

Managementul Calității

1 Introducere

Asigurarea calității (QA) este necesară pentru ca întregul Ciclu de Viață de Dezvoltare a Software (SDLC) să fie unul de calitate înaltă și este o etapă integrală și critică a fiecărui proiect de dezvoltare software-ului. Acest proces nu doar anticipează cerințele, dar și asigură un software sigur și stabil garantând urmarea tuturor standardelor și procedurilor din cadrul companiei.

Procesul de asigurare a calității constă în verificarea tuturor soluțiilor software dezvoltate și asigurarea că ele întrunesc specificațiile și așteptările Beneficiarului. Asigurarea calității este un aspect critic începând cu etapa de inițiere și finisând cu etapa de lansare a software-ului.

Calitatea este definită în cele din urmă de Beneficiarul final al sistemului și arată cât de aproape este de cerințele și scopul Proiectului.

Managementul calității în acest proiect se bazează pe recomandările PMBOK PMI și se concentrează pe 2 domenii:

- Managementul calității proceselor de proiect;
- Managementul calității sistemului (funcționalități dezvoltate).

2 Asigurarea și controlul calității proiectului

Pentru a asigura calitatea implementării contractului/proiectului, atât se implementează proceduri de lucru, se numesc persoanele responsabile și, se comunică echipa proiectului și se instruește eficient aceasta, precum și se stabilesc criteriile de monitorizare și control al proiectului.

Activitățile de asigurare a calității includ, dar nu se limitează la:

- Documentarea proceselor de proiect (procesele sunt acoperite de prezentul plan de management al proiectului și de regulile și dispozițiile privind documentația de licitație);
- Monitorizarea implementării proiectului prin raportarea obligațiilor, organizarea ședințelor, menținerea registrelor aferente;
- Efectuarea auditurilor de proces și ajustarea documentației de proiect, dacă este necesar;
- Implementarea modificărilor pentru eficientizarea anumitor procese;
- Aprobarea de către Beneficiar a documentației proiectului;
- Identificarea parametrilor de calitate ai proiectului care sunt determinați de:

- Buget: respectarea bugetului alocat proiectului (prin controlul bugetului alocat, a cheltuielilor și veniturilor);
- Timp: respectarea constrângerilor de timp în conformitate cu planul de implementare a proiectului (rapoarte de progres al proiectului, întâlniri dedicate);
- Lista livrabilelor pentru fiecare etapă a proiectului;
- Identificarea punctelor de control;
- Documente: asigurarea că documentația proiectului este înregistrată pe tot parcursul contractului, cele mai recente versiuni ale documentelor sunt comunicate echipei de proiect, starea și forma documentelor sunt bine definite, coordonarea/aprobarea documentelor se face în timpul alocat;
- Performanța echipei de proiect.

Controlul calității se realizează prin:

- Validarea și acceptarea livrabilelor;
- Monitorizarea programului de implementare;
- Analiza registrului de riscurilor, problemelor, documentelor.

Rezultatele controlului calității vor fi analizate în timpul ședințelor de calitate, unde vor fi formulate recomandări pentru remedierea unui proces ineficient (dacă sunt identificate) și registrele legate de proiect vor fi finalizate, procesele defecte vor fi actualizate (ajustarea planului de comunicare, plan de management al riscurilor, ajustarea planului de instruire, ajustarea planului de gestionare a resurselor etc.).

2.1 Responsabilul

#	Responsabil	Activități
1	Coordonatorul proiectului	Oferirea sprijinului în procesul de planificare, control și planificare
2	Managerul de proiect	Planificarea, asigurarea și controlul calității proiectului; Asigurarea respectării regulilor enumerate în documentația proiectului; Controlul versiunilor documentelor și fluxului de documente; Controlul raportării; Acordarea de sprijin metodologic și coaching echipei de proiect;

	Revizuirea, coordonarea și aprobarea proceselor și documentelor cu Beneficiarul.
--	--

3 Asigurarea și controlul calității software-ului

Scopul managementului calității sistemului constă în capacitatea sistemului de a îndeplini scopul Proiectului.

