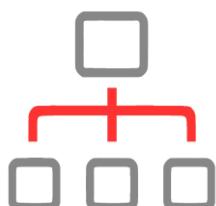


Informatika Fakultatea

Informatika Ingeniaritzako Gradua

▪ Gradu Amaierako Lana ▪

Software Ingeniaritzza



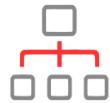
ProMeta

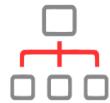
Softwarearen garapenerako prozesuen definizio eta
ezarpenerako sistema metaereduetan oinarrituta

Egilea: Julen Etxaniz Aragoneses

Tutorea: Juan Manuel Pikatza Atxa

2021eko iraila





Laburpena

Proiektuaren helburua metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema eraikitzea da. Izan ere, softwarea garatzeko egokiak diren hainbat metodologia existitzen dira. Garrantzitsua da metodologia horien informazioa ereduen bidez definitzea, etorkizunean malgutasunez kudeatu eta hobekuntzak egin ahal izateko. Gainera, beharrezkoa da ereduen informazioa erabiliz metodologia ezartzen duen sistema eraikitzea, garapen taldeak proiektuetan erabiltzeko. Proiektuaren garapenerako, OpenUP metodologia erabili da eta proiektuaren dokumentazioa eta memoria idazteko CCII-N2016-02 estandarra.

Hitz gakoak: metaeredua, garapen metodología, aurkezpen araua, softwarearean bizi-zikloa, MDE, CMS, sistematizazioa, automatizazioa.

Resumen

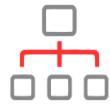
El objetivo del proyecto es construir un sistema para la definición e implementación de procesos de desarrollo de software basado en metamodelos. De hecho, existen varias metodologías que son adecuadas para el desarrollo de software. Es importante definir la información de estas metodologías a través de modelos para que se puedan gestionar de forma flexible en el futuro y se puedan realizar mejoras. Además, es necesario construir un sistema que establezca una metodología utilizando información del modelo para uso de los equipos de desarrollo en los proyectos. Para el desarrollo del proyecto se utilizó la metodología OpenUP y el estándar CCII-N2016-02 para la redacción de la documentación y memoria del proyecto.

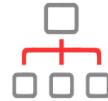
Palabras clave: metamodelo, metodología de desarrollo, norma de presentación, ciclo de vida del software, MDE, CMS, sistematización, automatización.

Abstract

The objective of the project is to build a system for the definition and implementation of software development processes based on metamodels. In fact, there are several methodologies that are suitable for software development. It is important to define the information of these methodologies through models so that they can be managed flexibly in the future and improvements can be made. In addition, it is necessary to build a system that establishes a methodology using information from the model for use by development teams in projects. The OpenUP methodology was used for the development of the project and the CCII-N2016-02 standard for the drafting of the project documentation and memory.

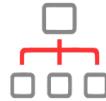
Keywords: metamodel, development methodology, presentation rule, software life cycle, MDE, CMS, systematization, automation.



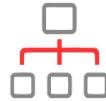


Aurkibide Orokorra

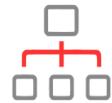
Laburpena.....	3
Resumen.....	3
Abstract	3
Aurkibide Orokorra.....	5
Irudien Aurkibidea	13
Taulen Aurkibidea.....	17
Laburdurak	19
1 Sarrera	21
2 Helburuak	23
2.1 Arazoaren Planteamendua	23
2.2 Produktuaren Planteamendua.....	24
2.3 Interesatuen Deskribapena.....	24
3 Aurrekariak	25
3.1 Software Kalitatea.....	25
3.2 Bezeroen Eskakizun Gogorrak.....	25
3.3 Garapen Metodologik.....	26
3.4 Proiekturen Aurkezpenerako Arauak	28
3.5 Prozesuetan Oinarritutako Garapena	28
3.6 Ezagutzaren Kudeaketa eta Inferentzia Motorrak	30
3.7 Ereduak Bideratutako Ingeniaritza	32
3.8 Edukiak Kudeatzeko Sistemak.....	33
3.9 ProWF Sistema	33
4 Egungo Egoera	37
4.1 Deskribapena	37
4.2 Abantailak eta Desabantailak	37
4.2.1 Abantailak.....	37
4.2.2 Desabantailak	38
4.3 Proposatutako Hobekuntzak	38
4.4 Prestakuntza	40
5 Arauak eta Erreferentziak.....	41
5.1 Aplikatutako Legedia eta Araudia	41
5.2 Bibliografia	42
5.3 Metodoak.....	43



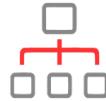
5.3.1	OpenUP.....	43
5.3.2	ABRD.....	43
5.3.3	MDE	43
5.4	Tresnak.....	43
5.4.1	Git	44
5.4.2	GitHub.....	44
5.4.3	GitHub Pages	44
5.4.4	Toggle Track.....	44
5.4.5	Java	45
5.4.6	Eclipse IDE.....	45
5.4.7	EPF Composer.....	46
5.4.8	Xtext.....	46
5.4.9	XSLT	46
5.4.10	Visual Studio Code.....	46
5.4.11	Microsoft Office	47
5.4.12	PlantUML.....	47
5.4.13	Draw.io	47
5.4.14	Drupal.....	48
5.4.15	XAMPP.....	48
5.4.16	Pantheon	49
5.4.17	MariaDB	49
5.4.18	phpMyAdmin.....	49
5.4.19	PHP	50
5.5	Ereduak.....	50
5.5.1	CCII-N2016-02.....	50
5.5.2	OpenUP.....	50
5.5.3	ABRD.....	50
5.5.4	UMA.....	50
5.5.5	Ecore	51
5.6	Metrikak.....	51
5.6.1	Irismena	51
5.6.2	Denbora	51
5.6.3	Kostua	51
5.6.4	Kalitatea	51
5.7	Prototipoak	52



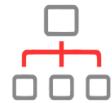
5.7.1	ProMeta ModelEditor	52
5.7.2	ProMeta IO-System	52
6	Definizioak eta Laburdurak.....	53
6.1	Laburdurak	53
6.2	Definizioak	55
6.2.1	Bizagi.....	55
6.2.2	BPM	55
6.2.3	CCII.....	55
6.2.4	CCII-N2016-01.....	55
6.2.5	CCII-N2016-02.....	55
6.2.6	CMMI	55
6.2.7	CMS.....	55
6.2.8	DOT	55
6.2.9	Drupal	55
6.2.10	IBM	55
6.2.11	ISO	55
6.2.12	OpenUP	56
6.2.13	PlantUML.....	56
6.2.14	ProMeta	56
6.2.15	ProWF.....	56
6.2.16	RUP	56
6.2.17	SPICE.....	56
6.2.18	UML.....	56
6.2.19	UNE.....	56
6.2.20	URPS	56
6.2.21	Workflow	56
6.2.22	WYSIWYG	56
7	Hasierako Betekezunak.....	57
7.1	Betekezun Funtzionalak	57
7.2	Betekezun Ez-Funtzionalak	57
7.3	Sistemaren Ezaugarríak.....	58
7.3.1	Erabilgarritasuna.....	58
7.3.2	Fidagarritasuna.....	58
7.3.3	Errendimendua	58
7.3.4	Mantengarritasuna	58



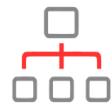
7.4	Sistemaren Interfazeak	58
7.4.1	Itxura eta Sentsazioa	58
7.4.2	Diseinu eta Nabigazio Betekizunak.....	58
7.4.3	Trinkotasuna.....	58
7.4.4	Erabiltzailearen Pertsonalizazio Betekizunak	58
8	Irismena	59
8.1	Bizi-zikloa	59
8.2	LDE Diagrama.....	59
8.3	Artefaktuak	60
9	Hipotesiak eta Murriztapenak	63
9.1	Hipotesiak	63
9.2	Murriztapenak	63
10	Aukeren Azterketa eta Egingarritasuna	65
10.1	Arkitektura	65
10.1.1	Bizagi	65
10.1.2	ProWF.....	66
10.1.3	ProMeta	66
10.2	CMS.....	67
10.2.1	Wordpress.....	67
10.2.2	Drupal.....	68
10.2.3	Joomla	68
10.3	Dokumentazioa <i>Hostinga</i>	68
10.3.1	GitHub Pages	68
10.3.2	GitLab Pages	68
10.3.3	Netlify	69
10.4	Drupal <i>Hostinga</i>	69
10.4.1	000webhost.....	69
10.4.2	Heroku.....	69
10.4.3	Acquia.....	69
10.4.4	Pantheon	69
10.4.5	Platform.sh	69
10.5	Datu-base Kudeaketa Sistema	69
10.5.1	MySQL	69
10.5.2	PostgreSQL	69
10.6	Datu-basearen Sorrrera.....	70



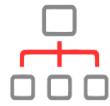
10.6.1	Inferentzia Motorra.....	70
10.6.2	Teneo.....	70
10.6.3	Xtext	70
10.7	Lanerako Ingurunea	70
10.7.1	Makina Birtuala	70
10.7.2	Ordenagailu Pertsonala.....	70
10.8	Bertsio Kontrola	70
10.8.1	GitHub	70
10.8.2	GitLab	70
10.9	Metaeredua	71
10.9.1	SPEM	71
10.9.2	UMA	71
10.10	Metodologia	71
10.10.1	OpenUp	71
10.10.2	RUP.....	71
10.10.3	ABRD.....	71
10.11	Metodologia Editorea	71
10.11.1	EPF Composer	71
10.11.2	Rational Method Composer	71
10.11.3	Editore Grafikoa	71
10.11.4	Testu Editorea	72
10.12	Prozesua Bistaratu	72
10.12.1	DOT.....	72
10.12.2	XPDL	72
10.12.3	Webgunea	72
10.13	Denboraren Kontrola.....	72
10.13.1	Toggl Track	72
10.13.2	Clockify	72
10.13.3	WakaTime	72
10.14	Diagramak.....	72
10.14.1	PlantUML.....	72
10.14.2	Draw.io	73
10.14.3	UMLDoclet.....	73
11	Proposatutako Sistemaren Deskribapena	75
11.1	Azpisistemak	75

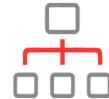


11.2	ModelEditor	75
11.2.1	Arkitektura	75
11.2.2	Erabilpen Kasuak	76
11.2.3	Diseinua.....	77
11.2.4	Dokumentazioa	78
11.3	IO-System	89
11.3.1	Arkitektura	89
11.3.2	Erabilpen Kasuak	90
11.3.3	Diseinua.....	92
11.3.4	Dokumentazioa	93
11.4	Proba.....	103
11.5	Hedapena.....	104
11.6	Etorkizunerako Hobekuntzak.....	105
12	Arriskuen Analisia	107
12.1	Arriskuak	107
12.2	Arriskuen Analisia	109
13	Proiektuaren Antolamendua eta Kudeaketa	112
13.1	Proiektuaren Antolamendua	112
13.1.1	Taldekideak eta rolak	112
13.1.2	Informazio-sistema.....	112
13.1.3	Komunikazio-kanalak	112
13.2	Proiektuaren Kudeaketa	112
13.2.1	Integrazioaren Kudeaketa	112
13.2.2	Irismenaren Kudeaketa	113
13.2.3	Epeen Kudeaketa.....	113
13.2.4	Kostuen Kudeaketa	113
13.2.5	Kalitate Kudeaketa	114
13.2.6	Giza Balia bideen Kudeaketa	114
13.2.7	Komunikazioen Kudeaketa.....	114
13.2.8	Arriskuen Kudeaketa	115
13.2.9	Erosketen Kudeaketa	115
13.2.10	Interesatuen Kudeaketa	115
14	Denbora Planifikazioa	116
14.1	Mugarriak.....	116
14.2	Lan-atazak.....	117



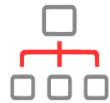
14.3	Gantt Diagrama.....	118
14.4	Iterazioak	118
14.5	Neurtutako denborak	120
14.5.1	Denbora hilabeteka.....	120
14.5.2	Hilabete bateko denbora	120
14.5.3	Aste bateko denbora.....	121
14.5.4	Denbora atazaka	121
14.5.5	WakaTime	122
14.6	Desbiderapenak	122
15	Aurrekontuaren Laburpena	124
16	Dokumentuen Lehentasun Ordena	126
	ERANSKINAK	128
I.	Memoriaren Eranskinak	128
A1	- Sarrerako Dokumentazioa.....	128
A2	- Analisia eta Diseinua	129
A3	- Tamaina eta Esfortzu Estimazioa.....	129
A4	- Kudeaketa Plana	130
A5	- Segurtasun Plana	130
A6	- Beste Eranskinak	130
II.	Sistemaren Espezifikazioa	131
III.	Barne Informazioa	131
1.	Barne Kudeaketa.....	131
2.	Artefaktuen Txantiloia	132
3.	Aurreko Projektuak	133
4.	Ingurunea	133
5.	Tresnak	134
6.	Trebatzeko Materialak	134
IV.	Egilearen Kudeaketa	135
1.	Arriskuen Analisia	135
2.	Denbora Planifikazioa	135
3.	Ondorioak	135



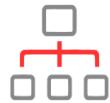


Irudien Aurkibidea

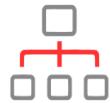
1.1. Irudia. Proiektuaren webgunean memoria bistaratu.....	21
1.2. Irudia. Webguneko eranskinen eta barne informazioaren menuak.....	22
3.1. Irudia. RUPen prozesu iteratiboa, bizi-zikloaren faseak, iterazioak eta jarduerak.....	27
3.2. Irudia. OpenUP-en geruzak: mikro-gehikuntzak, iterazio bizi-zikloa eta proiektu bizi-zikloa.....	27
3.3. Irudia. CCII-2016N-02 araua betetzen duen proiektuaren webguneararen egitura.....	28
3.4. Irudia. 2018ko BPM Suiten Koadrante Magikoa, Gartner.....	29
3.5. Irudia. 2019ko The Forrester Wave txostena, Forrester.....	30
3.6. Irudia. EHSISen garapen ingurunea.....	31
3.7. Irudia. Programazio tradizionalaren eskema.....	31
3.8. Irudia. Erregeletan oinarritutako sistemek eskema: klaseak, objektuak, erregelak eta egitateak....	31
3.9. Irudia. MDERen 3 abstrakzio maila: eredu, metaeredua eta meta-metaeredua.....	32
3.10. Irudia. CMS batek eskaizten dituen funtzionalitate nagusiak.....	33
3.11. Irudia. ProWF Workflow Editor azpisistemaren arkitektura.....	34
3.12. Irudia. ProWF IO-System azpisistemaren arkitektura.....	35
5.1. Irudia. ABRD metodologiaren prozesuko faseak.....	43
5.2. Irudia. ProMeta proiektuko webgunearen GitHub biltegia.....	44
5.3. Irudia. Toggle Track tresnan hilabete bateko egun bakoitzeko denbora.....	45
5.4. Irudia. ModelEditor azpisistemako proiektuak Eclipse IDE editorean.....	45
5.5. Irudia. OpenUP bizi-zikloa EPF Composer tresnan.....	46
5.6. Irudia. ProMeta dokumentazio webgunearen errepositorioa Visual Studio Code-n.....	47
5.7. Irudia. PlantUMLren lengoain idatzitako sekuentzia-diagrama.....	47
5.8. Irudia. Draw.io diagrama adibidea.....	48
5.9. Irudia. Proiektuan sortutako Drupal webgunea.....	48
5.10. Irudia. XAMPP kontrol panela Apache eta MySQL hasieratzeko.....	49
5.11. Irudia. ProMeta webguneko Pantheon kontrol panela.....	49
5.12. Irudia. phpMyAdmin tresnarekin datu-baseko taulak kudeatzen.....	50
5.13. Irudia. UMA metodologiako metodoaren eta prozesuaren kontzeptuak.....	51
8.1. Irudia. OpenUP metodologiako bizi-zikloaren faseak.....	59
8.2. Irudia. LDE diagrama.....	60
10.1. Irudia. Bizagi Modeler erabiliz sortutako prozesua.....	65
10.2. Irudia. BPMN eta workflowetan oinarritutako arkitekturak.....	66
11.1. Irudia. ModelEditor eta IO-System azpisistemen arkitektura.....	75
11.2. Irudia. ModelEditor azpisistemaren arkitektura.....	76
11.3. Irudia. ModelEditor azpisistemako erabilpen kasuen eredu.....	76
11.4. Irudia. UMA metaeredua graphviz formatuan.....	77
11.5. Irudia. UMA metaereduaren zati bat graphviz formatuan.....	78
11.6. Irudia. OpenUP bizi-zikloa EPF Composer tresnan.....	78
11.7. Irudia. OpenUP metodologiaren webgunea.....	79
11.8. Irudia. Eraldaketa egiteko XSLT fitxategia.....	80
11.9. Irudia. EMF proiektuaren sorrera Eclipsen.....	80
11.10. Irudia. Uma metaeredua importatzen Eclipsen.....	81
11.11. Irudia. Uma.ecore fitxategia Eclipsen.....	81
11.12. Irudia. Uma metaereduaren klase diagrama.....	82
11.13. Irudia. Editore grafikoaren kodearen sorrera.....	82
11.14. Irudia. Editore grafikoaren proiektuak Eclipsen.....	82

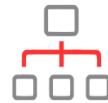


11.15. Irudia. Editore grafikoa martxan jartzeko konfigurazioa.....	83
11.16. Irudia. Editore grafikoarekin library.uma fitxategia bistaratzen.....	83
11.17. Irudia. Xtext proiekta sortzen Eclipsen.....	84
11.18. Irudia. Uma metaeredutik Xtext proiekta sortzen.....	84
11.19. Irudia. Xtext proiektuaren eta lengoaiaren izenak aukeratzen.....	85
11.20. Irudia. Testu editorearen proiektaak Eclipsen.....	85
11.21. Irudia. Xtext proiekta kokoatzen lengoaiaren gramatika.....	85
11.22. Irudia. Editore grafikoa ezkerrean eta testu editorea eskuinean.....	86
11.23. Irudia. Umal-etik uma-ra itzultzen duen serializatzailaren kodea.....	87
11.24. Irudia. Datu-basearen datuak dituen SQL fitxategia.....	88
11.25. Irudia. ModelEditor-en paketeak eta paketeen arteko dependentziak.....	88
11.26. Irudia. Pakete diagrama adibidea.....	89
11.27. Irudia. Klase diagrama adibidea.....	89
11.28. Irudia. IO-System azpisistemaren arkitektura.....	90
11.29. Irudia. IO-System azpisistemako erabilpen kasuen eredua.....	91
11.30. Irudia. OpenUP metodologiako rolak multzokatuta.....	91
11.31. Irudia. Proiekta kudeatzailak bete beharreko ataza eta artefaktuak.....	92
11.32. Irudia. Datu-base erlazionalaren diseinua.....	92
11.33. Irudia. Drupal datu-base baten diseinua.....	93
11.34. Irudia. ProMeta webguneko Pantheon kontrol panela.....	94
11.35. Irudia. Drupal webgunera datu-basetik edukia importatzeko moduluak.....	95
11.36. Irudia. ProWF sistemaren hasierako orria.....	95
11.37. Irudia. Drupal webgunearren hasierako orria.....	96
11.38. Irudia. Drupal webguneko edukia ikusteko orria.....	96
11.39. Irudia. Drupal webguneko metodologiak ikusteko orria.....	97
11.40. Irudia. Drupal webgunean OpenUP metodologia ikusteko orria.....	97
11.41. Irudia. Drupal webgunean maiatzean gehitutako edukia.....	98
11.42. Irudia. Drupal webguneko glosario orria.....	98
11.43. Irudia. Drupal webgunean edukia bilatzeko aukera.....	99
11.44. Irudia. Drupal webguneko saioa hasteko orria.....	99
11.45. Irudia. Drupal webguneko kontua sortzeko orria.....	100
11.46. Irudia. Drupal wenguneko pasahitza berrezartzeko orria.....	100
11.47. Irudia. Drupal webguneko erabiltzailearen kontuaren orria.....	101
11.48. Irudia. Drupal webguneko kontua editatzeko orria.....	101
11.49. Irudia. Drupal webguneko OpenUP metodologiaren orria.....	102
11.50. Irudia. Drupal webguneko metodologia sortzeko orria.....	102
11.51. Irudia. Drupal webgunean rolak kudeatzeko orria.....	103
11.52. Irudia. Drupal webgunean baimenak kudeatzeko orria.....	103
12.1. Irudia. Arriskuen inpaktu eta probabilitate analisia.....	110
14.1. Irudia. Mugarri garrantzitsuen diagrama.....	116
14.2. Irudia. Gantt diagrama.....	118
14.3. Irudia. Proiektaaren denbora hilabeteka.....	120
14.4. Irudia. Hilabete bateko denbora.....	121
14.5. Irudia. Aste bateko denbora.....	121
14.6. Irudia. Proiektaaren denbora atazaka.....	122
0.1. Irudia. Eranskinak webguneko menuan.....	128
0.2. Irudia. Memoriaren Eranskinak webguneko menuan.....	128



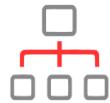
0.3. Irudia. Sarrerako Dokumentazioa webguneko menuan.....	129
0.4. Irudia. Analisia eta Diseinua webguneko menuan.	129
0.5. Irudia. Beste Eranskinak webguneko menuan.	130
0.6. Irudia. Sistemaren Espezifikazioa webguneko menuan.	131
0.7. Irudia. Barne Informazioa webguneararen menuan.....	131
0.8. Irudia. Barne Kudeaketa webguneko menuan.	132
0.9. Irudia. Artefaktuen Txantiloia webguneko menuan.	133
0.10. Irudia. Aurreko Proiektuak webguneko menuan.	133
0.11. Irudia. Injurunea webguneko menuan.	134
0.12. Irudia. Tresnak webguneko menuan.	134
0.13. Irudia. Trebatzeko Materialak webguneko menuan.	135





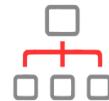
Taulen Aurkibidea

2.1. Taula. Arazoaren Planteamendua	23
2.2. Taula. Produktuaren Planteamendua.....	24
2.3. Taula. Interesatuen deskribapena	24
3.1. Taula. CMMI-ren 5 heldutasun-mailak.....	26
7.1. Taula. Hasierako betekizun funtzionalak.....	57
7.2. Taula. Hasierako betekizun ez-funtzionalak.....	57
8.1. Taula. OpenUP metodologiako bete diren artefaktuak domeinutan sailkatuta.	61
10.1.Taula. CMS ezberdinien ezaugarrien konparaketa.....	67
12.1. Taula. Identifikatutako arriskuen zehaztasunak.....	109
13.1. Taula. Kalitate kudeaketako rolak eta ardurak.....	114
14.1. Taula. Proiekktuko mugarrir garrantzitsuak.	116
14.2. Taula. Lan-atazen zehaztasunak.....	117
14.3. Taula. Proiekktuko iterazioen fasea, helburuak eta datak.	120
15.1. Taula. Aurrekontua giza baliabideen eta erreminten kostuekin.	124
15.2. Taula. Software taldeko kideen kostu baremoa.....	125

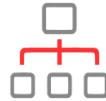


Laburdurak

ABRD	Agile Business Rules Development
BETRADOK	Betekizunen Trazabilitatea eta Dokumentazio sorkuntza automatikoa
Bizagi	BPM hornitzalea
BPM	Business Process Management
BPMN	Business Process Model and Notation
CCII	Consejo de Colegios de Ingeniería Informática
CCII-N2016-01	Norma de Visado de Proyectos y Actuaciones Profesionales en Ingeniería Informática
CCII-N2016-02	Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática
CLIPS	C Language Integrated Production System
CMMI	Capability Maturity Model Integration
CMS	Content Management System
CSS	Cascading Style Sheet
DOT	Grafoak definitzeko lengoiaia
Drupal	CMS hornitzalea
DSL	Domain Specific Language
Ecore	Meta-metaeredua
EHSIS	Erabakiak Hartzen Laguntzeko Sistemen Sortzailea
EHU	Euskal Herriko Unibertsitatea
EMF	Eclipse Modeling Framework
EPF	Eclipse Process Framework
Freemium	Dohako oinarrizko zerbitzuak gehi ordaindu beharrekoak
GrAL	Gradu Amaierako Lana
HTML	HyperText Markup Language
IBM	International Business Machines
IDE	Integrated Development Environment
ISO	International Organization for Standardization
LDE	Lan Deskonposaketa Egitura



MDE	Model-Driven Engineering
NASA	National Aeronautics and Space Administration
OpenUP	Open Unified Process
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PlantUML	Plant Unified Modeling Language
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
ProMeta	Professional Metamodel
ProWF	Professional Workflow
RUP	Rational Unified Process
SOA	Service-oriented Architecture
SPEM	Software Process Engineering Metamodel
SPICE	Software Process Improvement and Capability dEtermination
SQL	Structured Query Language
SWEBOK	Software Engineering Body of Knowledge
UMA	Unified Method Architecture
UML	Unified Modeling Language
UNE	Una Norma Española
URPS	Usability, Reliability, Performance and Supportability
Workflow	Lan-fluxua
WYSIWYG	What You See Is What You Get
XAMPP	Cross-platform Apache, MariaDB, PHP and Perl
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XPDL	XML Process Definition Language
XSLT	eXtensible Stylesheet Language Transformations



1 Sarrera

Dokumentu hau Julen Etxaniz Aragonesesen Informatika Ingeniaritzako Gradu Amaierako Lanaren memoria da. Egilearen inguruko informazio gehiago jakiteko <https://julenetxaniz.eus/> webgunea kontsulta daiteke. Lan hau Euskal Herriko Unibertsitateko (UPV-EHU) Donostiako Informatika Fakultatean landu da, Juan Manuel Pikatza izanik tutorea.

Dokumentu honetan ProMeta proiekturen inguruko aspektu guztiak azalduko dira. ProMeta izena prozesu edo profesional eta metaeredu hitzetatik dator. Portadan ikusten den proiektuaren logoak metaeredu bat irudikatzen du. Proiektuaren izenburua **Softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema metaereduetan oinarrituta** da. Beraz, proiektuak bi helburu nagusi ditu: garapeneko prozesuen definizioa eta prozesuaren ezarpenera.

Proiektua OpenUP metodologia erabilita gauzatu denez, bertako pausoak jarraitu dira. Gainera, dokumentazio formala eta profesionala lortzeko asmoz, ezaguna den CCII-N2016-02 arau estandarra eta Informatika Fakultateko GrAL eredua erabili dira memoriaren eta proiektuaren webgunearen atalak definitzeko.

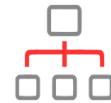
Proiektuaren webgunea honakoa da: <https://juletx.github.io/ProMeta/>. Webgune honen helburua ProMeta proiektuaren dokumentazio guzti biltzea eta proiektuko bezero zein interesdunek eskura izatea da. Memorian atal batzuetan webgunean dauden eranskinak aipatuko dira.

Webgunea bi zatitan dago banatuta, ezkerrean nabigazio menua dago eta eskuinean nabigazio menuan aukeratutakoa agertuko da. Ezkerreko menua erabiliz, proiektuko edozein dokumentu ikustea lortu daiteke: memoria, eranskinak, posterra, barne kudeaketarako dokumentuak etab. Gainera, webguneak CCII-N2016-02 estandarra betetzen duenez, bezero aldeko teknikari edo ebaluatzaileak oso azkar identifikatuko du behar duen dokumentua. Behin menuko aukera batean klikatzen denean, zati nagusian PDF formatu gisa irekiko da dokumentua.

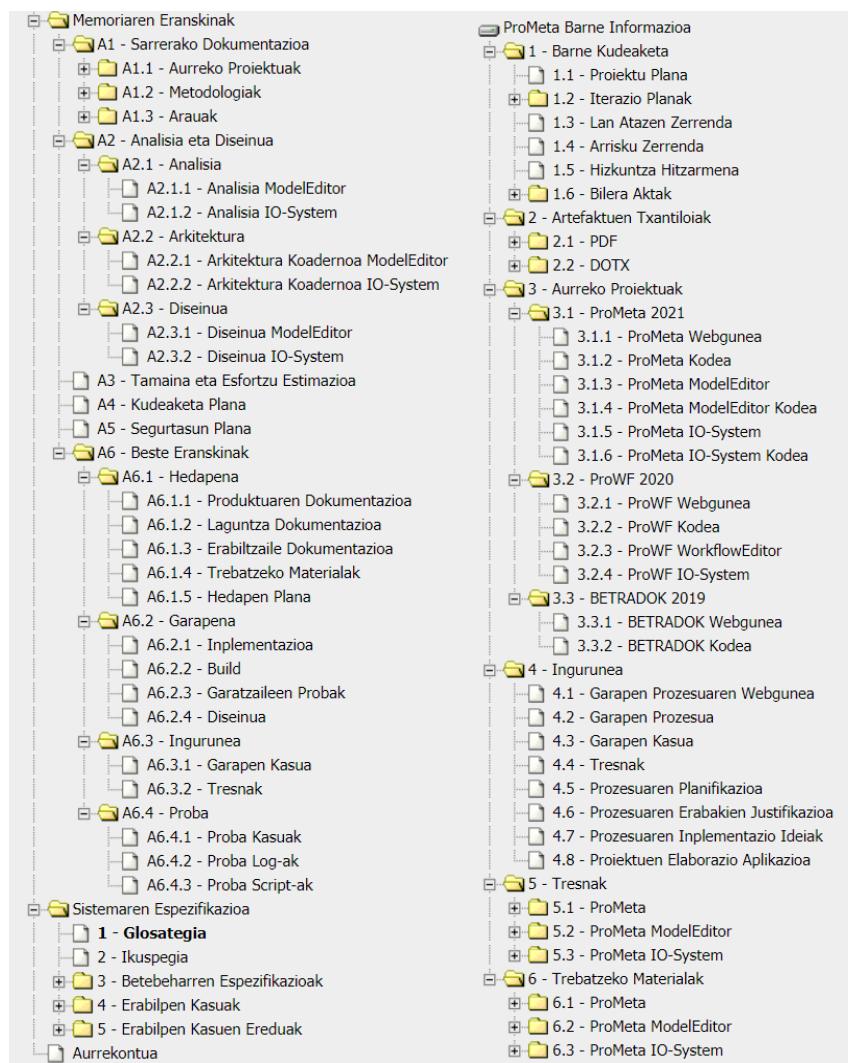
Adibidez, 1.1. Irudian memoria bistaratzeko aldean. Memoriako ataletara nabigatu daiteke menuko aukerak erabiliz edo PDF dokumentuko menua erabiliz. PDF dokumentuko menuan maila guztietako izenburuak agertzen dira, aurkibide orokorrean bezala.

The screenshot shows the '1 Sarrera' (Introduction) section of the ProMeta project website. The left sidebar contains a navigation tree for the project, including sections like 'Portada', 'Laburpena', 'Aurkezpena', 'Aurkikide Orokorra', 'Memoria', 'Hizkuntza', 'Sistemanen Espezifikazioa', 'Aurrekontuak', and 'Skerrak'. The main content area displays the introduction text in a dark-themed browser window. The header of the browser window reads 'Memoria' and '1 / 81'. The footer of the browser window includes the university logo, the text 'INFORMATIKA FAKULTATEA FACULTAD DE INFORMÁTICA', and 'Informatika Ingeniaritzako Gradua'. Below the browser window, there is a footer for 'Grado Amaierako Lan' (Graduate Work) with the text 'Software Ingeniaritza'. At the bottom right, there is a logo for 'ProMeta' with the text 'Metaereduetan oinarritutako softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema'.

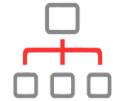
1.1. Irudia. Proiektuaren webgunean memoria bistaratu.



Esan bezala, memoriaz gain dokumentu eta esteka asko daude webgunean. 1.2. Irudian menuko dokumentu gehienak ikus daitezke. Ezkerrean memoriaren eranskinak, sistemaren espezifikazioa eta aurrekontua daude, CCII arauak eta OpenUP metodologiak zehazten dituenak. Eskuinean proiektuaren barne informazioa agertzen da, bezeroari ez zaiona bidaliko. Bertan proiektuaren barne kudeaketa, txantiloia, aurreko proiektuak, ingurunea, tresnak eta trebatzeko materialak daude.



1.2. Irudia. Webguneko eranskinen eta barne informazioaren menuak.



2 Helburuak

Izenburuak dioen moduan, ProMeta **Softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema metaereduetan oinarrituta** da. Beraz, proiektuak bi helburu nagusi ditu: garapeneko prozesuen definizioa eta prozesuaren ezarpena.

Lehenengo helburuan, softwarearen garapeneko prozesuak definitzeko metaeredu bat definitu beharko dugu eta gutxienez OpenUP metodologiaren eredu. Eredua aldatu ahal izateko, editore grafiko bat eta testu editore bat sortuko ditugu. Bi editoreen arteko bateragarritasuna bermatu beharko dugu, edozein momentutan bien artean aldatu ahal izateko.

Bigarrengoa, metodologien ereduak informazioa erabiliz prozesua ezartzen duen webgune bat sortu beharko dugu. Horrek garapen taldeari prozesua jarraitzen lagunduko dio, bakoitzak uneoro egin behar duena argi utziz. Webguneak prozesuen informazioa eta proiektuena bistaratzen eta aldatzeko aukera emango du, garapen prozesurako behar den informazio guztia bateratzuz.

Enpresa edo garatzaileen ikuspegitik bi dira helburuak. Alde batetik, **ekoizpen-prozesu sistematiko bat izatea**, ezinbestekoa etengabeko hobekuntza gauzatu eta kalitezko produktuak sortzeko. Bestetik, **ekoizpen-prozesu hori sistematizatzeko baliabideak metodologia, arau eta estandarretatik ateratzea**. Beraz, ekoizpen-prozesuko artefaktuak berrerabili ahal izateko azpiegitura teknologiko bat sortu beharko dugu.

Informatika Ingeniaritza Graduko ikasle bezala, Software Ingeniaritzan espezializatuta, lan honen egilearen helburua proiektuaren motibazioarekin bat dator: software garapenerako garrantzitsuak diren aspektuak bereganatzea. Adibidez, gaur egungo bezeroen eskakizun ez-funtzionalak betetzea, softwarearen kalitateari dagozkionak.

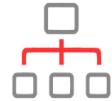
Proiektuaren helburuak hobeto ulertzeko, komenigarria da hiru aspektu zehaztea: arazoa, produktua eta interesatuak. Ondorengo azpiataletan deskribatuko dira horiek taulen bidez.

2.1 Arazoaren Planteamendua

Proiektuaren testuingurua ulertu ahal izateko, lehenik konpondu nahi dugun arazoa ulertu behar da. 2.2. Taulan agertzen den informazioa jakin behar dugu: arazoa zein den, nori eta nola eragiten dion eta gure soluzioa zein den.

Arazoaren Deskribapena	Proiektu informatikoen elaboraziorako ezagutza erabilgarria pilatzen da zenbait jarduera modu automatikoen egiteko, adostutako gida bati jarraituz. Gida hauek softwarea garatzeko prozesuetan, metodologietan eta estandarretan oinarritzen dira. Beharrezko da gida hauek definitu eta kudeatu ahal izatea, mantendu eta hobetu ahal izateko. Horrek giden edukia eta horiek exekutatzen dituen sistema definitzea eskatzen du. Honetan oinarritura bezeroen CMMI eskaerak bete daitezke.
Interesatu Kaltetuak	Proiektu informatikoak garatzen dituzten erakundeak.
Arazoaren Eragina	Estandarizazio maila baxua software proiektuen garapen prozesuan. Desadostasunak, atzerapenak eta akatsak eragiten ditu honek.
Soluzioaren Abantailak	Softwarea garatzeko prozesua definitu, kudeatu, mantendu eta hobetzeko erraztasuna. Garapen denbora asko gutxituko du eta bizi-ziklorako metodologia bat erabiliz estandarizazio-maila handituko du.

2.1. Taula. Arazoaren Planteamendua.



2.2 Produktuaren Planteamendua

Arazoa ulertu ondoren, produktuaren inguruko planteamendua azalduko dugu 2.2. Taulan. Bezeroa eta beharra, produktua, alternatibak eta horiekiko abantaila eta hobekuntzak zein diren jakin behar dugu.

Bezeroa	Software proiektuak garatzen dituen enpresa, domeinuko bezeroen epaien menpe dagoena.
Beharra	Softwarea garatzeko prozesuaren bizi-zikloa definitu eta exekutatzen duen sistemaren beharra, prozesua mantendu eta hobetu ahal izateko.
Produktua	ProMeta: Softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema metaereduetan oinarrituta.
Abantaila	Softwarea garatzeko prozesua definitu, kudeatu, mantendu eta hobetzeko erraztasuna.
Alternatiba	ProWF: Software proiektuen elaboraziorako workflowetan oinarritutako sistemaren sorkuntza eta bizi-zikloa definitzeko metodologia baten ezarpena. Ez ditu OpenUp bizi-zikloaren fase guztiak, hasiera eta elaborazio fasearen zati bat bakarrik.
Hobekuntzak	ProMeta sistemak metaereduak erabiltzen dituenez erraztasuna ematen du aldaketak egiteko. Etorkizunean sistemaren atal bat aldatzea erabakitzenean, metaeredua edo ereduak aldatzea nahikoa da. OpenUp metodologiaren bizi-zikloa osatzea du helburu. IO-System sistemaren editorea hobetzea ere aurreikusten da. Gainera, sistema zerbitzari batean jarriko da, edozein erabiltzailerentzat eskuragarri egon dadin.

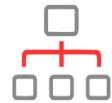
2.2. Taula. Produktuaren Planteamendua.

2.3 Interesatuen Deskribapena

Arazoa eta produktuaren planteamendua jakinik, proiektuaren interesatuak eta horien erantzukizunak deskribatzen dira 2.3. Taulan.

Izena	Deskribapena	Erantzukizunak
Software Enpresa	Software proiektuak sortu eta sortutako proiektuaren partaideak aukeratzen ditu.	Proiektua sortu Partaideak aukeratu
Administratzailea	Sistema kudeatzeaz arduratzen da.	Erabiltzaileen kontuak kudeatu Sisteman aldaketak egin
Proiektuko Kudeatzailea	Software proiektuaren liderra da, interesdunen arteko harremanak koordinatzen ditu, eta proiektuko taldea helburuetan enfokatuta egotea du ardura.	Proiektuaren ardurak eta rolak esleitu Prozesuaren gida automatikoak jarraitu
Proiektuko Analista	Software proiektuaren eskakizunak eta betebeharrok harrapatzen ditu.	Proiektuaren eskakizunak eta beharrak eskuratu Prozesuaren gida automatikoak jarraitu
Proiektuko Prozesu Ingeniarria	Software proiektuaren eskakizunak eta betebeharrok aztertz, prozesuak sortzen ditu.	Prozesua sortu/editatu, metaeredu eta ereduetan aldaketak eginez. Prozesuaren gida automatikoak jarraitu
Proiektuaren beste taldekideak	Software proiektuaren arkitektoak, garatzaileak, testerrak...	Prozesuaren gida automatikoak jarraitu

2.3. Taula. Interesatuen deskribapena.



3 Aurrekariak

Kapitulu honetan, proiektua burutu ahal izateko garrantzia izan duten iraganeko elementu esanguratsuak jasotzen dira. ProMeta proiektua ProWF proiektuaren jarraipena denez, aurrekariak antzekoak dira eta aurreko projektuan oinarrituta daude.

Hasteko, software kalitatea eta gaur egungo bezeroen eskakizun gogorrak aztertuko dira. Ondoren, bezeroen eskakizunengatik agertu diren beharrak azalduko dira: softwarea garatzeko metodologiak eta proiektuak aurkezteko araua. Hori azaldu eta gero, *BPM (Business Process Management)* sistemek eskaintzen dituzten abantailak eta sortzen dituzten menpekotasunak aztertuko dira. Jarraian, inferentzia motorrak eta haien ahalmena komentatuko da, *CLIPS* eta *EHSIS* inguruneak aztertuz. Segidan, *Ereduak Bideratutako Ingeniaritza (MDE)* gauzatzeko softwarea garatzeko metodologia azalduko da eta *Edukiak Kudeatzeko Sistemak (CMS)* softwareak deskribatuko dira. Amaitzeko, ProWF sistemaren ezaugarriak aztertuko dira, aurretik azaldutako kontzeptuak kontuan hartuz.

3.1 Software Kalitatea

Softwarearen kalitatea bezeroen beharrak asetzan dituen softwarearen ezaugarriak kontrolatzean eta ziurtatzean datza, bezero eta erabiltzaileen onurarako. Softwarea produktua ez da fabrikatzen eta ez da fisikoki degradatzen, baina garapen-prozesu bat du. Hala ere, softwarea ez da ia inoiz perfektua izaten, asebete behar den bezeroaren arazoetan ezaugarri berriak agertzen direlako. Proiektu ororen helburua ahalik eta kalitate oneneko softwarea ekoiztea da, bezero eta erabiltzaileen itxaropen eta betekizunak gaindi ditzan. Softwareak akatsak eta gorabeherak izan ditzake, baina ez dira edozein ekipo fisikoren antzekoak, horietan ausazko hutsegiteak eman baitaitezke. Softwarearen kasuan, denak sistematikoak dira eta, ondorioz, zuzendu beharrekoak.

Softwarearen ziurtagiria bere kalitatea ziurtatzeko prozesuaren ondorioa da, baina ez da inoiz azken helburua. Softwarearen kalitatea ez da ziurtatzen, bermatu behar direnak kalitatezko softwarea eraikitzeko prozedurak dira. Prozedurek, bezeroek eskatutako kontsentsu maila altuko nazioarteko estandarretan oinarritutakoak izan behar dute eta, herrialde bakoitzean, normalizazio agentzia ofizialaren berariazko lan-taldeak egokitutakoak. Prozeduren kalitate-eredu ezberdinak daude, garrantzitsuenak *CMMI* eta *SPICE* dira, helburu berdina dutenak. Kalitate-ereduek, softwarea garatzeko praktika onenak definitzen dituzte, softwarea garatzen duten erakundeen prozesuak hobetzen laguntzeko. Ziurtagiri-emaileek, praktika on horiek hartu eta beraien emaitza neurgarriak egiaztatzen dituzte garatzaileekin lankidetzan. Ebaluazio-prozesu batean parte hartu ahal izateko implikatu guztiak ziurtagiri egokiaren jabe izan behar dira.

3.2 Bezzeroen Eskakizun Gogorrak

2007ko urriaren 30ean onartutako *Ley de Contratos del Sector Público* legeak, BOE-261-2007-18874, bere 69. artikuluan, hornitzaireek kalitate bermeak erakusteko arau europarrei erreferentzia egin eta, dagozkien ziurtagiriak arauarekin bat datozen erakundeek emanak izan behar dutela ezarri zuen.

Ondoren, aurreko legearekin bateratuta, 2011ko azaroaren 14an Estatuko Aldizkari Ofizialean onartu zen legearen ostean, BOE-A-2011-17887, bezzeroen eskakizunak sendo gogortu ziren, softwarearen kalitateari dagokionez. Lege horren 80. artikuluak kalitatea bermatzeko arauak betetzen direla egiaztatzea zuen helburu, horretarako erakunde independenteek emandako ziurtagiriak beharrezkoak ziren merkatuak exijitzen bazituen. Erakunde horiek kalitatea bermatzeko Europako arau jakin batzuei erreferentzia egin

edo baliokideak izan behar ziren. Hortaz, lege horren ostean software garapenean kalitatea bermatzeko *CMMI* edo *SPICE* erakundeei lotutako ziurtagiriak lortzea beharrezkoa bilakatu zen.

CMMI erakunde baten softwarea garatzeko prozesuaren heldutasuna ebaluatzen eta neurtzen metodoak erabiltzen dituen prozesu bat da. Erakundeen heldutasun-maila neurtzeko bost etapa bereizten ditu.

