



Metodologia de dezvoltare și implementare a
produsului software – model Waterfall
(cascade model)

CUPRINS

1. INFORMAȚII GENERALE.....	3
2. ALGORITMUL DE ACȚIUNE	4
3. ETAPELE CICLULUI DE VIAȚĂ	5
4. PROCESE	15
5. PROIECTARE ȘI DOCUMENTARE.....	16
6. ASPECTE POZITIVE ȘI NEGATIVE ALE METODOLOGIEI	17
7. UTILIZAREA METODOLOGIEI	19
8. ROLUL FURNIZORULUI	19

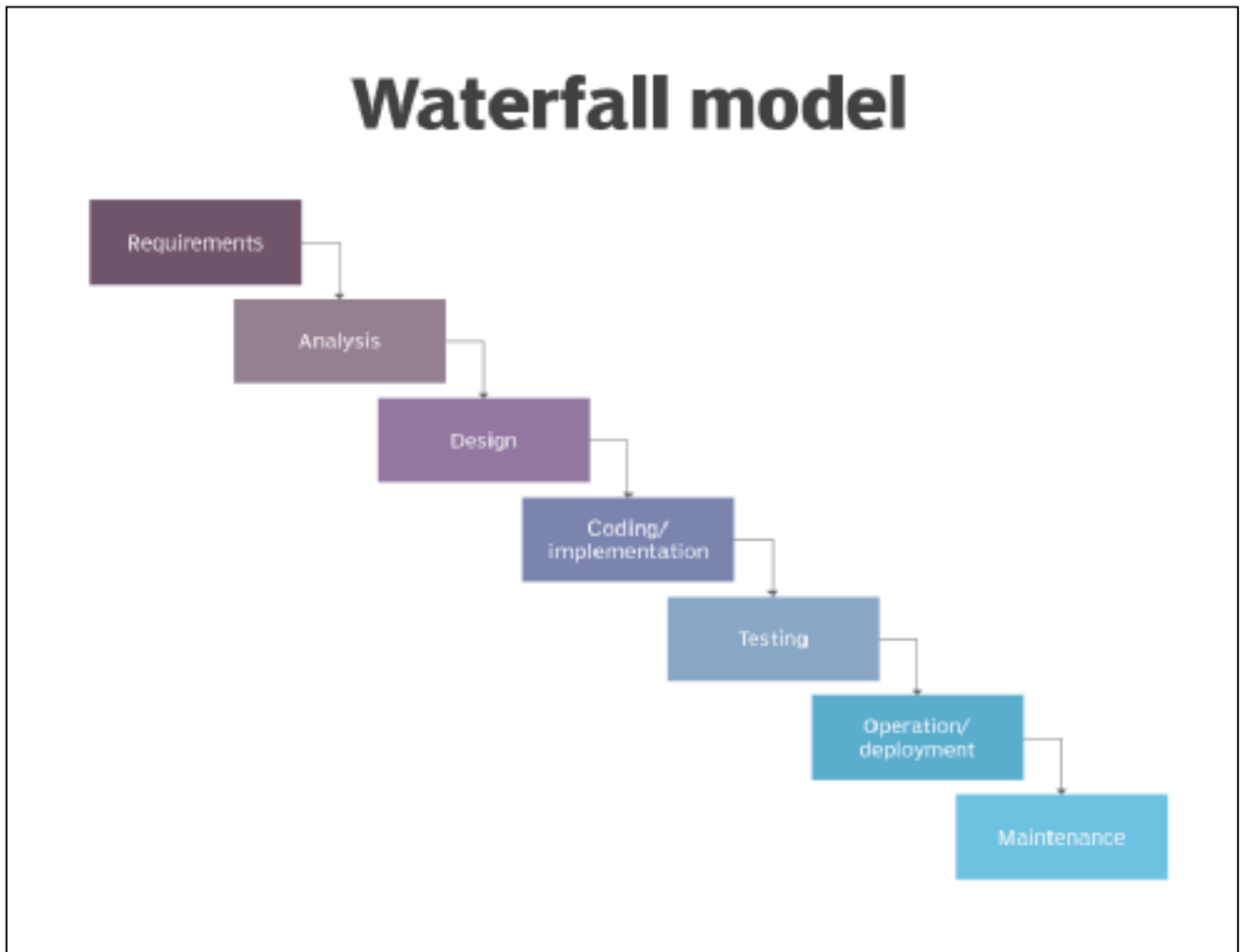
1. INFORMAȚII GENERALE

Modelul în cascadă (Waterfall) este un model de dezvoltare software în care procesul de realizare a aplicației urmează un flux secvențial, trecând consecutiv prin fazele de analiză a cerințelor, proiectare, implementare, testare, livrare și suport.

