



IKER
GAZTE
NAZIOARTEKO
IKERKETA EUSKARAZ

IV. IKERGAZTE NAZIOARTEKO IKERKETA EUSKARAZ

2021eko ekainaren 9, 10 eta 11a
Gasteiz, Euskal Herria

ANTOLATZAILEA:
Udako Euskal Unibertsitatea (UEU)

INGENIARITZA ETA ARKITEKTURA

**Diseinuaren ezagutzaren biltzea
software artefaktuetan; Produktu
Lerroen Ingeniaritzan
oinarritutako proposamena**

*Xabier Garmendia, Oscar Díaz
eta Maider Azanza*

91-98 or.
<https://dx.doi.org/10.26876/ikergazte.iv.03.11>



Diseinuaren ezagutzaren biltzea software artefaktuetan; Produktu Lerroen Ingeniaritzan oinarritutako proposamena

Garmendia, X.¹, Diaz, O.¹, Azanza, M.¹

Donostiako Informatika Fakultatea
Euskal Herriko Unibertsitatea (UPV/EHU)

xabier.garmendiad@ehu.eus

Laburpena

Ikerketa Softwarea (IS) ikerketa-helburu baterako ikertzaileek garatzen duten softwarea da. Tamalez, ikerketa softwarearen aplikazio gutxi irauten dute behin sortu zituen ikerketa-proiektua amaitu eta gero eta, honen ondorioz, softwarea dakarten ikerketak errepikatzea eta berrerabiltzea zaila bihurtzen da. Askotan, ikertzaileek hutsetik antzeko arazoetarako softwarea eraikitzen dute, gurpila aldiro berrasmatuz. Diseinuaren Zientzia Ikerketa delakoaren testuinguruan, arazo honek Diseinuaren Printzipioek, ikerketa guztiek aurrera egin beharko luketen ezagutzak, hainbat artefaktuen artean sakabanatuta egotea dakar, ezagutza orokor bat galaraziz. Arazo honi aurre egiteko, lan honetan software komertziala garatzeko ondo ezarrita dagoen Produktu Lerroen Ingeniaritza (PLI) IS-motako softwareari aplikatzea aztertuko da.

Hitz gakoak: Ikerketa Softwarea, Softwarearen berrerabilpena, Design Science Research, Software Produktu Lerroak (SPL).

Abstract

Scientific Software (SS) is software that is developed by researchers to fulfill a research aim. Unfortunately, one of the problems associated with SS is that it seldom survives after the research work is concluded, and, as a result, research that involves the development of software becomes difficult reproduce. Researchers usually build software for similar problems from scratch, reinventing the wheel every time. In a Design Science Research (DSR) context, this problem causes Design Principles, sets of knowledge that carry all research forward, to be scattered among various artifacts, preventing a holistic knowledge. To fight this back, in this work we will analyze the application of the Software Product Line Engineering (SPLE) approach, which is well settled in the development of commercial software, to scientific software.

Keywords: *Scientific Software, Software reuse, Design Science Research, Software Product Lines (SPL)*

1. Sarrera eta motibazioa

Ikertzaile baten helburua orain arte landu gabeko ezagutza eta ekarpen berriak aurkitzea da. Ikerketa burutu ostean lortzen den ezagutza berri hori normalean argitaratutako lanen bitartez komunikatzen eta zabaltzen da, adibidez, artikuluen, liburuen edota tesien bidez. Argitaratutako lan hauetan ikerketa prozesuan zehar berretsitako ekarpen berriak adierazten dira, beste ikertzaileei ikerketarako bide berriak sortzeko edota erreferentzia-puntu bezala erabiltzeko balio dutenak.

Tamalez, argitaratzen diren lan asko ez dira errepikakorak. Hau da, bertan argitaratutako emaitzak ezin dira berretsi ikerketa prozesuan zehar erabilitako baliabideak askotan ez direlako partekatzen argitalpenarekin batera (Chen *et al.*, 2019). Argitalpena, erabilitako datuak, eta jasotako emaitzak eskutik joan behar dira ikertzaileek egindako lanak beste ikertzaile batzuen eskuetan ere emaitza berdinak lortzen dituela frogatzeko. Gainera, baliabideen gabeziak, emaitzak egiaztatzeaz gain, erabili eta sortu diren baliabideen berrerabilpena ekiditen du. Ingeniaritzako arlo askotan ikerketa prozesuan arazo honi protagonista berri bat gehitzen zaio. Ingeniaritza ikertzaileek sarri ikerketa prozesuan zehar software berriak garatzen dituzte haien ikerketa esparruko arazoei aurre egiteko eta hauek artikulatu batekin batera argitaratuak izan ohi dira haien ikerkuntzaren emaitza gisa (von Krogh eta Haefliger, 2010).

Ikerketa Softwarea (IS), software komertziala izan beharrean, testuinguru akademikoan ikertzaileek ikerketa helburu baterako garatzen duten softwarea da. Ikerketa softwareak mota desberdinekoak izan daitezke, horien artean denbora luzez garatuak izan diren software pakete handiak edota idatzitako kode lerro gutxi dituzten programa txikiak aurkitu daitezkeelarik (Hettrick, 2016).

