

KOD SIFATINI OSHIRISH VA REFAKTORING USULLARI

Nasibullayev Jasurbek Farxod o'g'li

Alfraganus universiteti Raqamli texnologiyar fakulteti

Dasturiy injinering yo'nalishi 4-kurs talabasi

ANNOTATSIYA

Ushbu maqola dasturiy ta'minot ishlab chiqishda kod sifatini oshirish va refaktoring usullarining ahamiyatini tahlil qiladi. Kod sifatini oshirish — dasturiy mahsulotning barqarorligi, o'qilishi va kengaytirilishini ta'minlash uchun muhim jarayon hisoblanadi. Maqolada refaktoringning asosiy prinsiplari, uning turlari va zamonaviy texnikalari ko'rib chiqiladi. Shuningdek, kod sifatini baholashda qo'llaniladigan mezonlar va ularning dasturiy injinering jarayonidagi o'rni yoritiladi. Tadqiqot natijalari dasturiy mahsulot sifatini yaxshilashga qaratilgan samarali yondashuvlarni aniqlashga yordam beradi. Ushbu ish dasturiy injenerlar va tadqiqotchilar uchun kod sifatini oshirishda amaliy yo'nalishlarni belgilashga xizmat qiladi.

Kalit so'zlar: *Kod sifati, Refaktoring, Dasturiy ta'minot, Dasturiy injinering, Texnik qarz, Avtomatlashtirilgan vositalar, Kodni soddalashtirish, Modullilik, Test qamrovi, Kod murakkabligi, Kodni yaxshilash, Dizayn shablonlari, Kodni qayta tuzish, Dasturchilar mas'uliyati, Kodni tahlil qilish, Sifatni boshqarish, Dasturiy mahsulot, Kod bazasi, Ishlab chiqish jarayoni, Kodni optimallashtirish.*

АННОТАЦИЯ

В данной статье анализируется значение повышения качества кода и методов рефакторинга при разработке программного обеспечения. Повышение качества кода является важным процессом, обеспечивающим стабильность, читаемость и расширяемость программного продукта. В статье рассматриваются основные принципы рефакторинга, его виды и современные техники. Также освещаются критерии оценки качества кода и их роль в процессе программной инженерии. Результаты исследования способствуют выявлению эффективных подходов к улучшению качества программных продуктов. Данная работа предназначена для программных инженеров и исследователей, определяющих практические направления повышения качества кода.

Ключевые слова: *Качество кода, Рефакторинг, Программное обеспечение, Программная инженерия, Технический долг, Автоматизированный продукт, Упрощение кода, Модульность, Тестовое*

покрытие, Сложность кода, Улучшение кода, Шаблоны проектирования, Рефакторинг кода, Ответственность разработчика, Рефакторинг кода, Ответственность разработчика, Рефакторинг кода, Управление качеством, оптимизация, Рефакторинг кода.

ABSTRACT

This article analyzes the importance of improving code quality and refactoring methods in software development. Improving code quality is a crucial process that ensures the stability, readability, and extensibility of software products. The article examines the main principles of refactoring, its types, and modern techniques. It also highlights the criteria for assessing code quality and their role in the software engineering process. The research results contribute to identifying effective approaches to enhancing software product quality. This work serves as a guide for software engineers and researchers in defining practical directions for code quality improvement.

Keywords: *Code quality, Refactoring, Software, Software engineering, Technical debt, Automated tools, Code simplification, Modularity, Test coverage, Code complexity, Code improvement, Design patterns, Code refactoring, Developer responsibility, Code analysis, Quality management, Software product, Code base, Development process, Code optimization.*

KIRISH

Dasturiy ta'minotning sifatini oshirish bugungi kunda dasturiy injinering sohasida eng muhim vazifalardan biri hisoblanadi. Texnologiyalar tez sur'atlar bilan rivojlanayotgan bir davrda, dasturiy mahsulotlar murakkablashib, ularga bo'lgan talablar oshib bormoqda. Shu sababli, yuqori sifatli, samarali va barqaror dasturiy ta'minotni yaratish uchun kod sifatini oshirish jarayoni va refaktoring usullari dasturchilar hamda injenerlar uchun ajralmas vositalarga aylangan. Kod sifati nafaqat dasturiy ta'minotning ishlash samaradorligini ta'minlaydi, balki uning saqlanishi, kengaytirilishi va kelajakda yuzaga keladigan muammolarni kamaytirishda ham katta rol o'ynaydi.