Maila	Etapa	Prozesuaren egoera
I	Hasierakoa	Kaotikoa da, kontrol zaila du, errektiboa da
II	Kudeatuta	Projektuen ezaugarriak ditu, askotan errektiboa da
III	Definituta	Erakundeen ezaugarriak ditu, proaktiboa da
IV	Kuantitatiboki kudeatuta	Erdi-mailako prozesua da, kontrolatuta dago
V	Optimizazioa	Etengabeko hobekuntzan dago

3.1. Taula. *CMMI*-ren 5 heldutasun-mailak.

*CMMI*ren 3. heldutasun-maila lortzea ezinbestekoa zen merkatuan mantentzeko, hori dela eta, hainbat enpresa kexatu ziren merkatutik kanpo geratzeagatik, baina alferrikakoa izan zen. Horren adibidea, 2016an *OESIA NETWORKS, S.L.* enpresak jarritako erreklamazioa da, *Recurso 0006-2016*, honek *CMMI*ren 3. heldutasun-maila lortzear zegoela erreklamatzen zuen merkatuan mantentzeko, tramitazioren faltan zegoen. Hala ere, administrazioaren erantzuna errekurtsoa baistearen aurka egotea izan zen, lehen aipatutako 80. artikulua ez betetzeagatik. Arrazoia enpresak oraindik ez zuela eskatutako kalitate maila bermatzen zuen ziurtagirik edo ziurtagiriaren baliokiderik izan zen eta gainera, mala hori zuten software enpresa asko zeudela argudiatuz.

Ebidentzia horiek direla eta, gaur egun proiektu informatiko baten bezeroek dute pisu handiena edo agintea, haien exijentzia gogorrak betetzea funtsezkoa da merkatuan mantentzeko. Bezeroen behar eta ametsetan fokatzea, 2018ko *CMMI 2.0* bertsioan eta estandar internazionalen sendoki indartuta geratuta da.

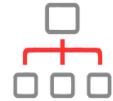
3.3 Garapen Metodologiak

Gaur egun pentsaezina bilakatu da software proiektu bat aurrera ateratzea metodologiarik jarraitu gabe. Software munduan artisautza lanak ez du etorkizunik, metodologia baten ezarpenak hori ekiditeko balio du. Software ingeniariek edozein momentutan jakin behar du zer egin, noiz eta nola, bestela arazoak eta galerak agertzenko probabilitatea handitzu joango da proiektuaren garapena luzatzen doan heinean.

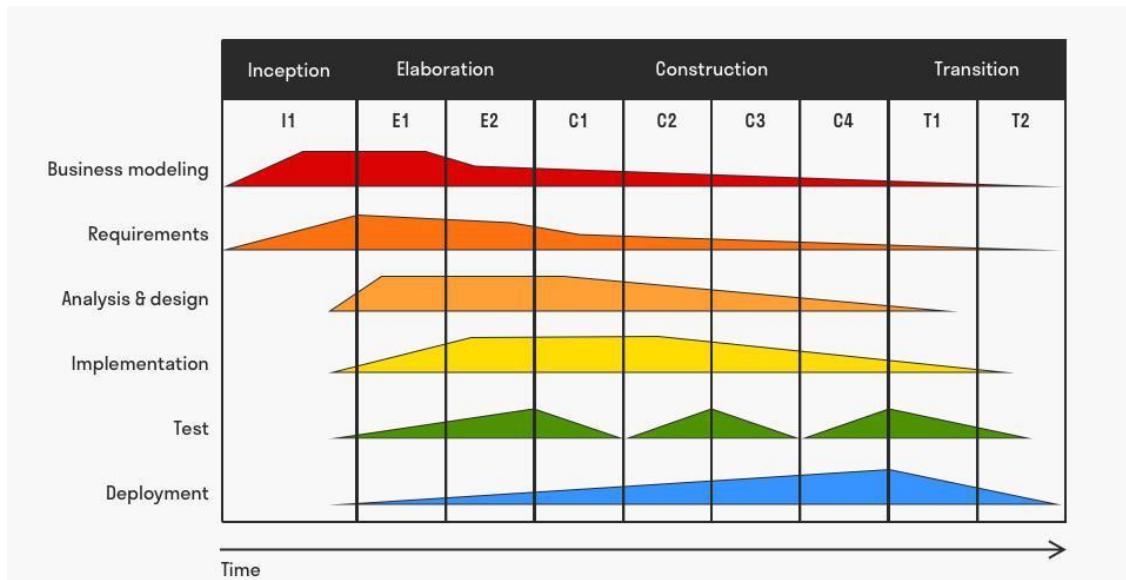
Gainera, proiektuak zerotik egitea garestiegia da software garapen enpresentzat. Berrerabilpenean oinarritutako metodologia bat aukeratuz, enpresaren kostu ekonomikoak gutxituko dira epe laburrean.

Gehien erabiltzen direnak metodologia arinak edo bizkorak dira. Arina kontzeptua softwarea sortzeko urratsak arindu behar direlako sortzen da, urrats bakotzeko prototipo bat sortuz. Giza interakzioetan zentratzen da, aldeen arteko elkarritzeta-fluxuari eutsi ahal izateko, garapen dinamikoagoa eta parte-hartzaileagoa ahalbidetzeko. Metodologia bizkorrek garapen-sistema egokitzale bat erabiltzen dute, eta ez prediktiboa. Horrek esan nahi du lantaldeak buruan duela nahi duen emaitza, baina ez daki zehatz-mehatz zer produktu mota sor dezakeen.

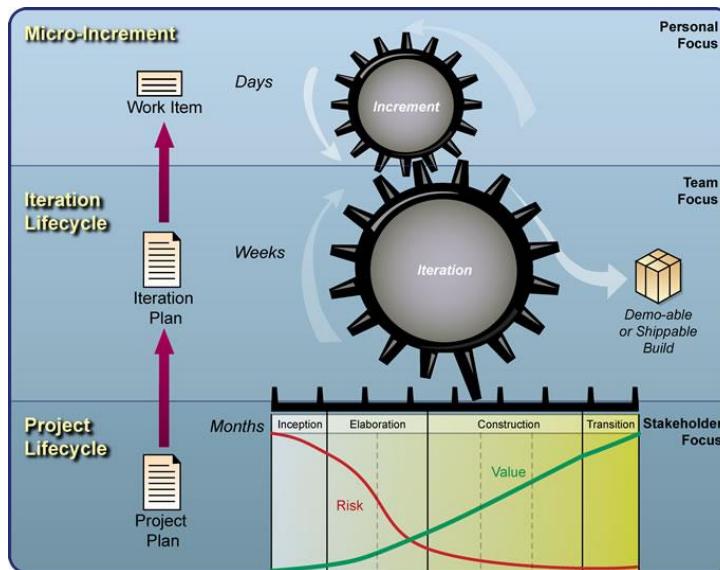
Software garapenaren metodologia arinen artean RUP aurkitzen da. Softwarea garatzeko prozesu iteratibo bat da, Rational Software Corporation erakundeak sortua, IBM enpresaren dibisio bat. 3.1. Irudian ikus daiteke RUP prozesuaren egitura. Hala ere, RUP ez da zehatz-mehatz jarraitu behar den



prozesua, baizik eta prozesu moldagarria da, garapen-erakundeek eta software-proiektuen taldeek egokitzeko asmoarekin, bakoitzaren premietarako egokiak diren elementuak hautatuz.

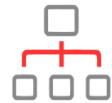


3.1. Irudia. RUPen prozesu iteratiboa, bizi-zikloaren faseak, iterazioak eta jarduerak.



3.2. Irudia. OpenUP-en geruzak: mikro-gehikuntzak, iterazio bizi-zikloa eta proiektu bizi-zikloa.

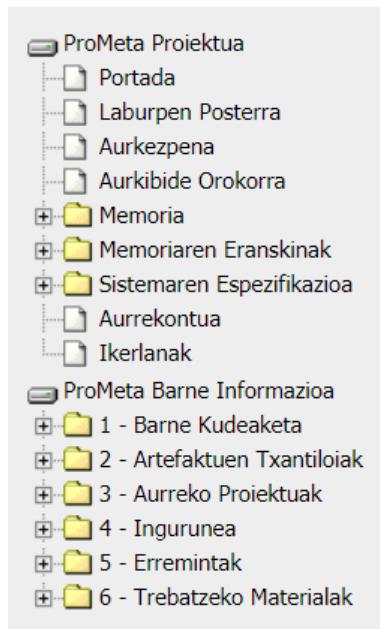
RUPen oinarritutako metodologia simple eta erabiliena OpenUP da. Metodologia horrek RUPen funtsezko ezaugarriak gordetzen ditu, garapen iteratiboa, erabilpen-kasuak, arriskuen kudeaketa eta arkitekturan oinarritutako ikuspegia bultzatzen duten agertokiak barne. RUPen erabiltzen ez diren aukerako zati gehienak baztertu eta elementu asko bateratzen ditu. Emaitza prozesu askoz simpleagoa da, eta RUP printzipioekiko leiala izaten jarraitzen du. 3.2. Irudian ikus daiteke OpenUP prozesuaren laburpen bat. Proiektuaren tamainaren arabera OpenUPtik RUPera pasatzea komenigarri ikusiko dugu, informazio zehatzagoa gorde ahal izateko.



3.4 Proiektuen Aurkezpenerako Arauak

Proiektuen garapenean zehar sortzen diren dokumentazio multzoaren aurkezpenerako ezinbestekoa da ezarrita dagoen araudi ofiziala betetzen dela bermatzea, nazioarteko estandarren oinarritutakoa. Helburu nagusia proiektuan parte hartzen duten alderdi guztien aldeko dokumentazioa osoa eta gardena izatea izanik, bezeroaren gogobetetasuna handituz.

Naziortean ISO erakundeak ezarritako estandarrak jarraituz, Spainian araurik finkatuenak *UNE* eta *CCII* erakundeek aurkeztutakoak dira: **CCII-N2016-02. Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática.** 3.3. Irudian ikusten den moduan, zehatz-mehatz deskribatzen du nola egin behar den Informatika Ingeniaritzako proiektu baten dokumentu-zehaztapena. Proiektuko dokumentazioa egiteko ereduak eta dokumentazioaren antolamendua deskribatzen du ere. Erreferentiazko nazioarteko esparru eta estandarrak kontuan hartzen ditu. Hala nola, *UNE 157801:2007 – Informazio-sistemen proiektuak egiteko irizpide orokorak*, *UNE-ISO 21500:2013 – Proiektua zuzentzeko eta kudeatzeko jarraibideak eta PMIren PMBOK – Proiektuen Zuzendaritzarako Oinarrien Gida*. Proiektu baten bisa edo ikuskaritzak egiteko **CCII-N2016-01** araua ere eskuragarri dago.



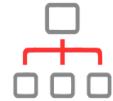
3.3. Irudia. CCII-2016N-02 araua betetzen duen proiektuaren webgunearren egitura.

Proiektuak aurkezteko araudien gabeziak proiektu informatiko batean gatazkak ekarri ditzake. Hau da, proiektuan esku hartzen duten aldeentzat nahi ez diren ondorioak eragin, bezero, hornitzaire zein interesdunen arteko gatazkak sortuz. Epaitegietan bukatzea oso kaltegarria da edozein software enpresentzako.

3.5 Prozesuetan Oinarritutako Garapena

Business Process Management (BPM) negozio-prozesuak definitzera eta aplikatzen bideratutako metodologia edo ikuspegি estrategiko gisa ikus daiteke. Negozio-prozesuak konplexuak eta dinamikoak dira. Gainera, malguak izan behar dute, negozioa nabarmen aldatzten joango delako eta etengabe eguneratu behar direlako.

Hona hemen BPMrekin lan egitean lortzen diren abantailak:



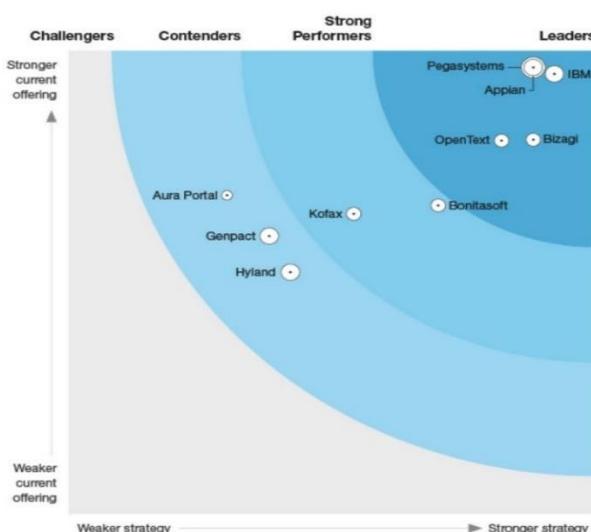
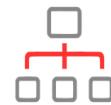
- Ataza errepikakorrik ezabatzea eta automatizatzea.
- Eraginkortasuna handitzea: prozesuetan akatsak minimizatzuz, itxarote-denbora murriztuz, giza esku-hartzeak murriztuz eta lana berregitea saihestuz
- Negozio-arauak betetzen direla ziurtatzea.
- Zerbitzu-maila bermatzea, salbuespenak maneiatuz, egoeren jarraipena eginez, gertakariak mailakatuz, prozesuen sendotasuna eta trazabilitatea bermatuz, etab.
- Lan egiteko modua aldatzeko aukera eskaintza, eragiten duen inpaktua murriztuz eta etengabe hobetuz.

Gartner eta *Forrester* aholkularitza-enpresek BPM merkatua ikertzeko, urtero, txosten bat argitaratzen dute. Txosten horiek patentatutako datu kualitatiboak aztertzeko metodoetan oinarritzen dira, merkatuaren joerak frogatzeko, hala nola zuzendaritza, heldutasuna eta parte-hartzaleak.

Gartnerren Koadrante Magikoa industria teknologikoaren azpisektore nagusiei buruz egiten duen merkatu-ikerketa bati erantzuten dio. Bertan, merkatu-joerak, tartean dauden enpresta nagusiak eta heldutasun teknologikoa besteak beste aztertzen dira. 3.4. Irudian lau koadrantetan banatutako grafiko gisa aurkezten da. X ardatzak exekutatzeko gaitasuna adierazten du, eta Y ardatzak, berriz, balio-proposamen osoa. Ezkerretik eskuinera eta goitik behera, sektore bakoitzeko enpresak honako koadrante hauetan kokatzen dira: *challengers*, *leaders*, *niche players* eta *visionaries*. Antzeko informazioa aurkitu dezakegu 3.5. Irudian agertzen den *Forrester Wave* txostenean. Bi txosten ezagun hauek, ondo kokatutako hornitzaillek komertzialki erabiltzen dituzte.



3.4. Irudia. 2018ko BPM Suiten Koadrante Magikoa, Gartner.



3.5. Irudia. 2019ko The Forrester Wave txostena, Forrester.

Bizagi bi txostenetan agertzen da, ondo kokatuta, gainerako “liderrak” tamaina handiko enpresa eta erakundeen prozesu kudeaketan fokuratzen dute. *Bizagi* bezalako teknologia erabiliz arkitektura konplexuko web-aplikazioa implementatu daiteke, prozesu, erregelea eta *workflowak* exekutatzeko motorrak integratuak dituena. Apustu hori garestia izan daiteke eta proiektu edo enpresaren menpekotasun teknologikoa handituko sortu. Beste apustu merkeago bat, metodologia eta estandarretara hurbiltzen joateko, teknologia propioa sortzea da, *workflow-lengoia* propioa sortuz eta lengoia horretan adierazitako ezagutza exekutatzeko inferentzia motorrak erabiliz.

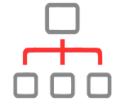
3.6 Ezagutzaren Kudeaketa eta Inferentzia Motorrak

ProWF proiektuan teknologia propioa sortzea erabaki zen, lan-fluxu edo *workflowsen* bidezko lengoia propioa sortuz eta berekin adierazitako ezagutza inferentzia motorren bidez exekutatu.

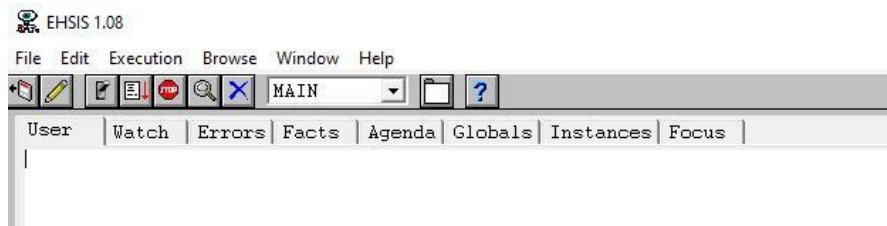
CLIPS softwareak inferentzia motor baten erabilera errazten digu, sistema adituak ekoizteko eta exekutatzeko garapen-ingurune bezala erabili daiteke. *NASA* sortu eta erabilia jabari publikoan utzita dago. Bere lengoiaiak erregelak, objektuetara bideratutako programazioa eta programazio prozedurala erabiltzen du ezagutza adierazteko. Lengoia simplea baino oso ahaltsua da, hurrengo ezaugarrietan nabarmenduz:

- **Garraigarritasuna.** C lengoaiak ematen diona.
- **Integrazioa edo zabalkortasuna.** Programazio prozeduralari esker funtzi berri ahaltsuak sora daitezke. *Service-oriented Architecture* (SOA) estandarra erabilita BPM edo beste edozein sistemekin integragarria da.
- **Interakzio edo diziplinarteko garapena.** Formakuntza ezberdinako pertsonen ideiak azkar implementatzeko aukera lengoia simple eta ahaltsu bat erabilita, objektu eta erregeletan oinarrituta.
- **Egiaztapen edo balidazio errazagoa.** Horretarako, funtzi bereziak erabiliz.

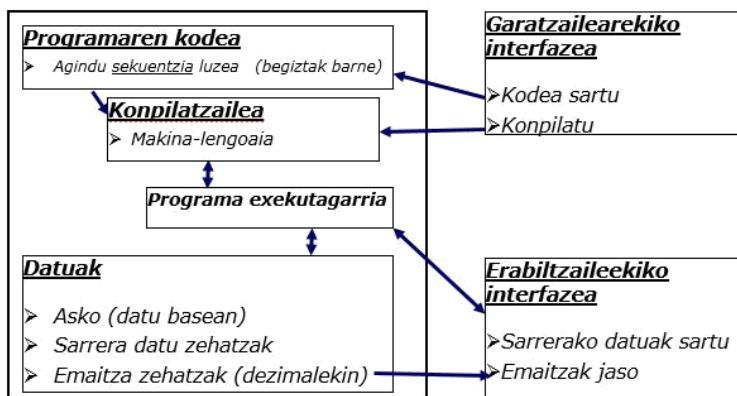
EHSIS, ordea, Euskal Herriko Unibertsitateko (EHU) *ERABAKI* taldeak hedatutako ingurunea da, *CLIPS 6.04*, *FuzzyCLIPS¹⁴ 6.04*, objektuetara eta gertaeretara bideratutako programazioa, interfazeen garapena, komunikazioa eta leioetan oinarritutako ingurunea integratzen duena. *EHSIS* inguruneak (ikusi 3.6. Irudia), softwarea garatzeko baliabide tradizionaletan oinarritutako aplikazioak sortzeaz gain (ikusi 3.7.



Irudia), COOL¹⁵ lengoia erabilita, ezagutzan oinarritutakoak ere sor ditzake sistemaren arkitektura egoki batekin baliatuz.

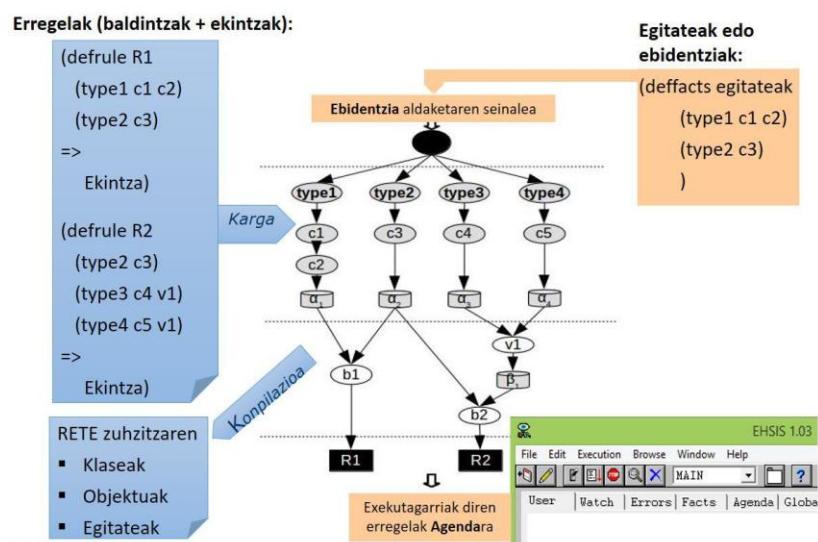


3.6. Irudia. EHSISen garapen ingurunea.

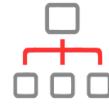


3.7. Irudia. Programazio tradizionalaren eskema.

Erregelak idazteko lengoia simplea denez, klase eta objektuak erabilita ere, domeinuko ezagutza adierazteko ondo diseinatutako lengoia bat lortuz gero, ezagutza hori exekutatzeko kodea automatikoki sortzea ere posible da. Gainera, inferentzia motorrarekin abiadura handiko exekuzio eraginkorra lortu daiteke kode guztia RETE sare bezala gordeta (ikusi 3.8. Irudia), hau “konpilazio” mota bat bezala erabili daiteke. RETE sarearen egitura bereziak RETE algoritmoaren abiadura, egitate eta erregela askorekin ere, aprobetxatzeko aukera ematen du. Erreminta hau, bere eraginkortasuna eta jabari publikoa dela eta, konplexutasun handiko proiektuetan erabili daiteke produktu mantengarriak lortzeko.



3.8. Irudia. Erregeletan oinarritutako sistemen eskema: klaseak, objektuak, erregelak eta egitateak.



EHSISen garapen inguruneak badu bere produkzio bertsioa, *EHSIS_RT* deitutakoa. Web-zerbitzuetarako bertsioa ere badu, Mairi deitutakoa.

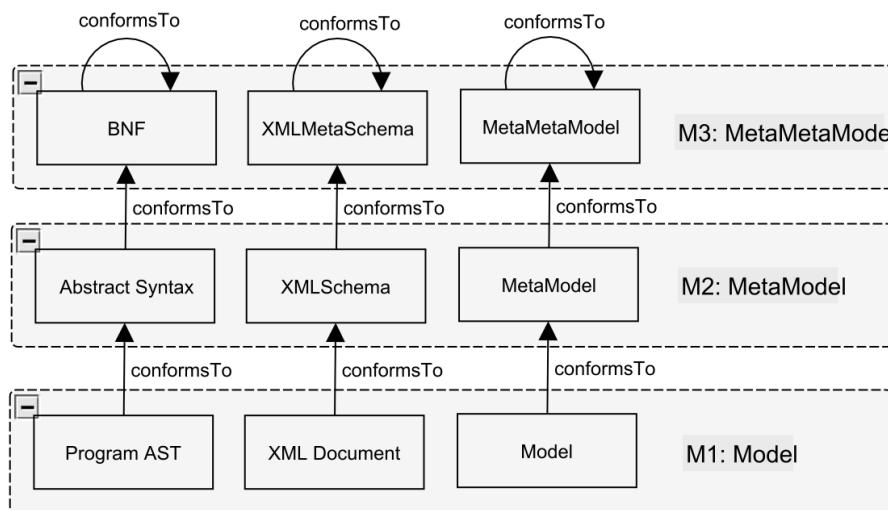
3.7 Ereduek Bideratutako Ingeniaritzar

Model Driven Engineering (MDE) edo eredu bidezko ingeniaritzar softwarea garatzeko metodologia da. Domeinu ereduak erabiltzen ditu, hau da, arazo zehatz bat lotutako gai guztiak eredu kontzeptualak. Hori dela eta, aplikazioen domeinu jakin baten ezagutzaren eta jardueren irudikapen abstraktuak nabarmenak du helburu, kontzeptu informatikoetan sartu gabe.

MDEren helburua produktibitatea handitzea da. Horretarako, sistemen arteko bateragarritasuna maximizatzen du eredu estandarizatuak berrerabiliz, diseinu prozesua simplifikatu aplikazioaren domeinuko diseinu patroi erreplikakorren ereduak bidez eta sistemak lan egiten duten pertsonen eta taldeen arteko komunikazioa sustatu praktika onen estandarizazioaren bidez.

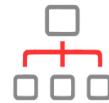
MDEren modelatze paradigma bat eraginkorra dela deritzo bere ereduek domeinua ezagutzen duen erabiltzaile baten ikuspuntutik zentzu badute eta sistemak ezartzeko oinarri gisa balio badute. Ereduek produktuen kudeatzaileen, diseinatzaileen, garatzaileen eta aplikazioaren domeinuko erabiltzaileen arteko kolaborazioarekin garatzen dira. Ereduek amaitzen doaz heinean, softwarea eta sistemak garatzea ahalbidetzen dute.

MDEn 4 modelatze maila daudela esan dezakegu. Maila handiagoa den heinean abstrakzio-maila igotzen doa. 3.9. Irudian ikus daitezke 3 abstrakzio-maila altuenak: eredua, metaeredua eta meta-metaeredua.



3.9. Irudia. MDEren 3 abstrakzio maila: eredua, metaeredua eta meta-metaeredua.

- **M0. Instantziak.** Maila hau exekuzio sistemari dagokio. Maila honetan negozioko elementuak daude, edo mundu errealeko elementuen errepresentazioak (software errepresentazioak).
- **M1. Eredua.** Eedu honek software sistemaren itemak adierazten dituzten kontzeptuak dauzka. M1 mailan dauden kontzeptuek M0 mailan dauden instantziak kategorizatu edo sailkatzen dituzte.



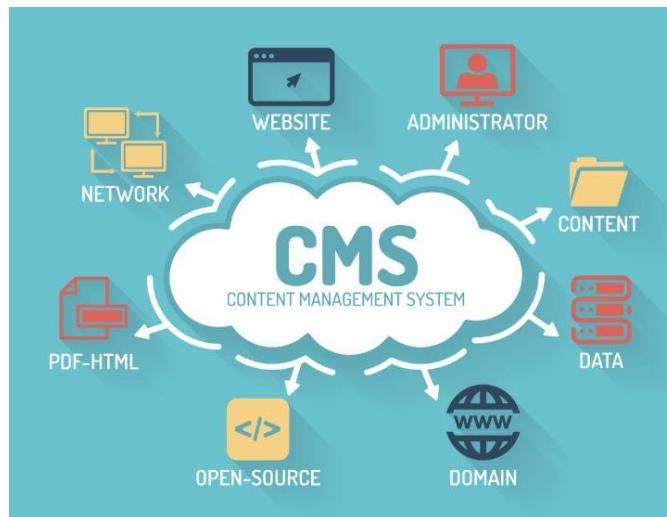
- **M2. Metaeredua.** M1 mailan dauden kontzeptuen inguruan arrazoitzeko beharrezkoak diren kontzeptuak dauzka. M2 mailako elementu batek M1-eko elementuak espezifikatzen ditu. Ereduen eredu honi metaeredu esan ohi zaio.
- **M3. Meta-metaeredua.** M2 mailan dauden kontzeptuen inguruan arrazoitzeko beharrezkoak diren kontzeptuak dauzka. M3 mailako elementuek M2-ko elementuak kategorizatzen ditu. Meta-metaeredu esan ohi zaio.

ProMeta proiektuan, MDE ModelEditor azpisistema garatzerako erabiliko da.

3.8 Edukiak Kudeatzeko Sistemak

Content Management System (CMS) izeneko softwareak dokumentuak eta bestelako edukiak antolatu eta kudeatzeko erremintak dira, webguneak eta web edukiak kudeatzeko balio dutenak. Gaur egun sistema asko daude arlo honetan, bai kode irekikoak eta baita jabedunak ere.

Edukiak kudeatzeko sistema zerbitzarian kokaturik egon ohi da. CMS batera atzipena erabiltzaile motaren araberako maila ezberdinetan egin daitake. Adibidez, edukiaren sortzaileek sistema osatuko duten dokumentuak sortuko dituzte. Argitalpen-teknikariek dokumentu hauek aipatu, onartu edo baztertuko dituzte. Editoreen burua gure web orrian eduki hori argitaratzeaz arduratuko da. Dena interfaze grafiko baten bidez kontrolatuko da, honen erabilera erraztu ahal izateko. Gainera, CMS-ek hainbat funtzionalitate eskaintzen dituzte, 3.10. Irudian ikus daitezkeenak.



3.10. Irudia. CMS batek eskaintzen dituen funtzionalitate nagusiak.

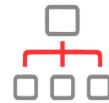
ProWF eta ProMeta proiektuen IO-System sistema garatzeko Drupal CMSa erabili da.

3.9 ProWF Sistema

Esan bezala, ezinbestekoa da ProWF sistema [1] ulertzea proiektu hau horren jarraipena delako. Horretarako, bertan zehazten diren rolak, sistemaren azpisistemak eta osagaiak deskribatuko dira.

ProWF sistemaren erabiltzaileek hurrengo rol hauekin lan egin behar dute:

- **Prozesu Sortzailea:** sortutako workflow-lengoia baliatuz, softwareen garapenerako bizi-zikloa ezartzen duen metodologia definituko duena.



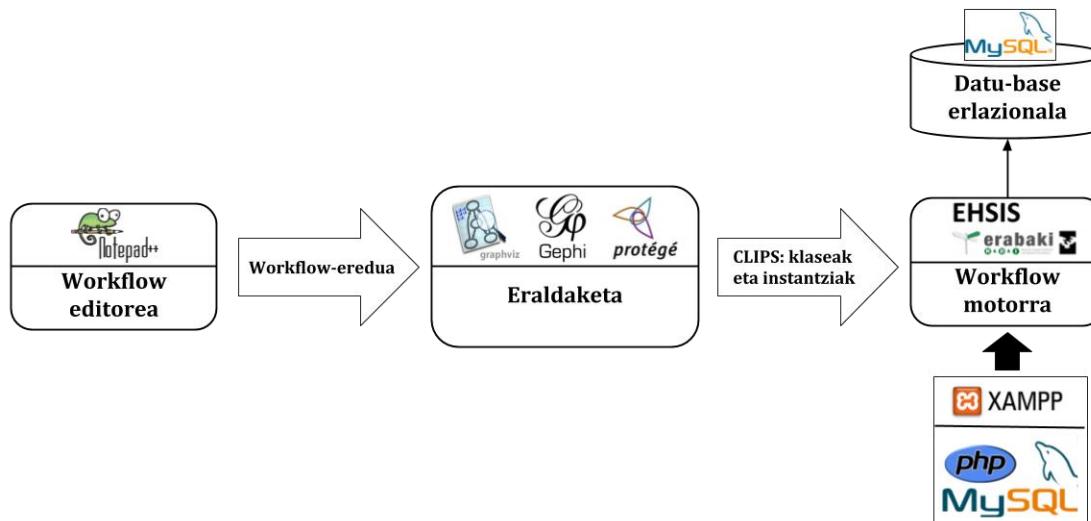
- **Analista:** sortutako workflow-ereduak adierazten duen bizi-zikloa exekutatzearaz eta aplikatzearaz arduratuko da.
- **Proiectuko Zuzendaria:** proiectuak sortu eta proiectuko partaideen rolak esleituko ditu.
- **Kalitate Arduraduna:** workflowaren exekuzioan zehar sortuko diren artefaktuen kalitatea bermatzea du helburu, balorazioak emanez.

ProWF sistema bi azpisistema ezberdinietan bananduta dago:

- **Workflow Editor:** sortutako workflow-lengoia erabiliz workflow-ereduak sortzeko balio du, ondoren, eredu horiek workflow motorrak exekutatzeko eraldatuko ditu. Testu-editore bat izango da. Rolak: Prozesu sortzailea.
- **IO-System:** workflow kudeatzaileraren sarrera/irteerak maneiatzea du helburu. CMS baten bitartez kudeatutako web-aplikazioa izango da. Rolak: Proiectuko zuzendaria, analista eta kalitate arduraduna.

3.11. Irudian ikusten den moduan, hauek dira **Workflow Editor** azpisistemaren arkitekturaren osagaiak:

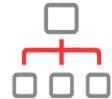
- **Workflow editorea:** definitutako lengoia grafiko bat erabilita, testu editore batekin, DOT lengoiaiaren bitartez workflow-ereduak sortzeko balio du. “Workflow-lengoiaaren Eskuliburu” dokumentuan jarri dira lengoia grafikoaren xehetasunak.
- **Eraldaketa-prozesua:** DOT lengoiaian idatzitako workflow-eredua CLIPS lengoaiako klase eta instantzieta eraldatzea du helburu. Horretarako, “Workflow Editor - Eskuliburu” dokumentu jarraitu behar da.
- **Datu-base erlazionala:** sortutako workflow-ereduen informazio gordetzeaz arduratzen da.
- **Workflow motora:** CLIPS lengoaiara eraldatutako workflow-ereduaren klaseak eta instantziak baliatuz, workflowa exekutatzeko erregela batzuen bidez, informazio guztia datu-base erlazionalean gordeko du.



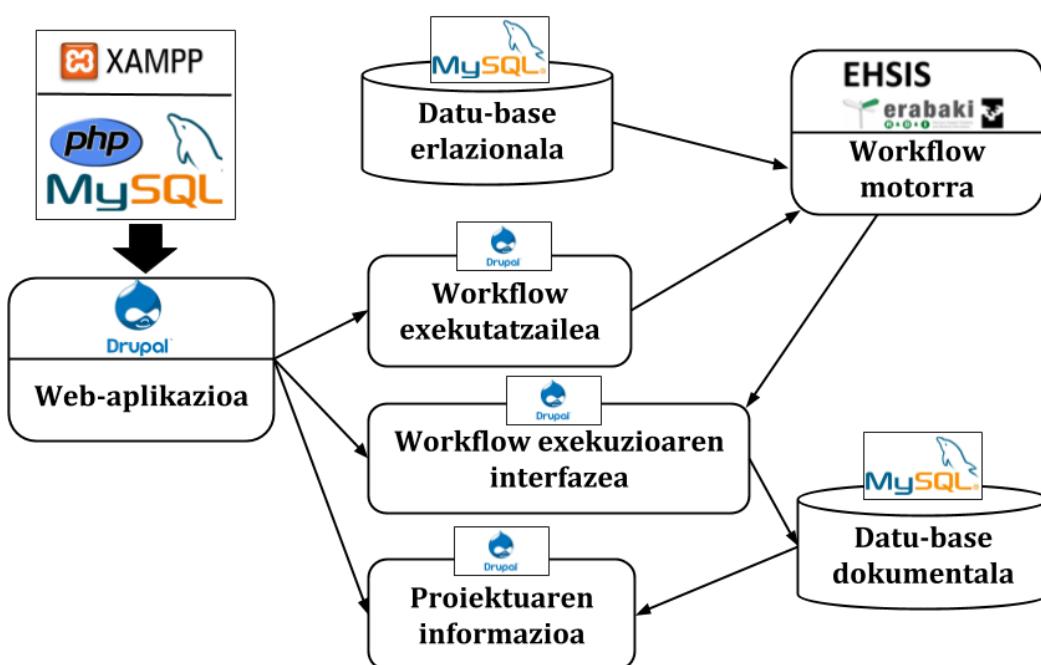
3.11. Irudia. ProWF Workflow Editor azpisistemaren arkitektura.

3.12. Irudian ikusten den bezala, hauek dira IO-System azpisistemaren arkitekturaren osagaiak:

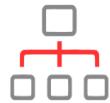
- **Web-aplikazioa:** CMS baten bitartez, lokalean erabiltzeko (zerbitzari batean ere kokatu daiteke) pentsatua dagoen sistema kudeatzailea. Hurrengo osagai nagusiak integratuko ditu:
 - **Workflow exekutatzalea:** workflow motorrari aginduak emango dizkio eta ondoren, bere irteerak jasoko ditu.

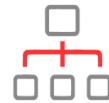


- **Workflow exekuzioaren interfazea:** erabiltzaileak exekuzioan zehar ikusiko duena da, workflow motorraren irteeran adierazitako ekintzak betetzeko interfazea da.
- **Proiektuaren informazioa:** CCII-2016N-02 araua jarraituz, erabiltzailearen dokumentuak antolatuta dituen webgune antzeko bat da. Informazioa datu-base dokumentaletik eskuratuko du.
- **Workflow motorra:** Datu-base erlazionalari konsultak eginez, workflow-eredua prozesatuko du eta jardueraz-jarduera joango da egin beharreko ekintzak irteera lez bueltatuz.'
- **Datu-base erlazionala:** sortutako workflow-ereduen informazioa gordetzeaz aparte, erabiltzaileen proiektu eta artefaktuen informazioa biltegiratuko du.
- **Datu-base dokumentala:** erabiltzaileek idatzitako dokumentuen kokapena eta sekzio bakoitzaren edukia gordeko du.



3.12. Irudia. ProWF IO-System azpisistemaren arkitektura.





4 Egungo Egoera

Kapitulu honetan, proiektuaren egungo egoera deskribatuko da, ProWF aurreko proiektuaren egoera eta egilearen prestakuntza azalduz.

4.1 Deskribapena

Egungo egoera ulertzeko, ezinbestekoa da proiektu honen aurrekaria den ProWF proiektuaren egoera ulertzea. Proiektuaren izenburuak dioen moduan ProWF sistemak *software projektuen elaboraziorako workflowetan oinarritutako sistemaren sorkunza eta bizi-zikloa definitzeko metodologia baten ezarpena* ahalbidetzen du. Sistemak bi osagai nagusi ditu, workflow editorea eta sarrera irteera sistema.

Horretarako, proiektuan zehazten diren abantailak eta desabantailak aztertu eta osatuko ditugu. Gainera, proposatzen diren hobekuntzak ere zehatzko ditugu, eta gure ideiekin osatu. Izan ere, hobekuntza horietako batzuk aurrera eramango dira ProMeta proiektuan.

4.2 Abantailak eta Desabantailak

ProWF proiekuan sistemaren abantaila eta desabantaila batzuk identifikatu ziren. Osatu egin dira proiektua aztertzeraoan identifikatu diren beste batzuekin. Garrantzitsua da hauek kontuan hartzea ProMeta proiektuaren planteamendua egiteko.

4.2.1 Abantailak

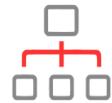
Sistemaren abantailen artean bi mota aurki ditzakegu, workflow-lengoaiari lotutakoak eta workflowetan oinarritutako sistemari lotutakoak.

Workflow-lengoaiari lotutakoak:

- Sortutako workflow-ereduen irudien nabigagarritasunak garbitasuna eta ulergarritasuna ematen dio prozesuari. Gainera, OpenUP metodologiaren webgunean agertzen diren formak eta koloreak erabiltzen ditu.
- Workflow-eredua aldagarria da, baldin eta sortutako lengoia grafikoa errespetatzen bada. Bestela, workflow-lengoia exekutatzeko erregelak egokitu beharko dira.
- Lengoaiak softwarearen bizi-zikloaren ezaugarri esanguratsuenak harrapatzen ditu.

Workflowetan oinarritutako sistemari lotutakoak:

- Interfaze simple eta intuitiboa du, itxura profesionalarekin.
- Drupal CMSari esker, erabiltzaileen erregistro eta kudeaketa erraza du.
- Workflow motorra, workflow-lengoia erabiliz sortutako edozein prozesu exekutatu dezake, prozesuaren objektuak automatikoki sortzen dira eta prozesuak exekutatzeko erregelak berrerabili daitezke.
- Workflow motorraren prozesaketa-denbora asko murrizten da, erabiltzen dituen instantzia eta erregelak RETE sarean “konpilatu” izanari esker.



4.2.2 Desabantailak

Sistemaren desabantailak ere multzo berdineta sailka daitezke.

Workflow-lengoaiari lotutakoak:

- Workflow-eredua nabigagarria denez, hainbat fitxategi eraldatu behar dira CLIPS lengoaiako klase eta instantziak sortzeko. Prozesu errepiakor eta neketsua da.
- OpenUP bizi-zikloko workflow eredu eskuz definitu beharra. Horrek prozesua definitzeko denbora asko behar izatea eragiten du. Hobe izango litzateke webguneko informazioa erabiliz automatikoki sortzea.

Workflowetan oinarritutako sistemari lotutakoak:

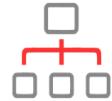
- Web-aplikazioak lokaleko instalazioa behar du. Zerbitzari batera eraman daiteke eta horrela instalazio prozesua asko murriztuko litzateke, bakarrik Workflow Editor azpisistemaren osagaia instalatzu.
- Drupalen bidez sortutako web-aplikazioak ez ditu erantzun azkarrak ematen. Gunearen orrialdez aldatzean kargatu behar diren modulu eta beste aspektuek errrendimendua murrizten diote.
- Elaboratzen den proiektuaren webguneko editorea testu soilean dagonez, ez du aukerarik ematen formatua emateko. Adibidez, ezin da letra Iodia erabili eta ez dago taulak betetzeko aukerarik.
- Webgunean OpenUP artefaktuen informazioa betez ez dira beraien txantiloiaaren formatuko dokumentuak lortzen.
- Drupaletik sortzen diren datuak kanpoko datu-base baten daude. Horrek Drupalek eskaintzen dituen aukerak murrizten ditu, datuak bistaratzea eta editatzeko aukerak, adibidez.

4.3 Proposatutako Hobekuntzak

ProWF proiektuan hobekuntza interesgarri asko proposatzen ziren. Garrantzitsua da horiek kontuan hartzea ProMeta proiektua haren jarraipena baita. Gainera, hobekuntza posible gehiago ere gehitu ditugu aurrekoak osatzeko.

ProWF proiektuan hurrengo hobekuntzak proposatzen dira etorkizunerako:

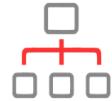
- OpenUP bizi-zikloko workflow eredu amaitu eta ahal bada, hobetu. Prozesuan gelditzen diren fasesak gehitu eta bigarren fasea (*elaboration*) guztiz definitu. Horretarako, “*Workflow-lengoaiaren Eskuliburua*” eta “*Workflow Editor – Eskuliburua*” dokumentuak jarraituz.
- *Workflow*-ereduen eraldaketa-prozesu errepiakorra ekiditeko metaeredu bat definitzea, DOT lengoia deskriptiboaren eta COOL lengoiaaren arteko eredu bat sortuz, urrats bakar baten bidez eraldaketa eginez eta kanpoko softwareak (Gephi, Protégé) erabiltzea ekidituz. Produktibilitatea, azkartasuna eta mantengarritasuna bilatuz.
- *Workflowak* kudeatzeko sistema zerbitzari batean jartzea. Zerbitzari batean egonda, erabiltzaileak ez du instalaziorik beharko.



- *Workflowak* kudeatzeko sistemean, *IO-System* azpisisteman, artefaktuen sekzioak idazterako orduan *HTML* edo *WYSIWYG* motako testu-editore bat implementatzea. Softwarearen bizi-zikloa definitzen duten metodologia askotan taulak eta Excel orriak bete behar dira, prototipo honetan, ordea, ez dago taulak txertatzeko aukerarik.
- Bezero ezberdinenei esker gaitasuna izateko asmotan, metodologia ezberdinak integratzen dituen garapen-prozesuak definitzea.
- Gure enpresak ondo egiten duena garapen-prozesuan sartzea. Hori CMMI 2.0 kalitate-ereduak eskatzen du. Prozesu berrikin integratzeko lanak konplexuak izan daitezke.
- Garapen-prozesua grafikoki adieraztea xehetasun maila handiagorekin eta funtzionalitate gehiagorekin. Lehen fase batean, lan-fluxuen eredua erabiliz eta, bigarren fase batean, BPMN estandarrak definitzen duen lengoia grafikoa erabiliz, partekatze eta adoste lanak erraztu eta azkartzeko.
- Garapen-prozesua beste metodologia batzuen baliabideekin edo adostasun-maila handiko artefaktuen txantiloiekin aberastea, adibidez, *RUP* metodologia arina.
- CMMI 2.0 kalitate-ereduaren 2. maila lortzeko garapen-prozesua osatzea.
- CMMI 2.0 kalitate-ereduaren 3. maila lortzeko garapen-prozesua osatzea.

Hobekuntza posible gehiago ere identifikatu dira proiektua aztertzerakoan:

- Drupaletik zuzenean inferentzia motorrari deitu ahal izatea, tarteko fitxategirik erabili gabe. Orain sistemak erabiltzaileari esaten dio fitxategi bat exekutatu behar duela.
- OpenUP bizi-zikloko workflow eredua eskuz definitu beharrik ez edukitzea. Horrek prozesua definitzeko denbora asko behar izatea eragiten du. Hobe izango litzateke webguneko informazioa erabiliz automatikoki sortzea.
- Prozesuaren informazioa gordetzeko metaeredua eta ereduak erabiltzea. Metaereduak erabiltze malgutasuna ematen du etorkizunean eraldaketak egiteko komeni den formatura.
- Drupaletik sortzen diren datuak kanpoko datu-base baten gorde ordez Drupalen datu-basean gordetzea. Horrek Drupalek eskaintzen dituen aukerak aprobetxatzen ditu, datuak bistaratzeko eta editatzeko aukerak, adibidez.
- Drupal webguneararen itxura hobetu, defektuzko itxura aldatuz. Itxura egoki bat aurkitu webgunerakomenu hedagarriak onartzen dituena.



