Lucru in echipa pentru dezvoltarea produselor software

Introducere – Simona Virlan

Metodologii – Ioana Frincu

Activitati – Ioana Frincu

Roluri in echipa – Simona Virlan

Concluzii – Ioana Frincu

Bibliografie - Simona & Ioana

Cuprins

[Introducere 2](#_Toc441090459)

[Metodologii 2](#_Toc441090460)

[Agile 3](#_Toc441090461)

[Activitati 6](#_Toc441090462)

[Identificarea nevoii 6](#_Toc441090463)

[Planificarea 7](#_Toc441090464)

[Design-ul 7](#_Toc441090465)

[Implementare, testare si documentare 7](#_Toc441090466)

[Livrare si mentenanta 7](#_Toc441090467)

[Roluri în echipă 7](#_Toc441090468)

[Studiu de caz: Echipele de dezvoltare software Microsoft 9](#_Toc441090469)

# Introducere

 Dezvoltarea software este un proces al programării, documentării, testării și fixării bug-urilor implicate în crearea și întreținerea aplicatilor și framework-urilor dintr-un ciclu de viață software ce duc la un produs software. Termenul se referă la procesul de scriere și întreținere a codului sursă , dar include și conceptul produsului software dorit până la realizarea finală a produsului. Prin urmare, dezvoltarea software poate include o nouă dezvoltare, prototipuri, modificarea, reutilizarea, întreținerea sau orice alte activități ce pot avea legătură cu produsele software. [2]

 Dezvoltarea produselor software este o activitate complexă ce necesită un grup de persoane care lucrează în mod eficient în echipă. Studiile (Hoegl & Gemuenden, 2001; Liang et al., 2012; Henderson & Lees, 1992) au arătat că performanță unei echipe eficiente este cu mult mai mare față de cea a persoanelor care acționează independent. O eficientă ridicată este o sarcină dificilă ce necesită investiții și angajamete. În acest proiect, vom prezența principalele caracteristici ce diferențiază lucrul în echipă de lucrul individual. [9]

# Metodologii

Metodologia reprezinta un set de reguli, metode si postulate care sunt folosite pentru a interpreta sau rezolva diferite probleme in interiorul unei discipline. Spre deosebire de un algoritm, o metodologie nu e o formula ci un set de practici. [1]

In cazul dezvoltarii software, un proces de dezvoltare (metdologie de dezvoltare, model sau ciclu de viata) este un framework folosit pentru a structura, planifica si controla procesul dezvoltarii sistemelor informationale. [2]

Dintre metodologiile de dezvoltare software amintim [3]:

* Agile Software Development Methodology
	+ Minimizeaza riscurile prin dezvotarea in intervale de timp mici (iteratii);
	+ Se axeaza pe comunicarea in timp real;
	+ Documentatia scrisa este in mica cantitate.
* Crystal Methods
	+ Performanta este direct dependenta de oameni, interactiune, comunitate, talent;
	+ Fiecare echipa e compusa din oameni unici. prinurmare, fiecare echipa trebuie sa issi aleaga propriul tip de planificare
	+ Cristal provine de la fatetele numeroase ale unei pietre pretioase ale caror substratusi reprezinta valorile si principiile si fiecare fateta prezinta un element, cum ar fi tehnicile;
* Dynamic Systems Development Model (DSDM)
	+ Implicare activa a user-ului;
	+ Autoritate crescuta a echipelor;
	+ Concentrare pe livrare frecventa a produselor;
	+ Concentrare pe “potrivirea” intre scopul business-ului ca si criteriu esential de acceptare a unui livrabil;
	+ Dezvoltare incrementata si iterativa pentru a asigura o solutie de business convergenta;
	+ Schimbari reversibile in dezvoltare;
	+ Cerintele sunt stabilite la nivel inalt;
	+ Testarea e integrata intr-un ciclu de dezvoltare
	+ Colaborare si cooperare intre toti cei implicati;
* Scrum
	+ scopul de a imbunatati dramatic performanta si eficienta;
	+ Permite crearea de echipe auto-organizate
* Waterfall (a.k.a. Traditional)
	+ Metda rigida si liniara
	+ Are scopuri distincte pentru fiecare faza de dezvoltare, iar fiecare faza o succede pe urmatoarea, fara a se putea reveni asupra fazelor care au trecut;
	+ Avantaj: Permite impartirea in departamente si controlul managerial;

## Agile

Una dintre metodologiile cele mai populare si aplicata cu succes la proiecte este Agile. Pe aceasta vom prezenta in capitolul de fata intrucat este raspandita si ne permite sa ne legam de lumea practica a business-ului si a dezvoltarii in echipa.

Agile Software Development reprezinta un set de metode in care cerintele si slutiile evolueaza prin colaborare intre echipe cu diferite functionalitati, aut-organizate. Promoveaza planificarea adaptiva, dezvoltarea evolutionara, livrare mai devreme de terment, imbunatatire continua si incurajeaza raspunsul rapid si flexibil la schimbare.

