



## DASTURIY TA'MINOTGA BO'LGAN TALABLARNI TEKSHIRISH VA HUJJATLASHTIRISH

Muallif: **Xudayqulov Azizbek**

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

 [azizbekxudayqulov456@gmail.com](mailto:azizbekxudayqulov456@gmail.com)

Ilmiy rahbar: **Suyarov Akram**

*Samarqand iqtisodiyot va servis instituti, dotsent*

 [akramsuyarov@mail.ru](mailto:akramsuyarov@mail.ru)

**Annotatsiya:** Mazkur maqolada dasturiy ta'minotga bo'lgan talablarni to'plash, tekshirish va hujjatlashtirish bosqichlari haqida batafsil tushuncha beriladi. Dasturiy tizim foydalanuvchining ehtiyojlarini to'g'ri tushunib, to'g'ri hujjatlashtirmasa, loyiha muvaffaqiyatsizlikka uchrashi mumkin. Shuning uchun talablar bilan ishlash dasturiy ta'minot muhandisligining eng muhim bosqichidir.

**Kalit so'zlar:** talablar, foydalanuvchi ehtiyoji, hujjatlashtirish, funksional talablar, notalablar, verifikatsiya, validatsiya.

### 1. Kirish

Dasturiy ta'minot ishlab chiqishda eng muhim qadam bu – **foydalanuvchining ehtiyojini aniqlash**dir. Agar mijoz yoki foydalanuvchi tizimdan nimani kutayotganini aniq tushunmasak, eng zo'r texnologiyalar bilan ishlab chiqilgan dastur ham foydasiz bo'lib qoladi. Shu sababli dastlabki bosqichda talablarni to'plash, tekshirish va hujjatlashtirish juda muhim.

### 2. Talablar tushunchasi



**Talab** – bu foydalanuvchining dasturdan kutayotgan imkoniyatlari, xatti-harakatlari yoki tizimga qo‘yiladigan texnik yoki funksional shartlar majmuasidir.

Talablar ikki asosiy turga bo‘linadi:

### 2.1. Funksional talablar

- Dastur nima qilishi kerakligi ko‘rsatiladi
- Masalan: foydalanuvchi ro‘yxatdan o‘tishi, buyurtma berishi, ma’lumotlar qidirishi

### 2.2. Nofunksional talablar

- Dastur qanday ishlashi kerakligini ifodalaydi
- Masalan: xavfsizlik, ishlash tezligi, foydalanuvchilarga qulaylik, tizim yuklamasiga bardoshlilik

## 3. Talablarni tekshirish (verifikatsiya va validatsiya)

Talablar to‘plangach, ular albatta **tekshiruvdan o‘tkazilishi** kerak. Bu jarayon ikki bosqichga bo‘linadi:

### 3.1. Verifikatsiya (Verification)

- Bu talablar to‘g‘ri yozilganmi?
- Mantiqiy xatolar yo‘qmi?
- Har bir talab tushunarli va aniqmi?

### 3.2. Validatsiya (Validation)

- Bu talablar foydalanuvchining ehtiyojini to‘liq qamrab olganmi?
- Mijoz haqiqatda bunday tizimni xohlaptimi?
- Loyiha bo‘yicha noto‘g‘ri tushunchalar yo‘qmi?



**Misol:** Agar foydalanuvchi “tezkor buyurtma tizimi” so‘rasa, bu “tezkor” degani nima? 1 soniya ichida ishlashi kerakmi yoki 1 daqiqa? Shuni aniqlash validatsiyaga kiradi.

#### 4. Talablarni hujjatlashtirish

Talablar to‘plangach va tekshirilgach, ular **rasmiy hujjatga** aylantiriladi. Bu jarayon quyidagi shakllarda amalga oshiriladi:

##### 4.1. Talablar spetsifikatsiyasi (SRS – Software Requirements Specification)

- Bu hujjatda barcha funksional va nofunksional talablar yoziladi
- Har bir talab raqamlanadi, aniq shaklda ifodalanadi
- Grafiklar, diagrammalar (UML) yordamida tushunariligi oshiriladi

##### 4.2. Use Case diagrammalar

- Tizim foydalanuvchi bilan qanday ishlashini ko‘rsatadi
- Harakatlar ketma-ketligi, foydalanuvchi rollari tasvirlanadi

##### 4.3. Talablar jadvali

| Raqam | Talab tavsifi                                  | Talab turi   | Ustuvorlik | Izoh                       |
|-------|------------------------------------------------|--------------|------------|----------------------------|
| 1     | Foydalanuvchi ro‘yxatdan o‘tishi kerak         | Funksional   | Yuqori     | Telefon raqam orqali       |
| 2     | Tizim 2 soniyadan ortiq javob bermasligi kerak | Nofunksional | O‘rta      | Yuklama testida aniqlanadi |



## Xulosa

Dasturiy ta'minotni yaratishda talablar ustida ishlash eng muhim bosqichlardan biridir. To'g'ri to'plangan, tekshirilgan va hujjatlashtirilgan talablar:

- Loyiha muvaffaqiyatini oshiradi
- Jamoa a'zolari uchun aniq yo'l xaritasi bo'ladi
- Noto'g'ri tushunchalarning oldini oladi
- Tizim sinovini va rivojlanishini osonlashtiradi

Shuning uchun loyiha boshida talablarni puxta tekshirish va ularni aniq shaklda yozib olish har doim katta foyda keltiradi.

## Foydalanilgan adabiyotlar

1. Sommerville, I. (2016). *Software Engineering*
2. Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach*
3. Karl E. Wiegers, Joy Beatty (2013). *Software Requirements*
4. [www.visual-paradigm.com](http://www.visual-paradigm.com) – UML for Requirements
5. [www.atlassian.com](http://www.atlassian.com) – How to write requirements effectively
6. Karimov M. (2022). *Dasturiy tizimlar talablari*, TATU