

# Dekomponavimo kortų metodas sudėtingoms informacinėms sistemoms dekomponuoti

**Gediminas Krasauskas**

Vilniaus universitetas, Matematikos ir informatikos fakultetas,  
Didlaukio g. 47, LT-08303 Vilnius  
*gediminas.krasauskas@mif.stud.vu.lt*

---

**Santrauka.** Šiame straipsnyje pateikiami sudėtingų informacinių sistemų dekomponavimo, būtino siekiant mažinti artefakto, kurį reikia įgyvendinti, sudėtingumą, tyrimo rezultatai. Programų sistemų kūrimo srityje yra atlikta įvairių tyrimų, skirtų dekompozicijos procesui analizuoti ir kiekybiniais dekomponavimo rezultatų vertinimams atlikti. Tačiau programų sistemų kūrime naudojami „skaidyk ir valdyk“ metodai negali būti tiesiogiai taikomi sudėtingoms informacinėms sistemoms dekomponuoti, pamatuoti dekompozicijos gerumą ir palyginti su kitomis alternatyvomis. Pasiūlytasis dekomponavimo kortų metodas, kuris apima struktūrinį dekompozicijos vertinimą atsižvelgiant į kiekybinius atributus, charakterizuojamas kaip hierarchinis ir nedarantis konkrečių techninių sprendimų aukštame abstrakcijos lygmenyje. Gauti rezultatai sudaro prielaidas tolesniems tyrimams vykdyti.

**Raktiniai žodžiai:** sistemų dekomponavimas, informacinės sistemos, sudėtingos sistemos, kokybės funkcijų sklaidos metodika, struktūrinis sudėtingumas.

---

## 1 Įvadas

Kuriant sudėtingas informacines sistemas dekomponavimo procesas turi būti pagrįstas ir metodiškai vykdomas. Tai yra vienas iš pirminių tokių sistemų kūrimo gyvavimo ciklo etapų. Nuo dekompozicijos kokybės tiesiogiai priklauso visos kuriamos sistemos ir jos dalių kokybė bei sudėtingumas.

Už sistemos dekompoziciją yra atsakingi sistemų architektai (toliau – architektai). Tačiau kiekvieno architekto kompetencijos, patirtis ir požiūris į kuriamas sistemas skiriasi. Dėl šios priežasties yra tikėtina, kad kiekvieno iš architektų atlikto sistemos dekomponavimo rezultatai skirsis. Prie skirtingų rezultatų prisideda ir tai, kad nėra visuotinai pripažintų universalių standartų arba metodų šiai veiklai atlikti, vyrauja architektūros stilių įvairovė. To

pasekmė – sistemų kūrimo projektų nesėkmės, kurias reikšmingai nulemia nekokybiška dekompozicija, sukuriant sistemas, kurios nepasiekia norimų tikslų, arba padidina sistemos sudėtingumą iki nevaldomo.

Pateiktos priežastys leidžia teigti, kad sudėtingų informacinių sistemų dekomponavimas yra aktuali problema. Prasmingi rezultatai šioje srityje, tokie kaip metodiniai pagrindai, kiekybinis dekompozicijos matavimas ir dekompozicijos alternatyvų vertinimas, gali ne tik palengvinti architektų darbą, bet ir užtikrinti tokias kuriamų sistemų savybes, kaip kokybė ir potencialus sudėtingumas.

Šiame straipsnyje pateikiamas atlikto tyrimo rezultatas – parodoma, koku būdu įvertinti atliktą dekompoziciją ir kaip gauti kiekybinius vertinimo rezultatus, šitaip užtikrinant skirtingų dekompozicijos alternatyvų palyginimo galimybes.

Antrame skyriuje apžvelgiami kitų autorių, nagrinėjusių panašią problemą, vykdyti tyrimai ir rezultatai. Trečiame skyriuje aptariamas dekomponavimo kortų metodas bei jo dalys ir kuris yra grindžiamas kokybės funkcijų sklaidos metodikos principais. Ketvirtame skyriuje pateikiamos dekompozicijos matavimo ir vertinimo galimybės taikant dekomponavimo kortų metodą. Penktame skyriuje pateiktos darbo išvados.