În conformitate cu modelul Waterfall, dezvoltatorul de sistem parcurge etapele într-o succesiune strictă, fără revenire la fazele anterioare:

- În prima etapă se finalizează complet definirea cerințelor, rezultând o listă detaliată de cerințe funcționale și nefuncționale ale sistemului software.
- Odată ce cerințele au fost clar definite și aprobate, se trece la etapa de proiectare, în cadrul căreia sunt elaborate documente tehnice care descriu modul în care cerințele vor fi implementate din punct de vedere arhitectural și logic.
- După finalizarea completă a proiectării, programatorii implementează soluția tehnică conform specificațiilor.
- Etapa următoare presupune dezvoltarea componentelor individuale de către diferite echipe de programatori, în conformitate cu planul stabilit.
- Ulterior, după finalizarea implementării și livrării interne, are loc testarea și depanarea produsului software. În această fază sunt identificate și corectate defectele apărute în etapele anterioare.
- În final, produsul software este implementat în mediul de producție, se realizează instruirea utilizatorilor, derularea exploatării pilot și ulterior se asigură suportul continuu, inclusiv corectarea erorilor și adăugarea de funcționalități noi.

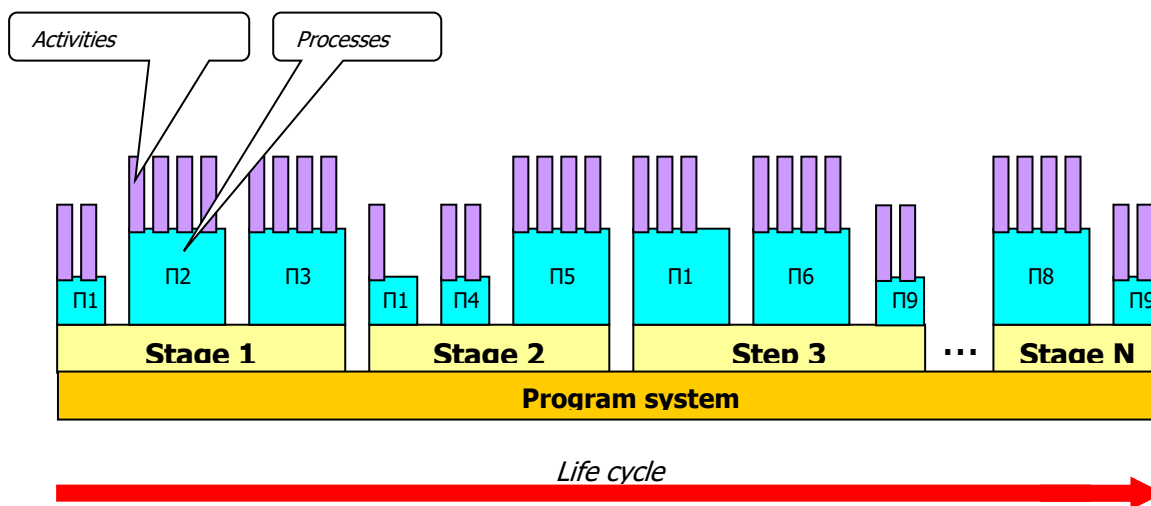
2. ALGORITMUL DE ACȚIUNE



Algoritmul acțiunilor în metodologia "Waterfall"

Pentru implementarea proiectului, alegem o modificare a modelului Waterfall — modelul "Sashimi" sau modelul Waterfall cu faze suprapuse. În acest model, la fel ca în metodologia originală, fazele se succed, dar se suprapun în timp.

Ciclul de viață al unui sistem software conform metodologiei Waterfall-Sashimi constă într-o serie de etape în care sistemul software este planificat, creat, implementat, operat, întreținut și dezafectat. Acest lucru este ilustrat printr-un model tipic al ciclului de viață al sistemului software prezentat în figura de mai jos.



Modelul ciclului de viață al sistemului software conform metodologiei Waterfall

3. ETAPELE CICLULUI DE VIAȚĂ

Etapele ciclului de viață sunt organizate în secvențe care pot să se suprapună și/sau să se repete, în funcție de limitele, dimensiunea, complexitatea, nevoile în schimbare și capacitățile de implementare ale sistemului software. Acestea sunt iterative și se realizează prin acțiuni pas cu pas.

Modelul ciclului de viață al metodologiei Waterfall pentru un sistem software constă în șase etape:

1. Analiza cerințelor proiectului. Se definesc cerințele programului pentru domeniul informațional al sistemului.
2. Proiectare. Se dezvoltă și se formulează o specificație tehnică logică și coerentă a sistemului software. Detalierea sistemului.
3. Dezvoltare. Realizarea unui proiect complet funcțional.
4. Testarea produsului. Exploatarea de test a produsului.
5. Implementare. Include instalarea produsului, instruirea personalului și acceptarea oficială a produsului.
6. Suport. Furnizarea de asistență tehnică pentru produs după lansarea comercială..

