

Gestió clàssica de projectes *vs.* desenvolupament àgil

Marcos Bermejo
Marc Florit
Ramon G. Sedó

PID_00236238

Temps de lectura i comprensió: **2 hores**





Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

Índex

1. Introducció	5
2. La complexitat del projecte multimèdia	6
2.1. Problemes en el desenvolupament de productes multimèdia	7
2.2. Per què apareixen aquests problemes?	8
3. Gestió clàssica de projectes	9
3.1. Fases de la gestió clàssica de projectes	9
3.2. Resultats de la gestió clàssica de projectes	10
4. El desenvolupament àgil	12
4.1. Persones i interaccions per sobre de processos i eines	13
4.2. Programari que funciona per sobre de documentació exhaustiva	13
4.3. Col·laboració amb el client per sobre de negociació contractual	14
4.4. Resposta al canvi per sobre del seguiment d'una planificació	15
4.5. Diferents metodologies àgils	15
Bibliografia	17

1. Introducció

Aquest mòdul es divideix en tres parts:

- 1) Primer parlarem sobre els projectes multimèdia i analitzarem la seva complexitat intrínseca.
- 2) Veurem les eines i processos clàssics en la gestió de projectes.
- 3) I finalment veurem noves metodologies, processos i tècniques de gestió de projectes multimèdia que intenten corregir els problemes més habituals que sorgeixen en el complex desenvolupament del projecte.

2. La complexitat del projecte multimèdia

Els **productes multimèdia**, en general, són projectes complexos i es basen en la incertesa per definició. Aplicat a l'entorn del desenvolupament de productes multimèdia, podem dir que un **projecte és complex** quan tant la tecnologia que cal utilitzar per a resoldre un problema com els coneixements sobre els requisits del projecte no són coneguts d'una manera clara al principi del procés.

Referència bibliogràfica

Ralph Stacey. "Strategic management and organizational dynamics". A: **Ken Schwaber; Mike Beedle.** *Agile software development with Scrum.*

Els requisits tendeixen a ser complexos amb molta facilitat. De fet, podem parlar de requisits simples si:

- El client els pot capturar tots i transmetre'ls al desenvolupador, i aquest és capaç d'entendre'ls completament.
- No hi ha diverses parts interessades en el projecte amb interessos divergents.
- El client sap exactament el que necessita.

Com es pot veure, això difícilment és assolible. A part, hem de tenir en compte que els productes multimèdia són desenvolupats per persones que treballen en grup. Si les persones ja som complexes des del punt de vista individual, aquesta complexitat s'incrementa a bastament quan treballem en grup.

En les metodologies tradicionals, altrament dites *predictives*, s'ha intentat lluitar contra aquesta complexitat de manera que tant els requisits com la tecnologia –tots dos aspectes inherents al desenvolupament de productes multimèdia– desapareguessin de l'equació, donant com a resultat fases molt grans de presa de requisits, anàlisis llargues d'aquests requisits i contractes massa tancats i restrictius.

Malgrat els esforços efectuats, es difícil lluitar contra la complexitat dels productes multimèdia i aquestes aproximacions no han portat a la realització de projectes més bons, sinó a tenir projectes amb problemes.

2.1. Problemes en el desenvolupament de productes multimèdia

Segons un article de la IEEE (<http://spectrum.ieee.org/computing/software/why-software-fails/0>), els factors habituals pels quals un producte multimèdia falla són els següents:

- Objectius del projecte poc realistes o poc articulats.
- Estimacions poc acurades dels recursos necessaris per a portar a terme el projecte.
- Requisits del sistema mal definits.
- Informació pobra sobre l'estat del projecte.
- Riscos no gestionats.
- Comunicació insuficient entre els usuaris, els desenvolupadors i els clients.
- Utilització de tecnologia poc madura.
- Incapacitat de gestionar la complexitat del producte.
- Pràctiques de desenvolupament descuidades.
- Gestió ineficaç del projecte.
- Pressions comercials.

En un estudi realitzat per BCS (<http://www.bcs.org/content/ConWebDoc/19584>), podem veure que dels 214 projectes analitzats de tots els àmbits duts a terme amb metodologies predictives, el 23,8% van ser cancel·lats i el 32,2% es van acabar per sobre del pressupost o del temps previst. Això vol dir que prop d'un 55% dels projectes no van tenir el final desitjat. Altres estudis corroboren aquestes dades. Si tinguéssim en compte la satisfacció final del client amb el producte que es lliura, segurament aquests percentatges encara s'incrementarien més.