## 2 Susiję darbai

Nemažai tyrimų yra atlikta sistemos dekomponavimo strategijų analizės srityje. Tačiau nėra aišku, kuris dekomponavimo būdas labiausiai tinkamas konkrečiam sistemos tipui arba jos dekompozicijai konkrečiame abstrakcijos lygmenyje. Literatūroje dažniausiai galima aptikti tokius dekomponavimo būdus: funkcinę [1], probleminių freimų [2], agentų [3], kintamumo [4], surinkimo [5] ir kitas. Straipsnyje [6] yra pateikiamas apibendrintas dekomponavimo būdų klasifikavimas, išskiriantis 11 skirtingų galimų dekompozicijos kategorijų kombinuojant sistemos struktūrų, elgsenų ir tikslų kriterijus.

Taip pat yra atlikta įvairių tyrimų sistemų sudėtingumo srityje ir kaip jį sąlygoja dekompozicija. Straipsnis [7] išskiria 3 sistemų sudėtingumo tipus: struktūrinį, dinaminį ir organizacinį. Autoriai teigia, kad struktūrinis sudėtingumas priklauso nuo sistemos vidaus, t. y. jos sudedamųjų dalių. Dinaminis sudėtingumas priklauso nuo sistemos elgsenos, funkcijų, o organizacinis sudėtingumas yra veikiamas organizacijos struktūros ir sistemos vystymo procesų.

Struktūriniam sudėtingumui matuoti nemažai autorių pateikia įvairių kiekybinių matavimo būdų. Vienas iš jų straipsnyje [8] pateikia kompleksinį matavimą atsižvelgiant į 3 kriterijus: individualių sudedamųjų dalių sudėtingumą, ryšių tarp sudedamųjų dalių sudėtingumą ir topologinį sistemos sudėtingumą. Straipsnis [9] taip pat siūlo analogišką struktūrinio sudėtingumo matavimą, kuriuo remiantis yra atsižvelgiama į dekompozicijos granuliarumą, t. y. sudedamųjų dalių kiekį. Literatūroje galima rasti topologinio sistemų sudėtingumo matavimus grafų struktūrose pasinaudojant matricos energija [10].

Straipsnyje [11] pateiktos 8 struktūrinio sudėtingumo metrikos, suskirstytos pagal moduliarumą, pakartotinį panaudojimą ir sistemos vidinę hierarchiją, o [12] autoriai pateikia 9 metrikas, kurių dalis yra skirta struktūriniam sudėtingumui matuoti. Ryšiams tarp sudedamųjų dalių sudėtingumui pamatuoti straipsnio [9] autoriai siūlo pagal 3 aspektus: suderinamumu, savalaikiškumu ir konfidencialumu, o šie aspektai papildomai turi individualią vertinimo skalę. Straipsnio [12] autoriai papildomai siūlo įtraukti ryšių kryptingumą ir pabrėžia ryšių ciklą svarbą sudėtingumo matavimui. Kai kurie autoriai siūlo matricomis paremtos metodikos taikymą sistemos sudedamųjų dalių tarpusavio ryšiams analizuoti, kuri yra vadinama DSM [13].

### **3 Dekomponavimo kortų metodas sudėtingoms sistemoms dekomponuoti**

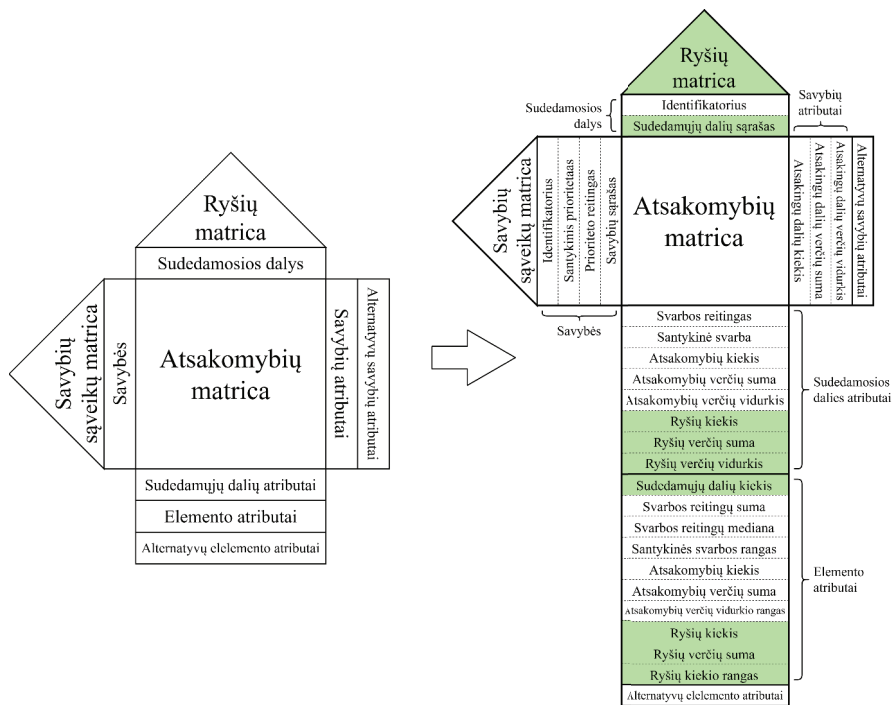
Tyrimo metu sudarytas dekomponavimo kortų (DK) metodas savo esme grindžiamas kokybės funkcijų sklaidos (KFS) metodika [14]. DK metodas yra skirtas sudėtingoms sistemoms dekomponuoti ir jį taikant galima pasiekti šiuos tikslus:

- transformuoti sistemos reikalavimus į sistemos architektūrinius sprendinius;
- užtikrinti, kad vykdant dekompoziciją nebūtų ignoruoti ar praleisti jokie reikalavimai;
- įvertinti ir palyginti skirtingas dekompozicijos alternatyvas ir pasirinkti labiau tinkamą.

DK metodas – tai KFS metodikos ir jos kokybės kortos pritaikymas sudėtingų sistemų dekomponavimo procesui. Originali KFS metodika yra skirta naudotojų poreikiams transformuoti į produkto inžinerines charakteristikas.

Po pirmojo taikymo Japonijos automobilių pramonėje ši metodika buvo adaptuota daugelyje skirtingų sričių, apimant ir programų sistemų kūrimą [15].

Siekiant užtikrinti sisteminių sudėtingų sistemų dekomponavimo procedūrą, DK metode išlaikomi pagrindiniai KFS metodikos principai ir specializuota bei išplėsta KFS kokybės korta: dekomponavimo kortoje aprašomos norimo dekomponuoti sistemos elemento savybės ir sudedamosios dalys. Be to, kortose turi būti tiek savybių, tiek sudedamųjų dalių kiekybiniai atributai. Galiausiai, perėjimas į žemesnį lygmenį atliekamas kiekvienai sudedamajai daliai sukuriant po naują dekomponavimo kortą.



1 pav. Dekomponavimo kortos dalys.

1 pav. pateiktos kortos vaizduoja konkrečios sistemos arba jos sudedamosios dalies dekompoziciją. Prieš atliekant dekompoziciją reikalavimų inžinieriai pateikia sąrašą reikalavimų, kuriuos turi tenkinti dekomponuojamas elementas (savybių sąrašas). Tada, atsižvelgdamas į reikalavimų rinkinį,

architektas parengia architektūrinį sprendinį: dekomponuoja elementą į sudedamąsias dalis (gaunamas sudedamųjų dalių sąrašas). Kiekvienai tokiai sudedamajai daliai architektas priskiria atsakomybę realizuoti konkrečius reikalavimus (atsakomybių matrica) bei užpildo likusius kortos atributus. Perėjimas į žemesnį abstrakcijos lygmenį vykdomas kiekvienai sudedamajai daliai sukūrus atskirą dekomponavimo kortą ir naują reikalavimų rinkinį (savybių sąrašas).

1 pav. taip pat yra pavaizduoti papildomi dekomponavimo kortos atributai. Kairėje pusėje yra pateikta abstrakti kortos struktūra, kurioje atributai suskirstyti į grupes, o dešinėje pusėje pateikta detali kortos struktūra. Kadangi straipsnyje nagrinėjamas dekompozicijos struktūrinis vertinimas, šiame kontekste aktualūs yra tik dalis kortos atributų. Šią aibę sudaro: sudedamosios dalys (sudedamųjų dalių sąrašas), ryšių matrica, sudedamųjų dalių atributai (ryšių kiekis, ryšių verčių suma, ryšių verčių vidurkis) ir dekomponuojamo elemento atributai (sudedamųjų dalių kiekis, ryšių kiekis, ryšių verčių suma, ryšių verčių vidurkis). Išvardinti atributai dešinėje kortoje (žr. 1 pav.) paryškinti išsiskiriančia spalva.