ANALIZA CERINȚELOR PROIECTULUI

Aceasta este, probabil, cea mai responsabilă și importantă etapă în crearea unui sistem software de succes. Toate informațiile colectate sunt utilizate pentru a planifica abordarea de bază a proiectului.

În plus, în această fază se planifică cerințele de asigurare a calității și se identifică diverse riscuri asociate proiectului. Rezultatul analizei constă în identificarea diferitelor abordări tehnice care pot fi utilizate pentru implementarea cu succes a proiectului, cu riscuri minime.

În această etapă, echipa de specialiști și părțile interesate convin asupra tuturor detaliilor viitorului produs, răspunzând la întrebări esențiale:

- Care este scopul viitorului produs?
- Ce probleme ar trebui să rezolve?
- Pentru ce este creat produsul?

După clarificarea acestor aspecte, toți participanții au o viziune comună și corectă asupra viitorului produs. Acest lucru facilitează dezvoltarea și minimizează riscurile în crearea software-ului.

PROIECTAREA ȘI DOCUMENTAREA CERINȚELOR

După finalizarea analizei inițiale a cerințelor, următorul pas este definirea clară și documentarea completă a cerințelor pentru produs, urmată de aprobarea acestora de către Client. Dacă scopul primei etape a fost înțelegerea și analiza cerințelor, atunci în această etapă toate obiectivele trebuie formalizate și documentate — acest lucru oferă protecție ambelor părți implicate.

În această fază, echipa de specialiști colaborează cu reprezentanții Clientului și, în unele cazuri, chiar și cu potențiali utilizatori finali, pentru a colecta toate detaliile relevante privind dezvoltarea proiectului. Procesul pornește de la cercetarea pieței și continuă până la definirea stack-ului tehnologic și a funcționalităților viitorului produs.

Toate aceste informații sunt ulterior înregistrate în documentația proiectului, care stă la baza planificării etapelor de dezvoltare, a cronogramului și a termenelor de execuție pentru livrabilele succesive.

După ce cerințele funcționale și stack-ul tehnologic au fost clarificate, se poate trece la proiectarea tehnică și arhitecturală. În această etapă, echipa de dezvoltare elaborează arhitectura viitorului sistem, utilizând tehnologia selectată. Se creează un design adaptiv și orientat pe experiența utilizatorului, se planifică legătura dintre partea de interfață (frontend) și server (backend), se definesc modulele și se proiectează sistemul de securitate al aplicației..

DEZVOLTARE

Etapa de dezvoltare reprezintă faza în care codul sursă este efectiv scris. Specialiștii tehnici utilizează documentația proiectului, prototipurile, designul și arhitectura pentru a crea aplicația sau site-ul web viitor.

Sarcinile sunt distribuite între membrii echipei conform specializării fiecăruia:

- Dezvoltatorii back-end sunt responsabili pentru crearea logicii serverului și asigurarea comunicării eficiente între interfața utilizatorului și server.
- Administratorii de baze de date gestionează structura și integritatea datelor, asigurând stocarea și accesul optim la informații.
- Dezvoltatorii front-end creează interfața responsivă a aplicației web, garantând o experiență utilizator fluidă și intuitivă.

Rezultatul acestei etape este un produs software funcțional, pregătit pentru testare și implementare ulterioară.

TESTARE

Crearea unui produs funcțional nu reprezintă finalul ciclului de dezvoltare. După faza de dezvoltare urmează etapa de testare. În această etapă, echipa compară aplicația concepută în faza de analiză a cerințelor cu cea realizată după implementare. Se răspunde la întrebări esențiale precum:

- Toate funcționalitățile planificate și integrările funcționează corect?
- Toate butoanele/interfețele reacționează corespunzător?
- S-a obținut rezultatul final specificat în cerințele inițiale?

Înainte de trecerea la exploatarea comercială a sistemului, este esențială verificarea performanței în toate modurile prevăzute în specificație, inclusiv în „condiții extreme”. De exemplu, dacă este planificată introducerea distribuită a datelor de către 100 de utilizatori simultan, trebuie verificat dacă sistemul poate procesa corect activitatea simultană a acestora. Aceasta este ideea de bază a testării de încărcare (load testing).

Testarea de încărcare

Are scopul de a prezice comportamentul sistemului în condiții reale și extreme, de a detecta erori, de a monitoriza performanța și disponibilitatea în condiții variate. Obiectivele includ:

- Evaluarea performanței aplicației în următoarele faze:

- Dezvoltare;
- Exploatare pilot;
- Mentenanță și lansări (patch-uri, actualizări).

- Optimizarea aplicației.
- Selectarea platformei hardware/software optime și a configurației serverului:

Categorii esențiale de testare:

Testare de performanță (load testing)

Simularea unui volum mare de utilizatori care accesează simultan diferite module:

- Determinarea timpului de execuție pentru diverse operațiuni la diferite nivele de încărcare;
- Stabilirea performanței maxime posibile a sistemului

Testare de stres (stress testing)

Testarea comportamentului sistemului la sarcini peste limita planificată:

- Evaluarea capacității sistemului de a reveni la parametri normali după suprasarcină;
- Esențială pentru sisteme critice unde eșecul ar avea costuri ridicate.