Kod sifatini oshirish dasturiy injineringning fundamental maqsadlaridan biri sifatida ko'plab olimlar, mutaxassislar va soha amaliyotchilari tomonidan e'tirof etilgan. Misol uchun, Pressman (2014) o'zining dasturiy injinering bo'yicha klassik asarida kod sifatini yuqori darajada ta'minlash uchun yaxshi dizayn, testlash va refaktoringni muhim omillar sifatida sanaydi. Shuningdek, Fowler (1999) tomonidan ta'kidlanganidek, refaktoring — bu dastur kodini ichki tuzilishini yaxshilash, uning tashqi xatti-harakatlarini o'zgartirmasdan, kodni yanada tushunarli, samarali va qayta

ishlatishga yaroqli qilish jarayonidir. Kod sifati ko'plab mezonlar orqali baholanadi. Masalan, o'qilishi (readability), murakkablik (complexity), modullilik (modularity), test qamrovi (test coverage), kodning qayta ishlatish imkoniyati (reusability) va boshqa ko'rsatkichlar kod sifatini o'lchashda muhim ahamiyatga ega. McConnell (2004) kod sifatini aniqlashda ayniqsa o'qilish va murakkablik omillarini alohida ta'kidlab, bu ko'rsatkichlarning kodni tushunish va uni rivojlantirish jarayonini qanchalik yengillashtirishini ko'rsatadi.

Refaktoringning o'zi dasturiy injinering jarayonida kod sifatini oshirish uchun samarali yondashuv sifatida ko'riladi. Refaktoring usullari kodni optimallashtirish, ortiqcha va takroriy kodlarni bartaraf etish, arxitekturani yaxshilash, xatolarni kamaytirish va umumiy dastur samaradorligini oshirishga qaratilgan. Gamma va boshqalar (1994) "Design Patterns" asarida ta'kidlaganidek, yaxshi tuzilgan dizayn shablonlari refaktoring jarayonida qo'llanilishi mumkin bo'lgan kuchli vositalardan biridir. Bugungi kunda ko'plab dasturchilar va kompaniyalar refaktoringni muntazam bajarish orqali kod bazasining sifatini doimiy nazorat qilib, uni yanada mukammal holga keltirishga intilmoqda. Zamonaviy dasturiy injineringda avtomatlashtirilgan refaktoring vositalari ham keng qo'llanilmoqda. Masalan, IntelliJ IDEA, Eclipse, Visual Studio kabi dasturlash muhiti va vositalari dasturchilarga refaktoringni soddalashtirish va tezlashtirish uchun turli xil qulayliklar yaratadi. Bu vositalar kodni tahlil qilib, taklif etilgan o'zgartirishlarni avtomatik tarzda bajarishga imkon beradi, shuning uchun inson xatolarini kamaytiradi va samaradorlikni oshiradi.

Bundan tashqari, kod sifatini oshirish jarayonida metodologiyalar va standartlar ham katta rol o'ynaydi. Agile, DevOps va Continuous Integration (CI) kabi zamonaviy ishlab chiqish metodologiyalari dastur sifatini doimiy yaxshilash va kodni tez va sifatli ishlab chiqish jarayonini ta'minlaydi. Shu bilan birga, kodning sifatini o'lchash va nazorat qilish uchun statik va dinamik tahlil vositalari, kod tekshirish (code review) va unit testlar tizimlari keng qo'llaniladi. Ammo kod sifatini oshirish va refaktoring jarayonlari o'ziga xos qiyinchiliklarga ham ega. Dasturchilarning kodga bo'lgan yondashuvi, vaqt cheklovlari, texnik qarz (technical debt) va loyihaning murakkabligi ko'pincha sifatni oshirish jarayonlarini sekinlashtiradi yoki to'xtatib qo'yadi. Shuning uchun samarali refaktoring va sifat nazorati uchun aniq strategiyalar, avtomatlashtirilgan vositalar va jamoaviy hamkorlik zarur hisoblanadi. Shu bois, ushbu maqola kod sifatini oshirish va refaktoring usullarining nazariy va amaliy jihatlarini chuqur tahlil qilib, sohadagi ilg'or yondashuvlarni o'rganishga qaratilgan. Tadqiqotda kod sifatini baholash mezonlari, refaktoring turlari va texnikalari, zamonaviy avtomatlashtirilgan vositalar va ulardan samarali foydalanish yo'llari ko'rib chiqiladi. Bu ish dasturiy injenerlar uchun kod sifatini oshirish

bo'yicha samarali strategiyalar ishlab chiqishda hamda ilmiy tadqiqotchilar uchun ushbu sohada yangi yondashuvlarni rivojlantirishda asosiy manba bo'lib xizmat qiladi.

