

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Kursinis darbas

**Pakartotinis kodo panaudojimas pirminio kriptovaliutų
platinimo (ICO) išmaniuosiuose kontraktuose**

(Code review in initial coin offering (ICO) smart contracts)

Atliko: 3 kurso 1 grupės studentė

Agnė Mačiukaitė (parašas)

Darbo vadovas:

lekt. Gediminas Rimša (parašas)

Vilnius
2018

Turinys

Ivadas	2
1. Kintamumo modeliavimas (Variability modeling)	3
1.1. Savybių modeliavimas (Feature modeling)	3
1.1.1. Savybė	3
1.1.2. Savybių modelis	3
1.1.3. Savybių modeliavimo sistemos	3
1.2. Sprendimų modeliavimas (Decision modeling)	3
2. Programinės įrangos produktų linija	4
2.1. Produktų linijos savybių modeliavimas	4
2.2. Savybėmis orientuotas programavimas	4
3. Blockchain	5
3.1. Blockchain 1.0	5
3.1.1. Kriptovaliutos	5
3.2. Blockchain 2.0	5
3.2.1. Išmanusis kontraktas	5
3.2.2. Pirminis kriptovaliutų platinimas (ICO)	5
3.2.3. Ethereum	5
4. Pirminio kriptovaliuto platinimo išmaniojo kontrakto savybių modeliavimas	6
4.1. Savybės	6
4.2. Savybių modelis	6
4.3. Savybėmis grįstas išmanusis kontraktas pirminiui kriptovaliutų platinimui	6
Rezultatai	7
Išvados	8
Literatūra	9

Įvadas

Programinės įrangos produktų linija yra nauja paradigma programų kūrime, kuri kuria programas naudodamasi jau žinomomis esminėmis savybėmis, taip išvengdama produktų kūrimo nuo nulio. Toks kūrimas susideda iš savybių išskyrimo ir jų įgyvendinimo produkte. Gerai išskirtos produkto ypatybės padeda sukurti lengvai pernaudojamą programą. Savybės turi būti atrinktos atsižvelginat į jų paplitimą bei kintamumą srityje. Produktų linijos susiaurinimas iki srities padidina pernaudojamą bei panaudojamumą [LKL15]. **PABAIGTI**

Savybių modeliavimas yra pagrindinis metodas atrinkti bei valdyti bendrąsias ir kintamas savybes produktų linijoje. Programinės įrangos šeimos gyvavimo pradžioje savybių modelis padeda išskirti pagrindines savybes, kurios gelbsti kuriant naują rinką ar norint išlikti jau esamoje. Taip pat savybių modelis leidžia išskirti rizikingas savybes, nuspėti, kokia yra visos programos ar atskirų savybių kaina. Vėliau savybių modeliavimas padeda išskirti variacijos taškus programinės įrangos architektūroje [CHE04]. Savybių modeliavimas yra populiariausias programinės įrangos produktų linijų kurime nuo pat pirmojo jo pristatymo [KCH⁺90]. Taip yra todėl, nes savybės yra pakankamai abstraktus konceptas padedantis efektyviai bendrauti suinterasuotoms šalims. Savybių modeliavimas yra intuitivus ir efektyvus būdas žmonėms išreikšti savybių paplitimą ir kintamumą programinės įrangos šeimoje [KL13].

Tyrimo problema: Išmaniųjų kontraktų technologijos yra pakankamai jaunos, dėl to projektavimo šablonai bei pakartotinai panaudojamo kodo bazė dar tik formuojasi. ICO kontraktai tiražuojami kopijavimo su modifikacijomis būdu.

Tyrimo tikslas: Ištirti sutelktinio finansavimo kriptovaliutomis (ICO) išmaniuosius kontraktus, nustatyti koks funkcionalumas yra pastovus, o koks - kintantis, bei pasiūlyti būdus pakartotinio panaudojamumo laipsniui didinti. Įvade apibūdinamas darbo tikslas, temos aktualumas ir siejami rezultatai.

1. Kintamumo modeliavimas (Variability modeling)

1.1. Savybių modeliavimas (Feature modeling)

Citavimo pavyzdžiai: cituojamas vienas šaltinis [Bat05];

1.1.1. Savybė

1.1.2. Savybių modelis

1.1.3. Savybių modeliavimo sistemos

1.2. Sprendimų modeliavimas (Decision modeling)

2. Programinės įrangos produktų linija

2.1. Produktų linijos savybių modeliavimas

2.2. Savybėmis orientuotas programavimas

3. Blockchain

3.1. Blockchain 1.0

3.1.1. Kriptovaliutos

3.2. Blockchain 2.0

3.2.1. Išmanusis kontraktas

3.2.2. Pirminis kriptovaliutų platinimas (ICO)

3.2.3. Ethereum

4. Pirminio kriptovaliuto platinimo išmaniojo kontrakto savybių modeliavimas

4.1. Savybės

4.2. Savybių modelis

4.3. Savybėmis grįstas išmanusis kontraktas pirminiui kriptovaliutų platinimui

Rezultatai

Išvados

Išvadose ir pasiūlymuose, nekartojant atskirų dalių apibendrinimų, suformuluojamos svarbiausios darbo išvados, rekomendacijos bei pasiūlymai.

Literatūra

- [Bat05] Don Batory. Feature Models, Grammars, and Propositional Formulas. *LNCS*, tom. 3714, p.p. 7–20, 2005. URL: <http://www.cs.utexas.edu/ftp/predator/splc05.pdf>.
- [CHE04] Krzysztof Czarnecki, Simon Helsen ir Ulrich Eisenecker. Staged Configuration Using Feature Models. 2004. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.1586%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.
- [KCH⁺90] Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak ir A. Spencer Peterson. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, tom. 18 numeris 3-4. 1990. ISBN: 1111111111. DOI: 10.1080/10629360701306050.
- [KL13] Kyo C. Kang ir Hyesun Lee. Variability Modeling. *Syst. Softw. Var. Manag.* P.p. 25–42. 2013. ISBN: 978-3-642-36582-9. DOI: 10.1007/978-3-642-36583-6. URL: <http://link.springer.com/10.1007/978-3-642-36583-6>.
- [LKL15] Kwanwoo Lee, Kyo C. Kang ir Jaejoon Lee. Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. 2015. DOI: 10.1007/3-540-46020-9_5. URL: <https://www.researchgate.net/profile/Kwanwoo%7B%5C%7DLee/publication/221553200%7B%5C%7DConcepts%7B%5C%7Dand%7B%5C%7DGuidelines%7B%5C%7Dof%7B%5C%7DFeature%7B%5C%7DModeling%7B%5C%7Dfor%7B%5C%7DProduct%7B%5C%7DLine%7B%5C%7DSoftware%7B%5C%7DEngineering/links/558bdbde08ae591c19d8d3ce.pdf>.