Ingeniaritzako arloetan ez ezik, beste ikerkuntza-arlo desberdinetan software baten garapenaren beharra ere eman daiteke. Adibide bezala, geografiari buruzko ikerketetan ikertzaileek softwarea garatzen dute aplikazio desberdinetarako. Sistema hauen proposamenen artean ikertzaileak aldaketa geofisikoen jarraipen zehatzagoa egiteko softwareak eta analisi hobeak egiteko algoritmo geometrikoak aurkitzen dira (Steiniger eta Bocher, 2009). Ekonomiaren arloan berriz, ikertzaileek, nahiz eta programazioari buruz jakintza handirik ez izan, askotan software paketeak garatzen dituzte analisi estatistiko eraginkorrak aurrera eraman ahal izateko (Yalta eta Yalta, 2010). Beraz, ingeniartzako arloetan gain, beste eremu batzuetan ere aurkitu dezakegu ikerketa softwarea.

Ikerketa lanetan sortzen diren softwarearen partekatzeak eta berrerabilpenak ikerketa diziplina bakoitzaren eraginkortasuna bultzatu dezake (von Krogh eta Haefliger, 2010). Zoritzarrez, ikuspegi hori lortzetik urrun dago. Ezagutzaren eta ikerkuntza softwarearen berrerabilpena gutxi landuta dagoen gai bat da. Eta arazo honen ondorioak areagotu egiten dira ikerketa eremua *Software Ingeniaritza (SI)* bada, non garatutako softwarea ikerketaren emaitza funtsezkoa den. Hemen aurkeztutako lan honek software ingeniartzako testuinguruan ezagutzaren eta ikerkuntza softwarearen gaineko berrerabilpena bultzatzen duen proposamen bat aztertuko du, berrerabilpena buruan duen ikerkuntza bat sustatuz.

2. Arloko egoera eta ikerketaren helburuak

Ikerketa softwarea ikerketa-prozesuen ezaugarri diren irekitasun (*openess*), biziraupen (*survivability*), errepikakortasun (*reproducibility*) eta pertsonalizatzeko ahalmen (*customizability*) printzipioek gidatu beharko lukete (Johannesson eta Perjons, 2014). Zoritzarrez, praktikan zaila da ikuspegi hori aurrera eramatea. Ikerketa softwarearen aplikazio gutxi irauten dute behin sortu zituen ikerketa proiektua amaitu eta gero (adibidez, doktorego-tesi bat). Software Ingeniaritzan, doktorego-denboraren %60 arte softwarearen garapenean aritzeak ahalegin inbertituaren neurriaren parte bat ematen du, eta ez da beti behar bezain errentagarria. Zehazki, zaila da Softwarearen Ingeniaritzako ikerketa errepikatzea, zaila baita ikerketaren emaitza bezala lortzen den ikerketa softwarea autoreak ez diren ikertzaileek abian jartzea.

Izan ere, 2020ko amaieran, Nazioarteko Software Ingeniaritza Kongresuan (*International Conference on Software Engineering - ICSE*) ikerketa prozesuan software bat garatu dutela aipatzen duten 800 lanen gainean burututako azterketa batek erakutsi zuen soilik %58ak publikoki eskuragarri uzten duela, lanaren erreplikazioa, errepikapena, hedadura eta egiaztapena galaraziz (Heumüller *et al.*, 2020). Honen ondorioz, egungo egoeran ISa ez da bakarrik errentagarritasun gutxikoa, gainera, Softwarearen Ingeniaritzako eta Informazio Sistemetak ikerketaren errepikagarritasuna arriskuan jartzen du. Lan honetan, gai hori Diseinuaren Zientzia Ikerketa (*Design Science Research - DSR*) delakoaren testuinguruan landuko dugu, eta gailu terminoa erabiliko dugu ISa aipatzeko.

DSR Informazio Sistemen eta Informazioaren Teknologia alorretan jatorria duen ikerketa metodologia da. Hala ere, gaur egun beste ikerketa arlo desberdinetara ere zabaldu da, adibidez, medikuntzaren ingeniartzan, bioteknologian eta arkitekturan (von Krogh eta Haefliger, 2010). DSR azterketa zientifikoaz eta gailuak sortzeaz arduratzen da, eta interes orokorreko arazo praktikoak konpontzea du helburu (Johannesson eta Perjons, 2014). Gailu baten garapena ez da, ordea, halabeharrez DSR ikerketa bat. "Ikerketa" izateak esan nahi du gailua orokortu daitekeela hasieran arazoa sortu zuen testuingurutik kanpo. "Ikerketa" bihurtzeko soluzioa orokortzeko behar honek bi faktoreren arteko oreka bilatu beharra dakar. Alde batetik, gailuak ebaluatu ahal izateko adinako xehetasun maila izan behar du. Bestetik, gailua orokortu egin behar da ebaluazioaren konkretutik "ikerketak" eskatzen duen orokortzera pasatzeko (Winter, 2008). Beraz, eta gailu komertzialak ez bezala, DSR gailuak ez dira berez helburu bat, baizik eta *Diseinuaren Printzipioetan (DP)* aurrera egiteko bitarteko bat baizik, hots, arazoaren eta konponbide-espazioaren arteko erlazioetan aurrera egiteko bitartekoak. DPak irizpide orokorrak dira, gailuen diseinua jatorrian aztertutako gain bestelako testuinguruetan erabiltzea baimentzen dutenak, eta, horren bidez, haien proiektzioa (hau da, aplikazio-irismen zabalagoa dutenak) eta konfiantza (hau da, ebaluazio sendagoak dituztenak) zabaltzen dituztenak (Vom Brocke *et al.*, 2020). Gailuak proiektzio eta konfiantza handiagoko testuinguruen aurrean jartzean, DPak orokorrak eta sendagoak bihurtzen dira. Beraz, DPak etengabeko fluxuan dauden ezagutza bat dira hainbat testuingurutan probatzen diren heinean.

