwards W. Deming després de que la guerra destruís l'economia Japonesa, va establir les bases pel retorn amb èxit de l'escenari internacional dels negocis i va implantar les idees en el millorament de la qualitat i la reducció de costos.

Va convèncer de que la qualitat japonesa podria convertir-se en la millor del món amb els mètodes que proposava i moltes empreses comencen a aplicar conceptes de prevenció i assegurament de la qualitat.

Qualitat total

Entre 1960 i 1970 van sorgir els principis de qualitat total i es poden destacar alguns personatges que descriuen teories i enfocaments de la millora de la qualitat.

A Estats Units els principals autors i precursors de la qualitat total son: Philip B. Crosby, Edwards W. Deming, Armand V. Feigenbaum i Joseph M. Juran.

Al Japó trobem com a principals autors: Kaoru Ishikawa, Shegeru Mizuno, Shigeo Shingo i Genichi Taguchi.

La teoria de la qualitat total té les arrels en àrees com: el desenvolupament de les organitzacions, les teories de la motivació i el comportament humà, la teoria del lideratge i molts altres. La combinació entre aquests conceptes i les aportacions dels diferents fundadors es crea la teoria de la qualitat total.

Els diferents fundadors de teories i principis relacionats amb la qualitat total coincideixen en un conjunt de idees:

- ✓ Aquesta filosofia ha de ser impulsada des dels alts càrrecs.
- ✓ Es un problema de tots.
- ✓ Ha de ser orientada al consumidor.
- ✓ Es un procés de millorament continu.
- ✓ Requereix d'una educació permanent, tant a dirigents com treballadors.
- ✓ Necessita una medició permanent que indiqui quin és el cost del incompliment.

A partir del 80

Es dona una empenta a la millora de la qualitat com a supervivència d'una organització. Es caracteritza per la direcció estratègica de la qualitat, ja que la qualitat aconseguida en un producte és una estratègia aplicada a tota l'empresa.

A partir del 1990 l'enfocament no només es focalitza al mercat també es reflexa en el coneixement de les necessitats i expectatives dels clients, per construir una organització que satisfaci als clients. La responsabilitat de la qualitat s'enfoca des dels líders de l'organització.

L'organització de Normes ISO des d'un inici el seu objectiu era millorar la qualitat i d'aquest organisme sorgeix ISO 9000 integrades per un conjunt de models i document sobre la gestió de qualitat.

Origen de les Normes ISO 9000

Al 1987 es va publicar la primera edició de la família ISO 9000. Conté les normes sobre l'Assegurament de la qualitat i a partir d'aquest moment les normes ISO adopten un estàndard mundial referent a la gestió de qualitat i les sorgeixen noves versions com ISO 9001 i 9004 que tenen major congruència en estructura i contingut i es basen en 8 principis de l'administració de la qualitat que detallen les millors pràctiques:

- Organització enfocada al client: compren les necessitats actuals i futures dels clients satisfent els seus requisits i expectatives.
- Lideratge: Els líders estableixen els propòsits i direcció de l'organització.
- Participació de personal: l'implicació del personal es necessària pel benefici de l'organització.
- Enfocament basat en processos: els resultats desitjats s'aconsegueixen amb major eficiència quan els recursos i activitats relacionades es gestionen com un procés.

- Enfocament del sistema per la gestió: la millora i eficiència d'una organització ve donada per la gestió, identificació i enteniment dels processos.
- Millora continua: Objectiu permanent de l'organització.
- Enfocament objectiu cap a la presa de decisions basades en fets: les decisions efectives es basen en el anàlisi de dades i informació.
- Relació mútuament beneficiosa amb el proveïdor.

Els sistemes de qualitat basats en normes i estàndards son acceptats mundialment i son escollits per empreses disposades a aconseguir un satisfactori grau de qualitat i millora continua.

Els estàndards i normes

Ofereixen mètodes i procediments compatibles amb les polítiques internes de cada empresa.

- La qualitat dels productes i serveis han de complir amb les necessitats, gustos i expectatives dels clients.
- Costos competitius.
- Flexibilitat. Major gama de productes i menors temps d'entrega obtenint la satisfacció dels clients i millor posició al mercat.

Actualment la qualitat s'associa al compliment de les necessitats, ús i satisfacció del client i això provoca exigències sobre el producte i la necessitat d'obtenir productes de qualitat.

El Producte

Es un concepte relacionat amb la qualitat que ha anat evolucionant al llarg dels anys. Es defineix com el resultat d'activitats o processos i pot ser tangible (materials) i / o intangible (coneixements o conceptes).

La ISO 9000 estableix una classificació dels productes i també aplicable una combinació d'ells:

- Materials (hardware): Peces, components, muntatges, etc.
- Suports lògics (software): programes, procediments, informació, dades, etc.
- Materials processats: matèries primes, líquids, sòlids, etc.
- Serveis: bancs, transports, etc.

De totes les formes el producte és el resultat d'un procés combinant recursos i activitats relacionades, com tècniques, mètodes, equips, instal·lacions, i que transformen elements d'entrada en elements de sortida.

La qualitat d'un producte ha sigut definida de moltes maneres però sempre resulta imprescindible per les empreses.

Actualment

El mercat és exigent i la societat disposa de productes similars els quals es poden diferenciar en la qualitat i competir en les expectatives que busquen els clients.

La qualitat és aplicable a qualsevol producte i els productes software han de contemplar la qualitat en totes les etapes del desenvolupament del software.

El producte ha de contemplar factors de qualitat: fàcil manteniment, de confiança, facilitat d'ús, etc. ja que son productes realitzats per persones destinat a persones i la qualitat no només es centra en el producte sinó que ha de tenir en compte quin és el destí pel qual s'està desenvolupant. Si una aplicació està molt ben feta, codificada, planificada i estructurada, però l'usuari final no entén o no disposa de la facilitat d'ús, que l'aplicació disposi d'un codi estable i potent no serveix de gran cosa.

La qualitat és mesurable des de diferents aspectes:

- *Qualitat interna*

Mesurable a partir de característiques intrínseques, com el codi font.

- *Qualitat externa*

Mesurable en el comportament del producte, com les proves.

- *Qualitat d'ús*

Mesurable per l' utilització del producte.

La qualitat d'un producte software es diferencia de la qualitat d'altres productes fabricats ja que conté característiques especials:

- El software és un producte mental.
- No es fabrica, es desenvolupa.
- Si s'apliquen els factors adequats, no es deteriora amb el temps.
- El manteniment pot resultar més costos que el hardware.
- El software que conté errors no es rebutjat, sinó que és de baixa qualitat respecte altres productes.

Quan es parla de qualitat software sorgeixen dubtes alhora de conceptualitzar, per això existeixen estàndards i normes que descriuen la representació d'aquests conceptes:

- Els models de qualitat resolen dubtes davant del concepte de qualitat i l'abastament de cada concepte relacionat.
- La gestió de qualitat i assegurament de la qualitat resolen la forma en que es tracta la qualitat, les polítiques, les activitats planificades per obtenir la qualitat esperada.

Algunes definicions:

Walter A. Shewhart (1891 - 1967)

Pare del control estadístic de qualitat. Entenia la 'Qualitat' com un problema de variació, el qual pot ser controlat i es pot prevenir mitjançant l'eliminació a temps de causes que el provoquen (gràfics de control).

Shewhart introdueix el Control estadístic de qualitat i va ser el primer en reconèixer que en tota producció industrial es dona variació en el procés.

Philip B. Crosby (1926 - 2001)

Primer en el si de ITT i després com a consultor independent ha estat un dels grans referents en temes de la qualitat. És molt conegut per la seva frase '*Quality is Free*', fent un joc de paraules en quan a 'Free' que es pot traduir com a lliure i com a gratis. La idea és que fer les coses bé a la primera no costa, en canvi els errors tenen importants costos afegits.

Confirma que la qualitat està basada en quatre principis:

- Qualitat és defineix com compliment de requisits.
- El sistema de qualitat és prevenció.
- L'estàndard de realització és zero defectes.
- La mesura de la qualitat és el preu del incompliment.

Joseph M. Juran (1904 - 2008)

Un dels més renombrats consultors en temes de qualitat. Aplicava el principi de Pareto 80/20, en tant els 80% dels errors obeïen només al 20% de les causes, per tant, treballant sobre el 20% tindríem el resultat del 80, tot i això cal treballar també l'altre 80, ell ho anomenava 'els pocs vitals i els molts trivials'.

El Control de la qualitat va ser el seu punt fort i va divulgar i donar èmfasis a utilitzar com una part integral del control administratiu.

El millorament de la qualitat es compon de tres tipus d'accions: control de qualitat, millora de nivell i planejament de la qualitat.

Armand V. Feigenbaum (Nascut al 1922)

Pare de l'expressió Control Total de la qualitat (TQC) que després es derivaria en Gestió de la Qualitat Total (TQM).

Control Total de la Qualitat és un sistema eficaç per a la integració del desenvolupament, manteniment, millora de la qualitat i els esforços dels diversos grups de l'organització, amb el fi de que la producció i serveis permetin la plena satisfacció del client.

El Control de qualitat resulta efectiu quan s'inicia amb el disseny del producte i es finalitza només quan es trobi en mans del consumidor satisfet.

William Deming (1900 - 1993)

Més que un autor va ser un gran divulgador, sobretot al Japó, dels conceptes de la qualitat aplicada a la indústria.

El cicle de Deming, originari però de Walter Shewhart és probablement la seva aportació més coneguda. A Japó els premis de qualitat són els premis Deming i proposa la qualitat d'un sistema software governat des de la qualitat dels processos utilitzats pel desenvolupament i manteniment enfocat a la millora de la qualitat.

La qualitat és defineix, des del punt de vista del client, com qualsevol cosa que augmenta la satisfacció.



3.2 Assegurament de la qualitat

L'assegurament de la qualitat és una estratègia fonamental a les organitzacions. Es pot definir com l'esforç total per plantejar, organitzar, dirigir i controlar la qualitat en un sistema amb l'objectiu de donar al client productes amb la qualitat adequada, és a dir, assegurar que la qualitat sigui el que ha de ser.

Un producte ha d'aconseguir un nivell acceptable, però aquest pot variar depenent del tipus i dels mètodes escollits i seguits en les diferents organitzacions.

Com es pot identificar quants defectes o errors pot tenir un producte per considerar que s'està produint amb la qualitat esperada?

Doncs depèn de l'organització, perquè si es tracta d'una indústria un error d'un 0,1% (1 de 1000) pot ser acceptable, però si es tracta d'un hospital en un determinat sector pot ser no acceptable.

Les necessitats del client s'exposen en els requisits i aquests descriuen amb detall la qualitat amb la que ha de resultar el producte per ser acceptable.

Els requisits son la base de les mesures de la qualitat i sinó estan ben definits i entesos provoca la falta de qualitat en els productes.

- Els requisits es desenvolupen abans d'iniciar un desenvolupament i poden ser modificats en les diferents etapes del producte.
- S'expressen de forma funcional i documentats.
- Els requisits de societat es tenen en compte també. Son aquells establerts per lleis, normes, seguretat, etc.

L'assegurament de la qualitat conté un conjunt d'activitats planificades i sistemàtiques necessàries per obtenir la confiança del producte software i satisfà els requisits de qualitat establerts. Inicialment s'estableixen aquest conjunt d'activitats mitjançant el qual es desenvolupa el producte.

Es un procés que defineix com aconseguir la qualitat del software. S'ocupa d'escollir els estàndards que han de ser aplicats i les eines i mètodes que suporten aquests estàndards.

Els estàndards moltes vegades es consideren de difícil aplicació a les empreses perquè estableixen normatives per omplir documentació i realitzar tasques d'una certa manera, els equips poden considerar els estàndards com una pèrdua de temps que no afecten a les seves tècniques de desenvolupament de les tasques.

Si s'estudia una mica sobre aquests estàndards hi ha aspectes que els equips troben evidents però que provoquen fallides per no ser aplicats o ser aplicats sense prèvia planificació el que pot provocar haver de realitzar regressions en algunes tasques ja implementades.

L'assegurament de la qualitat es porta a terme durant tot el procés de desenvolupament d'un software i engloba:

- ✓ La gestió de la qualitat.
- ✓ Mètodes i eines efectives en l'Enginyeria de software.
- ✓ Revisions periòdiques de seguiment del producte.
- ✓ Estratègia i pla de proves.
- ✓ Control de la documentació i modificacions realitzades al llarg del projecte.
- ✓ Aporta procediments basats en estàndards de desenvolupament software.
- ✓ Conté mecanismes de mesura, report i generació d'informes.

Gestió de la qualitat

Engloba el Control de Qualitat i l'Assegurament de la Qualitat. Mentre el Control de Qualitat recull les inspeccions, revisions, proves i medis operatius utilitzats per satisfer els requisits per la qualitat, l'Assegurament de la Qualitat abasta l'aplicació i compliment de les activitats planificades que demostren la satisfacció dels requisits per la qualitat tant internament en l'organització com externament pels clients.

La gestió de la qualitat suposa la garantia de la qualitat dels productes prèviament pactada amb el client, en el termini establert, amb el menor cost, millor preu, aconseguint els beneficis esperats per l'empresa i alhora pel client.

Entorn al control intern de l'empresa, el concepte 'Sistema de qualitat' engloba la gestió, direcció i control d'una organització respecte a la qualitat, fent-se possible amb la participació i compromís de tots els membres de l'organització per aconseguir els objectius de qualitat proposats. No s'encarrega de l'inspecció dels requisits amb el client, sinó a les accions internes de l'organització.

Qualitat d'un procés i d'un producte software

La qualitat d'un procés es reflexa a la qualitat d'un producte. La gestió de la qualitat entorn als processos de desenvolupament afecten directament a la qualitat al llarg del projecte i en el seu alliberament.

Si la qualitat no es té en compte perquè el producte sigui de fàcil utilitat i manteniment, fiable, eficient, etc. no es podrà arribar a l'assegurament de la qualitat del producte desenvolupat patint sèries conseqüències en el mercat.

Al inici d'un projecte sembla que el més important sigui arribar a les dates d'entrega de l'usuari final i començar lo abans possible a desenvolupar el codi, però aquest fet pot portar greus impactes ja que la qualitat del producte no serà la esperada i forçarà a situacions de refer el codi de l'aplicació per no estudiar i analitzar prèviament aspectes que garanteixin la qualitat del producte.

Es difícil determinar la qualitat, però és un aspecte que s'ha de trobar present des de l' inici d'un projecte per assegurar la qualitat del producte.

La gestió de la qualitat implica:

- ✓ Definir estàndards en els processos, les revisions a realitzar, en quin moment i com executar-les, etc.
- ✓ Supervisar el desenvolupament per assegurar que es segueixin els estàndards.
- ✓ Realitzar informes del procés pels gestors de projectes i pels usuaris finals.

Un problema de l'assegurament de la qualitat basada en processos és que l'equip de QA insisteixi en els estàndards de processos independent al software a desenvolupar.

La qualitat dels processos ha de ser valorada abans d'implementar una aplicació i en el cas de no ser acceptada per l'equip de QA no haver de desfer projectes ja iniciats.

Planificació de la qualitat

La planificació de la qualitat defineix i documenta com es compliran els requisits i objectius de qualitat:

- ✓ Planificació del producte: engloba l' identificació, classificació, objectius i restriccions.
- ✓ Planificació administrativa i operativa: abasta la preparació del sistema de qualitat de l'organització i la programació.
- ✓ Preparació de plans de qualitat i disposició el millorament de la qualitat.
- ✓ Identificació i adquisició de medis de control, processos, equips, recursos per aconseguir la qualitat desitjada.
- ✓ Assegurar la compatibilitat amb el disseny, procediments d'inspecció i de documentació.
- ✓ Identificació de requisits amb el temps suficient per desenvolupar de forma efectiva.
- ✓ Identificació de les verificacions en les diferents etapes del producte.
- ✓ Aclariment de les normes d'acceptació per totes les característiques i requisits del producte.

Es un procés en el qual es desenvolupa un pla de qualitat pel projecte. Aquest pla defineix la qualitat desitjada del software i descriu com s'ha d'aconseguir aquesta.

Sense la definició d'un pla de qualitat els implicats en un projecte poden seguir direccions diferents en el mateix projecte.

El pla de qualitat defineix els mètodes, eines i estàndards apropiats que poden variar pels diferents projectes. Humphrey, 1989 en el seu llibre sobre gestió del software suggereix una estructura per un pla de qualitat: ³

- *Introducció del producte*

Conté la descripció del producte, al mercat al que es dirigeix i les expectatives de qualitat.

- *Plans de producte*

Conté les dates de finalització del producte, les responsabilitats i plans per la distribució i servei.

- *Descripcions del procés*

Conté els processos de desenvolupament i de servei a utilitzar pel desenvolupament i administració del producte.

- *Metes de qualitat*

Conté les metes i plans de qualitat pel producte, inclou la identificació i justificació dels atributs de la qualitat.

- *Riscos i gestió de riscos*

Conté els riscos claus que podrien afectar a la qualitat del producte i a les accions per abordar aquests riscos.

3.3 Objectius d'un projecte de qualitat

Cada aproximació al desenvolupament de software està basada en uns objectius. Per això la metodologia seguida en un projecte ha de recollir els objectius generals del desenvolupament, és a dir, que els objectius estiguin implementats en la metodologia de desenvolupament.

Els objectius s'han de saber abans de començar un projecte amb claredat i amb un previ estudi del projecte a tractar. A vegades la definició dels objectius no és feina fàcil, però sempre hi ha aspectes que s'han d'aconseguir: Cost, Termini i Resultat.



³ [ISHUM] Ingeniería del Software, Ian Sommerville (Humphrey, 1989)

El Cost i el Termini normalment son pactats amb l'usuari final i s'ha de respectar i estan definits des del principi. El Termini és l'objectiu que fàcilment pot ser necessari la seva mobilitat i no respecta lo suficient i és amb el que es pot demostrar i mesurar el grau de qualitat de la gestió del projecte.

El Resultat del projecte és l'objectiu més significatiu però una dolenta gestió dels demés no porta a un bon resultat i lo important no és arribar a un resultat en el cost i termini prèviament exposat, sinó arribar a un Resultat esperat i desitjat amb la qualitat necessària.

La modificació de qualsevol dels objectius afecta als demés ja que no es poden separar, és decisió de la organització el fet de modificar qualsevol d'ells o la seva combinació depèn de l'estratègia seguida en cada projecte.

La combinació dels tres objectius no és única, es pot exposar una zona de validesa en la que el projecte pot desplaçar-se depenent de la disponibilitat dels recursos existents. Les tècniques de gestió de projectes han de considerar les actuacions relacionades amb les desviacions dintre de la zona permesa i aplicar mesures correctores per evitar problemes addicionals. Això implica que alguna desviació es podria provocar per arribar a complir altres objectius com 'la satisfacció del client'.

La satisfacció del client és un objectiu que estableix l'èxit del projecte. Per molt que es compleixi els altres objectius, que es realitzi el projecte en el termini i cost establert i que s'arribi un resultat final, si la satisfacció del client no és positiu no es compleixen els objectius d'un projecte.

4 Processos i models de qualitat software

El procés de desenvolupament software descriu el cicle de vida d'un producte software des de la seva entrega, la utilització i el manteniment fins a la fi d'aquest producte. Establir un model de cicle de vida de software ajudarà a dur un projecte amb èxit i menys possibilitat d'errors i detecció ràpida dels que puguin sorgir al llarg del desenvolupament.

Els models de millora dels processos software indiquen pautes que les organitzacions han de seguir per millorar els processos de desenvolupament i així millorar la qualitat del producte.

Els processos de prova descriuen com el producte software ha de ser provat des de el seu inici. Igual que hi ha processos de millora per al procés de desenvolupament, hi ha models de millora dels processos de prova.

També existeixen models que avaluen per poder millorar aquests models i defineixen mides per poder avaluar el procés i el producte, sempre amb la finalitat de millorar la qualitat.

S'inicia el capítol amb la descripció del cicle de vida del software i la importància de definir un cicle de vida.

Per tal de situar el projecte en el entorn de desenvolupament del software en els següents punts es detalla les etapes i processos del cicle de vida del software basats en els estàndards desenvolupament software i qualitat sobre el cicle de vida i assegurement de la qualitat del software.

Un cop definits els processos es resumeixen alguns models de cicle de vida del software com model lineal seqüencial, en espiral, en V, etc.

Posteriorment es presenten els models relacionats amb la qualitat i millora dels productes i processos de software com l'estàndard ISO/IEC 9126 i el model *Capability Maturity Model Integration* (CMMI) i models de proves *Test Management Approach* (TMAP) *Testing Maturity Model* (TMM) i *Test Process Improvement* (TPI).

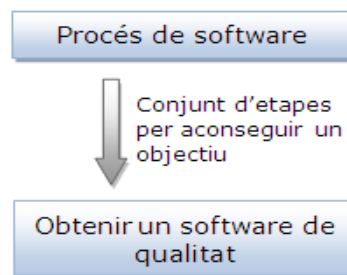
4.1 Procés i cicle de vida del Software

El cicle de vida del software és un procés de desenvolupament que segueix el software des de que és un simple projecte fins que es deixa d'utilitzar havent estudiat el seu origen, l'anàlisi, les funcionalitats, les restriccions, realitzar el seu disseny, la construcció, les proves, la instal·lació, la implantació, el manteniment, etc... totes les fases per les que passa un producte al llarg de la seva vida i la forma en que es relacionen entre sí aquestes fases.

Els sistemes de software requereixen de temps i esforç pel seu desenvolupament i s'han d'utilitzar durant un període de temps superior. Durant aquest temps de desenvolupament i ús del software s'identifiquen diferents etapes que en conjunt són les que constitueixen el cicle de vida del software, que en cada cas depenen de

les característiques del software es treballarà el cicle de vida de manera diferent. Aquestes etapes consten d'un conjunt d'activitats requerides per desenvolupar un sistema de software que varien depenent de la organització i del tipus de sistema de desenvolupar.

No obstant, no existeix un procés estàndard de fabricació del software, pot ser un procediment, un estàndard o una política. Cada cas es pot resoldre de manera diferent però hi ha unes pautes generals per organitzar el procés, hi ha activitats que es repeteixen cada vegada en la construcció de qualsevol software. Aquestes activitats necessiten d'una organització en la seva realització per això s'estableixen les metodologies de desenvolupament de software.



Per què és important el cicle de vida d'un software?

- El cicle de vida d'un software facilita el control sobre els temps en que és necessari aplicar tot tipus de recursos (personal, equips, subministraments, ...) al projecte.
- El control de qualitat també es facilita si la separació entre les fases es correspon amb els punts que s'han de verificar (mitjançant comprovacions sobre els productes parcialment obtinguts). Amb una bona definició de les fases es podrà validar al llarg del desenvolupament de l'aplicació i poder garantir que el software compleix amb els requisits per la aplicació.
- Amb fases ben definides es pot garantir la verificació dels procediments en el desenvolupament del software assegurant que els mètodes utilitzats son els apropiats.
- La pràctica acumulada en el disseny dels models de cicle de vida per situacions diverses permet que ens beneficiem de l'experiència adquirida utilitzant un enfocament o un altre que millor s'adapti als requeriments desitjats.

Aquest estudi s'origina pel fet de que és molt costos rectificar els errors que es detecten tard dintre d'una fase d'implementació. El cicle de vida permet que els errors es detectin lo abans possible i així permet als desenvolupadors del software alliberar un software de més qualitat.

4.1.1 Etapes i elements del cicle de vida del software

Gràcies a la metodologia establerta un projecte es divideix en mòduls més petits també denominades etapes, i les accions que corresponen a cadascuna de les etapes defineixen les entrades i sortides de cadascuna d'elles i així administrar el projecte adequadament.

Les etapes es poden delimitar de certa manera però l'ordre o l'execució d'aquestes depèn del model escollit pel projecte a determinar. Des d'un punt de vista general es pot considerar que el cicle de vida d'un software té etapes clarament definides, encara que poden haver-hi moltes més:

- *Planificació*

Es realitza el plantejament detallat de la gestió del projecte, temporal i econòmicament.

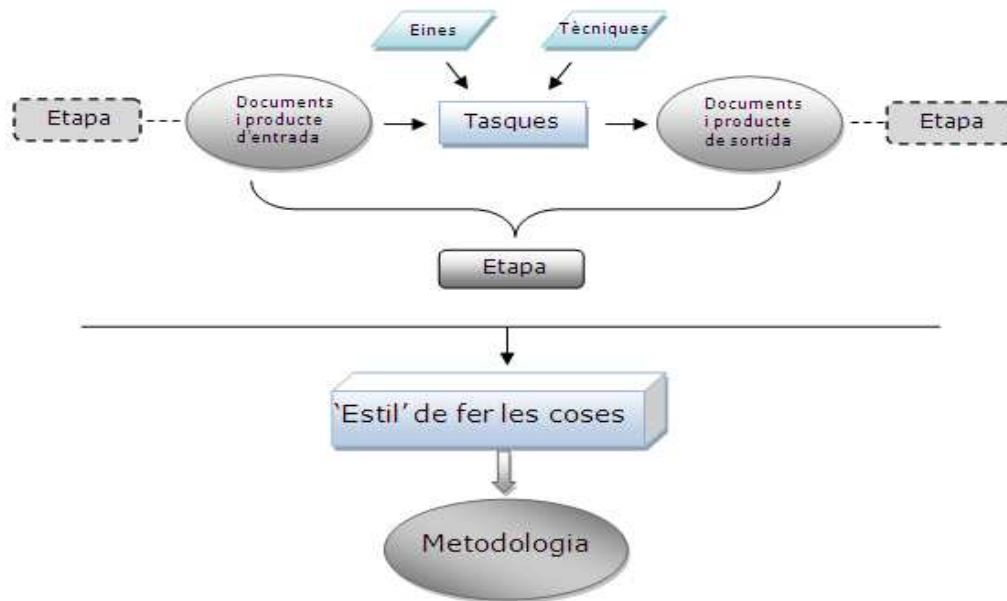
- *Implementació*

Es representa amb el conjunt d'activitats que componen la realització del producte.

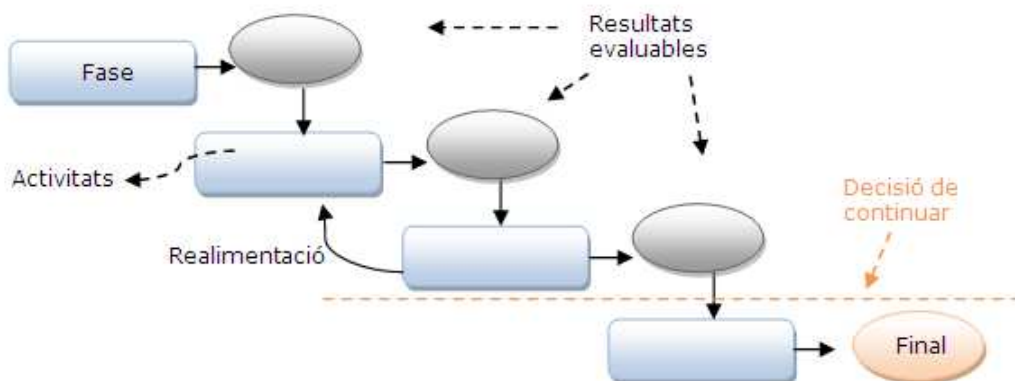
- *Producció*

El projecte entra en una etapa de definició on es presenta al usuari final o client, saben que funciona correctament i respon als requeriments sol·licitats en el seu moment. Aquesta etapa és molt important no només perquè s'espera l'acceptació del client sinó perquè poden sorgir imprevistos d'instal·lació provocant costos no previstos.

