

10 Aukeren Azterketa eta Egingarritasuna

Kapitulu honetan proiektuko soluzioaren alternatibak, balorazio-irizpideak eta hautatutako aukeren justifikazioak agertzen dira.

10.1 Arkitektura

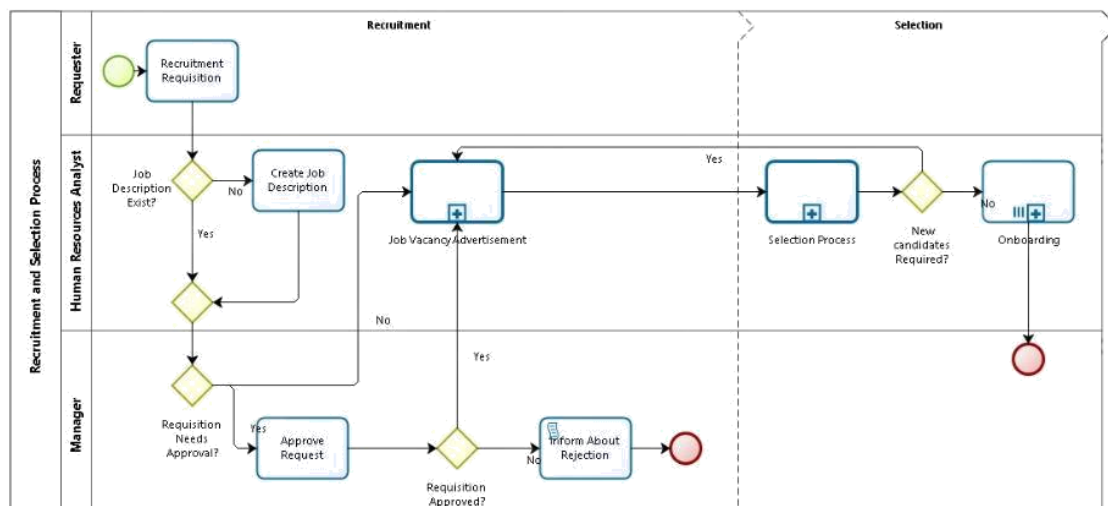
Soluzioaren arkitekturari dagokionez, ProWF proiektuan bi aukera aztertu ziren: *Bizagi* bezalako softwarearen bitartez prozesuetan (BPM) oinarritutako web-aplikazioa eraikitzea edo arkitektura propioa sortzea. Proiektu honetan aukera berri bat gehitu da.

10.1.1 Bizagi

Bizagi Softwarearen Kalitatea irakasgaia erabilitako softwarea da. Bertan, *Bizagi Modeler* editorearen bitartez prozesuak sortu eta ondoren, *Bizagi Studio* tresnan negozio-erregelak adieraziz, datu-basea konfiguratuz, formularioak definituz, web-zerbitzuak integratuz eta beste hainbat aspektu ukituz, sortutako prozesuan oinarritutako web-aplikazio bat sortu zen. Sortutako web-aplikazioak itxura oso profesionala zuen eta bere sorkuntza ez zen izan batere zaila.

Proiektu honetarako bideragarria izango litzateke software hau erabiltzea, hurrengo bi arrazoiengatik:

- Lan-fluxua definitzeko *Bizagi Modeler* editorea erabili daiteke, bizi-zikloa definitzen duten metodologiek faseak eta rolak erabiltzen dituztelako, 10.1. Irudian ikus daitekeen bezala. Beraz, *Bizagi Modeler* erabiliz, bere lengoaiaren (BPMN) arauak errespetatuz, *workflow* azpi-lengoaia bat sortu izango litzateke.
- Soluzioaren sarrera/irteerak kudeatzeko *Bizagi Studioren* bitartez sortutako web-aplikazioa erabili daiteke, bere datu-baseen kudeaketa, negozio-erregelak, formularioak eta web-zerbitzuak baliatuz.



10.1. Irudia. Bizagi Modeler erabiliz sortutako prozesua.

Arkitekturaren soluzio honek, ostera, bi desabantaila nagusi ditu:

- Arkitektura guztia kanpoko tresna baten bitartez eraikitzeak izugarrizko menpekotasun teknologikoa sortuko du. Garapen-prozesuaren konplexutasuna gorakorra izango denez, sistema osoa, hasieratik, *Bizagiren* menpe jartzea arriskutsua izan daiteke eta ez da batere komenigarria produktuaren aldaketa eta hobekuntzarako.