4.4 Prestakuntza

Proiektu honen egileak bazituen proiektu honetarako erabilgarriak diren hainbat ezagutza, Informatika Ingeniaritzako Graduko hainbat irakasgaitan ikasitakoak. Esaterako, software proiekturen, softwarearen bizi-zikloaren, metodologia zein estandarren oinarrizko ezagutzak.

Softwarearen Kalitatea irakasgaien proiektu honetan interesa duten hurrengo ekintzak jorratu ziren:

- *BPMn* oinarritutako software bat probatu, *Bizagi*. Software horren bidez, prozesuetan oinarritutako web-aplikazioa bat sortu zen. Lehenengo, *Bizagi Modeler* softwarearen bitartez prozesua modelatu, eta ondoren, prozesu horretan oinarritutako web-aplikazioa eraiki zen *Bizagi Studio* softwarearekin.
- *OpenUP* metodologia jarraitzen zuen proiektu bat osatu, softwarearen bizi-zikloa definituz. Ez ziren metodologiako artefaktu guztiak bete, baina bai hasierako fasekoak, betekizunen ingeniaritzari buruzkoak.
- Proiektu bat aurkezteko webgunea sortu eta antolatu *CCII-2016N-02* araua jarraituz.

Softwarearen Garapen Industriala irakasgaien beste gai hauek landu ziren:

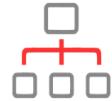
- Model Driven Engineering (MDE) edo ereduek bideratutako ingeniaritzaren oinarrizko kontzeptuak: metaeredua eta eredua.
- Domain Specific Language (DSL) edo domeinu zehatzeko lengoaien sorrera.
- ATLAS Transformation Language (ATL) erabilera ereduen arteko eraldaketak egiteko.
- Eclipse Modelling Framework (EMF) tresnen erabilera MDE-rako.

Web Sistemak irakasgaien ikasitakoa:

- XAMPP-en erabilera webguneak ordenagailu lokalean garatzeko.
- PHP programazio lengoaiaren erabilera web garapenerako.
- MYSQL-ren erabilera web garapenerako.

Bestetik, egileak prestakuntza zuen erabilgarriak izan diren beste gai batzuetan:

- Git eta GitHub-en erabilera bertsio kontrolerako.
- GitHub Pages-en erabilera webgune estatikoak sortzeko.
- Java eta Eclipse tresnen erabilera.



5 Arauak eta Erreferentziak

Kapitulu honetan, proiektuan zehar erabilitako araudia, bibliografia, metodoak, tresnak, ereduak, metrikak eta prototipoak deskribatuko dira.

5.1 Aplikatutako Legedia eta Araudia

Hainbat esparrutako legedia eta araudia aplikatzen da. Alde batetik, GrALari eta dokumentazioari buruzkoak eta bestetik erlazionatuta dauden administrazio publikoaren legeak.

Gradu Amaierako Lanen inguruko bete beharreko arautegia:

- [UPV/EHUko gradu amaierako lanen araudia](#). Euskal Herriko Unibertsitatean gradu amaierako lana egin eta defendatzeari buruzko arautegia.
- [Informatika Fakultateko gradu amaierako lanen araudia](#). Informatika Fakultateko Gradu Amaierako Lanari buruzko arautegia.
- [BOE-A-2009-12977](#). Informatika Ingeniaritzako Graduko edo Ingeniaritza Teknikoko titulazioak bete beharreko konpetenzia profesionalak eta Gradu Amaierako Lanen izaera profesionala ezartzen duen Errege Dekretua.

Administrazioa Publikoak ezarritako Sektore Publikoko Kontratuenei legeak eta aurkeztutako kexak:

- [BOE-261-2007-18874](#). 30/2007 Legea, urriaren 30ekoa, Sektore Publikoko Kontratuenei buruzkoa.
- [BOE-A-2011-17887](#). 3/2011 Legegintzako Errege Dekretua, azaroaren 14koa, Sektore Publikoko Kontratuenei buruzko Legearen testu bategina onartzen duena.
- [BOE-A-2017-12902](#). 9/2017 Legea, azaroaren 8koa, Sektore Publikoko Kontratuena, Europako Parlamentuaren eta Kontseiluaren 2014ko otsailaren 26ko 2014/23/EB eta 2014/24/EB zuzentarauen transposizioa egiten duena Espainiako ordenamendu juridikora. 93. Artikulua: Kalitatea bermatzeko arauak betetzen direla egiaztatzea.
- [Recurso nº 6/2016 Resolución nº 100/2016](#). Kontratu baliabideen administrazio zentralaren erresoluzioa OESIA NETWORKS, S.L enpresaren errekursoari.

Kalitate-eredu eta giden inguruko informazio eta baliabideak:

- [CMMI-DEV, V1.3](#). Improving processes for developing better products and services.
- [CMMI 2.0](#). Capability and performance model. Berarekin batera bakiabide multzo bat dakin:
 - Eredua erraztua (erredundantziak murriztuz) eta azken joera metodologikoak gehituta
 - Ebauazio metodoa erakundeen maila jakiteko.
 - Prestakuntza ikastaroak dagozkio ziurtagiriak lortzeko.
 - Sistema eta erreminta berriak. Kalitate-eredua online atzitzeko eta ebauaziok egiteko.
 - CMMI V2.0 hartzeko edo CMMI 1.3-tik migratzeko gidak.
- [PMBOK](#). Project Management Body of Knowledge.
- [SWEBOK](#). Software Engineering Body of Knowledge.

Proiektu honen dokumentazioen antolaketarako eta proiektuaren aurkezpenerako aplikatu den araua CCII-N2016 estandarra da.

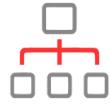
- [CCII-N2016-01](#). Ingeniaritza informatikoko proiektuen ikuskaritza edo bisa egiteko araua CCII N2016-01 estandarra da. Estandarrak dokumentuen osotasuna berrikusteko zerbitzuen prozesua deskribatzen du.

- [CCII-N2016-02](#). Estandar honek ingeniaritza informatikoko proiectuen dokumentazioaren antolaketa eta bere aurkezpena zehazten du. Memoria eta bere eranskinak estandar honen arabera antolatu dira, baita memoriarekin batera entregatu den webgunearren antolaketa.

5.2 Bibliografia

Jarraian, proiectuan zehar informazioa bilatzeko eta datuak lortzeko erabili diren erreferentzia bibliografikoak zerrendatuko dira.

- [1] J. Rojo, «ProWF proiectua: Software proiectuen elaboraziorako workflowetan oinarritutako sistemaren sorkunta eta bizi-zikloa definitzeko metodologia baten ezarpena. Gradu Amaierako Lana.,» 2020. [Online]. Available: <https://juletx.github.io/ProWF/>.
- [2] Eclipse Foundation, «OpenUP: Open Unified Process,» 2012. [Online]. Available: <https://420-gel-hy.github.io/EPF/openup/index.htm>.
- [3] Eclipse Foundation, «ABRD: Agile Business Rules Development,» 2012. [Online]. Available: <https://420-gel-hy.github.io/EPF/ARBD/index.htm>.
- [4] L. Torvalds, «Git Documentation,» [Online]. Available: <https://git-scm.com/doc>.
- [5] Microsoft, «GitHub Documentation,» [Online]. Available: <https://docs.github.com/>.
- [6] Microsoft, «GitHub Pages Documentation,» [Online]. Available: <https://docs.github.com/en/pages>.
- [7] Eclipse Foundation, «Xtext Documentation,» [Online]. Available: <https://www.eclipse.org/Xtext/documentation/index.html>.
- [8] Drupal, «Drupal Documentation,» 2021. [Online]. Available: <https://www.drupal.org/documentation>.
- [9] Apache Friends, «XAMPP Documentation,» [Online]. Available: <https://www.apachefriends.org/docs/>.
- [10] Pantheon, «Pantheon Documentation,» 2021. [Online]. Available: <https://pantheon.io/docs/>.
- [11] CCII, «Norma CCII-N2016-01: Norma de Visado de Proyectos y Actuaciones Profesionales en Ingeniería Informática,» 2016. [Online]. Available: <https://juletx.github.io/ProMeta/Proiectua/Memoriaren%20Eranskinak/A1%20-%20Sarrerako%20dokumentazioa/CCII-N2016-01.pdf>.
- [12] CCII, «Norma CCII-N2016-02: Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática,» 2016. [Online]. Available: <https://juletx.github.io/ProMeta/Proiectua/Memoriaren%20Eranskinak/A1%20-%20Sarrerako%20dokumentazioa/CCII-N2016-02.pdf>.
- [13] P. Kruchten, «RUP: Rational Unified Process,» [Online]. Available: https://www.researchgate.net/publication/220018149_The_Rational_Unified_Process--An_Introduction.



[14] J. Legarda, «BETRADOK proiektua: Betezizunen trazabilitate inpaktu-analisi automatikoa eta dokumentazio formalaren sorkuntza automatikoa modeloetan oinarritutako ekosistemetan. Gradu Amaierako Lana,,» 2019. [Online]. Available: <https://juletx.github.io/BETRADOK/>.

5.3 Metodoak

Proiektuan hainbat metodo erabili dira, garapena baldintzatu dutenak.

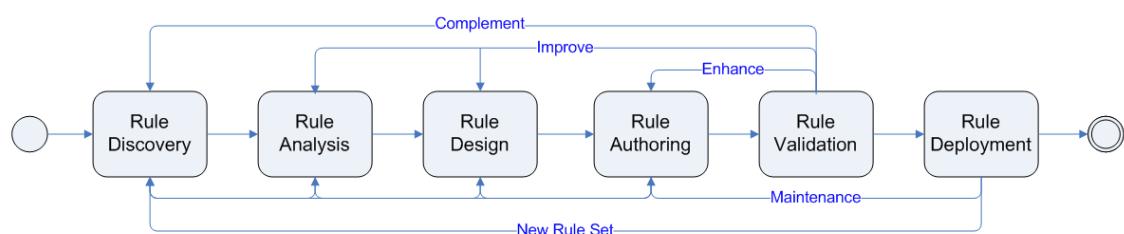
5.3.1 OpenUP

OpenUP softwareko garatzeko metodo eta prozesu bat da, teknologien sektoreko enpresa multzo batek proposatutakoa, zeintzuk 2007an Eclipse Fundazioari dohaintzan eman zioten [2]. 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.

5.3.2 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 [3]. Metodologia honek erregelen prozesua eta sistema hedatzeko beharrezkoak diren arkitektura diciplinak zehazten ditu. 5.1. Irudian metodologiaren prozesua ikus daiteke, fase eta jarduerekin.



5.1. Irudia. ABRD metodologiaren prozesuko faseak.

5.3.3 MDE

Model Driven Engineering (MDE) edo eredu bidezko ingeniaritza softwarea garatzeko metodologia da. Domeinu ereduak erabiltzen ditu, hau da, arazo zehatz bat lotutako gai guztiak eredu kontzeptualak. Hori dela eta, aplikazioen domeinu jakin baten ezagutzaren eta jardueren irudikapen abstraktuak nabamentzea du helburu, kontzeptu informatikoetan sartu gabe.

5.4 Tresnak

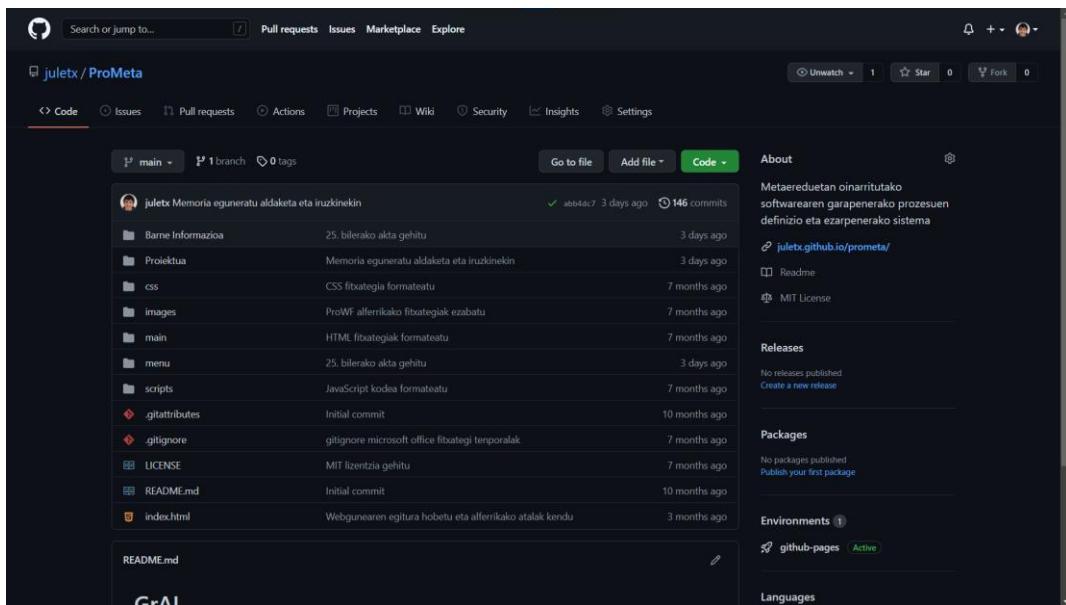
Atal honetan erabili diren tresna nagusien deskribapen labur bat egingo da.

5.4.1 Git

Linus Torvaldsek garatutako bertsio-kontrol software bat da [4]. Hain zuzen, produktu edo konfigurazio batean egin daitezkeen aldaketak kudeatzeko programa da. Proiektuaren fitxategi guztiak bertsio-kontrolerako erabili da, hainbat biltegitan banatuta.

5.4.2 GitHub

Bertsio kontrolerako web-ostatu zerbitzua da, Git erabiltzen duena [5]. Gehienbat iturburu koderako erabiltzen da. Git-en bertsio-kontrol banatu eta iturburu-kode kudeatzaile funtzionalitate guztiak eskaintzeaz gain bere ezaugarri propioak gehitzen ditu. Proiektuan Git biltegi guztiak igotzeko erabili da. Adibidez, 5.2. Irudian dokumentazio webgunearen biltegia ikus daiteke.



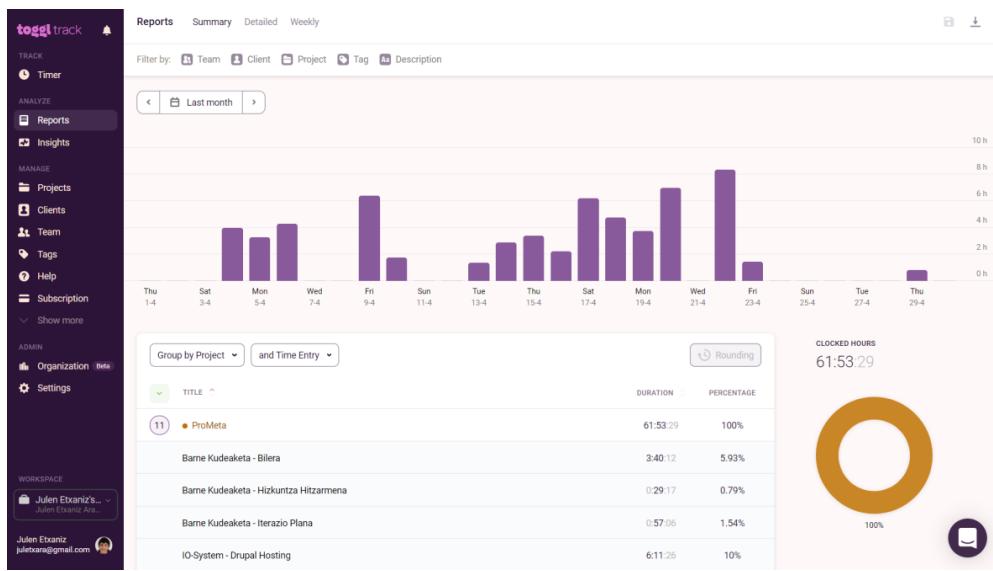
5.2. Irudia. ProMeta proiektuko webgunearen GitHub biltegia.

5.4.3 GitHub Pages

GitHub-ekin integratuta, automatikoki eraikitzen da webgunea kodea GitHub-era igotakoan [6]. Konfigurazio oso erraza, biltegitik 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.

5.4.4 Toggle Track

Proiektuko denbora neuritzeko erabili den tresna. Temporizadore baten bidez ataza bakoitzean pasatako denbora neurtu daiteke. Eskuz ere aldatu daiteke denbora ahaztu egin bazaigu. Abisatu egiten du inaktibo bagaude eta denbora neurzen ari bagara. Aspaldian denborarik ez badugu neurtu ere abisatzen du, ez ahazteko. Denbora proietutan, atazatan eta kategoriatan sailkatu daiteke. Edozein momentuko denboraren estatistikak ikusi eta deskargatu daitezke. Adibidez, 5.3. Irudian hilabete bateko egun bakoitzeko denbora ikus daiteke.



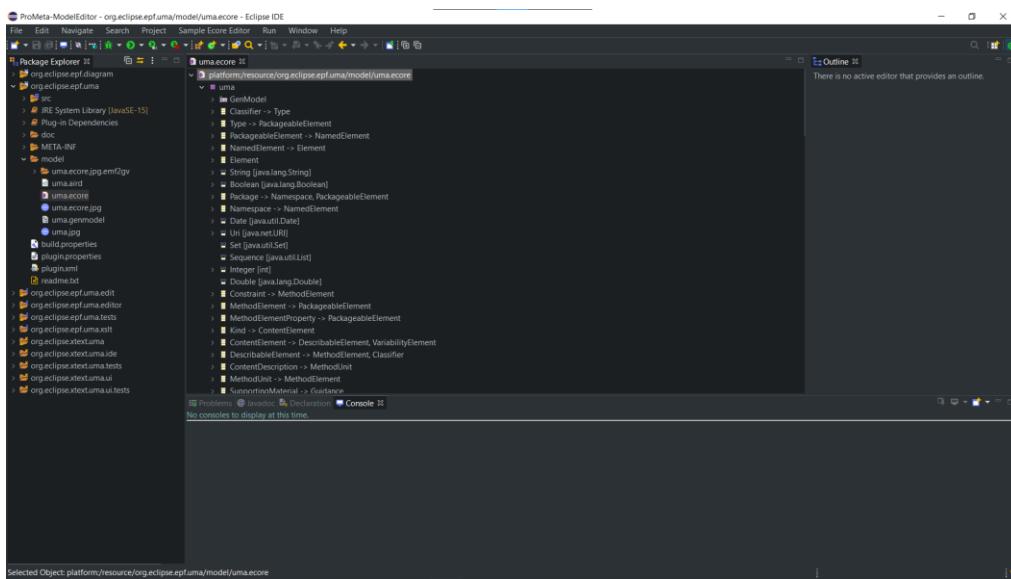
5.3. Irudia. Toggle Track tresnan hilabete bateko egun bakoitzeko denbora.

5.4.5 Java

Sun Microsystems-ek garatutako programazio lengoia eta plataforma informatikoa da. Plataforma makina birtual bat da eta Java programazio lengoia eta garapen tresnak erabiliz garatutako aplikazioak exekutatzeko gai da. Proiektuan Java 8 bertsioa erabili da EPF Composer-erako eta Java 11 eta 15 Eclipseko garapenerako. Java programazio lengoia erabili da ModelEditor azpisistemaren garapenerako.

5.4.6 Eclipse IDE

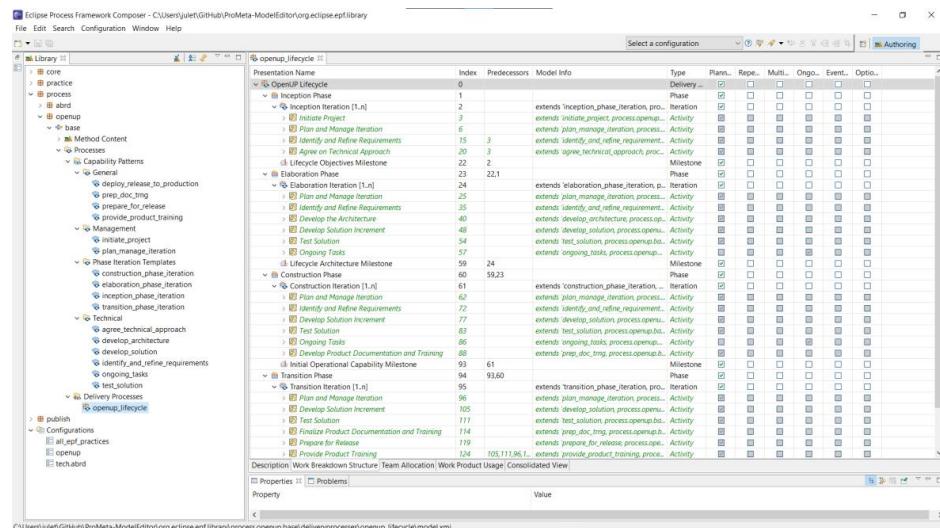
Kode irekiko software plataforma bat da. Ematen zaion erabilera nagusia Javaz programatzeko garapen ingurune integratuarena (IDE) da. Proiektuan zehar ModelEditor azpisistemaren garapenerako erabili da, 5.4. Irudian ikus daitekeen bezala.



5.4. Irudia. ModelEditor azpisistemako proiektuak Eclipse IDE editorean.

5.4.7 EPF Composer

EPF Composer tresna erabili da metodologien informazioa eskuratu eta bistaratzeko. Gainera, metodologia aldatzeko erabil daiteke eta metodologien webgunea sortzeko. 5.5. Irudian ikus daiteke EPF Composer tresnaren erabilera adibide bat.



5.5. Irudia. OpenUP bizi-zikloa EPF Composer tresnan.

5.4.8 Xtext

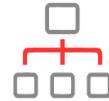
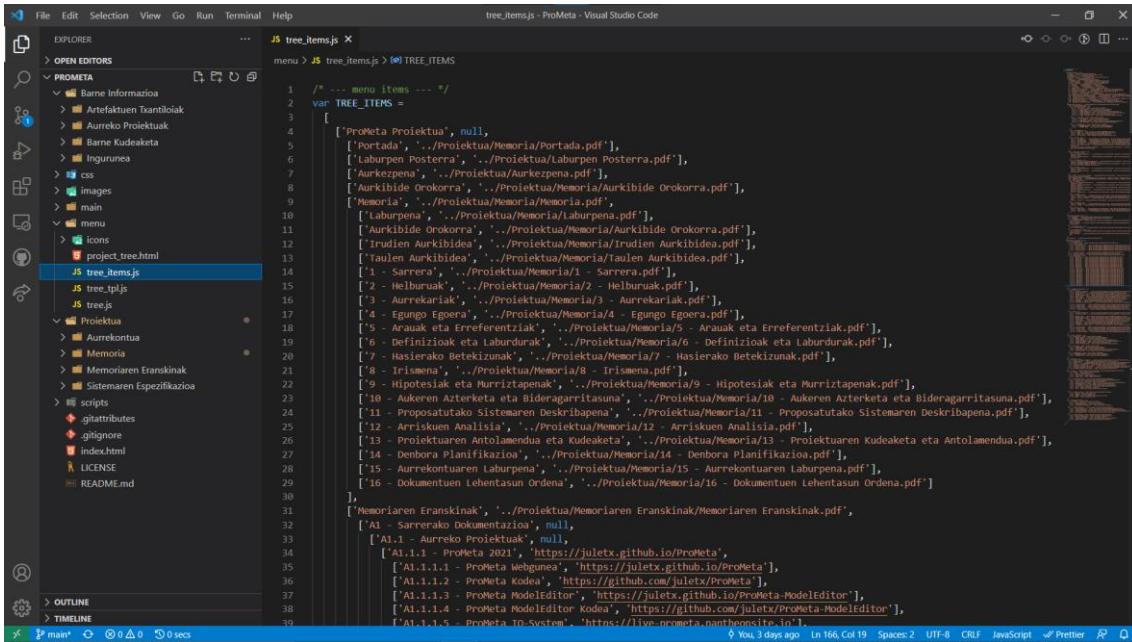
Programazio lengoaiak eta domeinu espezifikoko lengoaiak (DSL) garatzeko kode irekiko ingurunea da [7]. Xtext-ek analizatzaile bat, sintaxi zuhaitz abstraktuaren klase eredu eta Eclipse-n oinarritutako IDE pertsonalizagarria sortzen ditu. Proiektuan testu editorea sortzeko eta SQL kodea sortzeko erabili da.

5.4.9 XSLT

eXtensible Stylesheet Language Transformations (XSLT) XMLn oinarritutako lengoia da, XML fitxategiak transformatzeko balio duena. Proiektuan ModelEditor azpisisteman XMI eredu UMAra transformatzeko erabili da.

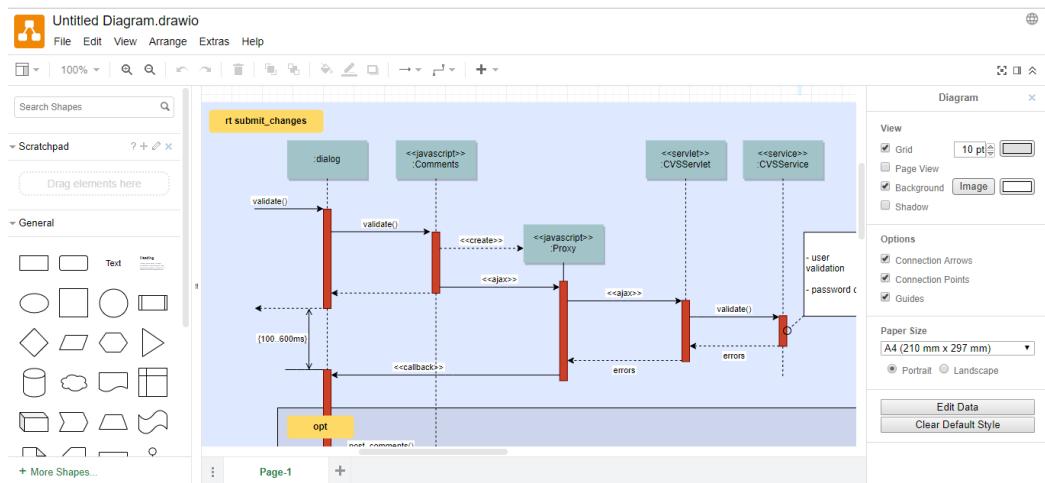
5.4.10 Visual Studio Code

Microsoftek garatutako kode editorea. Bertsio kontrola, sintaxi nabarmentzea eta kode osatze automatikoa bezalako aukerak eskaintzen ditu. Software librea eta doakoa da. Proiektuan zehar hainbat fitxategi mota editatzeko erabili da. IO-System sistemaren garapenera editore nagusia izan da. Gainera, bertsio kontrolerako ere erabili da, *commit* eta *push* guztiek bertatik eginez. 5.6. Irudian ProMeta dokumentazio webgunearen errepositorioa ikus daiteke Visual Studio Code-n.

```

File Edit Selection View Go Run Terminal Help
tree_items.js - ProMeta - Visual Studio Code
menu > JS tree_items.js > [?] TREE_ITEMS
1 /* --- menu items --- */
2 var TREE_ITEMS =
3 [
4     ['ProMeta Projekta', null,
5         ['Portada', '../Proiektua/Memoria/Portada.pdf'],
6         ['Laburpen Posterra', '../Proiektua/Laburpen Posterra.pdf'],
7         ['Aurkezpena', '../Proiektua/Aurkezpena.pdf'],
8         ['Aurkibide Orokorra', '../Proiektua/Memoria/Aurkibide Orokorra.pdf'],
9         ['Memoria', '../Proiektua/Memoria/Memoria.pdf'],
10        ['Laburpena', '../Proiektua/Memoria/Laburpena.pdf'],
11        ['Aurkibide orokorra', '../Proiektua/Memoria/Aurkibide Orokorra.pdf'],
12        ['Irudien Aurkibidea', '../Proiektua/Memoria/Irudien Aurkibidea.pdf'],
13        ['Iaulen Aurkibidea', '../Proiektua/Memoria/Iaulen Aurkibidea.pdf'],
14        ['1 - Sarreia', '../Proiektua/Memoria/1 - Sarreia.pdf'],
15        ['2 - Helburuak', '../Proiektua/Memoria/2 - Helburuak.pdf'],
16        ['3 - Aurrekarriak', '../Proiektua/Memoria/3 - Aurrekarriak.pdf'],
17        ['4 - Egungo Egoera', '../Proiektua/Memoria/4 - Egungo Egoera.pdf'],
18        ['5 - Arauak eta Erreferentziak', '../Proiektua/Memoria/5 - Arauak eta Erreferentziak.pdf'],
19        ['6 - Definizioa eta Laburdurak', '../Proiektua/Memoria/6 - Definizioa eta Laburdurak.pdf'],
20        ['7 - Hasierako Betezkizunak', '../Proiektua/Memoria/7 - Hasierako Betezkizunak.pdf'],
21        ['8 - Irismena', '../Proiektua/Memoria/8 - Irismena.pdf'],
22        ['9 - Hipotesiak eta Murriztapenak', '../Proiektua/Memoria/9 - Hipotesiak eta Murriztapenak.pdf'],
23        ['10 - Aukeren Azterketa eta Bideragarritasuna', '../Proiektua/Memoria/10 - Aukeren Azterketa eta Bideragarritasuna.pdf'],
24        ['11 - Proposatutako Sistemaren Deskribapena', '../Proiektua/Memoria/11 - Proposatutako Sistemaren Deskribapena.pdf'],
25        ['12 - Arriskuen Analisia', '../Proiektua/Memoria/12 - Arriskuen Analisia.pdf'],
26        ['13 - Proiektuaren Antolamendua eta Kudeaketa', '../Proiektua/Memoria/13 - Proiektuaren Kudeaketa eta Antolamendua.pdf'],
27        ['14 - Dendoba Planifikazioa', '../Proiektua/Memoria/14 - Dendoba Planifikazioa.pdf'],
28        ['15 - Aurrekontuaren Laburpena', '../Proiektua/Memoria/15 - Aurrekontuaren Laburpena.pdf'],
29        ['16 - Dokumentuen Lehenetasun Ordena', '../Proiektua/Memoria/16 - Dokumentuen Lehenetasun Ordena.pdf'],
30    ],
31    ['Memoriaren Eranskinak', '../Proiektua/Memoriaren Eranskinak/Memoriaren Eranskinak.pdf'],
32    ['A1 - Sarreko Dokumentazioa', null,
33        ['A1.1 - Aurreko Projektuak', null,
34            ['A1.1.1 - Prometa 2021', 'https://juletx.github.io/ProMeta'],
35            ['A1.1.1.1 - Prometa Webgunea', 'https://juletx.github.io/ProMeta'],
36            ['A1.1.1.2 - Prometa Kodea', 'https://github.com/juletx/ProMeta'],
37            ['A1.1.1.3 - Prometa ModelEditor', 'https://juletx.github.io/ProMeta-ModelEditor'],
38            ['A1.1.1.4 - Prometa ModelEditor Kodea', 'https://github.com/juletx/ProMeta-ModelEditor'],
39            ['A1.1.1.5 - Prometa To-System', 'https://live-prometa.namhoonsite.io/'],
40        ],
41    ],
42]
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
856
857
858
858
859
859
860
861
862
863
864
865
866
866
867
868
868
869
869
870
871
872
873
874
875
876
876
877
878
878
879
879
880
881
882
883
884
885
886
886
887
888
888
889
889
890
891
892
893
894
895
895
896
896
897
897
898
898
899
899
900
901
902
903
904
905
905
906
906
907
907
908
908
909
909
910
911
912
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1
```

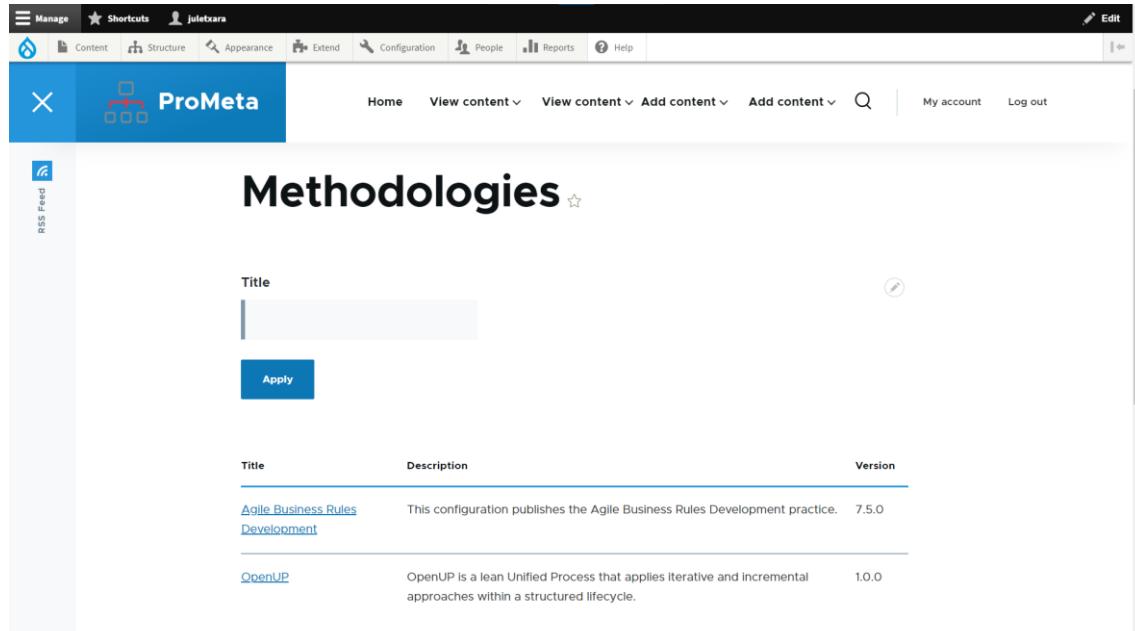


5.8. Irudia. Draw.io diagrama adibidea.

5.4.14 Drupal

Drupal edukiak kudeatzeko sistema edo CMS librea, modularra eta oso konfiguragarria da [8]. Sistema dinamikoa da, hau da, zerbitzariaren edukiak modu finkoa biltegiratu beharrean, orrien testu-edukia eta beste konfigurazio batzuk datu-base batean biltegiratzen dira eta web-ingurune bat erabiliz editatzen dira. 5.9. Irudian proiectuan sortutako Drupal webgunea ikus daiteke.

ProMeta IO-System sistemaren interfaze moduan erabili da, datuen sarrera/irteerak kudeatzeko.



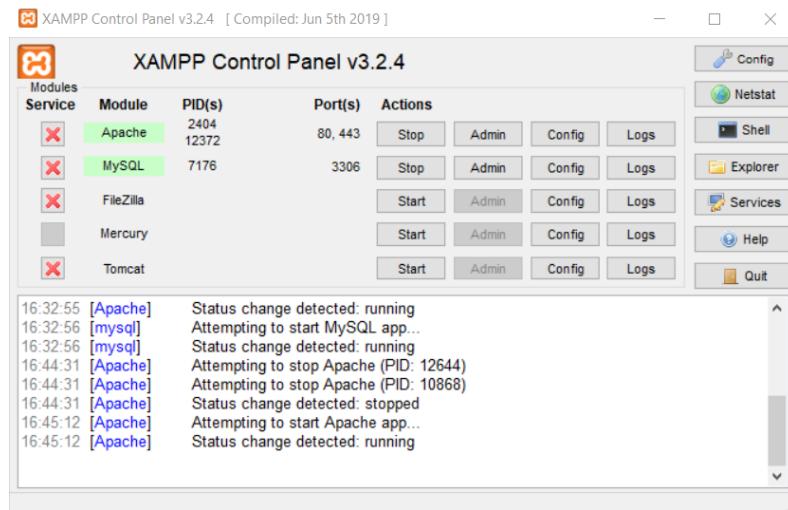
Title	Description	Version
Agile Business Rules Development	This configuration publishes the Agile Business Rules Development practice.	7.5.0
OpenUP	OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle.	1.0.0

5.9. Irudia. Proiectuan sortutako Drupal webgunea.

5.4.15 XAMPP

XAMPP software libreko pakete bat da, nagusiki MySQL datu-baseen kudeaketa sistema eta Apache web zerbitzaria integratzen duena [9]. 5.10. Irudian ikus daiteke XAMPPen erabileraren adibide bat.

Proiectuan honetan Drupal webgunea lokalean garatzeko erabili da.



5.10. Irudia. XAMPP kontrol panela Apache eta MySQL hasieratzeko.

5.4.16 Pantheon

Pantheon kode irekiko Drupal eta WordPress webguneetarako hosting plataforma da [10]. 5.11. Irudian Pantheon kontrol panela ikus daiteke.

The screenshot shows the ProMeta interface on the Pantheon platform. On the left, there's a sidebar with links for Multidev, Dev (which is selected), Test, and Live. Below that are links for Visit Development Site and Site Admin. The main area has tabs for Code (selected), Development Mode (set to Git), SFTP, and Git. A message says "There are no updates". The Commit Log section shows a recent update: "Updated drupal/core to 9.1.9 and 25 others" by "drupal/core (9.1.7) to (9.1.9)" and "drupal/core-recommended (9.1.7) to (9.1.9)".

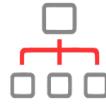
5.11. Irudia. ProMeta webguneko Pantheon kontrol panela.

5.4.17 MariaDB

MariaDB datu-baseak kudeatzeko sistema erlazional libre bat da. Oracle Corporationek MySQLren jabea zen Sun Microsystems erostean MySQLren jatorrizko garatzaileetako batzuk GNU Lizentzia Publiko Orokorra lizenziapean libre mantentzeko asmoz sortutako fork bat da. Datu-base erlazional moduan erabili da proiektuan.

5.4.18 phpMyAdmin

phpMyAdmin PHP-n idatzitako tresna da MySQL administrazioa web orrien bidez kudeatzeko. Proiektuan datu-base erlazionalak kudeatzeko erabili da, 5.12. Irudian ikus daitekeen moduan.



The screenshot shows the phpMyAdmin interface. On the left, the database structure for 'methodology' is displayed, showing various tables like activities, activity_tasks, artifacts, disciplines, domains, input_artifacts, iterations, iteration_activities, iteration_tasks, methodologies, output_artifacts, phases, phase_activities, phase_tasks, practices, processes, roles, role_artifacts, role_sets, role_tasks, tasks, task_sections, and templates. On the right, a table dump for the 'activities' table is shown, listing 45 rows. The columns include Acción, Tabla, Filas, Tipo, Cotejamiento, Tamaño, and Residuo a depurar. The table has 45 rows and a total size of 16.0 KB.

Acción	Tabla	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
Examinar Estructura Buscar Insertar Vaciar Eliminar	activities	45	InnoDB	utf8mb4_general_ci	16.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	activity_tasks	135	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	artifacts	64	InnoDB	utf8mb4_general_ci	48.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	disciplines	18	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	domains	16	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	input_artifacts	72	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	iterations	10	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	iteration_activities	24	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	iteration_tasks	47	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	methodologies	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	output_artifacts	74	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	phases	15	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	phase_activities	19	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	phase_tasks	9	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	practices	17	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	processes	4	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	roles	23	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	role_artifacts	44	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	role_sets	4	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	role_tasks	98	InnoDB	utf8mb4_general_ci	32.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	tasks	86	InnoDB	utf8mb4_general_ci	64.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	task_sections	256	InnoDB	utf8mb4_general_ci	288.0 KB	-
Examinar Estructura Buscar Insertar Vaciar Eliminar	templates	20	InnoDB	utf8mb4_general_ci	16.0 KB	-
Número de filas	1,094	InnoDB	utf8mb4_general_ci	1.0 MB	0 B	

5.12. Irudia. phpMyAdmin tresnarekin datu-baseko taulak kudeatzen.

5.4.19 PHP

PHP (PHP: Hypertext Preprocessor) interpretatutako programazio lengoia bat da, batez ere webgune dinamikoak sortzeko erabili ohi dena. Datu-base sistema ugarirekin funtzionatzeko aukera izatea eta sistema eragile gehienetarako eskuragarri izatea dira beronen abantaila nagusiak. Projektuan Drupal-erako programazio lengoia moduan erabili da.

5.5 Ereduak

Projektuan hainbat eredu erabili dira, batzuek dokumentazioarekin erlazionatuak eta beste batzuk implementazioarekin.

5.5.1 CCII-N2016-02

Arauen atalean aipatu den moduan, estandar honetan oinarrituta antolatu da memoria eta proiektuaren webgunea. CCII erakundeak sortutako araua da, ingeniaritzako informatikoko proiektuetarako dokumentazioaren egitura eta beharrezkoak diren dokumentu eta sekzioak definitzen dituena.

5.5.2 OpenUP

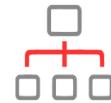
OpenUP metodologiaren eredua erabili da. Eredu horrek metodologiaren informazio guztia du. Metodologia hau jarraitzeko, bere webgunean artefaktu batzuen txantiloia daude eskuragarri. Txantiloia horiek jarraituz OpenUP metodologiaren bitarte sortutako artefaktuak idatzi dira.

5.5.3 ABRD

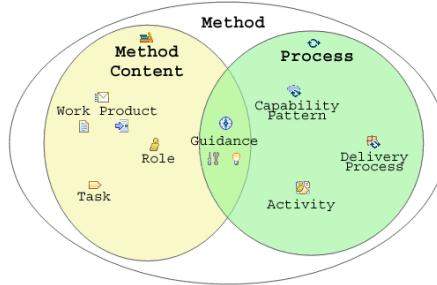
Agile Business Rules Development metodologiaren eredua ere erabili da. Eredu horrek metodologiaren informazio guztia du. Metodologia hau jarraitzeko, bere webgunean artefaktu batzuen txantiloia daude eskuragarri.

5.5.4 UMA

Unified Method Architecture metaeredua erabili da OpenUP eta ABRD ereduak definitzeko. Metaeredu honen helburua edozein metodologia modelatu ahal izatea da. 5.13. Irudian ikusten den moduan, UMA



metodoaren edukia prozesutik bereizten da. Horrela, metodoaren edukia hainbat prozesutan berrerabil daiteke.



5.13. Irudia. UMA metodologiako metodoaren eta prozesuaren kontzeptuak.

5.5.5 Ecore

EMF-ko Ecore meta-metaeredua erabili da UMA metaeredua definitzeko. Meta-metaeredu honen helburua edozein metaeredu modelatu ahal izatea da. Ecore oinarri moduan hartuta definitzen dira metaeredu guztiak, baita Ecore bera ere.

5.6 Metrikak

Proiektuaren helburuekin erlazionatutako 4 metrika nagusi daude: irismena, denbora, kostua eta kalitatea.

5.6.1 Irismena

Proiektuaren irismena neurtzeko atazak definitu dira eta bakoitzaren denbora estimazioa egin da. Kontuan hartuta proiektua eta projektuko 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.

5.6.2 Denbora

Denboraren kontrola egiteko ataza bakoitzean pasatako denbora neurtu da, Toggle Track aplikazioaren kronometroa erabiliz. Ondoren, neurtutako denbora estimatutakoarekin konparatu da, eta horren arabera erabakiak hartu dira.

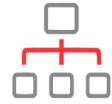
5.6.3 Kostua

Proiektuaren kostua neurtzeko aurrekontua egin da. Bertan, giza baliabideen eta erreminten kostuak zehaztu dira. Erreminta guztiak doakoak izatea espero denez, kostua kudeatzea erraza da. Giza baliabideen kostua jakiteko projektuko kide bakoitzaren ordu kopurua eta orduko kostua biderkatzen dira.

5.6.4 Kalitatea

Proiektuaren kalitatea neurtzea garrantzitsua da onargarria dela ziurtatzeko. Kudeaketa planean zehazten den moduan zaintzeko hiru aspektu kontrolatu behar dira: implementazioa, funtzionalitateak eta dokumentazioa.

