Taraqqiyot spektri
ILMIY JURNALI
Vol 2 № 11 2026

SUN’IY INTELLEKTNING DASTURLASH SOHASIDAGI O’RNI VA KELAJAK ISTIQBOLLARI

Suyunov Sarvar Solijon o’g’li

OXUS University kompyuter ilimlari 501-26 guruh talabasi 100105, O’zbekiston,
xaydaralitursunboyev9@gmail.com

Tursunboyev Xaydarali To’lqin o’g’li

OXUS University kompyuter ilimlari 501-26 guruh talabasi 111500, O’zbekiston,
xaydaralitursunboyev1@gmail.com

Yusupov Samandar Anvarovich

OXUS Universiteti katta o‘qituvchisi Toshkent 100208, O‘zbekiston

Samandar75781111@gmail.com

Annotatsiya. Maqolada sun’iy intellekt (SI) vositalarining dasturiy ta’minot yaratish jarayoniga ta’siri tahlil qilinadi: kodni avtoto‘ldirishdan tortib, loyiha ustida mustaqil ishlaydigan “agent”larga qadar bo‘lgan rivojlanish bosqichlari, so‘nggi so‘rovnomalar va tadqiqotlar natijalari, SI ning afzallik va xavflari, dasturchi kasbining kelajagi hamda talabalar uchun amaliy tavsiyalar keltiriladi.

Kalit so‘zlar: sun’iy intellekt, dasturlash, katta til modellari, GitHub Copilot, AI agent, vibe coding, kelajak kasblari.

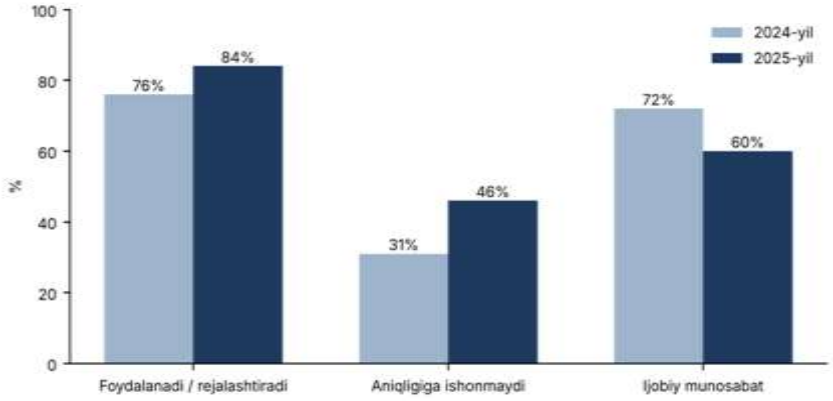
Kirish: Dasturlash o‘tgan yetmish yil davomida doimiy ravishda “yuqori abstraksiya” sari rivojlandi: mashina kodlaridan assemblerga, so‘ng yuqori darajali tillarga, ob’ektga yo‘naltirilgan yondashuvga, ochiq manbali kutubxonalar va bulutli xizmatlarga o‘tildi. Har bir bosqichda dasturchi mashinaga yaqin mayda tafsilotlardan xalos bo‘lib, masalaning mohiyatiga ko‘proq e’tibor qaratdi. Sun’iy intellekt bu zanjirdagi navbatdagi va ehtimol eng katta sakrashdir: endi kompyuterga odam tilida topshiriq berib, unga ishlaydigan kod yozdirish mumkin. Shu sababli bugungi kunda “SI dasturchilarning o‘rnini bosadimi?” degan savol har bir kompyuter ilmlari talabasini qiziqtirmoqda. Ushbu maqolada bu savolga faktlar va raqamlar asosida javob izlaymiz.

Yil	Voqea va uning ahamiyati
2021	OpenAI Codex modeli asosida GitHub Copilot paydo bo‘ldi: dasturchi kod yozayotganda keyingi qatorlar va butun funksiyalarni taklif qiladi.
2022	DeepMind’ning AlphaCode tizimi dasturlash musobaqalarida ishtirokchilar orasida o‘rtacha darajaga (taxminan yuqori 54%) erishdi. Shu yilning noyabrida ChatGPT chiqdi va suhbat orqali kod yozish ommalashdi.
2023	GPT-4 kabi kuchliroq modellar, IDE ichidagi chat-yordamchilar, kod bazasini “tushunadigan” vositalar tarqaldi.
2024–2025	Agent bosqichi: SI faqat maslahat bermaydi, balki fayllarni o‘qiydi, terminal buyruqlarini bajaradi, testlarni yurgizadi va xatolarni o‘zi tuzatadi (Cursor, Claude Code, Copilot agent rejimi va boshqalar). 2025-yil fevralida A. Karpathy “vibe coding” atamasini ommalashtirdi — kodni deyarli o‘qimay, faqat tavsif orqali yaratish uslubi.

SI dasturlash jarayonining qaysi bosqichlarida qo‘llanadi?