Testare de fiabilitate (reliability testing)

- Determinarea duratei de funcționare fără erori;
- Stabilitatea sistemului în condiții de încărcare medie pe perioade extinse;
- Identificarea:
 - Scăpărilor de memorie;
 - Configurațiilor incorecte;
 - Rebooturi neplanificate ale serverului.

Testare de configurație (configuration testing)

- Evaluarea impactului modificărilor de configurare asupra performanței
- Compatibilitate cross-platform / cross-browser
- Identificarea „bottleneck”-urilor în modulele sistemului
- Determinarea configurației hardware optime
- Verificarea compatibilității cu OS-uri și aplicații terțe

Pașii implicați:

1. Crearea unei matrice de acoperire pentru toate configurațiile posibile
2. Prioritizarea acestora
3. Planificarea ciclului de viață al testelor
4. Organizarea efectivă a testării conform priorităților

Testare de volum (volume testing)

Această etapă evaluează stabilitatea și performanța aplicației în condițiile procesării unor volume mari de date și a unui număr crescut de utilizatori. Se urmărește determinarea limitei maxime de utilizatori activi simultan fără degradarea performanței.

Absenta unei astfel de testări poate crește semnificativ riscurile operaționale și poate conduce la eșecul soluției într-un interval scurt după implementare. Pentru a preveni aceste riscuri, este esențial ca testarea să fie organizată corespunzător încă din faza de implementare, fiind aplicabilă și sistemelor deja în producție.

Dacă aplicația nu funcționează conform așteptărilor, problemele identificate în această etapă sunt comunicate echipei de dezvoltare pentru remediere. După efectuarea corecțiilor necesare, produsul este pregătit pentru următoarea fază a ciclului de viață.

Secvența activităților necesare pentru efectuarea acestui tip de testare:

1. Se creează o matrice de acoperire care descrie toate configurațiile posibile ale sistemului.
2. Se efectuează prioritizarea configurațiilor.
3. Se planifică ciclul de viață al testării (IP lifecycle).
4. În final, se organizează testarea tuturor configurațiilor majore, conform priorităților stabilite.

IMPLEMENTARE

Etapa de implementare a sistemului reprezintă una dintre cele mai importante faze din perspectiva distribuției responsabilităților între Client și Furnizor. Rolul principal al Furnizorului constă în instalarea modulelor pe serverul de producție, prin migrarea configurației finale din mediul de test.

Totuși, prin formarea corespunzătoare și implicarea activă a specialiștilor Clientului în procesul de creare și implementare a sistemului, o serie de activități pot fi preluate de aceștia, inclusiv:

- Asigurarea funcționalității rețelei locale (LAN) pentru toate stațiile de lucru automatizate ale utilizatorilor.

- Configurarea serverului de producție.
- Stabilirea listei și a numărului de stații de lucru care urmează să fie utilizate în etapa OP (operare pilot).
- Instalarea și configurarea posturilor de lucru, inclusiv alocarea drepturilor de acces.
- Pregătirea informațiilor actualizate pentru fiecare modul al sistemului, inclusiv gestionarea nomenclatoarelor care ar putea să-și fi pierdut relevanța între faza de testare și cea de implementare.
- Organizarea mecanismelor de actualizare a datelor din sistem, inclusiv integrarea cu sisteme externe.

Scopul principal al acestei etape este pregătirea sistemului pentru exploatarea pilot, prin instruirea utilizatorilor finali, introducerea datelor inițiale și emiterea documentației necesare privind trecerea în regim pilot.

Activități majore ale etapei de implementare:

- Asigurarea îndeplinirii necondiționate a tuturor cerințelor de pregătire a modulelor pentru lansarea în regim de operare pilot (conform unui document dedicat).
- Introducerea și validarea datelor inițiale (de exemplu: solduri de început) în fiecare modul.
- Simularea acțiunilor reale ale utilizatorilor în paralel cu aplicațiile deja existente, direct la posturile de lucru.
- Elaborarea, coordonarea și aprobarea regulamentelor interne de interacțiune între departamentele clientului implicate în utilizarea sistemului în regim OP.
- Integrarea modulului cu alte module sau sisteme externe implementate anterior.

Condiții necesare pentru lansarea etapei OP:

- Modulele sistemului au fost migrate cu succes și funcționează pe serverul de producție și pe posturile de lucru ale utilizatorilor, cu drepturi de acces configurate corespunzător.
- Toate nomenclatoarele lipsă au fost completate și introduse în sistem, fiind reconciliate corespunzător.
- Documentația finală a fost aprobată de către Client.