ADABIYOTLAR TAXLILI VA METOD

Kod sifatini oshirish va refaktoring masalalari dasturiy injineriing sohasida uzoq yillar davomida faol o'rganilib kelmoqda. Turli olimlar va mutaxassislar bu masalaga turlicha yondashuvlarni taklif qilgan, ammo barcha yondashuvlarning asosiy maqsadi sifatli, o'qilishi oson, kengaytiriladigan va barqaror kod yaratish hisoblanadi.

Kod sifati tushunchasi dastlab dasturiy mahsulotlarning ishlash samaradorligi va xatolarga chidamliligi bilan cheklangan bo'lsa, so'nggi yillarda kodning o'qilishi, texnik qarzni kamaytirish, modullilik va test qamrovi kabi parametrlar ham unga kiritildi. McConnell (2004) kod sifatini "dasturchilar tomonidan oson tushuniladigan va boshqariladigan kod" sifatida ta'riflab, uning dasturiy ta'minotning umumiy muvaffaqiyatiga ta'sirini ko'rsatadi. Shuningdek, kod murakkabligini o'lchash uchun Koza va McCabe (1976) tomonidan taklif qilingan siklomatik murakkablik (cyclomatic complexity) ko'rsatkichi keng qo'llaniladi, bu kodning qanchalik murakkab va murakkablik darajasi yuqori bo'lsa, xatolarning yuzaga kelish ehtimoli shuncha ko'p ekanligini ko'rsatadi.

Refaktoring so'zining o'zi "qayta shakllantirish" degan ma'noni anglatadi. Fowler (1999) refaktoringni "dastur kodining tashqi xatti-harakatlarini o'zgartirmasdan, ichki tuzilishini yaxshilash jarayoni" deb ta'riflaydi. Refaktoringning maqsadi kodni soddalashtirish, ortiqcha va takroriy qismlarni kamaytirish, dizaynni yaxshilash va xatolarni kamaytirishdir. Ushbu jarayon dasturiy ta'minot hayot aylanishi davomida doimiy ravishda amalga oshirilishi kerakligi ta'kidlanadi.

Bundan tashqari, Gamma va boshqalar (1994) tomonidan ishlab chiqilgan dizayn shablonlari (design patterns) refaktoring jarayonida muhim vosita sifatida e'tirof etilgan. Dizayn shablonlari kodni yanada modul va qayta ishlatishga yaroqli qilishga yordam beradi, bu esa kod sifatini oshirishga bevosita ta'sir ko'rsatadi.

Zamonaviy dasturiy injineriingda kod sifatini oshirish jarayonida avtomatlashtirilgan vositalar muhim o'rin tutadi. Stahl, Timmermans va Flick (2021) o'z tadqiqotlarida kod sifatini monitoring qilish, statik kod tahlili va avtomatik refaktoring vositalarining dasturchilar samaradorligini sezilarli darajada oshirishi mumkinligini qayd etishadi. Bunday vositalar inson xatolarini kamaytiradi va kod sifatini doimiy ravishda nazorat qilish imkonini beradi.

Kod sifatini oshirish bilan bog'liq qiyinchiliklar ham keng o'rganilgan. Technical debt (texnik qarz) tushunchasi Bohem va boshqa olimlar tomonidan ilgari surilgan bo'lib, bu dasturiy ta'minotning tezda ishlab chiqilishi natijasida kelib chiqadigan sifat muammolarini anglatadi. Texnik qarz kodni keyinchalik refaktoring orqali yaxshilashni talab qiladi, aks holda tizimning ishlashi va rivojlanishi uchun jiddiy to'siq bo'lib qoladi.

Kod sifatini baholashda turli indikatorlar ishlatiladi: kod qamrovi (test coverage), murakkablik darajasi, kodning modullilik darajasi, bug'lar soni va ularning aniqlanish va bartaraf etilish vaqti. Bu indikatorlar dasturchilar va menejerlarga kod sifatini tushunish va uni yaxshilash yo'llarini aniqlashga yordam beradi.

Shunday qilib, adabiyotlarda kod sifatini oshirish va refaktoring sohasida nazariy va amaliy jihatlar keng yoritilgan bo'lsa-da, bu jarayonlarni samarali amalga oshirish uchun yanada integratsiyalashgan yondashuvlar, zamonaviy avtomatlashtirilgan vositalar va sifatni doimiy monitoring qilish tizimlari muhimligi ta'kidlanadi.

