

Tema 1- PSSC

1.Principiile SOLID

Sunt specifice organizării agile. Este important ca codul să fie structurat după aceste principii, fiindcă atunci când intervine vreo modificare, aceasta să poată fi realizată fără mari dificultăți.

1.Principiul Singurei Responsabilități(SRP) presupune faptul ca o clasă trebuie să aibă un singur motiv pentru care să fie modificată. Fragilitatea este specifică acestui principiu, și înseamnă că o modificare asupra sistemului produce o ruptură design-ului.

2.Principiul Deschis-Închis(OCP) susține faptul că: modulele, clasele și funcțiile trebuie să fie deschise pentru extensii, dar închise pentru modificări. Totodată trebuie refactorizat design-ul astfel încât modificările ulterioare să nu schimbe codul existent.

3.Principiul de substituire Liskov(LSP) face referire la tipurile de bază. Acestea trebuie să poată fi substituite de tipurile derivate.

4.Principiul inversării dependenței(DIP) presupune ca modulele de pe nivelurile ierarhie superioare să nu depindă de cele de pe nivelurile inferioare. De asemenea, detaliile trebuie să depindă de abstractizare, nu invers.

2. Arhitectura software

Definiție: Este rezultatul unui set de decizii de afaceri și tehnice. Schița este influențată de mai multe părți, astfel încât execuția va suferi modificări. Arhitectul trebuie să le cunoască.

Scop: perspectiva generală asupra structurilor, componentele și cerințele importante ale proiectului.

Nivelele de arhitectură: 1. Arhitectura de Întreprindere->arhitecturile de mai jos; 2.Arhitectura software->definirea și relațiile dintre componente; 3. Arhitectura hardware->CPU, spațiu de stocare,mediu de transfer; 4. Arhitectura organizațională->relațiile de business, structura organizațională; 5.Arhitectura informațională->accesibilitatea ușoară a datelor.

3.Indicatori de calitate

Se folosesc pentru stabilirea calității unui produs software.

Exemple:

1.Complexitatea metrică este un indicator deosebit de important. Măsoară mentenabilitate codului, care este un element esențial pentru calitatea software.

2.Statului de stabilitate măsoara zilele de la ultimul build failure până la cel actual. Acest factor ajută echipa înainte ca acesta să ajungă în faza de testare.

3.Acoperirea codului

4.Statistici privind accidentele: este special pentru aplicațiile smartphone-urilor.
Monitorizează aplicația și transmite datele.

5.Distribuția majoră a problemelor: poate diminua costul bug-urilor.

6.Proporția automatizării testelor: măsurarea este raportul dintre testele automate și toate cazurile de testare definite(pe categorii de caracteristici).