Pentru a asigura calitatea sistemului, Furnizorul, împreună cu Beneficiarul, va întreprinde următoarele acțiuni:

- Actualizarea planului de testare și acceptanța, inclusiv scenarii de testare, în care vor fi specificate criteriile de calitate a sistemului.
- Elaborarea documentației sistemului atât a proiectului tehnic care este un document la nivel de sistem, cât și a instrucțiunilor pentru utilizarea și gestionarea sistemului pentru a asigura continuitatea și eficiența sistemului.
- Instruirea utilizatorilor și administratorilor beneficiarului pentru a asigura o bună înțelegere a sistemului.
- Asigurarea întreținerii și asistența după punerea în funcțiune a sistemului (inclusiv instruirea utilizatorilor beneficiari pentru a utiliza aplicația Help Desk).
- Comunicare activă între echipele de proiect și alte părți implicate.

Controlul sistemului se realizează prin testarea continuă a sistemului de către dezvoltatori, echipa de testare și echipa de utilizatori cheie a Beneficiarului. Rezultatele testelor sunt incluse în jurnalul de erori și discutate de ambele părți pentru a lua deciziile corecte.

În urma testării, se pot face recomandări pentru îmbunătățirea sistemului fie prin eliminarea erorilor, fie prin eficientizarea proceselor proiectului sau prin ajustarea scopului proiectului.

Managerul de testare va monitoriza funcționalitatea sistemului pentru a se potrivi cu documentația aprobată. Erorile identificate vor fi monitorizate până când vor fi rezolvate.

3.1 Responsabilul

Fiecare participant a proiectului este responsabil să acționeze în conformitate cu scopul proiectului din caietul de sarcini, contract și documentația proiectului.

Responsabili pentru managementul calității proiectului sunt:

#	Responsabil	Activități
1	Managerul de proiect	Planificarea procesului de testare, acceptare, instruire
2	Coordonatorul tehnic	Training și suport metodologic; Dezvoltarea proceselor de testare și control SI.
3	Managerul de testare/Asigurare calitate	Organizează echipa de testare și asigură procesul de testare conform planului; Avizarea rapoartelor de testare; Informează managerului de proiect cu privire la rezultatele testelor și livrarea rapoartelor de testare a managerului de proiect.

4 Monitorizarea și raportarea progreselor

4.1 Principiile de bază

Monitorizarea este o analiză regulată a resurselor, activităților și rezultatelor proiectului. Procesul de monitorizare se bazează pe un sistem coerent de colectare a informațiilor (rapoarte, analize și indicatori).

În cadrul acestui proces, managerul testare va monitoriza performanța și va identifica permanent variațiile din planul de proiect. Nu în ultimul rând, va controla și modificările făcute și va recomanda acțiuni preventive dacă anticipează posibilele probleme care pot apărea.

Factorii cheie care vor contribui la succesul procesului de monitorizare și control sunt: planificarea;

- măsurarea performanței;
- gestionarea riscurilor;
- analiza varianței
- acțiuni corective.

Ofertantul va utiliza sistemul de management al proiectului implementat în companie pentru documentarea și urmărirea lucrărilor efectuate în raport cu planul de proiect și va măsura performanța obținute peste cheltuielile suportate. Aceasta înseamnă că revizuirile, auditurile și reuniunile privind starea proiectului vor fi efectuate periodic.

În cadrul acestor orientări, se vor face următoarele:

- Evaluarea riscurilor, analiza riscurilor, gestionarea a riscurilor;
- Asigurarea calității;
- Analiza variațiilor din calendar și bazele de costuri (financiare);
- Documentarea acțiunilor corective pentru fiecare variație întâlnită (fiecare acțiune corectivă este stabilită, înregistrată și urmărită până la finalizare).

4.2 Ce monitorizăm și cum

Instrumentul de bază utilizat de QA pentru a monitoriza și evalua iterativ cu scopul asigurării că proiectul progresează în conformitate cu planurile prestabile sunt următoarele acțiuni: monitorizarea, evaluarea și controlul proiectului.

Monitorizarea proiectului include activități precum:

- Colectarea de informații despre dezvoltarea proiectului și factorii de influență (echipă, context);
- Diseminarea concluziilor evaluării și a modificărilor aduse echipei.

Managerul de proiect colectează/compară informații despre:

- starea proiectului în raport cu planul de proiect;
- costurile suportate în legătură cu planul;
- evoluția factorilor de risc;
- evoluția contextului proiectului;
- cantitatea și calitatea muncii efectuate de fiecare membru al echipei;
- îndeplinirea sarcinilor de către subcontractanți/parteneri;
- coeziunea și comunicarea în cadrul echipei.