- *Bizagi Modeler freemium* bat da, hau da, zerbitzu basikoak dohain eskaintzen ditu eta zerbitzu aurreratuago edo bereziengatik zerbait ordaindu behar da. *Bizagi Studio*, ordea, *suite* ofimatikoa da, enpresen erabilpenerako paketeak eskaintzen ditu bere produktua erosiz. Hori dela eta, *Bizagirekin* arkitekturaren kostu ekonomikoa handia izango litzateke.

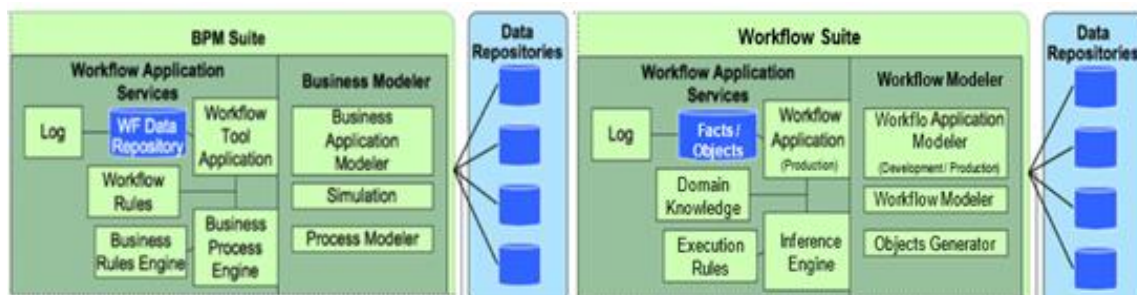
10.1.2 ProWF

Esan bezala, ProWF proiektuan erabakia **arkitektura zerotik eraikitzea** izan zen. Hurrengo kapituluan sartuko gara soluzioaren deskribapenean.

Workflowetan oinarritutako web-aplikazio bat zerotik eraikitzea apustu handia zen, denbora gehiago eman beharko litzateke arkitekturaren garapenean, baina, ordea, pisuzko arazoak zituen:

- *Bizagiren* desabantailak oso kaltegarriak ziran. Honela, menpekotasun teknologikoa saihestu eta kostu ekonomikoa asko gutxitzen da.
- *Workflow*-lengoaia guztiz propioa sortzea komenigarria zen, lengoaia hori exekutatzeko motor propioa ere, bere erregela eta berezitasunekin.
- Gaur egun, sarrera/irteeren kudeaketarako, *CMS* baten erabilerarekin web-aplikazioen sormena ez da hain zaila eta soluzio profesionalak lortu daitezke.

Hala ere, workflowetan oinarritutako arkitektura hau zerotik eraikitzea ez da lan erraza eta sistema konplexua bat sortu behar da, gainera, *Bizagik* eta *BPMN*¹⁸ estandarrak eskaintzen dituzten funtzionalitate asko galduko ditugu. 10.2. Irudian BPMn eta workflowetan oinarritutako arkitekturak ikus daitezke.



10.2. Irudia. BPMn eta workflowetan oinarritutako arkitekturak.

10.1.3 ProMeta

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

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

Proiektu honetan saiakera bat egin da BPM eta workflow-lengoiarik erabili gabe. Izan ere, eredutik sortutako datu-basea eta Drupal-eko moduluak bakarrik erabili dira. Drupal CMSaren funtzionalitateak aprobetxatzeko ahal den guztia bere interfazea erabiliz eraiki da. Adibidez, edukia kanpoko datu-basean gorde ordez, Drupal-eko eduki motetan gordetzea erabaki da.

10.2 CMS

ProWF proiektun, soluzioaren datu zein informazioaren sarrera/irteerak kudeatzeko web-aplikazio bat sortzea erabaki zenez, CMS bat erabiltzea adostu zen. CMS baten bitartez web-aplikazioaren administrazioa eta kudeaketa ahalbidetzen da eta itxura profesionala duen emaitza lortu daiteke.

Hasieratik *Drupal* erabiltzea gomendatu zuen proiektuaren tutoreak, Juan Manuel Pikatzak, baina *Drupal* erabiltzen hasi baino lehen merkatuan zeuden beste CMSak aztertu behar ziren ere. Hiru CMS aztertu ziren nagusiki: *Wordpress*, *Joomla* eta *Drupal*.