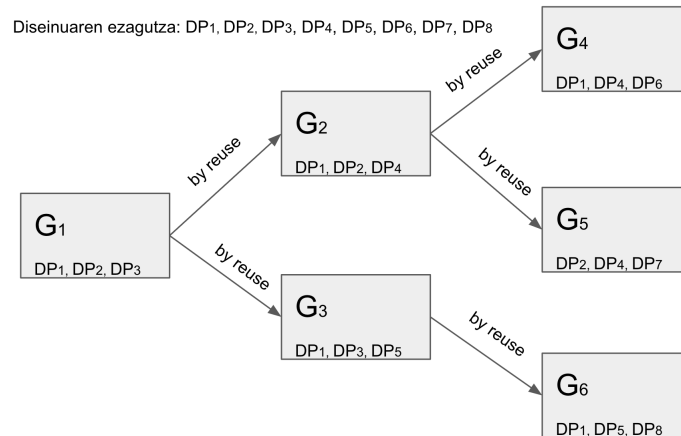
DPak funtsean ebolutiboak badira, azpiko gailuek ere hala izan beharko lukete, DPek aurrera egiteko beharrezko bitartekoa baitira. Vom Brocke *et al.*-ek ikuspegi hori lantzen dute bi egokitzapen mota desberdinez. Batetik,

erabilerarako egokitzapena (hau da, gailuak bere funtzioa egungo testuinguruan egiteko duen gaitasuna) eta, bestetik, **eboluziorako egokitzapena** (hau da, gailuak denboran zehar problemaren espazioan gertatzen diren aldaketetara egokitzeko duen gaitasuna) (Vom Brocke *et al.*, 2020). Hala ere, autore berek onartzen dute, orain arte, “ikasketa” gehienak DSR proiektu bakar batean oinarritzen direla, proiektu horren barruan DPak lortzeko; aldiz, zenbait proiekturen bidez egindako ezagutza- eta eboluzio-bilketa gutxitan hartzen da proiektuaren aurrekari edo ekarpenzat (Vom Brocke *et al.*, 2020). Horren ondorioz, ikerketa softwarea ez da berrerabiltzen ez egokitzen.

Ikertzaileek softwarearen garapenaren kasuan, sarritan garapen-erabaki suboptimoak hartzen dituzte, ebaluaziora azkar iristeko. Ebaluaziorainoko denbora merkataritza munduko merkaturainoko denboraren baliokidea da. Azken kasuan, garapen-erabaki suboptimoek zor teknikoa (*technical debt*) deritzona eragin lezakete, hau da, garatzaileek irtenbide azkar baten bila hartzen dituzte software-garapenaren erabakiak inplementazio-denbora murrizteko (Lenarduzzi *et al.*, 2019). Laburbilduz, erabilerarako egokitasunari lehentasuna emateak, eboluziorako egokitzapena arriskuan jar lezake. Hau bereziki kaltegarria izan daiteke DSR proiektuetan, eboluziorako egokitzak beste ikertzaile batzuek gailuak berrerabiltzea eta eskuratzea eragotzi baitezake, eta, beraz, DSR ikerketaren metatze- eta esplorazio-izaera oztopatu. Arazoaren ondorioz jabetzen gara, baina galderak bere horretan jarraitzen du, ikerketa softwarean berrerabilpena ezberdina den software komertzialaren berrerabiltzetik. Garapen horietako bakoitzaren helburuetan aurki daiteke erantzuna.

Softwarearen berrerabilpenak dituen onura komertzialen artean daude softwarearen kalitatea, kostua eta produktibitatea hobetzea (Frakes eta Kang, 2005). Onura hauek ikerketa softwarearako ere balio dute. Hala ere, ikerketa softwarearen azken helburua uneko DPetan aurrera egitea izan beharko luke. DSR gailuak ez dira arrakastatsutzat jotzen DPak finkatzen ez badituzte, haien kalitatea, kostua edo produktibitatea kontuan hartu gabe. Zorritzarez, ikerketa-komunitateak ez du esaten duena egiten: nahiz eta ikerketa asko egin softwarea berrerabilpenari buruz, DSR gailuen artean ez dago berrerabilpenik edo, batez ere, kopiatu-bereganatu (*clone&own*) (Vom Brocke *et al.*, 2020) teknikaren bidez lortu da. Kasu horretan, produktu berri bat lehendik dagoen produktu bat kopiatuz hasten da, eta, ondoren, haren zatiak egokitzen dira baldintza berriak betetzeko. Epe laburrera kostuak aurrezten dituen arren, kopiatu-bereganatzeak gailu desberdinak sortzea dakar testuinguru bakoitzarentzat, hau da, DPen ulermen holistikoa emateko ahalegina atzeratu egiten da. Ondorioz, kopiatu-bereganatu praktikak “arrisku-jardueratzat” har daitezke DSR ikuspegitik, DPak hainbat gailutan sakabanatuta egongo baitira (ikus 1. irudia).

