

Predmet : Razoj Softvera 2

Gorka Java

Gorke priče

Student

Pejčić Biserka

Profesor

Filipović Vladimir

Sadržaj prezentacije

- Strašne priče u programiranju
- Tehnike za traženje i rešavanje antipaterna
- Primeri antipaterna u drugim indusrtijama

Antipaterni u životu

Da bih uspjeli da prevaziđemo probleme neophodno je primeniti 3 pravila:

- Naučiti da koristimo alate i tehnike stručnjaka – posećivanje konferencija radi učenja najbolje prakse. Novo okruženje i alati koji će načiniti razliku u projektima.
- Raditi isto što i stručnjaci – korišćenje obrazaca koji su detaljan uspešan nacrt za teške arhitektonske probleme.
- Učiti na greškama ljudi pre mene – dobri planovi mogu poći po zlu i bitno je znati reagovati.

Korišćenje dizajniranih obrazaca naglašava pozitivno

Dizajnirani obrazci su rešenje za nastale probleme u datom kontekstu. Dobar dizajnirani obrazac treba da predstavlja rešenje koje je uspešno implementirano nekoliko puta.

Razlozi za upotrebljavanje dizajniranih obrazaca:

- Dokazani dizajnirani obrazci umanjuju rizik. – korišćenjem dokazanog nacrtu za rešenje povećavam svoje šanse uspeha
- Dizajnirani obrazci štede vreme i energiju. – efikasno korišćenje vremena i truda drugih za rešavanje teških problema
- Dizajnirani obrazci poboljšavaju veštinu i razumevanje - Korišćenjem dizajniranih obrazaca mogu da poboljšam svoje znanje o domenu i traženje novih načina predstavljanja kompleksnih modela.

UML se koristi kao jezik za obrazce

UML skuplja zajedno nekoliko alata koji podržavaju objektno orjentisan jezik.

Koncepti:

- Scenario (slučajevi upotrebe)
- Povezanost klasa (klasni dijagrami)
- Povezanost objektnih interfejsa (dijagrami sekvence)
- Stanje objekta (dijagrami stanja)

Antipaterni se uče iz negativnih iskustava

Antipaterni – književna forma koja objašnjava često korišćena rešenja problema koja prouzrokuje odlučno negativne posledice.

Svako ko prati softversko inženjerstvo izbliza zna da veliki procenat softverskih projekata propada.

Neupoznatost sa istorijom softverskog inženjerstva ili eksponencionalni rast troška popravljavanja problema u ciklusu razvojnog procesa, možemo da zavaravamo sebe misleći da sad uzmemo prečicu a popravljamo kaskade.

Neki dobro poznati antipaterni

Neki primeri dobro poznatih antipaterna:

- Slatke prečice – optimizacija praznog prostora.
- Optimizacija po ceni čitljivosti – u većini slučajeva čitljivost globalno je mnogo bitnija nego optimizacija.
- Dupliraj programiranje – širenje grešaka. Kopije kodova su manje testirane od originala.
- Korišćenje pogrešnog algoritma za rešavanje – algoritmi sortiranja
- Korišćenje pogrešne klase za rešavanje – b-tree heš tabele i nizovi

Načini razumevanja antipaterna

Antipaterni pokušavaju da odrede koje greške se često dešava, zašto su napravljene i koje ispravke u procesu ih mogu sprečiti.

Bitni koraci u razumevanju antipaterna:

- Pronalaženje problema – greška, algoritam loših performansi ili nečitljivi metod.
- Ustanoviti obrazac neuspeha
- Refaktorisanje koda sa greškama
- Objavljivanje rešenja –učiti druge ljude o svojim pronalascima
- Ustanoviti slabosti procesa – okruženje, alati, rokovi, nesavršenstvo
- Rešavanje procesa – efektivno građenje barijere između zdravog preduzeća i bolesti.

Učenje iz industrije

U mnogim situacijama iz realnog života principi obrazaca i antipaterna se kombinuju. Primeri

Želimo da otkrijemo rupe u procesu i proceduru koja može da sputava naše kupce ili čak i nas u budućnosti.

Želimo da znamo kada se veliki sistemski problemi dešavaju.

Detektivski posao

- Problemi u bazama sadrže nagrade bogatstva. – praćenje metrike kvaliteta u formi grešaka na bazi.
- Rana provera performansi može da ukaže na mane u dizajnu – ističe greške u dizajnu
- Česte provere koda i pomoć mentora – uparivanje početnika i iskusnog programera
- Krajnji korisnici neuobičajeno dobro zapažaju – mišljenje krajnjeg korisnika može biti veoma korisno
- Strancima će koristiti razgovor– lakše otvaranje kad vide da hoće neko da pomogne a ne da okrivljuje
- Uspostavljanje uzorka – treba imati procese i procedure za otklanjanje grešaka, Nikad ne možemo otkloniti sve greške, ako proces nije korektan napraviće se više grežaka nego što će se srediti

Sređivanje antipaterna

Proces stalnog poboljšavanja koda kroz rekonstrukciju za jasnoću ili fleksibilnost i uklanjanje redundantnog i nekorišćenog koda zove se sređivanje.

Ako nije pokvareno ne popravljaj ga.

Zaključak

Sadržaj knjige

Korišćeni alati

Za koga je knjiga namenjena