Modul în care managerul de proiect colectează și analizează informații va fi:

- ședințe de proiect;
- cu ajutorul software-ului de gestionare a proiectelor, a sistemului Ticketing, Știri, forumuri;
- utilizarea listelor de discuții;
- utilizarea rapoartelor de testare;
- prin comunicare directă cu membrii echipei.

În urma monitorizării, managerul de proiect va avea informațiile necesare pentru a evalua proiectul și, și după comunică echipei rezultatele evaluării și deciziile luate.

Recomandări pentru faza de diseminare a evaluării și concluziile deciziei:

- evitarea comparațiilor între membrii echipei sau propunerea de modele sau anti-modele diferite;
- corespunde întotdeauna unui model sau anti-model, fii un tip generic de membru al echipei, conform standardelor de lucru;
- toți membrii implicați în proiect trebuie tratați în mod egal și nediscriminatoriu - în caz contrar, membrii echipei se pot simți nereprezentativi, dezavantajați, nedoriti;
- Se va oferi feedback constant membrilor echipei pentru a fi conștienți de situație, pentru a putea acționa rapid prin a fi informați și pentru a nu fi tensionați la audituri regulate.

4.3 Evaluarea proiectului

Evaluarea proiectului:

- Este efectuată în principal de către QA sub îndrumarea directă a managerului de proiect. Vor contribui membrii echipei, echipa de management a Beneficiarului și subcontractanții;
- Are loc după etapa de colectare a informațiilor de monitorizare;
- Evaluarea proiectului se referă la:
 - determinarea stării proiectului în raport cu planul proiectului (respectarea termenelor, calitatea proiectului și respectarea planului bugetar);
 - evaluarea echipei.

Planul de proiect include următoarele:

- WBS detaliat ale proiectului (asocierea activităților - resurse - durata);
- Planificarea temporală a activităților WBS (diagrama Gantt);
- Planul de risc al proiectului (care sunt riscurile, cum pot fi prevenite, planuri de acțiune în cazul apariției lor).

Pentru a evalua obiectiv și precis starea proiectului, valorile estimate și valorile reale (STD) vor fi calculate în termeni de timp și cost.

4.4 Planul de control a proiectului

Urmărirea planului

Progresul calendarului proiectului se desfășoară în mod continuu în cadrul proiectului și este raportat în mod regulat beneficiarului. Managerul de monitorizare și evaluare colectează informații despre progresul la fiecare nivel de sarcină, care sunt apoi comparate cu calendarul de realizare a sarcinilor. În plus, Managerul de proiect va identifica variațiile actuale sau viitoare pentru toate etapele proiectului.

Raportarea performanței

Informațiile privind starea proiectului sunt măsurate în raport cu planul de implementare planificat, iar rezultatul acestor măsurători este raportat în rapoartele de progres.

Revizuirea calendarului

Dacă QA identifică o problemă în planul de implementare a proiectului, el / ea va efectua următoarele activități cu echipa:

- Identifică nivelul problemei;
- Analizează impactul acesteia asupra planului de implementare;
- Identifică alternativele posibile, analizând riscul asociat fiecărei alternative.

În urma acestor activități, QA recomandă și apoi implementează acțiunile corective necesare.

Monitorizarea variațiilor de progres

Progresul actual poate fi diferit de progresul planificat din mai multe motive. QA are responsabilitatea de a identifica abaterile de la planul de proiect, cauzele acestor abateri, tendințele la nivelul fiecărei sarcini și de a corecta aceste abateri.

Rezolvarea variațiilor în curs

Odată ce o variație de calendar a fost identificată și cuantificată, QA are câteva sugestii pentru rezolvarea acesteia. În funcție de cauza abaterii, acțiunea poate fi: realocarea resurselor, reatribuirea sarcinii sau grupurilor de sarcini sau corectarea performanței echipei.

Urmărirea acțiunilor corective

Acțiunile corective sunt atent monitorizate pentru a se asigura că acestea ajută la recuperarea abaterii de la planul de implementare înainte de atingerea termenului limită programat sau înainte de propagarea abaterii către planul de implementare.

4.5 Raportarea progresului în proiect

Conform metodologiilor de lucru descrise în această ofertă tehnică, au fost stabilite mecanisme de comunicare pentru a se asigura că informațiile necesare sunt generate și utilizate într-un mod eficient. În acest context:

- Analizele de progres sunt utile pentru a revizui progresul comparativ cu planul. Aceasta poate fi, de asemenea, o oportunitate pentru prezentarea și discutarea rapoartelor scrise sau pentru o evaluare orală a problemelor și problemelor actuale;
- Rapoartele de progres ale proiectului oferă un rezumat al progresului proiectului, inclusiv informații cheie din indicatorii fizici și financiari, inclusiv în graficele logice și de activitate și în graficul resurselor.