Instruirea utilizatorilor

Instruirea se realizează pe baza principiilor fundamentale de utilizare a sistemului, prin exemple de control, individual pentru fiecare modul. Utilizatorii sunt familiarizați cu interfața aplicației, regulile de lucru și funcționalitățile de bază. În cazul existenței ghidurilor metodologice, acestea vor fi puse la dispoziția utilizatorilor pentru studiu suplimentar.

Etapele principale de instruire:

1. Instruirea utilizatorilor-cheie în etapa de implementare, înainte de organizarea testelor.
2. Instruirea utilizatorilor-cheie înainte de instalarea pe serverul de test (banc).
3. Instruirea utilizatorilor finali înainte de lansarea în regim de exploatare pilot.

Așa cum se poate observa, această divizare a etapelor este determinată de trei repere principale:

În primul rând, este necesar să fie explicate principiile de funcționare ale sistemului deja configurat înaintea primei testări cu utilizatorii-cheie.

Următorul reper — instalarea pe serverul de testare (banc de lucru) — este important ca etapă intermediară, în cadrul căreia utilizatorii-cheie vor fi implicați din nou, de această dată pentru a descrie funcționarea sistemului și interfețele într-un mod mult mai detaliat, astfel încât să fie identificate și eliminate toate observațiile critice rămase înainte de trecerea la operarea pilot (OP).

Desigur, până la acest moment, toți specialiștii IT care vor fi implicați în suportul sistemului trebuie să cunoască în detaliu arhitectura internă a soluției și să fie capabili să efectueze revizii sau să genereze cerințe de modificare, dacă este necesar.

În cele din urmă, este organizată cea mai amplă instruire pentru restul angajaților care vor lucra efectiv cu sistemul. În acest caz, sistemul este prezentat nu din perspectiva arhitecturii tehnice, ci din punctul de vedere al utilizatorului final — axat pe principiile de introducere, procesare și regăsire a datelor.

Indiferent de specificul sistemului, al domeniului de aplicare sau al etapei de instruire, următoarele activități sunt necesare pentru organizarea eficientă a procesului de formare::

Pasul 1: Identificarea utilizatorilor-cheie

În primul rând, sunt identificați „utilizatorii-cheie” – unul sau doi specialiști din fiecare unitate funcțională (împărțirea poate fi logică, nu neapărat organizațională).

Sprijinul acestor persoane este esențial în etapa următoare, în cadrul instruirii utilizatorilor finali, deoarece, pe lângă faptul că pot oferi observații privind utilizarea sistemului și necesitatea unor îmbunătățiri, experiența lor practică ajută ceilalți utilizatori să înțeleagă modul de lucru în sistem și să nu apeleze la echipa de suport a Furnizorului pentru orice întrebare.

Este important de menționat că, în funcție de amploarea și specificul sistemului, acești specialiști pot lucra nu doar cu partea de utilizare a sistemului, ci și cu deplasarea și configurarea independentă a modulelor funcționale, în funcție de cerințele utilizatorilor.

Din acest motiv, formarea, certificarea și implicarea lor activă trebuie să aibă loc mult înainte de etapa OP, deoarece majoritatea angajaților vor interacționa cu sistemul pentru prima dată abia în faza de operare pilot sau industrială, atunci când acești utilizatori-cheie vor deveni formatori și puncte de sprijin.

Cursurile de instruire pentru utilizatorii-cheie (pe domenii funcționale) sunt organizate de echipa Furnizorului și pot include certificare.

Forma de organizare poate fi:

- în format fizic (cu trainer din partea Furnizorului, la sediul Clientului sau Furnizorului);
- la distanță – în cazul în care prezența fizică nu este posibilă sau oportună.

Webinariile și teleconferințele devin din ce în ce mai populare, oferind eficiență în timp și costuri reduse..

Pasul 2: Instruirea utilizatorilor finali

Până la acest punct, este necesar ca un număr suficient de specialiști din echipa de implementare a Clientului să fie capabili să:

- instaleze și configureze componentele funcționale ale sistemului la stațiile de lucru;
- instruiască utilizatorii finali;
- ofere suport legat de configurarea standard;
- mențină sistemul în regim operațional.

Metodologia instruirii poate varia în funcție de dimensiunea companiei, proiectului sau preferințele Clientului. În practică, instruirile se desfășoară:

- în mai multe sesiuni;
- în grupuri mici (10–15 persoane);
- formate în funcție de rolul fiecărui tip de utilizator.

Temele abordate includ:

- Prezentare generală a obiectivelor sistemului;
- Navigarea și utilizarea interfeței;
- Funcționalități de bază;

- Funcționalități specifice rolului;
- Sesiuni de întrebări și răspunsuri.

Uneori, instruirea este urmată de configurarea parametrilor necesari direct la stațiile de lucru (ex. integrare cu conturi Outlook, setări de sincronizare), în funcție de caracteristicile aplicației.

