

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

# **Kodo skirstymo į paketus šablonų tyrimas**

## **Analysis of code packaging patterns**

Bakalaurinio darbo planas

Atliko: 4 kurso 3 grupės studentė  
Martyna Ubartaitė

Darbo vadovas: Gediminas Rimša

Vilnius – 2024

## Problema ir jos aktualumas

Teisingai įgyvendintas kompiuterinės sistemos dizainas yra vienas iš kritinių sėkmingo verslo elementų. Tam, jog verslas išlaikytų stabilų augimą yra būtina sukurti sistemą, kuri sumažintų atotrūkį tarp organizacijos tikslų ir jų įgyvendinimo galimybių. Mąstant apie programinio kodo dizainą, kodo paketų kūrimas, klasių priskyrimas jiems ir paketų hierarchijos sudarymas paprastai nėra pagrindinis prioritetas, tačiau tai parodo praleistą galimybę padaryti sistemos dizainą labiau patikimu[Sho19], suprantamu [Eli10] ir lengviau palaikomu. Modernios sistemos yra didžiulės, programinis kodas yra padalintas į daugybę failų, kurie išskaidyti per skirtingo gylio direktorijas, todėl apgalvotai išskirstytas programinis kodas daro daug didesnę įtaką kodo kokybei, nei gali atrodyti iš pirmo žvilgsnio. Norint išsiaiškinti, kaip efektyviausiai gali būti skaidomas programinis kodas, reikalinga atlikti skirstymo šablonų analizę, suprasti jų privalumus bei trūkumus.

## Darbo tikslas

Šio darbo tikslas – identifikuoti šablonus kodo skirstymui į paketus ir pagal aprašytus kriterijus įvertinti jų įtaką sistemos palaikomumui.

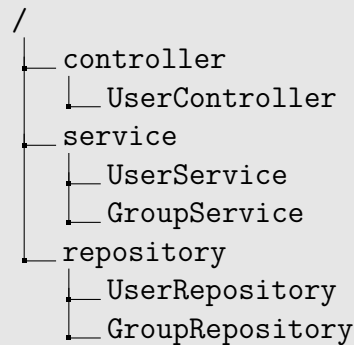
## Keliami uždaviniai

1. Išskirti gerai įgyvendinto kodo požymius
2. Aprašyti skirstymo į paketus šablonus, remiantis pavyzdžiais teorinėje medžiagoje bei egzistuojančiose atviro kodo sistemose
3. Pasiūlyti kriterijus, įvertinančius kodo suskirstymo šablono įtaką sistemos kokybei
4. Pasirinkti kelias sistemas ir pertvarkyti jų failų struktūrą pagal aprašytus šablonus
5. Naudojant pertvarkytas sistemas, įvertinti kiekvieną kodo skirstymo šabloną pagal pasiūlytus kriterijus
6. Pateikti rekomendacijas, kokius šablonus kodo skirstymui tinkamiausia naudoti

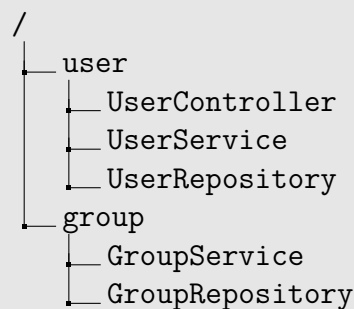
## Numatomas darbo atlikimo procesas

1. Remiantis teorine medžiaga aprašomi gerai įgyvendinto kodo požymiai, užtikrinantys sistemos stabilumą ir palaikomumą
2. Aprašomi kriterijai, kuriuos naudojant galima įvertinti kodo suskirstymo įtaką sistemos kodo kokybei, pavyzdžiui:
  - Komponentų skaičius, priklausomas nuo pasirinkto komponento[Mah03]
  - Tiesioginės ir netiesioginės priklausomybės (matomumas)[Ala07]
3. Galimų kodo skirstymo šablonų išskyrimas pasitelkiant teorinę informaciją [Sho11] ir egzistuojančių sistemų architektūrą. Aprašomo šablono pavyzdžiai:

- Pagal techninį sluoksnį – klasės, priklausančios vienam techniniam sistemos sluoksniui, laikomos tame pačiame pakete, taip išlaikant aiškią paketų priklausomybių kryptį:



- Pagal komponentą – klasės atskiriamos ir grupuojamos tarpusavyje pagal tai, kokią funkcionalumą jos teikia. Tai gali būti techninis funkcionalumas arba dalykinės srities veiksmas:



- Atviro kodo projektų pasirinkimas. Pasirenkami skirtingo tipo projektai, užtikrinant objektyvesnę šablonų analizę skirtingose srityse. Galimi tipai:
  - Taikomoji programinė įranga, teikianti paslaugas įrangos naudotojams. Pavyzdžiui, internetinė programėlė priminimams ir darbams užsirašyti
  - Techninė programinė įranga, naudojama taikomosios programinės įrangos duomenų saugojimui, siuntimui, paieškai. Pavyzdžiui, duomenų bazės, pranešimų eilės, talpyklos (angl. *cache*)
  - Programinės įrangos įrankiai, skirti naudoti kitose sistemose supaprastinant programinį kodą, naudojant jau įgyvendintas funkcijas. Pavyzdžiui, Java programavimo kalbos Spring karkasas internetinių programėlių kūrimui
- Pasirinktų projektų paketų struktūros pertvarkymas pagal pasirinktus skirstymo šablonus
- Pertvarkytų projektų įvertinimas, naudojant išskirtus kriterijus, nustatant, kokią įtaką skirtingi skirstymo šablonai turi sistemos kokybei
- Rezultatai

## Laukiami rezultatai

1. Identifikuoti kodo skirstymo šablonai, remiantis teorine informacija ir praktikoje sutinkamais pavyzdžiais

2. Sukurti kriterijai, įvertinantys sistemos paketų struktūros indėlį sistemos kokybei
3. Pasirinkti projektai pertvarkyti pagal kodo skirstymo šablonus
4. Įvertinus šablonus sukurtais kriterijais, pateiktos rekomendacijos kodo skirstymo šablonų naudojimui

## Šaltiniai

- [Ala07] C. B. Alan MacCormack John Rusnak. The Impact of Component Modularity on Design Evolution: Evidence from the Software Industry. 2007 [žiūrėta 2024-03-04]. Prieiga per internetą: [https://www.hbs.edu/ris/Publication%20Files/08-038\\_187f1243-7d1e-464a-bddd-bf7c09873284.pdf](https://www.hbs.edu/ris/Publication%20Files/08-038_187f1243-7d1e-464a-bddd-bf7c09873284.pdf).
- [Eli10] M. Elish. Exploring the Relationships between Design Metrics and Package Understandability: A Case Study. 2010 [žiūrėta 2024-03-04]. Prieiga per internetą: [https://www.researchgate.net/publication/221219583\\_Exploring\\_the\\_Relationships\\_between\\_Design\\_Metrics\\_and\\_Package\\_Understandability\\_A\\_Case\\_Study](https://www.researchgate.net/publication/221219583_Exploring_the_Relationships_between_Design_Metrics_and_Package_Understandability_A_Case_Study).
- [Har23] K. B. Haris Mumtaz Paramvir Singh. Identifying refactoring opportunities for large packages by analyzing maintainability characteristics in Java OSS. 2023 [žiūrėta 2024-03-04]. Prieiga per internetą: <https://www.sciencedirect.com/science/article/abs/pii/S0164121223001127>.
- [Hou11] H. A. Houari A. Sahraoui Stéphane Ducasse. Modularization Metrics: Assessing Package Organization in Legacy Large Object-Oriented Software. 2011 [žiūrėta 2024-03-04]. Prieiga per internetą: [https://www.researchgate.net/publication/221200426\\_Modularization\\_Metrics\\_Assessing\\_Package\\_Organization\\_in\\_Legacy\\_Large\\_Object-Oriented\\_Software](https://www.researchgate.net/publication/221200426_Modularization_Metrics_Assessing_Package_Organization_in_Legacy_Large_Object-Oriented_Software).
- [jav17] javadevguy. Happy Packaging! 2017 [žiūrėta 2023-11-01]. Prieiga per internetą: <https://javadevguy.wordpress.com/2017/12/18/happy-packaging/>.
- [Kle17] M. Kleppmann. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media, 2017.
- [Mah03] D. R. Mahmoud O. Elish. Investigation of Metrics for Object-Oriented Design Logical Stability. 2003 [žiūrėta 2024-03-04]. Prieiga per internetą: <https://core.ac.uk/download/pdf/146513146.pdf>.
- [Sad21] M. Sadowski. Divide Your Codebase by Domains and Features To Keep It Scalable. 2021 [žiūrėta 2023-11-02]. Prieiga per internetą: <https://betterprogramming.pub/divide-code-by-domains-and-features-and-keep-it-scalable-bb5bd66cf3d2>.
- [San] M. Sandin. Four Strategies for Organizing Code. [Sine anno] [žiūrėta 2024-03-04]. Prieiga per internetą: <https://medium.com/@msandin/strategies-for-organizing-code-2c9d690b6f33#:~:text=This%20article%20outlines%20four,worked%20with%2C%20pleasurable%20and%20not>.

