

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Pakartotinis kodo panaudojimas pirminio
kripto valiutų platinimo (ICO) išmaniuosiuose
kontraktuose**

Code review in initial coin offering (ICO) smart contracts

Kursinis darbas

Atliko: 3 kurso 1 grupės studentė
Agnė Mačiukaitė (parašas)

Darbo vadovas: lekt. Gediminas Rimša (parašas)

Vilnius
2018

TURINYS

ĮVADAS	2
1. SAVYBIŲ MODELIAVIMAS	4
1.1. Savybė	4
1.2. Savybių modelis	4
2. SAVYBIŲ MODELIAVIMAS PIRMINIO KRIPTOVALIUTŲ PLATINIMO IŠMANIE- SIEMS KONTRAKTAMS	5
2.1. Savybė	5
2.2. Savybių modeliavimas	5
REZULTATAI	6
IŠVADOS	7
LITERATŪRA	8
SAVOKŲ APIBRĖŽIMAI	10
SANTRUMPOS	11

Įvadas

Programinės įrangos pernaudojimas leidžia naudoti programas keliuose projektuose. Tai yra svarbi strategija programinei įrangai norint padidinti sistemos efektyvumą ir kokybę. Taikant pernaudojamumą programuotojai naudojami jau įgyvendintu kodu, kurį keičia taip, kad jis atitiktų dabartinio projekto reikalavimus [RR03].

Pirminio kriptovaliuto platinimo (angl. initial coin offering, toliau ICO) metu įmonė parduoda specializuotus kripto-žetonus žadėdami, kad žetonai veiks kaip mainų priemonė gaunant paslaugas įmonės platformoje. Žetonų pardavimas kuria kapitalą pradiniam įmonės platformos kūrimui nors nėra įsipareigojimo dėl būsimos paslaugos kainos (žetonais ar kitaip) [CG18]. Satoshi Nakamoto išleidęs baltąjį popierių (angl. whitepaper) [Nak08] įvykdė ICO ir taip surinko finansavimą pirmajam blockchain ir kriptovaliutai Bitcoin. Bitcoin - skaitmeniniai pinigai, kurių pavedimai vyksta internete naudojantis decentralizuota vieša duomenų baze - blockchain [Swa15]. Šiuo metu du populiariausi blockchain yra Ethereum ir Bitcoin [LCO⁺16]. Ethereum be savo kriptovaliutos turi ir kitą svarbų funkcionalumą - išmaniuosius kontraktus - Turing complete programą, kuri leidžia rašyti decentralizuotas aplikacijas [But14]. Solidity - populiariausia kalba naudojama rašyti išmaniesiems kontraktams [Dan17]. Problema - išmaniųjų kontraktų technologijos yra pakankamai jaunos, dėl to pakartotinio kodo panaudojimo bazė dar tik formuojasi. ICO kontraktai yra tiražuojami kopijavimo su modifikacijos būdu.

Programinės įrangos produktų linija (angl. product line software engineering, toliau PLSE) naudojama įmonėse pakartojamumui susijusiuose programinės įrangos produktuose numatyti. PLSE suteikia bendrą architektūrą ir pernaudojamą kodą programinės įrangos kūrėjams [SVB01]. Toks kūrimas susideda iš savybių išskyrimo ir jų įgyvendinimo produkte. Gerai išskirtos produkto ypatybės padeda sukurti lengvai pernaudojamą programą. Savybės turi būti atrinktos atsižvelginant į jų paplitimą bei kintamumą srityje [LKL15]. Naudojantis PLSE produkto kūrėjai gali fokusuotis produkto specifikacijoje, o ne bendrų savybėse [SVB01].

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 PLSE kūrime 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].

Šio darbo tikslas - ištirti pirminio finansavimo kriptovaliutomis (ICO) išmaniuosius kontraktus, nustatyti, kokios savybės yra pastavios, o kokios - kintamos bei pasiūlyti būdus kodo pernaudojamumui didinti.

Tikslui pasiekti išsikelti uždaviniai:

1. Apžvelgti savybių modeliavimą programinės įrangos produktų linijos sričiai

2. Surinkti virš 100 išmaniųjų kontraktų skirtų ICO
3. Išskirti surinktų kontraktų savybes į pastovias ir kintančias
4. Pasiūlyti ICO išmaniuosius kontraktus pagal išrinktas savybes

1. Savybių modeliavimas

Savybių modeliavime bendri ir kintami bruožai yra modeliuojami iš produkto savybių perspektyvos PLSE, kuri yra suinteresuotos šalies interesas. Originalus savybių modeliavimas - FODA [KCH⁺90] - paprastas modelis, kuris savybes skirsto pagal tai iš ko jos susideda bei pagal bendrumą ir specializaciją naudojant AND/OR diagramas. Savybės yra suskirstytos į būtiną, alternatyvą ir pasirenkamą pagal bendrus ir kintamus bruožus. Savybių atributai taip pat gali būti dokumentuojami [KL13].

1.1. Savybė

Savybės yra pagrindinis produkto skiriamasis bruožas. Skirtingi srities analizės metodai terminą „savybė“ apibūdina šiek tiek kitaip. FODA [KCH⁺90] savybę apibūdina kaip pastebimą ir skiriamą sistemos charakteristiką, kuri yra matoma įvairioms suinteresuotoms šalims. Svarbu, kad savybių modeliavime turi būti fokusuojamasi ties bendrumo ir skirtumų srityje identifikacija, o ne ties bendrų savybių supratimo išskyrimu. Iš kitos pusės neapibrėžta savybių paskirtis daro sunkumų formuluojant jos semantiką, rezultatų valdymą bei automatinį pagalbo suteikimą.