Pasul 3: Perfecționare profesională / recalificare

Este important să fie luată în considerare necesitatea unor sesiuni de instruire periodice, organizate pentru dezvoltarea continuă a competențelor angajaților. Acestea pot fi împărțite, în mod condiționat, în trei categorii principale:

Instruire planificată după modificări ale sistemului – organizată atunci când sunt adăugate funcționalități noi, implementate module suplimentare sau introduse modificări majore în arhitectura existentă.

Reinstruire periodică – destinată perfecționării cunoștințelor angajaților care dețin deja competențe profesionale în utilizarea sistemului și au nevoie de actualizare în funcție de noile cerințe sau standarde interne.

Instruire la cerere – parte a activităților de suport pentru utilizatori în etapa de exploatare. Se desfășoară în situațiile în care un utilizator întâmpină o problemă specifică sau are nevoie de o sesiune metodologică aplicată.

Trebuie menționat că, în cazul anumitor tipuri de sisteme — în special cele asociate cu managementul unor obiective care, în caz de control deficitar, pot pune în pericol viața sau sănătatea oamenilor — simpla instruire nu este suficientă. În aceste situații, se organizează testări suplimentare de cunoștințe, iar accesul la sistem este permis doar pe baza promovării acestor evaluări, prin obținerea unui certificat personal de competență.

Exemple relevante includ sisteme informatice utilizate pentru:

- managementul siguranței traficului feroviar și aerian;
- lansarea de obiecte spațiale;
- controlul proceselor din industria chimică;
- alte activități cu risc ridicat.

În mod evident, la planificarea oricărui program de instruire trebuie avute în vedere și următoarele aspecte:

- experiența angajaților;
- nivelul de cunoștințe perceput și demonstrat;

- abilitățile de leadership, de management și de organizare.

Versiunea finalizată și testată a software-ului este lansată în producție. Serviciul de suport sau clientul colectează feedback de la utilizatori. Dacă în această etapă sunt identificate erori care nu au fost detectate în timpul testării, dar care corespund cerințelor produsului, acestea sunt transmise dezvoltatorilor pentru a fi remediate în următoarea etapă a ciclului de viață.

SUPORT ȘI MENTENANȚĂ

După lansare, proiectul intră în faza de mentenanță. În această etapă are loc o monitorizare continuă a funcționării corecte a produsului, a performanței sistemului, a securității și a riscurilor de învechire tehnologică.

Specialiștii tehnici remediază problemele apărute în utilizarea resursei software. Aceasta poate include tratarea erorilor reziduale care nu au putut fi corectate înainte de lansare, precum și rezolvarea noilor incidente semnalate de utilizatori. În cazul proiectelor de amploare, perioada de mentenanță este adesea mai extinsă decât în cazul proiectelor de dimensiuni mai reduse.

În cadrul procesului de suport, sunt evidențiate următoarele tipuri principale de activități:

- Suport de garanție
- Mentenanță corectivă
- Mentenanță evolutivă (de îmbunătățire)
- Mentenanță adaptivă

Suport de garanție

Reprezintă activitățile de suport și servicii garantate de dezvoltatorul sistemului în cazul apariției unor defecțiuni sau erori în perioada de garanție. Aceste intervenții sunt reglementate contractual, fiind realizate pe cheltuiala furnizorului și incluzând toate obligațiile asumate privind produsele livrate.

Mentenanță corectivă

Vizează corectarea erorilor identificate în timpul testării sau în utilizarea curentă. Utilizatorii se așteaptă ca problemele apărute („bug-uri”) să fie remediate prompt, iar pierderea feedbackului lor poate genera nemulțumiri. De aceea, acest tip de mentenanță este tratat cu o atenție deosebită.

Mentenanță evolutivă (de îmbunătățire)

Constă în adăugarea de funcționalități noi sistemului. Solicitățile provin de regulă de la utilizatori, însă și analiștii de business pot genera cerințe de îmbunătățire pe baza observațiilor din operare.

Mentenanță adaptivă

Implică modificări ale sistemului pentru a-l adapta la noi condiții hardware sau software. Schimbările de infrastructură, de interfață sau de structură a datelor pot impune refactorizarea unor componente pentru a le menține funcționale. Nu se extinde neapărat funcționalitatea, ci se asigură continuitatea acesteia într-un nou mediu operațional.

Aplicarea modificărilor presupune, în cele mai multe cazuri, intervenții directe în codul sursă și declanșează un lanț complet de activități de modernizare software – de la analiză, la testare și livrare către utilizatori. Pentru eficientizarea acestor procese sunt dezvoltate metodologii care reduc efortul de muncă, cum ar fi abordarea orientată pe obiect, larg utilizată în dezvoltarea modernă.