- [Sho11] M. A. Shouki Ebad. Software Packaging Approaches - A Comparison Framework. 2011 [žiūrėta 2024-03-04]. Prieiga per internetą: [https://www.researchgate.net/publication/220757169\\_Software\\_Packaging\\_Approaches\\_-\\_A\\_Comparison\\_Framework](https://www.researchgate.net/publication/220757169_Software_Packaging_Approaches_-_A_Comparison_Framework).
- [Sho19] M. A. Shouki A. Ebad. Investigating The Effect of Software Packaging on Modular Structure Stability. 2019 [žiūrėta 2024-03-04]. Prieiga per internetą: [https://d1wqtxts1xzle7.cloudfront.net/75578419/pdf-libre.pdf?1638471828=&response-content-disposition=inline%3B+filename%3DInvestigating\\_the\\_Effect\\_of\\_Software\\_Pac.pdf&Expires=1709553794&Signature=BWU2DSSXDNujfvTl1UWYspcqNlbW1Fgg~doSlc6JoeK7XXJ5bGLPB1B1yBD0tnojJ0yNuWZzQPfpTjd~yff0hlxnM4GyB2gNMuGZyXsDBXWQuD66kZpwWdluJ63GWjvs28T2ArRpaekqk9JAc-Icun18nyonYJ~W4pIViibjHA4k9yN5r1FZRSWFeUSHpRmflEpJ1opD2Nh889TquLxsA2DhPP3L\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/75578419/pdf-libre.pdf?1638471828=&response-content-disposition=inline%3B+filename%3DInvestigating_the_Effect_of_Software_Pac.pdf&Expires=1709553794&Signature=BWU2DSSXDNujfvTl1UWYspcqNlbW1Fgg~doSlc6JoeK7XXJ5bGLPB1B1yBD0tnojJ0yNuWZzQPfpTjd~yff0hlxnM4GyB2gNMuGZyXsDBXWQuD66kZpwWdluJ63GWjvs28T2ArRpaekqk9JAc-Icun18nyonYJ~W4pIViibjHA4k9yN5r1FZRSWFeUSHpRmflEpJ1opD2Nh889TquLxsA2DhPP3L_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA).
- [Stu13] D. J. Sturtevant. System design and the cost of architectural complexity. 2013 [žiūrėta 2024-03-04]. Prieiga per internetą: <https://dspace.mit.edu/handle/1721.1/79551>.
- [Suh12] R. A. K. Suhel Ahmad Khan. Analyzability Quantification Model of Object Oriented Design. 2012 [žiūrėta 2024-03-04]. Prieiga per internetą: <https://www.sciencedirect.com/science/article/pii/S2212017312003647>.