## 4 Dekompozicijos vertinimas ir pasirinkimas

Sistemų sudėtingumui matuoti naudojami įvairūs matai. Vienas pagrindinių informacinių sistemų sudėtingumo matų yra struktūrinis (dar vadinamas topologiniu) sudėtingumas. Tą sąlygoja šiandienos inžinerinių sistemų sudėtingumas, glūdintis jų persipynusioje architektūroje, kurią įgyvendina daug sudedamųjų dalių ir jų tarpusavio sąveika. Todėl, norint įvertinti sudėtingą sistemą arba palyginti ją su kitomis sistemomis, skaitinis jos struktūrinio sudėtingumo vertinimas yra privalomas.

Struktūrinis sudėtingumas nusakomas nagrinėjant sistemos struktūrą ir kompoziciją. Detaliau aptarsime dekomponavimo kortų metode naudojamus dekompozicijos struktūrinio sudėtingumo matus, nusakomus posistemų kiekiu ir jų tarpusavio sąveika.

*Posistemų kiekis.* Posistemų kiekis turi tiesioginę įtaką sistemos sudėtingumui. Jeigu sistema vienu būdu būtų išskaidyta į 2, o kitu būdu į 5 vienodo sudėtingumo dalis, antras dekompozicijos būdas lemtų aukštesnį sistemos sudėtingumą. Didelis sudedamųjų dalių kiekis sąlygoja daugiau potencialių tarpusavio ryšių, daugiau kūrimo ir palaikymo pastangų.

Natūralu, kad dekompozicija nėra prasminga, jeigu posistemų kiekis yra lygus 0 arba 1. Posistemų kiekiui esant lygiam 0, tai reiškia, kad sistemai dekompozicija nėra taikoma ir ji yra traktuojama kaip atominis neskaidomas elementas. Jeigu posistemų kiekis lygus 1, tuomet, neprarandant informacijos, ji gali būti dekomponuojama tik į save pačią.

*Posistemų tarpusavio ryšiai.* Tai aspektas, nusakantis kaip konkreti posistemė sąveikauja su kitomis posistemėmis. Nuo jo tiesiogiai priklauso visos sistemos sudėtingumas. DK metodas matuoja tarpusavio ryšius dviejuose skirtinguose kontekstuose: elemente, kuris dekomponuotas ir kiekvienoje elemento sudedamojoje dalyje. Matuojant tarpusavio sąveikos aspektą galima naudoti kelis matus.

Pirma, tai ryšių kiekis. Kuo daugiau struktūrinis elementas turės ryšių, tuo jis bus sudėtingesnis: reikalaus daugiau integracijų kūrimo, transakcijų apdorojimo, testavimo ir palaikymo pastangų. Kadangi posistemės viena su kita palaiko ryšį interfeisų pagalba, jų pokyčio rizika ir kaina smarkiai išauga dėl priklausomybių sukkelto inertiškumo. Tačiau visiškai išvengti tarpusavio ryšių negalima. Kiekviena posistemė turi turėti bent vieną ryšį su kita posisteme, kitaip ji negalėtų būti sistemos dalis – tokiu atveju ši posistemė būtų atskira nepriklausoma sistema. Ryšių kiekis gaunamas taikant formules (1) – sudedamosios dalies kontekste ir (2) – dekomponuojamo elemento kontekste:

$$f(x) = \sum_{i=1}^n 1, \text{ tiems } i, \text{ kuriems } x \neq x_i \text{ ir } C(x, x_i) > 0, n > 0 \quad (1)$$

ir

$$\alpha = \frac{\sum_{i=1}^n f(x_i)}{2}, n > 0. \quad (2)$$

Čia  $f(x)$  – sudedamosios dalies  $x$  ryšių su kitomis dalimis kiekis;  $\alpha$  – bendras ryšių kiekis dekomponuotame elemente;  $n$  – sudedamųjų dalių kiekis;  $C(x, x_i)$  – tarp sudedamųjų dalių  $x$  ir  $x_i$  egzistuojantis ryšio stiprumas.

