

Arkitektura Koadernoa ModelEditor

1. Helburua

Dokumentu honek deskribatu egiten ditu proiektuan zehar, eta zehazki ModelEditor azpisistemaren atalean, filosofia, erabakiak, baldintzak, justifikazioak eta garrantzizkoa izan daitekeen edozer.

2. Arkitektura helburuak eta filosofia

Sortutako editore grafikoa eta testu editorea erabiliz prozesuaren eredua sortzeko eta editatzeko aukera emango du. Sistema honen ardura prozesu ingeneriari rola izango du.

Azpisistemak hurrengo helburu nagusiak ditu:

- Metodologiaren informazioa editatu EPF Composer erabiliz.
- Metodologiaren informazioa editatu editore grafikoa erabiliz.
- Metodologiaren informazioa editatu testu editorea erabiliz.
- Metodologiaren informazioa datu-basean gordetzeko SQL kodea sortu.

3. Hipotesiak eta menpekotasunak

Metaereduen erabilerak malgutasuna ematen du proiektuaren hurbilpena aldatzeko edo etorkizunean beste hurbilpen batzuk probatzeko.

Definitutako garapen prozesuak software proiektuen elaborazioa gidatu eta kontrolatuko du. Prozesua aldatzeak sistemaren portaera eta datu-basea automatikoki aldatzea ekarriko du.

IO-System azpisistemaren arkitektura ModelEditor azpisistemarekin erlazionatuta dago. Izan ere, ModelEditor-ekin definitutako ereduaren datuak erabili beharko ditu. Hala ere, dependentzia ez da ProWF proiektuan bezain zuzena, ModelEditor sistemaren arkitektura abstraktuagoa delako.

Sistemarako tresna moduan Eclipse erabiltzen denez honekiko dependentzia izango du. Gainera, Java bertsioekiko dependentzia ere egongo da. Adibidez, EPF Composerekin derrigorrez Java 8 32 bit erabili behar da.

4. Betekizun arkitekturalki esanguratsuak

Sistemaren betekizunak Ikuspegia eta Betebeharren Espezifikazioa dokumentuetan definitu dira.

5. Erabakiak, mugak eta justifikazioak

ProMeta proiektuan sistemaren abstrakzio-maila igotzea erabaki da. Metodologiaren informazioa workflow lengoaiaren edo BPMNn eskuz definitu ordez, metaeredu eta ereduen bidez gordetzen da. Horrela, ereduaren informazioa berrerrabil daiteke eta beste ereduak eraldaketak egin. Ereduen informazioa zuzenean kodea sortzeko ere erabil daiteke, beste metaereduen beharrik gabe.

Hurbilpen honek ziurtatzen du etorkizunean aurreko aukeretako edozein edo beste batzuk aukeratu ahal izatea. Izan ere, BPMNren eta workflow-lengoaiaren metaereduak izanik, eraldaketa egitea ez litzateke horrenbeste lan izango. Gainera, ereditutako datu-baserako SQL kodea edo webgunerako HTML kodea sortzeko aukera ere egongo litzateke.

6. Mekanismo arkitekturak

- **Metodologiak editatu:** Metodologiak editatu EPF Composer erabiliz edo editore grafiko eta testuala erabiliz.
- **Metodologiaren webgunea sortu:** Metodologiaren webgunea sortu EPF Composer erabiliz.

- **Metodologiaren datu-basea sortu:** Metodologiaren informazioa datu-basean gorde SQL aginduekin. Ondoren, informazio hori IO-System azpisistemak erabiliko du.

7. Geruzak edo arkitektura frameworka

- **EPF Composer:** Metodologiak modu grafikoan editatzeko aukera ematen du. Metodologiaren webgunea sortu daiteke bertatik.
- **Formatu Aldatzailea:** Metodologiaren formatua aldatzen du XMItik UMARA editore grafikoak erabili ahal izan dezan.
- **Editore Grafikoa:** Metodologiak modu grafikoan editatzeko aukera ematen du, metodologiaren jatorrizko ikonoak erabiliz.
- **Testu Editorea:** Metodologiak testu bidez editatzeko aukera ematen du, metaeredutik sortutako gramatika erabiliz.
- **Editore Sinkronizatzailea:** Metodologiaren informazioa editore batekin aldatutakoan bestea ere aldatzeaz arduratzen da.
- **Kode Sortzailea:** Metodologiaren informazioa datu-basean gorde ahal izateko SQL kodea sortzen du.