Les etapes consten de tasques i un aspecte essencial dintre de les tasques del desenvolupament és la 'Documentació' que es troba present en totes les etapes. Cada etapa té com entrada un o varis documents procedents d'etapes anteriors i produeix uns altres documents de sortida.



Les tasques d'un projecte componen fases successives que determinen el cicle de vida d'un software. Segons el model de cicle de vida aplicat la successió de fases es poden ampliar en bucles de realimentació, de manera que el que es considera un fase es pot executar més d'una vegada al llarg del projecte i recollint tota la informació dels resultats entremitjos que es van obtenint de cada execució.



Per seguir amb un control adequat de les diferents fases és necessari especificar els resultats avaluable de les execucions, és a dir, dels productes entremitjos que resulten de les tasques de cada fase.

Els diferents elements que integren el cicle de vida son les fases i les entregues.

- **Fases:**

Resulta del conjunt d'activitats relacionades amb l'objectiu de desenvolupar un software. Aquestes activitats s'agrupen en tasques que es poden compartir en el cicle de vida del software. La agrupació i compartició de les tasques fa imprescindible l'assignació de recursos (persones, materials, ...)

- **Entregues:**

Son els productes entremetjats que generen les fases. Poden ser materials (components, equips) o poden no ser materials (documents, software). Aquestes entregues permeten avaluar l'estat del projecte mitjançant comprovacions respecte els requisits funcionals i condicions prèviament establertes i així poder solucionar possibles errors en el menor avanç possible del projecte. D'aquesta manera es pot prendre decisions al llarg del desenvolupament del projecte en cas de possibles desviacions.

ISO IEEE/EIA 12207 Cicle de vida d'un software

La ISO, International Organization for Standardization, a la seva norma ISO/IEC 12207 defineix el cicle de vida del software com un marc de referència que conté les activitats i les tasques involucrades en el desenvolupament, la explotació i el manteniment d'un producte software, amb un abastament des de la definició fins la finalització del seu ús.

Aquest estàndard té com a objectiu principal proporcionar una estructura comú per tot el personal (tècnics, gestors, desenvolupadors, .. implicats en el desenvolupament del software per utilitzar un llenguatge comú i aquest llenguatge s'aconsegueix amb processos ben definits.

4.1.2 Processos del cicle de vida del software

El conjunt de pràctiques realitzades per obtenir un resultat i desenvolupar software, ja siguin eines, tècniques, materials, persones, ... tot el conjunt és el procés.

Tot procés que segueix el desenvolupament del software és el cicle de vida del software, descriuen els passos que relacionen les activitats, restriccions i recursos que produeixen una determinada sortida.

El procés respon a les preguntes: QUÉ ? i COM ?, en canvi un projecte respon a les preguntes: QUI ? I QUAN ?



Els processos es classifiquen en tres tipus:

ISO 12207

5 processos principals

8 processos de suport

3 processos de la organització

1 model de cicle de vida

➤ Processos principals:

- Adquisició: activitats i tasques de compra d'un producte, sol·licitud d'ofertes, selecció de subministradors i gestió de tasques des de l'adquisició fins l'acceptació.
- Subministrament: activitats i tasques que realitza el subministrador, proposta per la petició del comprador, identificació de procediments i recursos per garantir l'execució del projecte.
- Desenvolupament: Anàlisi, disseny, codificació, integració, proves, instal·lació i suport.
- Explotació: Explotació del software, suport als usuaris i activitats i tasques que s'apliquen al procés.
- Manteniment: Per modificar el software (error, deficiència o millores), consistència i migració o retirada del software.

Els processos principals depenen de cada situació en particular i resulten útils per compradors, desenvolupadors, operadors i personal de manteniment.

Els processos de suport i de la organització han d'existir independentment de la organització i projecte executat.

➤ Processos de suport:

- Documentació: Registre de la informació, activitats per planificar, dissenyar, desenvolupar, produir, editar, distribuir i mantenir documents.
- Gestió de la configuració: Control de les modificacions i versions, registre i informe d'estat de les modificacions, assegurar la correcta correcció, control de l'entrega, manipulació i emmagatzematge.
- **Assegurament de qualitat (QA):** asseguruen que els productes de treball i processos compleixen amb els requisits predefinitos.
 - Es desenvolupa una estratègia per dur a terme el assegurament de la qualitat.
 - Es produeix i manté un assegurament de la qualitat.
 - S'identifiquen i es registren els problemes o disconformitats amb els requisits principals.

- Es verifica el compliment per part dels productes, processos i activitats dels estàndards, procediments i requisits aplicables.
 - Validació
 - Verificació
 - Revisió continua
 - Auditoria
 - Resolució de problemes.
- Processos de la organització (gestió, formació del personal, millores del procés d'infraestructura, a nivell organitzatiu estaria fora del projecte):
- Gestió: Planificació, revisió, avaluació, seguiment i control.
 - **Gestió de la qualitat:** L'objectiu principal es aconseguir la satisfacció dels clients, monitoratge de la qualitat dels productes i serveis, a nivell d'organització i del projecte, demostrant i assegurant que aquests satisfan els requisits dels clients. La qualitat es pot assegurar tenint en compte els següents punts:
 - Establir *objectius* de qualitat basats en els requisits de qualitat definits prèviament pel client i l'organització.
 - Desenvolupant una *estratègia* global per aconseguir els objectius definits.
 - Establir un sistema de gestió de qualitat per implementar aquesta estratègia.
 - Realitzant les activitats de *control i assegurament* de la qualitat identificada.
 - *Monitoratge* dels objectius de qualitat, és necessari demostrar visualment la informació.
 - Prendre les *accions* oportunes quan no s'aconsegueixen els objectius de qualitat.
 - Infraestructura: Hardware, software, eines.
 - **Procés de millora:** Valorar, mesurar, controlar i millorar els processos del cicle de vida.
 - Aquest procés és imprescindible per dur a terme una millora continua dels processos utilitzats al llarg del cicle de vida del software i així mantenir les necessitats de negoci dels productes.
 - Les fonts d'informació que poden proporcionar les entrades pel canvi son: resultats de la valoració dels processos, auditories,

informes de satisfacció del client, eficiència i efectivitat de la organització i cost de la qualitat.

- L'estat actual dels processos podria determinar-se amb un procés de valoració:
 - establiment dels processos
 - valoració dels processos
 - millora dels processos
- Formació: Formació del personal, desenvolupament del material de formació, pla de formació.

ISO IEEE/EIA 12207 Cicle de vida d'un software:

Objectius del procés d'assegurament de la qualitat (QA)

- *Identificar i planificar les activitats d'assegurament de la qualitat pel procés o producte*
- *Identificar estàndards, metodologies, procediments i eines per dur a terme les activitats d'assegurament de la qualitat i la seva adaptació al projecte.*
- *Identificar els recursos i les responsabilitats per la realització de les activitats d'assegurament de la qualitat.*
- *Establir i garantir la independència dels responsables de dur a terme l'assegurament de la qualitat.*
- *Realitzar les activitats identificades d'assegurament de la qualitat en consonància als plans i procediments rellevants.*
- *Aplicar els sistemes de gestió de qualitat d'organització en el projecte.*

Plans per l'assegurament de la qualitat

- *Informació d'un pla genèric per l'assegurament de la qualitat del software.*
- *Estàndards de qualitat, metodologies, procediments i eines per dur a terme les tasques d'assegurament de qualitat.*
- *Procediments per la identificació, col·lecció, classificació i disposició dels registres de qualitat.*
- *Recursos, programacions i responsabilitats per dirigir les activitats d'assegurament de la qualitat.*
- *Activitats i tasques seleccionades a partir dels processos de suport, com la Verificació, Validació, Revisions, Auditories i Resolució de Problemes.*

4.1.3 Models del cicle de vida del software

Els models del cicle de vida del software serveixen per definir les fases durant el seu desenvolupament, proporcionen l'ordre en el qual un projecte ha de realitzar les seves tasques i les principals diferències entre els diversos models tenen relació amb:

- *Abastament del cicle de vida* depenent del punt on es vulgui arribar amb el projecte, és a dir, si es vol saber si és viable el desenvolupament del producte, si es vol saber el desenvolupament complet o si es vol el desenvolupament complet i a més el manteniment i actualitzacions.
- *Qualitat i quantitat d'etapes* en que es divideix el cicle de vida, depenen del cicle de vida i del projecte.
- *Estructura i successió d'etapes*, si hi ha realimentació entre elles i si hi ha llibertat per repetir-les.

El primer model utilitzat als principis del software va ser "Code & Fix" que té dos etapes: Primer escriure el codi que resolgui un problema donat i segon solucionar els problemes del codi. Aquest model no té en compte les necessitats del client ja que primer codifica i després pensa en els requeriments, disseny, proves i manteniment.

El model lineal seqüencial divideix l'activitat global del projecte en fases amb un enfocament seqüencial, això suposa que cadascuna es realitza una sola vegada, és molt fàcil de planificar, molt rígida i s'ha de tenir tota la informació abans de començar. Cada fase comença quan s'ha acabat l'anterior, per poder passar a la següent fase s'han de complir tots els objectius marcats i al final de cada fase tant usuaris com personal tècnic tenen l'oportunitat de revisar conjuntament el progrés del projecte.

Les etapes són: Captura de requisits, Anàlisi i Disseny, Codificació, Proves i Manteniment.

La verificació i validació (proves) pretén demostrar que un sistema és conforme amb les especificacions i que resolen els requisits del client.

Com desavantatge les especificacions mai són correctes i el client no veu res fins al final.

Com avantatge és molt senzill de gestionar i les etapes es troben ben diferenciades i és el més efectiu a nivell de costos perquè permet una planificació on fer un seguiment és més simple (baseline).

El model iteratiu realitza iteracions dels cicles de vida en cascada i de cadascuna s'obté un prototip fins satisfer els requisits. Amb aquest model pot afectar a la qualitat del software però com avantatge el client pot obtenir informació en cada prototip i pot tenir una versió abans. En el desenvolupament iteratiu s'entrega un sistema complet des del principi i després es millora la funcionalitat de cada subsistema amb cada versió. Entre els més populars hi ha el *Rational Unified Process* (RUP) i *eXtreme Programming* (XP).

El model incremental realitza iteracions obtenint diferents entregues de forma incremental així es pot controlar millor la qualitat, els errors greus afecten només a la última iteració i es poden realitzar entregues al client. Els requisits són prioritzats en les primeres entregues. La desavantatge respecte l'anterior és que la gestió és més complicada, els serveis del sistema amb prioritat més alta tendeixen a ser més provats i pot ser difícil ajustar els requisits als increments.

El model en espiral té en compte els riscos, s'obtenen prototips i es repeteix l'espiral fins a satisfer els requisits. Amb aquest model és possible controlar la qualitat, el client pot anar aprovant cada prototip, cada volta en l'espiral representa una fase del procés, és a dir, no hi ha fases fixes, cada volta en l'espiral determina les activitats a realitzar.

La definició d'un model que s'adapti a un pla de proves ha de respectar les següents etapes:

- Anàlisi de les proves
- Disseny de les proves
- Execució de les proves
- Avaluació dels resultats i introducció de millores

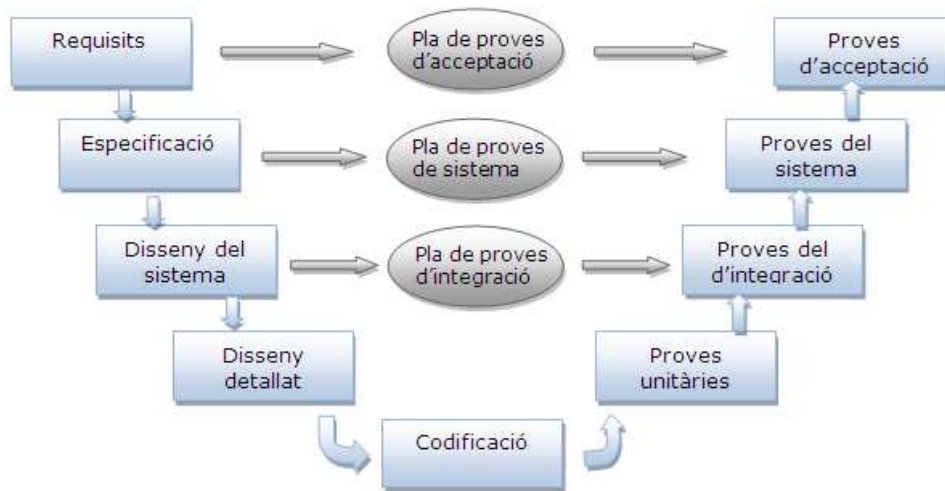
Model en V

El model de proves més extens és el model en V (Verificació i Validació) ja que a diferència que altres models clàssics, extens les proves al llarg de tot el cicle de vida del software.

El model en V és una variació del model lineal seqüencial que demostra com es relacionen les activitats de prova amb l'anàlisi i el disseny.

Es similar al model lineal seqüencial però té una part que té com a finalitat realitzar les proves i integració a cadascuna de les etapes. Té una avantatge important respecte el model lineal seqüencial i és que fa comprovacions a cadascuna de les etapes. Però com tots els models també té desavantatges: les proves d'acceptació es realitzen al final de cada etapa i es comencen un cop acabada la implementació i això provoca que es pugui anar enrere si es detecten errors provocant una pèrdua de temps i diners. Tampoc contempla retornar enrere a les etapes cosa que en la realitat pot passar fàcilment.

Encara que hi hagi aspectes negatius és un model que pot donar qualitat al software podent completar el model amb un desenvolupament iteratiu considerant la V com iteracions del projecte. Aquest model és suficient en molts projectes ja que des de les primeres etapes es preocupa del disseny de les proves que més tard seran executades i el pla de proves anirà evolucionant al llarg del projecte.



4.1.4 Estàndards del Cicle de vida dels processos

IEEE 1074-1998 Developing Software Life Cycle Processes

Defineix les activitats que constitueixen els processos necessaris pel desenvolupament i manteniment del software, ja sigui per part d'un sistema major o autònom.

Els processos de gestió i suport al llarg del cicle de vida.

IEEE/EIA (ISO/IEC) 12207 Information technology – Software life cycle processes

Estableix el marc comú dels processos de cicle de vida del software, implementa terminologia ben definida i descriu el cicle de vida (des de la definició dels requisits fins al final de la seva utilitat) i conté processos per adquirir i subministrar productes i serveis software.

Estableix el cicle de vida com un marc de referència que conté processos, activitats i tasques involucrades en el desenvolupament, la explotació i el manteniment d'un producte de software, enfocant la vida del sistema des de la definició dels requisits fins la finalització de l'ús.

- Indica els processos, activitats i tasques durant la adquisició de:
 - Un sistema que conté software
 - Un producte que conté software autònom
 - Un servei software

I durant el subministrament, desenvolupament, operació i manteniment de productes software.

- Proporciona processos per definir, controlar i millorar els processos de cicle de vida de software.
- El marc de l'estàndard està dissenyat per se adaptat a cada organització i projecte.
- El procés d'adaptació consisteix en la eliminació de processos, activitats i tasques no aplicables.
- Processos:
 - Principals: útils per persones que inicien o realitzen el desenvolupament, l'explotació o el manteniment del software durant el cicle de vida (compradors, subministradors, personal de desenvolupament, operadors, personal de manteniment del software)
 - Suport: serveixen de suport, contribueixen a l'èxit i qualitat del projecte software i es poden aplicar en qualsevol moment del cicle de vida del software.
 - De la organització: permeten establir, implementar i millorar la organització (gestió, formació, ...)
 - Adaptació: Permet adaptar l'estàndard a cada projecte i organització.

ISO 15288 Systems and Software Engineering - System Life Cycle Processes.

Activitats del procés de gestió de qualitat:

- Establir polítiques, estàndards i procediments de gestió de qualitat.
- Establir objectius de gestió de qualitat de la organització basades en la estratègia empresarial per la satisfacció del client.
- Definir les responsabilitats i autoritats per implementar la gestió de la qualitat.
- Avaluar i informar sobre la satisfacció del client.
- Dur a terme revisions periòdiques de plans de qualitat de projectes.
- Monitoratge de l'estat de les millores de qualitat dels productes i serveis.

4.2 Models de procés, qualitat i millora

A aquest capítol es presenten alguns models de qualitat que formen el conjunt de pràctiques pel cicle de vida del software enfocat en el desenvolupament de projectes.

Al parlar de processos de desenvolupament de software, no es parla només en termes tècnics perquè està demostrat que la majoria de problemes que sorgeixen als projectes son organitzatius. Cada cop més organitzacions es preocupen per l'implantació de models de qualitat en el desenvolupament del software.

Els models serveixen d'eines per guiar a les organitzacions a la millora continua i la competitivitat per aportar bons productes i serveis al mercat. La implantació de sistemes de qualitat aporten beneficis a les organitzacions ja que a més de reduir costos aconseguixen la satisfacció dels clients i la millora per la motivació dels empleats.

Alguns dels factors que determinen la qualitat del software son:

- ✓ *Correcció* → Fa el que jo vull?
- ✓ *Fiabilitat* → És fiable tot el temps?
- ✓ *Eficiència* → S'executarà bé des de el meu hardware? I de la millor manera?
- ✓ *Seguretat* (integritat) → És segur?
- ✓ *Facilitat d'ús* → Està dissenyat per ser utilitzat?

4.2.1 Models estàndards

Els models estàndards ajuden a definir i conceptualitzar la qualitat Software i així poder arribar a mesurar i planificar la qualitat d'un producte.

Els estàndards son útils perquè:

- ✓ Reuneixen lo millor de les bones pràctiques i utilització del desenvolupament software.
- ✓ Engloben els coneixements que son patrimoni d'una organització.
- ✓ Proporcionen un marc per implementar procediments d'assegurament de la qualitat.
- ✓ Proporcionen continuïtat entre el treball de les diferents persones.

Dins dels estàndards de l'Enginyeria del Software hi ha els estàndards relacionats amb el procés software, a continuació alguns exemples:

Família ISO 9000 → QUALITAT

- IEEE – *Institute of Electrical and Electronics Engineers*
- ISO – *International Organization for Standardization*
- IEC – *International Electrotechnical Commission*

Normes ISO 9000 (anys 90)

Son estàndards per la gestió de qualitat de qualsevol procés de producció adoptat per la majoria d'organitzacions. ISO 9000 garanteix el Assegurament de la Qualitat gràcies a la fiabilitat dels processos i es pot aplicar a qualsevol negoci independentment dels productes i serveis oferts.

- *Punts forts*: Factor competitiu per les empreses, estalvi de temps i evita haver de demostrar la qualitat moltes vegades, implantat en tot tipus d'organitzacions i garantia de fer les coses ben fetes.
- *Punts dèbils*: Estàtic i car, en qüestió de temps deixa de ser competitiu, adoptat per moltes organitzacions per cobrir expedient i hi ha diferències en les interpretacions de les clàusules de l'estàndard.

Cal destacar les normes d'assegurament de qualitat:

ISO 9001 – Assegurament de la qualitat en organitzacions que el seu procés abasta des del disseny fins al servei postvenda.

ISO 9002 - Assegurament de la qualitat en organitzacions que la seva activitat es basa en fases de producció i instal·lació.

ISO 9003 - Assegurament de la qualitat en organitzacions que la seva activitat consisteix en inspecció i assajos finals.

ISO 9126

Es un model de qualitat com estàndard sobre mètriques que cobreixen els aspectes sobre la qualitat del software i estableix les característiques que han de ser avaluades d'un producte software.

El model descriu la qualitat d'un producte en dos parts:

- Qualitat interna i qualitat externa especifiquen les sis característiques de la qualitat, que son: Funcionalitat, confiança, utilitat, eficiència, manteniment i portabilitat.

- Qualitat en l'ús especifica quatre característiques de la qualitat d'ús, com son: Eficàcia, Productivitat, Seguretat física i Satisfacció. Aquesta és la qualitat conjunta de les sis característiques de la qualitat del producte.

La qualitat del procés contribueix a millorar la qualitat del producte i la qualitat del producte millora la qualitat de l'ús.

Cada característica de la qualitat del producte ha de ser especificada i avaluada utilitzant mètriques validades. Les mètriques internes s'apliquen a un producte de software durant el disseny i la codificació, les mètriques externes utilitzen mesures d'un producte derivat del comportament del producte del que forma part mitjançant proves, operacions i funcionament del software executat i les mètriques de qualitat d'ús mesuren el grau en el que un producte resol les necessitats requerides pels usuaris.

Factors de la qualitat software

ISO planteja un model genèric de qualitat a la norma ISO 9126 i es divideix en 6 conceptes: ⁴

➤ *Funcionalitat*

- El software satisfà les necessitats requerides i declarades.
- El producte proporciona els resultats correctes amb la precisió necessària.
- El software interactua amb més d'un sistema.
- Seguretat per prevenir els accessos no desitjats, protecció de dades i programes.
- El software compleix amb els estàndards, lleis i protocols legals.

➤ *Fiabilitat*

- Capacitat d'un sistema per mantenir el seu nivell de rendiment i recuperació de dades.
- Capacitat per evitar la quantitat de fallides amb la que podria fallar.

➤ *Facilitat d'ús*

- Esforç necessari per l'ús i la valoració individual de l'ús, per part d'un conjunt d'usuaris.
- Capacitat d'operar i control d'usuaris.

⁴ [ISO9126] La calidad del software y su medida, Norma ISO/IEC 9126

➤ *Eficiència*

- Relació entre el nivell de prestacions d'un sistema i el volum de recursos utilitzats en condicions declarades.
- Temps de resposta òptim en les diferents funcions.

➤ *Fàcil manteniment*

- Esforç necessari per realitzar modificacions específiques.
- Capacitat del software per evitar accions no esperats després de les modificacions.

➤ *Portabilitat*

- Capacitat d'un sistema per ser transferit d'un entorn a un altre.
- Esforç per instal·lar un software en un entorn específic i adaptar a entorns ja existents.

Model de McCall ⁵

El model proposat per McCall (1977) estableix que la qualitat d'un producte software es centra en tres aspectes de mesura del software:

➤ *Operació del producte*

- Correcció. Fa el que vull?
- Fiabilitat. Ho realitza de manera fiable?
- Eficiència. S'executa de la millor manera des del hardware?
- Seguretat. És segur?
- Facilitat d'ús. Està dissenyat per ser provat?

➤ *Capacitat de suportar canvis (revisió del producte)*

- Facilitat de manteniment. Es pot corregir?
- Flexibilitat. Es pot modificar?
- Facilitat de prova. Es pot provar?

➤ *Adaptació a nous entorns (transició del producte)*

- Portabilitat. Es pot utilitzar des de una altra hardware?
- Reutilitzable. Es podria tornar a utilitzar algunes parts del software?
- Fàcil d'interactuar. Pot interactuar amb una altre sistema?

⁵ [McCall] La calidad del software y su medida. ISO/IEC 9126, Factores del modelo de McCall

Estàndards IEEE

Estan orientats a l'assegurament de la qualitat a nivell de projecte, a continuació alguns exemples:

IEEE 730 – proporciona plans d'assegurament de qualitat software.

IEEE 829 – Documentació de proves de software.

IEEE 1012 – Verificació i validació del software.

IEEE 1028 – Revisions del software.

IEEE 1061 – Mètriques per productes i processos.

L'estàndard IEEE 829-1998, IEEE Standard for Software Test Documentation

La Documentació és important en el procés de proves i aquest estàndard descriu els tipus de documents que poden produir-se durant el procés de prova. Estableix un pla de proves a seguir dividit en vuit etapes amb cadascuna el seu tipus de document. L'estàndard especifica el format d'aquests documents però no com s'ha d'establir ni quin criteri seguir en relació al contingut amb aquests documents. Els documents exposats son:

- *Pla de prova*: un document de planificació de la gestió que mostra la forma en que el test es durà a terme, qui ho farà, què es posarà a prova, quan temps trigarà, quina cobertura ha de tenir (nivell de qualitat que es requereix).
- *Prova de Disseny*: es detallen les condicions de prova i els resultats esperats així com aprovar els criteris del testing.
- *Cas de prova*: especificant les dades del test pel seu ús en la gestió de les condicions d'assaig definides en l'especificació del disseny de test.
- *Procediment del test*: detalla com executar cada un dels assaigs, incloent un conjunt de condicions prèvies i els passos que s'han de seguir.
- *Informe del test de transmissió*: la presentació d'informes sobre els components del software durant la prova han progressat d'una etapa de proves per a la pròxima.
- *Registrar prova*: el resultat de les proves, qui les ha executat, qui les ha observat, dirigit, en quin ordre, l'aprovació, etc.
- *Informe d'incidents de prova*: el detall per a qualsevol prova que no s'obtingui el resultat esperat i altra informació destinada a aclarir per què una prova ha fallat. La raó del control dels incidents és que una discrepància entre els resultats esperats pot passar per una sèrie de raons que no sigui un error en el sistema. Els resultats obtinguts que aparentment poden ser erronis poden ser provocats per un error en el test, una mala interpretació o

un problema en l'execució de les proves. L' informe consta de tots els detalls de l'incident, tant el resultat real com l'espera't i tots els detalls possibles que aportin una evidència per la seva resolució. L' informe també inclourà, si és possible, una avaluació de l' impacte en les proves del incident.

- *Informe de resum de la prova:* un informe de gestió de qualsevol informació important descoberta per les proves realitzades, incloses les avaluacions i l'esforç en la qualitat de les proves i les estadístiques derivades dels informes d'incidents. Aquest document final s'utilitza per indicar si el software testejat es considera acceptable per en funció de si o no ha complert les especificacions definides prèviament pel projecte.

4.2.2 CMMI (*Capability Maturity Model Integration*)

Capability Maturity Model Integration (CMMI) és la evolució del *Capability Maturity Model* (CMM) i permet la millora continua en més disciplines a més del software. Va ser desenvolupat a la Universitat Carnegie Mellon (SEI – Software Engineering Institute) i té com a objectiu millorar i avaluar els processos de desenvolupament i manteniment de productes de software de les organitzacions. Les àrees de procés del CMMI representen un conjunt d'activitats que faciliten el camí a la millora i marquen objectius a complir, les diferents àrees de procés son: Enginyeria, Gestió de Procés, Suport i Gestió de projectes.

En relació al procés de proves l'Enginyeria conté Verificació, Validació, Integració de Producte però no cobreixen totes les necessitats del procés de proves.

La representació del CMMI enfoca a mesurar la millora del procés utilitzant nivells de maduresa. Es basa en cinc nivells de maduresa:

➤ **Inicial (Nivell 1)**

L'èxit depèn dels esforços personals i els processos ben definits.

➤ **Repetible (Nivell 2)**

S'estableixen polítiques i procediments per dur a terme un projecte. Una funció de qualitat assegura que es compleixin aquests procediments.

- **Assegurament de la qualitat (QA):** Els processos de qualitat s'han de formalitzar i s'han de documentar de les proves. Reuneix el conjunt d'activitats planificades i constant que son requerides per assegurar que el software complirà amb els criteris de qualitat esperats. S'ha de planificar abans de desenvolupar el software, no abans ni durant el projecte.

➤ **Definit (Nivell 3)**

S'adopta un procés software i s'adapta a cada projecte. El procés de verificació i validació corresponen a aquest nivell on els processos per cada àrea ja es troben entesos, descrits en estàndard, procediments, eines i mètodes.

