

Date: 27<sup>th</sup> November-2025

## DASTURIY TA'MINOTNI MODELASHTIRISH

Ilmiy rahbar: G'anixojayeva Dilfuza

Zaxro Salimova Ulug'bek qizi

Alfraganus universiteti Raqamli texnologiyalar fakulteti kompyuter injenering 3-kurs talabasi

**Annotatsiya:** Ushbu maqolada dasturiy ta'minotni modelashtirish jarayoni, uning nazariy asoslari, turlari va zamonaviy dasturiy muhitlardagi qo'llanishi tahlil qilinadi. Model yaratishning maqsadi dasturiy tizimni loyihalash, tahlil qilish, sinovdan o'tkazish va ekspluatatsiya bosqichlarida aniqlik va ishonchlilikni oshirishdan iboratdir. Maqolada UML (Unified Modeling Language), BPMN (Business Process Model and Notation) kabi modelashtirish vositalarining samaradorligi ham yoritilgan.

**Kalit so'zlar:** dasturiy ta'minot, modelashtirish, UML, tizimli tahlil, dasturiy injiniring, algoritmik modellar.

**Аннотация:** В статье анализируется процесс моделирования программного обеспечения, его теоретические основы, типы и области применения в современных программных средах. Целью создания модели является повышение точности и надежности этапов проектирования, анализа, тестирования и эксплуатации программной системы. Также рассматривается эффективность инструментов моделирования, таких как UML (Unified Modeling Language), BPMN (Business Process Model and Notation).

**Ключевые слова:** программное обеспечение, моделирование, UML, системный анализ, программная инженерия, алгоритмические модели.

**Abstract:** This article analyzes the process of software modeling, its theoretical foundations, types and applications in modern software environments. The purpose of creating a model is to increase the accuracy and reliability of the design, analysis, testing and operation stages of a software system. The article also discusses the effectiveness of modeling tools such as UML (Unified Modeling Language), BPMN (Business Process Model and Notation).

**Keywords:** software, modeling, UML, systems analysis, software engineering, algorithmic models.

### Kirish

Bugungi kunda axborot texnologiyalari va dasturiy injiniring sohasi insoniyat faoliyatining barcha jabhalariga chuqur kirib bordi. Raqamli iqtisodiyotning shakllanishi, avtomatlashtirilgan boshqaruv tizimlari, sun'iy intellekt, katta ma'lumotlar (Big Data) va bulutli hisoblash (Cloud Computing) texnologiyalarining rivojlanishi natijasida dasturiy mahsulotlarga bo'lgan talab keskin ortib bormoqda. Shu sababli, dasturiy ta'minotni yaratish, tahlil qilish, sinovdan o'tkazish va ekspluatatsiya jarayonlarini samarali tashkil etish masalasi zamonaviy ilm-fanning eng dolzarb yo'nalishlaridan biri hisoblanadi.



**Date: 27<sup>th</sup> November-2025**



Dasturiy ta'minot tizimlari murakkablashgani sari ularni an'anaviy yondashuvlar orqali loyihalash va ishlab chiqish qiyinlashmoqda. Har qanday murakkab tizimni muvaffaqiyatli yaratish uchun uni avvalo modellashtirish zarur. Modellashtirish – bu real yoki mo'ljallangan tizimning soddalashtirilgan, lekin uning asosiy xususiyatlarini ifodalovchi nusxasini yaratish jarayonidir. Dasturiy ta'minotni modellashtirish orqali tizimning strukturasi, funksional imkoniyatlari, komponentlar o'rtasidagi bog'liqliklar hamda ularning xatti-harakatlari tahlil qilinadi. Bu esa loyihaning aniq konsepsiyasini shakllantirish, xatoliklarni erta aniqlash va resurslardan oqilona foydalanish imkonini beradi.

Zamonaviy dasturiy injiniringda modellashtirishning o'rni beqiyosdir. U nafaqat tizim arxitekturasini yaratishda, balki dasturiy mahsulotni hujjatlashtirish, tahlil qilish va takomillashtirish jarayonlarida ham muhim vosita hisoblanadi. Ayniqsa, UML (Unified Modeling Language) va BPMN (Business Process Model and Notation) kabi xalqaro standartlarga asoslangan modellashtirish vositalari dasturchilarga tizimni grafik ko'rinishda ifodalash, jamoaviy ish jarayonini muvofiqlashtirish hamda loyihaning umumiy sifatini oshirish imkonini beradi.

Dasturiy ta'minotni yaratish bo'yicha yondashuvlar – bu dasturiy mahsulotlarni ishlab chiqish jarayonida qo'llaniladigan usullar, metodlar va strategiyalardir. Turli xil loyihalarga va mijoz talablariga qarab, dasturiy ta'minotni ishlab chiqishda quyidagi asosiy yondashuvlar qo'llaniladi:

1. An'anaviy (Klassik) yondashuvlar: Bu yondashuvlar uzoq yillardan beri dasturiy ta'minot ishlab chiqishda qo'llanib kelinadi.