- Implementazioa.** Softwarearen ekoizpenerako azpiegitura teknologikoak arkitektura konplexua izango du BPM sistema arrakastatsuak dutenen ideia, pausuz pausuko hurbilpen batean, jarraitzen delako. Sistematizazioa eta automatizazioa oinarri bezala hartuta. Ondorioz, menpekotasun teknologiko mugatua suposatu behar duten osagai asko izan behar ditu, denak irekiak azpiegituraren bilakaera errazteko.



2. **Funtzionalitateak.** Kalitatea, definizioz, bezeroen beharrak asebetetzearekin lotzen denez, azpiegitura aplikatuta kudeatzen diren proiektu errealkien dagozkien bezeroen nahiak asebetetzeko besteko emaitzak (kode eta dokumentazioa) sortu behar dituzte. Azpiegitura teknologikoa bezero errealen beharren konplexutasun gorakorrera egokitzea beharko da.
3. **Dokumentazioa.** Softwarearen ekoizpenerako azpiegitura teknologikoa proiektu askoren emaitza izango denez, ezagunak diren estandarrak erabilita eraiki behar da. Hau da, informazioaren konplexutasunera hobeto egokitzen den metodologia jarraituz adieraziko da hurrengo proiektuek beharko duten informazio guztia.

5.7 Prototipoak

Proiektuan bi prototipo nagusi garatu dira, azpisistema bakoitzari dagozkionak, ModelEditor eta IO-System.

5.7.1 ProMeta ModelEditor

Sortutako editore grafikoa eta testu editorea erabiliz prozesuaren ereduak sortzeko eta editatzeko aukera emango du. Sistema honen ardura prozesu ingeniarri rolak izango du.

5.7.2 ProMeta IO-System

CMS baten bitartez kudeatutako web-aplikazioa izango da. Helburua metodologia jarraitzen duten proiektuen informazioa gordetzea da. Rol bakoitzak metodologian dituen ataza berdinak bete beharko ditu.

6 Definizioak eta Laburdurak

Atal honetan memorian zehar agertzen diren laburduren esanahiak eta termino garrantzitsuenen definizioak daude.

6.1 Laburdurak

ABRD	Agile Business Rules Development
BETRADOK	Betekizunen Trazabilitatea eta Dokumentazio sorkuntza automatikoa
Bizagi	BPM hornitzalea
BPM	Business Process Management
BPMN	Business Process Model and Notation
CCII	Consejo de Colegios de Ingeniería Informática
CCII-N2016-01	Norma de Visado de Proyectos y Actuaciones Profesionales en Ingeniería Informática
CCII-N2016-02	Norma Técnica para la realización de la Documentación de Proyectos en Ingeniería Informática
CLIPS	C Language Integrated Production System
CMMI	Capability Maturity Model Integration
CMS	Content Management System
CSS	Cascading Style Sheet
DOT	Grafoak definitzeko lengoiaia
Drupal	CMS hornitzalea
DSL	Domain Specific Language
Ecore	Meta-metaeredua
EHSIS	Erabakiak Hartzen Laguntzeko Sistemen Sortzailea
EHU	Euskal Herriko Unibertsitatea
EMF	Eclipse Modeling Framework
EPF	Eclipse Process Framework
Freemium	Dohako oinarrizko zerbitzuak gehi ordaindu beharrekoak
GrAL	Gradu Amaierako Lana
HTML	HyperText Markup Language
IBM	International Business Machines

IDE	Integrated Development Environment
ISO	International Organization for Standardization
LDE	Lan Deskonposaketa Egitura
MDE	Model-Driven Engineering
NASA	National Aeronautics and Space Administration
OpenUP	Open Unified Process
PDF	Portable Document Format
PHP	Hypertext Preprocessor
PlantUML	Plant Unified Modeling Language
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
ProMeta	Professional Metamodel
ProWF	Professional Workflow
RUP	Rational Unified Process
SOA	Service-oriented Architecture
SPEM	Software Process Engineering Metamodel
SPICE	Software Process Improvement and Capability dEtermination
SQL	Structured Query Language
SWEBOK	Software Engineering Body of Knowledge
UMA	Unified Method Architecture
UML	Unified Modeling Language
UNE	Una Norma Española
URPS	Usability, Reliability, Performance and Supportability
Workflow	Lan-fluxua
WYSIWYG	What You See Is What You Get
XAMPP	Cross-platform Apache, MariaDB, PHP and Perl
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XPDL	XML Process Definition Language

XSLT eXtensible Stylesheet Language Transformations

6.2 Definizioak

6.2.1 Bizagi

Bi produktu osagarri dituen softwarea da, prozesuen modelatzale (Bizagi Modeler) bat eta BPMren suite ofimotiko bat (Bizagi Studio).

6.2.2 BPM

Sigla(ingelesez), Business Process Management. Enpresei prozesuak automatikoki modelatzeko, implementatzeko eta exekutatzeko aukera ematen dien software teknologia da.

6.2.3 CCII

Sigla(gaztelera), *Consejo de Colegios de Ingeniería Informática*. Estatu-mailan informatika ingeniariz guztiek errepresentatu eta bateratzen dituen antolakundea da. Iku, gainera: [CCII, webgunea](#).

6.2.4 CCII-N2016-01

CCII erakundeak sortutako araua, Ingeniaritzako informatikoko proiektuen ikuskaritza edo bisa egiteko estandarra. Estandarrak dokumentuen osotasuna berrikusteko zerbitzuen prozesua deskribatzen du [11].

6.2.5 CCII-N2016-02

CCII erakundeak sortutako araua, ingeniaritzako informatikoko proiektuetarako dokumentazioaren estruktura eta beharrezkoak diren dokumentu eta sekzioak definitzen dituena [12]. Informazio gehiago: <https://www.ccii.es/norma>.

6.2.6 CMMI

Capability Maturity Model Integration. Software-sistemak garatzeko, mantentzeko eta erabiltzeko, prozesuak hobetzeko eta evaluatzeko eredu da, CMMI Institutuak administratutakoa.

6.2.7 CMS

Sigla(ingelesez), Content Management System. Dokumentuak eta bestelako edukiak antolatu eta kudeatzeko softwarea da, normalean web-aplikazioa.

6.2.8 DOT

Testu lauan idatzitako lengoia deskriptiboa da. Grafoak deskribatzeko modu simple bat eskaintzen du, gizakiek eta konputagailuek ulertzeko modukoa.

6.2.9 Drupal

Edukiak kudeatzeko sistema edo CMS librea, modularra eta oso konfiguragarria. Iku, gainera: [Drupal, webgunea](#).

6.2.10 IBM

Sigla(ingelesez), International Business Machines. Informatikarekin lotutako tresnak, programak eta zerbitzuak ekoitztu eta merkaturatzen dituen enpresa multinazionala da.

6.2.11 ISO

International Organization for Standardization edo Estandarizaziorako Nazioarteko Erakundea nazioarteko estandarrak ezartzen dituen erakunde bat da, hainbat estatutako estandarizazio erakundeek osatua.

6.2.12 OpenUP

Open Unified Process softwarea garatzeko metodo eta prozesu bat da, Eclipse Fundazioak garatua [2]. Rational Unified Process (RUP) metodologiaren azpimultzo minimoa da. Proiectua iteraziotan banatzen du eta eta proketuaren bizi-zikloaren lau fasetan banatzen du: Hasiera, Elaborazioa, Eraikuntza eta Trantsizioa. Informazio gehiago: [OpenUp 1.0](#) edo [OpenUp 1.5](#).

6.2.13 PlantUML

Testu laua *UML* diagrametan bihurtzeko balio duen software irekia. Ikus, gainera: [PlantUML](#), [webgunea](#).

6.2.14 ProMeta

Softwarearen garapenerako prozesuen definizio eta ezarpenerako sistema metaereduetan oinarrituta. Proiektu honen izena ingeleseko hitzetatik eratutako hitz-jokoa da. “Pro” *professional* edo *process* hitzetik dator. “Meta” *metamodel* hitzaren laburdura da eta metaeredu esan nahi du.

6.2.15 ProWF

Software proiekturen elaboraziorako *workflow*etan oinarritutako sistemaren sorkunta eta bizi-ziklo definitzeko metodologia baten ezarpena. Proiektu honen aurrekariaren izena ingeleseko hitzetatik eratutako hitz-jokoa da. “Pro” *professional* hitzetik dator eta profesionala esan nahi du, “WF” *workflow* hitzetik datorren laburdura da eta lan-fluxu esan nahi du.

6.2.16 RUP

Sigla(ingelesez), Rational Unified Process. Rational Software enpresak garatutako software-prozesu bat da [13]. Objektuetara bideratutako sistemak aztertu, diseinatu, implementatu eta dokumentatzeko erabiltzen den metodologia estandarra.

6.2.17 SPICE

Software Process Improvement and Capability dEtermination. ISO/IEC 15504. Garapen-prozesuak hobetzeko, ebalutzeko, informazio-sistemak eta software-produktuak mantentzeko eredua da.

6.2.18 UML

Unified Modeling Language (Modelaketarako lengoia bateratua) sistemak zehaztu, diseinatu eta eraikitzeko lengoia da, printzipioz objektuei orientatutako programaziorako prestatuta dagoena. UML aplikazio baten garapen fase guziak modelatzeko lengoia homogeneo bat definitzen saiatzen da, bezeroaren zehaztapenetatik hasita programatzailaren diseinu xehera arte.

6.2.19 UNE

Sigla(gazteleraz), Una Norma Española. Comités Técnicos de Normalización (CTN) batzordeak sortutako arauen, arau experimentalen eta txostenen (estandarrak) multzoak dira.

6.2.20 URPS

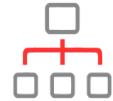
Sigla (Ingelesez). Usability, Reliability, Performance and Supportability. Softwarearen kalitate ezaugarriak dira: Erabilgarritasuna, Fidagarritasuna, Errrendimendua eta Mantenigarritasuna.

6.2.21 Workflow

Aspektu operacionalekin lan-aktibitate bat deskribatzeko egiten den irudikapena. Irudikapen horretan atazak nola egituratzen diren, zein den atazen arteko ordena eta nola sinkronizatzen diren, nolakoa den atazen informazio-fluxua eta atazen betetzearen jarraipena nola egiten den grafikoki deskribatzen da.

6.2.22 WYSIWYG

Sigla (Ingelesez), What You See Is What You Get. Testu-prozesadoreei eta beste testu-editore batzuei aplikatutako esaldi bat da, azkenengo emaitza zuzenean erakutsiz dokumentu bat idazteko aukera ematen duena.



7 Hasierako Betekizunak

Kapitulu honetan proiektuaren hasierako betekizun funtzionalak eta ez-funtzionalak azaltzen dira. Gainera, sistemak izan behar dituen kalitate ezaugarriak eta sistemaren interfazeen ezaugarriak deskribatzen dira.

7.1 Betekizun Funtzionalak

Betekizun funtzionalek sistemaren beharrak eta horiek betetzeko ezaugarriak finkatzen dituzte. 7.1. Taulan datu horiek ikus daitezke lehentasunarekin eta entrega datarekin batera. Betekizun hauek goimailan definitu dira, aurrerago zehaztuko da gehiago bakoitzaren inguruan.

Beharra	Ezaugarriak	Lehentasuna	Entrega Data
Software garapeneko prozesuaren definizioa	Software garapeneko prozesuen metaeredua definitu. Gutxienez OpenUP metodologiaren eredu definitu.	Altua	2021/06/20
Editore grafikoa eta testuala	Metaeredua erabiliz ereduak aldatzeko editoreak sortu. Editoreen arteko bateragarritasuna bermatu.	Altua	2021/06/20
Prozesuaren datu-basea	Prozesuen informazioa gordeko duen datu-basea definitu eta datuak gorde.	Altua	2021/06/20
Prozesuaren webgunea	Web interfazea garatu. Web kodea garatu. Webgunea zerbitzari batean jarri.	Altua	2021/06/20

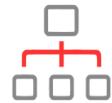
7.1. Taula. Hasierako betekizun funtzionalak.

7.2 Betekizun Ez-Funtzionalak

Betekizun ez-funtzionalek sistemaren funtzionalitateekin zuzenean erlazionatuta ez dauden betebeharrok deskribatzen dituzte. 7.2. Taulan betekizun ez-funtzionalak eta horien lehentasuna eta entrega data ikus daitezke. Orokorean, betekizun funtzionalek baina lehentasun txikiagoa dute, sistemaren funtzionamendurako ez baitira ezinbestekoak.

Betekizuna	Lehentasuna	Entrega Data
Dokumentazioa eta proiektuaren webgunea CCII estandarraren arabera.	Altua	2021/06/20
Proiektuan OpenUp metodologia jarraitzea.	Altua	2021/06/20
Garapen prozesua metodologia eta estandarrek esaten duten moduan eratzea.	Altua	2021/06/20
Erreminta, metodologia eta ezagutzaren aldetik eman daitezkeen aldaketen aurrean, soluzioa malgua eta egokigarria izatea.	Ertaina	2021/06/20
Proiektuaren garapenerako doakoak eta libreak diren tresnak erabiltzea.	Ertaina	2021/06/20
Proiektuko osagaien dokumentazioa eta eskuliburuak	Ertaina	2021/06/20

7.2. Taula. Hasierako betekizun ez-funtzionalak.



7.3 Sistemaren Ezaugarriak

Software kalitatearekin erlazionatutako URPS ezaugarriak deskribatuko dira. Garrantzitsua da argi edukitzea sistemak izan behar dituen kalitate ezaugarriak.

7.3.1 Erabilgarritasuna

Sistemak erabilgarritasun altua izango du, erabiltzeko, ikasteko eta memorizatzeko erraza izango da. Ez da ikastarorik beharko aplikazioa erabili ahal izateko, software talde bateko kideentzat intuitiboa izango denez erraz ikasiko baitute. Hala ere, sistemak erabiltzailea laguntzeko eskuliburuak izango ditu, erabiltzaileak zalantzak dituenean kontsultatu ahal izateko.

7.3.2 Fidagarritasuna

Sistemak fidagarritasun altua izango du. Honek esan nahi du ia beti eskuragarri egon behar duela, hutsegiteek eragin txikia izan behar dutela eta hauetatik azkar berreskuratuko dela. Fidagarritasuna bermatzeko, sistema monitorizatuko da arazo potentzialak azkar identifikatu eta ekiditeko.

7.3.3 Errendimendua

Sistemak errendimendu altua izango du. Honek esan nahi du erantzun denbora azkarra edukiko duela eta aldi berean hainbat konexio onartuko dituela. Hasieratze eta amaitze denbora ere azkarra izango da. Horretarako, garrantzitsua izango da zerbitzariak ahalmen nahikoa izatea.

7.3.4 Mantengarritasuna

Sistemak mantengarritasun altua izango du, hau da, instalatzeko, konfiguratzeko, eguneratzeko eta mantentzeko erraza izango dela. Erabiltzaileek web bidez erabiliko dute sistema, eta beraz ez dute ezer instalatu beharrik izango. Instalazioa, eguneratzeak eta mantenua zerbitzarian egingo dira.

7.4 Sistemaren Interfazeak

Sistemaren erabiltzaile interfazeen ezaugarri garrantzitsuak deskribatuko dira. Sistemaren interfazeak egokiak izatea ezinbestekoa da erabiltzailearen esperientzia ona izateko.

7.4.1 Itxura eta Sentsazioa

Orrialdean erabilitako kolore eta itxurari dagokionez, koloreak kontuz aukeratuko dira irakurgarritasuna bermatzeko. Gainera, kontzeptuak OpenUP prozesuan erabiltzen diren antzeko koloreekin adieraztea izango da helburua, lortura errazagoa izan dadin. Erabiliko diren menuak eta aukerak software proiektuetan aritzen direnentzat ulerterrazak eta ezagunak izan behar dira.

7.4.2 Diseinu eta Nabigazio Betekizunak

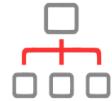
Nabigazio menuan funtzionalitate nagusiak bilduko dira, eskuragarri egon daitezten. Menuan agertzen diren aukerak erabiltzaile motaren araberakoak izango dira, eta menua atzigarria izan behar du edozein momentutan. Sistemaren atal desberdinak modu egokian antolatuta egongo dira. Sistema *responsive* izango da, pantailaren tamainaren arabera itxura automatikoki aldatuko da.

7.4.3 Trinkotasuna

Interfazeak trinkotasuna mantendu beharko du, itxuraz berdinak diren elementuek funtzionalitate antzekoa izan behar dute. Gainera, interfazeetako elementuek ohiko portaera izango dute, erabiltzaile gehienentzat beste sistemetatik ezaguna dena.

7.4.4 Erabiltzailearen Pertsonalizazio Betekizunak

Ez da aurreikusten erabiltzailean pertsonalizazio aukerarik izatea. Baliteke, interfazearen itxura edo hizkuntza aldatzeko aukeraren bat gehitzea.

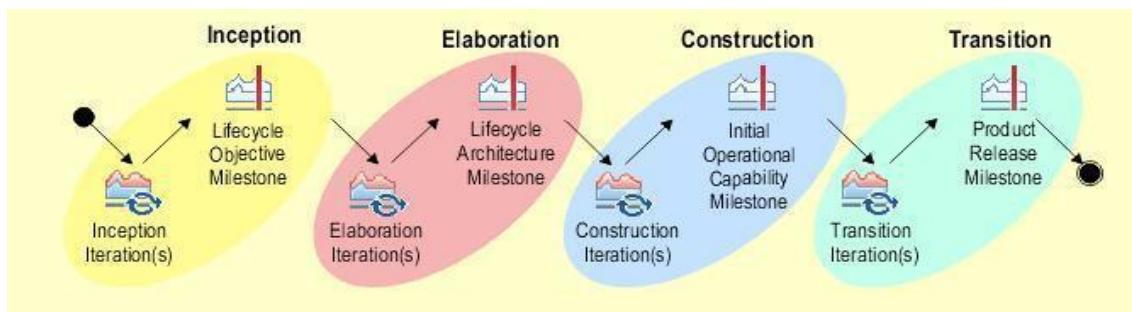


8 Irismena

Kapitulu honek proiectuaren irismena definitzea eta proiectuak sortutako entregagarriak zerrendatzea du helburu.

8.1 Bizi-zikloa

Proiectu honen irismena finkatzeko, OpenUP metodologiaren bizi-zikloa jarraitu da. 8.1. Irudian ikusten den moduan bizi-ziklo hori lau fasez osatuta dago: hasiera, elaborazioa, eraikuntza eta trantsizioa.



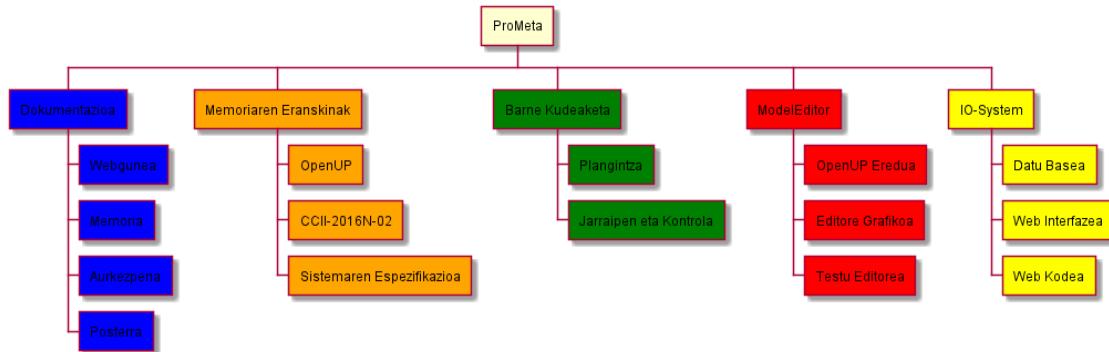
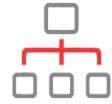
8.1. Irudia. OpenUP metodologiako bizi-zikloaren faseak.

Proiectuaren kasuan **elaborazio** faseraino iritsiko da, **produktuaren lehenengo prototipoa ateraz eta dokumentazioa sortuz**. Beste proiectu batzuen bidez hobekuntzak egingo dira. Denbora mugatuko proiectua izanik, eraikuntza eta trantsizio faseak kangoan geratu dira. Hurrengo zerrendan deskribatzen dira irismena definitzen duten ezaugarriak:

- OpenUP metodologiak eskatutako dokumentuak betetzea. Horretarako OpenUP metodologiak bere webgunean eskaintzen dituen txantiloia jarraituz.
- CCII-N2016-02 arauak eskatzen dituen dokumentuak betetzea. Ingeniaritza informatikako proiectu profesional baten dokumentazioa ere profesionala izan dadin, arau estandar bat erabiltzea oso garrantzitsua da.
- Proiectuaren webgunea osatu. Webgune honetan jarritako dokumentuak CCII-N2016-02 arauak eskatzen duen dokumentazio egitura jarraituz. Bertan, proiectuaren memoria, memoriaren eranskinak, OpenUP metodologiarekin sortutako dokumentu guztiak eta proiectuarekin zerikusia duten hainbat aspektu agertuko dira.
- Proiectuaren memoria idaztea.
- Proiectuaren defentsa prestatza. Horretarako, memorian idatzitako aspektu guztiak laburbiltzen dituen aurkezpen bat prestatuz.
- Proiectuaren posterra egitea.

8.2 LDE Diagrama

LDE diagramaren bitartez lan-pakete eta ataza nagusiak modu argian adieraz daitezke. 8.2. Irudian ikus daiteke LDE diagrama.



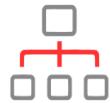
8.2. Irudia. LDE diagrama.

Hurrengo zerrendan proiektuko lan-pakete eta ataza nagusiak deskribatzen dira.

- **Dokumentazioa**
 - **Webgunea:** proiektuaren dokumentu guztiak biltzen dituen eta CCII-2016N-02 araua betetzen duen webgunea.
 - **Memoria:** proiektuaren memoriaren txantiloia prestatu eta memoria idatzi.
 - **Aurkezpena:** proiektuaren aurkezpena prestatu.
 - **Posterra:** proiektuaren posterra egin.
- **Memoriaren Eranskinak**
 - **OpenUP:** metodologiak zehazten dituen artefaktuak, hurrengo atalean zerrendatzen direnak.
 - **CCII-2016N-02:** arauak zehazten dituen memoriaren eranskinak.
 - **Sistemaren Espezifikazioa:** OpenUP metodologiak zehazten dituen artefaktuak.
- **Barne Kudeaketa**
 - **Plangintza:** OpenUP metodologiaren plangintza dokumentuak.
 - **Jarraipen eta Kontrola:** OpenUP metodologiaren jarraipen eta kontroleko artefaktuak.
- **ModelEditor**
 - **OpenUP Eredua:** OpenUP metodologiaren informazio guztia duen eredu.
 - **Editore Grafikoa:** metodologien ereduak grafikoki aldatzeko editorea.
 - **Testu Editorea:** metodologien ereduak testu bidez aldatzeko editorea.
- **IO-System**
 - **Datu Basea:** metodologiaren informazioa gordetzen duen datu-basea sortu eredutik.
 - **Web Interfazea:** webgunearen interfazea aldatu.
 - **Web Kodea:** webgunearen kodea programatu.

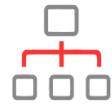
8.3 Artefaktuak

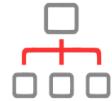
OpenUP metodologiari dagokionez, 8.1. Taulan ikusten dira bete behar diren dokumentuak, domeinuaren arabera sailkatuta. Esan bezala, hauek hasiera, elaborazio eta eraikuntza faseei dagozkien dokumentuak dira. Hala ere, baliteke denbora faltagatik dokumentu gutxi batzuk ez betetzea.



Domeinua	Artefaktua
Arkitektura	Arkitektura Koadernoa
Hedapena	Produktuaren Dokumentazioa Laguntza Dokumentazioa Erabiltzaile Dokumentazioa Trebatzeko Materialak
Garapena	Implementazioa Eraikuntza Diseinua
Ingurunea	Garapen Kasua Tresnak
Projektu Kudeaketa	Iterazio Plana Projetku Plana Lan-atazen zerrenda Arriskuen zerrenda
Betekizunak	Glosarioa Ikuspegia Betebeharren Espezifikazioa Erabilpen Kasuak Erabilpen Kasuen Eredua
Proba	Proba Kasuak Proba Log-ak Proba Script-ak

8.1. Taula. OpenUP metodologiako bete diren artefaktuak domeinutan sailkatuta.





9 Hipotesiak eta Murritzapenak

Kapitulu honetan, proiektuaren hasierako hipotesiak eta proiektuaren garapenerako ezarritako murritzapenak deskribatuko dira.

9.1 Hipotesiak

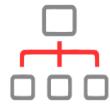
Proiektuarekin lanean hasteko hipotesi batzuk ateratzen ditugu aurreko lan eta datuetatik. Hurrengoak dira proiektuaren hasierako hipotesiak:

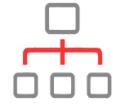
- Metaerduen erabilerak malgutasuna ematen du proiektuaren hurbilpena aldatzeko edo etorkizunean beste hurbilpen batzuk probatzeko.
- Definitutako garapen prozesuak software proiektuen elaborazioa gidatu eta kontrolatuko du. Prozesua aldatzeak sistemaren portaera eta datu-basea automatikoki aldatzea ekarriko du.
- CMS baten erabilera datuen sarrera/irteerarako irtenbide egokiena da. Webgune bat sortzeko aukera ematen duen tresna erabilerraza izateaz gain, ez da baliabide tekniko aurreratuetara etengabe jo behar. Kudeaketa, administrazioa eta mantentze-lanak egiteko laguntza ematen du kanpoko baliabiderik erabili gabe.
- Datu-base erlazionalak prozesu baten ezagutza gordetzeko modurik egokiena da, datuen independentzia, emaitzen koherenzia eta datu-basearen produktibilitatea handitzea lortuz.
- Sistema iteratiboki hobetzen joango denez, estandarretan oinarritutako garapenak bere mantenua eta hedapena errazten ditu.

9.2 Murritzapenak

Proiektu informatikoen bezeroen eskakizunen ondorioz, neurri batean mugatu egin dira proiektuarekin lotutako elementu batzuk egiteko kontuan hartu beharreko aukerak. Hurrengoak dira proiektuaren hasierako murritzapenak:

- Kostu ekonomikoa ez da batere aldatu hasierako planteamendutik. Izen ere, proiektuan zehar ez da kostu gehigarrik sortu behar, erabilitako teknologia guztiak doakoak direlako.
- Denborari dagokionez, ekainaren 20rako proiektua bukatzeko murritzapena bete behar da.
- Kalitateari dagokionez, proiektuak denbora eta kontu murritzapenen barruan kalitate onargarria izan behar du, hau da, bezeroari aurkezteko modukoa.
- Betekizunen ingeniaritza eta bizi-zikloa definitzen duen metodologia bat jarraitzea proiektuaren elaborazio eta garapenerako: OpenUP. Bezeroak egiten duen exijentzia da metodologia bat erabiltzea.
- Proiektuaren dokumentuen antolaketarako CCII-2016N-02 estandarra erabiltzea.
- Sortuko den software proiektuen elaboraziorako sistema web bidez atzigarria izan behar du.
- Beste proiektu batzuen oinarria izango denez, informazio transmisio eraginkorra ahalbideratzen duen dokumentazioa eraman behar du.





10 Aukeren Azterketa eta Egingarritasuna

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

10.1 Arkitektura

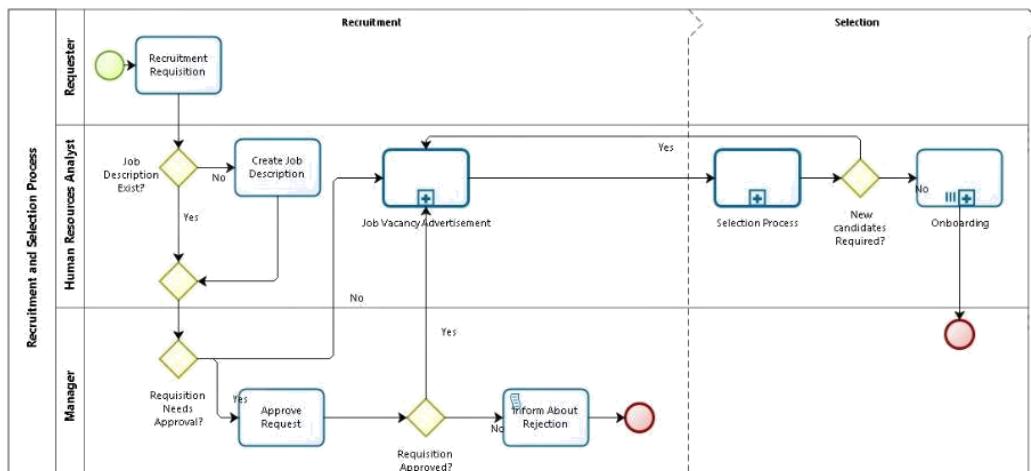
Soluzioaren arkitekturari dagokionez, ProWF proiekutan bi aukera aztertu ziren: *Bizagi* bezalako softwarearen bitartez prozesuetan (*BPM*) oinarritutako web-aplikazioa eraikitzea edo arkitektura propioa sortzea. Proiektu honetan aukera berri bat gehitu da.

10.1.1 Bizagi

Bizagi Softwarearen Kalitatea irakasgaian erabilitako softwarea da. Bertan, *Bizagi Modeler* editorearen bitartez prozesuak sortu eta ondoren, *Bizagi Studio* tresnan negozio-erregelak adieraziz, datu-basea konfiguratz, 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:

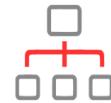
- Lan-fluxua definitzeko *Bizagi Modeler* editorea erabili daiteke, bizi-zikloa definitzen duten metodologiek faseak eta rolak erabiltzen dituztelako, 10.1. Irudian ikus daitekeen bezala. Beraz, *Bizagi Modeler* erabiliz, bere lengoiaaren (*BPMN*) arauak errespetatuz, workflow azpi-lengoia bat sortu izango litzateke.
- Soluzioaren sarrera/irteerak kudeatzeko *Bizagi Studio*ren 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 ofimatkoa da, enpresen erabilpenerako paketeak eskaintzen ditu bere produktua erosiz. Hori dela eta, *Bizagirekin* arkitekturaren kostu ekonomikoa handia izango litzateke.

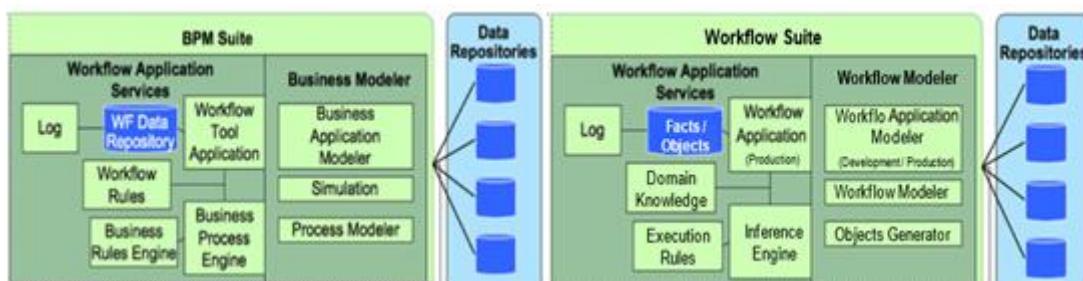
10.1.2 ProWF

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

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

- *Bizagiren* desabantailak oso kaltegarriak ziran. Honela, menpekotasun teknologikoa saihestu eta kostu ekonomikoa asko gutxitzen da.
- *Workflow-lengoia* guztiz propioa sortea komenigarria zen, lengoia 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 eraikitza 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 proiectuan sistemaren abstrakzio-maila igotzea erabaki da. Metodologien informazioa workflow lengoian edo BPMN-en esku definitu ordez, metaeredu eta ereduen bidez gordetzen da. Horrela, ereduaren informazioa berrerabil daiteke eta beste ereduetara eraldaketa egin. Eeduaren 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, BPMN-ren eta workflow-lengoaiaren metaereduak izanik, eraldaketa egitea ez litzateke horrenbeste lan izango. Gainera, eredutik datu-baserako SQL kidea edo webgunerako HTML kidea sortzeko aukera ere egongo litzateke.

Proiektu honetan saiakera bat egin da BPM eta workflow-lengoaiarik erabili gabe. Izan ere, eredutik sortutako datu-basea eta Drupal-eko moduluak bakarrik erabili dira.

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 nabarmenzen diren puntuekin.

Ezaugarria	<i>Wordpress</i>	<i>Joomla</i>	<i>Drupal</i>	Oharrak
Kode irekia	BAI	BAI	BAI	-
Dokumentazio simple eta ondo egituratuta	BAI	BAI	BAI	-
Komunitate aktiboa eta foroak	BAI	EZ	BAI	Hemen Wordpress da nagusiena.
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 gunearren itxura aldatzen jakidea 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 arrazoengatik:

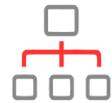
- Drupalen erraza da edukia gehitzea/sortzea. Eduki pertsonalizatu motak malguak dira eta aukera asko eskaintzen dituzte.
- Guneari gehitzeko hainbat modulu eskuragarri daude bere webgunean eta proiektu honetarako oso erabilgarriak diren moduluak aurkitu dira.
- Erabiltzaileak administratzea erraza da, 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 erakundeen txostenetan, CMS atalean, liderra den Acquia enpresak Drupal erabiltzen du oinarri bezala.

ProMeta proiektuak CMSari dagokionez antzeko helburuak dituenez, 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 exekutatzen duten inguruneetarako, GPL lizenziapean eta software librea da.



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

Arrakastaren arrazoieta bat garatzaile eta diseinatzaileen komunitate izugarria da, bere muinean programatzeaz edo komunitatearentzako pluginak eta txantiloia 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 lizenziarekin, 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 koherenzia.

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

10.2.3 Joomla

Webgune dinamiko 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) lizenziapean argitaratua.

Joomla-k bere funtzionamendurako datu-base kudeatzalearekin 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*

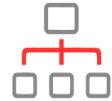
Proiektuaren webgunea eta sortutako produktuaren webgunea interneten publikoki eskuragarri egon daitezen hosting zerbitzu bat erabili behar da. Hauek batera publikatu daitezke edo aparteko webgune moduan. Doako aukerak bakarrik aztertu dira.

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 eraikitzeko 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 simpleagoa 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 simple bat eraikitzea bakarrik interesatzen zaigunez, aurreko aukerekin nahikoa daukagu.

10.4 Drupal Hostinga

10.4.1 000webhost

Ez dauka GitHubekin integrazioak 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 <https://devcenter.heroku.com/articles/github-integration>. Heroku-ren 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 simplean instalatzeko ez dago prestatuta.

10.4.3 Acquia

Ez dauka doako planik <https://www.acquia.com/choosing-right-acquia-cloud-platform>. Drupal modu simplean instalatzeko prestatuta dago.

10.4.4 Pantheon

Doako plana dauka <https://pantheon.io/plans/pricing>. GitHub-ekin ondo integra daiteke, nahiz eta ez den horren simplea <https://pantheon.io/docs/guides/collaborative-development>. Drupal modu simplean instalatzeko prestatuta dago. Drupal-erako honek ematen du aukera onena.

10.4.5 Platform.sh

Ez dauka doako planik <https://platform.sh/pricing/>. Drupal modu simplean instalatzeko prestatuta dago.

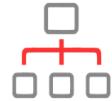
10.5 Datu-base Kudeaketa Sistema

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. Heroku erabiltzen badugu oso mugatuta dago, 5MB bakarrik eskaintzen dira doan <https://elements.heroku.com/addons/cleardb>.

10.5.2 PostgreSQL

Drupal-ekin bateragarria da, ematen dituen aukeretako bat da. Xampp-ek defektuz ez du eskaintzen, baina gehigarri moduan aktibatzeko aukera ematen du. Xampp-eko phpPgAdmin gehigarriaren bidez kontrola daiteke <https://beecreativos.com/postgresql-en-servidor-local-con-wamp-y-xampp/>. Heroku erabiltzen bada aukera egokia da, 1GB eskaintzen ditu <https://elements.heroku.com/addons/heroku-postgresql>. Webgunea Heroku-n eduki nahi badugu hau da aukera onena.



10.6 Datu-basearen Sorrera

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. Bainan, datu-basea oso handia izango litzateke eta beharrezkoa baina elementu gehiago izango lituzke. Gainera, Teneo zaharkitua dago eta Eclipseren eta EMF-ren bertsio berriein ez da bateragarria.

10.6.3 Xtext

Aukera onena Xtext eta Xtend erabiliz ereduko datuekin SQL INSERT-ak sortzea da. Honek lan dezentz euskatuko luke. Bainan, 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 erabilitako 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 dudan softwarea instalatuta. Makina birtualean softwarea instala daiteke baina errazagoa da nire konputagailuan falta den softwarea instatzea. 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

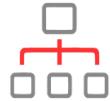
Bertsio kontrola oso garrantzitsua da, batik bat horrelako proiektuetan.

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

10.9.1 SPEM

Software Process Engineering Metamodel. SPEM metaeredua *cmoF* formatuan dago.

10.9.2 UMA

Unified Method Architecture. UMA metaeredua *ecore* formatuan dago.

10.10 Metodología

10.10.1 OpenUP

OpenUP softwarea garatzeko metodo eta prozesu bat da, teknologien sektoreko empresa 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 proietuko 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, simpleagoa da OpenUP metodologia ezartzea txikiagoa delako. Gainera, OpenUP metodologiarekin lan egiteko Eclipseren tresnak erabil daitezke. Tresna horiek libreak dira, eta egileak esperientzia du horiek erabiltzen.

10.10.3 ABRD

Agile Business Rules Development metodologiaren eredu ere erabili da. Horrela, bi metodologia erabiliz ziurtatzen da sistemaren egitura egokia dela.

10.11 Metodología Editorea

10.11.1 EPF Composer

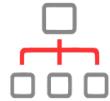
Metodologia definitzeko aukera argiena da, horretarako egindako tresna baita. Jadanik hainbat metodologia definituta daude tresna honekin: OpenUp, ABRD, Scrum... Metodologia horietako bat erabili nahi badugu ez dauagu ezer diseinatu beharrik. Horiek oinarri bezala hartuta ere errazagoa da beste metodologia bat definitzea. Metodologiaren webgunea sortzeko aukera ematen du, dokumentazio moduan erabili ahal izateko. Nahiko zaharkitua dago, Java 8 32 bitemko bertsioa eskatzen du.

10.11.2 Rational Method Composer

EPF Composer-en antzoko 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 simpleagoa den editorea edukitzea ondo etorriko litzateke. Ikono berdinak partekatuko lituzke, zuhaitz egitura simpleagoa edukiko luke.



10.11.4 Testu Editorea

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

10.12 Prozesua Bistaratu

10.12.1 DOT

ProWF proiektuan prozesua definitzeko eta bistaratzeko DOT lengoia erabili da. Beraz, aukera ona izan daiteke prozesua bistaratzeko. Hori bai, honek lan dezente gehituko luke. Izan ere, eredu DOT lengoaiara pasatzea eskatuko luke metaereduak erabiliz.

10.12.2 XPDL

XPDL ere aukera egokia izan daiteke estandarra delako. Baino, oraingoz DOT aukera hobea izango litzeteke, simpleagoa delako.

10.12.3 Webgunea

Prozesua bistaratzeko 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 definitza komenigarria da.

10.13 Denboraren Kontrola

10.13.1 Toggl Track

Temporizadore baten bidez ataza bakoitzean pasatako denbora neurtu daiteke. Eskuz ere aldatu daiteke denbora ahaztu egin bazaigu. Abisatu egiten du inaktibo bagaude eta denbora neurten 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 guztiarako aplikazioak daude, mugikorrerako ordenagailurako, nabigatzailerako, etab.

10.13.2 Clockify

Toggl Track-en antzeko funtzionamendua dauka. Doako planean aukera batzuk gutxiago eskaintzen ditu.

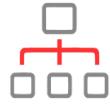
10.13.3 WakaTime

Plugin moduan instalatu daiteke editore askotan. Gure kasuan VSCode eta Eclipse editoreetan instalatu da, horiek izan baitira erabili diren editoreak. Denbora automatikoki neurten du eta metrikak sortzen ditu. Adibidez, programazio-lengoia 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

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.

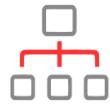


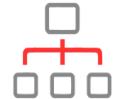
10.14.2 Draw.io

Hainbat motatako diagramak sortzeko aukera ematen duen softwarea da. Eskuz sortzen dira diagramak eta bi aukera daude aplikazioa erabiltzeko: online edo offline. Proiektaun zehar aplikazio hau erabili izan da dokumentazioan agertzen diren diagrama batzuk egiteko.

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 guzietarako.





11 Proposatutako Sistemaren Deskribapena

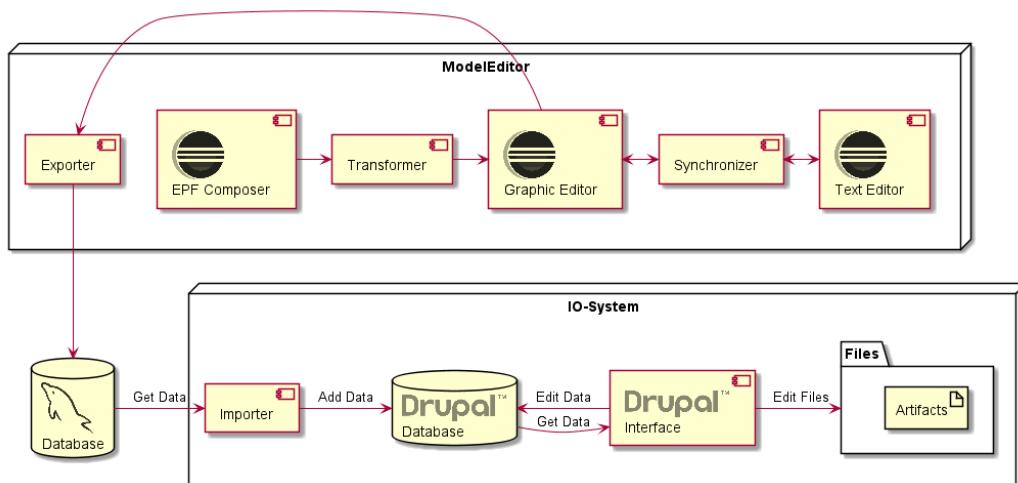
Kapitulu honetan planteatutako arazoa konpontzeko proposatzen den sistema, bere osagaiaak eta bere ezaugarriak deskribatzen dira.

11.1 Azpisistemak

ProWF sistema bi azpisistema ezberdinetan bananduta egongo da: **ModelEditor** eta **IO-System**.

- **ModelEditor:** Sortutako editore grafikoa eta testu editorea erabiliz prozesuaren eredu editatzeko aukera emango du. Sistema honen ardura prozesu ingeniarri rolak izango du.
- **IO-System:** CMS baten bitartez kudeatutako web-aplikazioa izango da. Helburua metodologia jarraitzen duten proiektuen informazioa gordetzea da. Rol bakoitzak metodologian dituen ataza berdinak bete beharko ditu.

Idea orokor bat egiteko, 11.1. Irudian sistema osoaren arkitektura ikus daiteke, azpisisteman banatuta.



11.1. Irudia. ModelEditor eta IO-System azpisistemen arkitektura.

Ondoren, azpisistema bakoitzaren hainbat aspektu azalduko ditugu: arkitektura, erabilpen kasuak, diseinua eta dokumentazioa.

Bukatzeko, sistemaren proba eta hedapena nola egingo diren azalduko da eta etorkizunerako hobekuntza posibleak aurkeztuko dira. Horrela, proiektu honi jarraipena ematea erraztuko da.

11.2 ModelEditor

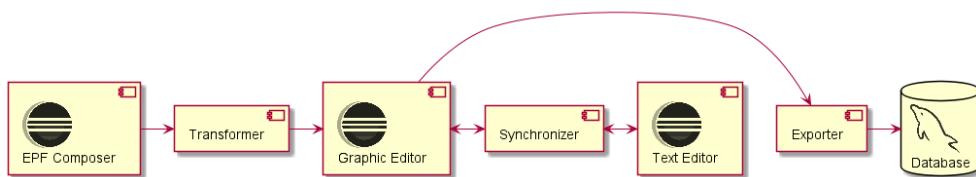
Sortutako editore grafikoa eta testu editorea erabiliz prozesuaren eredu editatzeko aukera emango du. Sistema honen ardura prozesu ingeniarri rolak izango du.

