SCIENCE HUB REVIEW
2026
Volume 1
Issue 5
www.sciencehubreview.uz
ILMIY ELEKTRON JURNAL

DASTURIY TA’MINOT VA UNING RIVOJLANISH TENDENSIYALARI

Muxtorov Farhodjon Furqatjon ogli

T.D.I.U.T.F buxgalteriya hisobi yo‘nalishi bo‘yicha 1-kurs

Annotatsiya

Maqolada dasturiy ta’minotning mohiyati, tasnifi va rivojlanish bosqichlari tahlil qilinadi, zamonaviy tendensiyalar (agile va DevOps madaniyati, bulutli va mikroservis arxitekturalar, sun’iy intellektga asoslangan dasturlash vositalari, low-code/no-code platformalar, kiberxavfsizlikka yondashuvning o‘zgarishi) tizimli ko‘rib chiqiladi. Tadqiqot ilmiy manbalar, sohaviy hisobotlar va qiyosiy tahlil asosida bajarildi. Natijada dasturiy ta’minot rivojining beshta asosiy yo‘nalishi aniqlandi va ularning o‘zaro bog‘liqligi ko‘rsatildi. Maqola xulosalari oliy ta’limda dasturiy injiniring kurslarini yangilash va mahalliy IT sohasini rivojlantirish uchun amaliy ahamiyatga ega.

Kalit so‘zlar

dasturiy ta’minot, dasturiy injiniring, agile, DevOps, bulutli hisoblash, mikroservislar, sun’iy intellekt, low-code, kiberxavfsizlik.

Kirish

Bugungi kunda dasturiy ta’minot iqtisodiyot, ta’lim, sog‘liqni saqlash, moliya va davlat boshqaruvining deyarli barcha sohalarida asosiy infratuzilma vazifasini bajaradi. Raqamli transformatsiya jarayonlarida u yordamchi vosita emas, balki tashkilotning raqobatbardoshligini belgilovchi strategik resursga aylangan. O‘zbekistonda "Raqamli O‘zbekiston – 2030" strategiyasi doirasida elektron hukumat xizmatlari, raqamli ta’lim platformalari va IT-parklar faoliyatining kengayishi ham dasturiy ta’minotga bo‘lgan ehtiyojni oshirmoqda.

Shu bilan birga, dasturiy ta’minotni ishlab chiqish usullari tez o‘zgarmoqda. 1970-yillarda keng tarqalgan ketma-ket (waterfall) modeldan (Royce, 1970) iterativ va moslashuvchan yondashuvlarga o‘tish (Beck va boshq., 2001), keyinchalik ishlab chiqish

va ekspluatatsiyani birlashtiruvchi DevOps madaniyati (Forsgren va boshq., 2018) hamda so‘nggi yillarda generativ sun’iy intellekt vositalarining dasturlash jarayoniga kirib kelishi sohada sifat jihatidan yangi bosqichni boshlab berdi. F. Bruksning mashhur tezisiga ko‘ra, dasturiy ta’minot murakkabligining muhim qismi uning tabiatidan kelib chiqadi va yagona "sehrli yechim" mavjud emas (Brooks, 1987). Shu sababli tendensiyalarni alohida emas, balki o‘zaro bog‘liq tizim sifatida o‘rganish dolzarb hisoblanadi.

O‘zbek tilidagi ilmiy adabiyotda dasturiy ta’minot rivojlanishining zamonaviy yo‘nalishlarini bir butun tizim sifatida yoritgan ishlar cheklangan. Mavjud tadqiqotlar asosan alohida texnologiyalarga (masalan, veb-dasturlash yoki ma’lumotlar bazalari) e’tibor qaratadi. Bu ilmiy bo‘shliqni to‘ldirish maqolaning asosiy maqsadini belgilaydi.