Rapoartele de progres vor fi elaborate într-un format standard care să permită compararea rapoartelor în timp. Scopul rapoartelor de progres va fi de a oferi actualizări de performanță în comparație cu etapele de referință.

Ofertantul va pregăti rapoarte pe toată durata contractului. Rapoartele vor acoperi toate activitățile proiectului și vor înregistra toate rezultatele obținute de furnizor.

Raportarea progresului lunară sau săptămânală

Aceste rapoarte vor include principalele progrese ale perioadei raportate, evoluția activităților și întârzierile, dacă acestea sunt semnificative, dificultățile întâmpinate, abaterile de la planul de activitate și consumul de resurse. Fiecare raport de activitate va fi însoțit de un raport financiar care detaliază alocarea experților/activității.

Beneficiarilor li se vor furniza, la cerere, rapoarte intermediare intermediare privind problemele specifice identificate de reprezentanții lor.

Rapoartele vor oferi, de asemenea, informații despre rezultatele fiecărui pas, soluțiile date și deciziile majore care trebuie luate în considerare atunci când se iau în considerare aspectele concrete ale activității.

Raportul final

Versiunea preliminară a Raportului va fi trimisă Beneficiarului cu cel puțin o lună înainte de sfârșitul perioadei de executare a contractului pentru analiză. Acesta va descrie întregul proces de implementare și va facilita evaluarea rezultatelor obținute atât în termeni calitativi, cât și cantitativi.

Raportul va include:

- evaluare a succesului și a constrângerilor majore pentru fiecare activitate și sarcină;
- realizările generale ale contractului;
- evaluare a realizării rezultatelor propuse în contract;
- recomandări pentru acțiuni viitoare în vederea asigurării durabilității activităților, a rezultatelor așteptate după finalizarea contractului și a măsurilor care trebuie luate de beneficiar în acest sens.

5 Cele mai bune practici – îmbunătățirea calității în orice aspect

Pentru a asigura și a îmbunătăți la maximum calitatea fiecărui proiect, inginerii Indrivo Software adoptă multe bune practici în fiecare aspect al proiectelor noastre, practic putem separa aceste practici în trei categorii: cele mai bune practici pentru programare, cele mai bune practici pentru testare și cele mai bune practici la nivel funcțional.

5.1 Cele mai bune practici pentru Programare

Aceste practici se concentrează în principal pe îmbunătățirea calității codurilor sursă, ceea ce va duce la produse mai ușor de întreținut. Aceasta înseamnă mai puține erori și mai puține costuri de întreținere.

Unit Test

Unit Test constă în scrierea unor fragmente de cod pentru a valida corectitudinea funcționalităților (cum ar fi funcții și clase). Acestea formează o rețea de siguranță pentru

prevenirea modificărilor neintenționate ale codului. Astfel, odată ce apare o eroare, testele de unitate pot ajuta programatorul să găsească problema și să o repare foarte rapid, având mai puțin efort de testare. Testele de unitate se efectuează cu acordul Beneficiarului.

Dezvoltare pe baza de test – TDD (Test Driven Development)

Test Driven Development cere dezvoltatorilor să scrie teste înainte de implementarea funcționalităților. Astfel, dezvoltatorii vor fi forțați să se gândească din punctul de vedere al utilizatorilor și vor face un design mai circumspect până a fi transpus în cod. Pe măsură ce problema de proiectare este evitată de TDD, complexitatea codurilor va fi păstrată la un nivel minim.

Revizuirea codului (Code Review)

Revizuirea codului este o practică demonstrată, care îmbunătățește considerabil calitatea codului. Prin revizuirea codului, multe probleme potențiale vor fi identificate, deviațiile în standarde vor fi corectate, și cel mai important, dezvoltatorii vor face schimb de experiență, iar toate împreună vor contribui la calitatea integrală a software-ului.

Refactor

Refactor este o modalitate disciplinată de a curăța codul și de a îmbunătăți designul. Refactoring-ul păstrează codurile ordonate și minimizează introducerea erorilor. Refactoring-ul este parte a fluxului de dezvoltare și toate proiectele necesită un refactoring a codului atunci când sunt efectuate modificări.