Alguns d'aquests problemes són especialment típics i preocupants:

- El client triga molt temps a poder utilitzar el resultat del projecte; mentre tant, el context canvia i els competidors treuen nous productes. Si es cancel·la el projecte, el client haurà gastat els diners per a no res.
- El projecte s'ha complicat més del que es podia esperar: hi ha retard i s'han d'accelerar els lliuraments. Comencen els pedaços i no hi ha temps per a proves ni per al control de qualitat.
- L'equip fa hores extres i està poc motivat: simplement es dediquen a complir ordres. Cadascú va a la seva.
- El client demana canvis! Aquests canvis són difícils de realitzar, ja que el projecte s'està acabant i no queda temps ni pressupost.
- Tothom acaba insatisfet: tant el client com l'equip.

Lectura complementària

Podeu trobar l'article a l'adreça d'Internet següent: <http://spectrum.ieee.org/computing/software/why-software-fails/0>

Referència web

Podeu trobar l'estudi realitzat per BCS a l'adreça d'Internet següent: <http://www.bcs.org/content/ConWebDoc/19584>

2.2. Per què apareixen aquests problemes?

D'una banda, les diverses fases s'allarguen més del previst i sempre en surten perjudicades les fases de proves i de possibles canvis.

De l'altra, el client només ha estat veient paper (documents de disseny) fins a una fase molt avançada del projecte. El client va a cegues, gasta diners en una cosa que no té cap tipus d'aplicació. Són documents, no ho pot fer servir.

Tanmateix, sempre apareixen les discussions entre el client i el proveïdor. L'un considera que l'altre no ha entès el que volia i que no s'estan complint els terminis previstos. L'altre pensa que el que se li demana no estava en l'abast del projecte i que hi perd diners. Tot això fa que es dediqui molt d'esforç a objectius que aporten poc valor.

És a dir, en el desenvolupament de productes multimèdia es compleix perfectament el *principi de Pareto*, segons el qual el 80% del valor d'un producte prové del 20% de les funcionalitats implementades. Per tant, invertim molt de temps, esforç i diners en coses que no es faran servir. Hem de fer pagar el client per això?

Sembla clar que aquesta no és la millor manera de desenvolupar un producte multimèdia. Tenim alguna alternativa que ens pugui funcionar més bé? En aquesta assignatura aprendrem que sí.

3. Gestió clàssica de projectes

3.1. Fases de la gestió clàssica de projectes

Vista la complexitat del projecte, la gestió clàssica d'un producte multimèdia la dividim en les cinc fases següents:

- 1) Presa de requisits
- 2) Anàlisi
- 3) Disseny
- 4) Construcció
- 5) Proves

1) Presa de requisits

A l'inici del projecte el client sap molt bé el que necessita; a més, és capaç de fer-se entendre perfectament i sap comunicar-nos les expectatives que té. Tanmateix, l'equip de desenvolupament sap d'una manera exacta el que ha de fer per a aconseguir dur a terme el projecte i es prenen uns requisits detallats. Quan aquesta fase s'acaba, ja no cal tornar a parlar amb el client perquè hem documentat, especificat i signat tot el que vol en el producte. Fins al final del projecte no el tornarem a veure, moment en què li lliurarem el que ens va demanar.

2) Anàlisi

Una vegada entra a la nostra empresa el document (habitualment, d'unes 200 pàgines o més) amb l'especificació dels requisits del producte (el **què**) recollits en la fase de presa de requisits, l'analista funcional el llegeix i l'entén perfectament; a partir d'aquest document és capaç d'elaborar un altre document, amb una anàlisi funcional detallada d'aquests requisits, juntament amb tota classe de mapes i diagrames.

3) Disseny

En aquesta fase, una vegada ha acabat la fase d'anàlisi, el dissenyador tècnic és capaç d'entendre perfectament el disseny funcional del qual hem parlat en la fase anterior, i extreu una sèrie de documents, diagrames i especificacions tècniques per deixar per escrit tot el que els programadors hauran de programar (el **com**), per tal de deixar clar des del principi quines característiques tècniques tindrà tot el producte multimèdia que estem desenvolupant.

4) Construcció

En aquesta fase, els programadors són capaços d'agafar el disseny tècnic i, sense necessitat de plantejar-se camins alternatius, només han de plasmar els documents de la fase de disseny en codi. Com que tot està perfectament detallat, no apareixerà cap error, els programadors només han de seguir les indicacions del disseny.