Metodlar

Tadqiqot sifat tahlili xarakteriga ega bo‘lib, quyidagi usullardan foydalanildi:

Adabiyotlarni tizimli tahlil qilish. Dasturiy injiniring bo‘yicha asosiy darsliklar, hakamlik qilinadigan jurnal maqolalari va xalqaro tashkilotlar (IEEE, ACM, ISO/IEC) standartlari ko‘rib chiqildi. Manbalar Scopus, Web of Science va Google Scholar bazalaridan "software engineering", "DevOps", "microservices", "AI-assisted programming", "low-code" kalit so‘zlari bo‘yicha tanlandi. Tanlash mezoni: so‘nggi 10 yil ichida nashr etilgan va keng iqtiboslangan ishlar, shuningdek sohaning fundamental asarlari.

Sohaviy hisobotlar kontent-tahlili. Dasturchilar o‘rtasida o‘tkaziladigan yirik so‘rovnomalar va platforma hisobotlari (Stack Overflow Developer Survey, GitHub Octoverse, DORA State of DevOps) ma’lumotlari tendensiyalarni aniqlash uchun qiyosiy o‘rganildi.

Tarixiy-qiyosiy usul. Ishlab chiqish modellari (waterfall, spiral, agile, DevOps) o‘zaro taqqoslandi.

Tizimlashtirish va umumlashtirish. Aniqlangan tendensiyalar tematik guruhlarga ajratildi va o‘zaro bog‘liqligi tahlil qilindi.

Tadqiqotning cheklovi shundaki, sohaviy hisobotlar o‘z-o‘zini baholashga asoslangan so‘rovnomalarga tayanadi va barcha mintaqalarni teng ifodalamasligi mumkin.

Shuningdek, texnologiyalar tez o‘zgargani sababli ayrim xulosalar vaqt o‘tishi bilan yangilanishni talab qiladi.

Natijalar

Dasturiy ta’minotning mohiyati va tasnifi

Dasturiy ta’minot (DT) – hisoblash qurilmalarining ishlashini ta’minlovchi dasturlar, ma’lumotlar va ularga tegishli hujjatlar majmuidir. Funksional vazifasiga ko‘ra u odatda uch asosiy guruhga bo‘linadi:

Guruh	Vazifasi	Misollar
Tizimli DT	Apparat resurslarini boshqarish, boshqa dasturlar uchun muhit yaratish	operatsion tizimlar, drayverlar, kompilyatorlar
Amaliy DT	Foydalanuvchining aniq vazifalarini hal qilish	matn muharrirlari, buxgalteriya tizimlari, ta’lim platformalari
Dasturlash vositalari	Dasturlarni yaratish, sinash va sozlash	IDE, versiyalarni boshqarish tizimlari, test freymvorklari

Zamonaviy sharoitda bu chegaralar xiralashmoqda: bulutli xizmatlar (SaaS, PaaS, IaaS) dasturiy ta’minotni mahsulotdan xizmatga aylantirdi.

Ishlab chiqish yondashuvlarining evolyutsiyasi

Tahlil dasturiy ta’minotni ishlab chiqishda to‘rtta asosiy bosqichni ajratish mumkinligini ko‘rsatdi:

Ketma-ket modellar (1970–1990-yillar). Talablar, loyihalash, kodlash, sinash va qo‘llab-quvvatlash bosqichlari qat’iy tartibda bajariladi. Kamchiligi: o‘zgarishlarga sezgirlik past, xatolar kech aniqlanadi.

Iterativ va spiral modellar (1990-yillar). Xavflarni boshqarish va takroriy takomillashtirish joriy etildi.

Agile yondashuvlar (2001-yildan). Agile manifesti odamlar o‘zaro hamkorligi, ishlaydigan mahsulot, mijoz bilan hamkorlik va o‘zgarishlarga javob berishni ustun qo‘ydi (Beck va boshq., 2001). Scrum va Kanban eng keng tarqalgan amaliyotlarga aylandi.

DevOps va uzluksiz yetkazib berish (2010-yillardan). Ishlab chiqish va ekspluatatsiya jamoalari integratsiyalashdi, build, test va deploy jarayonlari avtomatlashtirildi. Tadqiqotlarga ko‘ra, yetkazib berish tezligi va barqarorligi yuqori

jamoalar tashkiliy ko‘rsatkichlar bo‘yicha ham yaxshi natija beradi (Forsgren va boshq., 2018).

Zamonaviy rivojlanish tendensiyalari

Tahlil natijasida beshta asosiy tendensiya aniqlandi.