Compilarea zilnică (Daily Build)

Compilarea zilnică este o practică de a compila software-ului în fiecare zi. Astfel este minimizat riscul integrării și sunt evitate multe erori care ar apărea în integrarea pachetelor de software. Compilarea zilnică asigură faptul că dezvoltatorii se sincronizează, astfel încât orice defecte cauzate de neînțelegerea și lipsa de comunicare să poată fi rezolvate ușor și rapid.

5.2 Cele mai bune practici în Testare

Aceste practici se concentrează pe activități de testare, prin implementarea unor teste eficiente și eficace, calitatea produselor va fi garantată la un cost rezonabil.

Autotestare

Așa cum este indicat în fluxul de lucru de mai sus, fiecare programator din Indrivo ar trebui să facă auto-testul conform unei liste de verificare înainte de a comite modificarea. Scopul este de a asigura că rezultatele dezvoltatorilor sunt acceptabile. Prin auto-testare eficientă, multe probleme pot fi descoperite și remediate într-o etapă anterioară, cu costuri mai mici. Următoarele activități de testare vor beneficia, de asemenea, de acest lucru.

Testare încrucișată

Testarea încrucișată înseamnă că A testează ieșirile lui B în timp ce B testează ieșirile lui A. Întrucât toată lumea are punctele oarbe, testele încrucișate vor adăuga valoare calității software-ului. Un alt motiv pentru care testele încrucișate funcționează se datorează motivului psihologic că oamenii nu le plac greșelile să fie găsite de alții, vor fi mai prudenți dacă echipa adoptă testele încrucișate. În Indrivo, testarea încrucișată este o practică sugerată, dar multe echipe au adoptat-o pentru că au încercat-o și le merge bine.

Testare regresivă

Pentru a efectua un test de regresie reușit, toate cazurile de testare sunt rulate și trecute, fiecare modul al acestui software va fi testat complet. Procedând astfel, depistăm majoritatea problemelor (poate peste 95%) ascunse în software, reducând astfel șansele ca o problemă să fie lăsată în afara utilizatorilor. Oferă calității produsului final o garanție generală. În Indrivo, testarea regresivă va fi implementată atunci când o versiune urmează să fie lansată, echipa ar putea alocă două sau trei săptămâni pentru testare și remediere a erorilor, dar depinde de dimensiunea proiectului.

Testarea integrărilor

Testarea integrărilor este testarea implementată în timpul integrării, va asigura că integrarea nu va sparge softwareul și este, de asemenea, utilizată pentru a vă asigura că modulele aferente nu vor fi influențate de această integrare. Deoarece echipele Indrivo implementează integrarea continuă și construcțiile zilnice, vor implementa teste de integrare numai atunci când este necesar, executând o listă de verificare predefinită.

Testarea fluxurilor

Testarea adecvată a unui flux de lucru poate face activitățile de testare mai eficiente și mai eficace, ceea ce la rândul său îmbunătățește calitatea software-ului.

Clarificarea cerințelor de calitate

Primul pas este identificarea cerinței de calitate. Testerii Indrivo vor analiza foarte atent cerințele furnizate de client și vor comunica întrebările Beneficiarilor softwarelui, vor confirma, de asemenea, înțelegerea cerințelor de calitate (cum ar fi cerințele de performanță și cerințele de securitate etc.) înainte de a lua măsuri suplimentare.

Crearea strategiilor și planului de testare

Odată ce cerințele de calitate sunt clare, vor fi stabilite strategii de testare. De exemplu, ce tip de testare (cum ar fi testarea performanței, testarea securității, testarea instalării / dezinstalării, testarea compatibilității) trebuie implementată în acest proiect, frecvența cu care va fi executat fiecare tip de testare, când să le efectuați și cum să măsurați rezultatul etc. Apoi se va face un plan de testare pentru a ghida activitățile de testare primite.

Crearea cazurilor de testare

Pentru fiecare caracteristică (user story sau specificație) a software-ului, testerii noștri vor proiecta unele cazuri de testare, aceste cazuri vor indica modul în care fiecare caracteristică va fi testată și care vor fi rezultatele așteptate. Acestea vor fi utilizate atât în etapa de execuție a testului, cât și în testarea de regresie. Este foarte frecvent ca testerii să găsească multe probleme potențiale în timp ce proiectează cazuri de testare.

Efectuarea testului

La această etapă, planul de testare va fi executat. Aceasta implică o mulțime de activități, cum ar fi pregătirea mediilor necesare, pregătirea datelor și instrumentelor de testare, executarea cazurilor de testare și așa mai departe.