1. Irudia: Kopiatu-bereganatu teknikaren bidez sortutako gailuetan garatutako Diseinu Printzipioen sakanbanapenaren irudikapen bat.



DSR munduan, diseinuaren printzipioen zehaztapena eta egingarritasun-proba da gailua. Hori dela eta, gure helburua G_1, G_2, \dots, G_n gailuak ez bereizi gisa hartzea baizik eta produktu-lerro bereko partaide bezala hartzea da, “ikerketa-eremu” bera partekatzen dutelarik. Ikuspegi honetan, ezagutzaren gordailuzainak ez dira argitalpenak soilik, baizik eta DSR ikuspegiarekin bat etorritz, baita gailu berri bat ere, produktu-lerroa, zeinak biltzen baititu komuna dena eta aldakorra dena, eta, horren ondorioz, ikertzaileek “ikerketa-eremu” bera lantzeko orduan aurkitu dutena testuinguru, interesatu edo helburu desberdinak kontutan izanda. Beste era batera esanda, produktu-lerro bat diseinuaren printzipioekiko da literaturaren berrikuspen bat argitalpenekiko dena: ikerketa-gai jakin batean egindako lanaren laburpena, kasu honetan DSR erabiliz. Literaturaren berrikuspenak sistematikoak badira, gailuen berrerabilpenak ere sistematikoak izan behar du. Horretarako, garapena produktu-lerroen ezagutzan oinarritu daiteke.

Proiektu honen premisa nagusia da diseinu printzipioen bilketari mesede egin dakiokela gailuak berrerabil-tzeak. Zehazki, hau proposatu nahi dugu:

“Produktuen Lerroko Ingeniaritzaren ikuspegia, DSR gailuak garatzeko ikuspegi berrerabilgarri eta metagarri bezala Diseinu Printzipioetarako”

Produktu Lerroen Ingeniaritzak (PLI) softwarea berrerabiltzeko estrategia sistematikoa ezartzen du. Estrategia horren helburua domeinu baten barruan zati komunak (funtzionaltasun komuna) eta aplikazioen arteko aldagaiak identifikatzea da, aktibo berrerabilgarriak eraikiz, etorkizuneko garapen ahaleginak errazteko (Apel *et al.*, 2013). Bai zati komunak bai aldakortasuna ezaugarrien (*feature*) nozioaren bidez zehazten dira, hau da, erabiltzaileak ezagutu dezakeen produktuaren portaeren bidez. PLIk produktua konfiguratzeko ezaugarriak abstrakzio nagusi bezala hartzen ditu kontuan, bezero baten betekizunetatik dagozkien funtzionalitatea ematen dituzten software gailuetaraino lotura egiteko erabiltzen baitira. Beraz, softwarearen bizi-zikloan zehar gailuek eta prozesuek duten oinarria ezaugarriak dira (Apel *et al.*, 2013). PLIak garatzeko, elkarrekin lotutako bi prozesu bereizten dira:

- **Domeinuaren Ingeniaritza**, hau da, domeinua aztertzeke eta berrerabiltzeko plataforma garatzeko prozesua. Problemaren espazioan, prozesu horrek softwarearen produktu lerroaren irismena lantzen du, baita kasuan kasuko domeinuaren barruan aldakortasuna eta zati komuna identifikatzea ere. Soluzioaren eremuan, domeinuaren ingeniaritzak berrerabilpenerako (*for reuse*) garapena lantzen du, hau da, berrerabiltzeko diren software-osagai edo -zatiak.
- **Aplikazioaren Ingeniaritza**, hau da, berrerabiltzeko plataformaren aktiboetatik abiatuta produktu bat eraikitzeke prozesua. Arazoaren eremuan, bezero jakin baten beharrak aztertzeke baldintzen azterketa sartzen da. Soluzioaren eremuan, aplikazio-ingeniaritzaren helburua da produktu zehatzak lortzea, ez hutsetik, baizik eta berrerabiltzeko plataforman oinarrituta. Berrerabilpenean oinarritutako garapen gisa laburbiltzen da (*by reuse*).

Testuinguru horretan, hauek dira proiektuaren helburuak:

- **Domeinuaren Ingeniaritza birformulatzea ikerketa gailuetarako.** Zer esan nahi du “ikerketa-eremua” izateak? Zer teknika erabili daitezke “ikerketa irismenatarako”? Zer ikerketa bide egin behar lituzke produktu-lerroak? Nola islatzen da ikerketa helburuen lehenespena produktu lerroaren eremuan?
- **Aplikazioaren Ingeniaritza birformulatzea ikerketa gailuetarako.** Eta produktuak ezin badira lortu aktibo berrerabilgarrietatik? Eta produktua pertsonalizatzea ez bada salbuespena mundu komertzialean ez bezala?
- **Nola motiba daitezke ikertzaileak programatzaile-lanetan berrerabilpenerako garatzeko?** DSRrentzat gailuak ezinbestekoak dira, baina ez dira zertan berrerabilgarriak izan behar. Berrerabil daitezkeen gailuak sortzeak kostu gehigarriak ditu, haien birfaktorizazioa egin beharra adibidez. Irabaziak, ordea, ez dituzte beti lortzen kostuak ordaintzen dituztenek. Ikerketa maiz neurtzen da argitalpenei dagokienez. Oinarritzko gailuaz haratago edozein lan egiteak, ikerketa taldearentzat onuragarria izan arren, beste argitalpen batetik denbora ken lezake. Horretarako, politika egokiak behar dira, artikuluen egileetan edo eskerroneko aipamenetan gailuen berrerabilpena saritzeko.
- **Nola motibatu daitezke ikertzaileak, programatzaile-lanetan, berrerabilpenaren bidez garatzera?** Softwarea eskura izateak ez du esan nahi nahitaez erabili behar denik. Batzuetan, softwarea aldatzea eta egokitzea garestiagoa izan daiteke zerotik behar den funtzionaltasuna programatzea baino. Testuinguru horretan, ikertzaileak beste pertsona baten kodea berrerabiltzeari uko egin diezaioke, adibidez, kalitatezko dokumentaziorik ez badago. Nola handitu daiteke konfiantza?