11.2.1 Arkitektura

ModelEditor azpisistemak honako osagaia izango ditu. 11.2. Irudian ikus daitezke osagaia hauen arteko erlazioak.

- **EPF Composer:** Metodologiak modu grafikoan editatzeko aukera ematen du. Metodologiaren webgunea sortu daiteke bertatik.

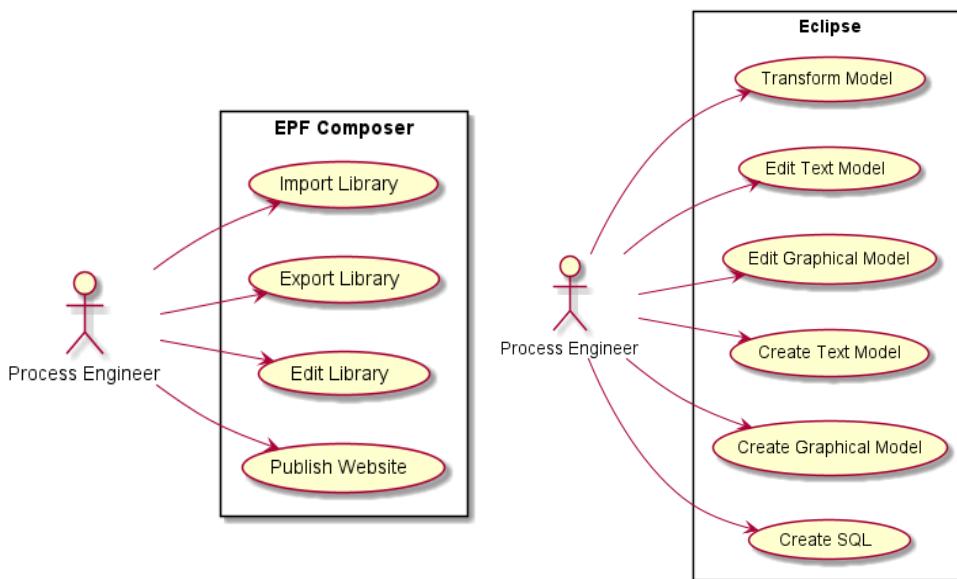
- **Formatu Aldatzailea:** Metodologiaren formatua aldatzen du XMLtik UM Ara editore grafikoak erabili ahal izan dezan.
- **Editore Grafikoa:** Metodologiak modu grafikoan editatzeko aukera ematen du, metodologiaren jatorrizko ikonoak erabiliz.
- **Testu Editorea:** Metodologiak testu bidez editatzeko aukera ematen du, metaeredutik sortutako gramatika erabiliz.
- **Editore Sinkronizatzalea:** Metodologiaren informazioa editore batekin aldatutakoan bestea ere aldatzeaz arduratzen da.
- **Kode Sortzailea:** Metodologien informazioa datu-basean gorde ahal izateko SQL kodea sortzen du.



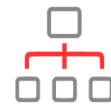
11.2. Irudia. ModelEditor azpisistemaren arkitektura.

11.2.2 Erabilpen Kasuak

Sistemaren betekizunak zehazteko beharrezko da erabilpen kasuak eta aktoreak zein diren jakitea. Azpisistemaren erabilpen kasuak bi multzotan bana daitezke: EPF Composer-enak eta Eclipse-renak. EPF Composer-etik metodologien liburutegiak importatu, esportatu eta aldatzeko aukera daukagu. Gainera, metodologiaren webgunea ere sor daiteke dokumentazio moduan erabiltzeko. Eclipsen ereduarekin zerikusia duten erabilpen kasuak dauzkagu: eredua eraldatu, sortu eta editatu. Horrez gain, SQL kodea sortzeko aukera ere badago. Erabilpen kasu hauek guztiak prozesu ingeniarik edo antzeko rolak duen pertsonak egin beharko ditu. Hala ere, metodologia bat adosterakoan erabiltzaile mota askok hartu beharko dute parte, guztiengandik beharretara egokitu behar baita. 11.3. Irudian ModelEditor azpisistemako erabilpen kasuen eredu ikus daiteke.

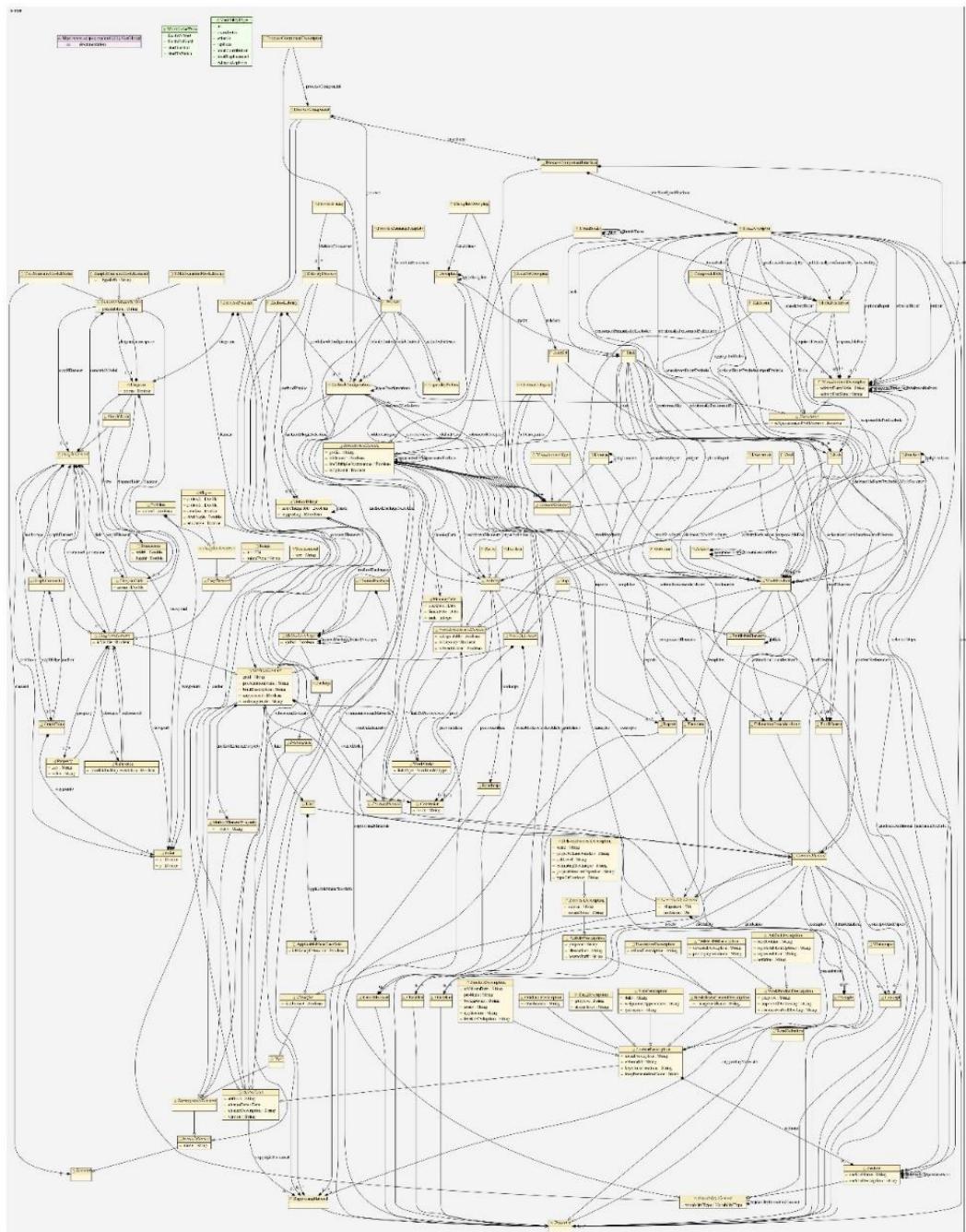


11.3. Irudia. ModelEditor azpisistemako erabilpen kasuen eredu.



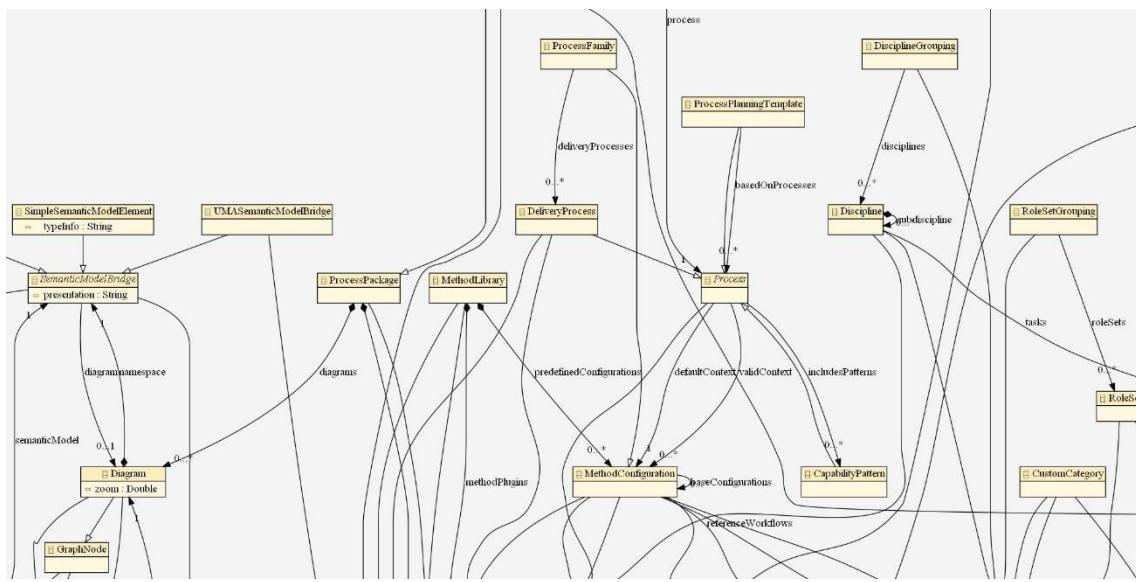
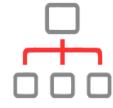
11.2.3 Diseinua

Metodologientzako metaeredu moduan UMA metaeredua erabiltzea erabaki da, SPEM estandarraren bertsio hobetua. Metaeredu hau oso konplexua da, edozein metodologia definitzeko beharrezko klase eta atributuak baititu. Metaereduaren tamainaren ideia bat egiteko 11.4. Irudia ikus daiteke.



11.4. Irudia. UMA metaeredua graphviz formatuan.

Diagrama handiegia denez, ez dira klaseen izenak ikusten. Horregatik, klaseak eta atributuak nolakoak diren ikusteko, 11.5. Irudian zati bat gertuagotik agertzen da. Bertan, ezagunak diren kontzeptu batzuk aurki ditzakegu: metodologia, prozesua, diagrama, disziplina, rola, etab. Klaseen arteko hainbat erlazio mota ere ikus daitezke.



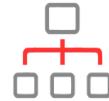
11.5. Irudia. UMA metaereduaren zati bat graphviz formatuan.

11.2.4 Dokumentazioa

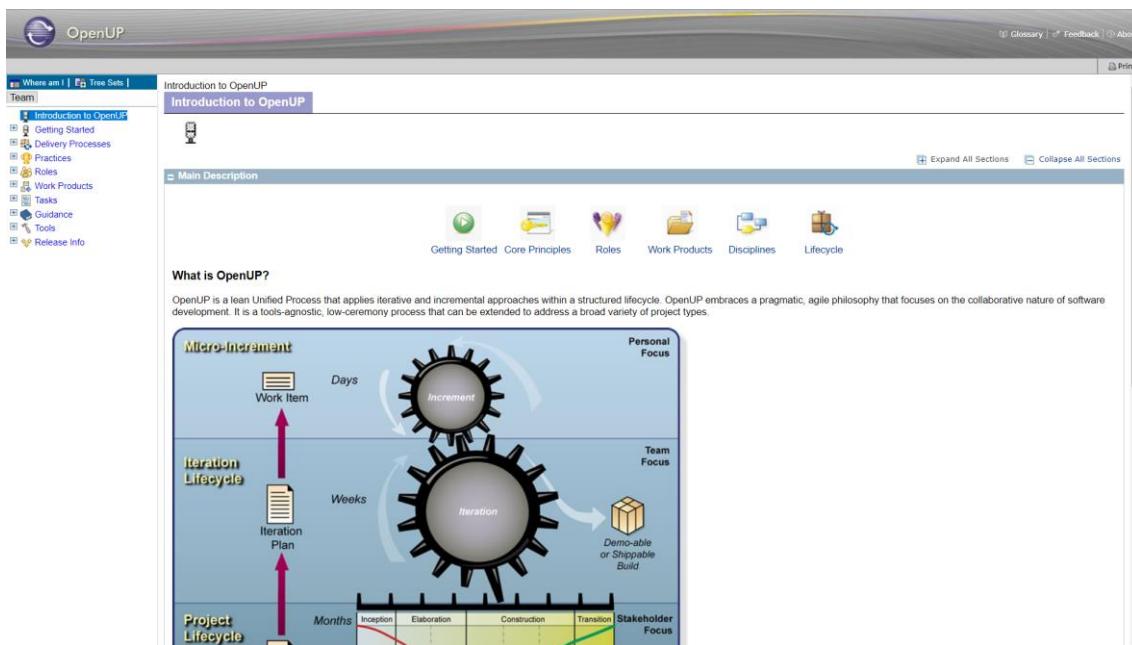
Garapenerako Eclipse-ren tresnak erabili dira orokorrean. EPF Composer eta Eclipse IDE izan dira tresna nagusiak baina Visual Studio Code ere erabili da editore moduan. Programazio lengoia nagusiak Java eta Xtend izan dira.

Hasteko, metodologia aukeratu behar dugu eta EPF Composer tresnan kargatu. Metodologiak XMI formatuan egon behar du eta UMA metaereduarekin bat eterri behar du. Metodologia propioa ere definidaiteke tresna horrekin existitzen den metodologia baten aldaketak eginez. Proiektu honetarako EPF Practices liburutegi osoa erabiltzea erabaki da. Bertan OpenUP eta ABRD metodologiak daude, 11.6. Irudian ikus datekeen moduan. Metodologia gehiago ere deskargatu daitezke Eclipseren webgunetik, Scrum eta XP adibidez. Etorkizunean RUP metodologia definitzeko aukera ere badago.

11.6. Irudia. OpenUP bizi-zikloa EPF Composer tresnan.

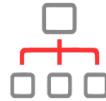


Behin metodologia kargatuta dugula hainbat aukera ditugu. Esan bezala, EPF Composer tresna bera erabil daiteke metodologian aldaketak egin nahi baditugu. Beste aukera garrantzitsu bat metodologiaren webgunea sortzea da, dokumentazio moduan oso erabilgarria dena. Adibidez, OpenUP metodologiaren webgunea ikus daiteke 11.7. Irudian.



11.7. Irudia. OpenUP metodologiaren webgunea.

Metodologia nahi dugun moduan dagoenean hurrengo pausura pasa gaitezke. Aurrerago ere edukiko dugu aldaketak egiteko aukera beharrezkoa bada. Arkitekturan ikus daitekeen moduan, hurrengo pausua XSLT erabiliz ereduan hainbat eraldaketa egitea da. Izan ere, dagoen bezala uzten badugu, eredua ez dator bat UMA metaereduarekin eta errore pila bat ematen ditu. 11.8. Irudian agertzen den XSLT fitxategiak 600 lerro baino gehiago dituen arren, nahikoa da Java fitxategi bat exekutatzeari eta segundu gutxi batzutan eraldatutako .uma fitxategiak lortzen ditugu.



```

</> Xmi2Uma.xslt ×

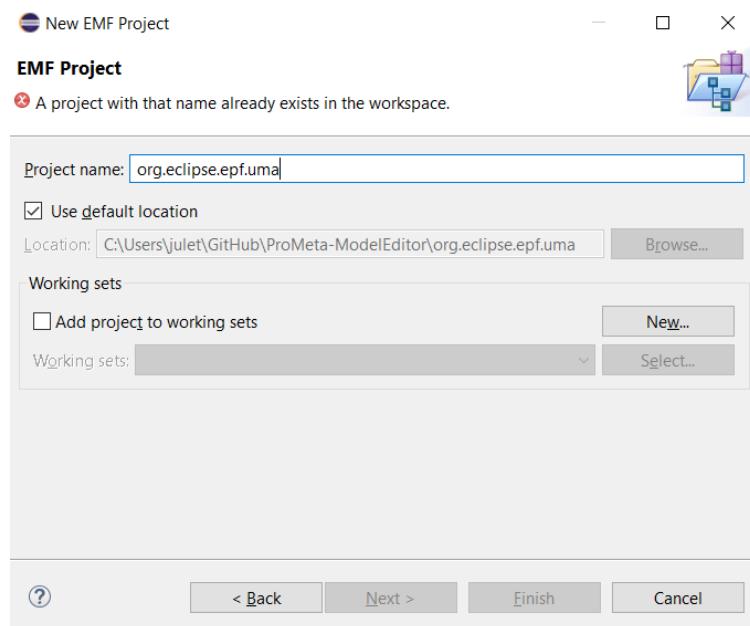
org.eclipse.epf.uma.xslt > xslt > </> Xmi2Uma.xslt

You, 5 months ago | 1 author (You)
1  <?xml version="1.0" encoding="UTF-8"?> You, 7 months ago • Transform xmi to uma and fix some
2  <xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xmi="http://
   www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:org.eclipse.epf.
   uma="http://www.eclipse.org/epf/uma/1.0.6/uma.ecore" xmlns:org.eclipse.epf.uma.
   resourcemanager="http://org.eclipse/epf/uma/resourcemanager.ecore">
3      <!--Identity-->
4      <xsl:template match="@*|node()">
5          <xsl:copy>
6              <xsl:apply-templates select="@*|node()"/>
7          </xsl:copy>
8      </xsl:template>
9      <!--Root-->
10     <xsl:variable name="root" select="'/org.eclipse.epf.openup.uma/src/'/>
11     <!--ResourceManager-->
12     <xsl:template match="org.eclipse.epf.uma.resourcemanager:ResourceManager"/>
13     <!--MethodLibrary-->
14     <xsl:template match="org.eclipse.epf.uma:MethodLibrary">
15         <xsl:copy>
16             <xsl:apply-templates select="@*|node()"/>
17             <xsl:element name="predefinedConfigurations">
18                 <xsl:attribute name="href">
19                     <xsl:value-of select="'configurations/all_epf_practices.
   uma#_QD87wFRHEd2CWscN8Mx6rg'"/>
20                 </xsl:attribute>
21             </xsl:element>
22             <xsl:element name="predefinedConfigurations">
23                 <xsl:attribute name="href">
24                     <xsl:value-of select="'configurations/openup.uma#_QN3nQBEHEdyM7Iu0sxfrPA'"/>
25                 </xsl:attribute>
26             </xsl:element>

```

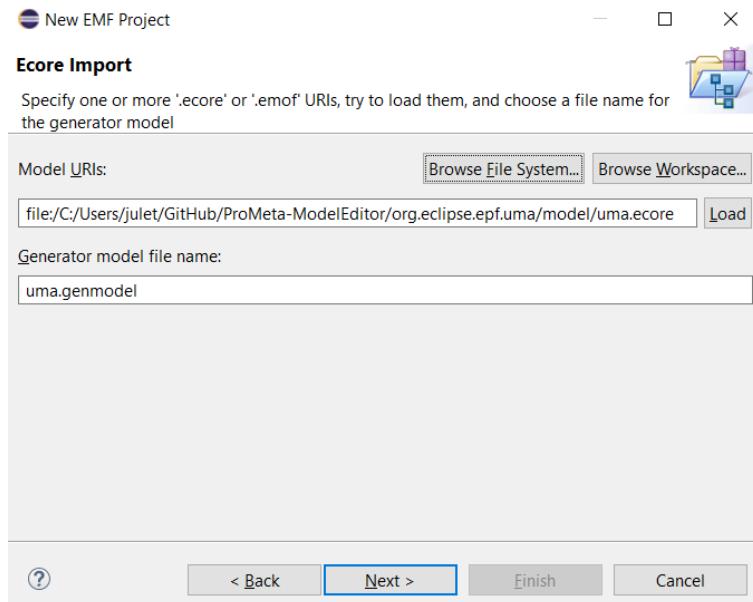
11.8. Irudia. Eraldaketa egiteko XSLT fitxategia.

Esan bezala, erabiliko dugun metaeredua UMA da, jadanik definituta dagoena. Beraz, lehenik metaeredu hori deskargatu behar da Eclipse-ren webgunetik. Metaereduarekin lan egiteko EMF proiektu bat sortuko dugu, 11.9. Irudian ikusten den bezala.



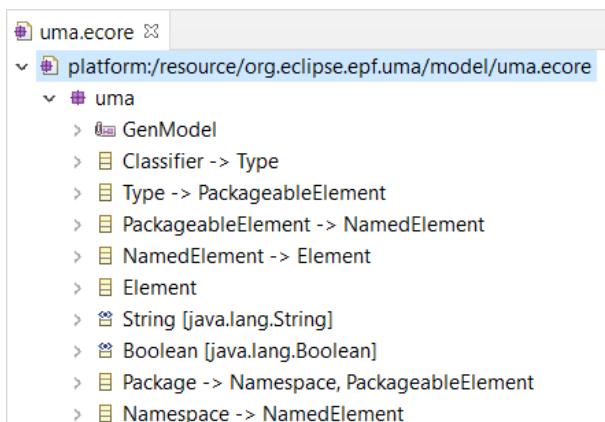
11.9. Irudia. EMF proiektuaren sorrera Eclipsen.

Proiektuaren izena aukeratu ondoren importatzeko *Ecore model* aukeratuko dugu eta 11.10. Irudian dagoen leihora eramango gaitu. Bertan `uma.ecore` metaeredua aukeratu beharko dugu eta *Next* botoia sakatu. Hurrengo lehioan *Finish* sakatzen badugu proiektua sortuko da.



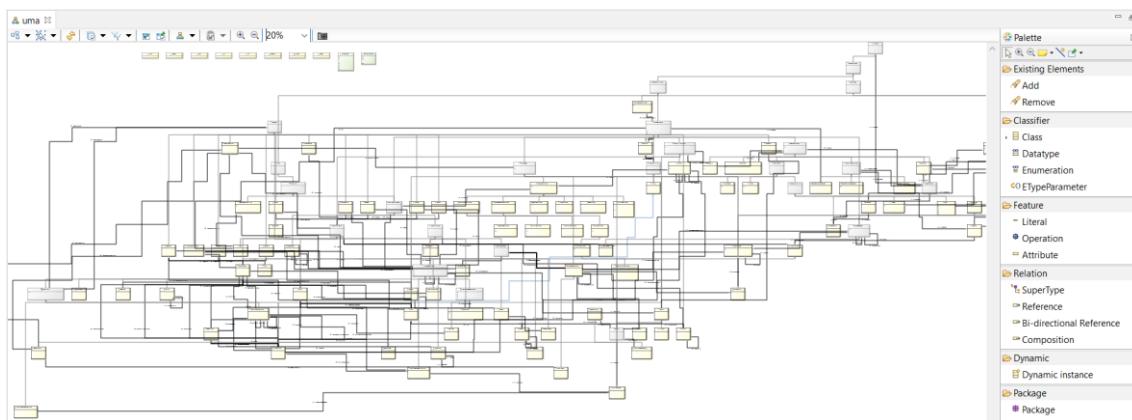
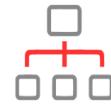
11.10. Irudia. Uma metaeredua importatzen Eclipsen.

Proiektu horrek `uma.ecore` eta `uma.genmodel` fitxategiak izango ditu. Fitxategi horiek *Sample Reflective Ecore Model Editor* erabiliz ireki eta editatu daitezke. Adibidez, `uma.ecore` fitxategiaren zati bat ikus daiteke 11.11. Irudian.



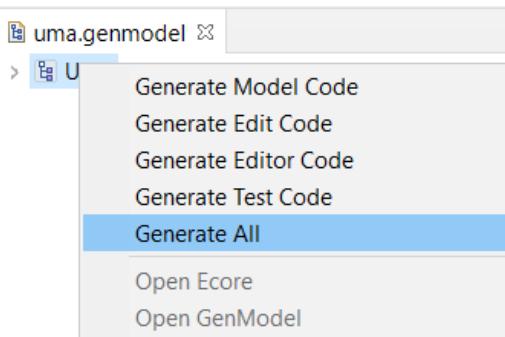
11.11. Irudia. Uma.ecore fitxategia Eclipsen.

Hurrengo pausua `uma.aird` fitxategia sortzea izango da, metaereduaren diagrama bat sortzeko klase eta atributu guztiekin. Horretarako, `uma.ecore` fitxategian *Initialize Ecore Diagram* aukeratuko dugu eta ondoren *Entities in a Class Diagram*. Diagramaren izena aukeratu eta *Finish* botoia sakatutakoan klase diagrama sortuko da. 11.12. Irudian ikus daitekeen moduan, metaeredua osoa handia da eta klase pila bat daude. Aldaketaren bat egitea nahiko bagenu eskuineko paleta erabili beharko genuke.



11.12. Irudia. Uma metaereduaren klase diagrama.

Behin metaereduaren proiektua prest daukagula, editore grafikoa sor dezakegu. Horretarako, uma.genmodel fitxategia ireki beharko dugu. Kodea sortzeko hainbat aukera daude, baina simpleena *Generate All* aukeratzea da, 11.13. Irudian ikusten den moduan. Izan ere, lehenengo 3 aukerak beharrezkoak dira editorea sortzeko eta 4. aukera probentzako da.



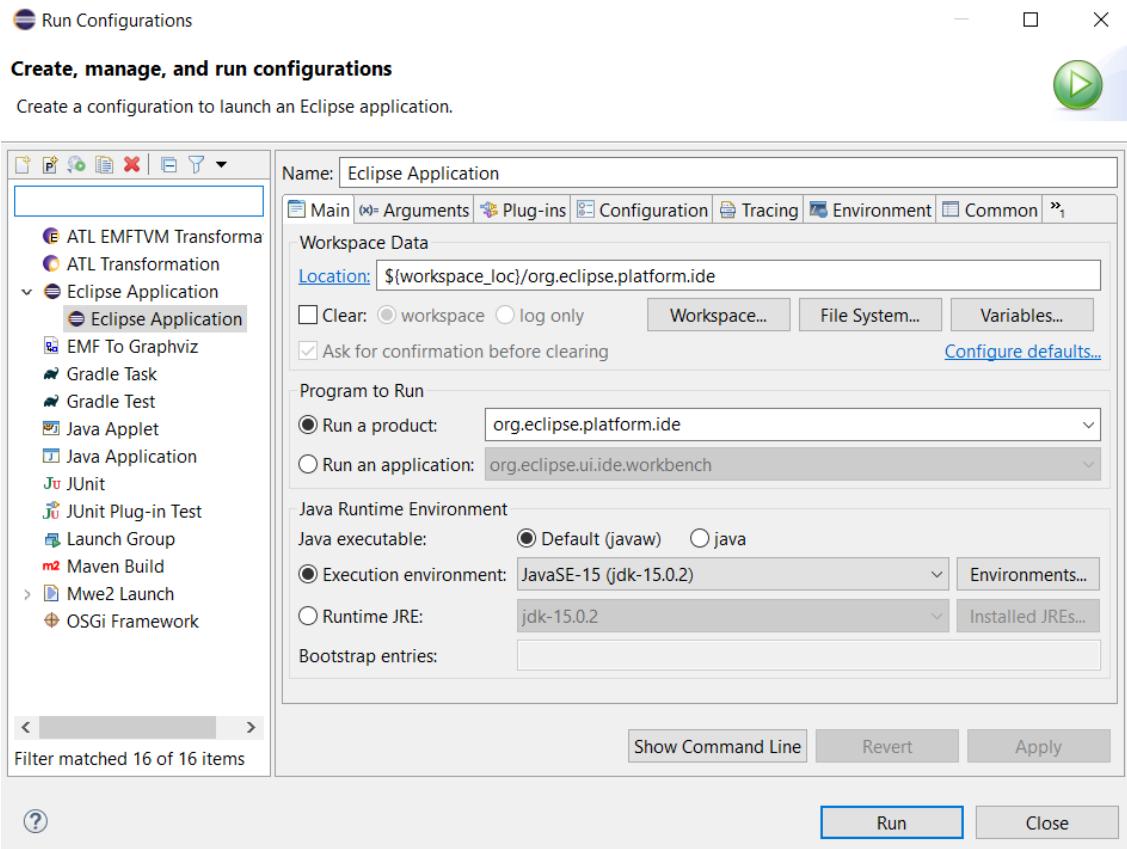
11.13. Irudia. Editore grafikoaren kodearen sorrera.

Lehenengo aukerak proiektuan bertan sortuko du ereduaren Java kodea. Beste aukera bakoitzeko Java proiektu berri bat sortuko da automatikoki, 11.14. Irudian ikusten den bezala. Honekin jadanik editore grafikoa daukagu, baina itxura hobea edukitzea nahi badugu ikonoak gehitu beharko ditugu org.eclipse.uma.edit proiektuan.

- > org.eclipse.epf.uma
- > org.eclipse.epf.uma.edit
- > org.eclipse.epf.uma.editor
- > org.eclipse.epf.uma.tests

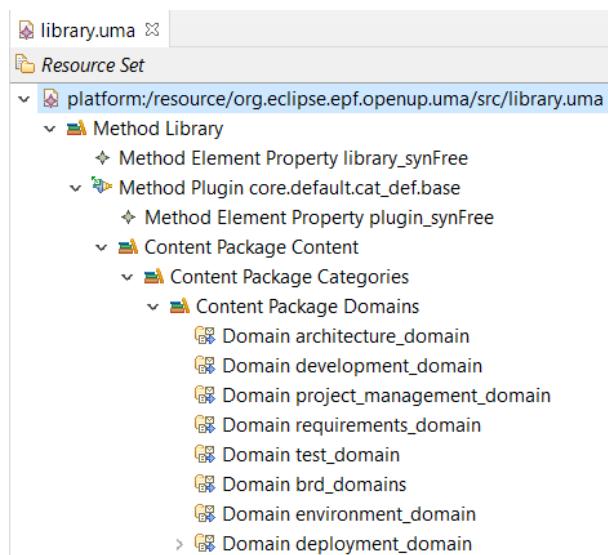
11.14. Irudia. Editore grafikoaren proiektuak Eclipsen.

Dena prest daukagunean editorea martxan jar dezakegu *Run As > Eclipse Application* aukeratuz. Irekkitzen dugun lehenengo aldia bada, konfiguratu egin beharko dugu. Adibidez, 11.15. Irudian agertzen den konfigurazioa aukera daiteke.



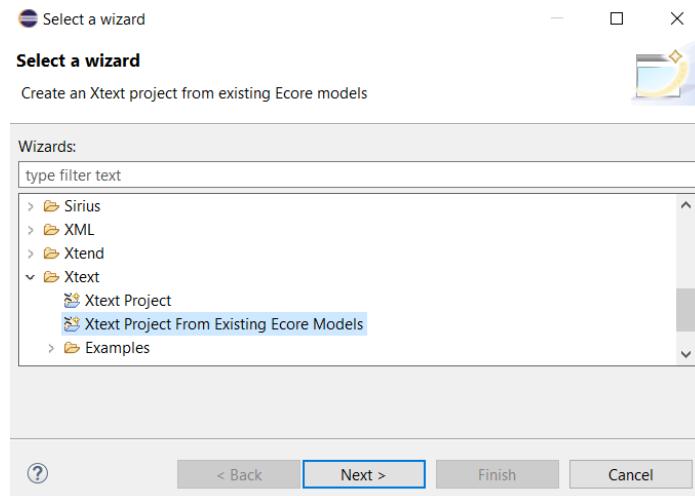
11.15. Irudia. Editore grafikoa martxan jartzeko konfigurazioa.

Martxan jartzen dugunean Eclipse aplikazio berri bat irekiko da. Bertan proiektu berri bat sortu beharko dugu eta XSLT eraldaketatik lortutako .uma fitxategiak kargatu. Edozein .uma fitxategi irekitzeko aukera izango dugu *Uma Model Editor* erabiliz. Editore honek EPF Composer-en antza du ikonoak eta zuhaitz egitura partekatzen dituelako. Adibidez, library.umä fitxategia irekitzen badugu metodologien eduki osoa bistaratu da, 11.16. Irudian ikus daitekeen bezala.



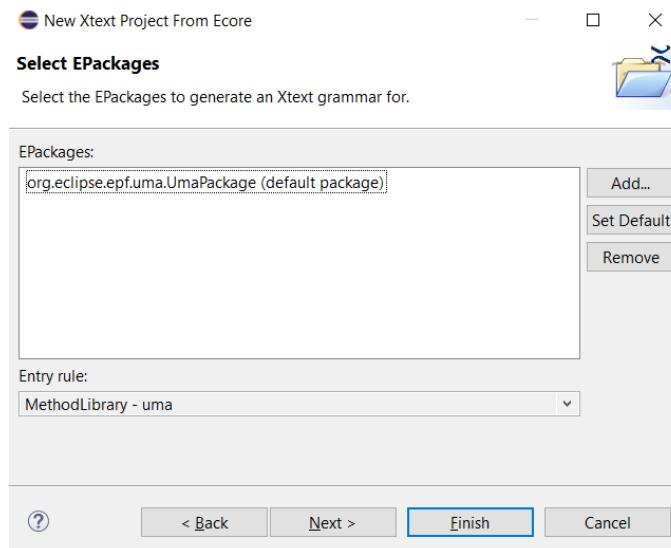
11.16. Irudia. Editore grafikoarekin library.umä fitxategia bistaratzten.

Editore grafikoa bukatuta dagoenez, testu editorearekin has gaitezke. Testu editorea sortzeko Xtext tresna erabiliko dugu. Helburua DSL bat sortzea denez, gramatika bat definitu beharko dugu. Metaeredua erabil dezakegu gramatika automatikoki sortzeko, 11.17. Irudian agertzen den *Xtext Project From Existing Ecore Models* aukera erabilita.



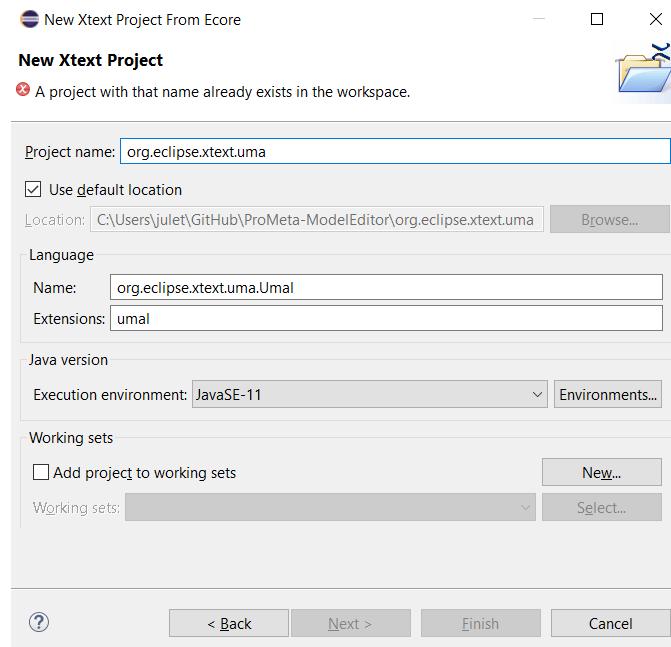
11.17. Irudia. Xtext proiectua sortzen Eclipsen.

Hurrengo lehioan `Add` botoia sakatu eta `uma.genmodel` fitxategia aukeratuko dugu, 11.18. Irudian ikus daitekeen bezala. Behoko aukeretan metaereduaren erroa aukeratu behar da, kasu honetan *MethodLibrary* dena.



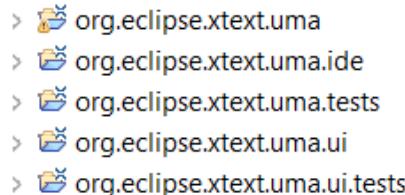
11.18. Irudia. Uma metaeredutik Xtext proiectua sortzen.

Hurrengo lehioan proiectuaren izena, lengoariaren izena eta fitxategien luzapena aukeratu beharko ditugu, 11.19. Irudian ikusten den moduan.



11.19. Irudia. Xtext proiektuaren eta lengoaiaren izenak aukeratzen.

Amaitutakoan 11.20. Irudian agertzen diren 5 Xtext proiektuak sortuko dira. Lehenengo proiektu nagusia da, gramatika eta fitxategi garrantzitsuenak izango dituena. Beste bi proiekturek editorearen kodea dute eta probarako bi proiektu ere badaude.



11.20. Irudia. Testu editorearen proiektuak Eclipsen.

Gramatika lehenengo proiektuan dago eta Umal.xtext izena du. Metaeredua oso handia denez, sortutako gramatikak 2600 lerro inguru ditu eta errore batzuekin sortzen da. Adibidez, 11.21. Irudian gramatikaren erroren zatia ikus daiteke.

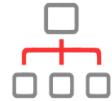


```

Umal.xtext
1 // automatically generated by Xtext
2 grammar org.eclipse.xtext.uma.Umal with org.eclipse.xtext.common.Terminals
3
4 import "http://www.eclipse.org/epf/uma/1.0.6/uma.ecore"
5 import "http://www.eclipse.org/emf/2002/Ecore" as ecore
6
7@MethodLibrary returns MethodLibrary:
8   'MethodLibrary'
9   guid=String0
10  '{'
11    ('name' name=String0)?
12    ('presentationName' presentationName=String0)?
13    ('briefDescription' briefDescription=String0)?
14    ('suppressed' suppressed=Boolean)
15    ('orderingGuide' orderingGuide=String0)?
16    ('authors' authors=String0)?
17    ('changeDate' changeDate=Date)?
18    ('changeDescription' changeDescription=String0)?
19    ('version' version=String0)?
20    ('kind' '(' kind=[Kind] ( "," kind+[Kind])* ')')?
21    ('copyrightStatement' copyrightStatement=[SupportingMaterial])?
22    ('ownedRules' '(' ownedRules+=Constraint ( "," ownedRules+=Constraint)* ')')?
23    ('methodElementProperty' '(' methodElementProperty+=MethodElementProperty ( "," methodElementProperty+=MethodElementProperty)* ')')?
24    ('methodPlugins' '(' methodPlugins+=MethodPlugin ( "," methodPlugins+=MethodPlugin)* ')')?
25    ('predefinedConfigurations' '(' predefinedConfigurations+=MethodConfiguration ( "," predefinedConfigurations+=MethodConfiguration)* ')')?
26  '}';

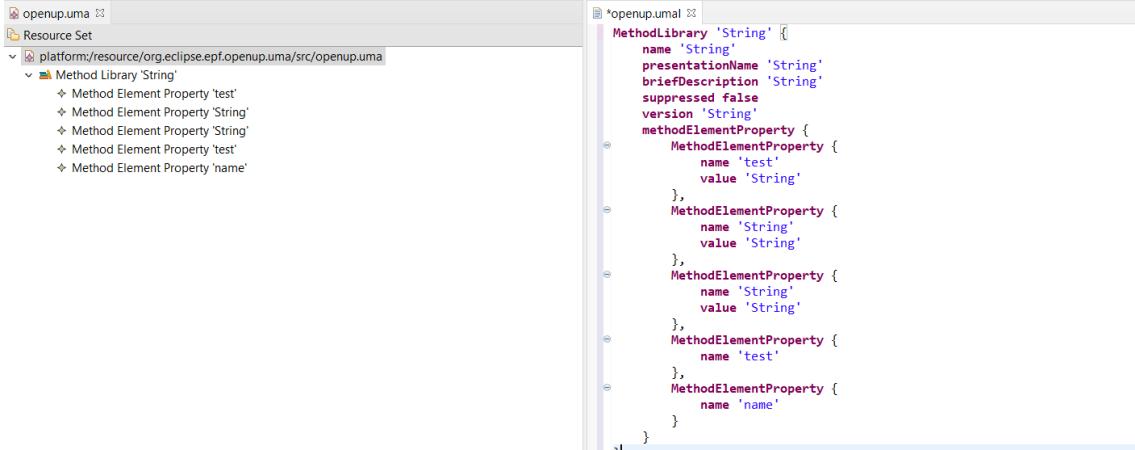
```

11.21. Irudia. Xtext proiektuko lengoaiaren gramatika.



Behin gramatika konpondutakoan `GenerateUmal.mwe2` fitxategiaan *Rus As > MWE2 Workflow* aukeratu beharko dugu editorearen kodea sortzeko. Gainerako proiektuetan editorearen Java eta Xtend kodea sortuko da.

Editorea exekutatzeko modua editore grafikoaren berdina da eta konfigurazio bera aukeratzen badugu, bi editoreak batera erabil ahal izango ditugu. Eredu osoa erabili ordez, adibide simple batekin hobeto ikus ditzakegu bi editoreak 11.22. Irudian.



```

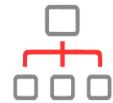
*openup.uma
MethodLibrary 'String' {
    name 'String'
    presentationName 'String'
    briefDescription 'String'
    suppressed false
    version 'String'
    methodElementProperty {
        MethodElementProperty {
            name 'test'
            value 'String'
        },
        MethodElementProperty {
            name 'String'
            value 'String'
        },
        MethodElementProperty {
            name 'String'
            value 'String'
        },
        MethodElementProperty {
            name 'test'
        },
        MethodElementProperty {
            name 'name'
        }
}

```

11.22. Irudia. Editore grafikoa ezkerrean eta testu editorea eskuinean.

Hala ere, oraindik aspektu garrantzitsu bat falta da, editoreak ez baitira elkarren artean sinkronizatzen. Helburua da bietako bat aldatzen badugu automatikoki segundu gutxi batzutan bestearekin sinkronizatzea. Horrela, nahiz eta bi fitxategi desberdin izan, edozein momentutan alda daiteke editorez. Hau garrantzitsua da editore bakoitzak bere abantailak dituelako. Editore grafikoak eredua modu bisualean ikustea ahalbidetzen du eta testu editoreak, berriaz, aldaketa errepikakorrak azkarrago egitea errazten du.