- **Verificació:** assegura que els productes interns compleixen amb les especificacions dels requeriments.
- **Validació:** valida de manera incremental els productes en relació a les necessitats de l'usuari.

➤ **Gestionat (Nivell 4)**

La qualitat d'un producte és mesurable i quantificable 'mètriques de software' que serveixen per detectar situacions corregibles.

➤ **Optimitzat (Nivell 5)**

El procés contínuament es millora utilitzant les mesures obtingudes en els processos anteriors.

4.2.3 PSP Personal Software Process

Es un model de millora del procés software format per un conjunt de descripció de processos, de mesures i de mètodes basats en la aplicació de mètodes avançats i tradicionals d'enginyeria i orientat a la millora individual de cada enginyer de software. Desenvolupat al 1997 per un dels creadors del model CMM i sorgí de la idea de que el CMM era aplicable a un únic enginyer i d'aquí va aparèixer PSP. Segons PSP per realitzar una bona feina un tècnic primer ha de conèixer el temps que necessita per realitzar bé la seva feina, segon es necessita una planificació abans de començar un projecte i tercer realitzar-lo d'una forma correcta. Finalment, s'ha d'analitzar els resultats obtinguts de cada activitat i utilitzar-los per millorar els processos, activitats i tasques.

4.3 Models de proves de software

Actualment els models més destacats i relacionats amb la millora en el procés de prova son: TMAP, TMMI i TPI. A continuació es presenten amb les característiques més destacades de cada model. Amb aquest estudi es pretén analitzar les diferents opcions que ens proporcionen per desenvolupar una metodologia pel procés de proves a International Softmachine Systems, s.l.

Existeixen models enfocats a la millora del procés del software. Aquests models sorgeixen degut a que els models existents s'obliden de molts detalls específics de les proves d'una aplicació. Tant les normes ISO 9000 com el model CMM avaluen i qualifiquen els processos de desenvolupament i aporten mecanismes de millora però cap dels dos es centra en el procés de proves del software per qualificar els atributs del mateix, avaluant i verificant si compleix amb els requeriments del client realitzant el software les funcionalitats esperades.

Aspectes de qualitat no contemplats als models CMM i ISO 9000

Al model CMM el concepte de maduresa al procés de proves no es troba contemplat i a més hi ha altres aspectes que no es troben del tot desenvolupats en el model, com ara: la capacitat per provar el software, el criteri de selecció del tipus de proves, la planificació de les proves, etc.

A les normes ISO 9000 en quan a la validació del producte només fa referència a que "el producte ha de ser validat sota condicions similars a les de l'usuari". Les normes fan menció a la documentació de les proves realitzades pel software però no analitza aspectes de la qualitat entorn al proves de proves del software o testing realitzat.

Els dos estàndards proposen mecanismes de millora continua i ajuden a la qualitat dels processos de desenvolupament però es necessitaven alguns estàndards que despenguessin de pràctiques i mètodes per la qualitat de les proves.

Els models enfocats al procés de proves més populars actualment son:

4.3.1 TMap® (*Test Management Approach*)

Model de proves de software que reconeix que un procés de proves estructurat i definit és fonamental per obtenir la qualitat desitjada d'un software i es defineix com un model per la solució a la gestió de proves. TMap no proporciona un marc de referència per establir nivells de maduresa del procés de proves, sinó que és un model que defineix com organitzar, planificar, executar i controlar les proves per obtenir un producte de qualitat, estableix una metodologia que abasteix tots els aspectes relacionats amb el test de software. TMAP és efectiu perquè intenta trobar a temps els defectes importants i té en compte els riscos del producte.

L'objectiu de TMap és reduir els costos i millorar la qualitat del software. Trobar els defectes lo abans possible, per això hi ha una incorporació del procés de proves en els projectes lo abans possible.

Millorant la qualitat s'obté la satisfacció dels usuaris, millor comunicació, automatització de proves adequada, reducció de defectes i visibilitat de la qualitat. Per reduir els costos cal una identificació ràpida dels defectes, menors caigudes a producció, estalviar costos en recursos, control de solapament dels diferents tipus de test i reutilització del Testware. El temps és un factor també implicat en la reducció de costos i millora de la qualitat ja que és possible reduir-lo i ser l'adequat en cada projecte si es reutilitzen els Testware, el desenvolupament és més ràpid, les coses es fan bé des del principi i es fan les coses adequades.

TMap es basa en quatre principis d'un procés de proves estructurat:

➤ **V = Cicle de Vida**

Un Cicle de Vida de les activitats d'acord amb el cicle de vida de desenvolupament del software i descriu les activitats de les diferents etapes. El cicle de vida consta de les següents fases:

- Planificació i control

Fase que defineix un pla de proves (Qui, Què, Quan, On i Com) de les activitats de prova a desenvolupar.

- Estratègia del Test (anàlisi de riscos).
- Preparació pla de testing (estimació de l'esforç).
- Gestió i Control.

- Preparació:

Determina si les especificacions del sistema son de suficient qualitat com per aconseguir l'èxit en l'execució de les proves.

- Avalua si la documentació admet testing (testability).
- Defineix infraestructura de l'entorn de proves.

- Especificació:

Crea els plans de proves, dissenya els casos de prova i defineix les dades de prova.

- Execució:

Execució de les proves manuals o automatitzades (iteratiu) i es realitza un anàlisi de les diferències entre els resultats previstos i els obtinguts i es realitza un informe dels defectes trobats si sorgissin.

- Finalització:

S'actualitzen i conserven els materials de les proves per una futura reutilització, es crea un informe final i s'avalua el procés de testing.

➤ **O = Organització**

Correcte incorporació del equip de testing a la organització i cadascun dels components del equip de testing ha de tenir les seves tasques i responsabilitats.

➤ **I = Infraestructura**

Infraestructura i eines adequades faran possible uns resultats òptims. L'entorn de proves ha de ser estable, representatiu i ha d'estar controlat, així com les eines del procés de testing son fonamentals.

➤ **T = Tècniques**

Les tècniques responen a com realitzar el procés de proves adequat:

- Estratègia de proves.
- Anàlisi dels punts de test.
- Disseny i especificacions del test.
- Anàlisi i report.

Son tècniques útils per la definició de l'estratègia basada en el risc, per donar suport en el procés de planificació, per estudiar i revisar la documentació i per especificar casos i aportar informes. Aquestes tècniques permeten un funcionament estructurat i repetitiu.



4.3.2 TMM (Test Maturity Model)

Model centrat en el procés de proves que neix complementar les deficiències del model CMM. Es descriu un Model de Maduresa específic en el procés de proves, no inclòs en el model CMM, en el qual la organització pot qualificar el procés d'avaluació i proves de software i organitzar les activitats de test amb un sistema de millorament continu.

Proposa 5 nivells de maduresa que determinen els diferents estats d'una organització en quan a la maduresa o estandardització del procés de proves. Respecte al procés de proves cada nivell representa una etapa d'evolució cap a la maduresa. Cada nivell consta de les seves activitats, tasques i responsabilitats que s'han d'aconseguir. A més s'estableix un model de valoració respecte als objectius de cada nivell beneficiós pels testers i personal involucrat en el procés.

També **TMMI** (Test Maturity Model Integration) com evolució de TMM és un model de referència dirigit al procés de prova. Defineix també cinc nivells de maduresa amb les seves àrees de procés corresponents, així com els objectius específics i genèrics i pràctiques específiques i genèriques de cada àrea de procés.

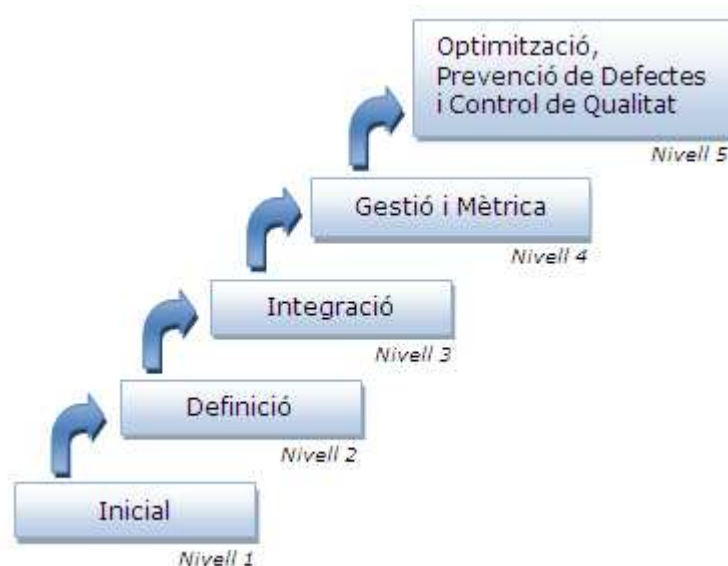
Maduresa del procés de proves del software

Un procés madur inclou la planificació, control, organització del projecte comú a tota l'empresa i acceptat per tot el personal per poder tenir una millora continua. Per aconseguir un procés madur s'han de complir uns objectius:

- Existeix un procés de test definit i l'organització ha de respectar les polítiques de test definides.
- Existeix un equip de test independent al equip de desenvolupament.
- Establir millores continues per garantir l'assegurament de la qualitat.
- Existeixen mètriques de les proves i procés de proves.
- L'equip conta amb tècniques, mètodes i eines adequades pel procés de test.
- Planificació i Control del procés de proves.
- Control de qualitat del producte per verificar defectes i l'anàlisi de resolució i mètriques establertes.

Nivells de maduresa de TMM

Cada nivell implica la maduresa del test al complir els objectius que estableix cada nivell. Aquests objectius es verifiquen amb els models d'avaluació (qüestionaris, valoracions, etc). Tots els implicats en el projecte managers, desenvolupadors, testers, etc., han d'aconseguir aquests objectius exposades mitjançant activitats, tasques i responsabilitats assignades a priori.



➤ **Inicial (Nivell 1)**

El procés de prova no es troba ben definit, es barreja el testing i el debug. No hi ha objectius de maduresa per aquest nivell, les proves es desenvolupen ad-hoc, és a dir, un cop el software ja ha estat implementat i l'objectiu demostra que el software funciona i no hi ha seguretat de que s'entrega un software de qualitat.

➤ **Definició (Nivell 2)**

Es distingeix la fase de proves al cicle de vida del software. S'estableixen els primers objectius per la fase de proves i depuració, aquestes dues fases es troben separades i l'objectiu principal de la prova en aquest nivell es mostrar que el software compleix amb les especificacions i es demostra amb l'execució de tècniques, mètodes, les eines corresponents per les proves i la documentació dels mètodes bàsics de proves.

A aquest nivell hi ha problemes de qualitat ja que el test s'inicia tard en el cicle i provoca una detecció de defectes en fases més avançades.

Alguns aspectes a destacar d'aquest nivell:

- La separació de la fase de proves i depuració és essencial per la maduresa del procés de proves ja que son diferents en quan a objectius, tècniques i mètodes, les proves son planificades i controlades en canvi no ho és la depuració.
- El procés de proves del software es troba inicialment planificat. La planificació ha d'incloure els objectius, anàlisi de risc, disseny, casos de prova, l'assignació de recursos i responsabilitats al personal implicat.
- L'organització ha de disposar de mètodes de test i especificar quines eines es troben disponibles i adequades per utilitzar en cada mètode. Resulta necessari tenir un equip per estudiar, avaluar i recomanar tècniques i mètodes per utilitzar i seguir la política proposada per l'ús dels mètodes i eines pels diferents tipus de projectes.

➤ **Integració (Nivell 3)**

La prova és part del cicle de vida del software, es comença a visualitzar la qualitat del software. La planificació de la prova comença a la fase de requeriments i continua a través del cicle de vida, seguint una variació del model V. Hi ha tècniques establertes per cada fase del test i orientades a cada tasca programada. Els objectius de la prova son establerts respecte als requeriments basats en les necessitats del client i son utilitzats pel disseny de les proves i els criteris d'èxit. Els objectius de maduresa son:

- S'estableix una estructura organitzada pel testing. Hi ha un grup responsable pel test, amb els rols i responsabilitats definides i organitzades. Les tasques corresponents resideixen en la planificació de les proves, l'execució, els estàndards, tècniques, mètodes,

mètriques, bases de prova, reutilització, seguiment i avaluació de les proves.

- Es posen en pràctica programes d'entrenament per millorar les habilitats de l'equip de testers. Han d'estar capacitats de forma que puguin realitzar el seu treball de forma eficient i efectiva.
- Les proves s'integren en el cicle de vida del software basat en el Model V. El mecanisme proposat permet als desenvolupadors i testers treballar organitzadament.
- El procés de test es troba sota control i es defineixen mètriques respecte el procés de test. La documentació també aporta plans de contingència en cas de desviacions dels objectius principals.

➤ **Gestió i Mètriques (Nivell 4)**

El procés de prova es mesurat i quantificat. Les revisions en totes les fases del procés de desenvolupament son reconegudes per l'organització com activitats de prova i control de qualitat. Els objectius de maduresa son:

- S'estableix un programa de revisions comú per tota l'organització i es realitzen per identificar, catalogar i eliminar defectes, provant els productes de forma efectiva i lo més aviat possible.
- S'inclouen mètriques per avaluar la qualitat de les proves, l'efectivitat del procés de prova i la productivitat del personal de prova. S'inclouen les mesures relacionades amb el progrés de les proves, els costos, les dades sobre els errors i els defectes.
- Es creen mecanismes que permeten estructurar i avaluar la qualitat del software abans i després de les proves realitzades o d'una versió a una altra per assegurar la qualitat del software segons els objectius establerts. Per fer-ho possible s'ha de definir polítiques, objectius i atributs de la qualitat del producte software per part de l'equip de test i demés implicats en el projecte.

➤ **Optimització, Prevenció de Defectes i Control de Qualitat (Nivell 5)**

Recull la gestió i mètriques de la fase anterior (Nivell 4) per obtenir una millora continua de la prova i una prevenció de defectes obtenint un control sobre la qualitat. Aquesta millora es possible amb els resultats obtinguts amb les eines de recol·lecció i anàlisi de dades relacionades amb el test. Els objectius de maduresa son:

- Prevenir els defectes: les organitzacions han d'aprendre de la seva història. Els defectes es registren, s'analitzen i amb s'identifiquen les causes dels errors trobant l'arrel de les fallides. D'aquesta manera es poden desenvolupar plans d'acció perquè no tornin a repetir-se els mateixos errors.

- S'utilitzen tècniques i eines per reduir els defectes i establir un control de la qualitat. S'estableixen objectius en els productes com fiabilitat, nivell de fallides per unitat de producte, etc.
- Optimització del Procés de Prova que s'ha de millorar contínuament en el projecte. Inclou identificar les pràctiques de prova que requereixen ser millorades, implementar millores, seguir el progrés, avaluació continua de noves eines i tecnologies a adoptar.

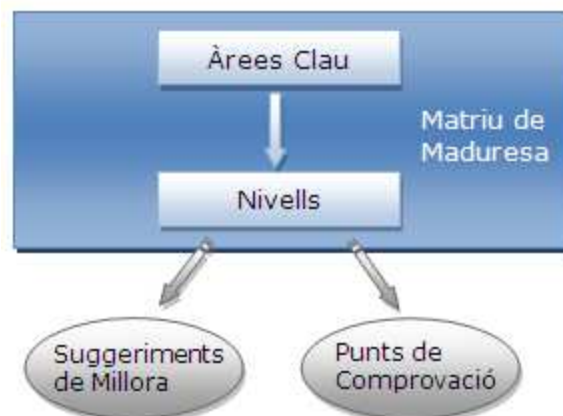
Per aconseguir un Nivell 4 o 5 resulta imprescindible que l'organització disposi d'un equip d'experts en proves independent de l'equip de desenvolupament ja que les característiques d'un Enginyer de Software de Proves son diferents a les d'un Enginyer de Software de desenvolupament, hi ha proves que les realitza el propi desenvolupador (com les proves de Caixa Blanca) i altres que les executa el tester (com les proves de Caixa Negra).

4.3.3 TPI® (*Test Process Improvement*)

Es un model de referència dirigit al procés de proves que proporciona i ajuda a identificar factors a millorar a la fase del test. A vegades la quantitat de documentació, processos, documentació, etc., pot provocar desordre i ofuscació a l'àrea del test. TPI proposa un conjunt de 20 àrees relacionades amb el test (estratègies, tècniques, entorn, etc.) i dintre de les àrees punts de verificació (checkpoints) que permeten avaluar el nivell de maduresa dintre de l'organització).

També es defineix una Matriu de Maduresa de Proves (MMP) com eina per visualitzar i interpretar les relacions existents entre les diferents àrees i els nivells de maduresa amb la situació present a estudiar.

TPI ofereix punts de comprovació i suggeriments de millora amb l'objectiu d'identificar els punts convenients a millorar, es poden determinar amb els nivells superiors que es desitgen aconseguir i a més el model proporciona consells o idees que ajuden a establir un nivell de maduresa de proves superior. Val a dir que TPI és una eina de millora en el procés de test que ajuda a definir i implementar millores i no realitza automàticament a la millora de cap procés.



Àrees Clau

Permeten millorar i estructurar el procés de prova amb la finalitat d'aconseguir un procés ben definit. Existeixen 20 àrees clau.

A continuació es descriuen les àrees clau exposades i definides per TPI:

Àrees Clau	Descripció
Estratègia de Test	Estableix una estratègia per localitzar de manera ràpida i menor cost possible els defectes de qualitat del software. Son importants les mides de detecció i la coordinació per evitar duplicitats o omissions de proves entre els testers. Defineix l'abastament d'aquestes proves, així com els requeriments i riscos de qualitat que suporten les proves.
Model de Cicle de Vida	Les fases del procés poden ser: Planificació, Preparació, Disseny, Execució i Finalització. A cada fase es realitzen les diferents activitats i per cada activitat s'han de definir: objectiu, entrada, procés, sortida, dependències, tècniques i eines, instal·lacions i documentació. És important la definició d'un cicle de vida i així poder planificar i controlar les activitats des del principi.
Moment de participació	El procés de proves ha de començar quan abans millor en el cicle de vida del software, això ajuda a trobar defectes abans i amb més facilitat i prevenir errors que com més avançat es troba el software més crítics resulten.
Estimació i Planejament	S'indiquen les activitats que es realitzaran en cada moment i els recursos necessaris. Així es coordinen les proves i les diferents activitats al llarg del projecte.
Tècniques de disseny de Proves	L'aplicació de diverses tècniques de disseny permet visualitzar la qualitat així com la profunditat i cobertura de les proves. D'aquesta manera s'incrementa la possibilitat de reutilitzar les proves.
Tècniques estàtiques de Test	Son tècniques que s'apliquen a la validació del software o avaluació de mètriques de qualitat sense la necessitat d'executar els programes.
Mètriques	Les mesures sobre el procés de prova i la qualitat del software son importants per avaluar les conseqüències de certes accions de millora, per comparar diferents processos i per poder expressar una opinió. Aquestes mesures es poden obtenir comparant les dades abans i després de realitzar les accions.
Eines de Prova	Un procés de proves amb eines adequades consumeix un menor temps, proporciona una major profunditat a les proves, incrementa la flexibilitat de les proves i s'obté major motivació pels testers. L'automatització de les proves es pot realitzar de varies formes depenent dels projectes i les organitzacions.
Entorn de Proves	L'entorn ha de ser establert perquè els resultats de les proves siguin òptims, alguns conceptes poden ser el hardware, software, instal·lacions, eines per creació de dades, etc. A l'entorn s'han de definir rols, responsabilitats, control, temps, disponibilitat i flexibilitat dels entorns reals de producció.

Entorn d'oficina	L'organització de la oficina influeix en la motivació del personal i l'eficiència del treball.
Compromís i Motivació	Son prerequisits importants per executar el procés de proves amb èxit. El procés de prova no només implica als testers, és imprescindible la cooperació i comunicació amb la resta d' implicats en el projecte a més de la disposició de temps, diners i recursos suficients per realització i execució de les proves.
Funcions de Proves i Capacitació	A més de saber executar les proves és necessari tenir altres coneixements i habilitats. El coneixement de l'organització, l'aplicació a provar, la capacitat i diverses tècniques també son importants per un test òptim.
Abast de la Metodologia	La metodologia adoptada ha de ser genèrica per ser aplicable a qualsevol situació però ha de ser suficientment detallada com per no pensar les coses cada vegada. Si la metodologia fos massa genèrica els procediments o funcions s'haurien de generar cada cop provocant una pèrdua de temps i cost.
Comunicació	La comunicació en un procés de prova dintre de l'equip de test i amb el resta de les parts com son desenvolupadors, client, usuaris, líders, etc., és part de l'estratègia de les proves gràcies a l' informació aportada per part dels implicats.
Informes	La informació reportada sobre la qualitat i el procés de desenvolupament del software tant al client com als col·laboradors del projecte.
Gestió de defectes	Els testers han de controlar el cicle de vida dels defectes i obtenir informes de qualitat de la tendència dels defectes trobats.
Administració del Testware (elements de prova)	Els productes de les proves s'han d'administrar per ser mantenibles i reutilitzables des de les fases de disseny o construcció. El procés de proves es pot interrompre amb l'entrega de versions errònies i això es pot evitar amb una adequada administració.
Administració del Procés de Prova	L'administració del procés és de vital importància per la realització de bones proves.
Revisió estructurada	Implica la validació de productes entremetjats com requeriments i disseny i així els defectes es troben a etapes inicials del procés de desenvolupament. Per realitzar la revisió no cal establir un entorn ni execucions d'aplicacions.
Proves de baix nivell (Caixa Blanca)	Inclou proves unitàries i d'integració i és necessari el coneixement de les propietats internes. Son realitzades pels desenvolupadors i s'executen a les etapes inicials abans de les proves de caixa negra.

Nivells

La forma en que les àrees claus son organitzades en el procés és el que determina la maduresa del procés. Existeixen els nivells A, B, C i D i cadascun millora l'anterior en concepte de temps, cost i qualitat. Cada nivell conté uns certs requeriments dels nivells previs, per exemple el nivell B satisfà els requisits de A i de B.

Àrea Clau	Nivell A	Nivell B	Nivell C	Nivell D
Estratègia de Test	Per una sola prova	Combinada amb proves de caixa negra	Combinada amb proves de caixa negra i caixa blanca	Combinada amb totes les proves i activitats de avaluació
Model de Cicle de Vida	Planejament, disseny i execució	Planejament, disseny, execució i finalització		
Moment de participació	Al acabar l'especificació	Al inici de l'especificació	Al iniciar la definició de requeriments	Al inici del projecte
Estimació i Planejament	Estimació i Planejament elemental	Estimació i planejament amb estadístiques		
Tècniques de disseny de Proves	Tècniques informals	Tècniques formals		
Tècniques estàtiques de Test	Proves en base a entrades	Llista de verificació (checklists)		
Mètriques	Envers al producte	Envers al procés	Del sistema	De la organització
Eines de Prova	Planejament i control	Execució i anàlisi de proves	Automatització de proves	
Entorn de Proves	Administrat i controlat	Entorn més convenient	Entorn a sol·licitud	
Entorn d'oficina	Adequat i a temps			
Compromís i Motivació	Assignació de pressupost i temps	Proves integrades en l'organització del projecte	Enginyeria de proves	
Funcions de Proves i Capacitació	Gerent de proves i testers	Suport i control	Assegurament de la qualitat	

Abast de la Metodologia	Específic al projecte	Per tota l'organització	Organització i optimització	
Comunicació	Interna	Del projecte (defectes, canvis, control)	A la organització	
Informes	Defectes	Progrés, activitats, defectes amb prioritats	Riscs i consells amb estadístiques	Consell
Gestió de defectes	Administració interna de defectes	Defectes extensos i Informes flexibles	Defectes del projecte	
Administració del Testware (elements de prova)	Administració i control intern de les proves	Administració i control extern de la base i objectes de les proves	Testware reutilitzable	Capacitat de rastreig: des de requeriments fins casos de prova
Administració del Procés de Prova	Planejar i fer	Planejar, fer, chequejar i reaccionar	Comprovar, reaccionar a l'organització	
Revisió estructurada	Tècniques	Estratègia		
Proves de baix nivell (Caixa Blanca)	Cicle de vida: Planificar, Disseny i Execució	Tècniques de disseny de caixa blanca	Estratègia de proves de caixa blanca	

Checkpoints: els requeriments per cada nivells estan donats punts de Verificació (checkpoints). En el moment en que resulta afirmatiu es pot qualificar en el nivell corresponent.

Un exemple de l'àrea Clau 'Administració de defectes' amb els checkpoints:

Àrea Clau 16 - Gestió de defectes				
Els testadors han de controlar el cicle de vida dels defectes i obtenir informes de qualitat de la tendència dels defectes trobats.				
Nivell	Punt verificació		OK?	Comentaris
A	<i>Gestió interna de defectes</i>			
	1	Les diferents etapes del cicle de vida dels defectes oposats són administrats, inclòs el retesteig		
	2	Els següents elements del defecte són registrats: nombre únic, persona que va registrar el defecte, data, categoria de serietat, descripció del problema, estat de la petició de canvi		
B	<i>Gestió extensa del defecte amb facilitat pels informes</i>			
	1	Es registra informació per al posterior anàlisi de tendències, es registrarà la següent informació: cas de prova, escenari, mòdul, prioritat de l'escenari, versió del software, causa (Probable i Definitiva), tots els estats pels quals ha passat incloent dates, descripció de la solució del problema, versió del testware en el qual el defecte és solucionat, persona encarregada de resoldre el problema		
	2	L'administració de defectes permet l'obtenció d'informes de resultats mitjançant l'ordenació de diferents maneres, amb mecanismes de filtre, etc.		
	3	Existeix algú responsable de l'administració de defectes i és duta amb propietat i consistència.		
	4	Les desviacions de l'estratègia coordinada són notificades, just després que es realitzi un ajustament substancial de l'estratègia combinada basada en els riscos.		
C	<i>Gestió de defectes de projectes</i>			
	1	L'administració de defectes és utilitzada íntegrament en el projecte		
	2	L'administració de la gestió de defectes té mecanismes de seguretat per a controlar que qui estigui designat ha realitzar els canvis és qui els realitza.		

Matriu de Maduresa del Test

Mostra les prioritats i dependències entre els nivells i àrees clau ja que no totes tenen la mateixa importància i a més estableix els passos de millora a seguir.

MMT	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Estratègia de Test		A					B				C		D	
Model de Cicle de Vida		A			B									
Moment de participació			A				B				C		D	
Estimació i Planejament				A							B			
Tècniques de disseny de Proves		A		B										
Tècniques estàtiques de Test					A		B							
Mètriques						A			B			C		D
Eines de Prova					A			B			C			
Entorn de Proves				A				B						C
Entorn d'oficina				A										
Compromís i Motivació		A				B						C		
Funcions de Proves i Capacitació				A			B			C				
Abast de la Metodologia					A						B			C
Comunicació			A		B							C		
Informes		A			B		C					D		
Gestió de defectes		A				B		C						
Administració del Testware (elements de prova)			A			B				C				D
Administració del Procés de Prova		A		B								C		
Revisió estructurada							A			B				
Proves de baix nivell (Caixa Blanca)					A		B		C					

5 Proves de software

El procés per la realització de proves pot resultar més problemàtic del que sembla. Realitzar proves porta molt temps i la planificació d'hores que es pugui establir per realitzar-les mai és suficient.

Les proves transmeten la qualitat del software a testear i moltes vegades per falta de temps poden ser insuficients per obtenir les mètriques reals i adequades.