1-tendensiya: Bulutli va mikroservis arxitekturalarga o‘tish. Monolit ilovalar o‘rniga mustaqil joriy etiladigan kichik xizmatlardan iborat arxitekturalar keng qo‘llanilmoqda (Newman, 2015). Konteynerlashtirish (Docker) va orkestratsiya (Kubernetes) texnologiyalari bu o‘tishni qo‘llab-quvvatlovchi asosiy infratuzilmaga aylandi. Afzalligi: masshtablash va mustaqil yangilash imkoniyati; qiyinchiligi: tizim murakkabligi va monitoring talablarining oshishi.

2-tendensiya: Sun’iy intellektga asoslangan dasturlash. Katta til modellari asosidagi yordamchilar kod yozish, refaktoring, test yaratish, hujjatlashtirish va xatolarni topishda qo‘llanilmoqda. Dasturchining roli kodni qo‘lda yozishdan natijani yo‘naltirish, tekshirish va arxitektura qarorlarini qabul qilishga siljimoqda. Shu bilan birga, avtomatik yaratilgan kodning to‘g‘riligi, xavfsizligi va litsenziya masalalari inson nazoratini zarur qiladi.

3-tendensiya: Low-code/no-code platformalar. Vizual konstruktorlar dasturlash bo‘yicha chuqur bilimga ega bo‘lmagan mutaxassislarga ("fuqaroviy dasturchilar") ichki ilovalar yaratish imkonini beradi. Bu tez prototiplash uchun qulay, ammo murakkab va yuqori yuklamali tizimlarda moslashuvchanlik cheklanadi.

4-tendensiya: Xavfsizlikning ishlab chiqish jarayoniga integratsiyasi (DevSecOps). Xavfsizlik tekshiruvlari ishlab chiqish oxiriga qoldirilmay, har bir bosqichga kiritilmoqda: kodni statik tahlil qilish, bog‘liqliklar zaifligini skanerlash, ta’minot zanjiri (supply chain) xavfsizligi. Xotira xavfsizligini ta’minlovchi tillarga (masalan, Rust) qiziqish ham shu yo‘nalish bilan bog‘liq.

5-tendensiya: Ochiq manbali dasturiy ta’minot va hamjamiyat ekotizimlari. Zamonaviy ilovalarning katta qismi ochiq kutubxonalar va freymvorklarga tayanadi. Ochiq manba hamkorlikni tezlashtiradi, lekin litsenziyalash va qo‘llab-quvvatlash barqarorligi bo‘yicha xavflarni ham keltirib chiqaradi.

Qo‘shimcha yo‘nalishlar sifatida chekka hisoblash (edge computing), WebAssembly, kvant hisoblashlari uchun dasturiy vositalar va barqaror ("yashil") dasturlash tamoyillarini qayd etish mumkin.

Tendensiyalarning o‘zaro bog‘liqligi

Aniqlangan tendensiyalar bir-biridan ajralgan emas. Bulutli arxitektura DevOps amaliyotlarini talab qiladi, DevOps esa avtomatlashtirish orqali xavfsizlik tekshiruvlarini jarayonga kiritish imkonini beradi. Sun’iy intellekt vositalari ishlab chiqish tezligini oshirib, low-code platformalar bilan birga dasturlash kirish chegarasini pasaytirmoqda. Natijada ishlab chiqish jarayoni tezlashadi, lekin sifat va xavfsizlikni nazorat qilish mas’uliyati ortadi.

Muhokama

Olingan natijalar dasturiy ta’minot sohasidagi asosiy siljish "kod yozish"dan "tizimni loyihalash va boshqarish"ga o‘tayotganini ko‘rsatadi. Bruksning ta’kidlashicha, dasturiy ta’minotdagi muhim qiyinchilik muammoni to‘g‘ri tushunish va modellashtirish bilan bog‘liq (Brooks, 1987). Sun’iy intellekt vositalari kodlash mehnatini yengillashtirsa-da, murakkablikning mohiyatiy qismini yo‘qotmaydi, balki talablarni aniq shakllantirish va natijani tanqidiy baholash ko‘nikmalarining ahamiyatini oshiradi.