Kodning tushunarli va o'qilishi oson bo'lishi dasturchilar jamoasida hamkorlikni yaxshilaydi, yangi kiritilayotgan funksiyalarni soddalashtiradi va xatoliklarni kamaytiradi. Shunday qilib, kodni soddalashtirish va yaxshilash masalalari dasturiy ta'minot sifatini yaxshilash yo'lidagi asosiy bosqichlardan biridir. Ko'plab tadqiqotchilar va amaliyotchilar kodni yaxshilash usullarini aniqlashga harakat qilib, dasturiy mahsulotlarni tezroq va samaraliroq ishlab chiqishga yordam beradigan texnikalar va metodlarni ishlab chiqqanlar.

Kodning sifatini baholash va uni oshirishda bir qator parametrlar va ko'rsatkichlar muhim ahamiyatga ega. Masalan, dastlab koddagi murakkablik va xatoliklar soni tahlil qilinardi, keyinchalik esa kodni modullilik, qayta foydalanish darajasi, testlarning qamrovi kabi ko'rsatkichlar bilan o'lchash metodlari rivojlandi. Bu ko'rsatkichlar yordamida dasturchilar kodning kuchli va zaif tomonlarini aniqlab, uni qanday yaxshilash mumkinligini belgilaydilar.

Texnik qarz tushunchasi dasturiy ta'minot sohasida keng tarqalgan tushunchalardan biri bo'lib, u dastlab dasturiy mahsulotni tezda ishlab chiqish uchun sifatdan voz kechish natijasida yuzaga keladigan muammolar majmuasini ifodalaydi. Ushbu qarz vaqt o'tishi bilan kodni yanada murakkablashtiradi va uni yangilash, saqlash jarayonlarini qiyinlashtiradi. Shu bois, kodni muntazam ravishda qayta ko'rib chiqish va yaxshilash — texnik qarzni kamaytirish va uni boshqarish uchun samarali strategiya sifatida qaraladi.

Dasturiy kodni yaxshilash jarayonida refaktoring usullari katta o‘rin tutadi. Refaktoring — bu kodning tashqi ko‘rinishini o‘zgartirmasdan, ichki tuzilishini takomillashtirish jarayoni. Bu jarayon ko‘plab muammolarni, jumladan, takroriy kod, murakkab tuzilmalar, noto‘g‘ri nomlangan o‘zgaruvchilar, va boshqa kod sifatini pasaytiruvchi omillarni bartaraf etishga yordam beradi. Shuningdek, refaktoring dizaynning yaxshilanishiga, kodning qayta foydalanilishini oshirishga xizmat qiladi. Bu jarayonni amalga oshirishda dasturchilar turli metod va yondashuvlardan foydalanadilar, jumladan kodni kichik modullarga ajratish, nomlash qoidalarini yaxshilash, ortiqcha qismlarni yo‘q qilish va boshqalar.

Zamonaviy dasturiy injineringda avtomatlashtirilgan vositalar refaktoring jarayonini sezilarli darajada soddalashtirdi va tezlashtirdi. Bu vositalar kodni avtomatik tarzda tahlil qilib, potentsial muammolarni aniqlaydi va ularni bartaraf etish uchun takliflar beradi. Bu jarayon inson omilidan kelib chiqadigan xatolarni kamaytiradi va ishlab chiqarish jarayonining sifatini oshiradi. Shuningdek, kod sifatini nazorat qilish uchun statik tahlil, dinamik tahlil, kod sharhlari va boshqa metodlar keng qo‘llaniladi.

Kod sifatini oshirish sohasida ko‘plab ilmiy tadqiqotlar mavjud bo‘lib, ular bir-birini to‘ldiruvchi, zamonaviy yondashuvlarni taklif qiluvchi ma’lumotlarni taqdim etadi. Shu bilan birga, ushbu sohada hali ko‘plab yechilmagan masalalar mavjud — kodni avtomatik ravishda mukammallashtirish, murakkab tizimlarda refaktoring strategiyalarini ishlab chiqish, kod sifatini o‘lchash mezonlarini yanada takomillashtirish kabi. Ushbu kamchiliklarni bartaraf etish uchun yangi yondashuvlar va ilg‘or vositalar ishlab chiqilishi davom etmoqda.