El procés de proves consisteix en confirmar que el producte compleix amb els requisits definits i l'objectiu de les proves de software es trobar els defectes lo abans possible.

Les proves s'han de plantejar com un procés de detectar errors en una aplicació. Si el tester busca els defectes o errors amb mentalitat de verificar que l'aplicació funciona correctament es deixarà moltes accions per realitzar ja que no és la manera d'interpretar un test.

Es important que el propòsit de l'equip de proves sigui trobar defectes, provant l'aplicació de manera que falli, intentant que es produeixin errors i demostrar que existeixen. Es realitzen proves per demostrar que una aplicació fa el que ha de fer però sempre contemplant que es busquen els defectes.

En aquest procés es defineix:

- El criteri de verificació pel producte software integrat que demostrin que es compleixen els requisits.
- Es verifica el software utilitzant els criteris definits.
- Es registren els resultats de les proves.
- Es defineix l'estratègia de regressió i s'apliquen per tornar a verificar en el cas que hi hagi modificacions.

Cada fase de proves, verificació, validació i qualitat es planifica, s'executa i es corregeix i això comporta disposar de mètodes, tècniques i eines.

Procés de gestió i planificació

La gestió i planificació de les proves estableixen els objectius a definir i permeten avaluar el risc i desviacions al llarg del projecte.

- La documentació és una part important del procés de proves i ha d'incloure:
 - Un pla de proves: metodologia de proves des del inici del projecte.
 - Una especificació del disseny de proves.
 - Especificació dels casos de prova.

- Especificació del procediment de prova.
- Revisions de seguiment al llarg del procés de proves.
- Informe de resultats i incidències o problemes.
- Test d'errors.
- La documentació ha d'estar al dia.
- Els gestors i encarregats han de verificar les despeses de cost i esforç: número de casos de prova especificats, casos de prova executats, casos de prova que han fallat, etc.
- La generació dels casos de prova es basa en el nivell de proves que s'estigui realitzant.
- L'entorn d'execució de les proves ha de ser compatible amb l'entorn de desenvolupament de software.
- Un cop definit els procediments de prova, s'executen amb el personal qualificat. Es realitza l'execució de manera consistent en els casos de prova documentats.
- Un cop executades les proves s'avaluen els resultats.
- Els problemes es gestionen a través d'un procés de resolució de problemes.
 - Problemes en la definició del cas de verificació i validació.
 - Problemes en la implantació del cas.
 - Problemes en les dades d'entrada pel cas.
 - Problemes amb el codi o element software subjecte a verificació i validació.
 - Problemes amb el disseny o origen del software subjecte a verificació i validació.
- Elements de la planificació:
 - Elements que han de ser provats i les seves característiques.
 - Infraestructura.
 - Activitats.
 - Objectius de les proves.
 - Mètodes i tècniques.
 - Estratègia de proves (basat en tècniques i objectius de proves).

- Mida de cobertura o verificació.
- Pla de proves.
- Informes i resultats.

Procés d'anàlisi i disseny

- Determinar els objectius de les proves.
- Transformar els objectius en condicions de proves i casos de prova concrets.

Procés d'implementació i execució

- Detall dels casos de prova.
- Preparació de l'entorn.
- Execució de les proves.

Procés d'avaluació de les proves

- Avaluar els resultats obtinguts de les proves.
- Realitzar els reports dels resultats.
- Recollir la informació de les execucions de les proves.

5.1 Verificació i Validació

En el procés de desenvolupament una aplicació ha de ser comprovada per assegurar que satisfà les seves especificacions i respon a la funcionalitat esperada per les persones que esperen el software.

La verificació i validació es donen al llarg de tot el desenvolupament del software, des de l'estudi dels requisits fins les proves finals del producte.

Son conceptes que no signifiquen el mateix, *Boehm*⁶ al 1979 va definir la diferencia entre els dos conceptes:

- **Verificació** → *Estem construint un producte correctament?*

Assegura que el software compleix amb les especificacions dels requeriments. Es comprova que el producte satisfà els requeriments funcionals i no funcionals.

⁶ [BOEHM] La calidad del software y su medida. ISO/IEC 9126, Modelo de Boehm

Com a mètodes de verificació: anàlisi, proves, demostracions, inspeccions, revisions, etc.

➤ **Validació** → *Estem construint un producte correcte?*

Assegura que el software satisfà les expectatives del client. Més enllà de les comprovacions de que el sistema satisfaci la seva especificació per demostrar que el software fa el que el client espera que faci.

Com a mètodes de validació: requeriments, disseny, prototips.

Per l'anàlisi i comprovació dels sistemes existeixen dos aproximacions complementaries:

- **Inspeccions de software:** analitzen i comproven els requeriments, diagrames de disseny i el codi font del programa. Es poden utilitzar a totes les etapes del procés.
- **Proves de software:** implica executar una implementació de software amb dades de prova. S'examina les sortides del software i el seu entorn per comprovar que el sistema funciona com es requereix.

5.2 Proves

Les proves son els processos que permeten verificar i revelar la qualitat d'un producte. S'integren dins de les fases del cicle del software de l'Enginyeria de software.

Es realitzen per determinar el nivell de qualitat efectuant mesures per comparar el producte software amb els seus objectius inicials i localitzar el màxim número de defectes:

- Es busca demostrar quins objectius no es compleixen.
- Si no existeixen objectius documentats no és possible realitzar proves.
- No significa provar la funcionalitat en el seu conjunt.
- S'utilitza com a base els objectius originals i la documentació del usuari per augmentar-la.
- No existeix un mètode, es proporcionen línies de base a la hora de preparar els casos de prova.

Les proves es dissenyen a partir de les especificacions de requeriments del software i normalment es preparen per un equip independent als desenvolupadors del sistema.

Cas de prova

Conjunt de valors d'entrada i condicions inicials d'execució, resultats esperats i condicions resultants d'execució, desenvolupat amb l'objectiu particular o condició de prova per comprovar un determinat camí d'execució o per unificar el compliment d'un requisit determinat [IEEE 610].

Formalment els casos de prova consisteixen en tres parts:

- *Visió general:* conté informació general sobre els casos de prova.
 - Identificador únic per futures referències.
 - Dissenyador del cas de prova.
 - Versió i nom del cas de prova on s'ha definit.
 - Requeriments que cobreix el cas de prova.
 - Descripció del propòsit i funcionalitat.
 - Dependències.
- *Activitat dels casos de prova:*
 - Configuració de l'entorn on s'executa el cas de prova.
 - Inicialització conté la descripció d'algunes accions que s'hagin de realitzar abans d'executar els casos de prova.
 - Finalització conté la descripció d'algunes accions que s'hagin de realitzar després d'executar els casos de prova.
 - Accions a realitzar per completar la prova.
 - Descripció de les dades d'entrada.
- *Resultats:*
 - *Resultat esperat:* és el comportament que s'especifica en els requeriments del software i amb unes condicions específiques d'execució.
 - *Resultat real:* és el comportament obtingut després de provar les diferents accions del sistema. Normalment hi ha un apartat on s'especifica l'estat correcte o fallida que s'ha obtingut.

Defectes, errors o bugs

Els defectes son característiques que poden causar que un component o sistema no realitzi les funcions requerides.

Un error és una equivocació en el qual el resultat no és correcte.

Un bug es defineix com qualsevol cosa que pot estar malament en el software. Quan es reporten bugs es fa referència a qualsevol defecte o error trobat en el sistema.

5.2.1 Mètodes dinàmics

Les proves són una tècnica dinàmica de verificació i validació, això significa que requereixen de l'execució del producte software.

L'objectiu és detectar els defectes analitzant la resposta del producte davant del conjunt de valors d'entrada.

ISO 12207 'Las fases y procesos de pruebas no están incluidas en los procesos de Verificación y Validación pero sí en este curso de métodos dinámicos para verificar y validar'.

Principis de les proves

- Les proves de software són organitzades, mesurades i ben definides integrades en el cicle de vida del desenvolupament del software. I s'han d'iniciar lo abans possible ja que la correcció de defectes és més difícil com més avançada es trobi el desenvolupament del producte.
- Les proves exhaustives no són possibles: no és possible provar totes les combinacions del comportament que pot portar un sistema per això els casos de prova han d'estar ben definits i avaluar els casos prioritaris i de major risc.
- Hi ha un grup de proves que s'ha de realitzar al llarg de tot el projecte, aquestes han d'estar definides i executar al inici de cada fase.
- S'ha de definir i adequar els objectius de les proves depenent del context. Pot ser que dos sistemes siguin semblants però que en context resultin diferents.

5.2.2 Requeriments del software

Els requeriments contenen l'anàlisi de les funcionalitats i observacions del sistema que ajuden a entendre l'aplicació a alliberar a l'usuari. Especifiquen que ha de fer i quines característiques ha de tenir el software.

Els requeriments d'un software s'han de poder comprovar per verificar si el comportament conté defectes o errors.

Els requeriments funcionals descriuen les funcions que el software ha de realitzar i els no funcionals són aquells que demostren les restriccions del software.

Amb els requeriments s'extrau el document d'especificacions dels requeriments del software. Aquest és necessari per ser revisat i analitzat per l'equip de proves i és imprescindible la presència del document per iniciar el procés de proves.

➤ **Especificació dels requeriments**

Es recull la informació de l'anàlisi de l'aplicació i detalla les funcionalitats i declaracions exposades.

Els requeriments son:

- Requeriments del sistema: son les funcionalitats a dissenyar, implementar i provar.
- Requeriments d'usuari: recull les declaracions del client i usuari final del sistema.

➤ **Validació de requeriments**

S'analitza el document per assegurar i verifiquen que el document defineix el software que els usuaris esperen. Durant aquesta fase es poden descobrir errors en el document de requeriments i s'ha de modificar per corregir els problemes detectats.

Es interessant que abans de iniciar la codificació i implementació del desenvolupament del software es validi el document de requeriments per no iniciar el procés i haver de tornar enrere per localitzar errors en els requeriments.

5.2.3 Objectiu del procés de proves

El procés de proves es centra en establir que el sistema satisfà els requisits funcionals i no funcionals i que no es comporta de forma inesperada.

DIJKSTRA 1972 'Las pruebas sólo pueden demostrar la presencia de errores, no su ausencia'.

Hi ha dos objectius en el procés de proves:

➤ **Proves de validació:** *demostrar que el sistema compleix amb els requisits.*

Les proves que es realitzen han d'abastar tots els requeriments i per totes les característiques del sistema.

En aquestes proves s'executen els casos de prova que demostren l'ús esperat del l'aplicació a provar i demostren que el sistema funciona correctament.

➤ **Proves de defectes:** *descobrir els defectes en el comportament del sistema que sigui no especificat i no requerit.*

Les proves comproven que el comportament és el desitjat i es comprova que no es donen comportaments no desitjats, càlculs incorrectes o corrupció de dades.

Es proven els defectes i els casos de prova executats son aquells que mostren els defectes i demostren que el sistema no funciona correctament.

Un model de procés de proves amb casos de prova, dades de prova i els resultats de la prova:



Els casos de prova són el conjunt de valors per verificar que es compleixen els requeriments específics, les dades de prova són les dades que existeixen abans de que una prova sigui executada i que afecta al sistema a provar (dades en una base de dades) i els resultats de la prova s'obtenen després de l'execució de les proves aportant el resultat real.

Les diferents responsabilitats en el procés de proves depèn de l'estructura de l'equip encarregat de realitzar el procés de proves. El que és important és que per organitzacions amb aplicacions complexes l'equip de Test és independent a l'equip de desenvolupadors i és per això que les proves s'han de detallar i han de ser informades amb els resultats obtinguts.

5.2.4 Tècniques i eines de les proves

Las proves es basen en un conjunt de casos de prova i és interessant que existeixin polítiques per escollir el conjunt de proves a executar segons el sistema a provar.

Hi ha característiques que sempre han de ser provades:

- S'haurien de provar totes les funcions a les que s'accedeix a través d'un menú.
- S'haurien de provar totes les combinacions de les funcions.
- S'haurien de provar les funcions amb dades correctes i incorrectes.

Les eines de proves garanteixen el seguiment de les activitats en el procés de proves i es poden classificar depenent de les accions a realitzar:

- Eines de gestió.
- Eines d'automatització.
- Eines de generació de casos de prova: especificació dels casos de prova i generació de dades.
- Eines d'execució i registre: comparadors, mecanismes d'execució, etc.
- Eines de seguiment.

Entre les tècniques de proves destaquen:

➤ **Tècniques de caixa blanca**

Son proves en les que s'utilitza l'estructura de control del disseny procedimental per derivar els casos de prova, normalment son proves executades pels desenvolupadors del codi de les aplicacions.

- Exerciten els camins independents de cada mòdul.
- Exerciten les decisions lògiques.
- Executen els bucles en els seus límits i amb els seus límits operacionals.
- Exerciten les estructures internes de dades per assegurar la validesa.

➤ **Tècniques de caixa negra** (*Proves funcionals*)

Aquestes proves es descriuen en més profunditat que les altres ja que en el present projecte s'analitza la metodologia exposada per l'equip de QA que s'encarrega d'aquestes proves.

Son proves també anomenades 'proves funcionals' en el que l'equip de proves valida el sistema per comprovar que compleix els requisits predefinits i que es tracta d'un sistema estable, confortable i de confiança amb l'objectiu de demostrar si el sistema funciona o no correctament. Els defectes es comuniquen a l'equip de desenvolupament per arreglar els errors detectats.

Les proves de caixa negra estudia les entrades i sortides en el sistema, d'aquí el nom de 'caixa negra' ja que el seu comportament només pot ser determinat observant les entrades i sortides.



Podem distingir les següents tècniques:

▪ **Partició equivalent**

Les dades d'entrada i de sortida es poden agrupar per característiques comuns i s'identifiquen les particions del sistema a testejar les quals poden ser utilitzades per dissenyar els casos de prova.

Les dades d'entrada han de ser vàlides i invàlides i en els dos casos s'obtenen uns resultats de sortida.

Aquesta tècnica selecciona subconjunts per trobar la majoria d'errors i els casos de prova s'han de generar amb les següents característiques:

- Cada cas de prova ha de tenir tantes entrades com sigui possible per reduir el número de casos de prova necessaris.
- Per cada variable analitzar el subconjunt de valors que han de tenir el mateix comportament. Amb un cas de prova que conté les mateixes característiques que un altre i es detecta un error, s'espera que l'error es produeixi també als demés casos de prova.
- Simplificar en lo possible.

Aquestes proves intenten trobar errors en les següents categories:

- Funcions incorrectes o absents.
- Errors d'interfase.
- Errors de rendiment.
- Errors d'inicialització o finalització.

- *Proves de valor límit*

Analitza els errors que sorgeixen en els límits de les proves. S'ha de provar els valors que es troben als límits de les funcions a testejar.

Es un bon complement de les Particions d'equivalència.

- *Proves d'estrés*

Avaluen el software sobrepassant els límits de les especificacions. Els més típics son els de mesurar la màxima capacitat que pugui suportar el producte.

- Màxim número d'activitats que es poden suportar simultàniament.
- Màxima qualitat de dades a ser processades en un temps determinat.

- *Taules de decisió*

Representen relacions lògiques de condicions d'entrada i de sortida.

Es presenten les dades en forma matricial amb les condicions, regles i accions associades.

Els casos de prova es consideren exposant el màxim de combinacions possibles per provar els diferents requeriments del sistema en tots els seus estats possibles.

- *Proves de cas d'ús*

L'actor interactua amb el sistema. Es prova el sistema en situació normal que duen a l'èxit.

Hi ha condicions inicials que s'han de complir i condicions finals a les que arribar a observar. Les proves es realitzen en escenaris reals un cop finalitzada l'aplicació. A vegades pot donar-se que l'usuari ja disposi de l'aplicació ja i es troben defectes no contemplats sobre l'ús real del sistema.

- *Proves d'injecció d'errors*

Avaluen el comportament del producte software injectant errors en el hardware operacional.

L'objectiu és provar en els seus mecanismes de tolerància d'errors.

- ✓ ***Proves no funcionals***

Identifiquen característiques principals del software que no es basen en les especificacions del software o en fa referència mínimament:

- *Facilitat d'ús*

Es la capacitat del software per ser utilitzat i de fàcil enteniment sota les condicions especificades [ISO 9126].

Depenent del tipus i perfil d'usuari s'estudia les accions del software: accés a l'aplicació, menús, drets, etc.

Les aplicacions han de ser visualment intuïtives i de fàcil enteniment afegint els passos a seguir, pantalles ben definides i veient la informació adequadament, etc.

- *Robustesa*

Grau en el que un component o sistema pot funcionar correctament en presència d'entrades invàlides o condicions d'estrès [IEEE 610].

El sistema ha de demostrar el millor comportament sense bloquejos, gestionant i controlant els diversos funcionaments resultant un sistema segur i fiable.

- ✓ *Tècniques basades en l'experiència*

Aquestes proves depenen de les persones que s'encarreguin de provar el sistema.

Son proves basades en l'experiència i que es basen en trobar els defectes lo abans possible.

5.2.5 Mètriques de les proves

Documentar, analitzar i mesurar les proves és important alhora de decidir si un procés de proves és de confiança i estable.

Mesurant les proves s'avaluen les accions correctives i l'impacte que han provocat les diferents accions realitzades.

Les proves han de mesurar objectius esperats, s'ha d'aplicar i planificar en els diferents processos i s'han d'avaluar en cada moment per conèixer l'estat en el que es troba el projecte de proves.

El compliment del pla de proves demostra els casos de prova no executats, els planificats i modificats i els percentatges de proves realitzades.

Al mesurar les proves s'obté la quantificació del procés utilitzat, la possibilitat d'avaluar i implantar millores en els processos i aconseguir la millora continua podent demostrar la qualitat de les proves.

En el procés de prova les mesures es poden utilitzar per:

- ✓ Visualitzar els resultats de les proves comprovant i verificant la prèvia planificació de les mateixes.
- ✓ Es determinen les accions correctives depenent dels resultats obtinguts.
- ✓ Reportant les proves en les diferents fases del procés de proves es pot analitzar l'estratègia i l'efectivitat de les proves per veure si és l'adequada i si el camí seguit és el més adequat o cal realitzar algunes accions al respecte.

6 Metodologia d'Assegurament de la Qualitat

En aquest apartat es presenten les principals característiques de la metodologia implantada en el procés de proves d'una aplicació i que servirà per posteriors aplicacions a provar.

MAQU ha sigut desenvolupada per proporcionar un marc de treball necessari per la millora del procés de proves que es desenvolupa de forma paral·lela a altres processos desenvolupats a l'empresa. Per obtenir MAQU s'ha analitzat i estudiat els models de processos de proves (TMAP, TPI, TMMI, etc...) i eines proposades en els punts anteriors fins adaptar una metodologia a l'àrea de proves de Softmachine. Un cop obtingut l'anàlisi dels resultats, MAQU s'ha definit per ser acceptat i establert a l'àrea QA i encara que és independent al procés seguit en el desenvolupament del producte, s'ha adaptat per ser compatible amb la els demés processos implantats al departament.

International Softmachine System s.l. parteix els seus projectes d'una metodologia de gestió de projectes PMI estable i consistent, la qual té definida uns processos de lideratge MAQU respecta i fa possible la comunicació amb la resta de mètodes implantats a l'organització.

TMMI i TMAP han sigut els models referencials per MAQU. TMAP ofereix una posada en marxa en l'exploració del software, la descripció d'un procés de proves efectiu i eficient, permet que l'organització compleixi amb els objectius claus del negoci i és un mètode eficient perquè és aplicable universalment centrat en la reutilització. Per una altre banda, TMMI avalua el nivell en el que es troba l'organització, prioritza l'identificació de la millora continua i estableix un marc d'equip de proves estable, constant i sistemàtic en les tasques. El model TPI, model de millora de proves com el TMMI, s'ha valorat per poder implantar alguns dels seus mètodes en un futur, però el TMMI és el que s'adequa millor a les necessitats que requereix el departament QA actualment.

Per establir una metodologia la feina més difícil plantejada és recopilar tota la informació històrica de les aplicacions, establir estàndards de documentació acceptats i adaptats a la resta del departament i generar bons plans de proves adaptats als antics per dur a terme una millora continua.

S'ha contemplat els objectius que es vol aconseguir amb MAQU però per començar s'estudien els punts negatius que cal evitar i que en moltes empreses dificulten un bon desenvolupament del Testing:

- L'equip de test o QA no redacta al 100% el joc de proves de les aplicacions i moltes vegades es limita a provar les funcionalitats de forma superficial sense provar totes les casuístiques.
- La gestió del pla de proves es troba en un document Word que s'ha de redactar cada vegada que s'ha de realitzar un test.
- Els errors detectats en el test no queden reflectits en els jocs de proves i algunes proves no consten en el pla de proves.

- No hi ha una manera exclusiva de redacció o traspàs d'errors de l'aplicació en procés de testing provocant problemes de comprensió entre els departaments de desenvolupament i test.
- El seguiment de les proves, problemes amb instal·lacions, retorns d'errors no solucionats a les versions, estadístiques d'errors, etc. No es troben correctament documentats i es tornen irrecuperables amb el pas del temps.

MAQU analitza i detalla com analitzar tots aquests punts clarificant les etapes a seguir fins a l'alliberament de l'aplicació.

El seguiment realitzat per establir MAQU ha sigut el següent:

1. Establir els objectius

Es vol millorar el manteniment dels productes, evitant les correccions contínues dels productes i millorar la eficiència de les proves dels productes abans d'entregar a l'usuari final amb la qualitat desitjada.

Els models exposats anteriorment no s'ajusten perfectament a les necessitats de les aplicacions de Softmachine, però s'utilitzen per extreure informació i idees dels diferents models. Son aplicacions especialitzades en uns productes i s'ha d'enfocar el desenvolupament del procés de proves a l'empresa, per això es decideix establir una metodologia a mida.

2. Estudi de la situació actual

S'ha estudiat i identificat les parts bones i dolentes dels mètodes seguits fins ara a l'àrea QA i així aconseguir la metodologia més adequada sempre millorable.

S'ha recollit idees i informació del personal de l'empresa i s'ha analitzat la manera de treballar dels equips de l'empresa. Per contemplar una metodologia acord amb els equips de la resta de l'organització.

3. Adaptació en els processos

Es processos han de ser adaptables al departament de desenvolupament i han de ser definits: identificar els processos a millorar, definir la proposta dels nous processos, dissenyar formularis pels processos, elaborar documentació i mètodes pel procés de proves.

Per verificar els processos s'han aplicat al nou mòdul a implementar amb el qual es presenta el projecte.

4. Ús i millora del procés de proves

El procés se seguirà utilitzant en posteriors projecte o manteniment dels ja instal·lats en usuaris finals. Quan es finalitza un projecte es realitza un anàlisi dels resultats verificant el seguiment dels processos de prova i si les proves elaborades han sigut suficients i eficients.

Les reunions periòdiques fan possible detectar problemes a temps en els diferents processos i veure el que s'ha de resoldre i l'impacte que es detecta.

L'objectiu de la millora del procés de prova és aconseguir refinar el procés inicial per poder adaptar-lo i flexibilitzar-lo als diferents projectes que es presentin.

6.1 Objectius en el procés de proves

L'objectiu és obtenir un software de qualitat i està comprovat que la qualitat ve determinada per la qualitat del procés de proves que s'utilitzi. Molts dels problemes que actualment existeixen es solucionen definint un procés de proves, gestionat i controlat, que comença a la fase de definició de requisits i és dur a terme de forma paral·lela amb el desenvolupament del software. En molts projectes no s'ha obtingut els resultats esperats per dolentes planificacions i per tenir un procés de proves mal definit. Amb aquest projecte es vol definir un bon procés de proves acord amb les aplicacions desenvolupades a Softmachine i amb les implicacions que això comporta.

El procés de prova definit és independent al procés seguit en el desenvolupament del producte i té com objectiu provar les funcionalitats del producte a partir de les seves especificacions. Es basa en l'anàlisi del risc del producte per definir les funcionalitats a provar (planificació del projecte) i la prioritat amb la que son provades (definició del pla de proves), s'organitza el projecte de prova en cicles que serveixen de guia pel procés de prova i que descriuen el cicle de vida del producte.

MAQU necessita un procés ben definit per:

1. Identificar i establir els objectius, polítiques i estratègies a implantar.
2. Desenvolupar i mantenir un pla de proves que reculli els aspectes necessaris per la gestió i control de les activitats de prova.
3. Definir les especificacions necessaris per obtenir bons i eficients casos de prova.
4. Analitzar i informar dels resultats obtinguts durant l'execució de les proves.
5. Identificar i establir les mides adequades per solucionar els possibles problemes trobats.
6. Per últim, es contempla i s'adapta el procés per ser millorable.

6.2 Relació entre el procés de proves i desenvolupament

El procés de desenvolupament té unes etapes definides en el cicle de vida del software i el procés de proves amb MAQU és independent però compatible al procés seguit pels desenvolupadors del software.

Es important definir la relació entre els diversos processos per establir l'abastament de cadascuna de les etapes de les proves. Indiscutiblement per provar una aplicació primer ha de ser programada i l'enllaç entre els desenvolupadors i QA ha de ser estable per realitzar les tasques en els temps planificats.

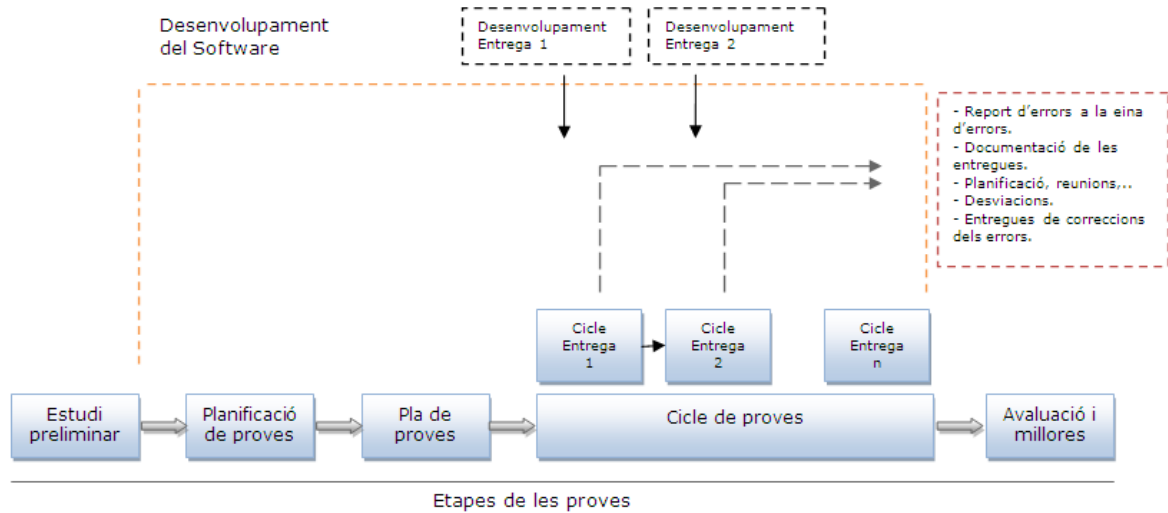
Les etapes en les que es troba el producte és diferent a l'àrea de desenvolupament que a l'àrea QA, normalment el desenvolupament del producte ha d'estar en etapes més avançades que el test que s'estigui realitzant, però sempre coordinat perquè si el codi del software es troba molt avançat i es detecten grans defectes o errors greus pot provocar haver de retrocedir el producte masses fases amb les conseqüències més greus com desplaçaments de dates d'entrega, noves planificacions de temps, aturades a altres tasques, reorganització de les fases, etc...