3. Ikerketaren muina

Lan honetan proposatutako ideiak, DSR testuinguru batean, Produktu Lerroen Ingeniaritzaren ikuspegia aplikatzea du helburu. Honen bidez lortu nahi duguna zera da: DSR metodologia jarraituz garatu diren gailuen berrerabilpen sistematikoa sustatzea, hauen arteko DP bilketa ahalbidetzeko.

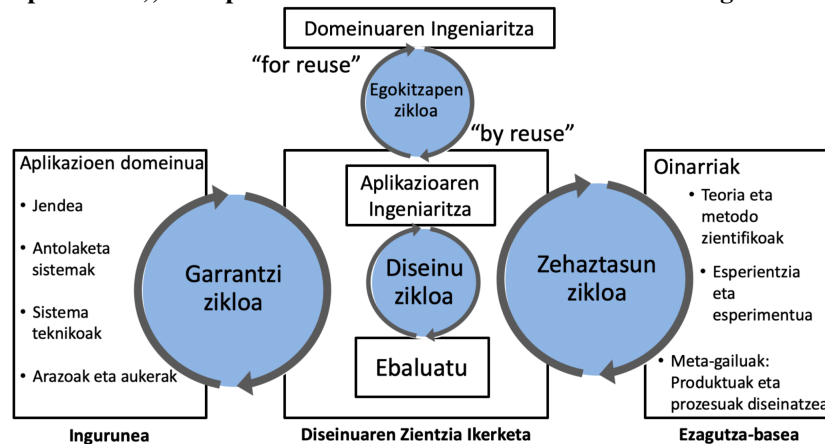
Hau lortzeko ezinbestekoa da DSR eta Produktu Lerroen Ingeniaritza aurrera eramaten duten prozesuak bateratzea. Alde batetik, aurreko atalean aipatu den bezala, PLI bi prozesuk osatzen dute: Domeinuaren Ingeniaritza eta Aplikazioaren Ingeniaritza. DSR metodologia aurrera eramateko ohiko prozesuak, berriz, Hevner autoreak 2007

lanean proposatutako hiru zikloko ikuspegi batek osatzen du, Garrantzi-zikloa (*Relevance Cycle*), Diseinu-zikloa (*Design-Cycle*) eta Zehaztasun zikloa (*Rigor Cycle*) biltzen dituenak.

- *Garrantzi-zikloa (Relevance Cycle)* DSR abiarazten du, eta praktika jakin baten barruan landuko den arazoa identifikatu eta aztertzen du. Arazoak interes orokorrekoa izan behar du, eta arazoaren balizko kausak azpimarratu behar dira. Ziklo honetan, arazoaren pertzepzioa ez ezik, soluzio bat onartzeko irizpideak ere lantzen dira.
- *Diseinu-zikloa (Design Cycle)* aurreko zikloan identifikatutako onarpen-irizpideen arabera arazoa konpontzen duen gailu baten diseinua eta garapena lantzen ditu.
- *Zehaztasun-zikloa (Rigor Cycle)* proiektuaren emaitzak literatura akademikoaren barruan kokatzen ditu, eta hartutako diseinu-erabakiak bermatzen dituzten oinarri zientifikoak ematen ditu. Zehaztasun-zikloak bermatu behar du sortutako diseinuak ikerketa-ekarpenak izatea, eta ez prozesu ezagunen aplikazioan oinarritutako ohiko diseinuak.

Lan honetan proposatutako ikuspegia aurrera eramateko asmoz, DSR metodologiaren hiru zikloko ikuspegia duen prozesuari Domeinuaren Ingeniaritza eta Aplikazioaren Ingeniaritza prozesuek barnertzen dituzten ideiak gehitzea proposatzen dugu, hauek biltzen dituen laugarren ziklo berri bat DSR prozesura txertatuz (Diaz *et al.*, 2021): Egokitzapen zikloa (*Fitness Cycle*) (ikus 2. irudia).

2. Irudia: Hevnerrek proposaturiko hiru zikloko ikerketa prozesuaren hedapena laugarren ziklo baten bitartez (Egokitzapen zikloa), DSR proiektu deberdinen arteko diseinuaren ezagutza sustatzeko.