Skirtumas tarp savybės ir konceptualios abstrakcijos (pvz.: funkcijos, objekto) yra tai, kad funkcijos ir objektai yra naudojami specifikuojant vidines sistemos detales. Kitaip, funkcijos ir objektai yra konceptualios abstrakcijos, kurios yra identifikuojamos iš vidinės sistemos pusės. Savybė - aiškiai matoma pagal charakteristiką, kuri gali išskirti produktą iš kitų. Todėl savybių modeliavimas turi išskirti iš išorės matomas charakteristikas produktuose bendrumo ir kintamumo atžvilgiu, o ne apibūdinti visas produkto modeliavimo detales (pvz.: funkcinis, objektais orientuotas modeliavimas). Suprantant produkto bendrus ir kintamus bruožus galima sukurti pernaudojamas funkcijas ir objektus [LKL15].

Savybėmis grįsti PLSE modeliai naudoja savybes kaip vienetus sudarytus iš:

- Veiksmų, kurie suteikiami vartotojams
- Reikalavimų
- Produkto konfigūracijos ir jos valdymo
- Kūrimo ir pristatymo klientams
- Parementrizacijos pernaudojamiems vienetais
- Produkto valdymo skirtingiems rinkos sektoriams [KL13]

1.2. Savybių modelis

2. Savybių modeliavimas pirminio kriptovaliutų platinimo išmaniesiems kontraktams

2.1. Savybė

2.2. Savybių modeliavimas

Rezultatai

Išvados

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

Literatūra

- [But14] Vitalik Buterin. A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM. 2014. URL: https://www.weusecoins.com/assets/pdf/library/Ethereum%7B%5C_%7Dwhite%7B%5C_%7Dpaper-a%7B%5C_%7Dnext%7B%5C_%7Dgeneration%7B%5C_%7Dsmart%7B%5C_%7Dcontract%7B%5C_%7Dand%7B%5C_%7Ddecentralized%7B%5C_%7Dapplication%7B%5C_%7Dplatform-vitalik-buterin.pdf.
- [CG18] Christian Catalini ir Joshua S Gans. Initial Coin Offerings and the Value of Crypto Tokens, 2018. URL: <http://creativecommons.org/licenses/by-nc/4.0/%20https://ssrn.com/abstract=3137213>.
- [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>.
- [Dan17] Chris Dannen. Introducing Ethereum and Solidity Foundations of Cryptocurrency and Blockchain Programming for Beginners Introducing Ethereum and Solidity Foundations of Cryptocurrency and Blockchain Programming for Beginners Introducing Ethereum and Solidity: Foundation. *Library of Congress Control Number*, 2017. DOI: 10.1007/978-1-4842-2535-6. URL: <http://smartcontracts.engineer/wp-content/uploads/2017/09/Etherium.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. *Systems and Software Variability Management*, 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>.
- [LCO⁺16] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena ir Aquinas Hobor. Making Smart Contracts Smarter, 2016. DOI: 10.1145/2976749.2978309. URL: <https://www.comp.nus.edu.sg/%7B%7Dloiluu/papers/oyente.pdf>.
- [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.
- [Nak08] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. URL: <https://bitcoin.org/bitcoin.pdf>.

- [RR03] T. Ravichandran ir Marcus A. Rothenberger. SOFTWARE REUSE STRATEGIES AND COMPONENT MARKETS. *Communications of the ACM*, 46(8), 2003. doi: 10.1145/859670.859678. URL: https://www.researchgate.net/profile/T%7B%5C_%7DRavichandran/publication/220423696%7B%5C_%7DSoftware%7B%5C_%7Dreuse%7B%5C_%7Dstrategies%7B%5C_%7Dand%7B%5C_%7Dcomponent%7B%5C_%7Dmarkets/links/54201aa60cf2218008d43cdb.pdf.
- [SDN⁺04] Marco Sinnema, Sybren Deelstra, Jos Nijhuis ir Jan Bosch. COVAMOF : A Framework for Modeling Variability in Software Product Families. 2004. doi: 10.1007/978-3-540-28630-1_12. URL: https://www.researchgate.net/profile/J%7B%5C_%7DBosch/publication/220789726%7B%5C_%7DCOVAMOF%7B%5C_%7DA%7B%5C_%7DFramework%7B%5C_%7Dfor%7B%5C_%7DModeling%7B%5C_%7DVariability%7B%5C_%7Din%7B%5C_%7DSoftware%7B%5C_%7DProduct%7B%5C_%7DFamilies/links/09e415086dcdf501dc000000.pdf.
- [SVB01] Mikael Svahnberg, Jilles Van Gurp ir Jan Bosch. On the Notion of Variability in Software Product Lines, 2001. URL: [http://www.ipd.hk-r.se/\[msv%7B%5C_%7D7Cjvg%7B%5C_%7D7Cbosch\]](http://www.ipd.hk-r.se/[msv%7B%5C_%7D7Cjvg%7B%5C_%7D7Cbosch]).
- [Swa15] Melanie Swan. *Blockchain. Blueprint for a new economy*. 2015, p. 149. ISBN: 978-1-491-92049-7. URL: <http://w2.blockchain-tec.net/blockchain/blockchain-by-melanie-swan.pdf>.
- [Tel94] Astro Teller. Turing Completeness in the Language of Genetic Programming with Indexed Memory, 1994. URL: <http://www.astroteller.net/content/3-work/2-papers/17-turing-completeness-in-the-language-of-genetic-programming-with-indexed-memory/turing.pdf>.

Sąvokų apibrėžimai

Turing complete - bet kuri sistema, kuri yra pakankamai galinga atpažinti visus galimus algoritmus [Tel94].

Santrumpos

PLSE - programinės įrangos produktų linija (angl. product line software engineering)

ICO - pirminis kriptovaliutų platinimas (angl. initial coin offering)