Els grups de treball no es poden sotmetre a problemes de comunicació, sincronització de les tasques, dolentes planificacions i administració dels projectes, etc... provocant errors catastròfics pel procés de desenvolupament del software. La interacció i coordinació entre els diferents equips és vital per obtenir bons resultats i en el temps planificat i desitjat per alliberar un producte de qualitat.

MAQU contempla les etapes, documentació, planificació i organització durant tot el cicle de vida del software en l'àrea QA. La relació entre el procés de desenvolupament i el procés de proves en alguns punts pot ser paral·lela i en altres depèn un procés de l'altre. A continuació es realitza una breu explicació de les etapes implementades:

L'Estudi Preliminar es realitza abans de començar el desenvolupament del producte, la planificació, definició del pla de proves i disseny de les proves es realitza paral·lelament a les tasques de desenvolupament del producte que ja s'estiguin realitzant (son independents les proves de test i el desenvolupament de l'aplicació), a continuació es desenvolupen les tasques del cicle de proves, configuració de l'entorn i execució de les mateixes (aquesta és l'etapa que pot provocar més desviacions en un producte, funcionalitats inesperades, errors no contemplats, entregues amb retràs, ...) i finalment s'avalua una conclusió del seguiment del projecte i un cop validat s'entrega l'aplicació a l'usuari final.

La coordinació entre les diferents àrees s'organitza per entregues dins d'una mateixa versió. És ideal que les diferents entregues es realitzin en els temps estipulats i que no hi hagi un avanç o retrocés en els diferents equips.



Aquest esquema genèric és aplicable a qualsevol projecte de proves i els processos s'adapten a cada projecte amb les característiques particulars de cadascun.

Hi ha punts a tenir en compte a l'hora de planificar el projecte de proves en relació al desenvolupament del mateix:

➤ **Document de requeriments**

L'equip QA ha de disposar del document de requeriments des de l'inici per començar amb l'Estudi preliminar, així es podrà detectar de manera ràpida qualsevol incident i ràpidament reaccionar per no arrossegar errors des del principi. El document de requeriments és alliberat a QA pel responsable de desenvolupament, acceptant possibles millores per part de QA durant les primeres etapes de proves.

➤ **Reunions de seguiment**

Durant la planificació de proves i el pla de proves es realitzen reunions per possibles dubtes funcionals, escollir les millors estratègies de test i obtenir un pla de proves efectiu per començar el Cicle de proves amb la primera entrega del producte a provar.

➤ **Entregues estables**

La primera entrega realitzada a l'àrea QA per l'equip de desenvolupament ha de ser estable amb les funcionalitats pactades de primera entrega, en cas de tenir moltes fallides provoca una desviació segura per la resta del projecte. En aquest cas, es prendrien mesures i alternatives per provar parts més petites però estables perquè sinó el cicle de proves cada cop és més extens i moltes proves han de repetir-se en següents cicles.

➤ **Canvis a la planificació**

Poden haver canvis en el pla de desenvolupament i això implica canvis en la planificació del cicle de proves. Tot desplaçament o modificació al desenvolupament afecta a les fases del Test, per això és necessari un seguiment estricte i documentat per les possibles regressions suportades.

➤ **Documentació compartida i per versions**

La documentació de les diferents projectes es troba localitzable en un espai d'emmagatzematge compartit, per versions i entregues disponible per les persones implicades en el projecte. Cadascú manté la informació que les seves tasques prèviament planificades requereixen.

6.3 Rols de QA

Els rols que s'han definit per l'àrea de QA son els següents:

➤ **Responsable del projecte de proves:**

Dirigeix i planifica el projecte. Te capacitat per liderar un projecte, coneixements en l'àrea de l'administració, experiència en treball en equip i és tècnic de proves. Activitats a desenvolupar:

ACTIVITATS RESPONSABLE DE PROVES	
AR1. Abastament i fases	Defineix l'abastament del projecte de prova i de la finalització de les fases junt amb els responsables d'àrea i desenvolupament.
AR2. Anàlisi preliminar	Realitza l'estudi preliminar de l'aplicació a desenvolupar per definir i planificar el projecte.
AR3. Anàlisi de riscos	Analitza el risc del software (també a l'estudi preliminar) detectant les funcionalitats més complexes i amb major probabilitat de defectes.
AR4. Planificació del projecte	Planifica el projecte de proves definint l'estratègia de proves i les tècniques a utilitzar.
AR5. Anàlisi de requeriments	Analitza el document de requeriments per generar el pla de proves més adequat.
AR6. Planificació del cicle de proves	Defineix i planifica el cicle de proves, les funcions a provar en cada cicle i les activitats a realitzar per l'equip QA.
AR7. Seguiment de fases	Realitza el seguiment en el procés de proves, assegura la qualitat dels productes en prova i valida el seguiment de les fases durant el projecte.
AR8. Avaluació de proves	Avalua les proves des de la planificació fins la finalització de les fases del cicle de prova i decideix si s'ha de definir noves o diferents proves en tot moment.

AR9. Avaluació de la qualitat	Avalua la qualitat en cadascuna de les fases i en l'alliberament del software realitzant un informe d'avaluació informatiu sobre els resultats obtinguts a l'àrea QA.
--------------------------------------	---

➤ **Dissenyador de les proves:**

Dissenya les proves segons el document de requeriments, té coneixements en disseny de proves, experiència en treball en equip i tècnic de proves.

ACTIVITATS DISSENYADOR DE PROVES	
AD1. Exploració general	Estudia el producte per conèixer ràpidament les noves funcionalitats.
AD2. Anàlisi preliminar de proves	Realitza l'estudi preliminar de l'aplicació a desenvolupar per definir i les proves del projecte.
AD3. Anàlisi de riscos de les proves	Analitza el risc de les proves detectant les funcionalitats més complexes i amb major probabilitat de defectes.
AD4. Planificació del cicle de proves	Defineix, planifica i defineix el cicle de proves amb el responsable del projecte de proves i les funcions a provar en cada cicle.
AD5. Definició de proves	Defineix i organitza com seran les proves. Les proves es divideixen en 3 test: Test d'exploració, test específic i test combinatori.
AD6. Administració de proves	Administra les proves definides junt amb el Tester i assisteix a algunes reunions si és necessari algun aclariment.
AD7. Disseny de les proves	Realitza el disseny de les proves, les analitza i fa un seguiment de les proves al llarg de tot el projecte.
AD8. Avaluació de proves	Avalua els resultats de les proves analitzant els incidents, errors detectats i documentació realitzada al llarg del projecte per possibles millores que pugui aportar l'estudi.

➤ **Tester:**

Executa les proves i reporta els resultats. Ha de tenir coneixements en desenvolupament de proves, en base de dades, comprensió de les fases de prova i tècnic en proves.

ACTIVITATS TESTER	
AT1. Organització cicle de proves	Planifica i organitza el cicle de proves amb el dissenyador de proves per iniciar les execucions de les proves.
AT2. Configuració de l'entorn	Instal·lar les eines necessàries i configurar l'entorn per provar el producte. L'entorn es troba definit al document de requeriments.
AT3. Test d'exploració	Execució del test d'exploració i documentar els resultats obtinguts.
AT4. Test específic (funcional)	Execució del test específic i documentar els resultats obtinguts.
AT5. Test de combinacions	Execució del test de combinacions i documentar els resultats obtinguts.
AT6. Report d'errors	Documenta, reporta i analitza els errors detectats a la fase de proves.
AT7. Validació i verificació de les proves	Realitza el seguiment de validació i verificació de les deteccions i correccions d'errors i reporta qualsevol incident detectat en el seguiment fins tancar els errors de cada entrega al llarg del projecte.
AT8. Documentació interna i manuals d'usuari	Documenta les funcionalitats a nivell intern de l'organització i els manuals pels usuaris finals.
AT9. Formacions	Realitza formacions per Consultoria i servei tècnic sobre els productes alliberats al mercat.
AT10. Avaluació i millores	Realitza el tancament de les diferents fases i avalua les tasques realitzades per suggerir millores per altres versions junt amb el responsable i el dissenyador de proves.

6.4 Desenvolupament

En aquest punt es realitza l'estudi de les diferents etapes mencionades en els punts anteriors: Estudi preliminar, Planificació de proves, Pla de proves, Cicle de proves i Avaluació.

MAQU contempla cinc etapes estàndards per totes les aplicacions. En el cas de sorgir desviacions es reacciona ordenadament, es coordinen les accions amb l'àrea de desenvolupament del producte i es realitzen les planificacions i solucions oportunes en cada situació possible.

MAQU estableix els rols que intervenen en cada etapa i després d'establir el pla de proves organitza el projecte en cicles de proves, on cada cicle correspon a una entrega realitzada des de l'àrea de desenvolupament a l'àrea QA.

En resum, s'ha analitzat i arribat a la conclusió que al involucrar el procés de proves abans del desenvolupament del software s'aconsegueix:

- ✓ Donar visibilitat al equip, de com es provarà un producte.
- ✓ Disminuir els costos de la correcció de defectes.
- ✓ Descobrir defectes lo abans possible, i si encara no s'ha iniciat el desenvolupament del codi és ideal.

6.5 Estudi preliminar

S'examina les tasques, equip, gestió i intervenció en el projecte per poder definir l'abastament de les proves del producte, és a dir, després d'haver realitzat les reunions i documentació necessària es determina quines funcionalitats s'han de provar i quines no.

Darrerament l'equip de desenvolupament allibera el Document de Requeriments final a l'àrea QA sobre el que es realitzarà tot l'estudi per obtenir els resultats pactats amb l'usuari i amb la qualitat que el producte requereix.

En aquesta etapa l'equip QA pot determinar modificacions en els requeriments que consideri necessàries i exposar dubtes o incongruències detectades en el document.

Junt amb l'estudi de tasques a realitzar s'estableixen els temps de desenvolupament i QA per arribar a concloure dates d'entrega a client.

Document de Requeriments

El document de Requeriments conté la nova versió a alliberar al mercat, la descripció de les funcionalitats a implementar, configuracions específiques i característiques de cadascuna de les accions que té el software.

L'especificació dels requeriments ha d'analitzar i detallar cadascuna de les possibles accions que es poden realitzar. D'aquesta manera el responsable del projecte de proves, el dissenyador de proves i el tester podran analitzar i dissenyar amb els mètodes més adequats per cada situació, els cassos de prova més efectius per comprovar, verificar i validar si el comportament real de l'aplicació és l'esperat i si conté defectes o errors.

Validació dels requeriments

S'analitza i revisa el document pel responsable de proves i en cas de detectar algun comportament no acceptable o inestable per l'àrea QA es comunica al gestor de projectes per acordar els incidents detectats.

Els errors o problemes detectats en el document s'han de corregir o modificar abans d'iniciar la implementació de l'aplicació ja que cal evitar al màxim les accions de regressió en qualsevol fase del projecte.

Rols que intervenen

Els rols del projecte de proves que intervenen en aquesta etapa i les activitats:

Responsable de projecte de proves	Dissenyador de les proves
AR1. Abastament del projecte de proves	AD1. Exploració general
AR2. Anàlisi preliminar	AD2. Anàlisi preliminar de les proves
AR3. Anàlisi de riscos	AD3. Anàlisi de riscos de les proves

Etapa 1. Inici de l'estudi

Primerament es determina la intervenció de l'equip QA en el producte, l'abastament i fases en les que es divideix el projecte i les prioritats o amenaces que aporta el nou producte.

Una visió amplia servirà per detectar el risc del producte a provar per començar a organitzar gestió administrativa necessària per iniciar la planificació.

El responsable i el dissenyador de les proves realitzen l'estudi preliminar per detectar possibles amenaces o desviacions no contemplades.

Etapa 2. Seguiment de l'estudi preliminar

El responsable de l'àrea de desenvolupament i de projecte de proves es reuneixen, dins del període de l'estudi preliminar, per determinar i pactar les dates d'entrega en les diverses fases i cicles del projecte, contemplant possibles desviacions i riscos depenent de les característiques de les entregues a testejar.

Un cop pactades les dates s'organitza una reunió amb el gestor de projectes que s'encarrega de gestionar el 'Project' on apareixen les tasques a desenvolupar pels diferents recursos de l'àrea de desenvolupament i de l'àrea QA.

Les reunions realitzades es detallen en actes documentades pel responsable de projecte de proves, evitant possibles desacords, oblitats o negligències.

Etapla 3. Anàlisi dels riscos

El responsable de projecte de proves i el dissenyador de proves, junt amb responsable de desenvolupament, realitzen una llista amb les funcionalitats més arriscades i més complexes que s'han d'analitzar amb més detall perquè tenen més risc i son més importants per l'usuari final i per l'aplicació.

A partir de l'anàlisi preliminar i l'exploració general s'analitzen els riscos associats a les noves funcionalitats i en el cas de no funcionar correctament son les primeres tractades per solucionar-les, son prioritàries davant d'unes altres.

L'objectiu principal d'aquesta etapa es realitzar grups de funcionalitats, cadascuna amb els seus riscos i definir un mapa general de les funcions del producte des de les que tenen més risc i prioritàries fins les que ho son menys. Aquesta divisió ajudarà molt a preparar la Planificació i Pla de proves en les etapes posteriors.

Documentació

La documentació acordada, incidents, solucions i temes tractats en les reunions es redacten pel responsable de proves i son posats a disposició per poder llegir-los en qualsevol moment per la resta dels assistents a la reunió i pels implicats en el projecte.

Es important que la documentació sigui pública i tot l'equip estigui al corrent de les negociacions o modificacions del projecte.

L'àrea de desenvolupament i l'àrea QA comparteixen directoris on es troba disponible la informació pactada, funcionals requerits i tota la documentació generada per la posada en marxa i seguiment del projecte.

Directori QA

A part del directori comú, QA disposa del seu directori on es prepara la documentació pertinent als jocs de proves, planificació de proves i documents generats en el projecte de proves.

En aquesta fase on es preparen les carpetes pel nou projecte, amb la documentació necessària i les plantilles estàndards de documentació que es van omplint al llarg del projecte de proves.

Es en aquest directori on el responsable i dissenyador de proves desaran la documentació de proves generada perquè estigui disponible per l'execució de les proves responsabilitat dels testers.

6.6 Planificació de proves

El document de requeriments es troba en mans de l'equip QA, ja es coneixen els punts crítics a verificar i validar, es pot procedir a la planificació de les proves depenent del risc del producte establert a l'estudi preliminar, així com l'establiment de les prioritats i documentació del pla de proves a seguir.

Aquest punt es podria començar junt amb l'estudi preliminar, però només aquelles tasques que no puguin ser refusades per l'usuari final, és a dir, preparació de tasques que no es veuen afectades pels canvis que s'estiguin realitzant en el document de presa de requeriment sinó s'estaria realitzant feina doble 'fer i desfer', arribant a obtenir una dolenta planificació de testing.

Rols que intervenen

Els rols del projecte de proves que intervenen en aquesta etapa i les activitats:

Responsable de projecte de proves	Dissenyador de les proves
AR4. Planificació del projecte	AD4. Planificació del cicle de proves
AR5. Anàlisi de requeriments	AD5. Definició de proves
AR6. Planificació del cicle de proves	

Etapa 1 Inici planificació

L'equip QA disposa del mapa general de funcionalitats a testear i s'ha d'establir una estratègia de proves depenent dels riscos i prioritats definides.

Es planifica l'abastament definitiu de les proves i es determina un calendari dels cicles de proves. Durant el projecte es realitzen diverses entregues que han d'estar controlades en tot moment segons el calendari planificat inicialment.

Totes les desviacions produïdes al llarg del projecte s'han de visualitzar en el Project posat a disposició i pactat des de l'estudi preliminar.

Etapa 2 Revisió requeriments

Es defineix amb detall el document de requeriments i es revisen les funcionalitats classificades com prioritàries i de més risc que s'han de tenir en compte alhora de planificar el cicle de proves.

S'estudien les especificacions per decidir si és possible generar les proves o es necessita implementar documentació. És necessari que les especificacions estiguin ben definides per poder definir les proves.

En el cas de que hi hagi moltes especificacions perquè es tracta d'un projecte extens es pot concretar en calendari una altra data per tractar les funcionalitats que

s'alliberaran en posteriors entregues. La primer entrega és en aquesta etapa la més important ja que és on es presentarà el producte amb les primeres funcionalitats implementades.

Eta3 3 Definició i cicle de proves

Es defineixen els calendaris pels cicles de proves i es conclouen les funcionalitats que seran provades a cicle.

Es pacten:

- ✓ Dates inici i fi dels cicles de proves.
- ✓ Data inici de report d'errors de les diferents entregues.
- ✓ Temps necessari per solucionar els errors detectats i generar l'entrega de les correccions. En moltes ocasions es faran coincidir amb la data d'inici d'un cicle per no generar masses entregues.

La decisió de les proves a executar a cada cicle depèn del risc definit a l'Estudi preliminar i s'utilitzen diverses tècniques per no repetir execucions de proves cada vegada.

Les proves a executar en un test poden ser infinites i cada vegada que es realitza una entrega per part de desenvolupament moltes d'elles s'haurien de tornar a provar per assegurar que es manté la qualitat des del cicle anterior.

Però el temps i cost no permet realitzar aquest comportament i moltes vegades hi ha proves que s'executen una vegada i no es tornen a executar. Per solucionar aquest dolent hàbit es realitza un test d'exploració al inici de cada entrega per verificar que l'entrega és vàlida i que es pot continuar amb el test específic de les funcionalitats i/o correccions de cicles anteriors. El fet de tornar a provar algunes proves ja executades amb anterioritat s'anomena 'Proves de regressió' i s'han de tenir en compte alhora de planificar les dates d'entregues.

Documentació

A la fase de planificació es redacta un document 'Test Pla de proves' que descriu les estratègies a seguir durant el desenvolupament de les proves. L'objectiu d'aquest document es recollir tota la informació necessària pel seguiment del projecte de proves sense tenir en compte els detalls de l'execució i resultats de les proves. Aquesta informació es recollirà en un altre document que fa el dissenyador de les proves especificant els casos de proves, així com les entrades i sortides esperades i procediments a seguir en cadascuna de les proves.

Contingut del document 'Test Pla de proves':

- ✓ Identificador del document.
- ✓ Data del document.

- ✓ Dades del projecte de test.
- ✓ Rols i responsabilitats.
- ✓ Control de versions i entregues.
- ✓ Abastament de les proves.
- ✓ Calendari del projecte de proves.
- ✓ Inici i fi dels cicles de proves.
- ✓ Condicions per iniciar el cicle de proves.
- ✓ Requeriments de l'entorn.
- ✓ Estratègies de test.
- ✓ Tècniques de proves.
- ✓ Eines de proves.
- ✓ Anàlisi de riscos.
- ✓ Revisions.
- ✓ Procediment pels report d'errors i seguiment.

6.7 Pla de proves

En el pla de proves es concreten les proves a realitzar, es gestiona els tipus de proves a aplicar a les funcionalitats dels diversos cicles de proves, es comença a documentar els plans de proves detallats i s'analitzen els casos de prova:

- ✓ Revisió de les entrades: es considera el document de requeriments, planificacions del projecte, test pla de proves, ... es secciona i s'estudia en profunditat cadascun dels casos proposats.
- ✓ Revisió dels tipus de test a executar (casos de prova).
- ✓ Automatitzacions: s'analitzen casos que es puguin automatitzar i àrees que s'identifiquin per realitzar proves de rendiment o de estrès (es l'àrea de desenvolupament la que s'encarrega d'aquest tipus de proves).
- ✓ Pla de regressió: s'estudien els cicles de proves a realitzar, és a dir, el número de vegades que s'ha de verificar els defectes trobats a les diferents entregues.

Es important realitzar les proves adequades i amb un previ estudi del disseny de les mateixes ja que si no es realitzen les proves adequades:

- ✓ El software que s'implementa no compleix amb els requeriments o els compleix parcialment.

- ✓ En lloc de produir o implementar noves funcionalitats es perd el temps arreglant errors.
- ✓ Les fallides del software poden tenir greus conseqüències.

Rols que intervenen

Els rols del projecte de proves que intervenen en aquesta etapa i les activitats:

Responsable de projecte de proves	Dissenyador de les proves	Tester
AR7. Seguiment de fases	AD6. Administració de proves	AT1. Organització cicle de proves
AR8. Avaluació de proves	AD7. Disseny de les proves	

A continuació es descriuen cadascuna de les etapes del Pla de Proves:

Etapa 1 Definició i organització de les proves

Es planifica la informació necessària per iniciar el disseny i execució de les proves de manera ordenada, amb tota la documentació apunt i les funcionalitats ben enteses i organitzades.

Es realitzen reunions amb els testers per traspassar les planificacions pactades i la documentació i funcionalitats a testear a cadascun dels cicles.

Etapa 2 Documentació

El disseny de les proves es recull en un document 'Test Disseny' que especifica les característiques a provar, els requisits que han de complir les proves i els resultats obtinguts de les proves, tots els camps seran omplerts pel tester alhora d'executar les proves.

Les proves es documenten de forma organitzada, totes tenen una entrada i una sortida i els passos a executar per cada prova. Per definir-les es poden utilitzar estratègies i tècniques que faciliten la identificació i elaboració de les proves.

El 'Test disseny' és un document Excel que té diferents fulls descrits a la primera fulla del document:

- ✓ 1r full: Especificacions del disseny de proves.
- ✓ 2on full: Test d'exploració
- ✓ 3r full: Test de proves específiques
- ✓ A partir del 4rt full: Test de combinacions

Els tests que s'especifica en els demés fulls Excel del document s'especifiquen els jocs de proves a provar i tots els elements a contemplar en cadascuna de les proves.

Aspectes a tenir en compte en la definició i disseny del Test Disseny:

- ✓ Una bona definició de les proves ajuda a no perdre temps alhora d'executar-les invertint el temps en pensar què s'ha de provar o com s'ha de provar.
- ✓ Unes proves efectives garanteixen la detecció d'errors molt més ràpida, podent solucionar-los de manera planificada, organitzada i oferint una major probabilitat d'entregar un producte de major qualitat.
- ✓ Totes les proves s'emmagatzemen en aquest document amb la possibilitat i objectiu de poder ser utilitzades en altres versions o funcionalitats semblants a testejar.

Etape 3 Disseny de les proves

L'ordre i importància en que es dissenyen les proves depèn de les prioritats establertes en l'estudi inicial.

A partir del document de requeriments i la documentació que hi ha disponible s'inicia el disseny dels casos de prova de l'aplicació. Donat que l'àrea de desenvolupament ja ha iniciat la codificació de l'aplicació, s'ha de tenir en compte que poden sorgir modificacions per això hi ha pràctiques que cal aplicar en el disseny de les proves:

- ✓ No crear documents de proves molt grans ja que costa molt de mantenir.
- ✓ No desenvolupar un gran volum de proves del mateix grup, ja que es pot fer massa extens i arribar a dates d'entrega i no tenir acabats els dissenys.
- ✓ Construir proves genèriques que es testegin en les diferents entregues, per no dissenyar proves iguals en diferents moments. Compactar al màxim.

Les aplicacions poden ser sotmeses a infinites proves, per això s'ha ideat tres tipus de test a realitzar alhora de testejar una entrega: Test d'exploració, test de proves específiques i Test de combinacions. Depenent de la complicació i magnitud del producte els diferents test seran més o menys extensos.

Per dissenyar les proves es coneixen els següents punts:

- ✓ Funcionalitats que han de ser provades.
- ✓ Conèixer i entendre les prioritats donades per l'anàlisi de riscos a les fases anteriors.
- ✓ Desenvolupar el disseny de les proves de manera clara i agrupar pels diferents tipus de proves.

- ✓ Dissenyar les proves a partir de les plantilles estàndards disponibles per cada versió definides al document 'Test Disseny'.

L'objectiu del disseny de les proves és especificar els casos a provar, descriure els passos per executar les proves correctament i establir els requisits de l'entorn específic de les diferents proves.

Etape 4 Validació de les proves

El responsable de projecte de proves valida les proves que s'han realitzat i en cas necessari aporta modificacions o casos a contemplar.

6.7.1 Tests i proves a dissenyar

A continuació es defineixen els diferents tests a dissenyar que posteriorment seran executats pels testers per detectar els possibles errors i ser reportats per solucionar abans de l'alliberament del producte.

El cicle de vida de les proves pot ser paral·lel amb el desenvolupament de l'aplicació i el disseny de les proves és una de les fases que es pot treballar de forma paral·lela. No cal disposar de l'aplicació per dissenyar les proves només cal disposar de la documentació necessària per desenvolupar-les.

- ✓ Document d'especificació de requeriments que conté la versió i funcionalitats sobre la que s'ha de dissenyar les proves oportunes.
- ✓ Document Test Pla de Proves que recull els riscos, prioritats, pactes, etc. de les proves.
- ✓ Document Test Disseny per implementar les proves.

6.7.2 Especificacions del disseny de proves

Es presenta a la primera fulla del document 'Test Disseny' i conté els següents punts:

- ✓ Identificador del document.
- ✓ La data del document.
- ✓ Identificació de la versió a testejar, de les funcionalitats i proves.
- ✓ Característiques a provar:

S'identifica les funcionalitats a provar i els aspectes a tenir en compte pel disseny de les proves.

- ✓ Especificació del procés de proves:
 - S'especifica els passos a seguir en l'execució de les proves.

- Configuració dels entorns en els diferents casos de prova.
 - Mètodes i eines a utilitzar.
 - Accions a realitzar abans, durant i després d'executar les proves.
 - Descripció d'algunes accions a realitzar en cas de donar-se determinades situacions no contemplades.
 - Requisits determinats per algunes proves.
- ✓ Establiment del criteri de defectes:

S'estableix un criteri per considerar els resultats reals de l'aplicació i classificar-los en defectes o proves superades positivament.

6.7.3 Test d'exploració

El test d'exploració es presenta a la segona fulla del 'Test Disseny' i conté proves que permeten identificar ràpidament el comportament de l'aplicació. Es detecta una primera visió del producte i es comença a familiaritzar amb el nou producte a testejar.

Aquesta exploració permet ràpidament confiar en l'entrega que desenvolupament ha alliberat a QA i si aquesta és vàlida es continua amb el Testing planificat restant. Aquest test es realitza al rebre cadascuna de les entregues prèviament pactades per l'alliberament de l'aplicació des de desenvolupament cap a QA.

En el cas de no superar el test d'exploració es valora si continuar amb el test o retornar l'entrega a desenvolupament per no haver superat l'exploració.

Proves claus

Per dissenyar el test d'exploració s'ha de tenir experiència i formació per saber distingir i classificar les diferents parts de l'aplicació i detectar quines son les proves que es consideraran claus per validar el producte ràpidament amb l'objectiu de trobar els defectes més ràpidament.

Proves eliminades

De igual forma, l'experiència ajuda a identificar quines son les parts que s'eliminen de l'exploració i que es deixaran per desenvolupar en altres casos de prova.

Casos de prova

Les proves claus son proves funcionals i estructurades que cobreixen el producte i no contenen totes les combinacions i resultats esperats de l'aplicació.

Els casos de prova de l'exploració majoritàriament son explicacions de com i quin camí s'especifica al document de requeriments per comprovar que el resultat real que ens ofereix l'execució es correspon amb el resultat esperat.

Les proves es centren en les parts de l'aplicació amb més risc i establertes amb prioritat i amb l'explicació de com realitzar la prova és tasca del tester executar aleatòriament i inspeccionar el producte per validar el cas de prova.