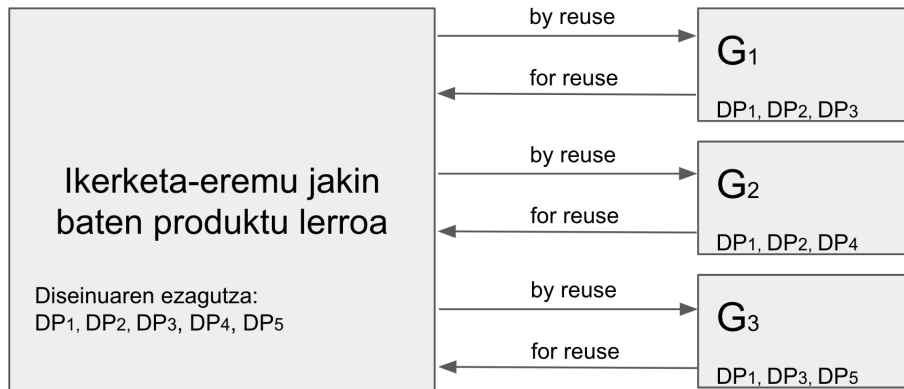
2. irudian ikus daitekeen bezala, Egokitzapen zikloak Diseinu zikloarekin elkar eragiten du. DSR gailuak Diseinu zikloan garatzen jarraitzen dira baina oraingoan produktu-lerroen ikuspegia erabiliz (*by reuse*). Hau da, prozesu berri honen bitartez gailuen garapena ez da hutsetik hasten, produktu-lerroan aurkitzen diren baliabideen berrerabilpenaz abiatuz baizik (Aplikazioaren Ingeniaritza). Beraz, oraingo honetan inplikazioa bikoitza da. Alde batetik, proiektu berriak bere beharrak aurreko proiektuetan garatutako ezaugarrien (*feature*) bitartez karakterizatzeko ahalegina egin behar du. Honek aurretik garatutako hainbat proiektutatik eratorritako esperientzien arteko alderaketa zehatzagoa egitea erraztuko du.

Beste alde batetik, proiektu berriak produktu-lerroak eskaintzen dituen baliabidetatik abiatzen bada litekeena da bere behar guztiak bete ezin izatea. Hau horrela baldin bada, gailu berriaren gainean moldaketak eta hedapenak egin behar du bere behar berriak bete ahal izateko. Produktu-lerroaren diziplinak planteatzen duen Domeinuaren Ingeniaritza prozesua kontuan hartzen badugu, egindako aldaketak eta hedapenak ez dira gailu berriari isolatuta geratzen. Planteamendu honek, garapen berriak sortutako esperientzia produktu-lerroa iraultzera eta integrazera behartzen du, etorkizuneko proiektu eta ikertzaileentzat moldaketak berrerabiltzeko prest eskainiz (*for reuse*).

DSR proiektuetako gailuen garapenean "*by reuse*" eta "*for reuse*" urratsak gehitzeak testuinguru bakoitzean gailuen oinarri izango den berrerabiltzeko plataforma inkrementalki haztea ahalbidetzen du, DSR proiektuetan zehar garatutako diseinuaren ezagutza eta gailuen iturburu kodea bertan metatuz (ikus 3. irudia).

Hau da planteatzen dugun ikerketaren muina. Ikuspegi berri hau hasierako faseetan dago eta oraindik ikerkuntza bide luzea dugu aurretik. Lan honetan proposatzen dugun abiapuntuari jarraipena emateko, ezin bestela izan,

3. Irudia: Produktu-lerroen teknikaren bitartez bideratutako gailu garapenaren irudikapen bat, non gailuen artean sakabanatutako Diseinu Printzipioak berrerabiltzeko plataforman biltzen diren.



hurrengo urratsetan DSR erabiliko dugu ikerketa-metodologia gisa. DSR ikerketak biltzen dituen hiru ziklo nagusi horien barruan, planteatzen ditugun hurrengo pausuak hauek dira:

Garrantzi-zikloa (Relevance Cycle). Ziklo honetan gure helburua da arazoa, hau da, diseinu printzipioen bilketa erraztuko duten mekanismo egoki eza, eta haren garrantzia ulertzea. Hori lortzeko zereginak hauek dira:

- Ikerketa softwarearen testuinguruan berrerabilpena karakterizatzea. Ikerketa softwareerako berrerabilpenaren berezitasunak identifikatzea du helburu ataza honek.
- Arazoaren azterketa kausala egin: zein dira arazoaren kausak? Motibazioa falta da software berrerabilgarria egiteko? Berrerabiltzaileen konfiantza falta? Eta zer ondorio ditu? Ondorioak, bai ikertzaileek inbertitutako denboran, bai gurpila berrasmatean edo lanak konparatzeko zailtasunak izan daitezke.
- Metrikak ezartzea, interesatuen helburuetan oinarritutako proiektuaren arrakasta neurtzeko. Interesatuak hauek dira: domeinu ingeniariak, produktu ingeniariak eta kudeatzaileak. Edo ikerketa-talde baten testuingurura itzulita: berrerabil daitekeen softwarea birfaktorizatzen duten doktoregaiak, hauek bere ikerketarako egindako softwarea berrerabiltzen (egokitzen) duten beste doktoregaiak, eta tesi-zuzendariak, batzuen eta besteen arteko transmisio-uhala direnak.
- Aurreikusitako ekarpenak:
 1. Kausa nagusiaren azterketa eta bibliografiatik lortutako euskarri osoa.
 2. Hirugarrenen softwarea eskuratzean gaur egun egiten diren jardueri buruzko galdetegiaren emaitzak.
 3. Proiektuaren arrakasta ebaluatzeko metrikaren zerrenda.