Dasturiy ta’minotni yaratish hayotiy sikli (SDLC) bir necha bosqichdan iborat bo‘lib, SI deyarli har birida yordam bera oladi:

1. Kod yozish va avtoto‘ldirish: takrorlanuvchi (boilerplate) kodni, API chaqiruvlarini, regulyar ifodalarni tez yozish.
2. Xatolarni topish (debugging): xato xabarini tahlil qilish, ehtimoliy sababni ko‘rsatish va tuzatish variantini taklif etish.
3. Test yaratish: unit-testlar va chegaraviy holatlarni avtomatik tuzish.
4. Kod ko‘rigi (code review) va xavfsizlik: zaif joylar, noto‘g‘ri uslub va samarasiz algoritmlarni aniqlash.
5. Refaktoring va migratsiya: eski kodni yangi til yoki kutubxonaga ko‘chirish.
6. Hujjatlashtirish: izohlar, README va API hujjatlarini yaratish.
7. O‘rganish: noma’lum kodni tushuntirish, tushunchalarni bosqichma-bosqich izohlash — “24/7 repetitor” sifatida.
8. Loyiha darajasidagi agentlar: bir nechta faylni o‘zgartirish, buyruqlarni ishga tushirish va natijani tekshirish orqali topshiriqni boshidan oxirigacha bajarish.
9. Raqamlar tilida: amaliyotda nima bo‘layapti?
10. Stack Overflow Developer Survey 2025 (177 ta mamlakatdan 49 000 dan ortiq texnologiya mutaxassisi) quyidagi natijalarni ko‘rsatdi:
11. Respondentlarning 84% i ish jarayonida SI vositalaridan foydalanadi yoki foydalanishni rejalashtiradi (2024-yilda — 76%). Professional dasturchilarning 51% i bu vositalardan har kuni foydalanadi.
12. SI natijasining to‘g‘riligiga 46% ishonmaydi (2024-yilda — 31%); ishonuvchilar 33%, “to‘liq ishonaman” deganlar esa atigi 3%.
13. SIga ijobiy munosabat 2024-yildagi 72% dan 60% ga tushdi.
14. Yangi SI-IDE’lar: Cursor — 18%, Claude Code — 10%, Windsurf — 5%. Eng ko‘p ishlatiladigan til modellari: OpenAI GPT — 81%, Claude Sonnet — 43%, Gemini Flash — 35%.



1-rasm. Dasturchilar o‘rtasida SI vositalariga munosabat dinamikasi, % (manba: Stack Overflow Developer Survey 2024–2025).

Ya’ni foydalanish ortmoqda, biroq ishonch kamaymoqda: dasturchilar SI ni ishlatadi, lekin natijani tekshirishda davom etadi.

Unumdorlikka ta’sir haqidagi tadqiqotlar ham turlicha natija beradi. GitHub 2022-yilda o‘tkazgan nazorat ostidagi tajribada (aniq, bir xil vazifa — HTTP-server yozish) Copilot’dan foydalanganlar vazifani taxminan 55% tezroq bajargan. Aksincha, METR tashkilotining 2025-yildagi tadqiqotida o‘z yirik ochiq manbali loyihalarida ishlayotgan tajribali dasturchilar SI vositalaridan foydalanganda vazifalarni o‘rtacha 19% sekinroq bajargan, garchi o‘zlari tezlashganini his qilishgan. Xulosa: SI ning foydasi vazifa turiga, dasturchi tajribasiga va loyiha murakkabligiga bog‘liq — u “sehrli tayoqcha” emas.

Afzalliklari va xavflari

Afzalliklari	Xavflari
Tezlik: oddiy va takrorlanuvchi ishlar sezilarli tezlashadi.	“Ishonchli noto‘g‘ri” javoblar (gallyutsinatsiya): mavjud bo‘lmagan funksiya yoki kutubxonani o‘ylab topishi mumkin.

Afzalliklari	Xavflari
Kirish to‘sig‘ining pasayishi: yangi boshlovchilar ham g‘oyani tez prototiplay oladi.	Xavfsizlik: yaratilgan kodda zaifliklar bo‘lishi, maxfiy ma’lumot (parol, API kalit)lar sizib chiqishi mumkin.
Shaxsiy o‘qituvchi: istalgan vaqtda tushuntirish va misollar.	Ko‘nikmalarning pasayishi: tayyor javobga haddan tashqari tayanish mustaqil fikrlashni susaytiradi.
Katta kod bazasida tez navigatsiya, test va hujjatlarni avtomatlashtirish.	Litsenziya va mualliflik huquqi, shuningdek javobgarlik masalalari hali to‘liq hal etilmagan.