6.7.4 Test de Proves específiques (funcional)

El test de proves específiques o funcional es presenta a la tercera fulla del 'Test Disseny' en una llista que presenta les funcionalitats a verificar i validar amb els resultats esperats establerts en el document de requeriments i pretén examinar l'aplicació intensament validant el comportament observat.

Amb les tècniques i mètodes escollits s'ha de dissenyar els casos de prova que demostrin les condicions vàlides i invàlides de l'aplicació.

Per això és imprescindible tenir unes especificacions i una versió executable del mateix. Es revisen les especificacions per analitzar i estudiar els casos de prova a partir d'elles, en el cas que no existeixin especificacions o que siguin incompletes, es treballa per analitzar les carències i els desenvolupadors s'encarreguen de millorar-les. Aquest és el comportament ideal, però el cert és que les dates de client són molt ajustades i s'ha d'intentar fer una feina ben feta des del principi del cicle per tal de no perdre temps que després repercutirà en la qualitat del software i el retràs de dates d'entrega.

L'objectiu del test funcional és revelar si existeixen errors en la funcionalitat del producte i que no es troben en desacord amb el comportament esperat de l'usuari final. L'aplicació en el moment de l'entrega ha de complir amb el que l'usuari necessita i no ha de contemplar errors de cap tipus, s'entrega una aplicació ben depurada, és a dir, exempta d'errors.

Amb l'objectiu de cobrir l'especificació de requisits els punts per dissenyar els casos de prova del test de proves funcionals són els següents:

- ✓ Identificador i nom del cas de prova.
- ✓ Anàlisi de les prioritats i riscos determinats en les primeres fases de proves.
- ✓ Accions a realitzar a les proves.
- ✓ Resultat real i esperat de les proves.
- ✓ Desenvolupament de les proves en grups, és a dir, les proves es relacionen entre elles depenent de l'origen especificat.
- ✓ Es desenvolupa una mapa de proves que mostra la correspondència entre les funcionalitats i els casos de prova exposats.
- ✓ Utilització de tècniques de caixa negra, anàlisi de valor límit, etc. que permeten detectar:
 - Un funcionament incomplet o incorrecte.
 - Errors a la interfase.

- Errors de dades.
- Problemes de rendiment.
- Problemes d'inici o finalització d'accions.
- Capacitat de les accions.
- Valors representatius d'un conjunt de dades.
- Fronteres, valors o combinacions conflictives.

El disseny conté uns camps específics estàndards per totes les versions i es poden afegir de nous en cas necessari. Aquestes modificacions sobre el disseny es valorarà per incloure en els estàndards per versions posteriors.

6.7.5 Test de Proves de combinacions

El test funcional identifica cadascuna de les funcionalitats a testejar però no detalla totes les combinacions possibles amb les accions i camps analitzats al test funcional.

Les combinacions son accions vàlides o invàlides proposades de forma seqüencial per obtenir un resultat real.

Per cada funcionalitat es realitza un test combinatori sobre les accions possibles a executar i totes les proves cobreixen la funcionalitat des de tots els punts de vista des d'on puguin ser executades.

Els casos de prova del test combinatori segueix els punts:

- ✓ Contempla casos vàlids i invàlids (proves de robustesa).
- ✓ No es fixa si el resultat és adequat i òptim, cerca defectes, errors, casos inesperats, etc.
- ✓ Contempla condicions normals o anormals de l'aplicació.
- ✓ Determinar la cobertura: No sempre és possible contemplar totes les combinacions, per això es determina el percentatge en relació a la totalitat de casos possibles.
- ✓ Les execucions de les proves es realitzen seguint el recorregut exposat pas a pas i especificant el resultat obtingut.
- ✓ Es construeix un mapa de proves amb els camins possibles i les proves es poden agrupar segons l'origen. S'utilitzen tècniques de caixa negra com Partició equivalent, taules de decisió, etc.
- ✓ S'identifiquen les entrades pertinents i es genera les combinacions per obtenir les sortides.

6.8 Cicle de proves

Al Cicle de proves s'executen les proves prèviament dissenyades i establertes pel dissenyador de proves. El cicle de proves son aquelles proves que s'executen en cada entrega que realitza l'àrea de desenvolupament a l'àrea QA, és a dir, cada cicle correspon i tindrà la numeració de l'entrega realitzada i finalitzarà el cicle quan es reportin els errors detectats en aquella entrega.

L'equip QA disposa de la primera entrega de l'aplicació, segons pactat a la planificació del projecte, i s'inicia el seguiment del cicle contrastant el comportament esperat del software amb el comportament real que es detecta en l'execució. Les diferents entregues correlatives poden ser funcionalitats o poden ser correccions d'errors reportats en anteriors entregues que s'han de solucionar abans de l'alliberament del producte a l'usuari final.

Les proves les executa el tester i els errors, diferències o defectes de l'aplicació son tractats amb el responsable de projecte de proves en les reunions de seguiment, d'aquesta manera s'avaluen les proves que s'estan realitzant i l'estat de les fases del procés de proves. Els errors acordats amb el responsable es reporten al departament de desenvolupament en les dates pactades per procedir a la seva resolució, documentant la prioritat, passos a seguir per la reproducció dels errors, etc.

Rols que intervenen

Els rols del projecte de proves que intervenen en aquesta etapa i les activitats:

Responsable de projecte de proves	Dissenyador de les proves	Tester
AR7. Seguiment de fases	AD8. Avaluació de proves	AT2. Configuració de l'entorn
AR8. Avaluació de proves		AT3. Test d'exploració
		AT4. Test específic
		AT5. Test de combinacions
		AT6. Report d'errors
		AT7. Validació i verificació de les proves
		AT8. Documentació interna i manuals d'usuari
		AT9. Formacions

6.8.1 Configuració de l'entorn

L'entorn de proves s'ha de preparar abans de l'execució de les proves per tal d'obtenir uns resultats reals i amb l'èxit esperat. L'entorn ha de ser lo més proper possible a l'entorn final en el que serà utilitzat el producte, així es disminueixen els problemes que moltes vegades sorgeixen degut a l'entorn de l'usuari final.

L'entorn de proves és establert pel tester abans d'executar les proves i aquest no pot ser preparat en el mateix entorn en el que s'ha desenvolupat l'aplicació ja que pot provocar no detectar errors no contemplats durant el desenvolupament. Depenent del tipus de test que estiguem executant utilitzarem un entorn o un altre, l'entorn més adequat es troba definit al document de Test Disseny prèviament estudiat i documentat.

La configuració de l'entorn es realitza al llarg de tot el desenvolupament del test, cada entrega de proves s'ha de configurar l'entorn abans d'iniciar el test corresponent.

Els recursos que es necessiten per configurar l'entorn són les següents:

- ✓ Entorn de desenvolupament de software i hardware (eines, sistemes operatius, bases de dades (Oracle, SQL), etc.
- ✓ Comunicacions (connexions, protocols, ...)
- ✓ Instal·lacions
- ✓ Documentació (Document de requeriments, Test Pla de proves i Test Disseny). Durant tot el cicle de proves es van omplint els documents i a les reunions de seguiment es reporta la informació al responsable del projecte de proves per detectar l'estat del projecte i les possibles incidències a destacar i solucionar com abans millor.

Gestió d'aplicacions, versions i entregues

El departament QA organitza la documentació en les diferents entregues pactades i coordinades amb l'àrea de desenvolupament de la versió a testear.

Una aplicació ve definida per uns paràmetres, aquests permetran identificar qualsevol activitat realitzada en el cicle de vida de l'aplicació i així en un futur si es necessita qualsevol informació d'un software és fàcilment identificable.

Els paràmetres que defineixen l'aplicació són:

- ✓ Versió del producte: la numeració és única per cada versió.
- ✓ Entorn en el que es desenvolupa la versió del producte.
- ✓ Producte i funcionalitats de la versió.

- ✓ Entregues de la versió: la versió disposa de diverses entregues, unes prèviament pactades i d'altres que s'afegiran al llarg del projecte per solucionar els defectes trobats en els tests executats.
- ✓ Documentació relacionada amb la versió.
- ✓ Eina de seguiment d'errors per versió.

Eina de seguiment d'errors

Els errors es reporten amb una eina de seguiment d'errors 'Bugzilla' ⁷ dissenyada per assegurar la qualitat del software on poden intervenir totes les persones involucrades en el desenvolupament d'un producte per reportar els defectes del software.

Aquesta eina permet organitzar els errors, tenint cadascun d'ells una numeració única, i fer un seguiment de múltiples versions. També permet organitzar per categories els diferents productes de les aplicacions i classificar els errors per la seva prioritat, per les seves característiques (crític, bloquejant, normal o millora).

El tester reporta els defectes trobats a través de l'eina omplint tots els camps demanats amb la possibilitat d'adjuntar imatges o documents per fer possible al desenvolupador una ràpida comprensió del error i així solucionar-lo amb més eficàcia i rapidesa, sense invertir el temps en comprendre l'error.

Bugzilla permet afegir comentaris als errors per qualsevol implicant del projecte amb accés a l'eina. S'identifiquen els responsables, es fan assignacions dels errors i es documenta la solució i impacte que ha tingut l'error.

⁷ [BUGZ] Eina de seguiment d'errors

Summary: contains all of the words/strings Search

Classification: Product: Component: Version: Target:

Unclassified add-ons.mozilla.org * mozilla.org 0.1 ...

Client Software AUS API 0.2 0.007

Components Bugzilla Accessibility 0.3 0.008

Server Software Calendar Account Manager 0.4 0.009

Client Support Camino Add-ons 0.5 0.010

A Comment: contains the string

The URL: contains all of the words/strings

Whiteboard: contains all of the words/strings

Keywords: contains all of the keywords

Status: Resolution: Severity: Priority: Hardware: OS:

UNCONFIRMED FIXED blocker P1 All All

NEW INVALID critical P2 DEC Windows 95

ASSIGNED WONTFIX major P3 HP Windows 98

REOPENED DUPLICATE normal P4 Macintosh Windows ME

RESOLVED WORKSFORME minor P5 PC Windows NT

VERIFIED INCOMPLETE trivial PocketPC Windows 2000

CLOSED MOVED enhancement SGI Windows XP

Windows Server 2003

Windows Vista

Email Addresses, Bug Numbers, and Votes

Any one of:

☒ the bug assignee

☐ the reporter

☐ the QA contact

☐ a CC list member

☐ a commenter

Any one of:

☒ the bug assignee

☒ the reporter

☒ the QA contact

☐ a CC list member

☐ a commenter

is

is

Bug Changes

Only bugs changed between: and Now

(YYYY-MM-DD or relative dates)

where one or more of the following changed:

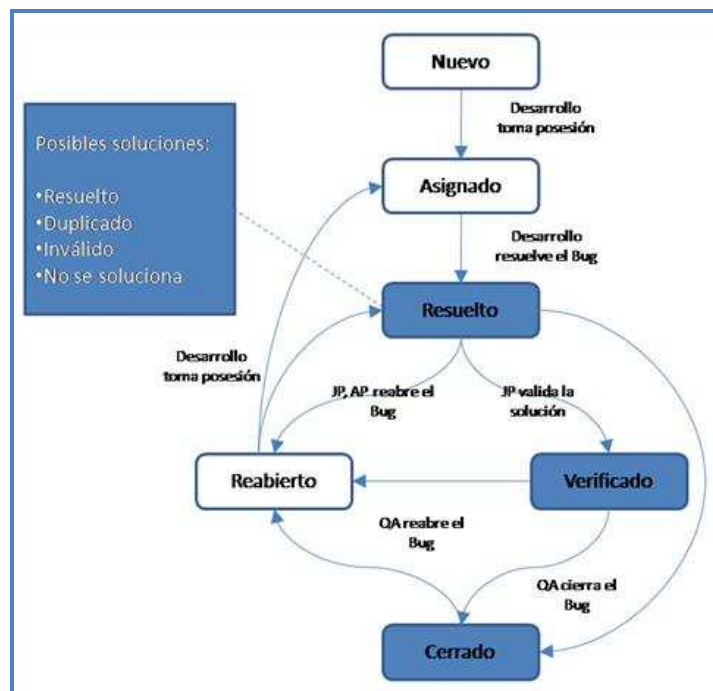
[Bug creation]

Alias

Assignee

CC list accessible?

Cicle de vida de l' error



MAQU disposa d'un document 'Report d'errors' que explica el procés de reportar errors, com es reporta un error, amb quins formats, quines característiques i passos s'han de seguir per reportar un error de manera correcta i el mètode a seguir en qualsevol situació. Cal descriure com es documenten els errors i quins passos s'han de descriure primer i com s'han de descriure pel tal de poder configurar l'entorn adequat per reproduir l'error fàcilment.

6.8.2 Execució de les proves

Es realitzen les proves per cada versió del producte. Les proves que s'apliquen, depenent del producte a testejar i dels errors amb els que s'alliberi el software a QA, es realitzaran més o menys entregues i el cicle de proves serà més extens, podent provar totes les proves planificades, o menys extens provant menys proves de les planificades perquè hi ha punts de l'aplicació on no es pot continuar el test (s'analitza quin tipus d'errors poden provocar aquestes situacions).

En el seguiment del cicle de proves s'executen els tests dissenyats al document 'Test Disseny' en el següent ordre: Test d'exploració, Test de proves específiques i test de proves de combinacions.

L'execució de les proves s'han de realitzar d'acord amb la planificació estipulada. En cada cicle de proves es van detectant errors i s'obtenen resultats que es van avaluant a mida que avança el test de forma disciplinada i amb l'ordre establert en el que hem d'executar les proves.

En el document de 'Test Disseny' es recullen:

- ✓ Identificador del document.
- ✓ Data del document.
- ✓ Funcions que estan essent provades.
- ✓ Entorn en el que s'estan desenvolupant les proves.
- ✓ Descripció de les execucions realitzades.
- ✓ Activitats d'entrada de les proves.
- ✓ Resultats que s'obtenen en l'execució de les proves.
- ✓ Impacte que provoca els incidents trobats en l'execució.
- ✓ Dates i temps en el que s'estan executant les proves.
- ✓ Comentaris, per si es vol afegir qualsevol detall de la prova.

6.8.3 Verificació i validació del software

La verificació i validació d'un producte es dur a terme al llarg de tot el cicle de vida oferint una major confiança en el desenvolupament i faciliten el manteniment del mateix. Perquè un sistema resulti robust ha de ser verificat i validat per evitar alliberar un producte amb debilitats i inestable podent provocar problemes importants o menys importants en els usuaris que treballin amb el producte.

S'està construint el producte correctament?

En la verificació del software es confirma l'existència d'entrades de dades adequades per a qualsevol activitat i que les sortides son les correctes i compatibles amb les entrades definides.

S'està construint el producte correcte?

En la validació del software es confirma que les funcions i execució dels requisits s'implementen correctament i completament en el producte final.

Estudi del model en V

En el Model en V el procés de desenvolupament i testing es realitza en paral·lel i permet validar i verificar cada etapa del desenvolupament en el moment en que s'està realitzant: quan finalitza una etapa, es realitza la verificació, mentre que l'etapa de testing realitzada paral·lelament és la validació.

- ✓ Fàcil d'entendre i simple.
- ✓ Ordenació de les activitats en la seqüència del temps.
- ✓ Abstracció dels nivells per mostrar gràficament la connexió entre les activitats de desenvolupament i de Testing.
- ✓ Organitza les activitats del testing en diferents nivells.

Com s'ha analitzat en el punt de models de cicle de vida del software, el model en V és el més aproximat en la relació entre desenvolupament i QA però l'etapa d'acceptació entra massa tard en el cicle de vida i MAQU vol establir un cicle paral·lel al desenvolupament del producte evitant les possibles desviacions lo abans possible en el cicle de vida del software. De totes maneres és un model analitzar per extraure idees fins aconseguir el model que necessita MAQU.

Model en V Procés de verificació

El propòsit és confirmar que cada sortida compleix els requisits específics.

Passos del procés de verificació:

1. Desenvolupament de l'estratègia per realitzar la verificació.
2. Identificació dels criteris de verificació per cada resultat.
3. Realització de les activitats de verificació requerides.
4. Identificació i registre dels defectes i errors trobats.
5. Report dels resultats de les activitats de verificació al document Test Disseny.

Model en V Procés de validació

El propòsit és confirmar que es compleixen els requisits per l'ús adequat i prèviament pactat del producte.

Passos del procés de validació:

1. Desenvolupament de l'estratègia per realitzar la validació.
2. Identificació dels criteris de validació per tots els resultats.
3. Realització de les activitats de validació requerides.
4. Identificació i registre dels defectes i errors trobats.
5. Report dels resultats de les activitats de validació al document Test Disseny.

Proves de regressió

Els cicles de proves contenen les execucions de les proves prèviament planificades i s'han de validar les correccions realitzades pels desenvolupadors dels incidents trobats en entregues anteriors.

Aquestes proves consisteixen en la verificació del producte després d'haver patit modificacions amb el fi de comprovar si el producte conté els resultats esperats i que les modificacions no han afectat a altres components i provocat altres errors. Aquestes proves es poden realitzar a qualsevol fase del procés de proves.

Els desenvolupadors realitzen les modificacions dels errors reportats i QA realitza les proves de regressió per assegurar que han sigut corregides correctament i que es pot procedir al tancament de l'error a través del Bugzilla. Els errors corregits han de presentar el comportament desitjat i en cas de no presentar aquest estat o aportar nous errors, implica que les proves s'hauran de tornar a executar i reportar el comportament localitzat.

Es en aquest procés quan es poden detectar més desviacions en la planificació dels cicles de proves i posteriorment com han de ser solucionats també pot provocar desviacions en les entregues següents a realitzar des de desenvolupament.

6.8.4 Report dels resultats

Als diferents cicles de proves es reporten els resultats de cadascuna de les entregues, és ideal que cada cicle hi hagi un report però a vegades i depenent de la gravetat dels errors QA necessita realitzar més d'un report encara que estigui dins del mateix cicle de proves, és a dir, dins del període de la mateixa entrega.

Un cicle de proves finalitza quan s'executen les proves corresponent al cicle i s'analitzen, s'avaluen i es comuniquen els resultats obtinguts, d'aquesta manera s'obté l'avaluació del cicle i la qualitat que el producte està adquirint a cada cicle.

Un cop finalitzades les execucions de proves d'una entrega, es reporten els últims incidents detectats als desenvolupadors per procedir a la correcció. Pot ser que abans d'aquest report d'errors haguessin hagut algun més, però aquest serà l'últim del cicle de l'entrega en qüestió.

Els passos que es segueixen són els següents:

1. S'analitzen i avaluen els defectes trobats al cicle de proves.
2. QA reporta els incidents detectats: a través de l'eina de seguiment d'errors (Bugzilla) assigna al responsable del departament de desenvolupament els defectes trobats.
3. El departament de desenvolupament allibera la següent entrega a QA per començar un nou cicle de proves.
4. L'alliberació de les diferents entregues finalitzen el cicle de proves present, això significa que a vegades poden sorgir desviacions en les entregues o en el report dels errors i podria ser que QA tingues una entrega abans de finalitzar el cicle. En aquesta situació igualment es finalitza el cicle i començaria el cicle següent, ja que és necessari que el cicle de proves que s'està realitzant coincideixi amb l'entrega que desenvolupament allibera a QA. El següent cicle de proves contindria les proves que han quedat pendents de l'anterior cicle afegides a les proves planificades pel següent cicle.
5. De la mateixa manera pot succeir en el cas contrari, pot ser que QA finalitzi el cicle de proves planificat i encara no tingui la següent entrega que allibera desenvolupament, en aquest cas QA inverteix el temps en avaluar els errors localitzats, omplir documentació pendent, realitzar reunions de seguiment, etc. Si s'estableix una bona planificació les dates de les entregues i cicles de proves coincideixen de manera que el software podrà ser entregat a l'usuari final en les millors condicions, amb la qualitat esperada i en les dates pactades.

A partir dels resultats es podran determinar accions de correcció i plans de millora que s'hauran de comunicar i planificar amb les persones implicades en el desenvolupament del software.

6.8.5 Avaluació de les proves

Es realitza una avaluació de les proves per determinar si les proves executades que s'estan realitzant són les adequades i s'analitzen les possibles millores pel procés de proves.

L'avaluació es realitza al llarg de tot el projecte per detectar lo més ràpid possible les desviacions o accions que cal corregir i prendre les decisions oportunes en cada moment i determinar aquells punts que s'ha de millorar.

Per avaluar les proves s'analitza els següents punts:

- ✓ Duració de testing.
- ✓ Defectes detectats a l'entrega.
- ✓ Desviacions produïdes.
- ✓ Documentació i informes generats.
- ✓ Estratègia de proves aplicada.
- ✓ S'analitzen debilitats localitzades.

Documentació

Per recollir l'avaluació al llarg del projecte de proves es genera un document 'Document d'Avaluació' que argumenta els punts anteriors per poder obtenir les diferents avaluacions al llarg del projecte amb les decisions i millores adoptades en les diferents situacions.

El document conté:

- ✓ Identificador del document.
- ✓ La data del document.
- ✓ Descripció de les proves i tests realitzats.
- ✓ Descripció de l'avaluació basat en els punts anteriors analitzats.
- ✓ Comentaris i millores conclosos.

Documentació interna i manuals d'usuari

El departament QA és el responsable d'alliberar a la resta de la organització al finalitzar el projecte, la documentació interna del producte i els manuals de la nova versió que surt al mercat.

➤ *Documentació interna:*

Públic per l'empresa i especifica les noves funcionalitats de les que disposa un nou projecte, implementacions o modificacions sobre alguna aplicació ja disponible en el mercat.

➤ *Manuals d'usuari de la versió:*

Públic per tots els usuaris finals i futurs clients interessats en els productes de l'organització.

Els manuals contenen la descripció de l'aplicació i dels passos a seguir per realitzar qualsevol acció amb exemples i imatges reals de l'aplicació per poder entendre amb facilitat l'ús de l'aplicació.

Formacions a l'organització

En finalitzar i alliberar les versions es realitzen les formacions oportunes pactades amb els responsables de les diverses àrees de l'empresa: Consultoria i Servei tècnic.

A les formacions es transmeten els coneixements funcionals, formes d'actuar, s'expliquen exemples prèviament preparats i es resolen els dubtes que puguin sorgir.

D'aquesta manera l'empresa disposa de la documentació necessària i de l'aprenentatge adequat per resoldre qualsevol incidència procedent d'un client.

Dites formacions son molt importants ja que son aquestes persones les que personalment parlen amb els clients i resolen els seus dubtes.

6.9 Avaluació i millores

S'avalua els resultats del procés de prova a través de la documentació, estadístiques, seguiments i documents d'avaluacions obtinguts per determinar les accions a realitzar per la millora dels mètodes establerts.

Els resultats obtinguts es comuniquen i documenten per posteriorment planificar les millores desenvolupades.

Rols que intervenen

Els rols del projecte de proves que intervenen en aquesta etapa i les activitats:

Responsable de projecte de proves	Tester
AR9. Avaluació de la qualitat	AT10. Avaluació i millores

En aquesta etapa el projecte es tanca amb el report final d'un informe que conté els següents punts a avaluar pel tester i el responsable de projecte de proves:

- ✓ Verificar temps i esforç en les diferents etapes del projecte.
- ✓ Valorar els mètodes i eines utilitzats.
- ✓ Revisar el compliment dels requisits.
- ✓ Revisar i analitzar els resultats del pla de proves.
- ✓ Conclusió dels resultats obtinguts en els finals de les diferents entregues de la versió a testear fins a l'alliberament.

- ✓ Avaluar i documentar la qualitat del producte resultant (ISO 9126):
 - Facilitat d'ús.
 - Funcionalitat adequada.
 - Fiabilitat.
 - Facilitat de manteniment.
 - Portabilitat.

Documentació

Per recollir l'avaluació de millores i accions correctives es redacta el 'Document Pla de millora' que conté:

- ✓ Identificador del document.
- ✓ La data del document.
- ✓ Descripció de l'avaluació i millores exposades per posteriors versions.
- ✓ Desenvolupament de les conclusions i accions acordades per gestionar i planificar pels posteriors projectes.
- ✓ Comentaris.

7 Cas pràctic aplicant MAQU

El software utilitzat pel present estudi és un nou mòdul implementat a l'aplicació Tempo Millenium de la qual disposen actualment molts clients ja en producció. L'aplicació ja consta d'altres mòduls i amb l'implementació d'aquest es donarà força al producte tant per clients que ja disposen de Tempo Millenium com per nous clients interessats en les novetats.

Per major facilitat de comprensió de les funcionalitats de l'aplicació es redacta un manual d'usuari que permet una visió més amplia dels passos a seguir per executar i entendre el funcionament de les diferents parts del software. La documentació del mòdul també forma part d'aquest projecte ja que el manual d'usuari és imprescindible pel client.

El manual d'usuari actualment es redactarà només en Castellà. Tempo Millenium és una aplicació que es disposa en Castellà però en un futur pròxim es disposarà de l'aplicació en Català i Anglès.

Per motius de confidencialitat, es presenten els mètodes i exemples de les proves funcionals però no detalladament les funcionalitats a provar ni els casos de prova generats. Això sí, el més important es mostrar el cas d'estudi sobre dades reals analitzades en el present projecte i la metodologia aplicada amb els seus avantatges i desavantatges obtinguts al llarg del projecte que servirà d'estàndard per posteriors anàlisi al procés de proves.

En el aquest capítol es desenvolupa la metodologia aplicada al procés de proves o testing en la implementació d'un nou mòdul a l'aplicació Tempo Millenium.

En el primer apartat es descriu l'aplicació Tempo Millenium sobre la que es realitza el projecte i una descripció del mòdul sobre el que es realitza el procés de proves i en el que es posa en pràctica MAQU, als següents apartats s'analitzen, documenten i avaluen les diferents fases desenvolupades a l'àrea de QA: Estudi preliminar, Pla de proves, Planificació de proves, Cicle de proves i Avaluació i millores detectades.

7.1 Aplicació Tempo Millenium

Actualment l'aplicació de Control Horari 'Tempo Millenium' consta de tres mòduls: Presencia, Accesos i Visites.

➤ *Control de Presencia*

Respon a la gestió d'entrades i sortides dels empleats a l'empresa. Els objectius poden ser diversos, des del control de saldo d'hores treballades fins a la gestió d'absentisme del personal.