Antra, tai ryšio stiprumo vertinimas. Ryšys, kuriuo dvi posistemės viena kitai asinchroniškai perduoda gigabaitus informacijos per sekundę užtikrinant maksimalų konfidencialumą bei informacijos nepraradimą stipriai prisidės prie sistemos sudėtingumo. Tačiau ryšys, kuriuo viena posistemė kitai kartą per parą perduos savo būseną *true* arba *false* formatu, menkai paveiks ben-

drą sistemos sudėtingumą. Sudedamųjų dalių tarpusavio sąveika DK metode vertinama trimis skirtingais tipais: neegzistuojančiu, silpnu ir stipriu.

Ryšio stiprumo matas leidžia įvertinti jo intensyvumą. Šis matas taip pat pravartus kai ryšių kiekis skirtingose dalyse yra vienodas – tokiu atveju sudėtingesnė dalis bus ta, kurios ryšių stiprumų vertės aukštesnės. Ryšių stiprumų suma gaunama taikant formules (3) – sudedamosios dalies kontekste ir (4) – dekomponuojamo elemento kontekste:

$$g(x) = \sum_{i=1}^n C(x, x_i), \text{ tiems } i, \text{ kuriems } x \neq x_i, n > 0 \quad (3)$$

ir

$$\beta = \frac{\sum_{i=1}^n g(x_i)}{2}, n > 0. \quad (4)$$

Čia  $g(x)$  – sudedamosios dalies  $x$  ryšių su kitomis dalimis stiprumų suma;  $\beta$  – bendra ryšių stiprumų suma dekomponuotame elemente;  $n$  – sudedamųjų dalių kiekis;  $C(x, x_i)$  – tarp sudedamųjų dalių  $x$  ir  $x_i$  egzistuojantis ryšio stiprumas.

Dar vienas matas, susijęs su ryšių stiprumu, yra sudedamosios dalies ryšių stiprumo vidurkis. Jis leidžia įvertinti, kokie yra vyraujantys sudedamosios dalies ryšiai ir yra gaunamas taikant formulę (5):

$$h(x) = \frac{f(x)}{g(x)}. \quad (5)$$

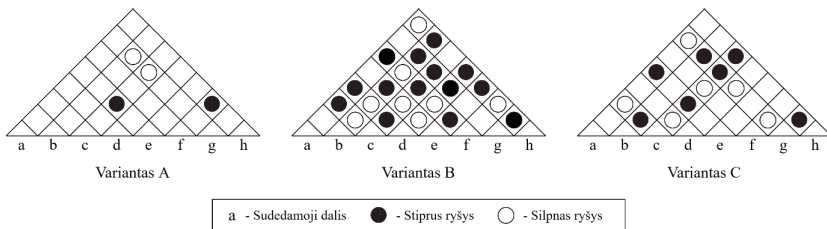
Čia  $h(x)$  – sudedamosios dalies  $x$  ryšių stiprumo vidurkis;  $g(x)$  – sudedamosios dalies  $x$  ryšių su kitomis dalimis stiprumų suma;  $f(x)$  – sudedamosios dalies  $x$  ryšių su kitomis dalimis kiekis.

Paskutinis matas, susijęs su ryšių stiprumu, yra visų dekomponuojamo elemento sudedamųjų dalių ryšių kiekio rangas. Šis atributas leidžia identifikuoti neproporcingai sudėtingas ir inertiškas elemento sudedamąsias dalis, apskaičiuojant skirtumą tarp didžiausių ir mažiausių ryšių kiekių turinčių dalių. Atributo dydis gaunamas taikant formulę (6):

$$\gamma = \max_{vi \in n} f(x_i) - \min_{vi \in n} f(x_i), [n] = \{1, \dots, n\}. \quad (6)$$

Čia  $\gamma$  – ryšių kiekio rangas;  $f(x)$  – sudedamosios dalies  $x$  ryšių su kitomis dalimis kiekis;  $n$  – sudedamųjų dalių kiekis.

Visi prieš tai aprašyti posistemų tarpusavio ryšių grupės atributai gautami iš ryšių matricos bei prisideda prie tendencijų atpažinimo. 2 pav. pateikiamas pavyzdys, t. y. trys skirtingi DK metodo ryšių matricos užpildymo variantai: A variante dauguma sudedamųjų dalių nesąveikauja tarpusavyje; B variante yra daug tarpusavio ryšių, iš kurių stiprių ryšių yra dauguma; C variante dvi sudedamosios dalys (c ir g) turi daugumą ryšių.