Beraz, bi itzultaile behar ditugu, uma-tik umal-era itzultzen duena eta umal-etik uma-ra itzultzen duena. Bigarren itzulketa nahiko sinplea da, kode sorreraren antzekoa baita. Izañ ere, Xtend-ek itzulketa hau errazten du serializatzailak definituz Xtend programazio lengoaia erabiliz. Lehenengoa zailagoa da, Xtend-ek uma fitxategien lengoaia detektatu behar baitu, eta umal lengoariaren antzeko tratamendua eman. Horretarako, Xtend-ek umal-erako autamatikoki sortu dituen fitxategietako batzuk eskuz sortu behar dira uma-erako. Adibide moduan serializatziale sinpleenaren kodea agertzen da 11.23. Irudian.



```

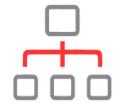
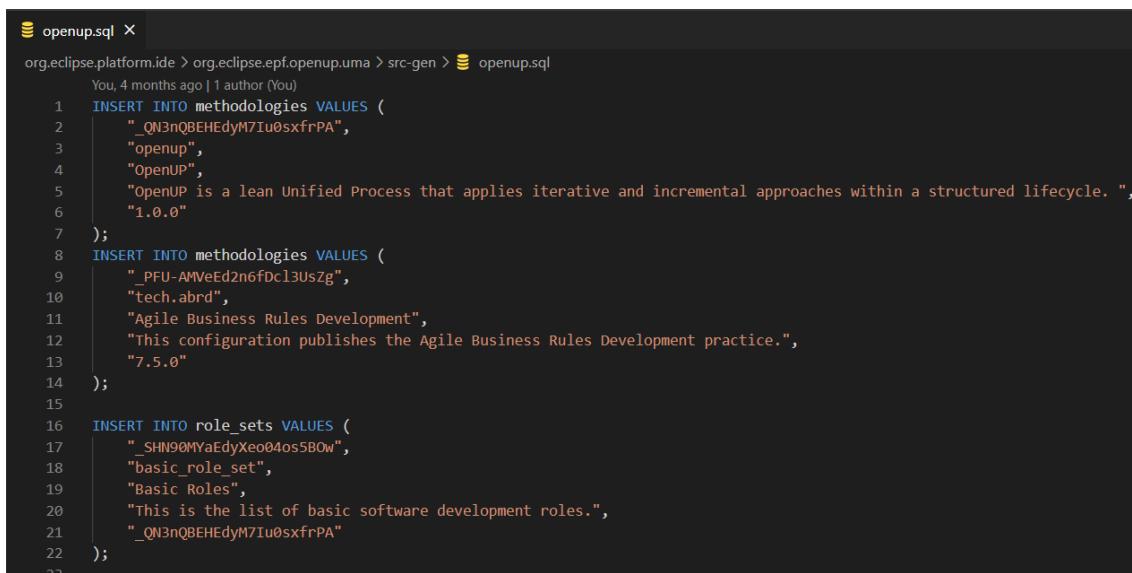
UmalGenerator.xtext
2+ * generated by Xtext 2.24.0
4 package org.eclipse.xtext.uma.generator
5
6@import java.io.IOException
17
18@/**
19 * Generates code from your model files on save.
20 *
21 * See https://www.eclipse.org/Xtext/documentation/303\_runtime\_concepts.html#code-generation
22 */
23@class UmalGenerator extends AbstractGenerator {
24
25@    override doGenerate(Resource input, IFileSystemAccess2 fsa, IGeneratorContext context) {
26        serialize(input, fsa)
27    }
28
29@    def serialize(Resource input, IFileSystemAccess2 fsa) {
30        UmaPackage.eINSTANCE.eClass()
31
32        val reg = Resource.Factory.Registry.INSTANCE
33        val m = reg.getExtensionToFactoryMap()
34        m.put("uma", new XMIResourceFactoryImpl())
35
36        val injector = new UmalStandaloneSetup().createInjectorAndDoEMFRegistration()
37        val resourceSet = injector.getInstance(ResourceSet)
38
39        val inputUri = input.URI
40        val outputUri = inputUri.trimFileExtension.appendFileExtension('uma')
41        val projectDir = System.getProperty("user.dir");
42        val directory = projectDir + "\\org.eclipse.platform.ide"
43        var outputPath = outputUri.toPlatformString(true)
44        outputPath = "file://" + directory + outputPath;
45
46        input.load(null);
47        EcoreUtil.resolveAll(resourceSet)
48
49        val xmiResource = resourceSet.createResource(URI.createURI(outputPath))
50        xmiResource.getContents().add(input.getContents().get(0))
51        try {
52            xmiResource.save(null)
53        } catch (IOException e) {
54            e.printStackTrace()
55        }
56    }
57 }

```

11.23. Irudia. Umal-etik uma-ra itzultzen duen serializatzailaren kodea.

Bukatzeko, aldaketak dauden bakoitzeari metodologiarekin informazio garrantzitsuena datu-base batean gordetzeko balio duen SQL fitxategia eguneratzen da. Editatzea bukatutakoan, fitxategi hori phpMyAdmin-etik importatuko dugu, datu-basean metodologiarekin informazio guztia gordetzeko.

Datu-basearen tamainaren ideia bat egiteko, sortutako fitxategiak 9400 lerro 1100 INSERT inguru dauzka. Argi dago, beraz, prozesu hau automatizatzearen erabilgarritasuna, informazio hori guztia eskuz gehitzeko denbora pila bat beharko genukeelako. 11.24. Irudian sortutako kodearen lehenengo 3 INSERT aginduak ikus daitezke.

```

openup.sql x
org.eclipse.platform.ide > org.eclipse.epf.openup.uma > src-gen > openup.sql
You, 4 months ago | 1 author (You)
1   INSERT INTO methodologies VALUES (
2     "_QN3nQBEHEdyM7Iu0sxfrPA",
3     "openup",
4     "OpenUP",
5     "OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle.",
6     "1.0.0"
7   );
8   INSERT INTO methodologies VALUES (
9     "_PFU-AMVeEd2ngfdcl3UsZg",
10    "tech.abrd",
11    "Agile Business Rules Development",
12    "This configuration publishes the Agile Business Rules Development practice.",
13    "7.5.0"
14 );
15
16   INSERT INTO role_sets VALUES (
17     "_SHN90MYaEdyXeo04os5B0W",
18     "basic_role_set",
19     "Basic Roles",
20     "This is the list of basic software development roles.",
21     "_QN3nQBEHEdyM7Iu0sxfrPA"
22 );
23

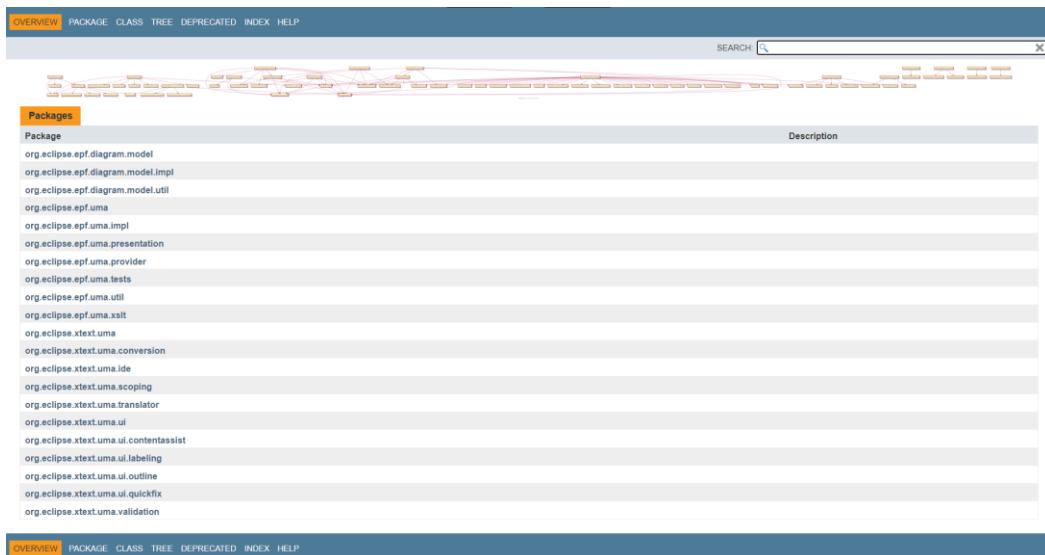
```

11.24. Irudia. Datu-basearen datuak dituen SQL fitxategia.

Kodea gordetzeko eta bertsio kontrolerako Git eta GitHub tresnak erabili dira. Dokumentazio webgunearen hosting-erako Gitub Pages erabili da. Java proiektuen dokumentazioa sortzeko modu ohikoa Javadoc erabiltzea da. Kasu honetan, plugin bat gehitu da, dokumentazioarekin batera diagramak automatikoki sortzeko.

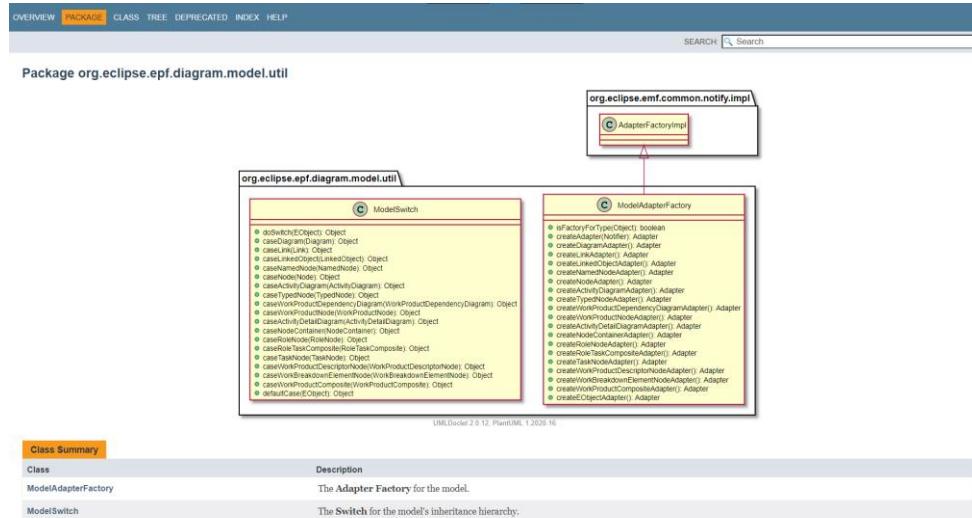
UMLDoclet tresnak Javadoc metadatuak erabiltzen ditu PlantUML diagramak automatikoki sortzeko eta HTML dokumentazioan txertatzeko. Diagramak irudi klikagarri gisa txertatzen dira eta pakete eta klaseko dokumentaziora estekatzen dira eskuragarri dagoenean. Pakete dependentzia, pakete eta klase diagramak sortzen ditu elementu guztiarako.

Dokumentazio webguneak Javadoc webgune arrunt baten itxura du, baina hainbat lekutan diagramak daude. 11.25. Irudian ModelEditor azpisistemako dokumentazio webgunearen hasierako orria agertzen da. Bertan paketeak zerrendatzen dira eta goiko aldean paketeen arteko dependentzia diagrama ikus daiteke.



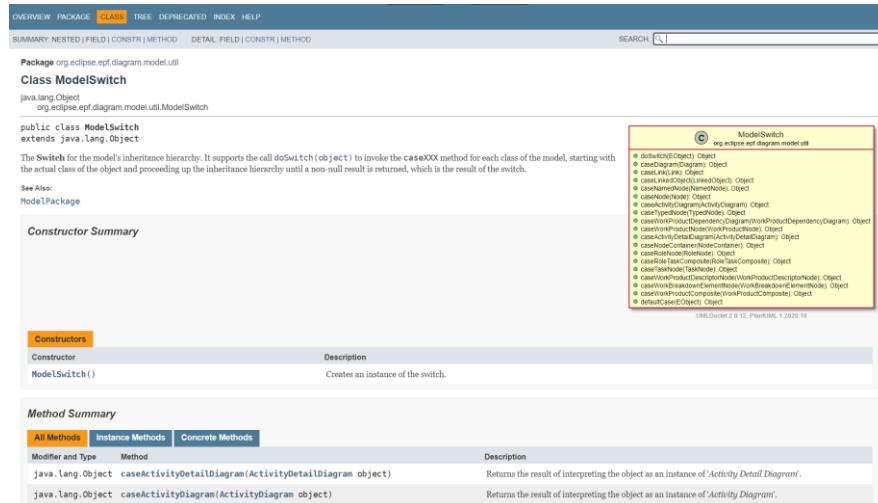
11.25. Irudia. ModelEditor-en paketeak eta paketeen arteko dependentziak.

Pakete bat aukeratzen badugu, diagrama eta klaseak bistaratu dira. 11.26. Irudian adibide bat ikus daiteke.



11.26. Irudia. Pakete diagrama adibidea.

Klase bat aukeratzen badugu, bere diagrama eta metodoak bistaratu dira. 11.27. Irudian ikus daiteke klase diagrama baten adibidea.



11.27. Irudia. Klase diagrama adibidea.

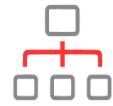
11.3 IO-System

Drupal CMSaren bitartez kudeatutako web-aplikazioa izango da. Helburua metodologia jarraitzen duten proiektuen informazioa gordetzea da. Rol bakoitzak metodologian dituen ataza berdinak bete beharko ditu.

11.3.1 Arkitektura

IO-System azpisistemak honako osagaiak izango ditu. 11.28. Irudian osagaien arteko loturak ikus daitezke.

- **Datu-basea:** Datu-basea erlazionalak ereduaren informazio garrantzitsuena gordeko du, Drupal webgunerako beharrezkoa izan daitekeena. Hau da, OpenUP eta ABRD metodologien faseak, iterazioak, jarduerak, atazak, artefaktuak, rolak, etab.



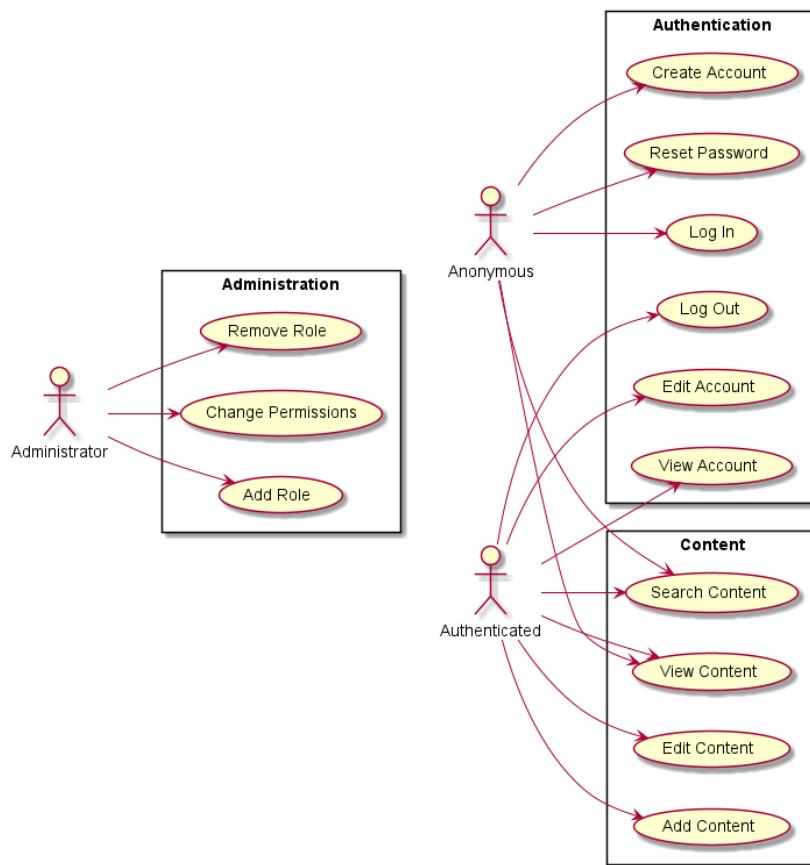
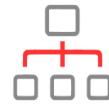
- **Drupal datu-basea:** Drupal sistemaren datu-basea eduki guzta gordetzeaz arduratzen da, fitxategiak izan ezik. Adibidez, erabiltzaileak, rolak, baimenak eta eduki motak gordetzean dira.
- **Drupal datu importatzalea:** Datu-importatzalea Drupal modulu multzo bat izango da. Hauen ardura aurretik aipatutako lehenengo datu-basetik Drupal datu-basera edukia importatzea da. Horretarako, nodoak sortu beharko dira, eta Drupal arduratu da edukia gordetzeaz.
- **Drupal fitxategiak:** Fitxategiak datu-basetik kanpo gordeko dira. Gure kasuan fitxategi gehienak artefaktuei dagozkienak izango dira, DOC eta PDF dokumentuak.
- **Drupal interfazea:** Interfazean edukia bistaratu eta aldatzeko aukera guztiak egongo dira. Esan bezala, edukia Drupal datu-basean gordeko da. Gainera, erabiltzailearen kontuekin zerikusia duten aukerak ere egongo dira. Horrez gain, administratzaileak aukera gehiagarri asko izango ditu webgunea kudeatzeko interfaze bidez.



11.28. Irudia. IO-System azpisistemaren arkitektura.

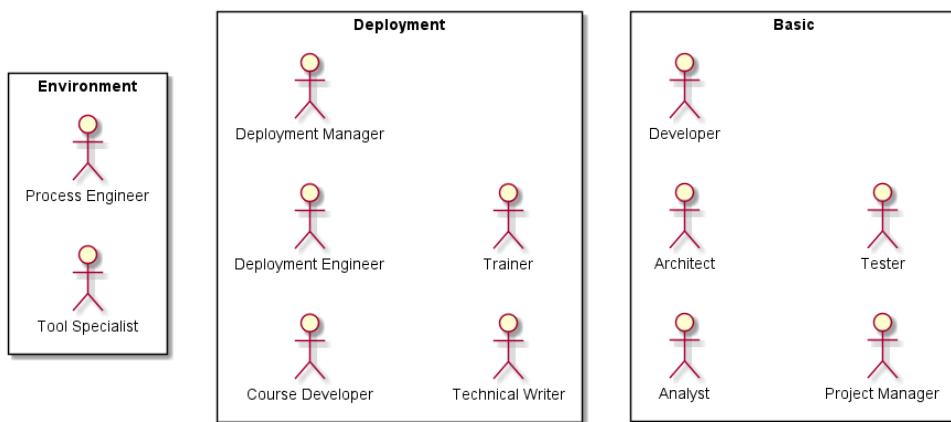
11.3.2 Erabilpen Kasuak

Sistemaren betekizunak zehazteko beharrezko da erabilpen kasuak eta aktoreak zein diren jakitea. Erabilpen kasuak 3 multzotan banatu daitezke: administrazioa, autentikazioa eta edukia. Aldi berean 3 erabiltzaile mota nagusi daude, anonimoa, autentikatua eta administraria. Erabiltzaile anonimoak autentikazioako ataza batzuk egiteko eta edukia ikusteko aukera bakarrik izango du. Erabiltzaile autentikatuak autentikazioko ataza batzuk egiteko eta edukia ikusi, editatu, gehitu eta bilatzeko auketak izango ditu. Administrariak erabiltzaile autentikatuak egin ditzakeen atazet gain administrazioko atzak izango ditu, rolei eta baimenei dagozkienak. 11.29. Irudian IO-System azpisistemako erabilpen kasuen ereduak agertzen da.



11.29. Irudia. IO-System azpisistemako erabilpen kasuen eredu.

Simplifikatzeko 3 rol nagusi jarri diren arren, erabiltzaile autentikatu mota asko daude, sistemako metodologietan dauden erabiltzaile adina. Adibidez, 11.30. Irudian ikusten da OpenUP metodologiako erabiltzaileak 3 multzotan banatu daitezkeela: ingurunea, hedapena eta oinarrizkoa.



11.30. Irudia. OpenUP metodologiako rolak multzokatuta.

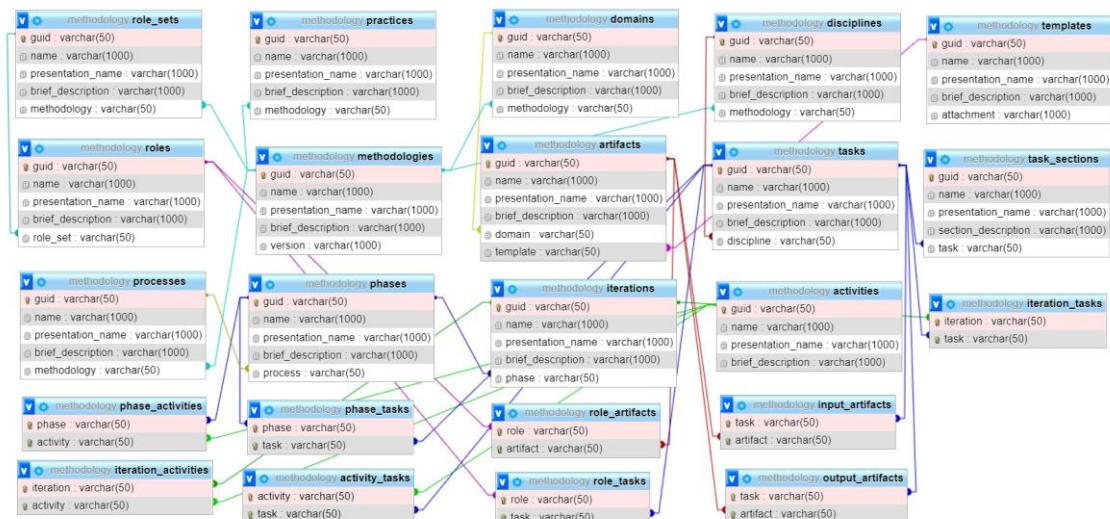
Erabiltzaile horietako bakoitzak OpenUP metodologian dagozkion ataza eta artefaktuak bete beharko ditu. Adibidez, proiektu kudeatzaileak 11.31. Irudian ikusten diren ataza eta artefaktuen ardura dauka. Sistema ideal batean rol bakoitzak beharrezkoak diren baimenak bakarrik eduki beharko lituzke. Hala ere, baimen guztiak kudeatzea oso konplexua denez, etorkizuneko hobekuntza moduan utzi da. Beraz, praktikan aurretik aipatutako 3 rol nagusiak bakarrik daudela esan dezakegu.



11.31. Irudia. Proiektu kudeatzaileak bete beharreko ataza eta artefaktuak.

11.3.3 Diseinua

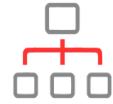
Metodologien informazio garrantzitsuena gordetzeko datu-base erlazionala erabili da. 11.1. Irudian ikusten den moduan, datu-basearen diseinuak eragina du bi azpisisteman. Atal honetan azaltzea erabaki da Drupal datu-basearekin lotura zuzenagoa duelako. Datu base honen diseinua nahiko konplexua da, 23 taula eta taulen arteko hainbat erlazio bainitu, 11.32. Irudian ikus daitezkeenak. Taula nagusiek identifikadore bat, izena, aurkezpen izen eta deskribapena dituzte. Gainera, atributu gehigarriak dituzte beste taulekin erlazioak adieratzeko. Gainerako taulak beste taulen arteko erlazioak adierazteko dira, bi taulen identifikatzaileak bakarrik dituzte.



11.32. Irudia. Datu-base erlazionalaren diseinua.

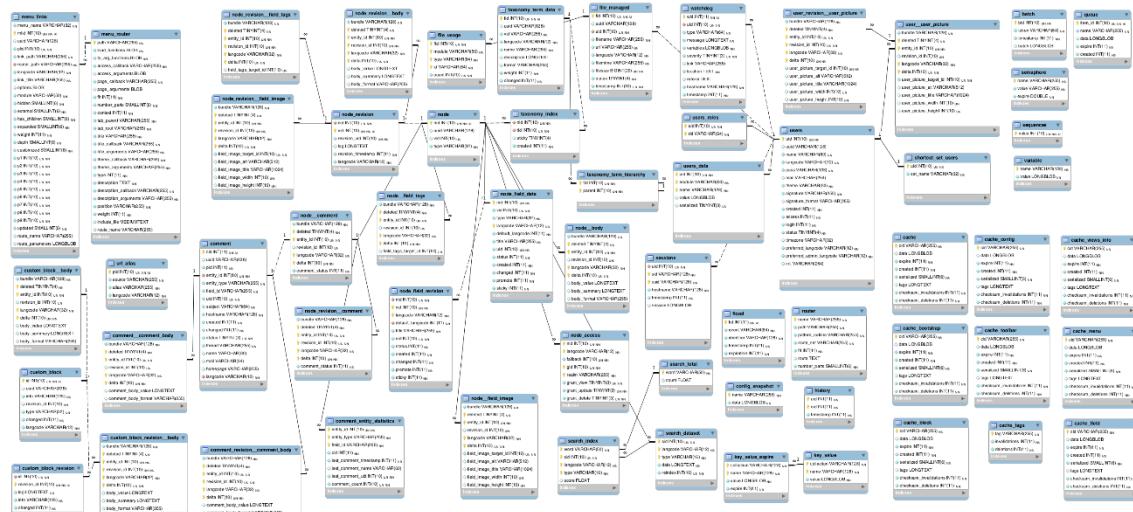
Jarraian, taula nagusien deskribapen labur bat egingo da diseinua hobeto ulertzeko:

- **methodologies:** metodologiaren datuak gordetzen ditu.
- **practices:** metodologiari dagozkion praktiken datuak gordetzen ditu.
- **roles:** prozesuko rolak gordetzen ditu.
- **role_sets:** metodologiari dagozkion rolen multzoak gordetzen ditu.
- **processes:** metodologik definitzen duen prozesuaren datuak gordetzen dira.
- **phases:** prozesuko faseen datuak gordetzen ditu.
- **iterations:** prozesuko iterazioen datuak gordetzen ditu.
- **activities:** prozesuko jardueren datuak gordetzen ditu.
- **tasks:** prozesuko atazen datuak gordetzen ditu.
- **disciplines:** prozesuko atazen multzoak gordetzen ditu.



- **task_sections**: prozesuko atazen sekzioak gordetzen ditu.
 - **artifacts**: prozesuko artefaktuen datuak gordetzen ditu.
 - **domains**: prozesuko artefaktuen multzoak gordetzen ditu.
 - **templates**: prozesuko artefaktuen txantiloia gordetzen ditu.

Datu horiek Drupal sistemara importatu direnez eduki moduan, Drupal-ek bere datu-basean gorde ditu. Drupal-ek atzetik erabiltzen duen datu-basearen diseinua jakitea beharrezkoa ez den arren, ideia bat egiteko sistema simple baten datu-basea ikus dezakegu 11.33. Irudian.



11.33. Irudia. Drupal datu-base baten diseinua.

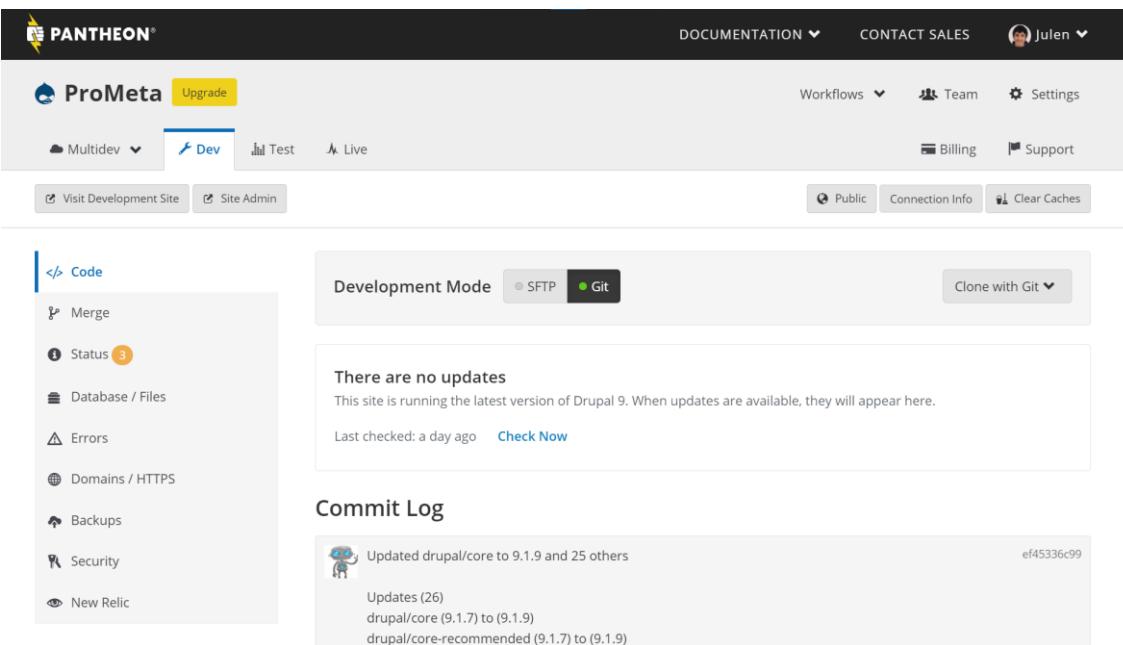
Gure datu-basearekin konparatuz ikus daiteke askoz konplexuagoa dela, Drupal-ek informazio guztia gordetzen du datu-basean gordetzen baitu. Gainera, IO-System sistemaren datu-basea oraindik handiago da, 113 taula ditu guztira. Izan ere, eduki mota asko definitu ditugu eta bakoitzarentzat hainbat taula sortzen dira. Gainera, kontuan hartu behar da metodologiez gain proiektuen informazioa ere gordetzen duela.

11.3.4 Dokumentazioa

Esan bezala, azpisistemaren garapenerako Drupal 9 erabili da. Uneoro azkenengo bertsio egonkorra erabiltzea erabaki da. Horretarako, aktualizazioak zeuden bakoitzean Drupal eguneratu da *composer* tresna erabiliz. Programazio lengoai nagusia PHP izan da, Drupal lengoai horretan idatzita baitago.

Webgunearen lokalean garatzeko XAMPP software libreko paketea erabili da. Honek MariaDB datubaseen kudeaketa sistema eta Apache web zerbitzaria integratzen ditu. Gainera, datu-basearen kudeaketa errazteko *phpMyAdmin* eskaintzen du, web nabigatzaile bidez erabiltzen dena. Editore moduan *Visual Studio Code* erabili da.

Kodea gordetzeko eta bertsio kontrolerako *Git* eta *GitHub* tresnak erabili dira. *Hosting*-erako *Pantheon* erabili da, Drupal webguneentzako doako *hosting*-a eskaintzen duen plataforma. Aldaketak lokalean egin eta probatu ondoren, *GitHub*-era eta *Pantheon*-era igo dira, webgunea eta kodea publikoki eskuragarri egoteko. *Pantheon*-en webgunearen edukia eguneratuta mantentzeko, beharrezko da fitxategiak eta datu-basea ere igotzea. 11.34. Irudian ProMeta webguneko *Pantheon* kontrol panela ikus daiteke.



11.34. Irudia. ProMeta webguneko Pantheon kontrol panela.

Drupal bezalako CMS batek eskaintzen dituen funtzionalitateak aprobetxatzeko, ahal den guzia bertako aukerak erabiliz egitea erabaki da. Hau da, edukia gordetzeko kanpoko datu-base bat erabili ordez Drupal-eko eduki motak erabiltzea erabaki da. Guztira 13 eduki mota sortu dira: metodologia, prozesua, iterazioa, jarduera, ataza, artefaktua, disziplina, domeinua, rola, proiektua, kidea eta proiektuko artefaktua. Eduki motak sortzeaz gain, bakoitzarentzak bista bat definitu da, eta menuko aukerak gehitu dira edukia errazago gehitu eta ikusteko.

Eduki guzia eskuz gehitzeak zentzurik ez duenez, garatu beharreko lehenengo funtzionalitatea kanpoko datu-basetik edukia hartu eta automatikoki Drupal-era gehitzea da. Drupal-en funtzionalitateak gehitzeko modurik onena moduluak direnez, eduki mota bakoitza importatzeko modulu bat garatu da, 11.35. Irudian ikus daitekeen bezala.

11.35. Irudia. Drupal webgunera datu-basetik edukia importatzeko moduluak.

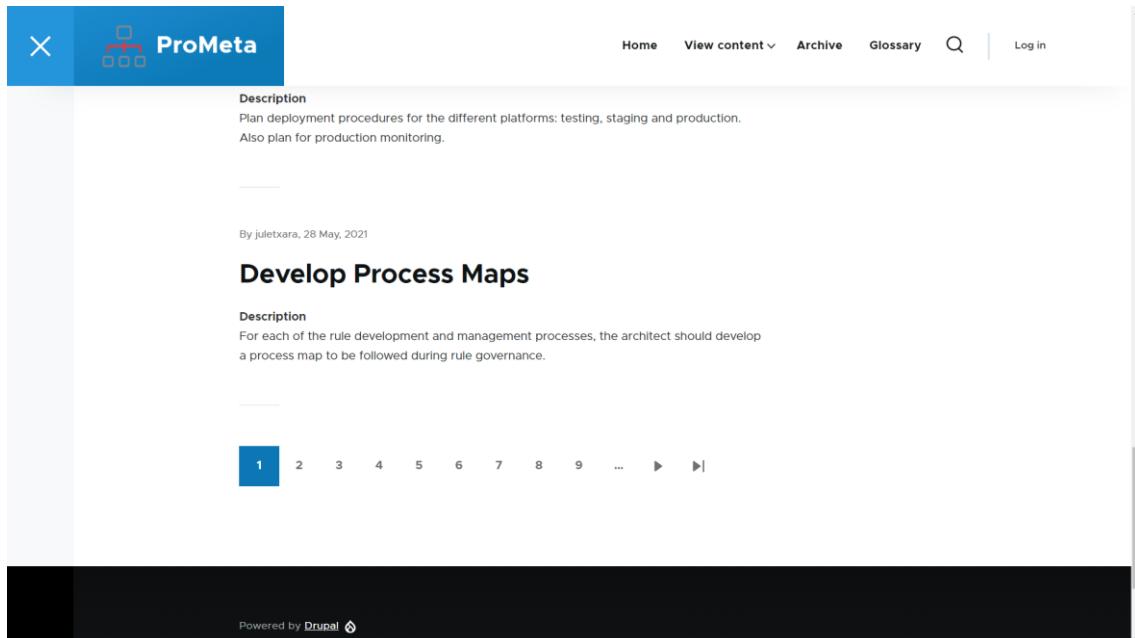
Gainera, Drupal-ekin batera etortzen diren modulu batzuk aktibatu dira funtzionalitateak gehitzeko. Bukatzeko, kanpoko bi modulu ere instalatu dira, [Admin Toolbar](#) administrazio menua hobetzeko eta [Menu Item Role Access](#) menuko baimenak kudeatzeko.

Itxurari dagokionez, Drupal-ek defektuz eskaintzen dituen gaiak oso zaharkituak daudenez, aldatzea erabaki da. Adibide moduan, ProWF sistemaren hasierako orria ikus daiteke *Bartik* gaiarekin. Beraz, webgune nagusirako *Bartik*-en ordez *Olivero* aukeratu da. Administrazio webgunerako, berriz, *Seven* gaia *Claro*-rekin ordezkatu da. Aldaketa simple honekin webguneak askoz itxura modernoagoa hartu du. Gainera, itxura ez ezik funtzionalitate gehigarriak ere eskaintzen dituzte, adibidez menu hedagarriak.

11.36. Irudia. ProWF sistemaren hasierako orria.

ProMeta webgunera sartutakoan hasierako orria bistaratzen da, 11.37. Irudian ikusten dena. Goiburuaren ezkerraldean sistemaren logoa eta izena agertzen dira eta eskuinaldean menua. Erdialdean sistemara gehitutako edukia agertzen da ordena kronologikoan. Oinean webgunea Drupal-ekin eginda dagoela dioen mezu bat bakarrik dago.

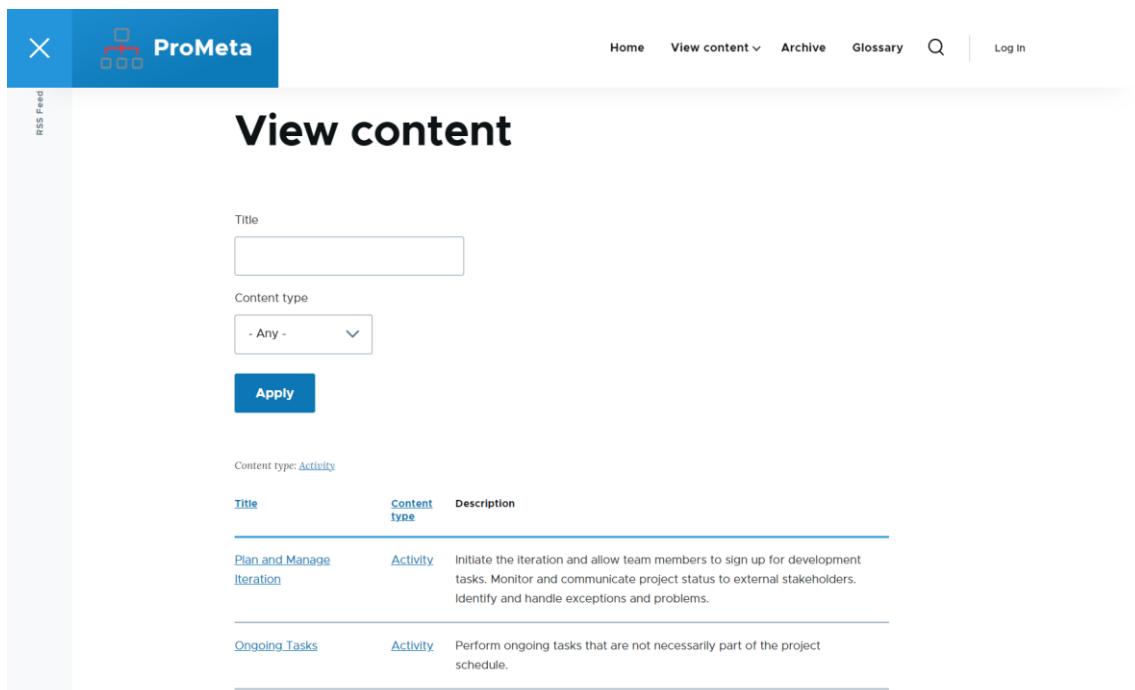
Menuaren ezkerraldean hainbat esteka daude: hasierako orriko joateko esteka, edukia ikusteko estekak, erregistroa eta glosarioa. Erdian bilaketak egiteko botoia dago eta eskuinean saioa irekitzekoa.



The screenshot shows a Drupal-based website for 'ProMeta'. The header includes the logo of the University of the Basque Country, the project name 'ProMeta' with a stylized tree icon, and navigation links for Home, View content, Archive, Glossary, and Log in. A search bar is also present. The main content area displays a 'Description' section for a process map, mentioning deployment procedures for testing, staging, and production, and a plan for production monitoring. Below this is a 'Develop Process Maps' section with its own description. A navigation menu at the bottom shows pages 1 through 9, with page 1 selected. The footer contains a 'Powered by Drupal' link.

11.37. Irudia. Drupal webgunearen hasierako orria.

Eduki ikusteko aukerari ematen badiogu, edukia bistaratuko da taula batean motaren arabera multzokatuta. Izenburu eta eduki motarekin bilaketak egiteko aukera ere badago, 11.38. Irudian ikusten den moduan.

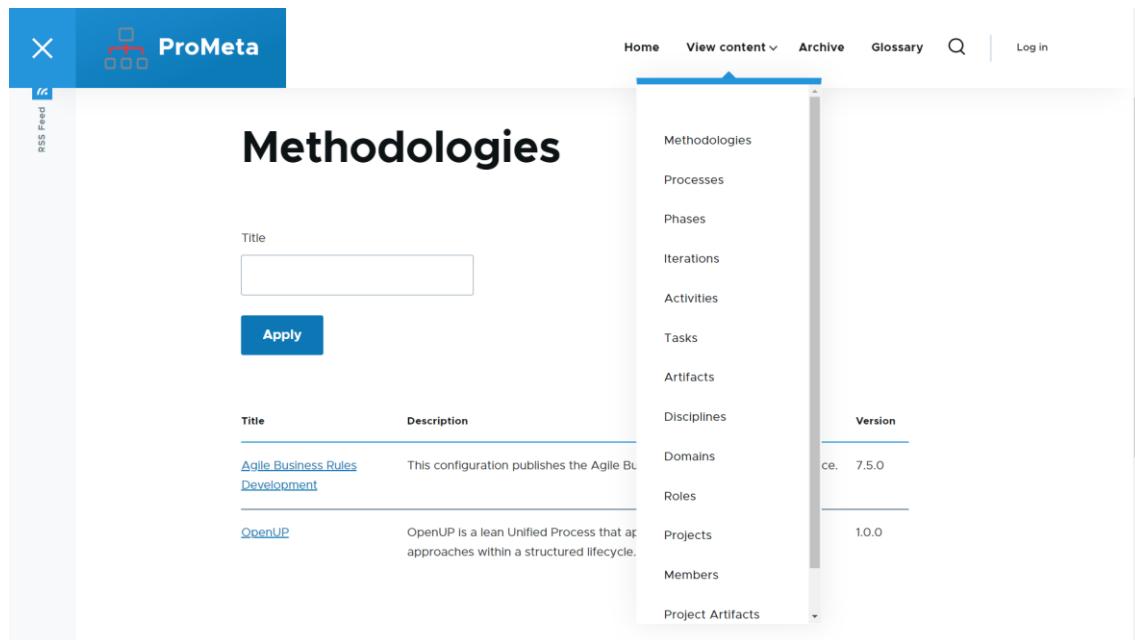


The screenshot shows the 'View content' page of the ProMeta website. It features a sidebar with an 'RSS Feed' link. The main content area has a title 'View content' and search fields for 'Title' and 'Content type' (set to 'Any'). A blue 'Apply' button is below these. A note states 'Content type: Activity'. The main table lists two items: 'Plan and Manage Iteration' (Activity) and 'Ongoing Tasks' (Activity). Both entries have a brief description. The table columns are 'Title', 'Content type', and 'Description'.

11.38. Irudia. Drupal webguneko edukia ikusteko orria.

Eduki mota zehatz bat bakarrik ikusi nahi badugu zehaztasun gehiagorekin, menu hedagarria erabil dezakegu. Guztira 13 eduki mota daude: metodologia, prozesua, fasea, iterazioa, jarduera, ataza,

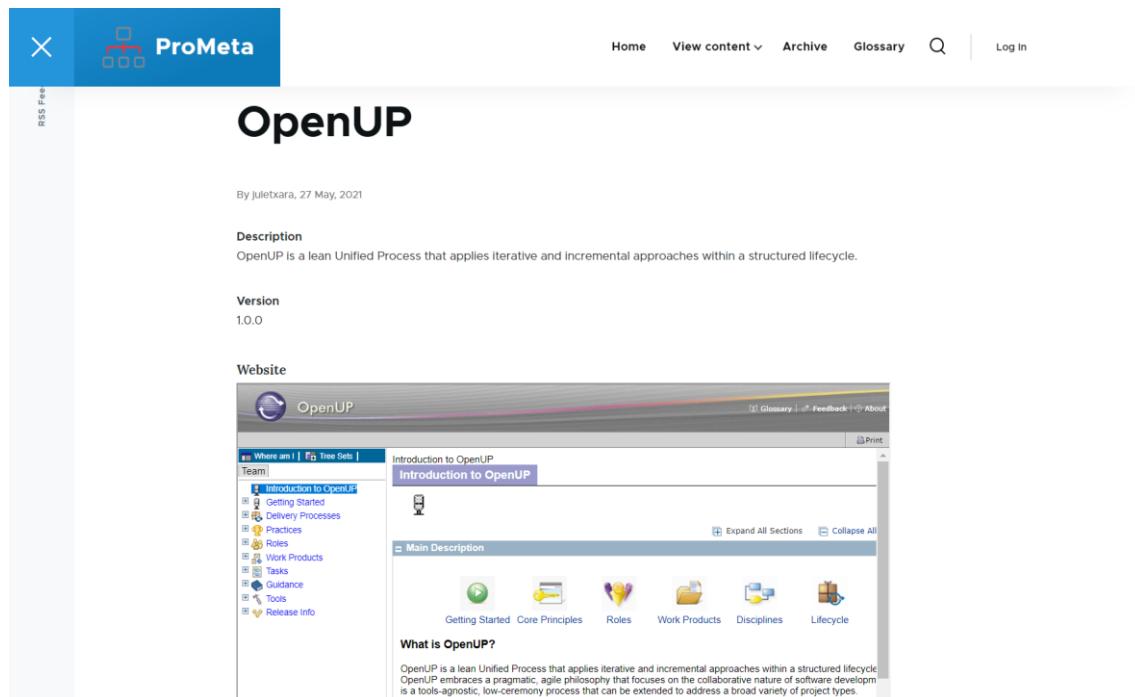
artefaktua, disziplina, domeinua, rola, proiectua, kidea eta proiectuko artefaktua. Adibidez, 11.39. Irudian metodologiak bistaratzen dira eta izenburuaren arabera bilaketak egin daitezke.



Title	Description	Disciplines	Version
Agile Business Rules Development	This configuration publishes the Agile Business Rules Development	ce.	7.5.0
OpenUP	OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle.	Projects	1.0.0

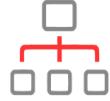
11.39. Irudia. Drupal webguneko metodologiak ikusteko orria.

Eduki bat zehaztasun gehiagorekin ikusi nahi badugu, izenburuan klik egin behar dugu. Adibidez, OpenUP metodologia aukeratzen badugu, bere webgunea bistaratuko da, 11.40. Irudian ikusten den bezala.



11.40. Irudia. Drupal webgunean OpenUP metodologia ikusteko orria.

Artxiboa aukeratzen badugu, edukia hilabeteka sailkatuta bistaratuko da. Adibidez, maiatza aukeratzen badugu, hilabete horretan gehitutako edukia bakarrik agertuko da, .



  **ProMeta**

Home View content Archive Glossary  Log in

Home > Monthly archive

May 2021

By juletxara, 28 May, 2021

Maintain Rules

Description
The goal of this task is to maintain the rules in order to correct faults or improve performance.

By juletxara, 28 May, 2021

Maintain Rules

Description
The goal of this task is to maintain the rules in order to correct faults or improve performance.

11.41. Irudia. Drupal webgunean maiatzean gehitutako edukia.

Glosarioan klik egiten badugu, berriz, edukia hasierako letraren arabera sailkatuta agertuko da. Adibidez, B hizkian klik egiten badugu, letra horrekin hasten den edukia bakarrik bistaratuko da, 11.42. Irudian ikus daitekeen bezala.