5) Proves

Finalment, el client valida el producte entregat, mitjançant la realització de proves manuals que miren de reflectir l'ús que es farà de l'aplicació. S'elabora una estratègia de nivells de prova, on es verifiquen els diferents components, mòduls, i el sistema sencer sobre diferents plataformes, entorns i contextos. L'equip de verificació (que habitualment no és el mateix que el de construcció) ho prova tot en el temps planificat. Finalitzades les proves, en les quals només es troben errors menors que es corregeixen immediatament, el client confirma que ha rebut el que esperava i no s'ha de canviar res.

3.2. Resultats de la gestió clàssica de projectes

En cadascuna de les etapes de gestió, en la documentació entrant hi ha tota la informació necessària per a realitzar la tasca corresponent, i d'aquí surten uns altres documents sense que en aquest procés de transformació s'hagi perdut informació pel camí. Tot aquest procés flueix amb total normalitat fins que el projecte s'acaba i es lliura al client exactament allò que mig any abans havia demanat.

Com hem dit a l'inici d'aquest mòdul, això és irreal. Mai no ha funcionat aquest plantejament en un producte multimèdia. Per què?

- Perquè en el moment d'engegar un projecte, quan el client ens expressa les necessitats que té mai no les sap comunicar perfectament; a més a més, nosaltres, com a proveïdors de productes multimèdia, mai no les sabem interpretar del tot.
- Perquè en tota aquesta anada i vinguda de documents, sempre hi ha errors, els quals s'amplifiquen d'una fase a l'altra. Un petit error en la presa de requisits esdevé un gran error en la fase de disseny, i quan arriba a la fase de construcció es converteix en un error incorregible, amb el qual hem de conviure indefinidament.
- Perquè el client només ha vist paper durant tres quartes parts de la durada del projecte. El client ha anat signant documentació, moltes vegades massa extensa, tècnica i complexa per a entendre-la, i no és fins al final que s'obté

el resultat de tot aquest temps d'espera. I el més probable es que no sigui allò que havia imaginat.

- Perquè, seguint aquest model, els canvis són sempre impossibles o massa traumàtics. Un canvi en la fase de construcció representa molt d'esforç, és sempre car i origen de molts enfrontaments amb el client. I la fase de construcció esdevé un etern "projecte finalitzat en un 97%". En aquest punt, els canvis d'última hora o els errors arrossegats al llarg de tot el procés fan que la data de lliurament se'ns tiri a sobre i, òbviament, el producte surt sense provar-lo i amb uns índexs de qualitat bastant mediocres.
- Perquè, amb aquesta gestió, suposem que les necessitats del client no variaran durant tot el projecte, i això mai no és cert, i encara més en el món multimèdia, en què en un mes poden canviar moltes coses.

4. El desenvolupament àgil

El 2001, disset enginyers d'alt nivell en el camp del desenvolupament de programari es van reunir a Utah per a explorar i compartir quin consideraven que havia de ser el futur del desenvolupament de programari. Dins del grup, hi havia molts dels proponents de metodologies emergents en aquell moment, com Scrum, Extreme Programming, Crystal, Feature Driven Development, i d'altres. Junts, van coincidir en el nom per al seu moviment: *agile* ('àgil').

En aquesta reunió els integrants van constituir l'**Agile Alliance** (Aliança Àgil), i van escriure *Manifesto for agile software development* (popularment, *The agile manifesto*; en català, *Manifest àgil*): un conjunt d'estaments que han servit com a declaració de valors i principis que han de seguir les metodologies àgils.

- **Persones i interaccions** per sobre de processos i eines.
- **Programari que funciona** per sobre de documentació exhaustiva.
- **Col·laboració amb el client** per sobre de negociació de contractes.
- **Resposta al canvi** per sobre del seguiment d'una planificació.

Els valors àgils semblen simples i obvis la primera vegada que hom els sent, i continuen sense modificar-se des del dia que els fundadors de l'Aliança Àgil els van publicar com a part del *Manifest àgil*.

Principis del Manifest àgil

- 1) La nostra prioritat principal és satisfer el client mitjançant el lliurament inicial i continu de programari avaluable.
- 2) Els canvis en els requisits són benvinguts, fins i tot en etapes tardanes del desenvolupament. Els processos àgils aprofiten el canvi com un avantatge competitiu del client.
- 3) Cal alliberar programari sovint, des d'unes quantes setmanes fins a uns quants mesos, amb preferència per l'escala més curta.
- 4) La gent de negocis i els desenvolupadors han de treballar plegats diàriament al llarg del projecte.
- 5) Construïm els projectes al voltant de persones motivades. Els oferim l'entorn i el suport que necessiten, i hi confiem per fer la feina.