Taulan ProWF proiektuan egindako konparaketa bat ikus daiteke erabakia hartzeko gehien nabarmentzen diren puntuekin.

Ezaugarria	Wordpress	Joomla	Drupal	Oharrak
Kode irekia	BAI	BAI	BAI	-
Dokumentazio sinple eta ondo egituratuta	BAI	BAI	BAI	-
Komunitate aktiboa eta foroak	BAI	EZ	BAI	Hemen Wordpress da nagusia.
Estentsio gehigarri eta moduluen hedapena	BAI	EZ	BAI	Joomlak estentsio gehigarriak ditu ere, baina ez askorik.
Erabiltzaile berrientzako erabilerraza	BAI	EZ	EZ	Joomla eta Drupalekin zaila izan daiteke hasieran bere konfigurazioa edo gunearen itxura aldatzen jakitea edo
Erabiltzaileen kudeaketa erraza	EZ	EZ	BAI	Drupalen bitartez rolak sortu/esleitu eta baimen espezifikoak eman daitezke
Programazio-lengoaia	PHP	PHP	PHP	-

10.1.Taula. CMS ezberdinen ezaugarrien konparaketa.

Azkenik, azterketa sakon bat egin eta aukera bakoitza ebaluatu ostean, *Drupal* CMSa erabiltzea izan zen erabakia, hurrengo arrazoiengatik:

- Drupalen erraza da edukia gehitzea/sortzea. Eduki pertsonalizatu motak malguak dira eta aukera asko eskaintzen dituzte.
- Guneari gehitzeko hainbat modulu eskuragarri daude bere webgunean eta proiektu honetarako oso erabilgarriak diren moduluak aurkitu dira.
- Erabiltzaileak administratzea erraza da, rol berriak sortu eta baimenak zehaztu ditzakeen sistema integratu batekin. Funtzionalitate hori oso komenigarria zen proiektu honentzat.
- Mundu mailan garrantzitsuenak diren teknologia saltzaileen sailkapenak argitaratzen dituzten Gartner eta Forrester erakundearen txostenetan, CMS atalean, liderra den Acquia enpresak *Drupal* erabiltzen du oinarri bezala.

ProMeta proiektuak CMSari dagokionez antzeko helburuak dituenaz, aurreko arrazoi guztiak mantentzen dira. Gainera, *Drupal* erabiltzeak orain beste abantaila bat du, aurreko proiektuaren zati batzuk berrerabiltzeko aukera.

10.2.1 Wordpress

2003ko maiatzaren 27an jarri zen abian, edozein motatako web orrialdeak sortzera bideratuta. Jatorrian blogen sorkuntzan arrakasta handia lortu zuen, baina geroago web orrialde komertzialak sortzeko tresna nagusietako bat bilakatu zen.

WordPress PHP hizkuntzan garatzen da MySQL eta Apache exekutatzeko inguruneetarako, GPL lizentziapean eta software librea da.

Helburu orokorreko CMS ezagunena da. 2019ko martxoan Interneteko gune guztien % 33,4k eta eduki kudeatzaileetan oinarritutako gune guztien %60,3k erabiltzen zuten.

Arrakastaren arrazoietakoa bat garatzaile eta diseinatzaileen komunitate izugarria da, bere muinean programatzeaz edo komunitatearentzako pluginak eta txantiloak sortzeaz arduratzen dena.

10.2.2 Drupal

Doakoa, modularra, erabilera anitzekoa eta oso konfiguragarria da. Artikuluak, irudiak, fitxategiak argitaratzea ahalbidetzen du eta beste zerbitzu gehigarri batzuk ere eskaintzen ditu, hala nola foroak, inkestak, bozketak, blogak, erabiltzaileen administrazioa eta baimenak.

Drupal sistema dinamikoa da: bere edukia zerbitzariaren fitxategi estatikoetan gorde beharrean, orrien testu edukia eta bestelako ezarpenak datu base batean gordetzen dira eta web ingurunea erabiliz editatzen dira.

Doako programa da, GNU/GPL lizentziarekin, PHP-n idatzia eta MySQL-rekin bateragarria. Erabiltzaileen komunitate aktibo batek garatu eta mantentzen du. Aipagarria da kodearen eta sortutako orrien kalitatea, web estandarrak errespetatzea eta sistema osoaren erabilgarritasuna eta koherentzia.