  **ProMeta**

Home View content Archive Glossary  Log in

Glossary

[A](#) (19) | [B](#) (17) | [C](#) (13) | [D](#) (100) | [E](#) (40) | [F](#) (1) | [G](#) (7) | [H](#) (40) | [I](#) (1) | [J](#) (6) | [L](#) (8) | [M](#) (7) | [P](#) (22) | [R](#) (35) | [S](#) (15) | [T](#) (25) | [U](#) (3) | [V](#) (5) | [W](#) (1)

Title	Author	Last update
Backout Plan	juletxara	Thursday, May 27, 2021 - 23:02
Build	juletxara	Thursday, May 27, 2021 - 23:02
Build Object Models	juletxara	Friday, May 28, 2021 - 16:58
Build Object Models	juletxara	Friday, May 28, 2021 - 16:58
Build Rule Project Structure	juletxara	Friday, May 28, 2021 - 16:58
Build Rule Project Structure	juletxara	Friday, May 28, 2021 - 16:58
Build Rule Project Structure	juletxara	Friday, May 28, 2021 - 16:58

11.42. Irudia. Drupal webguneko glosario orria.

Eduki bilatzeko botoian klik egiten badugu, idazteko barra zabalduko da. Idatzitakoarekin bat egiten duten edukiak bakarrik azalduko dira. Adibidez, 11.43. Irudian OpenUP hitza duten edukiak bakarrik agertzen dira bilaketan.

The screenshot shows a Drupal-based website with a blue header bar. The header includes links for 'Content', 'Structure', 'Appearance', 'Extend', 'Configuration', 'People', 'Reports', and 'Help'. The main navigation bar has items for 'Home', 'View content', 'Add content', 'Archive', 'Glossary', 'My account', and 'Log out'. A search bar at the top right contains the text 'OpenUP'. Below the header, a large search results page is displayed with the title 'Search for OpenUP'. It features a search input field with 'OpenUP' typed in, a 'Search' button, and links for 'About searching' and 'Advanced search'. The main content area is titled 'Search results' and lists an item by 'Julen Etxaniz' from May 27, 2021. The item is titled 'OpenUP Lifecycle' and describes it as a methodology designed for small teams. The URL for this item is [OpenUP Lifecycle](#).

11.43. Irudia. Drupal webgunean edukia bilatzeko aukera.

Orain arteko funtzionalitateak erabiltzeko ez zen beharrezkoa webgunean saioa hastea. Baino, edukia gehitu edo aldatu nahi badugu, saioa hasi beharko dugu. Saioa irekitzeko botoiari ematen badiogu beste orri batera bideratuko gaitu. Bertan 3 aukera izango ditugu: saioa hasi, kontu berria sortu eta pasahitza berrezarri. 11.44. Irudian ikusten den bezala, saioa hasteko erabiltzaile izena eta pasahitza eskatzen dira.

The screenshot shows the 'Log in' page of the ProMeta website. At the top, there is a blue header bar with the 'ProMeta' logo and navigation links for 'Home', 'View content', 'Archive', 'Glossary', and 'Log in'. Below the header, there are three buttons: 'Log in' (highlighted in blue), 'Create new account', and 'Reset your password'. The main content area is titled 'Log in' and contains two input fields: 'Username *' and 'Password *'. Each field has a placeholder text below it: 'Enter your ProMeta username.' and 'Enter the password that accompanies your username.'. A 'Log in' button is located at the bottom of the form.

11.44. Irudia. Drupal webguneko saioa hasteko orria.

Kontu berria sortzeko izena, abizena, posta, erabiltzaile izena eta argazkia eskatzen dira, 11.45. Irudian ikusten den moduan. Erabiltzaileari mezu elektronikoa bidaliko zaio posta baieztagatzeko eta ondoren pasahitza aukeratuko du.

ProMeta

[Home](#) [View content](#) [Archive](#) [Glossary](#) [Log in](#)

Create new account

First name

Last name

Email address *

A valid email address. All emails from the system will be sent to this address. The email address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by email.

Username *

Several special characters are allowed, including space, period (.), hyphen (-), apostrophe ('), underscore (_), and the @ sign.

Picture

Ningún archivo seleccionado

Your virtual face or picture.
One file only.
64 MB limit.
Allowed types: png gif jpg jpeg.

> [Contact settings](#)

[Create new account](#)

11.45. Irudia. Drupal webguneko kontua sortzeko orria.

Erabiltzaileari pasahitza ahazten bazaio berrezartzeako aukera izango du posta elektroniko bidez. Horretarako, bere erabiltzaile izena edo posta jarri beharko du, 11.46. Irudian ikus daitekeen moduan.

ProMeta

[Home](#) [View content](#) [Archive](#) [Glossary](#) [Log in](#)

[RSS Feed](#)

[Log in](#) [Create new account](#) [Reset your password](#)

[Home](#)

Reset your password

Username or email address *

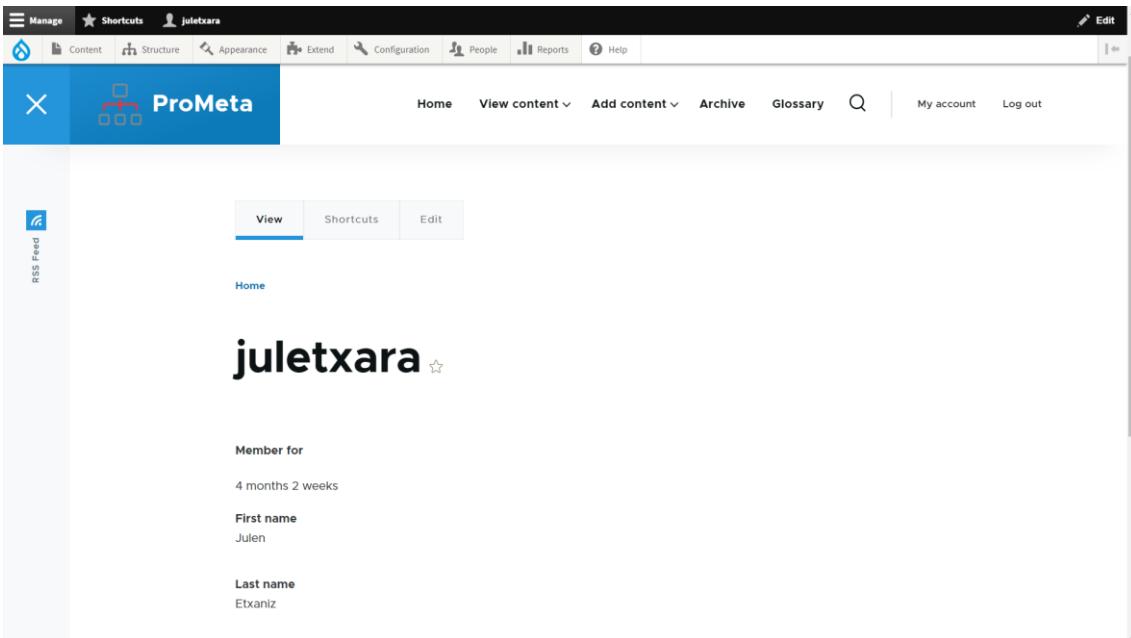
Password reset instructions will be sent to your registered email address.

[Submit](#)

11.46. Irudia. Drupal wenguneko pasahitza berrezartzeko orria.

Saioa irekitzen badugu, gure kontura eramango gaitu. Goiko aldean bi menu berri daudela ikus dezakegu 11.47. Irudian, lehenengo kontuari dagokiona eta bigarrena administrariaren menua. Gainera, menu

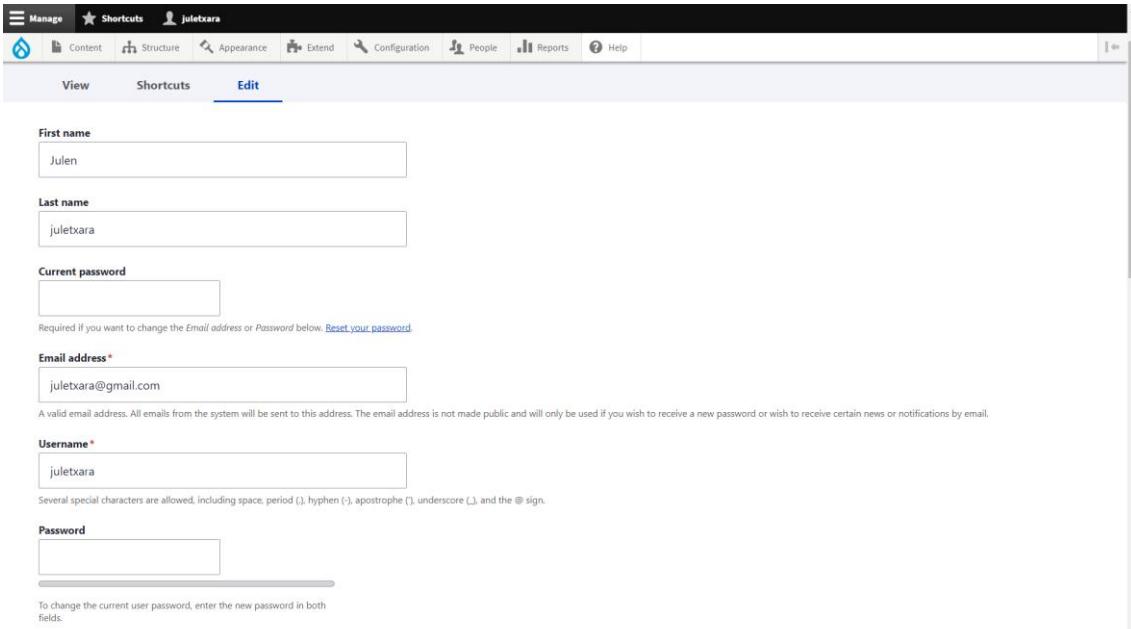
nagusian edukia gehitzeko, kontua ikusteko eta saioa ixteko aukerak agertuko dira. Kontuari dagokionez, lasterbideak definitzeko eta kontua aldatzeko aukerak izango ditugu.



The screenshot shows a Drupal-based web application interface. At the top, there's a navigation bar with links for 'Manage', 'Shortcuts', 'juletxara', 'Content', 'Structure', 'Appearance', 'Extend', 'Configuration', 'People', 'Reports', 'Help', 'Edit' (highlighted), 'Home', 'View content', 'Add content', 'Archive', 'Glossary', a search bar, and 'My account' / 'Log out'. Below the navigation is a blue header bar with the 'ProMeta' logo. The main content area has tabs for 'View', 'Shortcuts', and 'Edit' (which is currently selected). On the left, there's an 'RSS Feed' link. The main content displays a user profile for 'juletxara' with fields for 'Member for' (4 months 2 weeks), 'First name' (Julen), 'Last name' (Etxaniz), and an 'Email address' field containing 'juletxara@gmail.com'.

11.47. Irudia. Drupal webguneko erabiltzailearen kontuaren orria.

Kontua aldatzeko estekan sakatzen badugu itxura desberdina duen orrialde batera eramango gaitu, 11.48. Irudian azaltzen dena. Bertan kontuari dagozkion zehaztasunak aldatzeko aukera izango dugu: izena, posta, erabiltzailea, pasahitza eta rolak.



The screenshot shows a 'User edit' form for the user 'juletxara'. The 'Edit' tab is selected in the top navigation. The form contains fields for 'First name' (Julen), 'Last name' (juletxara), 'Current password' (empty field), 'Email address*' (juletxara@gmail.com), 'Username*' (juletxara), and 'Password' (empty field). There are also notes at the bottom: 'Required if you want to change the Email address or Password below.' and 'Reset your password.', and 'A valid email address. All emails from the system will be sent to this address. The email address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by email.' Additionally, there's a note about special characters allowed in the username.

11.48. Irudia. Drupal webguneko kontua editatzeko orria.

Lehen bezala OpenUP metodologia bistaratzen badugu, aukera gehigarriak ditugula ikus daiteke 11.49. Irudian. Edukia ikustearaz gain, editatzeko, ezabatzeko eta bertsioak ikusteko aukerak dauzkagu.

11.49. Irudia. Drupal webguneko OpenUP metodologiaren orria.

Eduka gehitzeko menu hedagarriaren bidez nahi dugun motako edukia gehitu dezakegu. Adibidez, metodologia aukeratzen badugu, metodologi sortzeko orria bistaratuko da, 11.50. Irudian ikusten den bezala. Eduka gorde aurretik ikusteko aukera ematen du, nola geratuko den jakiteko.

11.50. Irudia. Drupal webguneko metodologia sortzeko orria.

Orain arte aipatutako funtzionalitateak erabiltzaile anonimoak edo arrutak egin ditzakeenak dira. Erabiltzaile administrariak horiez gain webguneko edozein aspektu aldatzeko aukera dauka. Esaterako, rolak sortu ditzake eta erabiltzaileei rolak esleitu, 11.51. Irudian ikus daitekeen bezala.

Name	Operations
Anonymous user	Edit
Authenticated user	Edit
Administrator	Edit
Product Owner	Edit
Trainer	Edit
Technical Writer	Edit

11.51. Irudia. Drupal webgunean rolak kudeatzeko orria.

Rol bakoitzak dituen baimenak ere administrariak kudeatzen ditu, 11.52. Irudian ikusten den moduan. Horrela, nahikoa da rolaren baimenak behin zehaztea eta rol hori hainbat erabiltzaileri esleitzea.

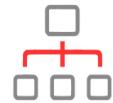
Permission	Anonymous user	Authenticated user	Administrator	Product Owner	Trainer	Technical Writer	Course Developer	Deployment Manager	Deployment Engineer	Project Manager	Any Role	Analyst
Adminster blocks	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>								
Administer comment types and settings	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>								
Administer comments and comment	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>								

11.52. Irudia. Drupal webgunean baimenak kudeatzeko orria.

11.4 Proba

Sistemaren probari dagokionez, proba kasuak zehaztu ez diren arren, sistemak funtzionamendu zuzena duela baieztagoan hainbat egiaztapen egin dira.

XSLT eraldaketa egin aurretik errore pila bat zeuden ereduau. Pixkanaka eraldaketa agindu berriak gehitu dira erroreak gutxitu, bukaeran errore guztiak konpondu arte. XLST eraldaketak ondo funtzionatzen ez



badu, editore grafikoak erroreak ematen ditu. Gainera, erroredun atalak ez dira bistaratzen, zuzen daudenak bakarrik. Beraz, errorerik ez badago eredu osoa bistaratuko da editorean. Horrek esan nahi du eraldaketa ongi egin dela eta eredu metaereduarekin bat datorrela. Gainera, eraldaketa bi metodologia erabiliz egin denez, ziurtasun handiagoz esan dezakegu ongi dagoela.

Editoreen sinkronizazioari dagokionez, antzeko zerbait gertatzen da. Editoreen arteko sinkronizazioak huts egiten badu, beste editoreak errorea emango du. Kasu honetan, editoreen arteko sinkronizazioak ongi funtzionatzen du eredu fitxategi bakarrean dagoenean. Hori bai, eredu hainbat fitxategitan banatuta badago erreferentziak erabiliz, huts egiten du. Erabilitako eredu fitxategi pila batean banatuta dagoenez, sinkronizazioa ez dabil ondo eta ez dira testu editorearen fitxategiak sortzen. Fitxategi berri batean eredu simple bat definitzen badugu, arazorik gabe sinkronizatzen dira editoreak.

Datu-basearen SQL kodea sortzerakoan, konprobatu behar dugu nahi ditugun datu guztien INSERT aginduak sortu direla. Aginduak zuzenak diren edo ez jakiteko, nahikoa da *phpMyAdmin* tresnan importatzeten dugunean errorerik ematen ez duela jakitea. Gainera, datu-basearen taulak definitzerakoan murriztapenak ongi zehazten baditugu, ez digu utzik betetzen ez dituzten datuak gehitzen. Adibidez, PRIMARY KEY erabilita ziurtatzen dugu taulen gakoak desberdinak direla eta FOREIGN KEY erabilita taulen arteko erreferentziak zuzenak direla.

Metodologiaren informazioa aurreko datu-basetik Drupal datu-basera pasatzeko, moduluen bidez mota bakoitzeko edukia gehitu behar da. Beraz, eduki motak ongi definitzen badira, sistemak ez du utzik edukia gehitzen ez baditu beharrezko eremu guziak. Adibidez, datuak ez baditugu ordena egokian importatzeten erroreak agertuko dira, erreferentziatutako elementuak ez direlako existitzen. Ez daukagu konprobatu beharrik datu-basean gorde al diren, webguneko interfazean agertzen badira datu-basean baitaude.

Drupal webguneak proiektuen informazioa gordetzeko balio duela konprobatzeko, proiektu honen informazioa gehitu da. ProMeta izeneko proiektu bat sortu da eta proiektuko kideak eta rolak gehitu dira. Ondoren, ProMeta proiektuaren dokumentazio webgunean dauden OpenUP artefaktu guziak gehitu dira, erabiltzaile bakoitzak berari dagozkionak.

11.5 Hedapena

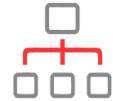
Proiektuaren dokumentazioa eta implementazioa publikoki eskuragarri egongo dira GitHub bidez eta webguneetan. Printzipioz, lana bukatu ondoren ere eskuragarri jarraituko dute, edozeinek kontsultatu ahal izan ditzan.

Proiektuaren dokumentazioaren kidea GitHub-en egongo da eskuragarri: <https://github.com/juletx/ProMeta>. Webgune hori automatikoki eraikiko da aldaketa bakoitzarekin <https://juletx.github.io/ProMeta> GitHub Pages erabiliz. GitHub Pages aukera ona da kasu honetan webgunea estatikoa delako.

Aurreko bi proiekturen webguneekin ere berdina egin dut, ProWF [1] eta BETRADOK [14]. ProWF proiektu honen aurrekaria denez kontsultatzeko behar dut. Eta BETRADOK proiektua antzekoa denez ongi etorriko zait ideiak hartzeko. ProWF proiektuaren errepositorioa <https://github.com/juletx/BETRADOK> eta webgunea <https://juletx.github.io/ProWF/>. BETRADOK proiektuaren GitHub errepositorioa <https://github.com/juletx/BETRADOK> eta GitHub Pages webgunea <https://juletx.github.io/BETRADOK/>.

Proiektuaren metaereduen atalaren implementazioaren kidea ere GitHub-eko errepositorio batean dago: <https://github.com/juletx/ProMeta-ModelEditor>. Kodearen dokumentaziorako webgune bat erabiliko da, aurreko kasuetan bezala GitHub Pages erabiliz <https://juletx.github.io/ProMeta-ModelEditor>.

Prozesuaren webguneak ere aparteko GitHub errepositorioa edukiko du: <https://github.com/juletx/ProMeta-IO-System>. ProWF proiektuaren IO-System ere errepositorio batean jarriko da: <https://github.com/juletx/ProWF-IO-System>.



Dokumentazioarekin egiten den bezala, ondo egongo litzateke webgunea aldaketa bakoitzarekin automatikoki eraikitzea. Edo gutxienez Git-en bidez kontrolatu ahal izatea kode lokala eta zerbitzarikoa. Kasu honetan webgunea dinamikoa denez, beste hosting bat aurkitu beharko da, Drupal-erako balio duena.

Aukeren azterketa sakona egin eta gero, [Pantheon](#) erabiltzea erabaki dut. Honek 3 webgune sortzeko aukera ematen du garapena errazteko: *Development* <https://dev-prometa.pantheonsite.io/>, *Test* <https://test-prometa.pantheonsite.io/> eta *Live* <https://live-prometa.pantheonsite.io/>. *Development* webgunea garapenerako erabiltzen da. *Live* webgunea erabiltzaileek edukia gehitzeko da. *Test* webgunea *Development*-eko hobekuntzak probatzeko erabiltzen da, *Live* webguneko edukiarekin. Webguneak *Test*-en funtzionatzen badu, *Live*-n ere funtzionatuko du.

Estrategia honekin ziurtatzen da nik eta tutoreak uneoro atal bakoitzaren azkenengo bertsioa konsultatu dezakegula. Honek tutorearekin errebisioak egitea errazten du. Gainera, *git* bertsio kontrolari esker egindako aldaketa guztiek ikus daitezke. Horrez gain, webguneak automatikoki eraikitzeak lana errazten du, ez baitaukat zerbitzari batera igotzen ibili beharrik aldaketak dauden bakoitzean.

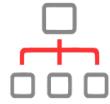
Proiektua amaitutakoan, lana [GAURen](#) matrikulatu behar da eta zuzendariak oniritzia eman behar du. Ondoren, ikasleak lana [ADDI](#) plataforma igoko du. Horrez gain, ikasleak bere lanaren posterra bidali behar du dif.gral@ehu.eus helbidera. Gainera, zuzendariak emandako makinara igoko da lana, proiektu honi jarraipena emateko eskuragarri egon dadin.

Proiektu berriekin domeinu honetan sakondu eta emaitza hobeak lortu ahal izateko, orain arte bezala, proiektu honen emaitzen **jabetza intelektuala** partekatua izango da egile eta tutorearen artean.

11.6 Etorkizunerako Hobekuntzak

Proposatutako sistema prototipo bat denez, oraindik bere kalitate maila igo daiteke. Prototipoaren bidez lortutako emaitzak oinarri bezala hartuta, sistemaren eta proiektuaren hurrengo hobekuntzak proposatzen dira etorkizunerako:

- Editore grafikoaren eta testu editorearen arteko sinkronizazioa hobetzea. Horrela, eredu osoa editore grafiko eta testu editorearekin aldatzeko aukera egongo litzateke.
- Testu editorean edukia automatikoki formateatzea irakurgarritasuna hobetzeko.
- Garapen-prozesua BPMN lengoia estandarra erabiliz definitzea, partekatze eta adoste lanak erraztu eta azkartzeko. Horretarako, metaereduak erabiliz ereduenean arteko eraldaketa egin daiteke automatikoki.
- Bezero ezberdin gehiagoren eskakizunak asetzeko gaitasuna izateko metodologia gehiagoren garapen-prozesuak definitzea, adibidez RUP eta Scrum metodologiak.
- Metodologiez gain proiektuak aurkezteko arauak ere garapen-prozesuan barneratzea, adibidez CCII-N2016-02 araua.
- IO-System azpisisteman rolak eta baimenak hobeto kudeatzea. Rol bakoitzari beharrezko baimenak esleitzea falta da, dagozkion atazak egiteko aukera bakarrik izan dezan.
- IO-System azpisisteman erabiltzaile bakoitzari beharrezkoa den edukia bakarrik bistaratzea. Adibidez, egin beharreko atazak eta bete beharreko artefaktuak.
- Drupal webgunean artefaktu bakoitzerako eduki mota desberdin bat sortzea. Horrela, artefaktu bakoitzaren sekzioak Drupalen bertan defini daitezke eta errazagoa da rolak eta baimenak kudeatzea.



- Drupal webgunean projektuen diagnostikoak egitea eta bistaratzea, egiteko falta dena jakiteko eta proiektuaren egoera ezagutzeko.
- CMMI 2.0 kalitate-ereduaren 2. maila lortzeko garapen-prozesua osatzea.
- CMMI 2.0 kalitate-ereduaren 3. maila lortzeko garapen-prozesua osatzea.

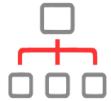
12 Arriskuen Analisia

Atal honetan proiektuan zehar identifikatutako arriskuen inguruko analisia egiten da.

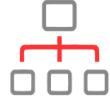
12.1 Arriskuak

12.1. Taulan identifikatutako arriskuen zehaztasunak agertzen dira. Proiektuaren fase guzietan identifikatutako arriskuak zerrendatzen dira eta horien probabilitatea eta inpaktua zehazten da, arazoen magnitudea neurtzeko. Gainera, arriskuei aurre egiteko mitigazio estrategiak definitzen dira.

ID	Data	Izena	Deskribapena	Inpaktu	Probabilitatea (%)	Magnitudea	Mitigazio Estrategia
01	25/02/2021	Lan ingurunearen prestakuntza	Lan ingurunearen instalazioan eta prestakuntzan agertu daitezken arazoak eta denbora galerak.	2	50	1,0	Instalazioan zehar egindakoa dokumentu batean idatzi, instalatuko dudan softwarearen espezifikazioak ondo irakurri.
02	25/02/2021	Lan tresnekin arazoak	Lan tresnak erabiltzerakoan ager daitezkeen arazoak: bertsio bateraezintasunak, pluginak, erroreak...	4	40	1,6	Instalatzerakoan ongi ziurtatu bertsioak bateragarriak direla. Beharrezkoak diren pluginak bakarrik instalatu.
03	25/02/2021	OpenUP ereduarekin arazoak	OpenUP eredua zaharra denez, baliteke guztiz osatua ez egotea eta bateraezintasunak egotea UMA metaereduarekin.	3	60	1,8	Ahalik eta bateragarrienak diren OpenUp eta UMA bertsiaoak aukeratu, ahal bada berrienak.
04	25/02/2021	Xtext-ekin arazoak	Xtext tresnarekin arazoak. Xtext tresnak askotan erroreak sortzen ditu. Normalean erraz konpontzen dira, baina gure eredua oso handia denez, zailagoa izan daiteke.	4	80	3,2	Xtext-ekin lan egitean aldaketak murriztu. Errorrik ez dagoenean lana aurreratzeko aprobetxatu.
05	25/02/2021	Tresnen aukeraketa okerra	Tresna okerra aukeratzeak lana asko atzeratu dezake. Izan ere, honek tresna aldatu beharra edo lana okerrago egitea eragin dezake.	3	50	1,5	Aukeren azterketa sakona egin tresnentzako. Beharrezkoa bada hainbat tresna probatu, egokiena zein den jakiteko.



06	25/02/2021	Memoriaren eta eranskinen arteko koherentzia falta	CCII-2016N-02 eta OpenUP metodologia jarraitu denez, eranskinetan dokumentu asko daude eta koherentzi faltak agertu daitezke dokumentu guztien artean.	1	50	0,5	Eranskin guztiak berrikusi, batez ere, proiektuaren hasieran idatzitakoak. Memoriak eranskinekiko lehentasuna duela zehaztu.
07	25/05/2021	Memoria idazteko denbora falta	Gerta daiteke memoria idazteko denbora nahikoa ez izatea, bukaerarako uzten bada. Oso garrantzitsua da memoria ondo idaztea.	5	50	2,5	Memoria osoa ez utzi bukerarako, pixkanaka kapitulu batzuk idazten joan nahiz eta proiektua bukatu gabe egon.
08	25/05/2021	Proiektua amaitzeko denbora falta	Baliteke proiektua amaitzeako denboraz juxtu ibiltzea. Horrek kalitatea jaistea eragin dezake, lana presaka egiteagatik.	4	75	3,0	Lana modu egokian antolatu eta konstantea izaten saiatu. Denbora aprobetxatu eta gauza garrantzisuenetan zentratu. Ez dago dena bukatu beharrik, etorkizuneko lan bezala utz daiteke.
09	25/05/2021	Egindako lanaren galera	Egindako lana galtzeak lana berregin behar izatea eragin dezake. Honek denbora galtzeaz gain frustrazioa eragiten du.	3	50	1,5	Bertsio kontrola erabili proiektuaren informazio guzta gordetzeko. Datu guztien segurtasun kopiak egin egunero.
10	25/05/2021	Ordenagailuar ekin arazoak	Nire ordenagailuarekin arazoak izateak denbora galtzeak eragin ditzake. Ordenagailua konpondu bitartean lana egin ahal ez izatea eragin dezake.	3	25	0,8	Alternatiba moduan lanerako tutoreak emandako makina birtuala edukitzea.
11	25/05/2021	Makina birtualarekin arazoak	Makina birtualekin arazoak izateak lanerako ingurune hori eskuragarri ez izatea eragin dezake. Nire kasuan ordenagailu pertsonala erabili dudanez ez dauka eragin handirik.	2	50	1,0	Ordenagailu pertsonala erabili garapenerako ingurune nagusi moduan. Makina birtuala alternatiba moduan eduki arazoren bat badago erabiltzeko.
12	25/05/2021	Proiektuaren planteamendu aldaketa	Proiektuan ziurgabetasun handia dagoenez, hurbilpen aldaketak egon daitezke. Honek atzerapenak eragin ditzake.	3	75	2,3	Proiektuaren osagaiak ahalik eta flexibleenak egiten saiatu. Horrela, hurbilpena aldatu arren egindako lana aprobetxatu daiteke.

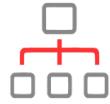


13		Metodologia gehiegia aukeratzea	Metodologia denak edo gehiegia aukeratzen baditugu ezarpena ez bukatzeko aukera daukagu.				Hasteko metodologia garrantzitsuenak bakarrik aukeratu. Ondoren beharrezkoa bada metodologia gehiago gehitu.
14		Metodologia konplexu bakarra aukeratzea	Metodologia konplexu bakarra aukeratzen badugu, lan handia hartuko dugu eta azpiegitura teknologikoa konplexua izango da. Gainera, gerta daiteke simpleagoa behar izatea.				Metodologia garrantzitsu bat baino gehiago aukeratu, simpleak eta konplexuak.
15		Metodologia simple bakarra aukeratzea	Metodologia simplegia aukeratzen bada, azkar bukatuko dugu, baina seguruenik aurrerago konplexuagoak beharko ditugu.				Metodologia garrantzitsu bat baino gehiago aukeratu, simpleak eta konplexuak.
16		CMMI 2.0 kalitate-ereduaren mailak azkarregi lortzen saiatzea	Hasieratik CMMI kalitate-ereduaren 2. edo 3. mailak lortzea handinahia izan daiteke eta arazoak ekar ditzake.				CMMI mailak lortzearen erabakia seguru gaudenean bakarrik hartu.

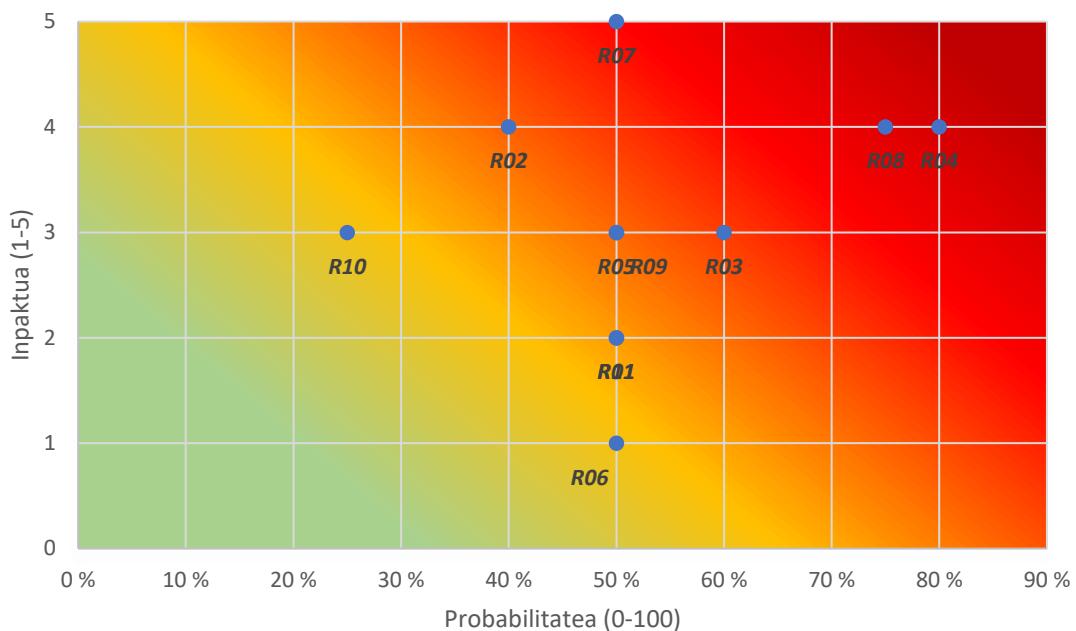
12.1. Taula. Identifikatutako arriskuen zehaztasunak.

12.2 Arriskuen Analisia

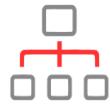
Aurreko atalean arriskuen analisia egiteko arriskuen inpaktu eta probabilitatea bidertzen dira, eta magnitudea kalkulatzen da. Horrela, arrisku garrantzitsuenak zein diren argiago izango dugu uneoro. 12.1. Irudian, berriz, arriskuen inpaktu eta probabilitate grafika batean ikus daiteke. Grafika honek arriskuen ikuspegi orokorra ematen digu, arrisku bakoitzaren larritasuna erakutsiz. Adibidez, arrisku garrantzitsuenak eskuinean eta goian agertzen dira gorriz.



Arriskuen Analisia



12.1. Irudia. Arriskuen inpaktu eta probabilitate analisia.



13 Proiektuaren Antolamendua eta Kudeaketa

Kapitulu honek proposatutako sistemaren proiektuaren antolaketa eta kudeaketa egiteko jarraitu beharreko plana deskribatzea du helburu. Honetarako, *Project Management Institutek* gomendatutako *PMBOK* gida jarraitu da.

13.1 Proiektuaren Antolamendua

Proiektuaren antolaketari dagokionez, garrantzitsua da hasieratik hainbat aspektu argi edukitzea: taldekieen rolak, informazio-sistema eta komunikazio-kanalak.

13.1.1 Taldekideak eta rolak

Proiektu hau Gradu Amaierako Lana denez, ni naiz proiektuaren kide bakarra, Julen Etxaniz. Beraz, nik hartu beharko ditut OpenUp metodologian beharrezkoak diren rol guztiak: Proiektu Kudeatzailea, Analista, Arkitektoa, Probatzailea, Garatzalea, etab. Proiektuaren tutorea Juan Manuel Pikatza izango da.

13.1.2 Informazio-sistema

Sistemaren hedapenean komentatu den moduan, proiektuaren dokumentazioa eta implementazioa publikoki eskuragarri egongo dira GitHub bidez eta webgune bidez. Printzipioz, lana bukatu ondoren ere eskuragarri jarraituko dute, edozeinek kontsultatu ahal izan ditzan. Gainera, amaitutakoan zuzendariak emandako makinara igoko da lana, proiektu honi jarraipena emateko eskuragarri egon dadin.

Estrategia honekin ziurtatzen da nik eta tutoreak uneoro atal bakoitzaren azkenengo bertsioa kontsultatu dezakegula. Honek tutorearekin errebisioak egitea errazten du. Gainera, git bertsio kontrolari esker egindako aldaketa guztiak ikus daitezke, eta egindako lana galtzea saihesten da. Horrez gain, webguneak automatikoki eraikitzeak lana errazten du, ez baitaukat zerbitzari batera igotzen ibili beharrik aldaketak dauden bakoitzean.

13.1.3 Komunikazio-kanalak

Gehienetan etxetik egingo dut lana. Batzuetan EHuko liburutegian ere arituko naiz lanean. Bilera gehienak BBC bidez online egingo ditugu eta beste batzuk presentzialki Informatika Fakultatean. Printzipioz, bilerak astero egingo ditugu aurretik adostutako ordu batean. Beharrezkoa bada eguna edo ordua aldatu dezakegu. Bilerez gain posta elektronikoa erabiliko dugu elkarrekin komunikatzeko.

13.2 Proiektuaren Kudeaketa

Aurretik aipatu den bezala, proiektuaren kudeaketa PMBOK gida-liburuaren arabera egin da. Hori dela eta, liburu horrek aipatzen dituen kudeaketa-motak deskribatzen dira atal honetan.

13.2.1 Integrazioaren Kudeaketa

Atal honen helburua proiektuan zehar ematen diren prozesu desberdinak identifikatu, definitu, konbinatu, bateratu eta koordinatzeko eman beharreko pausuak kudeatzea da.

Proiektuaren hasiera tutorearekin egin beharreko lana adostutakoan izan zen. Lan eskaintza publikoan aukeratutakoa izan zen. Enpresarik ez dagoenez, ez dago bestelako akordiorik sinatu beharrik.

Proiektuaren plangintza eramateko OpenUP metodologiaren Proiektu Plana erabili da, helburuak eta irismena definitzeko. Gainera, CCII-N2016-02 estandarraren arabera, proiekutuko kudeaketako beste esparru batzuetako kudeaketa planak ere egin dira, hurrengo ataletan daudenak.

Projektua zuzentzeko eta kudeatzeko ardura bi aktore garrantzitsuenena izango da: Julen Etxaniz egilea eta Juan Manuel Pikatza tutorea. Projektua epe motzetan zuzentzeko eta kudeatzeko OpenUP metodologia erabili da, hilabete bateko iterazioekin. Iterazioak definitzeko Iterazio Planak erabili dira.

13.2.2 Irismenaren Kudeaketa

Atal honen zeregina da proiektuaren irismena kudeatzea, hots, proiektuaren espezifikazioak eta denbora zeintzuk diren kontuan hartuta, proiektua zenbateraino garatu daitekeen estimatzea eta kudeatzea.

Proiektuaren ikuspegia bete den lehenengo dokumentua izan da. Izen ere, dokumentu horretan konpondu nahi dugun arazoa eta proposatutako produktua deskribatzen dira. Gainera, proiektuko interesatuen erantzukizunak ere zehazten dira.

Proiektuaren betekizunak definitzeko OpenUP metodologiako Ikuspegia eta Betebeharren Espezifikazioa dokumentuak erabili dira. Ikuspegiak betekizun funtzional eta ez-funtzionalak zehaztu dira. Betebeharren Espezifikazioan beste betekizunak definitu dira: kalitatea, interfazeak, konplimentua etab. Erabilpen Kasuen dokumentuek eta ereduek ere betekizunak ulertzten laguntzen dute.

Proiektuaren irismena zehazteko, LDE diagrama definitu da lan-pakete eta ataza nagusiekin. Proiektua lan-paketetan zatitzeko ondorengo atazak egin dira:

1. Emangarriak eta erlazionatutako lana identifikatu eta aztertu.
 2. LDE diagrama egituratu eta antolatu.
 3. LDE diagramako goi mailako elementuak, zehatzagoak diren maila txikiagoko osagarrietan zatitu.
 4. Identifikazio kodea garatu eta LDE diagramako elementuei kode hori esleitu.
 5. Emangarrien deskonposaketa maila egokia dela ziurtatu.

13.2.3 Epeen Kudeaketa

Dokumentu honen xedea da garrantzitsuak diren mugariak identifikatzea, eta proiektuan zehar egon daitezkeen arazoak direla eta epeak aldatu behar badira, gertakizun hori kudeatzeko plana zehaztea.

Atazak zehaztu eta sekuentziatzeko OpenUPeko Lan Atazen Zerrenda erabili da, atazen kudeaketa egiteko aproposak diren datuak zehazten dituen taula. Adibidez, ataza bakoitzaren lehentasuna, tamaina, iterazioak eta denbora estimazioa zehazten da.

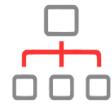
Atazen plangintza egiteko, Gantt diagrama erabili da. Atazen plangintza zehatza egitea oso zaila denez, diagrama honek iterazioko zehaztasuna izango du, hau da, hilabete bakoitzean egingo diren atazak zehaztuko ditu.

Proiektuaren mugarriei dagokionez, iterazio hasiera eta amaiererek eta proiektuaren entregarekin zerikusi duten mugariak zehaztu dira. Gerta daiteke proiektua epe horietan bukatzen denborarik ez izatea eta epea atzeratu behar izatea. Horrek proiektua uztailean aurkeztu ordez irailean aurkeztu beharra eragingo luke. Kontuan hartu behar da behin proiektuaren defentsa eskatzen denean derrigorrez aurkeztu behar dela, eta beraz ekainaren 11rako argi eduki behar dela noiz aurkeztu behar den.

13.2.4 Kostuen Kudeaketa

Atal honetan definitzen dira proiektua aurrera eramateko behar diren kostu ekonomikoak eta horiek nola kontrolatu.

Proiektuan egon daitezkeen kostu ekonomikoak aurreikusiak daudela onartzen da. Hori dela eta, pentsatzen da kostuak ez dutela plangintza eta kontrol berezirik beharko. Ahal den heinean, erabili beharreko softwarea librea izan behar da.



Beraz, estimatzen da tresna aldetik proiektuak ez duela kosturik izango. Hori bai, proiektuko kideen lan orduak kontuan hartuz, proiektuaren kostua zenbatekoa izango litzatekeen estimatzen da. Kostu hauek guztiak Aurrekontuan definitzen dira, doakoak diren tresnak ere zerrendatzu.

13.2.5 Kalitate Kudeaketa

Atal honetan proiektuan zehar helburu diren kalitate betekizunak ahalik eta hoheren gauzatzeko hurrengo jarraitu beharreko prozesua deskribatzen da.

Proiektuaren tamaina eta bereziki, egile bakar batek egindako proiektua dela kontuan hartuta, kalitatea zaintzeko hiru aspektu kontrolatu behar dira:

- **Implementazioaren kalitatea:** kode-mailako implementazioa egokia dela eta software arrunten patroiak jarraitzen direla bermatzeko.
- **Funtzionalitate eta ezaugarrien kalitatea:** produktuak bete behar dituen ezaugarri eta espezifikazioen kontrola.
- **Memoria, dokumentazioa eta aurkezpenaren kalitatea:** proiektuak atxikituta izan behar duen dokumentazio eta memoriaren kalitatea. Adostutako estandarrak betetzen diren (CCII eta OpenUP) ziurtatu behar da, bai eta artefaktu guztiak atal edo sekcio guztiek beteta daudela. Gainera, webgunearren kalitate kontrola egin behar da.

Proiektuko rolen eta kalitate kudeaketako arduren banaketa 13.1. Taulan ikus daiteke. Proiektuaren egileak etengabeen hiru aspektuak kontrolatu beharko ditu proiektuaren bukaerara arte. Proiektuaren tutoreari dagokionez, memoria, dokumentazioa eta aurkezpenaren kalitate-kontrola egiten lagunduko du. Horretarako, astean behin egiten diren bileretan jasotako feedbacka erabiliko da.

Rolak	Ardurak
Egilea: Julen Etxaniz	Implementazioaren kalitatea Funtzionalitate eta ezaugarrien kalitatea Memoria, dokumentazioa eta aurkezpenaren kalitatea
Tutorea: Juan Manuel Pikatza	Memoria, dokumentazioa eta aurkezpenaren kalitatea

13.1. Taula. Kalitate kudeaketako rolak eta ardurak.

13.2.6 Giza Baliabideen Kudeaketa

Atal honen helburua da proiektuan parte hartuko duten aktore guztiak kudeaketa deskribatzea da.

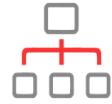
Proiektu hau Gradu Amaierako Lana denez, ni naiz proiektuaren kide bakarra, Julen Etxaniz. Beraz, nik harta beharko ditut OpenUp metodologian beharrezkoak diren rol guztiak: Proiektu Kudeatzailea, Analista, Arkitektoa, Probatzalea, Garatzalea, etab. Proiektuaren tutorea Juan Manuel Pikatza izango da.

Beraz, giza baliabideen kudeaketa nahiko simplea izango da. Egileak rol bakoitzarekin pasatako denbora kudeatu beharko da. Tutorearekin komunikazioa kudeatuko da, postaz eta bileretan.

13.2.7 Komunikazioen Kudeaketa

Komunikazioen kudeaketari begira, atal honetan definitzen dira zein pausu jarraitzen diren kideen artean komunikatzeko.

Ahalik eta komunikazio hoherena edukitzeko lehenengo interesatuak identifikatu dira. Ondoren, komunikazioak planifikatu dira. Komunikazorako bi kanal nagusi erabiliko dira, korreoa eta bilerak.



Korreo bidezko komunikazioa bilera baino lehenago komentatu beharreko aspektuetarako erabiliko da. Komunikazioa EHuko posta edo Gmail erabiliz gauzatuko da. Beraz, nahiko irregularra izango da, baliteke aste batzuetan hainbat aldiz idaztea eta beste batzuetan ezer ez.

Bilerei dagokionez, astean behin egingo dira adostutako orduan BBC bidez. Hala ere, tutorearekin adostuta eguna eta ordua alda daitezke beharrezkoa bada. Bilerak online egitea erabaki dugu Covid19-ren egoeragatik eta ordutegia aukeratzea errazagoa delako. Gainera, pantaila partekatzeko aukera izateak bileretan asko lagundi digu.