Diseinu-zikloa (Design Cycle). Ziklo honen helburua betekizunak eta tresna proposatzea da. Gure gailua produktu-lerroen informatika ingeniartzaren ikerketaren testuingurura egokitzea izango da. Produktu lerroen ingeniartzak hiru ziklo ditu: Domeinuaren Ingeniaritza, Aplikazioaren Ingeniaritza eta Kudeaketa Zikloa.

- Domeinuaren Ingeniaritza birformulatzea ikerketa softwareerako.
- Aplikazioaren Ingeniaritza birformulatzea ikerketa softwareerako.
- Ikertzaileek berrerabilpenerako garatzeko motibazioak bilatzea.
- Ikertzaileek berrerabilpenaren bidez garatzeko motibazioak bilatzea.
- Aurreikusitako ekarpenak:
 1. Produktu lerroen praktikak ikerketa-ingurunera egokitzea.
 2. ERRONKALine. Ministeritzak I+D RETOS programaren barruan finantzaturako OpenLine proiektua erabiliko da proba kasu gisa. OpenLine-n egindako gailuak berrerabiltzeko garatuko dira, baina, era

berean, hasieran garatzaileak izan ez diren ikertzaileek berrerabiliko dituzte ikerketa-talde berean. Helburua izango litzateke OpenLinek ez gailu bat sortzea, ONEKIN talde berari baina beste talde batzuei OpenLine-n garatutakoa berrerabiltzea ahalbidetuko liokeen ERRONKAline produktu lerro bat baizik. Horretarako, OpenLine proiektuak hedapen ekintzak aurreikusten ditu. Kasu honetan, ekintza horiek are zentzu handiagoa izango lukete; izan ere, zabalkundea ez litzateke ideien bidez soilik egingo, baizik eta, batez ere, hirugarrenek berrerabiltzeko sortutako gailuen bidez.

Zehaztasun-zikloa (Rigor Cycle). Ziklo honen helburua bikoitza da. Alde batetik, oinarritzko teoriak bilatzea, gailuen diseinuaren oinarrien euskarri. Eta bigarrenik, ERRONKAline gailua ebaluatzea, teoriari zer neurritan eusten dioten ikusteko.

- “Ezagutza biltzeari” buruzko lanak bilatzea.
- ERRONKAline ebaluatzea ONEKIN taldearen barruan
- ERRONKAline ebaluatzea produktu lerroen ezagutza duten beste ikerketa-taldeekin.
- ERRONKAline ebaluatzea produktu lerroen ezagutza EZ duten beste ikerketa-taldeekin.
- Aurreikusitako ekarpenak:
 1. Praktikei buruzko literatura berrikustea, ezagutza biltzeko (adibidez, literaturaren berrikuspen sistematikoak).
 2. ERRONKAlinen ebaluazio-txostenak, berrerabilpenerako garatuz ikuspegitik.
 3. ERRONKAlinen ebaluazio-txostenak, berrerabilpena erabiliz garatuz ikuspegitik.

4. Ondorioak

ISA oinarritutako lanen errepikatzea eta berrerabilpena ikerkuntza-komunitatean dagoen arazo erreala da, Software Ingeniaritzan batez ere. Ikerketa softwarearen aplikazio gutxi irauten dute behin sortu zituen ikerketa-proiektua amaitu eta gero, honen ondorioz, ikerketa errepikatzea zaila bihurtzen da. Askotan, ikertzaileek hutsetik antzeko arazoetarako softwarea eraiki behar izaten dute, gurpila aldiro berrasmatuz. Nahiz eta auzi nabarmena izan, oraindik gutxi landutako arazoa da. Honek, aurreko ikertzaile batek egindako lanei jarraipena ematea eta hedapenaren bidez bide berriak irekitzera ekiditen du. DSRren kasuan, arazo honek Diseinuaren Printzipioak hainbat gailuen artean sakabanatuta egotea dakar, ezagutza orokor bat galaraziz.

Lan honen helburu nagusia Diseinuaren Zientzia Ikerketa metodologia jarraitzen duten ikertzailek egindako lanen berrerabilpena sustatzea da. Horretarako, lan honetan software komertziala garatzeko ondo ezarrita dagoen Produktu Lerroen Ingeniaritza (PLI) IS-motako softwareari eta Diseinuaren Printzipioei aplikatzea aztertzen da. Hemen proposatzen den soluzioa ez da berrerabilpenaren behin betiko soluzio bat, software gailuen berrerabilpena errazteko eta modu sistematiko batean aurrera eramateko bide bat eskaintzea baizik. Etorkizunera begira espero da “ikerketa-eremu” desberdinek proposatutako berrerabilpen metodologia adoptatzea eremu horietan sortzen den ezagutzaren hedapena eta berrerabilpena hazi dadin.

5. Etorkizunerako planteatzen den norabidea

Lanaren hedapenari dagokionez hiru norabide bereizi daitezke proposatutako lanetik aurrera.