- 6) El mètode més eficient i efectiu per a compartir informació amb l'equip de desenvolupament són les converses cara a cara.
- 7) El programari que funciona és la principal mesura de progrés.
- 8) Els processos àgils promouen el desenvolupament sostenible. Els *sponsors*, els desenvolupadors i els usuaris han de ser capaços de mantenir el ritme indefinidament.
- 9) L'atenció contínua a l'excel·lència tècnica i al bon disseny milloren *agile*.
- 10) La simplicitat –l'art de maximitzar la feina no feta– és essencial.
- 11) Les arquitectures, els requisits i els dissenys més bons emergeixen d'equips autoorganitzats.
- 12) A intervals regulars, l'equip reflexiona sobre la manera de ser més efectiu i, consegüentment, afina i ajusta el context.

Però, posar aquests valors en pràctica en el dia a dia és més difícil del que sembla inicialment. En els subapartats següents els anirem desgranant.

4.1. Persones i interaccions per sobre de processos i eines

Un dels valors fonamentals de l'agilisme és que les persones que realitzen la feina són les que saben més bé com s'ha de fer. A diferència de la gestió tradicional de projectes, en què habitualment hi ha la figura de l'expert o gestor que diu a la resta de persones què han de fer i com ho han de fer, en el mètode àgil l'equip de persones que duu a terme la feina és **autoorganitzat**.

Això no implica que no es pugui suggerir a l'equip que faci servir una eina o una altra. Una de les tècniques més utilitzades per a l'estimació de tasques és el **planning poker**, però l'equip de desenvolupament i vosaltres potser no la coneixereu mai si no s'encoratja l'equip a investigar i buscar noves maneres de treballar, no solament a fer servir les que ja sabem que funcionen. Aquest concepte –buscar la millora contínua– és necessari per a aconseguir que l'equip de treball creixi i millori, amb la qual cosa serà cada cop més productiu en el futur.

4.2. Programari que funciona per sobre de documentació exhaustiva

Aquest principi àgil sovint porta també a males interpretacions.

La documentació és adequada quan serveix el propòsit de crear valor i portar el projecte endavant, de manera que permet intercanviar coneixement.

Per exemple, la documentació d'usuari és un recurs molt valuós en totes les eines de programari. Els problemes s'esdevenen quan el focus de l'equip es desvia del producte que construeixen, per centrar-se en altres tasques que els exigeix un determinat procés com és la documentació.

Quan es comença el procés de desenvolupament i es duen a terme grans esforços per a crear una documentació exhaustiva, se sacrifica l'oportunitat d'inspeccionar i adaptar-se a mesura que el projecte avança, de manera que s'aprèn dels errors que es cometen i s'ajusta el procés de desenvolupament al que realment necessita el projecte.

Una mala interpretació habitual és que els equips àgils no documenten o planifiquen; però, en la pràctica, dediquen més temps i energia a la planificació i documentació que els tradicionals, ja que la planificació s'actualitza i es corregeix contínuament.

En un desenvolupament àgil de productes multimèdia, la planificació i la documentació són pertot arreu al voltant de l'equip, en forma d'històries d'usuari, piles de requisits, proves d'acceptació i gràfics grans (molt grans!) i visibles; tots aquests aspectes són part de l'entorn per a una comunicació rica dins de l'equip.

4.3. Col·laboració amb el client per sobre de negociació contractual

Aquest valor àgil també busca evitar el gran disseny inicial, però ho fa posant l'èmfasi a mantenir un diàleg continu entre l'equip de desenvolupament i el client, tan obert i fluid com sigui possible. Els contractes són bons i necessaris, especialment quan protegeixen els interessos de les dues parts, tal com succeeix en els bons contractes.

Però, molts dels fundadors de l'Aliança Àgil eren consultors i coneixien els riscos que s'amaguen darrere dels contractes. Un dels més habituals en la nostra indústria és que el proveïdor es compromet a realitzar una certa quantitat de feina en un temps determinat, en què el profit que tregui del projecte dependrà de la rapidesa i el baix cost amb què sigui capaç de dur a terme els requisits mínims del client.

Els autors del *Manifest àgil* van concloure que un projecte basat exclusivament en un contracte posa èmfasi en el lloc incorrecte. Ells van preferir promoure un entorn col·laboratiu, en què el client i l'equip treballen conjuntament per aconseguir un objectiu que és compartit, l'èxit del projecte.