Principiile dezvoltarii pe baza Agile sunt [4]:

1. Prioritatea noastră este satisfacţia clientului prin livrarea rapidă şi continuă de software valoros.
2. Schimbarea cerinţelor este binevenită chiar şi într-o fază avansată a dezvoltării. Procesele agile valorifică schimbarea în avantajul competitiv al clientului.
3. Livrarea de software funcţional se face frecvent, de preferinţă la intervale de timp cât mai mici, de la câteva săptămâni la câteva luni.
4. Oamenii de afaceri şi dezvoltatorii trebuie să colaboreze zilnic pe parcursul proiectului.
5. Construieşte proiecte în jurul oamenilor motivaţi. Oferă-le mediul propice şi suportul necesar şi ai încredere că obiectivele vor fi atinse.
6. Cea mai eficientă metodă de a transmite informaţii înspre şi în interiorul echipei de dezvoltare este comunicarea faţă în faţă.
7. Software funcţional este principala măsură a progresului.
8. Procesele agile promovează dezvoltarea durabilă. Sponsorii, dezvoltatorii şi utilizatorii trebuie să poată menţine un ritm constant pe termen nedefinit.
9. Atenţia continuă pentru excelenţă tehnică şi design bun îmbunătăţeşte agilitatea.
10. Simplitatea--arta de a maximiza cantitatea de muncă nerealizată--este esenţială.
11. Cele mai bune arhitecturi, cerinţe şi design emerg din echipe care se auto-organizează.
12. La intervale regulate, echipa reflectă la cum să devină mai eficientă, apoi îşi adaptează şi ajustează comportamentul în consecinţă.



Cea mai populara modalitate de a implementa Agile este Scrum datorita simplitatii si flexibilitatii sale. Scrum are doar 3 roluri in echipa: Product Owner, Team si Scrum Master. Responsabilitatile unui project manager traditional sunt impartite intre aceste trei roluri.

O implementare de tip Scrum reprezinta o metoda iterativa continua, circulara:



Scrum este format din 5 intalniri[5] [6]:

1. Backlog Grooming (aka Backlog Refinement),
2. Sprint Planning: Product Owner-ul si echipa negociaza ce itemi din “lista de rezolvat” vor incerca sa rezolve
3. Daily Scrum (aka 15-minute standup): Zilnic, la aceeasi ora si in acelasi loc membrii echipei ofera informatii despre munca depusa pana in acel moment, planurile pentru ziua respectiva si provocarile intalnite.
4. Sprint Review Meeting: Dupa executia Sprint-ului (ziua de lucru) membrii echipei vor incrementa activitatea pana la acel punct.

Sprint Retrospective Meeting: Echipa reflecteaza la eficienta sa, isi analizeaza comportamentul si iau masuri, daca e cazul

# Activitati

Activitatile de baza in cadrul dezvoltarii unui produs software sunt:

1. Identificarea nevoii
2. Planificarea
3. Design-ul
4. Implementare, testare si documentare
5. Livrare si mentenanta

## Identificarea nevoii

Sursele de idei pentru dezvoltarea produselor software sunt provenite din cercetarea de piata, inclusiv surse demografice sau un partener de business. Ideile de obicei trec prin departamentul de marketing, (fezabilitate pentru piata). Pentru a asigura lista de lucruri de facut, o echipa de dezvoltare software impreuna cu o echipa de Resurse Umane stabilesc necesitatile.

## Planificarea

Ea este subiectul fiecarei activitati. Cea mai importanta faza este analiza cerintelor. Dat fiind ca clientii au de obicei o idee abstracta despre ce vor sa obtina, inginerii software transpun asta in termeni de tehnologie.

Dupa ce sunt adunate derintele clientului se face o analiza cu sccopul dezvoltarii (Scope document).

## Design-ul

 In momentul in care cerintele sunt enuntate, se creeaza documentul de design software. Acesta contine un design preliminar (sau high - level) a modulelor principale cu o imagine de ansamblu a componentelor.. Limbajul, sistemul si componentele hardware ar trebui sa fie cunoscute in acest moment. Apoi un document detaliat (low -level) este creat.

## Implementare, testare si documentare

Implementarea reprezinta partea in care inginerii software chiar programeaza. testarea este o parte integranta din proces si asigura recunoasterea defectelr cat de rapid posibil. Documentarea de face cu scopul unei referinte pentru dezvoltare ulterioara.

## Livrare si mentenanta

Livrarea are loc imediat dupa testare si aprobare pentru publicare, produsul devine vandabil sau este distribuit intr-un mediu anume. Mentenanta si imbunatatirea pentru a face fata defectelor ulterioare poate deveni anevoioasa si cerinte peste care s-a sarit pot forta redesign-ul solutiei software.