13.2.8 Arriskuen Kudeaketa

Atal honen zereginan proiektuan zehar gerta daitezkeen arriskuen identifikatzea, arrisku horiek ebazteko prozesua definitzea eta kontrolatzea da.

Lehenik eta behin, proiektua planteatutakoan arrisku-zerrenda bat egin da OpenUP metodologia erabiliz. Arrisku bakoitzarentzat analisi kualitatibo eta kuantitatibo egin da, inpaktua eta probabilitatea estimatuz. Horrez gain, mitigazio estrategiak definituko dira arriskuei aurre egiteko. Gainera, gerta daiteke proiektua garatu bitartean arrisku gehiago agertzea. Kasu horretan, arriskua zerrendatu eta analisia egingo da.

13.2.9 Erosketen Kudeaketa

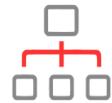
Atal honen helburua proiektuko erosketak kudeatzeko plana zehaztea da.

Kostuen kudeaketan azaldu den moduan, ez da aurreikusten erosketarik egin beharra. Izen ere, erabiliko diren tresna guztiak libreak izatea da helburua. Gainera, ordainpekoak diren software batzuk eskuragarri daude tutoreak eskainitako makina birtualean beharko balira. Hala ere, zerbait erozi beharko balitz egileak kudeatuko luke tutorearen laguntzarekin.

13.2.10 Interesatuen Kudeaketa

Atal honen helburua proiektuko interesatuak identifikatzeko, kudeatzeko eta kontrolatzeko pausuak zehaztea da.

Interesatuak eta horien erantzukizunak proiektuaren hasieran finkatuko dira Ikuspegia dokumentuan. Interesatu garrantzitsuenak kontuan hartzen direla ziurtatu beharko da proiektuaren garapenean. Esan bezala, interesatuen arteko komunikaziorako korreoa eta bilerak erabiliko dira.



14 Denbora Planifikazioa

Atal honen helburua da proiektuaren denbora aurre-planifikazioa azaltzea eta mugarri garrantzitsuak atazten denborak, iterazioak eta desbiderapenak zehaztea da. Denboraren jarraipena egiteko Toggle Track tresna erabili da. Tresna honekin proiektuko denbora guztiak kontrolatu dira, temporizadore bat erabiliz.

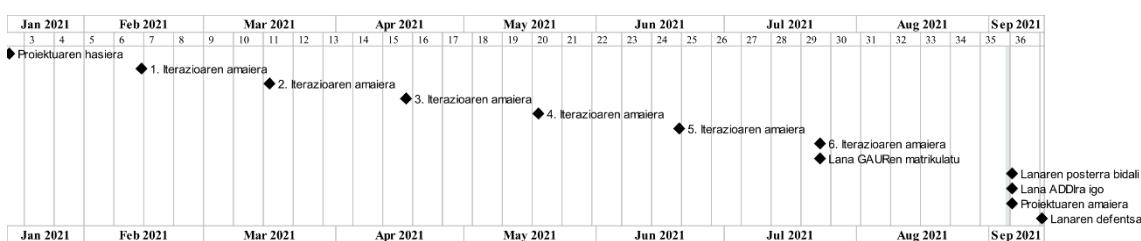
Azpiatazak iterazioka edo beharrezko momentuan sortu dira. Izan ere, oso zaila da aurreikustea ezkjakintasun handia duen proiektu batean zein atazetan egingo den lan hemendik hilabete batzuetara. Ordu kantitatea kontuan hartuko da, baina oso zaila da ataza bakoitzaren denbora eta epeak estimatzea. Horregatik, iterazioen planak eguneratzen joatea garrantzitsua da.

14.1 Mugarriak

Proiektua planifikatzeko garrantzitsua da mugarri garrantzitsuak biltzea eta momentu guztietan argi edukitzea. 14.1. Taulan eta 14.1. Irudian proiektuaren mugarri garrantzitsuenak eta datak azaltzen dira. Gehienek iterazio hasiera eta amaieretik edo entregekin dute zerikusia.

Mugarria	Data
Proiektuaren hasiera	2021/01/14
1. Iterazioaren amaiera	2021/02/14
2. Iterazioaren amaiera	2021/03/15
3. Iterazioaren amaiera	2021/04/16
4. Iterazioaren amaiera	2021/05/17
5. Iterazioaren amaiera	2021/06/20
6. Iterazioaren amaiera	2021/07/23
Lana GAURen matrikulatu	2021/07/23
Lanaren posterra bidali	2021/09/05
Lana ADDIra igo	2021/09/05
Proiektuaren amaiera	2021/09/05
Lanaren defentsa	2021/09/13

14.1. Taula. Proiektuko mugarri garrantzitsuak.



14.1. Irudia. Mugarri garrantzitsuen diagrama.

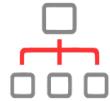
14.2 Lan-atazak

Lana modu egokian antolatu eta kontrolatzeko, ezinbestekoa da lana ataza eta azpiatazeten banatzea. Hala ere, ez da gomendagarria azpiataza gehiegi definitzea, kontrolatzeko eta ulertzeko zailagoa baita. Ataza hauek ez dira estatikoak, proiektua aurrera joan ahala berriak ager daitezke.

Hurrengo taulak adierazten du proiektuan zehar sortutako lan-ataza guztien zehaztasunak adierazten ditu: lehentasuna, tamaina, iterazioak, esfortzu estimazioa, lan orduak eta erreferentzia materiala.

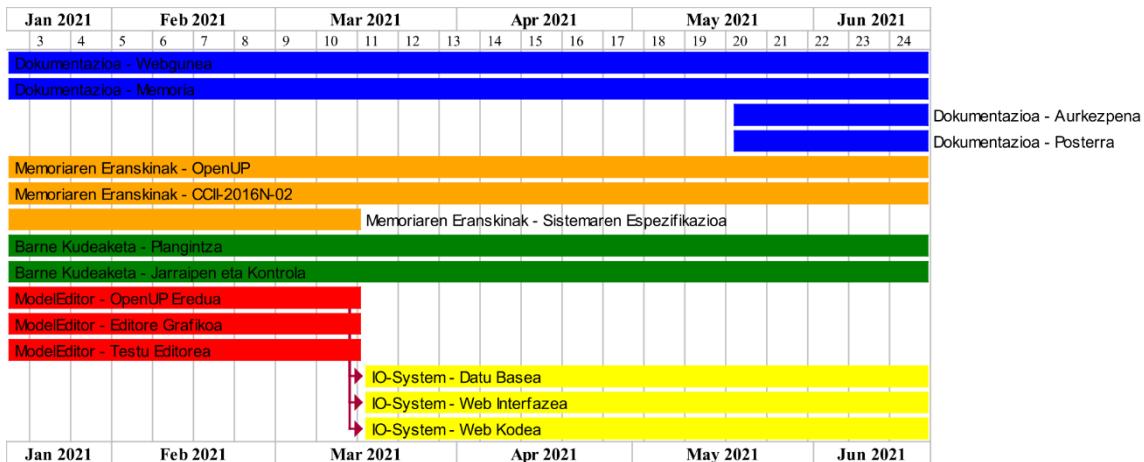
Izena	Lehentasuna	Tamaina	Iterazioak	Esfortzu estimazioa	Lan orduak	Erreferentzia materiala
Webgunea	1	3	I1,I2,I3,I4,I5			CCII-2016N-02
Posterra	2	2	I5			UPV/EHU
Aurkezpena	3	3	I5			UPV/EHU
Memoria	3	5	I1,I2,I3,I4,I5			UPV/EHU, CCII-2016N-02
Memoriaren Eranskinak - OpenUp	1	3	I1,I2,I3,I4,I5			OpenUP
Memoriaren Eranskinak - CCII-2016N-02	1	3	I1,I2,I3,I4,I5			CCII-2016N-02
Sistemaren Espezifikazioa	1	2	I1,I2			OpenUP
Plangintza	3	2	I1			OpenUP
Aurreko Proiektuak Aztertu	2	1	I1			ProWF, BETRADOK
Ingurunea Prestatu	2	1	I1,I2,I3,I4			Eclipse, Drupal
Barne Kudeaketa	1	3	I1,I2,I3,I4,I5			OpenUp
ModelEditor - OpenUp Eredua	1	5	I1, I2			OpenUP, EMF
ModelEditor - Editore Grafikoa	1	5	I1, I2			EMF
ModelEditor - Testu Editorea	1	5	I1, I2			Xtext
IOSystem - Datu Basea	1	3	I3			Drupal
IOSystem - Web Interfazea	1	5	I3,I4			Drupal
IOSystem - Web Kodea	1	5	I3,I4			Drupal

14.2. Taula. Lan-atazen zehaztasunak



14.3 Gantt Diagrama

Gantt diagramaren bitartez atazen garapen denborak zehaztu daitezke. Aurreko ataletan definitutako paketeak eta atazak kontuan hartuz, 14.2. Irudian agertzen den Gantt diagraman iterazio bakoitzeko lanaren estimazioa egin da. Atazak LDE diagraman zehazten direnak dira, eta kolore berdinak adierazi dira.

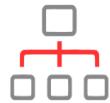


14.2. Irudia. Gantt diagrama.

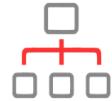
14.4 Iterazioak

Atal honetan, OpenUP metodologiaren bitartez jaso diren atazak iterazioetan zehar nolako banaketa izango duten deskribatuko da. Lan guztia bost iteraziotan banatu da, bakoitzak hilabete inguruko iraupena izanik. 14.3. Taulan iterazio bakoitzaren fasea, helburuak eta datak agertzen dira. Lehenengo bi iterazioak Hasiera fasekoak dira, hurrengo biak Elaborazio fasekoak eta azkena Eraikuntza fasekoak. Helburuak lanatzen antzekoak dira, baina informazio gehiagorekin.

Fasea	Iterazioa	Helburuak	Hasiera	Amaiera
Hasiera	I1	Iterazio Plana. OpenUp Hasiera faseko artefaktuak. Aurreko proiekturen informazioa ulertu. OpenUp txantiloia eskuratu. Memoriaren txantiloia sortu (Fakultateko Arautegiak hemen erabiliko diren metodo eta arauak gomendatzen dutenekin ez dator beti bat. Konponbide batzuk eratuko dira. CCII arauaren informazioa jaso. CCII araua betetzen duen webgunea eraiki. OpenUp metodologiaren informazioa bildu (metaeredua, eredua...). Lan egiteko tresnak eskuratu (Eclipse, EPF Composer, EHSIS, Drupal, etab.).	2021/01/14	2021/02/14



		ProMeta proiektuak sortuko duen produktuak jarraituko duen garapen prozesuaren OpenUp eredua definitu. Metaereduan oinarritutako eredu editore grafiko eta testu editorearen prototipoa.		
Hasiera	I2	Iterazio Plana. OpenUp Hasiera faseko artefaktuak. ProMeta proiektuak sortuko duen produktuak jarraituko duen garapen prozesuaren OpenUp eredua definitu. Metaereduan oinarritutako eredu editore grafiko eta testu editorearen prototipoa. Editore grafiko eta testu editorearen arteko sinkronizazioa. Eedu bera bi modutan editatzeko aukera. Drupal webguneak erabiliko duen datu-basea sortu eredutik.	2021/02/15	2021/03/15
Elaborazioa	I3	Iterazio Plana. Drupal webguneak erabiliko duen datu-basea sortu eredutik. OpenUp Elaborazio faseko artefaktuak. Web interfazea eraiki (Drupal). Web kodea garatu (Drupal). Webgunea. Memoria.	2021/03/16	2021/04/16
Elaborazioa	I4	Iterazio Plana. OpenUp Elaborazio faseko artefaktuak. Web interfazea eraiki (Drupal). Web kodea garatu (Drupal). Webgunea. Memoria.	2021/04/17	2021/05/17
Eraikuntza	I5	Iterazio Plana. OpenUp Eraikuntza faseko artefaktuak. Web interfazea eraiki (Drupal). Web kodea garatu (Drupal). Webgunea zerbitzari batean jarri. Webgunea. Memoria. Posterra.	2021/05/18	2021/06/20



		Aurkezpena.		
--	--	-------------	--	--

14.3. Taula. Proiektuko iterazioen fasea, helburuak eta datak.

14.5 Neurtutako denborak

Toggle Track aplikazioa erabiliz neurtu dira proiektuko denborak. Ataza bakoitzaren denbora neurtu denez, ataza bakoitzeko denbora jakiteko ez daukagu kalkulurik egin beharrik, aplikazioak zuzenean esaten digu. Gainera, nahi dugun denborak bistaratuz ditzakegu, adibidez hilabete bakoitzeko denbora edo aste bateko egun bakoitzekoa.

14.5.1 Denbora hilabeteka

Guztira hilabete bakoitzean pasatako denbora ikus dezakegun hurrengo irudian.

01-10-2020 – 30-06-2021

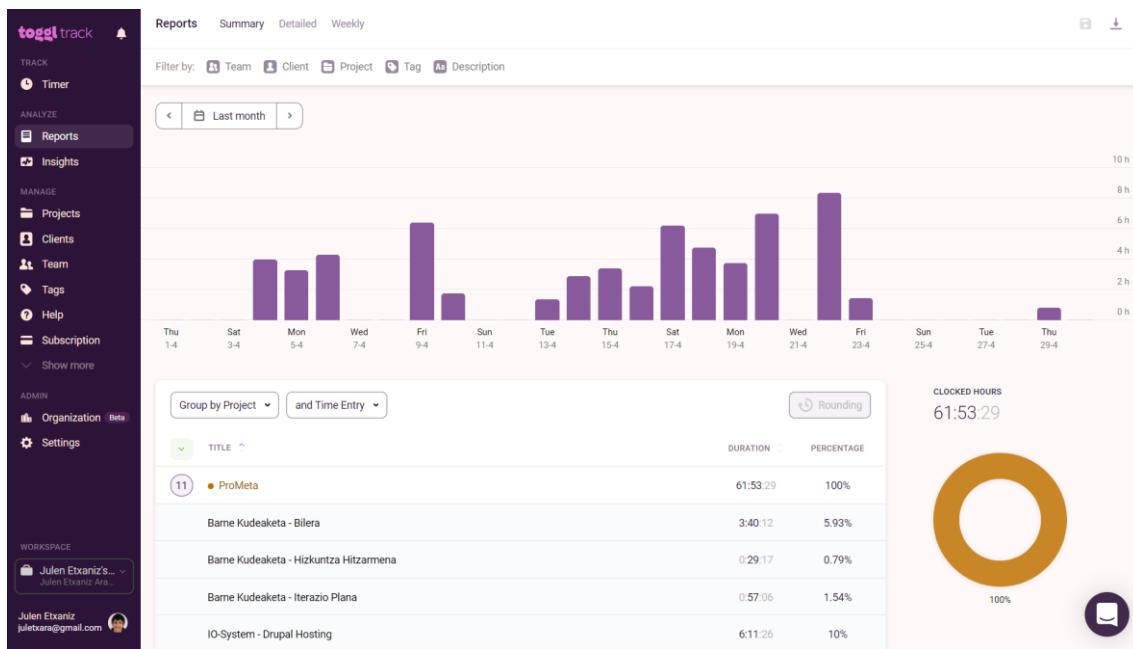
TOTAL HOURS: 293:10:46



14.3. Irudia. Proiektuaren denbora hilabeteka.

14.5.2 Hilabete bateko denbora

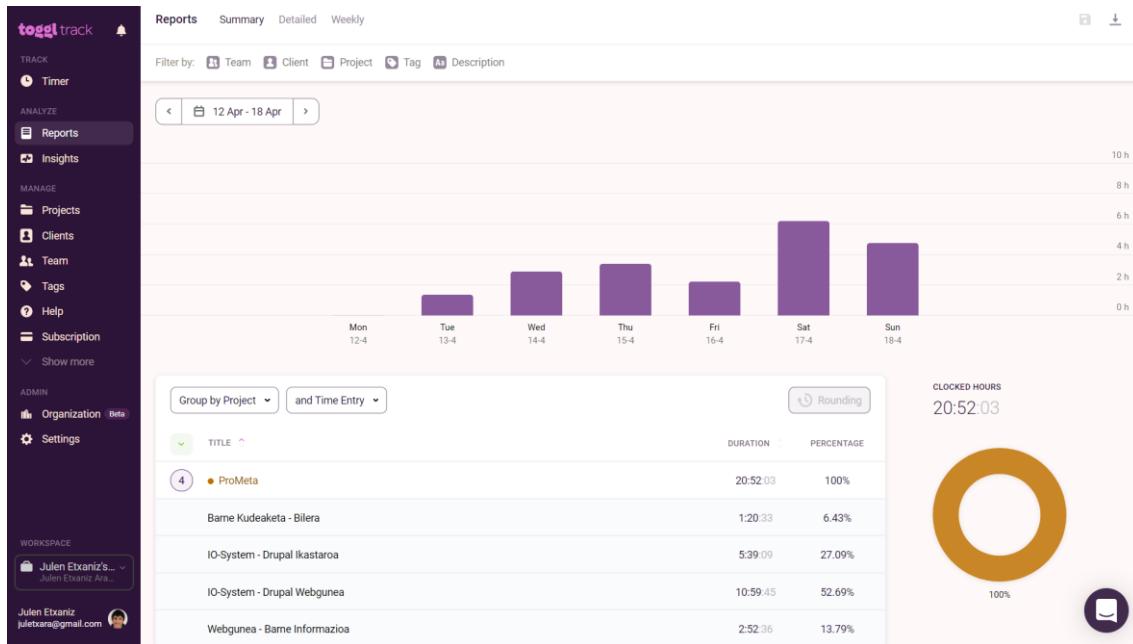
Hilabete bateko egun bakoitzeko denbora ikus dezakegu grafika batean. Gainera, azpian ataza bakoitzari eskainitako denbora eta denbora totala ikus ditzakegu.



14.4. Irudia. Hilabete bateko denbora.

14.5.3 Aste bateko denbora

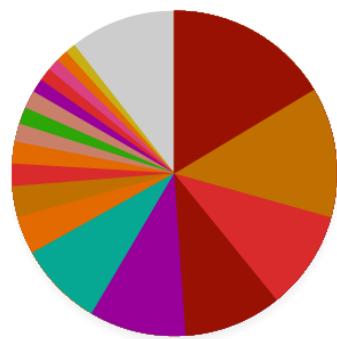
Aste bateko egun bakotzeke denbora ikus dezakegu grafika batean. Gainera, azpian ataza bakotzari eskainitako denbora eta denbora totala ikus ditzakegu.



14.5. Irudia. Aste bateko denbora.

14.5.4 Denbora atazaka

Ataza garrantzitsuenei guztira eskainitako denbora ikus dezakegu hurrengo irudian. Denbora gutxi eskaini zaien atazak grisez multzokatuta agertzen dira. Ataza horien denborak kontsulta daitezke arazorik gabe.



TIME ENTRY

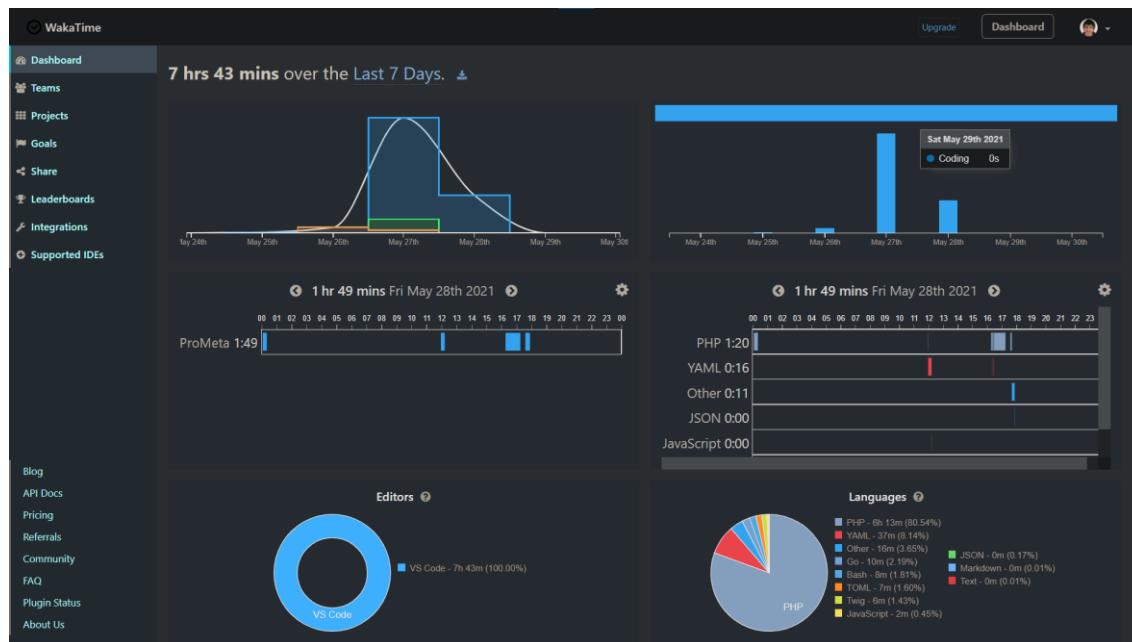
- ModelEditor - OpenUp Eredua
- IO-System - Datu Basea
- ModelEditor - Testu Editorea
- IO-System - Drupal Webgunea
- Barne Kudeaketa - Bilera
- ModelEditor - Editore Grafikoa
- Memoria
- Barne Kudeaketa - Aukeren Azterketa
- IO-System - Drupal Ikastaroa
- IO-System - Drupal Hosting
- Sistemaren Espezifikazioa - Ikuspegia
- IO-System - Aukeren Azterketa
- Webgunea - ProMeta
- Barne Kudeaketa - Proiektu Plana
- IO-System - ProWF
- Webgunea - Egitura Hobetu

	DURATION
48:04:22	
38:00:39	
28:47:26	
28:33:29	
28:05:06	
24:43:14	
10:50:18	
9:00:00	
6:39:09	
6:11:26	
5:21:13	
5:01:00	
5:00:00	
4:00:55	
3:59:01	
3:37:19	

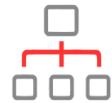
14.6. Irudia. Proiektuaren denbora atazaka.

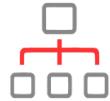
14.5.5 WakaTime

WakaTime tresnak kodetzen denborari buruzko estatistika zehatzagoak eskaintzen ditu. Adibidez, programazio-lengoai, editore eta proiektu bakoitzaren denbora ikus daiteke.



14.6 Desbiderapenak





15 Aurrekontuaren Laburpena

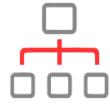
Aurrekontua sortzeko ALI (*Asociación de Titulados Universitarios Oficiales en Informática*) elkartea banatutako irizpide batzuk jarraitu dira. Batetik, giza baliabideen barne-kostuak eta kanpo-kostuak ateratzea bere ordu kopuruekin batera. Gure kasuen ez dago kanpo-kosturik. Bestetik, proiektua garatzeko behar izan diren erreminten kostua kalkulatzea.

Erabilitako erreminta guztiak doakoak izan dira, beraz, arkitektura propria eraikitzearen erabakia egokia izan da. Bizagiren arkitektura erabiliz bere lizenziak eta urteroko mantenuak proiektuaren kostua handituko lukete. Beste alde batetik, ez dira aurkitu erabilitako erreminten premium lizenziariak, baina egotekotan aurrekontuan sartu daitezke, sistemaren kalitatea handitzeko asmoz. 15.1. Taulan proposatutako sistema bideragarria izateko aurrekontua ikus daiteke:

PARTIDA		PARAMETROAK		TOTALA BEZ GABE	TOTALA BEZ BARNE
Giza Baliabideak		Orduak	Kostua		
1	Zuzendaria	0	100	0	0
2	Arkitektoa	0	70	0	0
3	Analista	0	70	0	0
4	Garatzalea	0	50	0	0
5	Probatzailea	0	50	0	0
Totala				0	0
Erremintak		Lizenzia	Mantenua		
1	XAMPP	0	0	0	0
2	Drupal	0	0	0	0
3	VSCode	0	0	0	0
4	PlantUML	0	0	0	0
5	Eclipse	0	0	0	0
6	Git	0	0	0	0
7	GitHub	0	0	0	0
8	Java	0	0	0	0
9	Microsoft Office	0	0	0	0
10	Pantheon	0	0	0	0
Totala				0	0
TOTALA				0	0

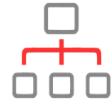
15.1. Taula. Aurrekontua giza baliabideen eta erreminten kostuekin.

Giza baliabideen kostua Ekonomia eta Ogasun Ministerioaren 26/2010 Esparru Akordioan oinarrituta dago. Bezeroari begira software proiektu baten kide bakoitzarentzako erabiltzen dugun baremoa 15.2. Taulan ikus daiteke.



Kidea	Kostua (€/ordu)
Proiektuko zuzendaria	100
Arkitektoa	70
Analista	70
Garatzailea	50
Probatzailea	50

15.2. Taula. Software taldeko kideen kostu baremoa.

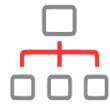


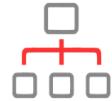
16 Dokumentuen Lehentasun Ordena

Dokumentazio luze honetan inkoherentiak egotea posible izan daiteke. Proiektuaren garapena luzea izan da, dokumentu asko idatzi dira eta gerta daiteke dokumenturen batean agertzen den baieztapen bat kontrajartzea beste dokumentu batean agertzen den baieztapen batekin edo dokumentu batean agertutako datu bat beste batean ezberdina izatea.

Hori dela eta, memoria izango da kontuan hartu beharreko informazioa inkoherentzien kasuan. Memoria dokumentu askoren bilketa da azken finean, baita proiektuaren azkenekoz idatzitako dokumentua. Horregatik, irakurleak memoria konsultatu beharko du zalantzarik izanez gero.

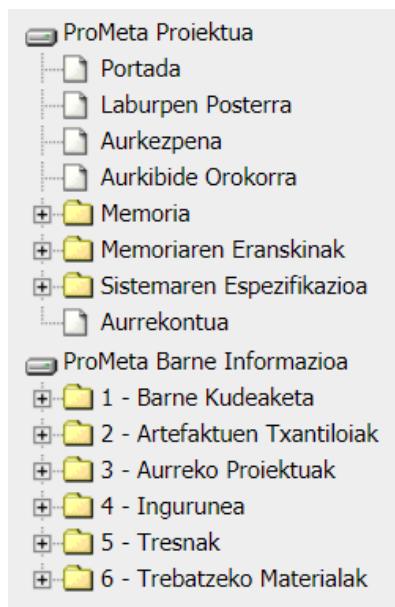
Hala ere, beti prest egongo naiz edozein zalantza edo arazo argitzeko. Nirekin kontaktuan jartzeko posta elektronikoa erabil daiteke: juletxara@gmail.com. Nahiago bada, nire webgunean kontakturako aukera gehiago daude: <https://julenetxaniz.eus/#contact>.





ERANSKINAK

Eranskin guztiak proiektuaren webgunean daude, <https://juletx.github.io/ProMeta/>. Webgune honen helburua ProMeta proiektuaren dokumentazio guztia biltzea eta proiektuko bezero zein interesdunek eskura izatea da. Eranskin batzuetan dokumentuak daude eta besteetan berriz estekak bakarrik. 0.1. Irudian ikus daitezkeen moduan, eranskinak 3 multzotan sailkatzen dira: Memoriaren Eranskinak, Sistemaren Espezifikazioa eta Barne Informazioa. Atal honetan eranskinen azalpen orokorra egingo da irudiek, eranskin bakoitza ikusteko webgunea konsultatu daiteke.



0.1. Irudia. Eranskinak webguneko menuan.

I. Memoriaren Eranskinak

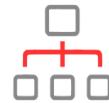
Memoriaren Eranskinak 0.2. Irudian ikus daitezkeen 6 multzotan sailkatzen dira: Sarrerako Dokumentazioa, Analisia eta Diseinua, Tamaina eta Esfortzu Estimazioa, Kudeaketa Plana, Segurtasun Plana eta Beste Eranskinak.



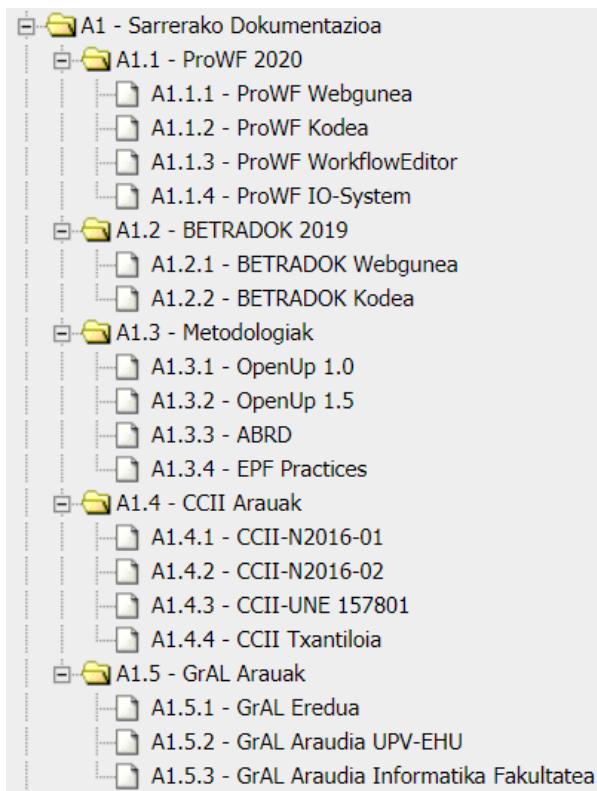
0.2. Irudia. Memoriaren Eranskinak webguneko menuan.

A1 - Sarrerako Dokumentazioa

Sarrerako Dokumentazioaren helburua proiektua eta dokumentazioa gauzatzeko egileak kontuan hartu beharreko dokumentazioa biltzea da. Atal honetan hainbat dokumentu eta webgune esteka daude, 0.3. Irudian ikusten den bezala. Alde batetik, proiektu honekin erlazionatuta dauden aurreko proiektuak



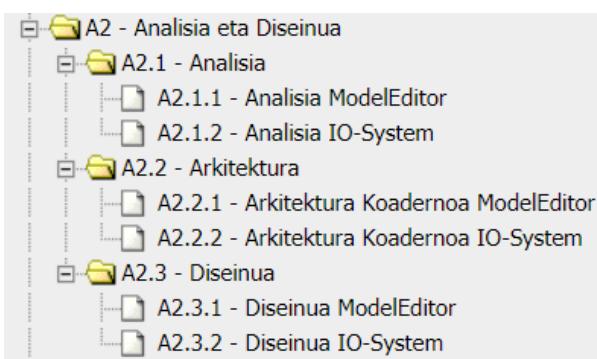
daude, ProWF eta BETRADOK. Bestetik, proiektua gauzatzeko erabilitako metodologia eta arauak biltzen dira, OpenUP, CCII eta GrALarekin zerikusia dutenak.



0.3. Irudia. Sarrerako Dokumentazioa webguneko menuan.

A2 - Analisia eta Diseinua

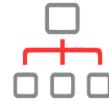
Analisia eta Diseinua eranskinak sistemaren Analisia, Arkitektura eta Diseinua biltzen ditu. Kasu honetan, azpisistema bakoitzenetan dokumentu bana egitea erabaki da, 0.4. Irudian ikusten den moduan.



0.4. Irudia. Analisia eta Diseinua webguneko menuan.

A3 - Tamaina eta Esfortzu Estimazioa

Tamaina eta Esfortzu Estimazioaren helburua da projektuaren tamaina eta esfortzuaren estimazioa azaltzea da. Beraz, mugariak, lan-atazak, iterazioak, LDE eta Gantt diagramak agertzen dira eranskin honetan.



A4 - Kudeaketa Plana

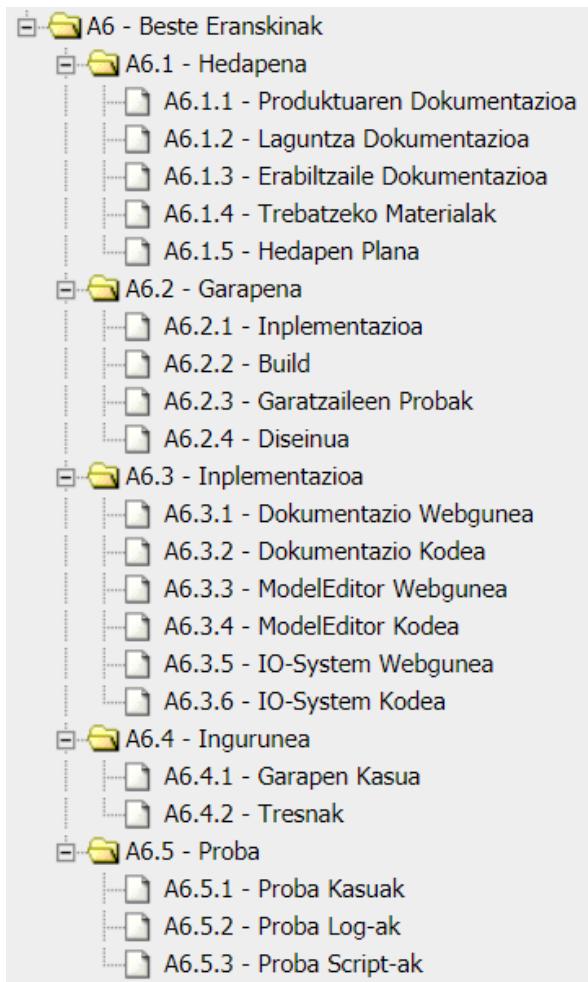
Kudeaketa Plana PMBOK gida-liburuaren arabera egin da. Hori dela eta, bertan aipatzen diren kudeaketa motak deskribatzen dira eranskin honetan: Integrazioaren, Irismenaren, Epeen, Kostuen Kalitateren, Giza Balia bideen, Komunikazioen, Arriskuen, Erosketen eta Interesatuen kudeaketa.

A5 - Segurtasun Plana

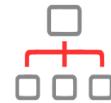
Segurtasun Planaren helburua da proiettuarekin erlazionatutako elementu eta entitateen segurtasuna definitzea eta proiettuan segurtasuna integratzeko neurriak definitzea. Horretarako, hedapen plana kontuan hartzea nahikoa da, bertan definitzen baitira proiettuaren segurtasunari eta konfidentzialtasunari buruzko aspektu garrantzitsuenak.

A6 - Beste Eranskinak

Beste Eranskinetan proiettuan zehar sortu diren beste eranskin batzuk agertzen dira, proiettua gehiago ulertzeko beharrezkoak direnak baina aurreko atalei ez dagozkienak. 0.5. Irudian ikusten bezala, OpenUP metodologiaren artefaktuak daude, 5 multzotan sailkatuta: Hedapena, Garapena, Implementazioa, Ingurunea eta Proba.



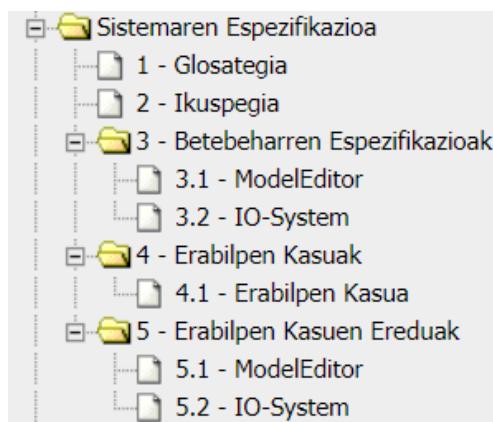
0.5. Irudia. Beste Eranskinak webguneko menuan.



II. Sistemaren Espezifikazioa

Atal honen helburua sistema osoaren espezifikazioak definitzea da, ahalik eta zehaztapen gehien emanez. Proiektauan software arina garatzeko OpenUp metodologia erabiltzen denez, honek zehazten ditu atal honetan egon behar diren dokumentuak, baita dokumentu horiek eduki behar dituzten sekcioak ere. 0.6. Irudian ikusten den moduan, honako dokumentuak daude:

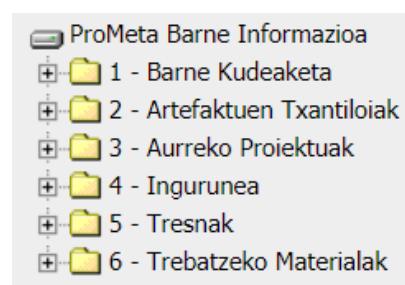
- Glosategia: proiektauan egon daitezkeen akronimo, laburdura edo hitz-berezien definizioak dituen bilduma da.
- Ikuspegia: proiektauren ikuspegi orokorra definitzen du eta proiektauren arazoa kokatzen du.
- Betebeharren Espezifikazioak: azpisistema bakoitzaren betebehrrak espezifikatzen ditu.
- Erabilpen Kasuak: proiektauan dauden erabilpen kasu garrantzitsuenen deskripzio bat biltzen duen dokumentua.
- Erabilpen Kasuen Ereduak: proiektauko erabilpen kasuen ereduko diagrama azpisistema bakoitzerako.



0.6. Irudia. Sistemaren Espezifikazioa webguneko menuan.

III. Barne Informazioa

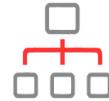
Barne Informazioan proiektauren garapenerako garrantzitsuak diren baina bezeroari bidaliko ez zaizkion dokumentuak gordetzen dira. 0.7. Irudian ikusten den bezala, Barne Kudeaketa, Artefaktuen Txantiloia, Aurreko Proiektuak, Ingurunea, Tresnak eta Trebatzeko Materialak daude.



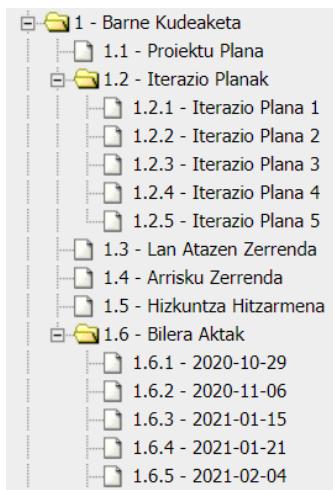
0.7. Irudia. Barne Informazioa webgunearren menuan.

1. Barne Kudeaketa

Barne Kudeaketan proiektauan zehar kudeaketarako erabilitako dokumentuak biltzen dira, 0.8. Irudian agertzen den moduan. Alde batetik, OpenUP metodologiak zehazten dituen artefaktuak daude: Proiektu



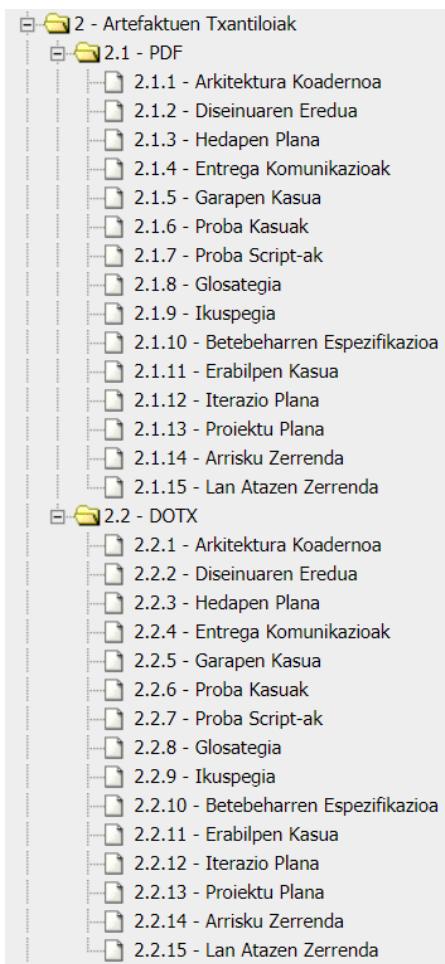
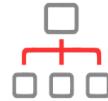
Plana, Iterazio Plana, Arrisku Zerrenda eta Lan Atazen. Bestetik, hizkuntza hitzarmena eta proiektaun zehar egindako bilera guztien aktak daude.



0.8. Irudia. Barne Kudeaketa webguneko menuan.

2. Artefaktuen Txantiloia

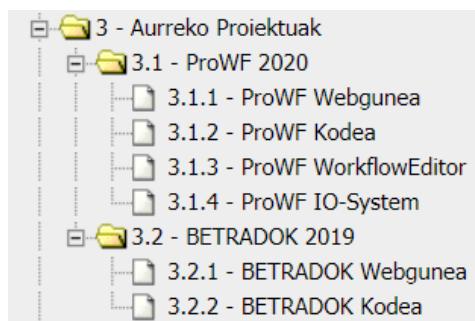
Artefaktuen Txantiloia atalean OpenUP metodologjak eskaintzen dituen txantiloiguztiak biltzen dira PDF eta DOTX formatuetan, beharrezkoa denean eskuragarri edukitzeko. 0.9. Irudian ikusten den moduan, 15 txantiloia daude baina ez dira guztiak erabili.



0.9. Irudia. Artefaktuen Txantiloia webguneko menuan.

3. Aurreko Proiektuak

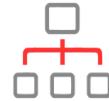
Aurreko Proiektuak atalean erlazionatutako proiektuen kode eta webguneetako estekak daude, 0.10. Irudian ikusten den moduan.



0.10. Irudia. Aurreko Proiektuak webguneko menuan.

4. Ingurunea

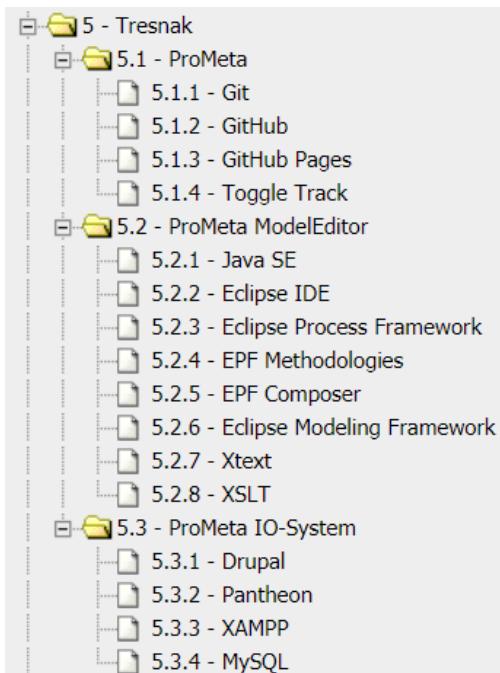
Ingurunean proiektuan jarraitutako garapen prozesuaren inguruko zehaztasunak agertzen dira, 0.11. Irudian agertzen direnak.



0.11. Irudia. Ingurunea webguneko menuan.

5. Tresnak

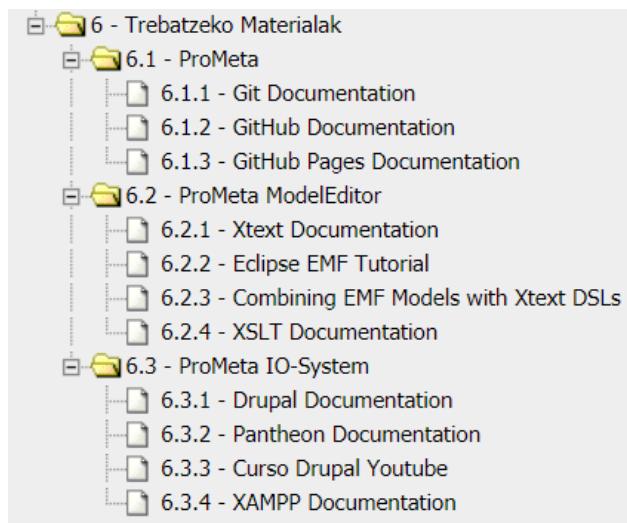
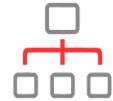
Tresnak atalean proiektua garatzeko erabilitako tresna garrantzitsuen webguneetara eramatzen duten estekak daude. 0.12. Irudian ikusten den bezala, 3 multzotan daude, dokumentazio webguneari eta bi azpisistemei dagozkienak.



0.12. Irudia. Tresnak webguneko menuan.

6. Trebatzeko Materialak

Trebatzeko Materialak atalean erabilitako dokumentazio eta tutorial garrantzitsuetara estekak daude. 0.13. Irudian ikusten den bezala, 3 multzotan daude, dokumentazio webguneari eta bi azpisistemei dagozkienak.



0.13. Irudia. Trebatzeko Materialak webguneko menuan.

IV. Egilearen Kudeaketa

1. Arriskuen Analisia
2. Denbora Planifikazioa
3. Ondorioak