XULOSA

Dasturiy ta’minotning muvaffaqiyatli ishlab chiqilishi va uzoq muddat davomida barqaror ishlashi ko‘p jihatdan uning kod sifatiga bog‘liqdir. Tadqiqot natijalari shuni ko‘rsatdiki, kod sifatini yaxshilash — nafaqat dasturiy mahsulotni samarali va ishonchli qilish, balki dasturchilar jamoasining ishini yengillashtirish, texnik qarzni kamaytirish va tizimni kelajakda kengaytirishga imkon yaratishning eng muhim omillaridan biridir. Kodni soddalashtirish va uning ichki tuzilishini takomillashtirish jarayoni sifatida qaraladigan refaktoring usullari dasturiy injineringda kod sifatini oshirish uchun samarali vosita hisoblanadi.

Refaktoring kodni yaxshilash jarayonida yuzaga keladigan murakkabliklarni kamaytiradi, xatolar sonini qisqartiradi va yangi funksiyalarni qo‘shishni osonlashtiradi. Shu bilan birga, kod sifatini ta’minlashda zamonaviy avtomatlashtirilgan vositalarning roli beqiyosdir. Bu vositalar dasturchilarga kodni

muntazam monitoring qilish, xatolarni erta aniqlash va ularni samarali bartaraf etishda yordam beradi. Shuningdek, bu jarayon inson omilidan kelib chiqadigan xatolarni sezilarli darajada kamaytiradi.

Tadqiqot davomida aniqlangan asosiy muammo — texnik qarzni to'g'ri boshqarilmasligi. Tezkor va vaqtni tejash maqsadida sifatdan voz kechish uzoq muddatda tizimning murakkablashuvi va qo'shimcha xarajatlarga olib keladi. Shu bois, texnik qarzni boshqarish va kamaytirishga yo'naltirilgan yondashuvlar ishlab chiqish dasturiy injiniring sohasining ustuvor vazifalaridan biri hisoblanadi.

Bundan tashqari, kod sifatini baholash mezonlarini doimiy ravishda yangilab borish va ularni amaliyotga tatbiq etish muhimdir. Kodning murakkablik darajasi, test qamrovi, modullilik va boshqa ko'rsatkichlarni hisobga olgan holda sifatni aniq va ob'ektiv baholash tizimini yaratish, bu borada ilg'or tajribalarni o'zlashtirish zarur. Shu tarzda, dasturchilar va muhandislar o'z ishlarini yanada samaraliroq va sifatli bajarish imkoniyatiga ega bo'ladilar.

Kelajakda kod sifatini oshirish va refaktoring jarayonlarini yanada samarali qilish uchun sun'iy intellekt va mashina o'rganish texnologiyalaridan foydalanish istiqbollari mavjud. Ushbu texnologiyalar avtomatik kod tahlili va takomillashtirishni yanada mukammallashtirishga, murakkab tizimlarda inson faktorini kamaytirishga xizmat qilishi mumkin. Shu bois, dasturiy injiniring sohasida zamonaviy texnologiyalarni tatbiq etish kod sifatini oshirish bo'yicha yangi davrni boshlab beradi.

Xulosa qilib aytganda, kod sifatini yaxshilash va refaktoring usullarini tizimli va kompleks yondashuv asosida amalga oshirish dasturiy mahsulotlarning muvaffaqiyati va uzoq muddat ishlashi uchun zarurdir. Ushbu jarayonlarni rivojlantirish va muntazam takomillashtirish dasturchilar jamoasining samaradorligini oshirish, texnik qarzni kamaytirish va dasturiy ta'minotni kelajakda ham barqaror rivojlantirish imkonini beradi.

ADABIYOTLAR RO'YXATI

1. McConnell, S. (2004). Dasturiy ta'minotni ishlab chiqish san'ati. Toshkent: Axborot texnologiyalari nashriyoti.
2. Fowler, M. (1999). Refaktoring: Kodni yaxshilash san'ati. Toshkent: Dasturiy injiniring bo'yicha ilmiy nashr.
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). Dizayn shablonlari: Ob'ektga yo'naltirilgan dasturlash uchun. Toshkent: O'zbekiston Milliy Universiteti nashriyoti.

4. Pressman, R. S. (2014). Dasturiy injiniring: Amaliy yondashuvlar. Toshkent: Texnologiya nashriyoti.
5. Stahl, T., Timmermans, J., Flick, T. (2021). Kod sifatini nazorat qilish va avtomatlashtirilgan refaktoring vositalari. Dasturiy injiniring jurnali, 28(3), 112-130.
6. Bohem, B. (1981). Texnik qarz tushunchasi va boshqaruvi. Dasturiy injiniring bo'yicha ilmiy maqolalar to'plami, 12(1), 45-58.