Drupal-en diseinua bereziki egokia da Interneteko komunitateak eraikitzeke eta kudeatzeko. Malgutasun eta moldagarritasunagatik nabarmentzen da, baita eskuragarri dauden modulu osagarrien kopuru handiagatik ere, webgune mota ugari egiteko egokia da.

10.2.3 Joomla

Webgune dinamikoa eta interaktiboak garatzeko aukera ematen du. Webgune bateko edukia modu errazean sortu, aldatu edo ezabatzeko aukera ematen du administrazio panel baten bidez. Kode irekiko softwarea da, PHP-n programatua edo garatua eta GNU General Public License (GPL) lizentziapean argitaratua.

Joomla-k bere funtzionamendurako datu-base kudeatzailearekin sortutako datu-basea behar du (MySQL da ohikoena), baita Apache HTTP zerbitzaria ere.

Ondo eratutako HTML kodea sortzea, blogen kudeaketa, artikuluak inprimatzeko ikuspegiak, albisteen flash-a, foroak, inkestak (inkestak), egutegiak, gune bilaketak integratuak eta hizkuntza anitzeko laguntza dira Joomla-rekin sor daitezkeen tresnetako batzuk. Gaur egungo joerak direla eta apustu handia egiten ari da merkataritza elektronikoaren alde.

10.3 Dokumentazioa *Hostinga*

Proiektuaren dokumentazioaren webgunea interneten publikoki eskuragarri egon daitezen hosting zerbitzu bat erabili behar da. Doako aukerak bakarrik aztertu dira eta GitHub Pages erabiltzea erabaki da.

10.3.1 GitHub Pages

GitHubekin integratuta, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. GitHub Pages-en konfigurazioa oso erraza, errepositoriotik bertatik egiten da. Webgune estatikoak bakarrik onartzen ditu, beraz, drupal-erako ez du balio, PHP erabiltzen baitu. Webguneak HTML, CSS, eta JavaScript-en idatzita egon daitezke edo Jekyll webgune estatiko sortzailea erabiliz.

10.3.2 GitLab Pages

GitHub Pages-en antzekoa da, baina aukera gehiago eskaintzen ditu. Webgune estatiko sortzaile guztiak onartzen ditu, Gastby, Jekyll, Hugo, etab. Hori bai, GitLab Pages-en webgune guztiak eraikitzeke Continuous Integration fitxategi bat behar da. GitHub-en ez da beharrezkoa, aukeretan aktibatzea nahikoa

da. Gure webguneak ez duenez erabiltzen webgune estatiko sortzailerik, GitHub Pages erabiltzea hobe da sinpleagoa delako.

10.3.3 Netlify

GitHub-ekin integratu daiteke, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Aurrekoek bezala, webgune estatikoak bakarrik onartzen ditu baina aukera zabalagoak ditu. Netlify-k edozein webgune estatikorako hosting-a eskaintzen du, funtzionalitate zabalagoekin. Kasu honetan webgune sinple bat eraikitzea bakarrik interesatzen zaigunez, aurreko aukerekin nahikoa daukagu.

10.4 Drupal *Hostinga*

Drupal webgunea interneten publikoki eskuragarri egon daitezen hosting zerbitzu bat erabili behar da. Doako aukerak bakarrik azertu dira eta Pantheon erabiltzea erabaki da.

10.4.1 000webhost

Ez dauka GitHubekin integraziorik eta beraz kodea eskuz igo beharko litzateke aldi bakoitzean. PHP kodea onartzen du, eta beraz Drupalerako balio du. 000webhost-ek MySQL datu-basea eskaintzen du, xampp-ekin bateragarria. ProWF proiektuan aukera hau erabili da proiektuaren webgunerako. Drupal webgunea ez da eskuragarri jarri.

10.4.2 Heroku

GitHubekin integratu daiteke, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan. Herokuren konfigurazioa GitHub Pages-ena baino zailagoa da. PHP kodea onartzen du eta beraz, Drupalerako balio du. Defektuz PostgreSQL datu-basea eskaintzen du. Xampp-ek, berriz, MySQL eskaintzen du. Drupal modu sinplean instalatzeko ez dago prestatuta.

10.4.3 Pantheon

GitHub-ekin ondo integra daiteke, nahiz eta ez den horren sinplea. Drupal modu sinplean instalatzeko prestatuta dago. Drupal-erako honek ematen du aukera onena.