Batetik, artikuluan komentatu den moduan, DSR software gailuak eta ezagutza sortzeko erabiltzen den metodologia da. Hala ere, DSRren gaitasuna handiagoa da, eta arazo praktikoei soluzioa emateko ez da zertan bakarrik softwarean oinarritu behar. DSR softwarea ez den beste gailu batzuk ere sortzeko erabili egiten da hala nola, irudikapen eraikuntzak, ereduak eta metodoak (Johannesson eta Perjons, 2014). Izan ere, software produktu-lerroak ez dira soilik softwarearen kodea berrerabiltzera mugatzen. Software produktu lerroek kodeaz gain, softwarearen betekizunak, diagramak eta dokumentazio anitzak biltzeko eta sistematikoki berrerabiltzeko ere erabiltzen dira. Gure lanean gailuak software gisa kontuan hartu ditugu, ezagutzaren eramaile bezala, software produktu-lerroen bidez berrerabiltzeko. Beraz, aipatutako gailu hauen gainean software produktu-lerroetan oinarritutako berrerabilpena nola moldatuko litzatekeen ikerkuntzarako bide berri bat ireki dezake.

Bestetik, gure lana DSR metodologian oinarritzen den moduan, beste norabide bat beste metodologia berri batean oinarritutako ikerkuntza-softwarearen berrerabilpena aztertzea izan daiteke. Gure lanak DSRren ezaugarriak

kontuan hartzen ditu (diseinuaren printzipioak, ezagutza, ziklo prozesuak...) eta honen berrerabilpena gailuetan islatutako ezagutzaren berrerabilpena eta hedapena bultzatzen du. Adibidez, DSRek bezala gailuak kontuan hartzen dituen beste metodologia bat *Action Research* (Sein *et al.*, 2011) da eta aztertu daiteke hemen planteatutakoa bertan nola aplikatu.

Azkenik, Software Ingeniaritzakoa ez den beste ikerketa-eremuetan garatutako softwarea kontuan hartu daiteke etorkizuneko ikerketa gisa. Arlo hauetan jarduten duten ikertzaileek ez dute ezagutzarik software ingeniartzari buruz, honek erronka bat suposatuz dezake haien softwarearen berrerabilpen sistematikorako.

6. Erreferentziak

- Apel, Sven, Don Batory, Christian Kästner, eta Gunter Saake. 2013. A development process for feature-oriented product lines. In *Feature-Oriented Software Product Lines*, 17–44. Springer.
- Chen, Xiaoli, Sünje Dallmeier-Tiessen, Robin Dasler, Sebastian Feger, Pamfilos Fokianos, Jose Benito Gonzalez, Harri Hirvonsalo, Dinos Kousidis, Artemis Lavasa, Salvatore Mele, eta others. 2019. Open is not enough. *Nature Physics* 15.113–119.
- Diaz, Oscar, Haritz Medina, eta Jeremias Perez Contell. 2021. Promoting design knowledge accumulation through systematic reuse: The case for product line engineering. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, p. 5429.
- Frakes, William B, eta Kyo Kang. 2005. Software reuse research: Status and future. *IEEE transactions on Software Engineering* 31.529–536.
- Hettrick, Simon. 2016. Research software sustainability: Report on a knowledge exchange workshop.
- Heumüller, Robert, Sebastian Nielebock, Jacob Krüger, eta Frank Ortmeier. 2020. Publish or perish, but do not forget your software artifacts. *Empirical Software Engineering* 25.4585–4616.
- Hevner, Alan R. 2007. A three cycle view of design science research. *Scandinavian journal of information systems* 19.4.
- Johannesson, Paul, eta Erik Perjons. 2014. *An introduction to design science*. Springer.
- Lenarduzzi, Valentina, Terese Besker, Davide Taibi, Antonio Martini, eta Francesca Arcelli Fontana. 2019. Technical debt prioritization: State of the art. a systematic literature review. *arXiv preprint arXiv:1904.12538*.
- Sein, Maung K, Ola Henfridsson, Sandeep Puroo, Matti Rossi, eta Rikard Lindgren. 2011. Action design research. *MIS quarterly* 37–56.
- Steiniger, Stefan, eta Erwan Bocher. 2009. An overview on current free and open source desktop gis developments. *International Journal of Geographical Information Science* 23.1345–1370.
- Vom Brocke, Jan, Robert Winter, Alan Hevner, eta Alexander Maedche. 2020. Special issue editorial—accumulation and evolution of design knowledge in design science research: A journey through time and space. *Journal of the Association for Information Systems* 21.9.
- von Krogh, Georg, eta Stefan Haefliger. 2010. Opening up design science: The challenge of designing for reuse and joint development. *The Journal of Strategic Information Systems* 19.232–241.
- Winter, Robert. 2008. Design science research in europe. *European Journal of Information Systems* 17.470–475.
- Yalta, A Talha, eta A Yasemin Yalta. 2010. Should economists use open source software for doing research? *Computational Economics* 35.371–394.

7. Eskerrak eta oharrak

Aurkeztutako ikerkuntza-lerroa *Hawaii International Conference on System Science* izena duen nazioarteko kongresuan artikulua hoberenera proposatua izan den (Diaz *et al.*, 2021) artikulua eratorria da. Oraingo ikerketaren helburuak bertan planteatutako ideiarri jarraipena ematea eta ekarpen berriak lortzea izango dira.

Lan hau UPV/EHUko ONEKIN ikerketa-taldean burutuko da eta Xabier Garmendiak EHUko Ikertzaileak presatzeko helburua duen kontratu bat jaso du tesia egiteko.