Raportarea erorilor

Dacă testerii găsesc erori în pasul anterior, vor înregistra erorile în sistemul de gestionare a erorilor (de exemplu, Mantis / Bug Net / Bug Tracker). Astfel dezvoltatorii le pot reproduce și repara.

Verificarea erorilor

Când bug-urile sunt remediate, starea lor va fi schimbată în „fix / rezolvat” și vor fi repartizate testerilor, apoi testerii trebuie să le verifice pentru a se asigura că aceste bug-uri nu mai există.

Analiza erorilor

Analizând starea și tendințele erorilor, veți avea o mai bună înțelegere a situației calității. Această activitate va genera niște informații valoroase, astfel încât să puteți decide dacă planul original trebuie ajustat. Indrivo efectuează o analiză amănunțită a erorilor atât în timpul cât și după faza de dezvoltare.

Rafinarea procesului de testare

La sfârșitul fiecărei iterații sau după lansarea unei versiuni majore, testerii Indrivo vor analiza dacă procesele și activitățile actuale de testare sunt suficient de eficiente și apoi le vor îmbunătăți. Procesele și activitățile rafinate vor fi aplicate în faza următoare, pentru a obține o calitate superioară.

5.3 Cele mai bune practici la Nivel Funcțional

Aceste practici sunt executate în mare măsură la nivel funcțional și nu pot fi sortate în activități de dezvoltare sau de testare, dar ajută la asigurarea calității.

Prototiparea

Deși scopul principal al realizării prototipului este comunicarea și confirmarea cerințelor cu clienții, calitatea va beneficia, de asemenea, foarte mult prin practicarea acestui lucru. Deoarece multe probleme de funcționalitate lipsă, probleme de neînțelegeri și probleme de experiență slabă a utilizatorului vor fi identificate și rezolvate foarte devreme, prototiparea evită o mulțime de bug-uri și economisește foarte mult costul calității. De asemenea, contribuie la calitate prin stabilirea unor criterii pentru sarcinile viitoare. Toate echipele Indrivo adoptă un fel de tehnologie de prototipare, unele dintre aceste prototipuri sunt realizate de clienții noștri, iar unele dintre ele sunt realizate de echipa Indrivo.

Lista Funcționalităților

Lista funcționalităților este o listă organizată care include toate caracteristicile software-ului și care este revizuită atunci când se schimbă cerințele. Deoarece lista funcționalităților poate fi utilizată în practicarea testelor de auto-testare și de regresie, aceasta va reduce la minim

șansele de a pierde ceva și, de asemenea, va ajuta echipa la descoperire problemelor inconsecvente.

Scrum review

Scrum Review este o practică provenită din metodologia Scrum, se practică la sfârșitul fiecărui Scrum Sprint (iterație). Echipa trebuie să demonstreze software-ul de lucru tuturor părților interesate odată ce au obținut vre-un livrabil, această activitate obligă echipa să livreze ceva care funcționează cu adevărat, mai degrabă decât aproximativ 90% din lucrări sunt deja făcute. De asemenea, oferă altor părți interesate șansa de a-și face contribuțiile, ceea ce poate îmbunătăți și calitatea. Este o practică sugerată în Indrivo, unele echipe au adoptat acest lucru și funcționează bine.

6 Măsuri de sustenabilitate

Utilizarea corectă a tehnicilor de management al proiectului va asigura îndeplinirea criteriilor esențiale de succes ale proiectului, astfel încât acesta să fie relevant, fezabil și durabil.

Sustenabilitatea execuției proiectului va fi realizată în modul următor:

- Proiectul este finalizat în termenii definiți în această ofertă;
- Proiectul este finalizat pe cheltuiala bugetului alocat;
- Livrabilele vor corespunde specificațiilor tehnice conform documentației de analiză;
- Produsul respectă cerințele de performanță funcțională;
- Procesul de management al proiectului este, de asemenea, adecvat în conformitate cu standardele;
- Riscurile proiectului sunt gestionate în mod adecvat;
- Cooperarea dintre Beneficiar și Ofertant este în conformitate cu așteptările;
- Proiectul respectă standardele de calitate;
- Produsul îndeplinește cerințele Beneficiarului;
- Produsul rezolvă problema Beneficiarului;
- Proiectul creează un impact economic pozitiv asupra companiei;
- Proiectul corespunde obiectivelor și sferei de implementare a proiectului.