10.5 Datu-base Kudeaketa Sistema

Metodologiaren informazioa gordetzen duen datu-baserako DBKS egokia aukeratzea garrantzitsua da. Erabiliko ditugun tresnen eta *hosting*-aren menpekoea da, ez baitaude guztiak eskuragarri. MySQL erabiltzea erabaki da.

10.5.1 MySQL

Drupal-ekin bateragarria da, ematen dituen aukeretako bat da. Xampp-ek eskaintzen duen datu-basea da, eta beraz garapen lokalerako egokia. Xampp-eko phpMyAdmin bidez kontrola daiteke.

10.5.2 PostgreSQL

Drupal-ekin bateragarria da, ematen dituen aukeretako bat da. Xampp-ek defektuz ez du eskaintzen, baina gehigarri moduan aktibatzeak aukera ematen du. Xampp-eko phpPgAdmin gehigarriaren bidez kontrola daiteke.

10.6 Datu-basearen Sorrera

Metodologiaren informazioa datu-basean gordetzeko prozesua automatizatzea ezinbestekoa da. Hainbat aukera azertu ondoren egokiena Xtext dela erabaki da.

10.6.1 Inferentzia Motorra

ProWF proiektuan inferentzia motorra erabiltzen da datu-basearen sorrerarako. Inferentzia motorra erabiltzen ez badugu bide honek ez dauka zentzurik.

10.6.2 Teneo

Teneo-k EMF eredu eta datu-base erlazionalen arteko mapaketa eskaintzen du <https://wiki.eclipse.org/Teneo>. Hibernate-rekin bateragarria da, objektu eta datu-base erlazional arteko mapaketarako tresna <https://hibernate.org/>. Aukera honek datu-basea automatikoki sortuko luke. Baina, datu-basea oso handia izango litzateke eta beharrezkoa baina elementu gehiago izango litzateke. Gainera, Teneo zaharkitua dago eta Eclipseren eta EMF-ren bertsio berriei ez da bateragarria.

10.6.3 Xtext

Aukera onena Xtext eta Xtend erabiliz ereduko datuekin SQL INSERT-ak sortzea da. Honek lan dezente eskatuko luke. Baina, flexibilitate asko emango luke, datu-basea nahi dugun bezala diseina dezakegu. Horrela, behar dugun informazioa bakarrik izango dugu datu-basean, eta errazagoa izango da honekin lan egitea.

10.7 Lanerako Ingurunea

Lanerako ingurune egokia aukeratzea garrantzitsua da. Baita ere ingurunearekin arazoren bat badago alternatibak izatea.

10.7.1 Makina Birtuala

Juanmak lanerako makina birtuala eskaini dit. Ordenagailu horretan aurreko proiektuak daude eta erabilgaitako softwarea instalatuta. Honetara VPN bidez konektatu beharko nintzateke. Honen arazoa da lokalean lan egitean baino makalago joango dela.

10.7.2 Ordenagailu Pertsonala

Ordenagailu pertsonala nire gustura konfiguratuta daukat eta erabiltzen dutan softwarea instalatuta. Makina birtualean softwarea instalatu daiteke baina errazagoa da nire konputagailuan falta den softwarea instalatzea. Izan ere, Drupal eta EHSIS bakarrik falta dira. Hori bai, bukaeran sistema zerbitzariko makina birtualera pasatuko da, besteek eskuragarri eduki dezaten.

10.8 Bertsio Kontrola

Bertsio kontrola oso garrantzitsua da, batik bat horrelako proiektuetan. Bertsio ezberdinak gordetzeko modu egokiena da, eta arazoren bat badago kodea berreskuratzeko aukera ematen du. GitHub erabiltzea erabaki da.

10.8.1 GitHub

Bertsio kontrolerako Git eta GitHub erabiltzea da aukera onena. Informazio guztia GitHub-eko errepositorio batean egongo da eta webgunea publikatuta egongo da. Horrela, beharrezkoa denean Juanmari erakutsi ahal izango diot egiten ari naizena. Gainera, bertsio kontrolak segurtasuna eta trazabilitatea ematen du.

10.8.2 GitLab

GitLab erabiltzea ere aukera ona izan daiteke, antzeko aukerak eskaintzen ditu. GitHub-ekin praktika gehiago daukat eta beraz ez dauka zentzurik GitLab-era aldatzeak. GitLab-ek bakarrik eskaintzen duen zerbait beharko bagenu orduan bai.