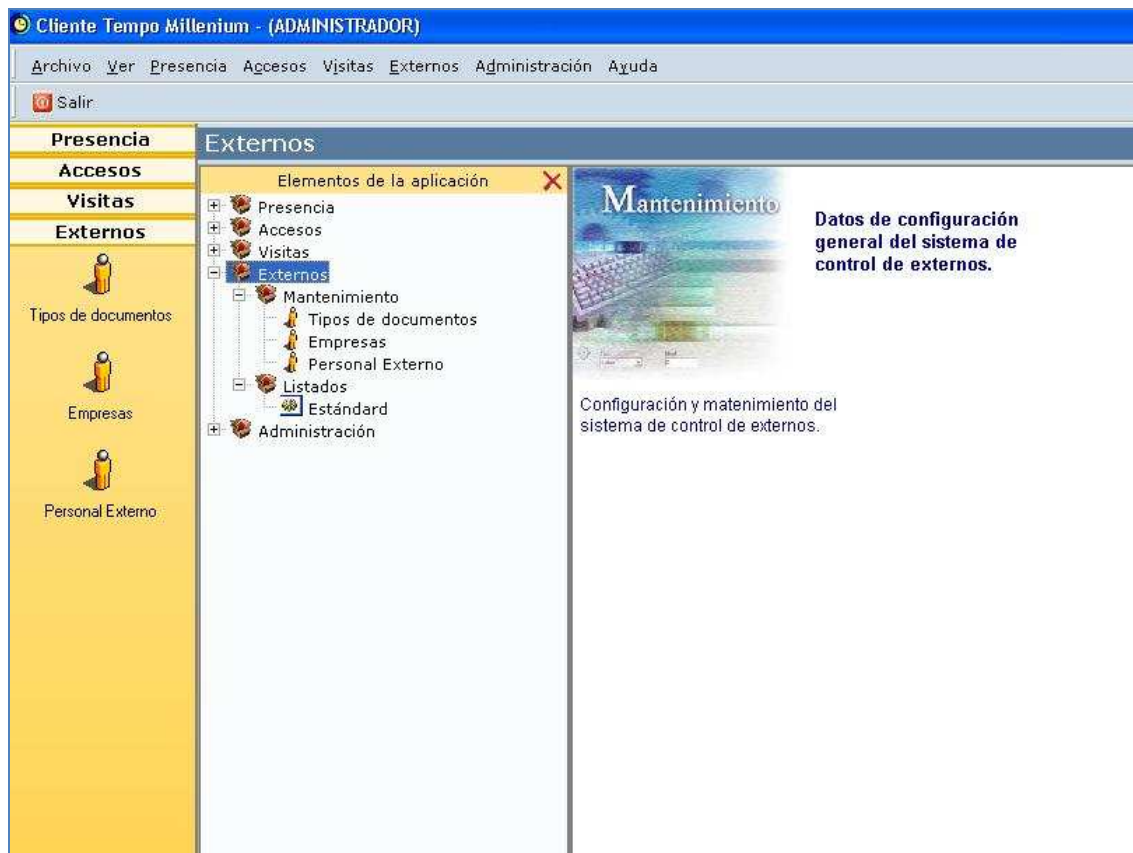
➤ *Control d'Accesos*

Basat en una aplicació i terminals de diferents tipus que permet restringir, autoritzar i el monitoratge de les entrades a àrees restringides de l'empresa.

➤ *Control de Visites*

Permet la gestió de les visites complint les lleis de riscos laborals.

Amb aquests tres mòduls l'aplicació és estable però el mercat cada cop exigeix més i la necessitat d'innovar i conservar la posició al mercat ha forçat a implementar un nou mòdul 'Personal Extern'. Aquest mòdul realitza tota la gestió de qualsevol persona externa que vagi a treballar a una empresa.



7.1.1 Mòdul Personal Extern

Fins ara l'aplicació Tempo Millenium contemplava dos tipus diferents de persones que podien accedir a l'empresa:

- ✓ *Empleats*: persones que treballen a l'empresa amb un contracte de treball i horari laboral.
- ✓ *Visites*: persones que realitzen visites a l'empresa amb la possibilitat de registrar l'entrada, la sortida, la localització de la visita dintre del recinte i la persona a visitar en el centre.

Faltava implementar un tipus de persones actualment bastant freqüents en les empreses, aquestes son les persones que son contractades per unes empreses i ofereixen els seus serveis a unes altres, a l'aplicació Tempo Millenium s'anomenen Persones Externes.

Les Persones Externes son totes aquelles persones autònoms, treballadors d'ETT, etc., que son contractats per prestar serveis i han de ser identificats, registrats i validats en el seu accés a l'empresa de forma similar que si es tractés d'un empleat habitual. Cada Extern quan arriba a una empresa a treballar està obligat a complir amb una sèrie de requisits per poder accedir al recinte per realitzar certes tasques.

Aquests requisits podrien ser Certificats per poder realitzar certes tasques, justificants de les empresa que els envia, etc., és a dir, una sèrie de documentació vigent i validada per poder accedir al centre per realitzar les feines corresponents.

En resum, la Persona Externa ha de seguir els següents passos:

1. Identificar-se amb les seves dades personals a la persona de Control (podria ser el vigilant de seguretat que es troba a l'entrada d'una empresa).
2. La persona de Control o encarregada de validar l'Extern demanarà la documentació necessària per deixar accedir a l'extern. En cas de ser un nou Extern se li obrirà una fitxa (amb la documentació corresponent que sol·licita l'aplicació).

Un cop a la fitxa de l'extern l'aplicació disposa d'uns semàfors que indiquen l'estat de la documentació de l'Extern:

- a. Semàfor en verd: indica que la persona te la documentació en regla i que pot passar sense problema.
- b. Semàfor en groc: indica que l'extern té documentació no entregada o caducada. En aquest cas la documentació de l'empresa de l'extern si està en regla (Hi ha dos tipus de documentació:
 - *Documentació de l'empresa de l'Extern* (documentació pactada amb l'empresa subcontractada i Recursos Humans de l'empresa que contracta els serveis)
 - *Documentació del Extern* (documents que porta la persona que es dirigeix a una empresa a realitzar tasques)

- c. Semàfor en vermell: indica que l'empresa de l'extern no te la documentació en regla, hi ha documentació no entregada o caducada.
3. Finalitzada la validació del Extern, l'empresa ja disposa de les dades personals i de la documentació en regla de la persona i se li ofereix una targeta per poder accedir a l'empresa per realitzar els serveis pactats.
4. Els moviments (fitxatges als terminals de l'empresa) tant siguin entrades com sortides (a departaments, a un menjador, etc.) es visualitzaran a la fitxa del Extern.

Aquest nou mòdul també disposa de llistats de manteniment (documentació incompleta, empreses, etc.) que mostren informació sobre la documentació que no està en regla, els moviments realitzats pels externs a l'empresa, etc.

A continuació la imatge de la pantalla principal del Personal Extern per entendre l'anàlisi de les proves realitzades:

7.2 Estudi preliminar

El responsable de projecte i dissenyador de proves es reuneixen per analitzar el Document de Requeriments i detectar les possibles desviacions. Encara no s'ha iniciat la implementació del mòdul per part de l'àrea de desenvolupament.

L'àrea QA disposa del temps per determinar l'abastament del projecte de proves i per realitzar l'anàlisi preliminar i Exploració general per alliberar la planificació del projecte de proves al responsable de projectes.

Les activitats en aquesta fase son les següents:

AR1+AD1. Abastament i fases+Exploració general

Es realitzen algunes reunions amb el responsable del projecte per establir l'abastament i fases del projecte de proves.

Es realitza una llista amb les funcionalitats del nou mòdul:

- ✓ Funcionalitat de les diferents pantalles.
- ✓ Menús dels que disposa el mòdul i funcionalitats de cada menú.
- ✓ Funcionalitats dels llistats del mòdul.
- ✓ Funcionalitats relacionades amb altres funcionalitats de l'aplicació Tempo Millenium.
- ✓ Funcionalitat de la gestió dels Externs.
- ✓ Funcionalitat d'Accessos a les instal·lacions.

AR2+AD2. Anàlisi preliminar+Anàlisi preliminar de les proves

Es realitza un anàlisi extens i detallat de les funcionalitats i es documenten aquelles incoherències, dubtes o desviacions no contemplades.

S'exposa a les reunions, tots els pactes i acords es modifiquen en el document de requeriments. L'equip QA no modifica el document de Requeriments, és tasca de l'equip de desenvolupament.

Inicialment es van detectar alguns aspectes no contemplats ràpidament es va valorar i modificar al document per contemplar-ho a l'equip de desenvolupament del producte.

AR3+AD3. Anàlisi de riscos+Anàlisi de riscos de les proves

L'especificació dels requeriments és bastant complert però quan es tracta d'una implementació tant extensa és necessària més d'una reunió per determinar l'anàlisi de les prioritats i riscos de l'aplicació.

Document 'Test Pla de Proves':

El responsable del projecte de proves redacta els acords respecte les dades generals del projecte i l'anàlisi i gestió de riscos determinats.

Les funcionalitats en les que es va detectar major risc van ser aquelles que es relacionen amb el semàfor 'Alarma' d'entrada a l'empresa, ja que determinen l'accés de l'Extern a les instal·lacions.

Les funcionalitats prioritàries eren aquelles que contemplen els moviments dels Externs, ja que han de tenir el mateix comportament que un empleat habitual. Els moviments detallen les entrades i sortides de les persones pels diferents accessos d'una empresa i és de vital importància que una empresa de Control horari no provoqui cap error en el sistema de fitxatge dels clients que ja treballen amb les nostres aplicacions o dels futurs clients.

Funcionalitats	Prioritat (1-3)	Riscos (0-2)	Observacions
Funcionalitat 1	1	1	
Funcionalitat 2	2	0	
Funcionalitat N	2	0	
Etc.			
Total funcionalitats prioritàries			
Total funcionalitats de risc			

7.3 Planificació de proves

AR4. Planificació del projecte.

El mòdul d'Externs és un projecte extens, això comporta realitzar una bona planificació per no provocar desviacions inesperades i la coordinació amb l'equip de desenvolupament ha de ser pactada i complerta per ambdues parts.

El responsable del projecte de proves realitza la planificació determinant amb el responsable d'àrea quines son les diferents entregues planificades des de l'àrea de desenvolupament. Aquestes entregues han de ser compatibles amb la planificació del projecte de proves i un cop definida la planificació i temps dels cicles de proves, funcionalitats a provar a cada cicle, anàlisi de les proves i eines i tècniques disponibles, etc. el responsable d'àrea procedeix a completar el 'Project' per incloure els temps i dates d'entregues de l'àrea de desenvolupament i l'àrea QA.

Es ideal que hi hagi una bona planificació en tot el departament, cadascú podrà realitzar les corresponents tasques, en les dates prefixades i de forma adequada i eficient però les dates no son inamovibles i en cas de detectar qualsevol desviació o inconvenient es comunica al responsable d'àrea en les reunions de seguiment setmanal a les quals assisteixen els responsables implicats de cada equip.

AR5. Anàlisi de Requeriments.

El responsable de proves analitza el document de requeriments per determinar cadascuna de les funcionalitats per generar el pla de proves més adequat.

Es determina les funcionalitats que es provarien amb prioritat i les de més risc per constatar amb el dissenyador l'ordre del disseny i execució de les proves i tècniques a utilitzar.

En la planificació van resultar:

- ✓ 5 funcionalitats de risc.
- ✓ 6 funcionalitats prioritàries.

- ✓ 21 funcionalitats restants, les quals son també dissenyades i exposades en els diferents cicles de proves per determinar l'ordre d'execució de les proves.

Les funcionalitats definides s'identifiquen i exposen en el Test Pla de Proves, juntament amb les tècniques, eines, estratègies, criteris, dates establertes dels cicles de proves i de report d'errors, etc.

AR6+AD4. Planificació del cicle de proves.

El responsable de proves i el dissenyador van definir, pactar i determinar els diferents cicles de proves coordinat amb les diferents entregues des de l'àrea de desenvolupament.

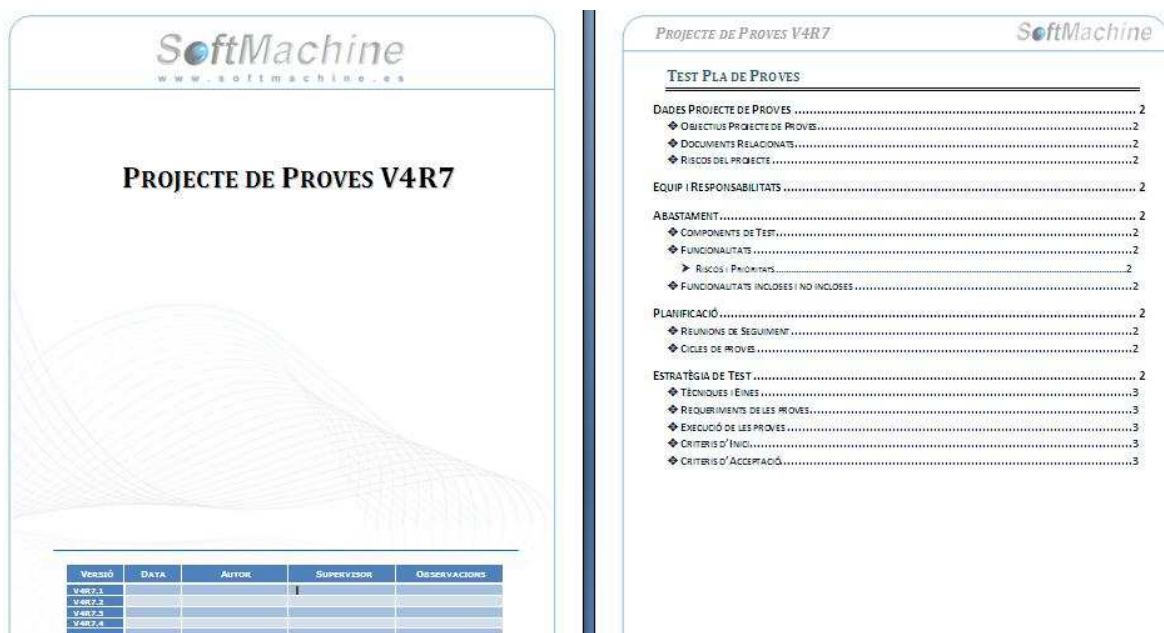
Cada cicle de proves disposa de les dates d'inici i fi, data de report d'errors, valoració dels temps per realitzar les execucions de les proves i les funcionalitats de cada cicle amb les prioritats i riscos establerts per cadascuna.

AD5. Definició de proves

Organitza com seran les proves i l'extensió dels diferents test de proves.

Documentació

El responsable del projecte de proves va documentar tota la informació pactada i planificada en el document 'Test Pla de Proves'. El document conté un apartat on detectar cadascuna de les modificacions realitzades i les dates en les que succeí el canvi respecte a la documentació anterior disponible a la primera pàgina del document per la permetre una ràpida visualització.



La planificació es va poder realitzar en uns 5 dies, dintre del termini inclòs a la planificació del projecte (realitzada amb MS Project). La documentació pertinent va

ser respectada i publicada per la resta d' implicats en el projecte i cadascú va procedir a realitzar les seves tasques a partir de la documentació alliberada.

Disposar d'aquesta informació va permetre organitzar i planificar les tasques per cadascuna de les parts i tenir una visió global i detallada del projecte amb la informació necessària recollida i endreçada en un mateix document.

7.4 Pla de proves

AD6. Administració de proves

Un cop definit i analitzat el Test Pla de proves, es realitzen reunions amb el tester per entendre les funcionalitats a testejar i traspasar la informació pactada en quan a planificacions realitzades. En el cas d'observar algun inconvenient en aquestes reunions es transmet al responsable del projecte de proves.

A la primera reunió que es realitzà no va sorgir cap problema, hi havia algunes funcionalitats del nou mòdul que els testers van analitzar i van realitzar les preguntes oportunes per entendre correctament el comportament.

El pla de proves pretén entendre:

- ✓ L'objectiu de les proves.
- ✓ L'abastament del projecte de proves i de cadascuna de les funcionalitats.
- ✓ Planificar i optimitzar els temps de proves.
- ✓ Respectar els recursos disponibles, aplicar les tècniques i mètodes adequats.
- ✓ Analitzar els costos i riscos detectats al producte a provar i alliberar respectant els estàndards de qualitat desitjada.

AD7. Disseny de les proves

El dissenyador comença a redactar el document 'Test Disseny' previ a disposar de l'executable del producte i degut a l'extensió del projecte primer va començar per les funcionalitats de major risc i les prioritàries i més tard va completar amb els test d'exploració, funcionals i de combinacions.

Les dates del disseny de proves van ser ajustades, es va determinar una durada aproximada de 2 setmanes i alguns tests de combinacions es van seguir realitzant quan el tester ja disposava de la primera entrega per part de desenvolupament.

Les desviacions i incidents detectats foren anotats al Test Pla de Proves per poder determinar la qualitat del disseny de proves i avaluar les millores per posteriors versions.

Els casos de prova dissenyats contemplen:

- ✓ Resultats que assegurin la qualitat del producte.
- ✓ Garanteixen el compliment de les normatives legals.

- ✓ Tècniques analitzades i controlades per cada cas de prova.
- ✓ Descripció del seguiment a realitzar de cada funcionalitat.
- ✓ Passos a seguir en el cas d'obtenir resultats no vàlids.
- ✓ Entrades, sortides i procediment de cada prova.
- ✓ Comportaments i normatives estàndards de l'aplicació. En cas de descriure proves assumint un comportament es documentat per informar al tester.
- ✓ Normes de qualitat en les que es basa la metodologia: Fàcil manteniment, funcionalitat, portable, confiança, eficiència, facilitat d'ús, etc.

Els casos de prova de les diferents funcionalitats es dissenya al document Test Disseny documentat a continuació:

Presentació del document:

Es defineix al primer full del document i presenta l'identificador i data del document amb un historial de les modificacions realitzades i pactades i les característiques principals de la versió a testear.

Test d'exploració:

Es defineix al segon full del document i s'especifica cadascuna de les funcionalitats, amb les accions a realitzar, els resultats esperats i una columna per validar el comportament real.

En el mòdul d'Externs s'ha analitzat les diferents funcionalitats i entre d'altres:

- ✓ Interfase d'usuari: Tipus de documents, Empreses, Personal Extern, Llistats i Registre d'Auditoria.
- ✓ Visualització de les pantalles.
- ✓ Manteniment de Pantalles: Filtres, buscadors, desplegable, menús, disposició de les dades, etc.
- ✓ Funcionalitat dels Llistats i execució.
- ✓ Funcionalitats de risc i prioritàries: Gestió d'alarmes, Control d'accés, etc.

Test específic o funcional:

Conté el detall de les funcionalitats determinades exposades en un mapa de proves, les accions a realitzar en cadascuna de les proves, els valors esperats i una columna per validar el comportament real al executar les proves.

En el mòdul d'Externs es detalla les diferents funcionalitats amb l'objectiu i valor esperat de cadascuna d'elles. No es realitzen totes les accions només de validació que l'aplicació funciona com s'espera que funcioni.

Test de combinacions:

Es detalla totes les combinacions possibles respecte les funcionalitats definides.

En el mòdul d'Extern:

- ✓ Interfase d'usuari: accés a les pantalles des dels diferents menús de l'aplicació. Accessos directes, menús d'eines i accés als elements de l'aplicació.
- ✓ Manteniment de pantalles: comportament desplegable, botons, Filtres, Flags, etc. i les accions possibles: 'Guardar', 'Nuevo', 'Nuevo Como', 'Cancelar', 'Salir', etc. realitzades des del menú, tecles funció o icones.
- ✓ Visualització, ordenació i comportament de dades d'entrada en les diferents pantalles.
- ✓ Missatges erronis o carència de missatges.
- ✓ Per cada funcionalitat les seves casuístiques mitjançant tècniques de caixa negra: valors límit, partició d'equivalència, etc.

1	Composante	Versión documento	Combinación Presentado	Combinación Caducidad	Visualización pantalla	OKKO	Valor real	Valor esperado
2	Externo		Presentado Empresa	Presentado Externo	Caducidad Empresa	Caducidad Externo	Datos	
3	Tipo documento	Fecha de creación	Of	Of	Of	Of	Of	Of
4	Editable							
5	Requerido							
6	Caducos							
7	Editable y Requerido							
8	Editable, Requerido y Caducos							
9	Requerido y Caducos							
10	Editable y Caducos							
11	Editable y Caducos							
12	Editable y Caducos							
13	Editable y Caducos							
14	Editable y Caducos							
15	Editable y Caducos							
16	Editable y Caducos							
17	Editable y Caducos							
18	Editable y Caducos							
19	Editable y Caducos							
20	Editable y Caducos							
21	Editable y Caducos							
22	Editable y Caducos							
23	Editable y Caducos							
24	Editable y Caducos							
25	Editable y Caducos							
26	Editable y Caducos							
27	Editable y Caducos							
28	Editable y Caducos							
29	Editable y Caducos							
30	Editable y Caducos							
31	Editable y Caducos							
32	Editable y Caducos							
33	Editable y Caducos							
34	Editable y Caducos							
35	Editable y Caducos							
36	Editable y Caducos							
37	Editable y Caducos							
38	Editable y Caducos							
39	Editable y Caducos							
40	Editable y Caducos							
41	Editable y Caducos							
42	Editable y Caducos							
43	Editable y Caducos							
44	Editable y Caducos							
45	Editable y Caducos							
46	Editable y Caducos							
47	Editable y Caducos							
48	Editable y Caducos							
49	Editable y Caducos							
50	Editable y Caducos							
51	Editable y Caducos							
52	Editable y Caducos							
53	Editable y Caducos							
54	Editable y Caducos							
55	Editable y Caducos							
56	Editable y Caducos							
57	Editable y Caducos							
58	Editable y Caducos							
59	Editable y Caducos							
60	Editable y Caducos							
61	Editable y Caducos							
62	Editable y Caducos							
63	Editable y Caducos							
64	Editable y Caducos							
65	Editable y Caducos							
66	Editable y Caducos							
67	Editable y Caducos							
68	Editable y Caducos							
69	Editable y Caducos							
70	Editable y Caducos							
71	Editable y Caducos							
72	Editable y Caducos							
73	Editable y Caducos							
74	Editable y Caducos							
75	Editable y Caducos							
76	Editable y Caducos							
77	Editable y Caducos							
78	Editable y Caducos							
79	Editable y Caducos							
80	Editable y Caducos							
81	Editable y Caducos							
82	Editable y Caducos							
83	Editable y Caducos							
84	Editable y Caducos							
85	Editable y Caducos							
86	Editable y Caducos							
87	Editable y Caducos							
88	Editable y Caducos							
89	Editable y Caducos							
90	Editable y Caducos							
91	Editable y Caducos							
92	Editable y Caducos							
93	Editable y Caducos							
94	Editable y Caducos							
95	Editable y Caducos							
96	Editable y Caducos							
97	Editable y Caducos							
98	Editable y Caducos							
99	Editable y Caducos							
100	Editable y Caducos							

AR7. Seguiment de fases

El responsable del projecte de proves ja realitza altres tasques d'altres projectes i realitza seguiments sobre l'estat del projecte. Es manté informat seguint la metodologia i estratègies definides al Test pla de proves.

AR8. Avaluació de proves

El responsable del projecte de proves avalua les proves definides pel dissenyador de proves, decideix si és necessari definir més proves o descartar algunes proposades pel dissenyador.

Algunes proves no van ser contemplades en un inici degut a una carència en el document de Requeriments. Aquesta desviació va ser ràpidament corregida i es van contemplar els casos de prova corresponents.

AT1. Organització cicle de proves

El tester disposa del Document de Requeriments, document Test Pla de proves i del Test disseny per començar amb l'anàlisi i estudi previ a disposar del primer executable del producte a testejar.

Es van realitzar algunes reunions amb el dissenyador de proves i es van contemplar els tipus de proves i tècniques utilitzades pel desenvolupament de les proves.

El tester es va organitzar la documentació i en quan rebés la primera entrega alliberada, QA ja disposava de la informació necessària per començar amb el primer Cicle de proves.

7.5 Cicle de proves

Els cicles de proves planificats es poden iniciar i ja es disposa del disseny de les proves per procedir a l'execució de les mateixes.

Cicle de proves 1:

L'àrea de desenvolupament ha alliberat la Entrega 1 a l'equip QA i es pot començar el procés de proves.

AT2. Configuració de l'entorn

Tempo Millenium disposa d'uns requisits tècnics a complir pels usuaris finals i és en aquest entorn que QA ha d'executar els casos de proves.

L'equip QA disposa d'un servidor propi amb màquines virtuals preparades i disponibles en diferents entorns hardware i software.

Tant el Servidor d'Aplicacions com el servidor de dades tenen requisits mínims de hardware i software per les instal·lacions del producte i en base a aquests es configura l'entorn el tester segons el detall del Test Pla de Proves.

En la configuració de l'entorn es procedeix a instal·lar l'Entrega 1 en els diferents entorns planificats i amb les bases de dades pertinents.

En el cas de detectar inconvenients es bloqueja el test i es procedeix reportar l'incident a solucionar, no es podria continuar amb el cicle de proves.

AT3. Test d'exploració

Un cop configurat l'entorn i instal·lat s'executa el Test d'Exploració.

En cas d'esdevenir defectuós es reporta els incidents i es valora la continuació del procés de proves.

El test d'exploració del mòdul d'Externs va aportar alguna modificació al Document d'especificacions de Requeriments i al Test Pla de Proves. Un cop contemplat es continua el procés de proves amb els següents tests a executar.

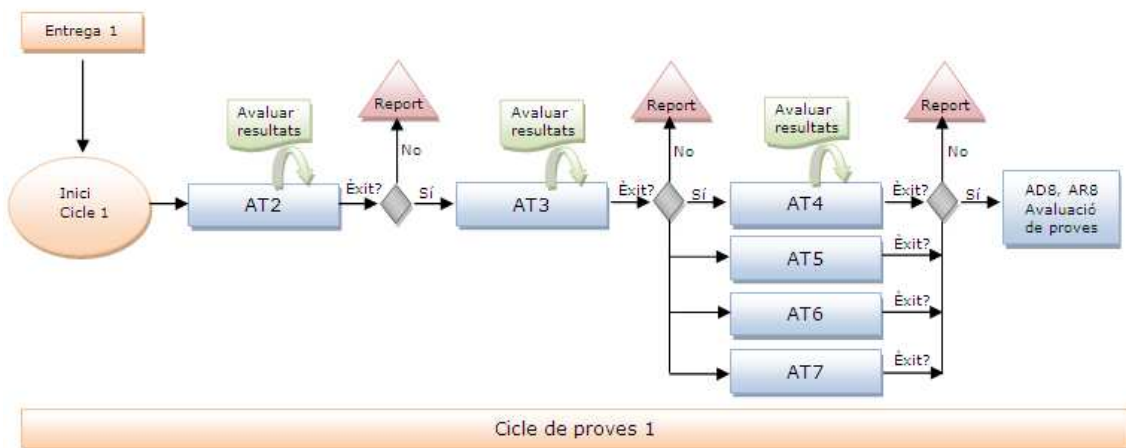
AT4. Test específic (funcional), AT5. Test de combinacions , AT7. Validació i verificació de les proves

El tester comença l'execució del test d'exploració i segueix amb la resta de tests dissenyats i verifica el contingut de l'aplicació respecte al comportament esperat.

El cicle de proves 1 conté unes funcionalitats prèviament definides i són aquestes les que el tester ha d'executar i no unes altres.

Els incidents localitzats es reporten en el moment a la eina d'errors 'Bugzilla' i fins que no arribi la data de report d'errors es mantenen a l'àrea de QA impossibilitant que cap component de l'equip de desenvolupament pugui visualitzar els defectes de l'aplicació. Els responsables d'àrea realitzen un seguiment dels incidents i defectes detectats.

L'eina de report d'errors permet visualitzar la quantitat de defectes reportats i en el document d'avaluació es valoren els defectes segons el valor crític i per cicles de prova.



