

# Tema PSSC

**S.O.L.I.D** este un acronim pentru primele 5 principii de proiectare orientate pe obiecte ( OOD ). Aceste principii ajută un programator să dezvolte programe mai ușor de întreținut și extinse. De asemenea facilitează un developer să factorizeze codul mai ușor și să lucreze în modul de dezvoltare de software agile.

S.O.L.I.D înseamnă:

- **S - Single-responsibility principle:** o clasă ar trebui să aibe un singur motiv pentru a se schimba, iar aceea clasă să facă un singur lucru.
- **O - Open-closed principle:** obiectele sau entitățile ar trebui să fie deschise pentru extensi, dar sunt închise pentru modificări.
- **L - Liskov substitution principle:** subclasa / copil derivate ar trebui să fie substituibilă pentru clasa de baza / părinte.
- **I - Interface segregation principle:** un client nu ar trebui niciodată obligat să implementeze o interfață pe care nu o folosește sau clienții să nu fie forțați să depindă de metodele pe care nu le utilizează.
- **D - Dependency Inversion Principle:** entitățile trebuie să depindă de abstractizări, nu de concreții. Acest lucru înseamnă că modul de nivel înalt nu trebuie să depindă de modul de nivel scăzut, dar ele ar trebui să depindă de abstractizări.

**Arhitect Software** are un rol important deoarece acesta trebuie să interacționeze cu clienții, cu managerii de produse și cu develop-eri pentru a înțelege, modela și oferi modele inițiale care pot fi construite. Arhitectul trebuie să revizuiască în mod constant codul pentru a asigura calitatea design-ului, evitând complexitatea. Acestea necesită, de obicei, lucrări practice în ceea ce privește dezvoltarea de prototipuri, contribuirea la tehnologiile de cod și de evaluare.

Un alt rol al acestuia este de a îndruma alte persoane după cum este necesar și un grad de modestie. O astfel de colaborare permite arhitectului să se familiarizeze cu abilitățile și interesele din echipa și să-și împărtășească cunoștințele cu restul echipei. Modestia este necesară pentru a se asigura că toată echipa este ascultată, deoarece poate avea o experiență sau o cunoaștere mai specifică pentru o anumită problemă.

**Software architecture** se referă la structurile de nivel înalt ale unui sistem software. Fiecare structură cuprinde elementele software, relațiile dintre ele și proprietățile elementelor și relațiilor. Arhitectura unui sistem software este o metaforă făcând analogie cu arhitectura unei clădiri. Aceasta funcționează cu un model al sistemului și al proiectului în curs de dezvoltare, care stabilește sarcinile necesare pentru a putea fi puse în execuție de către echipa de proiectare / dezvoltare. Arhitectura software vizează realizarea unor alegeri structurale fundamentale, care sunt costisitoare de schimbat odată implementate. Alegerea arhitecturii software include opțiuni structurale specifice de la posibilitățile de proiectare a software-ului.