10.9 Metaeredua

Garrantzitsua da prozesuak definitzeko metaeredu egokia aukeratzea, horrek mugatuko baititu defini ditzakegun prozesuak.

10.9.1 SPEM

Software Process Engineering Metamodel CMOF formatuan dagoen metaeredu estandarra da, OMG erakundeak sortua.

10.9.2 UMA

Unified Method Architecture metaereduaren helburua edozein metodologia modelatu ahal izatea da. SPEM metaereduan oinarrituta dago, baina ecore formatuan dago. Eclipse tresnarekin lan egiteko formatu hau behar da, EMF formatu horretarako prestatuta dagoelako

10.10 Metodologia

Prozesua definitzeko erabiliko ditugun metodologiak ongi aukeratzea garrantzitsua da. OpenUP eta ABRD metodologiak erabiltzea erabaki da.

10.10.1 OpenUP

OpenUP softwarea garatzeko metodo eta prozesu bat da, teknologien sektoreko enpresa multzo batek proposatutakoa, zeintzuk 2007an Eclipse Fundazioari dohaintzan eman zioten. Fundazioak lizentzia libre bezala argitaratu du eta eredu gisa mantentzen du Eclipse Process Framework (EPF) proiektuaren barruan.

Metodologia honek garrantzi handia izan du proiektu osoan zehar. Batetik, proiektuaren helburuetako bat metodologia baten definizioa eta ezarpena izan da eta OpenUP izan da aukeratutako metodologia. Bestetik, proiektuaren elaborazio prozesurako OpenUP metodologia jarraitu da, dokumentazioa bilduz eta proiektuaren kontrola eramanez.

10.10.2 RUP

Kontuan hartuta proiektua eta proiektuko taldea txikiak direla, OpenUP metodologia erabiltzea nahikoa da. Gainera, metodologiako lehenengo bi edo hiru fase egitearekin nahikoa izango da. Talde handiagoa edo proiektu konplexuagoa izango balitz agian RUP bezalako metodologia konplexuagoa bat beharko genuke.

Metodologiaren ezarpenari dagokionez, sinpleagoa da OpenUP metodologia ezartzea txikiagoa delako. Gainera, OpenUP metodologiarekin lan egiteko Eclipse-ren tresnak erabil daitezke. Tresna horiek libreak dira, eta egileak esperientzia du horiek erabiltzen.

10.10.3 ABRD

Agile Business Rules Development metodologiaren eredua ere erabili da. Horrela, bi metodologia erabiliz ziurtatzen da sistemaren egitura egokia dela. ABRD negozio erregela aplikazioa garatzeko metodologia da, erregela motorra eta erregelak kudeatzeko sistema erabiliz.

10.11 Metodologia Editorea

Behin metodologiak aukeratuta, metodologia horiek nola editatuko ditugun erabaki behar da. Kasu honetan 3 aukera ematea erabaki da: EPF Composer, editoe grafikoa eta testu editorea.

10.11.1 EPF Composer

Metodologia definitzeko aukera argiena da, horretarako egindako tresna baita. Jadanik hainbat metodologia definituta daude tresna honekin: OpenUp, ABRD, Scrum... Metodologia horietako bat erabili

nahi badugu ez daukagu ezer diseinatu beharrik. Horiek oinarri bezala hartuta ere errazagoa da beste metodologia bat definitzea. Metodologiaren webgunea sortzeko aukera ematen du, dokumentazio moduan erabili ahal izateko. Nahiko zaharkitua dago, Java 8 32 biteko bertsioa eskatzen du.

10.11.2 Rational Method Composer

EPF Composer-en antzeko tresna, baina IBMrena da eta ordainpekoa. RUP metodologia du oinarrian eta horrekin lana egin nahi badugu aukera interesgarria izan daiteke.

10.11.3 Editore Grafikoa

EPF Composer baino sinpleagoa den editorea edukitzea ondo etorriko litzateke. Ikono berdinak partekatuko lituzke, zuhaitz egitura sinpleagoa edukiko luke.

10.11.4 Testu Editorea

Editore grafikoa baino sinpleagoa den testu editore bat edukitzea ere komenigarria da. Batzuetan, editore grafikoa baino erosoagoa da testu editorea erabiltzea. Onena eredia bi editoreekin aldatzeko aukera izatea izango litzateke, eta edozein momentutan bien artean aldatzeko aukera izatea.