Kaskad modeli (Waterfall Model) bu dasturiy ta'minotni ishlab chiqish jarayonida bosqichma-bosqich harakatlanishni ta'minlovchi klassik yondashuv bo'lib, har bir bosqichni yakunlagandan keyin keyingi bosqichga o'tish imkonini beradi. Ishlab chiqish jarayoni qat'iy bosqichlarga bo'linadi: talablarni yig'ish → tizim dizayni → dasturlash → test → foydalanishga topshirish → texnik xizmat ko'rsatish. Bu yondashuv asosida ushbu hususiyatlari ochib beriladi, bosqichma-bosqich ketma-ketlikda amalga oshiriladi va har bir bosqich yakunlanmaguncha, keyingisiga o'tilmaydi. Bu model asosida tizimli yondashuv, rejalashtirish oson kabi afzalliklarga ega.

V-model (Verification and Validation Model) - bu kaskad modelining (Waterfall Model) takomillashtirilgan shakli bo'lib, har bir ishlab chiqish bosqichi uchun mos keluvchi sinov (test) bosqichi mavjudligini ta'minlaydi. Ushbu model verifikatsiya (tekshirish) va validatsiya (tasdiqlash) tamoyillari asosida ishlab chiqilgan. Bu modelda xatolar dastlabki bosqichlarda aniqlanadi va sifat nazorati kuchli kabi afzalliklarga ega.

Zamonaviy (Moslashuvchan) yondashuvlar: Zamonaviy yondashuvlar tezkorlik, moslashuvchanlik va foydalanuvchi talablariga tez javob berishga qaratilgan.

Agile (Tezkor yondashuv) – bu dasturiy ta'minot ishlab chiqish jarayonini moslashuvchan va iterativ (bosqichma-bosqich) shaklda tashkil etishga asoslangan yondashuvdir. Agile modelida loyihaning barcha bosqichlari kichik qismlarga bo'linib, doimiy ravishda takomillashtiriladi va mijoz fikr-mulohazalari asosida o'zgarishlar

**Date: 27<sup>th</sup> November-2025**

kiritiladi. Agile (tezkor yondashuv) foydalanuvchi bilan yaqin hamkorlik, o'zgarishlarga tez moslashish, qisqa muddatli iteratsiyalar (2-4 hafta) va ishchi dasturiy ta'minotni tez-tez taqdim etish kabi asosiy tamoyillarga ega. Scrum (Jamoa ma'lum muddatda (sprint) aniq vazifalarni bajaradi), Kanban (Ish jarayoni vizual taxtalarda kuzatiladi) kabi mashhur Agile metodologiyalari mavjud bo'lib moslashuvchan, tezkor javob berish, foydalanuvchi talablari tez amalga oshishi afzalliklariga ega.

Iterativ yondashuv (Iterative Model) – bu dasturiy ta'minotni ishlab chiqish yondashuvi bo'lib, unda dastur bir martalik to'liq ishlab chiqish o'rniga, takroriy bosqichlar (iteratsiyalar) orqali rivojlantiriladi va takomillashtiriladi. Har bir iteratsiya ishlaydigan dasturiy mahsulotning yaxshilangan versiyasini yaratishga xizmat qiladi. Xatolar va kamchiliklarni erta aniqlash va har bir bosqichda funksionallik ortib borash kabi afzalliklarga ega.

Spiral modeli (Spiral Model) – bu iterativ (takroriy) va xavf tahliliga asoslangan dasturiy ta'minot ishlab chiqish modeli bo'lib, u kaskad modeli va iterativ yondashuvning eng yaxshi jihatlarini birlashtiradi. Ushbu model katta hajmli va murakkab loyihalar uchun mo'ljallangan bo'lib, har bir bosqichda xavf tahlili amalga oshiriladi va loyiha iterativ ravishda rivojlantiriladi. Bu model risklarni boshqarish yaxshi yo'lga qo'yilganligi va katta va murakkab loyihalar uchun mos bo'la oladigan afzalliklarga ega.

Yangi texnologiyalarga asoslangan yondashuvlar

DevOps. (Development + Operations) – bu dasturiy ta'minot ishlab chiqish (Development) va IT operatsiyalar (Operations) o'rtasidagi hamkorlikni kuchaytirish uchun yaratilgan yondashuv. DevOps asosiy maqsadi – dasturiy mahsulotni ishlab chiqish, test qilish, yetkazib berish va joylashtirish jarayonlarini avtomatlashtirish va optimallashtirishdan iborat. Bu model dasturiy mahsulotni uzluksiz ishlab chiqish, test qilish va joriy etish, CI/CD (Continuous Integration/Continuous Deployment) texnologiyalaridan foydalanish kabi xususiyatlarga ega bo'lib dasturiy ta'minotni ishlab chiqish, joriy qilish jarayonlari tezlashtirish va xatolar tez tuzatilish kabi afzalliklarga ega.