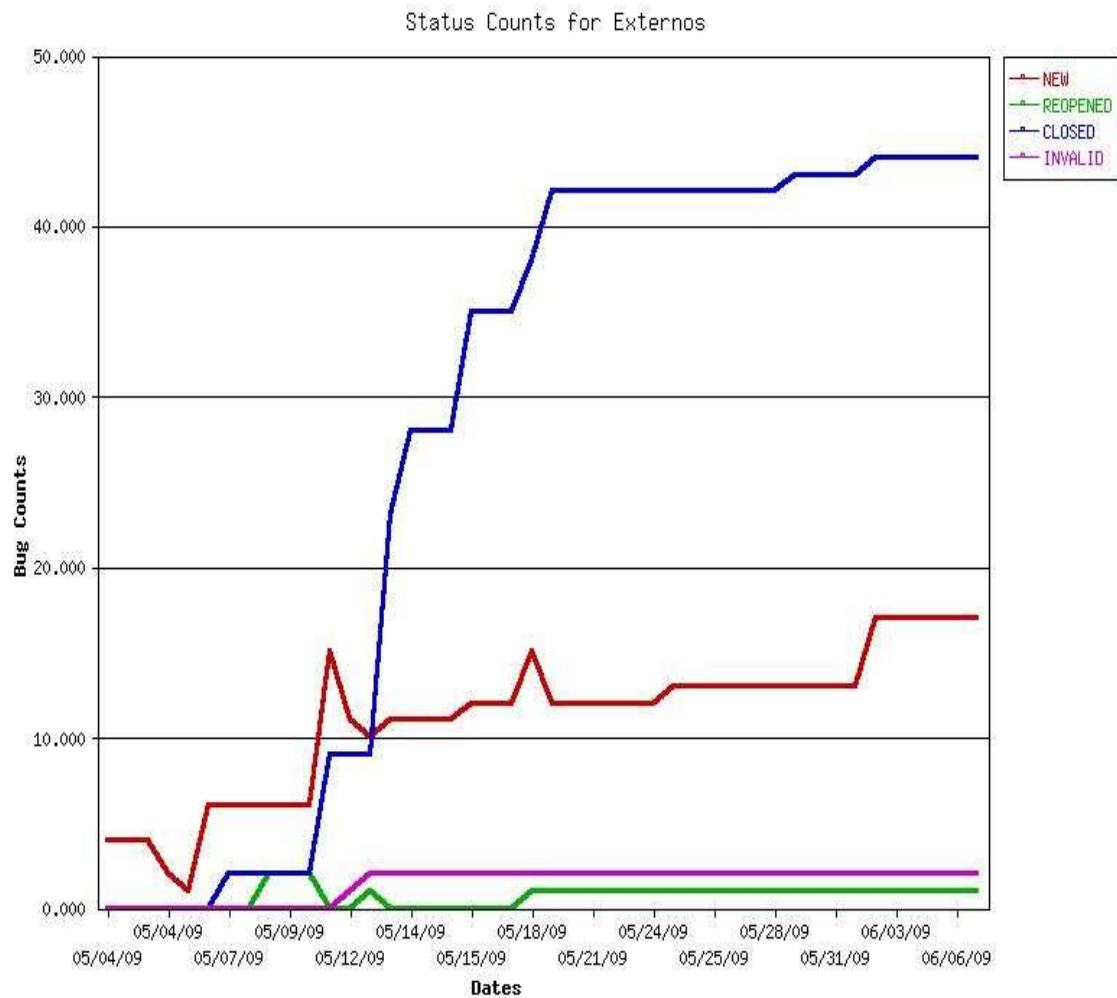
Cicles de proves posteriors:

Els següents cicles de proves s'executen de la mateixa manera. Es reporten els defectes en les dates establertes, l'àrea de desenvolupament ha d'entregar en els terminis corresponents i la solució dels errors reportats són pactats en quina entrega seran alliberats depenent de les funcionalitats de cada cicle de proves.

La prèvia planificació dels cicles de proves ha permès l'organització d'execució de les proves i un seguiment adequat dels incidents detectats.

Al llarg dels cicles de proves s'han detectat accions d'un valor bloquejant i crític, aquestes són les que provoquen alteracions més importants en el cicle de proves, provocant en alguns dels cicles problemes per finalitzar l'execució de les proves abans de la següent entrega des de l'àrea de desenvolupament.

L'obtenció de mètriques dels diferents cicles de proves permet identificar els defectes que han provocat dites accions i es recullen al document d'avaluació per definir les millores per posteriors versions.



Com s'observa en el gràfic es pot extraure la informació dels errors en els diferents estats en els que es pot assignar un bug o defecte.

AT6. Report d'errors

El report d'errors es realitza mitjançant l'eina de seguiment d'errors i el procés de seguiment es defineix i respecta pels equips per dur a terme un enteniment i metodologia comú entre ambdós equips que gestionen, localitzen i intervenen a l'eina implantada.

El document de 'Report d'errors' és públic per tot el departament i especifica el procés de seguiment d'errors:

BUG TRACKING SYSTEM		SoftMachine
ÍNDICE		
ÍNDICE		1
OBJETIVO		3
BUGS		3
➤ CAMPOS		3
▪ Product s y Components		3
▪ Status y Resolution		3
▪ Assigned To		3
▪ Summary		3
▪ Platform y OS		3
▪ Version		3
▪ Priority		4
▪ Severity		4
▪ Target (Target Milestone)		4
▪ Reporter		4
▪ Time Tracking		4
▪ Attachments		4
▪ Dependenc ies		4
▪ Comments		4
➤ CAMPOS PERSONALIZADOS		5
▪ Responsable		5
▪ Confirmado		5
▪ Reportado Por		5
CILCO DE VIDA		6
➤ ESTADO		6
➤ SOLUCIÓN		7
➤ SEVERIDAD		8
➤ COMENTARIOS		8
SEGURIDAD		8
➤ USUARIOS		8
➤ GRUPOS		9

El report d'errors s'organitza en les diferents entregues identificades a través de l'eina de seguiment d'errors i la informació necessària per establir mètriques alhora de realitzar l'estudi dels defectes provocats a la versió també s'extrau de l'eina de seguiment d'errors.

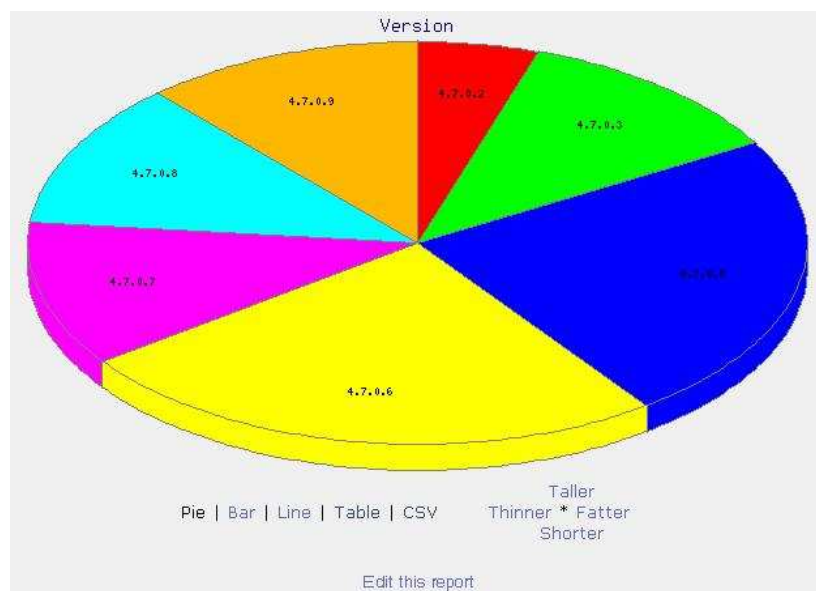
La informació es pot organitzar per determinar els defectes localitzats per entregues a Desenvolupament o Cicles de proves a QA. L'eina reflexa la versió i l'entrega en la que es detecta els errors i la versió i entrega en la que es solucionen mostrant l'històric sobre el defecte i tota la documentació, informació o comentaris realitzats per les persones implicades en el projecte.

El procés de seguiment en l'eina d'errors també es establert, documentat, provat i analitzat pel departament QA. El mètode a seguir és entès i respectat per ambdós equips: àrea de desenvolupament i àrea QA.

Com exemple, al mòdul d'Externs s'obtenen les següents dades entre l'entrega 2 i l'entrega 9 respecte al número de defectes detectats:

Reports Preferences Administration Help Log out vahessa	
Version	4.7.0.2 3
	4.7.0.3 7
	4.7.0.5 14
	4.7.0.6 15
	4.7.0.7 7
	4.7.0.8 7
	4.7.0.9 7
	Total 60
Pie Bar Line Table CSV	

L'eina de seguiment d'errors disposa de gràfiques per interpretar mètriques des de diferents punts de vista segons l'estratègia apropiada, com exemple les dues imatges següents:





La prioritat i severitat dels defectes es parametritza mitjançant altres gràfiques i anàlisi de dades, com exemple la imatge següent on s'exposa la prioritat (P1, P2 i P3) de cadascun dels components i la severitat en la que van ser classificats.

P1					
Component					
	Empresas	Listados	Personal externo	Tipos de documentos	Total
bloqueante	.	.	1	.	1
critica	.	.	.	.	0
normal	.	.	.	.	0
mejora	.	.	.	.	0
Total	0	0	1	0	1

P3					
Component					
	Empresas	Listados	Personal externo	Tipos de documentos	Total
bloqueante	.	.	.	.	0
critica	1	.	3	1	5
normal	1	2	4	2	9
mejora	2	.	.	.	2
Total	4	2	7	3	16

AR7. Seguiment de fases

El responsable del projecte de proves es informat de les desviacions i defectes localitzats, mitjançant les reunions de seguiment setmanals.

Els acords o conclusions exposades es documenten en 'Actes' per no perdre detall de les decisions.

AR8+AD8. Avaluació de proves

El responsable de proves i el dissenyador de proves avaluen les proves després de finalitzar cada cicle de proves.

Els defectes, millores o desviacions queden pactades en el document d'Avaluació.

AT8. Documentació interna i manuals d'usuari

Un cop alliberada la versió es publica la documentació interna i manuals d'usuari. És responsabilitat del tester sota la supervisió del responsable del projecte de proves, alliberar aquesta documentació en els formats i terminis establerts.

AT9. Formacions

Les formacions de les versions es pacten amb els responsables dels departaments implicats un cop alliberada la versió.

En el 'Project' inicial ja es va pactar aquestes dates per poder planificar futures instal·lacions a clients.

7.6 Avaluació i millores

AR9. Avaluació de la qualitat, AT10. Avaluació i millores

El document 'Pla de millora' recull les accions i conclusions de la versió provada.

Aquest document contempla accions preventives i correctores com a conseqüència de les desviacions trobades en el projecte, plans d'actuació que afecten a la qualitat del sistema, valoració dels objectius aconseguits i un resum de les mètriques de qualitat obtingudes al finalitzar el projecte.

El procés de proves del mòdul d'Externs ha patit algunes desviacions provocades per:

- ✓ S'ha implementat alguns processos que no estaven clarament definits.
- ✓ S'ha produït alguns canvis funcionals que impactaven directament sobre la qualitat del producte.
- ✓ S'ha implantat algunes proves inestables que han provocat regressions inesperades.

A partir de les dades obtingudes de les desviacions es pot estimar i valorar plans d'actuació pel millorament de les planificacions, optimització dels temps, control de les activitats i de la qualitat dels productes amb l'objectiu de millorar les estimacions en les futures versions.

El disseny de les proves es el primer objectiu a millorar començant per analitzar i estudiar els automatismes actuals disponibles al mercat.

El pla de millora d'aquesta versió testejada contempla l'automatització de proves per implantar a l'àrea QA.

8 Conclusions del projecte

En el moment d'iniciar el projecte la documentació de la qual disposava el departament QA no es trobava massa organitzada, QA disposava d'un espai d'emmagatzematge de documents amb informació de moltes versions de diverses aplicacions però la pèrdua de temps que provoca no tenir la informació ben documentada i estructurada suposa un estat caòtic alhora de planificar els projectes i assegurar la qualitat dels productes amb un historial de versions difícilment localitzable i amb documents dispersos amb diversos formats.

Alhora de iniciar un projecte de proves és imprescindible disposar de fonts d'informació i documents actualitzats per dur a terme amb èxit el projecte des del primer moment i durant el seguiment del mateix sense perdre qualitat del producte per no tenir localitzable la informació necessària o no tenir documentades les decisions pactades, etc.

Per això és important una metodologia que en tot moment indiqui quin és l'estat del projecte, com s'ha arribat a qualsevol punt i qui realitza cadascuna de les tasques. Treballant en equip, de forma planificada, amb els mètodes més efectius i adequats en els diferents projectes garanteix 'l'assegurament de la qualitat' dels productes a alliberar al mercat.

La metodologia MAQU destinada al departament QA ha proporcionat la reorganització de molta informació, ha proporcionat un enfocament pràctic, sistemàtic i quantitatiu per l'avaluació de la qualitat funcional dels productes.

L'estudi dels models, mètodes i eines d'Enginyeria de software per l'assegurament de la qualitat ha afavorit no només a la comprensió i anàlisi del software, si no que potencia la millora de la qualitat amb la que es desitja alliberar un producte. Tot i haver de destacar que s'han definit línies futures a implementar per facilitar les tasques de l'equip QA entorn al desenvolupament i execució de les proves.

MAQU garanteix una bona planificació d'un projecte de proves amb uns adequats i previstos resultats per satisfer als clients i a la organització, ja que garanteix resultats documentats, validats i verificats al llarg del cicle de vida del software evitant possibles desviacions o detectant els màxims errors per no alliberar-los a l'usuari final provocant sèries pèrdues i objectius no complerts.

Al llarg del projecte s'ha modificat documentació, s'ha creat informes estàndards i els mètodes necessaris per dur a terme qualsevol procés de proves dels productes a testear i s'ha implantat una eina d'errors, que actualment es troba en funcionament per tot el departament amb la possibilitat de saber en tot moment i per qualsevol dels implicats del projecte l'estat en el que es troba un projecte.

La implantació de la metodologia ha sigut acceptada per l'organització i és respectada i compatible amb la metodologia de desenvolupament del software. El més important a tenir en compte és que MAQU està obert a millores i en creixement constant i amb possibilitat de modificar les activitats necessàries definides en el procés de proves.

MAQU neix per establir mètodes de treball a l'àrea QA i està sent provada en els projectes i els resultats van demostrant i avaluant quines son les eines i mètodes a modificar o potenciar analitzant i estudiant constantment tècniques i mecanismes de millora per l'assegurament de la qualitat dels productes de Softmachine.

Situació Inicial

- ✓ QA sense metodologia definida.
- ✓ Descoordinació entre equip de desenvolupadors i test.
- ✓ Desenvolupadors finalitzen el desenvolupament i test comença les tasques: disseny de les proves, estratègies, execució de les proves.. Conseqüències caòtiques. Falta de temps i recursos.
- ✓ Sobrecàrrega de treball per testear.
- ✓ Falta d'informació en modificacions funcionals.
- ✓ Poca profunditat en les proves.
- ✓ Constants canvis per defectes localitzats.
- ✓ Desenvolupadors distant de l'equip de Test.
- ✓ Major cost en report d'errors i reproducció dels mateixos.
- ✓ No es consideren els impactes en les regressions al fer canvis als productes.
- ✓ Problemes en grup de proves = problemes de qualitat.

Situació Actual

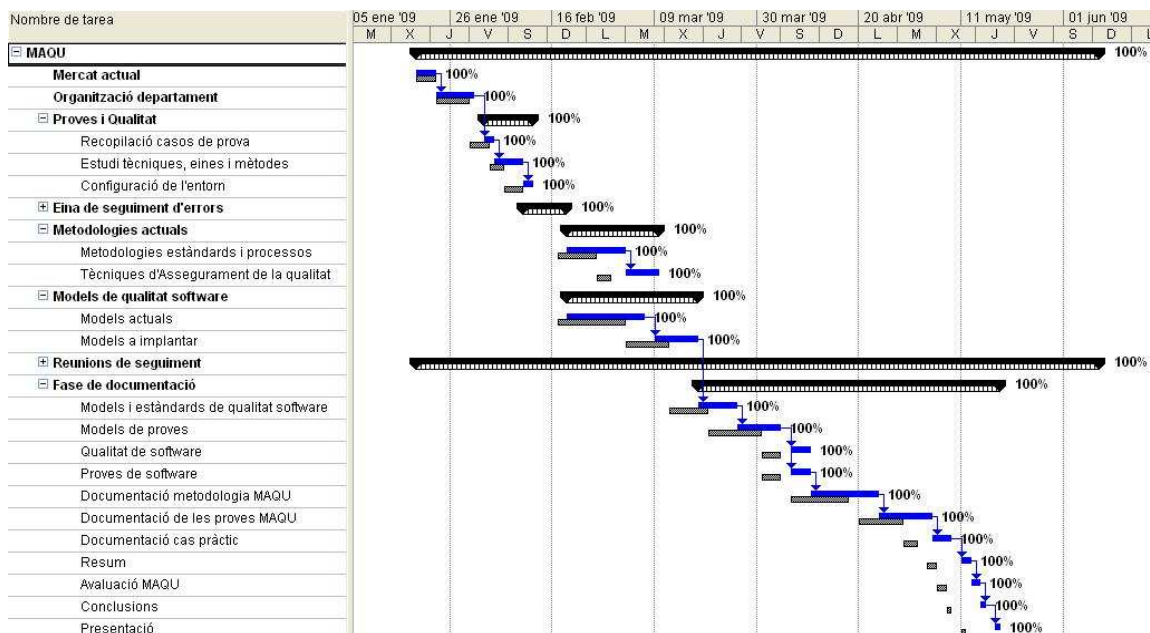
- ✓ Coordinació i planificació amb l'equip de desenvolupadors.
- ✓ Compatibilitat entre equips i sincronització adequada.
- ✓ Paral·lelisme del desenvolupament i el procés de proves.
- ✓ Visibilitat de l'estat del producte.
- ✓ Entregues periòdiques organitzades en cicles de proves.
- ✓ Definició i etapes predefinides en el procés de proves.
- ✓ Documentació, comunicació i compromís per part dels equips.
- ✓ Rols i responsabilitats al departament de QA.
- ✓ Report de defectes amb una eina d'errors amb control de versions.
- ✓ Facilitat de seguiment del projecte de proves i estat del procés de proves en les diferents etapes del desenvolupament del software.

8.1 Seguiment del projecte

Inicialment el projecte es va planificar per un termini de 250h i finalment s'ha obtingut 274h de treball. S'ha pogut implantar la metodologia i el cas pràctic però hi ha alguns aspectes que han provocat desviacions:

- ✓ Adaptació de l'entorn i eines a la metodologia a implementar i dissenyar.
- ✓ Configuració de nous mecanismes i eines al departament.
- ✓ Reunir i organitzar la documentació del departament.
- ✓ Prendre les decisions oportunes en les diferents reunions i adaptar-les a projecte.
- ✓ Estudi d'estratègies i mètodes de proves i d'assegurament de la qualitat.
- ✓ S'han realitzat continues revisions de la metodologia, dels processos i activitats definides al departament.
- ✓ Els terminis d'implantar MAQU en uns dels projectes reals de l'empresa han sigut ajustats.
- ✓ Recopilació de la informació per desenvolupar cadascun dels punts exposats al projecte.

A continuació es mostra el **Gantt de seguiment** que determina el temps finalment invertit al projecte:



El Gantt de seguiment respecte la planificació inicial mostra les desviacions que s'han produït. Per la quantitat de documentació recopilada, l'experiència adoptada i el seguiment constant del projecte les desviacions provocades a les diferents tasques no han resultat determinants per entregar el projecte finalitzat i en els períodes establerts, podria dir que els objectius han sigut assolits.

8.2 Línies futures

L'avaluació de MAQU ha deixat obert alguns punts els quals s'implementaran per facilitar les tasques de l'equip de qualitat i que s'aniran millorant amb eines per l'entorn de proves.

- Millora de la documentació.
- Automatismes:
 - *Casos de prova*: automatisme per crear casos de prova, introduint les dades necessàries dels casos de prova es generen els tests funcionals o combinatoris adequats.

S'estan estudiant possibles eines que garanteixen un bon disseny dels casos de prova però les aplicacions d'aquesta organització son massa complexes en quan a càlcul i estructura que s'està analitzant pas a pas la millor eina per la gestió de les proves.

- *Entorn de proves*: automatisme per preparar l'entorn de proves amb dades importades d'altres bases de dades. Ja s'ha proporcionat una eina per transformar les dades de les bases de dades però no disposa de les dades a importar havent de realitzar un procés manual de les dades a introduir.
- *Exportació de dades*: Els resultats obtinguts als tests exportats al documents que contenen la informació d'avaluació de les execucions. Actualment existeixen macros i demés mecanismes però es necessita un automatisme per facilitar les tasques als testers.
- *Comptador de temps en les execucions*: automatisme per comptar el temps de execució de les proves i saber així la planificació dels casos de prova amb més exactitud.

9 Bibliografia

- ➔ [CHSUM] The Standish Group, Chaos Summary 2008, 'The 10 Laws for Chaos', 1998 to 2006 Project Resolution.
- ➔ [GERPRO] Daniel Piorun. Gerencia de Proyectos. 20.01.2003. [En línea]. Article sobre el fracàs dels projectes.
http://www.degerencia.com/articulo/por_que_fracasan_los_proyectos
- ➔ [McCall], [ISO9126], [BOEHM] La calidad del software y su medida. Más allá del Análisis del Punto Función tradicional, ISO/IEC 9126. Jesús María Minguet Melián, Minguet, Juan Francisco Hernández Ballesteros, Hernández. Editorial Ramón Areces, 2003 ISBN8480046112.
- ➔ [ISHUM] Ingeniería del Software 7ª ed. Ian Sommerville, editorial Addison Wesley 2005, ISBN 84-7829-074-5.
- ➔ [BUGZ] Web oficial de la eina de seguiment d'errors, característiques, objectius, documentació i ajuda. <http://www.bugzilla.org/>

Llibres

- ➔ Test Process Improvement: a Practical Step by Step Guide to Structured Testing, editorial Addison Wesley Longman, 1999. ISBN 0-201-59624-5.
- ➔ Técnicas Cuantitativas para la gestión en la Ingeniería del Software. Autor Javier Dolado Cosín, editores Javier Tuya, Isabel Ramos y Javier Dolado. Editorial Netbiblo, 2007. ISBN 978-84-9745-204-5.
- ➔ Gestión Integral de la Calidad: Implantación, control y Certificación. Lluís Cuatrecasas Arbós. Edición: 2, Publicat por Gestión 2000, 2001. ISBN 84-8088-282-4.
- ➔ Gestión de la calidad total. Ediciones Díaz de Santos S.A., Mapcal, 1996. ISBN 8479782447, 9788479782443.
- ➔ Ingeniería del software: un enfoque práctico. Cuarta edición. Roger S. Pressman; tr. Rafael Ojeda Martín; Joaquín Sánchez Villoria; Virgilio Yagüe Galaup; Julio Zurdo Chávez; -- 4a. ed. en Español. Madrid [ES]: McGraw-Hill, 1998. ISBN: 84-481-1186-9. (PRESS98)

Enllaços relacionats

- ➔ Capability Maturity Model® Integration (CMMISM), Version 1.1 CMMISM for Systems Engineering and Software Engineering (CMMI-SE/SW, V1.1) Staged Representation CMU/SEI-2002-TR-002 ESC-TR-2002-002. December 2001 [En línia] <http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr002.pdf>
- ➔ Sogeti. Compañía de servicios IT y consultoría a nivell mundial. Empresa fundada por Serge Kampf en Grenoble (Francia) en 1967 y liderado por el CEO Paul Hermelin desde diciembre de 2001, Disponible documentació sobre TMAP i TPI ja que Sogeti és l'empresa fundadora dels models. [En línia] <http://www.es.sogeti.com/>
- ➔ Documentació de l'empresa Sogeti, Software Control & Testing. Expo:QA 26 de Novembre 2007 – Madrid: Verificación, Validación y pruebas del software, autora Patricia Rodríguez Dapena.
- ➔ Q-Course Quality & Organization: The Structured Testing Process (TMM, TPI, TMAP), autor Bob Legrand, edición 2, 2004, ISBN1411610202.
- ➔ Asistencia i documentació expo:QA Session a la jornada 'Regression Testing Day' amb tallers Hand On el dia 17 de Juny 2009 sobre automatitzacions de proves de Software. Jornades impartides per professionals en l'àrea de Gestió de Qualitat Software. <http://www.expoqa.com/sessions.html>
Sogeti realitza una presentació sobre metodologia d'automatització de proves, assoliments, problemes, tòpics i ROI (*Return On Investment*. Valor que mesura el rendiment d'una inversió, per avaluar l'eficiència d'un gasto que es realitza) que es pot esperar de l'automatització.
S'imparteixen tallers tècnics de HP (Quick Test Pro), IBM (Rational Functional Tester), Borland i Oracle on es poden provar i veure les eines líders del mercat.
- ➔ Adler P.S.: Practice and Process: The Socialization of Software Development. Paul S. Adler Marshall School of Business University of Southern California. Los Angeles CA 90089-0808 [En línia]. <http://www.si.umich.edu/ICOS/adlerpaper.pdf> (2003)
- ➔ ASQ Standards Group: ISO 9000:2000 Product Support Initiative [En línia]. http://qualitypress.asq.org/iso9000/ISO_Curves.pdf (2004)
- ➔ Rico D. F. Using Cost Benefit Analyses to Develop Software Process Improvement (SPI) Strategies [DACS Technical Report]. Rome, NY: Air Force Research Laboratory/Information Directorate (AFRL/IF), Data and Analysis Center for Software (DACS). (2000) <http://www.dacs.dtic.mil/techs/abstracts/rico.html>
- ➔ Evolución histórica de la calidad. Prof. Dr. Ing. Arturo Luis Romero y Lic. Sandor Luis Miranda. 10.08.2007 [En línia] <http://www.gestiopolis.com/administracion-estrategia/la-calidad-historia-conceptos-y-terminos-asociados.htm>
- ➔ TMMI Foundation. Reference Model version 2.0 released including Levels 2 and 3. Fundació TMMI i lloc web sense ànim de lucre per la millora de les organitzacions amb un enfocament comú [En línia]. <http://www.tmmifoundation.org/html/tmmiref.html>

- Connexions® is supported by the William and Flora Hewlett Foundation. Connexions® is a registered trademark of Rice University.
Web Connexions. Lloc Web per veure i compartir material educatiu organitzat en mòduls de coneixements que es poden organitzar com cursos, llibres, informes, etc. [En línia].
<http://cnx.org/content/m14738/latest/>, <http://cnx.org/content/m17461/latest/>
- Software Quality Assurance. James Back dedicat a la docència i consultoria de proves de software y asegurament de la qualitat és fundador de Satisfice. La Web oficial conté informació i articles sobre test de software. [En línia]
<http://www.satisfice.com/articles/et-article.pdf>
- Investigación en TBC Universidad, C.P. Leonel Chow Mak, Director LCEA y LDI Campus Mexicali [En línia] <http://garibays.com/calidad/calidad-02.html>

Agraïments

Voldria agrair a totes les persones que m'han acompanyat i animat al llarg del projecte.

Al meu director de projecte Xavier Verge, per escoltar les meves propostes, orientar-me i oferir-me els seus consells, ajudar-me i confiar en mi.

A la meva tutora Raquel López, vull agrair el suport constant, la positivitat i confiança demostrada al llarg del projecte, així com per donar-me la oportunitat de créixer a l'organització dia rere dia.

A International Softmachine System, s.l. per la seva col·laboració.

A la meva família i amics per la paciència demostrada i per patir tot aquest temps fins a l'arribada d'aquest dia amb mi.

A la meva parella per motivar-me, entendre'm i escoltar-me en els moments més difícils, així com agrair els moments compartits en aquest temps de constant esforç i sacrifici.

A tots vosaltres, gràcies per l'atenció.

Sabadell, Juny de 2009

Signat: Vanessa Gómez Monfort