Kelajak istiqbollari: Kelajak haqida mutaxassislar fikri bir xil emas, ammo ko‘pchilik quyidagi yo‘nalishlarni e’tirof etadi:

Rolning o‘zgarishi. Dasturchi “kod yozuvchi”dan “arxitektor, muharrir va tekshiruvchi”ga aylanib boradi: nima qurish kerakligini aniq ifodalash va natijani baholash qobiliyati qadrlanadi.

Agentlar va ko‘p-agentli tizimlar. Bir nechta SI agent tahlil, kod yozish, test va joylashtirishni hamkorlikda bajaradi; inson esa nazorat va strategik qarorlarni qabul qiladi.

Yangi mutaxassisliklar. AI/ML muhandisi, MLOps, kontekst va prompt muhandisligi, SI xavfsizligi, modellarni baholash (evals) kabi kasblar talabi oshadi.

Tabiiy til bilan dasturlash. Oddiy foydalanuvchilar ham kichik dasturlar yarata oladi, lekin murakkab, xavfsiz va yirik tizimlar uchun chuqur bilimli mutaxassislar zarur bo‘lib qoladi.

Kasb yo‘qolmaydi, o‘zgaradi. Kompilyatorlar paydo bo‘lganda ham dasturchilar yo‘qolmadi — ular murakkabroq masalalarni yecha boshladi. SI ham shunga o‘xshash ta’sir ko‘rsatishi ehtimoli katta.

O‘zbekiston konteksti: O‘zbekistonda raqamlashtirish va IT sohasini rivojlantirish davlat siyosatining ustuvor yo‘nalishlaridan biri: “Raqamli O‘zbekiston – 2030” strategiyasi, IT Park rezidentlari uchun beriladigan imtiyozlar, “Bir million o‘zbek dasturchisi” kabi o‘quv loyihalari hamda 2030-yilgacha mo‘ljallangan sun’iy intellekt texnologiyalarini rivojlantirish strategiyasi bunga misol bo‘ladi.

Yoshlar uchun bu — katta imkoniyat: SI vositalari til va resurs to‘siqlarini kamaytiradi, o‘zbek tili uchun ma’lumotlar to‘plamlari, tarjima va nutqni aniqlash tizimlari kabi mahalliy yechimlar yaratish esa hali to‘liq egallanmagan bo‘sh nisha bo‘lib turibdi.

Talabalar uchun amaliy tavsiyalar

1. Fundamentni mustahkam o‘rganing: algoritmlar, ma’lumotlar tuzilmasi, operatsion tizimlar, tarmoqlar va matematika — SI bergan javobni baholash uchun shart.
2. SI dan o‘qituvchi sifatida foydalaning: “kodni yozib ber” o‘rniga “bu kod qanday ishlaydi, nima uchun shunday yozilgan?” deb so‘rang.
3. Har bir kodni o‘qing, ishga tushiring va sinab ko‘ring: tushunmagan kodingizni loyihaga qo‘shmang.
4. Git, test yozish, debugging va xavfsizlik odatlarini erta shakllantiring.
5. Maxfiy ma’lumotni (parol, kalit, shaxsiy ma’lumot, yopiq kod) SI xizmatlariga yubormang.
6. Akademik halollikka rioya qiling: universitet va fan o‘qituvchisi qoidalariga muvofiq foydalaning.
7. Real loyihalar qiling: GitHub’da portfolio yarating, ochiq manbali loyihalarga hissa qo‘shing, ingliz tilini mustahkamlang.

Xulosa: Sun’iy intellekt dasturlashni o‘zgartirmoqda, ammo dasturchini almashtirmayapti: u takrorlanuvchi ishlarni tezlashtiradi, o‘rganishni osonlashtiradi, lekin masalani to‘g‘ri qo‘yish, tizimni loyihalash, natijani tekshirish va javobgarlikni zimmaga olish hamon insonning vazifasi bo‘lib qolmoqda. Kelajakda eng yutuqli bo‘lganlar — SI ni mohirona boshqara oladigan va ayni paytda fundamental bilimga ega mutaxassislar bo‘ladi.

Birinchi kurs talabasi uchun asosiy xulosa oddiy: SI bilan raqobatlashmang, uni o‘z kuchingizni oshiruvchi vosita sifatida o‘rganing — lekin “nima uchun ishlayapti?” degan savolni hech qachon qo‘ldan boy bermang.

Foydalanilgan adabiyotlar:

1. Stack Overflow. 2025 Developer Survey — AI. https://survey.stackoverflow.co/2025/ai
2. Stack Overflow. 2025 Developer Survey Reveals Trust in AI at an All Time Low (press release), 2025. https://stackoverflow.co/company/press/archive/stack-overflow-2025-developer-survey/
3. Chen M. va boshq. Evaluating Large Language Models Trained on Code (Codex), OpenAI, 2021. arXiv:2107.03374.
4. Li Y. va boshq. Competition-level code generation with AlphaCode. Science, 2022.
5. GitHub Research. Quantifying GitHub Copilot’s impact on developer productivity and happiness, 2022.
6. METR. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, 2025.