**2 pav.** Ryšių matricos užpildymo pavyzdžiai.

Naudojant DK metodą ir įvertinus dekompozicijas, galima pasirinkti tinkamą alternatyvą. Dekompozicijos alternatyvos yra lyginamos gretinant elemento atributų grupę (žr. 1 pav.), į kurią patenka šie struktūrinio sudėtingumo vertinimo atributai: dekomponuotų dalių kiekis, ryšių kiekis, ryšių verčių suma ir ryšių kiekio rangas. Konkretūs nurodymai, kurie architektui leistų vienareikšmiškai pasirinkti dekompoziciją, šio darbo kontekste nėra detalizuojami.

## 5 Išvados

Darbe yra pateiktas dekomponavimo kortų metodas sudėtingoms informacinėms sistemoms dekomponuoti ir toms dekompozicijoms vertinti. Metodas įgyvendina hierarchinį dekomponavimo būdą, kur pirmame lygmenyje identifikuojami sistemos elementai (posistemės), o žemesniame lygmenyje – tų elementų sudedamosios dalys. Kitaip sakant, sudėtingų informacinių sistemų dekomponavimo metu pirmajame lygmenyje nereikia taikyti dekomponavimo būdų (kitais žodžiais, rinktis konkretų architektūros stilių), kurie jau reiškia konkrečius techninius ribojimus. Gautos dekompozicijos tinkamumas nurodytų kriterijų ir sudėtingumo atžvilgiu turi būti vertinamas analizuojant atitinkamas kortos dalis, užpildomas pagal metodo taisykles.



Tai leidžia gauti skaitinį dekompozicijos struktūrinio sudėtingumo įvertį, kas yra privaloma, norint palyginti skirtingas dekompozicijas. Naudojant dekomponavimo kortų metodą, skirtingų dekomponavimo alternatyvų palyginimas vykdomas tarpusavyje gretinant šiuos atributus: dekomponuotų dalių kiekį, ryšių kiekį, ryšių verčių sumą ir ryšių kiekio rangą.

## Literatūra

- [1] Komoto, H., & Tomiyama, T. (2012). A framework for computer aided conceptual design and its application to system architecting of mechatronics products. *Computer-Aided Design*, 44(10), 931-946.
- [2] Colombo, P., Khendekb, F., & Lavazza, L. (2012). Bridging the gap between requirements and design: an approach based on problem frames and SysML. *Journal of Systems and Software*, 85(3), 717-745.
- [3] Holmgren, J., Persson, J., & Davidsson, P. (2008). Agent based decomposition of optimization problems. *First International Workshop on Optimization in Multi-Agent Systems*.
- [4] Löwy, J. (2019). *Righting software*. Addison-Wesley Professional.
- [5] Chiriac, N. (2011). Three approaches to complex system decomposition. *DSM 2011: Proceedings of the 13th International DSM Conference 2011*.
- [6] Koopman, P.J.A. (1995). A taxonomy of decomposition strategies based on structures, behaviours, and goals. *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 17179, 611-618.
- [7] Sheard S.A., & Mostashari A. (2010). A complexity typology for systems engineering. *Systems Engineering*. INCOSE International Symposium, 20(1), 933-945.
- [8] Sinha, K., & De, W.O.L. (2013). A network-based structural complexity metric for engineered complex systems. *IEEE International Systems Conference*, 426-430.
- [9] Luo A. (2019). A structural complexity metric method for complex information systems. *JSW*, 14(7), 332-339.
- [10] Sinha, K., & de Weck, O. L. (2016). Matrix energy as a measure of topological complexity of a graph. *arXiv preprint arXiv:1608.08456*.
- [11] Hornby G. (2007). Modularity, reuse, and hierarchy: measuring complexity by measuring structure and organization. *Complexity*, 13(2), 50-61.
- [12] Tamaskar, S., Neema, K., & DeLaurentis, D. (2014). Framework for measuring complexity of aerospace systems. *Research in Engineering Design*, 25(2), 125-137.
- [13] Eppinger, S.D., Whitney, D.E., Smith, R.P., & Gebala, D.A. (1994). A model-based method for organizing tasks in product development. *Research in Engineering Design*, 6(1), 1-13.
- [14] Akao, Y. (2004). *QFD: quality function deployment – integrating customer requirements into product design*. Productivity Press.
- [15] Chan, L.K., & Wu, M.L. (2002). Quality function deployment: a literature review. *European journal of operational research*, 143(3), 463-497.