4. PROCESE

În cadrul metodologiei Waterfall, fiecare proces aplicat în ciclul de viață al unui sistem software necesită un timp considerabil. Structura acestuia poate fi împărțită în trei faze, conform reprezentării din figura de mai jos.

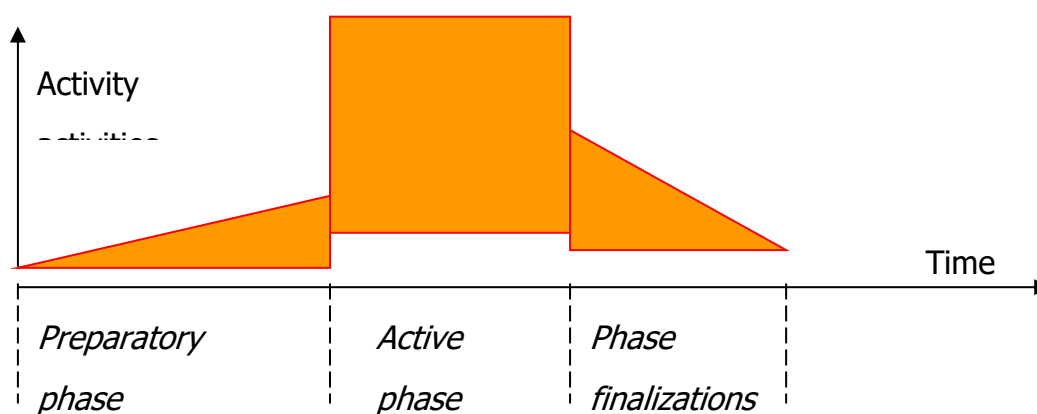


Figura - Structura de Execuție a Procesului

Faza de pregătire poate include familiarizarea cu domeniul de activitate, examinarea rezultatelor proceselor realizate anterior, precum și planificarea și calculele preliminare.

În timpul fazei active, procesul este executat propriu-zis. În descrierea proceselor din prezentul regulament tehnic, sunt acoperite exclusiv activitățile specifice fazei active.

În cadrul fazei de finalizare, pot fi desfășurate activități de analiză a rezultatelor, pot fi încheiate activități inițiate în faza activă, iar lucrările care nu sunt limitate de termenul final pot continua.

Pe baza acestor etape și procese, Furnizorul împreună cu Clientul elaborează un model de ciclu de viață al unui anumit sistem software.

Fiecare fază are un obiectiv distinct și contribuie la parcursul integral al ciclului de viață al sistemului software. Toate fazele sunt luate în considerare în procesul de planificare și execuție a acestui ciclu de viață.

Pentru implementarea fiecărei faze, organizația trebuie să dispună de:

- infrastructură adecvată,
- buget,
- componente software și hardware,
- instrumente tehnice,
- proceduri aprobate,
- resurse umane competente.

Toate aceste componente sunt planificate înainte de începerea fiecărei faze și sunt stabilite și/sau achiziționate, după caz.

Începutul și sfârșitul fiecărui pas și proces trebuie documentate în conformitate cu procedurile instituționale ale organizației.

Acest model presupune execuția strict secvențială și unică a fiecărei faze a proiectului. Tranziția de la o fază la alta este permisă doar după finalizarea cu succes a fazei precedente. Fiecare fază implică o planificare detaliată și asigurarea unei acurateți complete a rezultatului.

Această constrângere de secvențialitate strictă permite construirea unui proces de dezvoltare transparent și previzibil pentru Client.

5. PROIECTARE ȘI DOCUMENTARE

O particularitate esențială a acestei metodologii este necesitatea menținerii și actualizării constante a documentației de dezvoltare a produsului. Orice modificare trebuie coordonată cu Clientul. În absența acestui pas, un nivel insuficient de detaliere a cerințelor poate conduce la creșterea bugetului și extinderea termenelor proiectului — aspecte dificil de estimat corect în avans.

Reprezentantul autorizat al Clientului trebuie să fie activ implicat în procesul de dezvoltare. Acesta va avea trei responsabilități principale:

- Gestionarea portofoliului de produs. Product Owner-ul, împreună cu Business Analyst-ul, va actualiza regulat backlog-ul pentru a reflecta lista prioritizată a funcționalităților ce urmează a fi implementate.
- Răspuns la întrebările echipei de dezvoltare. Reprezentantul Clientului trebuie să fie disponibil pentru a oferi clarificări și a evita comunicarea excesiv formală care poate încetini proiectul. Pentru a livra un produs funcțional la finalul fiecărui sprint, este esențial ca echipa să aibă acces la informațiile necesare la momentul potrivit.
- Acceptarea livrabililor. La finalul fiecărui sprint, pachetele de lucru finalizate sunt prezentate Reprezentantului Clientului pentru acceptare. Acesta va valida rezultatele sau va transmite Furnizorului observații privind eventualele neconformități identificate, care trebuie rezolvate înainte de următorul sprint.