10.12 Prozesua Bistaratu

Prozesua bistartzeko modua izatea ezinbesteko da erabiltzaileek prozesua kontsultatu ahal izateko eta prozesuan hobekuntzak egiteko. EPF Composer-ekin sortutako webgunea nahiko dela erabaki da.

10.12.1 DOT

ProWF proiektuan prozesua definitzeko eta bistartzeko DOT lengoaia erabili da. Beraz, aukera ona izan daiteke prozesua bistartzeko. Hori bai, honek lan dezente gehituko luke. Izan ere, eredia DOT lengoiara pasatzea eskatuko luke metaereduak erabiliz.

10.12.2 XPD

XPD ere aukera egokia izan daiteke estandarra delako. Baina, oraingoz DOT aukera hobe izango litzateke, sinpleagoa delako.

10.12.3 Webgunea

Prozesua bistartzeko EPF Composer-ekin publikatzen den webgunea erabiltzea da aukera onena. Izan ere, webguneak metodologiaren informazio guztia dauka, eta honen sorrera automatikoa da. Beraz, prozesua EPF Composer-ekin definitzea komenigarria da.

10.13 Denboraren Kontrola

Denboraren kontrola modu zehatz eta antolatuan egiteko tresna egokia aukeratzeak asko laguntzen du. Hainbat aukera aztertu dira eta Toggl Track aukeratu da.

10.13.1 Toggl Track

Tenporizadore baten bidez ataza bakoitzean pasatako denbora neurtu daiteke. Eskuz ere aldatu daiteke denbora ahaztu egin bazaigu. Abisatu egiten du inaktibo bagaude eta denbora neurtzen ari bagara. Aspaldian denborarik ez badugu neurtu ere abisatzen du, ez ahazteko. Denbora proiektutan, atazatan eta kategoriatan sailkatu daiteke. Edozein momentuko denboraren estatistikak ikusi eta deskargatu daitezke. Gailu mota guztietarako aplikazioak daude, mugikorrerako ordenagailurako, nabigatzaileerako, etab.

10.13.2 Clockify

Toggl Track-en antzeko funtzionamendua dauka, denbora neurtzeko tresna batek eduki beharreko oinarritzko funtzionalitatea dauka. Doako planean aukera batzuk gutxiago eskaintzen ditu.

10.13.3 WakaTime

Plugin moduan instalatu daiteke editore askotan. Gure kasuan VSCode eta Eclipse editoreetan instalatu da, horiek izan baitira erabili diren editoreak. Denbora automatikoki neurtzen du eta metrikak sortzen ditu. Adibidez, programazio-lengoaia eta proiektu bakoitzean pasatako denbora erakusten du. Ez du balio editoreetatik kanpo pasatako denbora neurtzeko. Hala ere, Toggl Track-en osagarri ona da, estatistika zehatzagoak eskaintzen dituelako. Doako planean 2 asteko historia bakarrik erakusten du.

10.14 Diagramak

Diagramak beharrezkoak dira hainbat kontzeptu modu grafikoan adierazteko. Hainbat aukera aztertu ondore, diagrama guztiak egiteko PlantUML erabiltzea erabaki da.

10.14.1 PlantUML

Hainbat motatako diagramak sortzeko aukera ematen duen softwarea da. Eskuz sortzen dira diagramak eta bi aukera daude aplikazioa erabiltzeko: online edo offline. Proiektuan zehar aplikazio hau erabili izan da dokumentazioan agertzen diren diagrama gehienak egiteko, bezeroak dokumentazio ulergarria jaso dezan.

10.14.2 Draw.io

Hainbat motatako diagramak sortzeko aukera ematen duen softwarea da. Eskuz sortzen dira diagramak eta bi aukera daude aplikazioa erabiltzeko: online edo offline. Azkenean diagrama guztiak egiteko PlantUML tresna erabiltzea erabaki da.

10.14.3 UMLDoclet

Javadoc metadatuak erabiltzen ditu PlantUML diagramak automatikoki sortzeko eta HTML dokumentazioan txertatzeko. Diagramak irudi klikagarri gisa txertatzen dira eta pakete eta klaseko dokumentaziora estekatzen dira eskuragarri dagoenean. Pakete dependentzia, pakete eta klase diagramak sortzen ditu elementu guztietarako. ModelEditor azpisistemaren dokumentazioan diagramak gehitzeko aukera egokia da.