

Tema PSSC

Domain-Driven Design (DDD)

Designul unui sistem determină în mare măsură gradul de complexitate pe care îl poate atinge o aplicație. Atunci când complexitatea unui sistem este mult prea ridicată, acesta poate crea dificultăți de gestionare deoarece orice fel de schimbare sau încercare de extindere se poate transforma într-o sarcină dificilă.

Conceptul de **domain-driven design** (DDD) vine în întâmpinarea noastră pentru a ne plasa atenția în centrul aplicației, punând accentul pe complexitatea intrinsecă a domeniului de business. De asemenea, distingem domeniul de cod (unic pentru business) de subdomeniile care îl susțin (de obicei generice în natură, cum sunt banii sau timpul) și ne concentrăm eforturile asupra nucleului aplicației.

Premisele după care funcționează conceptul de domain-driven design sunt următoarele: fiecare aplicație software ar trebui să fie bazată pe un model, iar modelul ar trebui să fie bazat pe un domeniu.

DDD nu este o metodologie, ci mai mult un set de reguli despre designul arhitectural al software-ului, oferind o structură de practici pentru luarea deciziilor de proiectare care ajută în proiecte software care au domenii complicate. Aceasta abordare a fost introdusă de Eric Evans. În principal, designul bazat pe domenii se concentrează pe:

- Domeniul principal și logica domeniului
- Proiecte complexe pe modele ale domeniului
- Îmbunătățirea modelului de aplicație și rezolvarea problemelor legate de domenii emergente prin colaborarea cu experții în domeniu.

Atâta timp cât tehnologia infrastructurii este bine concepută, nu are importanță dacă designul nu cuprinde complexitatea domeniului. Un design de succes trebuie să se ocupe sistematic de acest aspect central al softului. Fiecare aplicație se referă la o activitate sau un domeniu de interes pentru utilizator. Aria din care face parte obiectul de interes al programului folosit de utilizator reprezintă domeniul aplicației.

În general vorbind, DDD este destul de util atunci când vine vorba de a ajuta echipa să creeze un model comun. Echipetele din partea unei companii și din partea dezvoltatorului pot folosi acest model pentru a comunica despre cerințele afacerii, entitățile de date și modelele de proces. Acest limbaj comun necesită a fi creat atunci când se lucrează cu modele de domeniu pentru a ușura munca de ambele părți. După definirea unui model de domeniu și după ce un limbaj comun este creat, scopul DDD este de a realiza o conexiune între modelul de domeniu și cod, conexiune care va fi menținută pe întreaga durată a procesului de dezvoltare a proiectului. Dacă este folosită corespunzător, această legătură asigură un proces bun de dezvoltare, iar în momentul în care implementarea este modificată, aceste schimbări se vor reflecta și în modelul de domeniu și viceversa. Acest proces face ca proiectul să fie mereu la curent cu ultimele schimbări din cadrul dezvoltării.

Domain-driven design oferă o abordare foarte bună pentru procesul de dezvoltare de zi cu zi, deoarece oferă un set bun de reguli și procese care pot ajuta la dezvoltarea mai rapidă și menținerea mai ușoară a unei aplicații software.