Deși nu este obligatoriu, Reprezentantul Clientului poate participa la ședințele echipei de dezvoltare pentru a urmări progresul și a reacționa prompt la eventuale blocaje.

Tot Reprezentantul Clientului va decide dacă produsul este gata pentru lansare conform planului de release.

Conform principiilor metodologiei de management de proiect, Clientul elaborează un concept de dezvoltare a produsului, precum și un roadmap pentru a urmări progresul și a asigura o evoluție coerentă a livrabililor.

Utilizăm această metodologie de dezvoltare datorită transparenței ridicate oferite în proces. Nivelul înalt de formalizare face ca gestionarea proiectului să fie mai facilă și predictibilă. Modelul în cascadă reduce riscurile și aduce claritate în derularea proiectelor complexe, mai ales în situațiile în care lucrează zeci de specialiști.

Adoptăm acest model în mod special în proiectele guvernamentale de mari dimensiuni, unde cerințele sunt clar definite și fixate de la început. Nu recomandăm utilizarea acestei metodologii pentru dezvoltarea de aplicații comerciale flexibile, orientate spre business.

6. ASPECTE POZITIVE ȘI NEGATIVE ALE METODOLOGIEI

Avantaje:

- Transparență ridicată a etapelor de dezvoltare și a fazelor proiectului
- Succesiune clară a activităților
- Stabilitate a cerințelor de-a lungul proiectului
- Control strict asupra managementului de proiect
- Facilitează elaborarea planului de proiect și formarea echipei de lucru

- Definirea clară a procedurii de control al calității

Dezavantaje:

- Proiectul trebuie să fie însoțit permanent de documentație actualizată. Actualizarea documentației este obligatorie și adesea redundantă
- Metodologia nu este suficient de flexibilă
- Clientul nu are posibilitatea de a interacționa din timp cu sistemul, de exemplu printr-o versiune pilot
- Utilizatorul final nu are ocazia să se acomodeze treptat cu produsul
- Toate cerințele trebuie cunoscute de la începutul ciclului de viață al proiectului
- Este necesar un management riguros și o monitorizare constantă; în caz contrar, proiectul riscă să iasă rapid din grafic
- Poate crea o impresie eronată asupra progresului proiectului (ex. formularea „45% complet” nu oferă valoare reală, fiind doar un indicator de stare pentru managerul de proiect)
- Nu permite remodelări pe parcurs – întregul proiect este dezvoltat într-o singură iterație

Dezavantajele de mai sus nu împiedică aplicarea cu succes a metodologiei Waterfall, în următoarele condiții:

- Clientul știe exact ce dorește să obțină. A analizat în prealabil conceptul, are o viziune clară asupra rezultatului final și nu intenționează să modifice cerințele în timpul dezvoltării.
- Clientul este de acord ca întregul proces de management al proiectului să fie asumat de către Furnizor. Nu își propune să participe activ în procesul de dezvoltare, control sau feedback, ci dorește doar să primească rezultatul final.
- Clientul este conștient de calendarul proiectului și de livrabilele așteptate la fiecare etapă.
- Echipa de dezvoltare cunoaște din timp tehnologiile și metodele de lucru ce urmează a fi utilizate și deține soluții standard pe baza cărora va livra produsul.

Modelul Waterfall este, de asemenea, o alegere potrivită în cazul în care echipa dezvoltă un produs complex, iar procesul de creare necesită respectarea unei secvențe stricte de pași și un buget substanțial.

7. UTILIZAREA METODOLOGIEI

Aplicăm această metodologie:

- Pentru proiecte de dimensiune medie și mare, care implică zeci de programatori și mai multe echipe de proiect distincte;
- În proiecte în care cerințele și limitele sunt clare, transparente și bine definite încă de la începutul ciclului de viață al proiectului.

Astfel, modelul în cascadă (Waterfall) presupune că tranziția de la o fază de dezvoltare la alta are loc doar după finalizarea completă și cu succes a fazei anterioare.

Modelul permite suprapunerea etapelor de implementare, dar nu presupune revenirea (backtracking) sau săritul etapelor – ordinea este secvențială, cu faze care se pot suprapune, dar nu inversa.

8. ROLUL FURNIZORULUI

Furnizorul este responsabil de gestionarea proiectului în conformitate cu planul de proiect și cu practicile agreeate împreună cu Clientul.

Responsabilitățile Furnizorului includ:

- Identificarea și mobilizarea resurselor necesare pentru desfășurarea activităților din aria sa de responsabilitate, conform planului de management al proiectului și nivelului de calitate convenit;
- Elaborarea documentației de arhitectură la nivel înalt pentru soluția IT propusă, care va include principalele aspecte de implementare: obiectivele de livrare, arhitectura sistemului, stack-ul tehnologic, infrastructura de implementare, componentele funcționale, machetele (mock-up-urile) principalelor interfețe etc.