Vegeu també

El mòdul "Nova relació entre client i proveïdor: contractes àgils" d'aquesta assignatura està dedicat íntegrament a la creació d'aquest tipus de contracte que veurem amb més profunditat.

4.4. Resposta al canvi per sobre del seguiment d'una planificació

Les organitzacions que basen el seu model a "seguir el pla establert" acostumen a tenir processos creats amb les millors intencions. Per a prevenir que el projecte tingui endarreriments, no es permeten canvis de funcionalitat no planificats o qualsevol altre disfunció que pugui afectar el pla inicial o, si de cas, se sotmeten a uns requisits durs, com revisar tota la planificació o fins i tot redactar un nou contracte o un annex a l'actual. Però, malgrat tot això, no funciona. Un control de canvis només té èxit en un entorn en què els canvis són controlables. En canvi, el desenvolupament de programari és un procés en què a mesura que avancem anem descobrint coses que no sabíem al començament, i que ens obligarà a fer canvis no esperats.

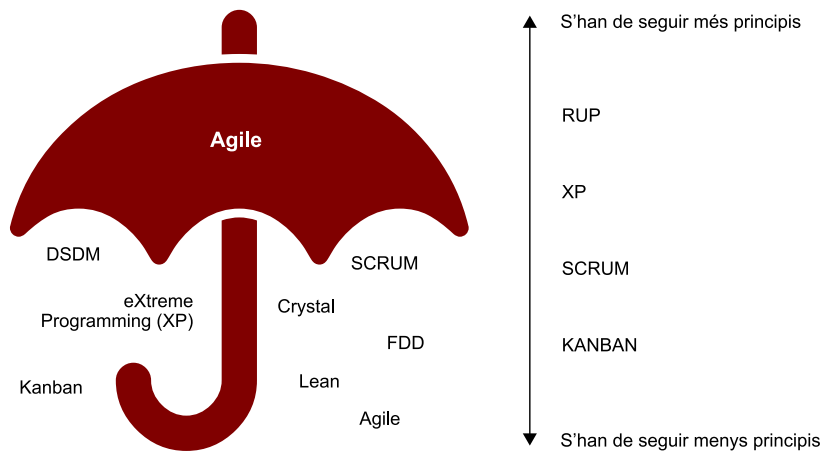
La planificació en els projectes de desenvolupament de productes multimèdia ha de ser fluida, no fixa, per al benefici de l'equip però sobretot per al benefici del producte que estem desenvolupant i, en última instància, per al benefici del client. Aquesta és la raó per la qual planifiquem per al canvi.

4.5. Diferents metodologies àgils

Partint dels 12 principis del Manifest àgil que hem vist a l'inici del punt 4, s'han anat desenvolupant diferents metodologies àgils: SCRUM, Kanban, Crystal, XP, FDD o RUP entre d'altres.

Totes elles parteixen dels 12 principis, però es diferencien en si segueixen o no més estrictament els principis esmentats.

Per a facilitar-ne la comprensió, en el gràfic següent podem veure un "parai-gua" que aixopluga les metodologies que parteixen dels principis àgils i una línia que defineix si aquestes metodologies es poden considerar més o menys àgils.



Font: basat en "Agile Umbrella" (<http://www.codeproject.com/KB/architecture/1064114/023.jpg>)

Tenint en compte aquesta varietat de metodologies àgils, els propers mòduls didàctics se centraran en les dues més utilitzades: SCRUM i Kanban.

Bibliografia

Diversos autors. *Manifesto for agile software development* [en línia].

<http://agilemanifesto.org/>

Martin, Robert C. (2012). *Agile software development: principles, patterns, and practices*. Upper Saddle River, NJ: Pearson Prentice Hall.

Ruby, Sam (2009). *Agile web development with rails*. Raleigh, NC: Pragmatic Bookshelf.

Shore, James (2008). *The Art of agile development*. Beijing / Sebastopol, CA: O'Reilly Media, Inc.

Anderson, David James (2004). *Agile management for software engineering: applying the theory of constraints for business results*. Upper Saddle River, NJ: Prentice Hall PTR, cop. 2004.

Highsmith, James A. (2004). *Agile project management: creating innovative products*. Addison-Wesley, cop. 2004.

Schwaber, Ken (2004). *Agile project management with Scrum*. Redmond, Wash.: Microsoft Press.

Schwaber, Ken (2002). *Agile software development with Scrum*. Upper Seadle River, NJ: Pearson Prentice Hall.

