

MURAKKAB DASTURIY TIZIM HAYOTIY SIKLINI TA'MINLASH UCHUN INSTRUMENTAL VOSITALARNI TANLASH VA TAHLIL QILISH

Muallif: Tursunov Elyor

*Samarqand davlat universiteti Urgut filiali,
Biznesni boshqarish va tabiiy fanlar fakulteti, 3-bosqich talabasi*

✉ telyorbek09@gmail.com

Ilmiy rahbar: Suyarov Akram

Samarqand iqtisodiyot va servis instituti, dotsent

✉ akramsuyarov@mail.ru

Annotatsiya

Mazkur maqolada murakkab dasturiy tizimlar hayotiy siklini samarali boshqarish uchun zarur bo'lgan instrumental vositalarni tanlash mezonlari va ularni tahlil qilish jarayoni yoritilgan. Dasturiy injiniring doirasida tizimlarning loyihalash, ishlab chiqish, testlash, joylashtirish va qo'llab-quvvatlash bosqichlarida qo'llaniladigan vositalar turlari va ularning imkoniyatlari ko'rib chiqiladi. Maqolada tanlashda e'tibor berilishi lozim bo'lgan muhim jihatlar va zamonaviy yechimlar misollar asosida tahlil qilinadi.

Kalit so'zlar: dasturiy injiniring, hayotiy sikl, instrumental vositalar, tizim tahlili, dasturiy ta'minot sifati, SDLC.

1. Kirish

Murakkab dasturiy tizimlar bugungi kunda ko'plab sohalarda (moliya, sog'liqni saqlash, transport, ta'lim va boshqalar) keng qo'llanilmoqda. Bunday tizimlarni yaratish jarayoni – bu oddiy dasturiy mahsulotdan farqli o'laroq, murakkab va uzoq davom etuvchi bosqichlarni o'z ichiga oladi. Dasturiy ta'minotning hayotiy siklini (Software Development Life Cycle – SDLC) samarali boshqarish esa, maxsus instrumental vositalar orqali amalga oshiriladi.

Ushbu maqolada aynan ushbu vositalarning ahamiyati, turlari va tanlash mezonlari haqida so'z yuritiladi. Maqsad – murakkab tizimlarda yuqori sifatni ta'minlash uchun qanday vositalarni tanlash lozimligini aniqlashtirishdir.

2. Asosiy qism

2.1. Dasturiy ta'minot hayotiy sikli bosqichlari

Murakkab dasturiy tizimlar quyidagi asosiy bosqichlar orqali amalga oshiriladi:

- Talablarni yig'ish va tahlil qilish
- Loyihalash (design)
- Kod yozish (developing)
- Testlash
- Joylashtirish (deployment)

• Texnik xizmat ko'rsatish (maintenance)

Har bir bosqichda alohida instrumental vositalar qo'llaniladi va ular turli mezonlar asosida tanlanadi.

2.2. Instrumental vositalarning asosiy turlari va ularning funksiyalari

Bosqich	Vosita misoli	Asosiy funksiyasi
Talablarni aniqlash	IBM DOORS, Jira, Confluence	Talablarni yozish, tasniflash, kuzatish
Loyihalash	Enterprise Architect, Draw.io	UML diagrammalar, tizim arxitekturasini
Dasturlash	Visual Studio, IntelliJ IDEA	Kengaytirilgan muharrir, kompilyatsiya
Testlash	Selenium, JUnit, TestRail	Avtomatlashtirilgan testlar, natijalarni kuzatish
Joylashtirish	Jenkins, GitLab CI/CD, Docker	CI/CD jarayonlari, konteynerlashtirish
Qo'llab-quvvatlash	Nagios, Zabbix, New Relic	Monitoring, ishlash samaradorligini tahlil qilish

2.3. Vositalarni tanlash mezonlari

Dasturiy vositani tanlashda quyidagi mezonlar muhim hisoblanadi:

- **Moslik:** Vosita loyihada qo'llanilayotgan texnologiyalar bilan mos kelishi kerak.
- **Ishonchlilik:** Uzoq muddatli foydalanishda barqaror ishlashi.
- **Qo'llab-quvvatlash:** Hamjamiyat yoki rasmiy texnik yordam mavjudligi.
- **Foydalanish qulayligi:** Interfeys tushunarli va ergonomik bo'lishi.
- **Litsenziya va narx:** Ochiq kodli va tijoriy variantlar o'rtasida tanlov.

2.4. Zamonaviy yondashuvlar va tajribalar

Bugungi kunda DevOps, Agile va CI/CD kabi metodologiyalar instrumental vositalarning integratsiyalashgan ishlashini talab etadi. Masalan, Jenkins orqali kodni avtomatik sinovdan o'tkazib, Docker yordamida joylashtirish va Kubernetes orqali tizimni kengaytirish mumkin.

Murakkab tizimlarda mikroxizmat (microservices) arxitekturasini keng qo'llanilishi sababli, vositalarning modullik va o'zaro integratsiyalashish xususiyatlari ham muhim ahamiyatga ega.

3. Xulosa

Murakkab dasturiy tizimlarni muvaffaqiyatli yaratish va ularni samarali boshqarish — to'g'ri tanlangan instrumental vositalar bilan bevosita bog'liqdir. Maqolada ko'rib chiqilganidek, har bir bosqich uchun maxsus vositalar mavjud va ularning tanlovi loyihaaning texnik, iqtisodiy va amaliy talablariga asoslanadi.

Zamonaviy dasturiy injiniring yondashuvlarida vositalarning integratsiyasi va avtomatlashtirish imkoniyati – tizim samaradorligini oshirishga xizmat qiladi. Shu sababli, har bir loyiha oldidan ehtiyojlarni tahlil qilib, vosita tanloviga ongli yondashuv muhim ahamiyat kasb etadi.

4. Foydalanilgan adabiyotlar

1. Sommerville, I. (2016). Software Engineering, 10th Edition. Pearson Education.
2. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach. McGraw-Hill.
3. Jalilov, A. (2020). Dasturiy injiniring asoslari. Toshkent: TATU nashriyoti.
4. Hüttermann, M. (2012). DevOps for Developers. Apress.
5. Fowler, M. (2010). Patterns of Enterprise Application Architecture. Addison-Wesley.
6. GitLab, Jenkins, Docker, Kubernetes rasmiy hujjatlari (docs.gitlab.com, jenkins.io, docker.com, kubernetes.io).