# Roluri în echipă

 Există o serie de mai multe roluri ce există în majoritatea proceselor de dezvoltare software. Un membru al echipei poate avea mai multe roluri și anumite roluri pot fi suprimate pentru un anumit tip de proiect, dar toate aceste roluri există sub o formă sau altă în fiecare proiect de dezvoltare software.

 Orice proiect de dezvoltare software are în centrul sau o echipă de oameni cu scopul de a satisfice nevoile colective ale părții interesate. Această echipă de bază se bazează pe o echipă mai extinsă pentru a își asigură contextul în care își desfășoară activitatea. Ambele echipe trebuie să realizeze o treabă bună pentru un rezultat mare.

 Există mai multe roluri, ce depind de metodologia urmărită. Fiecare persoană data are unul sau mai multe roluri ce se pot schimbă în timp.

Cele mai comune roluri pentru metodologia Agile sunt:

* lider - antrenorul echipei sau liderul proiectului este responsabil pentru a oferi facilități echipei, obținând resurse pentru această și pentru a o feri de probleme. Acest rol compensează îndemânărilor software dar nu și pe cele tehnice precum planificarea și programarea, activități care sunt mai bine lăsate în seamă întregii echipe.
* membrul echipei - acest rol se referă uneori la dezvoltatori sau programatori, este responsabil pentru crearea și livrarea sistemului. Acesta include modelarea, programarea, testarea și lansarea activităților, la fel de bine că și celelalte.
* beneficiarul produsului - este persoană responsabilă într-o echipă care se ocupă cu fixarea priorităților pentru luarea deciziilor, luarea deciziilor și transmiterea lor în timp util.
* partea interesată - este oricine care este utilizator direct sau indirect, “proprietarul de aur” care găsește proiectul, dezvoltatorul care lucrează la alte sisteme care integrează sau interacționează cu cel sub dezvoltare
* experții tehnici - uneori echipa are nevoie de ajutor de la experții tehnici. Ei sunt aduși temporar să ajute echipa să treacă peste o problemă dificilă și să împărtășească cunostinele cu ceilalți dezvoltatori ai echipei.
* experții în domeniu - sunt aduși să lucreze cu echipa, de exemplu un evaluator care să explice detaliile și cerințele sau sponsorul care să explice viziunea proiectului
* testerul independent - echipele eficiente au adesea un tester independent care lucrează în paralel pentru a valida munca lor. Acest rol este opțional, de obicei adoptat pentru proiectele de mare anvergură [7],[8]

 

Fig. Structura unei echipe Agile [7]

# Studiu de caz: Echipele de dezvoltare software Microsoft

 Deși compania Microsoft este o companie mare, cu mulți angajați, încă păstrează o structură ce se bazează pe echipe mici.

 Liderul lucrează cu un grup de manageri ce sunt experți tehnici seniori și mulți dintre ei sunt veterani ce lucrează pentru Microsoft de foarte mulți ani. Aceștia cunosc tehnologia și știu cum să facă bani cu îndemânările lor. Pe lângă acest nivel de administrare, compania mai are în general încă două nivele de administrare pentru dezvoltarea software. Fiecare nivel are un manager și în fiecare nivel există echipe mai mici conduse de către un manager de program. Această structură ierarhică a companiei nu reflectă fluxul autorității. Următoarea propoziție definește clar rolul unui manager de program într-o echipă: “Managerul de program este un lider; și coordonator; dar nu este șeful.” [11, p.77].

 Pentru ca fiecare persoană din echipă să se implice, membrii echipei software îi încurajează să își exprime ideile chiar dacă sunt negative sau pozitive.

 Structura de comunicare în Microsoft este necentralizată și conectată maxim, fiecare putând să comunice cu fiecare. Pentru a crește comunicarea între membrii echipei aproape toată dezvoltarea de produse noi se face pe un singur site după cum spune și Bill Gates: “Toată existența noastră cu mici excepții, aici pe un singur site, astfel încât orice interdependențe ar exista să poți să vezi persoana față în față...”

 [11, p.26] [12]

Bibliografie

[1] <http://www.businessdictionary.com/definition/methodology.html>

[2] <https://en.wikipedia.org/wiki/Software_development>

[3] <http://www.itinfo.am/eng/software-development-methodologies/>

[4] <http://agilemanifesto.org/>

[5] <http://scrumtrainingseries.com/>

[6] <http://scrumreferencecard.com/scrum-reference-card/>

[7] <http://www.ambysoft.com/essays/agileRoles.html>

[8] <https://flowchainsensei.wordpress.com/2011/02/25/the-many-roles-in-software-projects/>

[9] <http://www.slideshare.net/fdgiraldo/teamwork-in-software-engineering-projects>

[10] <http://www.unice.fr/crookall-cours/teams/docs/team%20Successful%20teamwork.pdf>

[11] Cusuman0, M.A. and R.W. Selby. 1995. Microsoft secrets. New York, NY: The Free Press.

[12] <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=617239>