Rapid Application Development (RAD) – bu tezkor dasturiy ta'minot ishlab chiqish modeli bo'lib, minimal rejalashtirish va maksimal prototiplash orqali dasturiy mahsulotni tezda ishlab chiqish tamoyiliga asoslanadi. Ushbu model an'anaviy yondashuvlardan farqli o'laroq, dasturiy ta'minotni tezroq ishlab chiqish, sinovdan o'tkazish va moslashtirish imkonini beradi. Bu modelning imkoniyatlari ishlab chiqish vaqti qisqa va foydalanuvchi fikri tez inobatga olinish afzalliklariga ega. Bu yondashuvlarni tahlil qilish orqali modellarni qiyosiy jadval asosida taqqoslashimiz mumkin.

| Yondashuv     | Afzalliklari                   | Kamchiliklari                 | Qo'llanilish sohalari       |
|---------------|--------------------------------|-------------------------------|-----------------------------|
| Kaskad modeli | Tuzilmaviy,<br>boshqarish oson | Moslashuvchan emas            | Kichik va barqaror talablar |
| Agile         | Moslashuvchan,<br>tezkor       | Katta loyihalarda<br>murakkab | Dinamik o'zg                |

**Date: 27<sup>th</sup> November-2025**

|          |                                  |                     | aruvchi talablar                  |
|----------|----------------------------------|---------------------|-----------------------------------|
| Iterativ | Bosqichma-bosqich takomillashuv  | Resurs talab qiladi | O'rtacha va katta loyihalar       |
| Spiral   | Risklarni boshqarish yaxshi      | Murakkab va qimmat  | Katta va murakkab loyihalar       |
| DevOps   | Uzluksiz integratsiya, tezkorlik | Texnik murakkablik  | Doimiy yangilanadigan mahsulotlar |

Bu fikrlar asosida quyidagi hulosalarni qilishimiz mumkin. Bo'lajak o'qituvchilarni raqamli kompetentligini rivojlantirishga mo'ljallangan dasturiy ta'minotni yaratishda ko'pincha yuqoridagi yondashuvlari samarali hisoblanadi. Chunki:

- O'qituvchilarning ehtiyojlari va texnologiyalardagi tez o'zgarishlar doimiy takomillashtirishni talab qiladi
- Foydalanuvchilarning fikr-mulohazalari tez hisobga olinadi.
- Moslashuvchanlik va tezkorlik ta'minlanadi.

Bu yondashuvlar orqali ishlab chiqilgan dasturiy mahsulotlar bo'lajak o'qituvchilarni AKT savodxonligi, raqamli resurslar yaratish, masofaviy ta'lim platformalaridan samarali foydalanish kabi ko'nikmalarini rivojlantirishga katta hissa qo'shadi.

#### **Xulosa**

Dasturiy ta'minotni modellashtirish zamonaviy dasturiy injiniringning ajralmas va eng muhim bosqichlaridan biri hisoblanadi. U dasturiy tizimlarning arxitekturasini chuqur tahlil qilish, ularning funksional va mantiqiy tuzilmasini aniqlashtirish hamda ishlab chiqish jarayonini tizimli boshqarish imkonini beradi. Modellashtirishning asosiy maqsadi – dasturiy mahsulotning sifatini oshirish, ishlab chiqish vaqtini qisqartirish va resurslardan oqilona foydalanishni ta'minlashdir.

Tadqiqot natijalari shuni ko'rsatadiki, dasturiy modellashtirish dasturiy tizimlarni yaratishda ko'plab afzalliklarni beradi. Birinchidan, modellar yordamida tizimni turli burchaklardan ko'rib chiqish, murakkab jarayonlarni soddalashtirilgan ko'rinishda tahlil qilish mumkin bo'ladi. Ikkinchidan, modellashtirish loyiha dastlabki bosqichlarida xatoliklarni aniqlash imkonini berib, keyingi texnik va iqtisodiy yo'qotishlarning oldini oladi. Uchinchidan, modellar dasturiy loyihalash jarayonida jamoaviy ishni muvofiqlashtiradi, chunki ular kommunikatsiya vositasi sifatida dasturchilar, tahlilchilar va menejerlar o'rtasida yagona tushuncha yaratadi.

#### **FOYDALANILGAN ADABIYOTLAR:**

1. Abdullayeva B.S. Формирование информационной компетентности школьника. Formation of information competence of students // Eastern European Scientific Journal.

**Date: 27<sup>th</sup> November-2025**

AURIS Kommunikations - und Verlagsgesellschaft mbH Dusseldorf. Germany: Ausgabe 3. 2018. – P. 183-187.

2. Abdullayeva B.S., Yaxyaev S.J.. Elektron ta'lim muhitida ta'lim jarayonini tashkil qilishning normativ-huquqiy asoslari // Respublika ilmiy-uslubiy konferensiya. – Toshkent, 2022, – B.301-302.

3. Прессман Р. С. Инженерия программного обеспечения: Практический подход. – Москва: Вильямс, 2019. – 960 с.

4. Коммервилл И. Инженерия программного обеспечения. 10-е изд. – Санкт-Петербург: Питер, 2020. – 832 с.

5. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. – 2nd ed. – Addison-Wesley, 2017. – 528 p.

6. Гуревич И. В. Моделирование программных систем. – Москва: Инфра-М, 2021. – 312 с.

7. ISO/IEC/IEEE 42010:2011. Systems and software engineering — Architecture description. – Geneva: ISO, 2011. – 45 p.