Bir tomondan, avtomatlashtirish va past kodli yechimlar ishlab chiqarish unumdorligini oshiradi. Ikkinchi tomondan, bu holat "texnik qarz", xavfsizlik zaifliklari va tushunib bo‘lmaydigan kodning to‘planishi xavfini kuchaytiradi. Shuning uchun muvaffaqiyatli jamoalar uchun texnologiyaning o‘zi emas, balki sifat nazorati madaniyati, kod ko‘rigi (code review) va avtomatlashtirilgan test amaliyotlari hal qiluvchi bo‘lib qolmoqda.

Ta’lim nuqtai nazaridan olingan natijalar dasturiy injiniring o‘quv rejalarini yangilash zarurligini ko‘rsatadi. Talabalarga dasturlash tillari bilan bir qatorda versiyalarni boshqarish, CI/CD, bulutli platformalar, xavfsiz dasturlash va sun’iy intellekt vositalaridan mas’uliyat bilan foydalanish ko‘nikmalarini berish maqsadga muvofiq. Bundan tashqari, kasbiy ingliz tili (ESP) bilimi muhim: sohaning asosiy hujjatlari, kutubxonalar va hamjamiyat muloqoti ingliz tilida bo‘lgani uchun texnik terminologiyani o‘zlashtirish dasturchining kasbiy raqobatbardoshligiga bevosita ta’sir qiladi.

Mahalliy kontekstda dasturiy ta’minot eksporti, davlat xizmatlarini raqamlashtirish va o‘zbek tilini qo‘llab-quvvatlovchi dasturiy mahsulotlar (nutqni aniqlash, mashina tarjimasi, OCR) rivojlanishi istiqbolli yo‘nalish hisoblanadi. Ushbu sohada mahalliy kadrlar tayyorlash va ochiq ma’lumotlar to‘plamlarini shakllantirish alohida e’tiborga loyiq.

Tadqiqotning cheklovlari: birinchidan, tahlil ikkilamchi manbalarga asoslangan; ikkinchidan, texnologiyalar tez o‘zgargani sababli tendensiyalar tez eskirishi mumkin. Kelgusi tadqiqotlarda O‘zbekistondagi IT kompaniyalari va dasturchilari o‘rtasida so‘rovnoma yoki chuqur suhbatlar o‘tkazish orqali mahalliy ma’lumotlar bilan to‘ldirish tavsiya etiladi.

Xulosa

Tadqiqot natijalari quyidagi xulosalarga olib keldi:

1. Dasturiy ta’minotni ishlab chiqish ketma-ket modellardan moslashuvchan (agile) va uzluksiz yetkazib berishga asoslangan DevOps yondashuvlariga evolyutsiya qildi.

2. Zamonaviy rivojlanishning beshta asosiy tendensiyasi aniqlandi: bulutli va mikroservis arxitekturalar, sun’iy intellektga asoslangan dasturlash, low-code/no-code platformalar, DevSecOps va ochiq manbali ekotizimlar.

3. Bu tendensiyalar o‘zaro bog‘liq bo‘lib, ishlab chiqish tezligini oshirish bilan birga sifat va xavfsizlik nazoratiga bo‘lgan talabni kuchaytiradi.

4. Dasturchi kasbi kodlashdan tizim loyihalash, natijani tanqidiy baholash va xavfsizlikni ta’minlashga yo‘nalmoqda, bu esa ta’lim mazmunini yangilashni talab qiladi.

Kelgusi tadqiqotlar mahalliy IT bozori ma’lumotlari asosida tendensiyalarning O‘zbekistondagi ko‘lamini empirik baholashga qaratilishi mumkin.

Adabiyotlar

1. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto for Agile Software Development. https://agilemanifesto.org

2. Brooks, F. P. (1987). No silver bullet: Essence and accidents of software engineering. Computer, 20(4), 10–19.

3. Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The science of lean software and DevOps. IT Revolution Press.

4. Newman, S. (2015). Building microservices: Designing fine-grained systems. O’Reilly Media.

5. Royce, W. W. (1970). Managing the development of large software systems. Proceedings of IEEE WESCON, 1–9.

6. Sommerville, I. (2016). Software engineering (10th ed.). Pearson.