



DASTURIY TA'MINOT ISHLAB CHIQUISHDA SHARSHARA MODELI

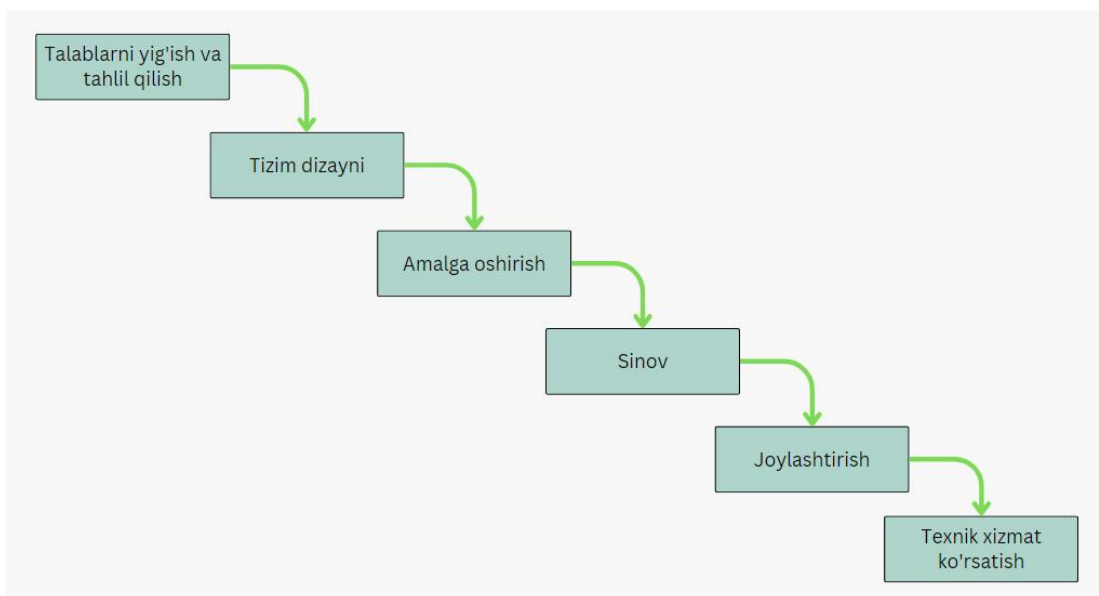
Narziyev Nosir Baxshilloyevich (TATU, DIF, katta o'qituvchi)

Qulmatov Qurvonali Zokirali o'g'li (TATU, DIF, talaba)

Annotatsiya: Ushbu maqolada sharshara modelining chuqur tahlili, uning tamoyillari, xususiyatlari, afzalliklari va cheklovlari haqida ma'lumotlar keltirilgan. Maqolada palapartishlik modelining ketma-ket tabiatini tasvirlash bilan boshlanadi, bunda har bir bosqich talablarni to'plashdan tortib to joylashtirishgacha ketma-ket yoritib boriladi.

Kalit so'zlar: Sharshara model, Unit testing, Integration testing, System testing, Acceptance testing, System Design, Implementation, Maintenance, Agile.

Sharshara modeli - chiziqli va bosqichma-bosqich yondashuviga amal qiladigan dasturiy ta'minotni ishlab chiqish jarayoni modeli. Bu model dasturiy ta'minotni ishlab chiqishda ishlatiladigan eng qadimgi va an'anaviy modellardan biridir. Model o'z nomini palapartishlikdagi suvga o'xshab, keyingi bosqichga o'tishdan oldin bajariladigan kaskadli harakatlar oqimidan olgan.



Sharshara modeli quyidagi ketma-ket bosqichlardan iborat:



1. Talablarni yig'ish va tahlil qilish (Requirements Gathering and Analysis): Ushbu bosqichda dasturiy ta'minotni loyihalashtiruvchi jamoa dasturiy ta'minot tizimining barcha talablarini aniqlaydi va hujjatlashtiradi. Dasturiy ta'minot tizimining talablari manfaatdor tomonlar, foydalanuvchilar va boshqa tegishli manbalardan yig'iladi va ular aniqlik, to'liqlik va amalga oshirish imkoniyatini ta'minlash uchun tahlil qilinadi.
2. Tizim dizayni (System Design): Tahlil qilingan talablar asosida tizimni loyihalash bosqichi boshlanadi. Dizayn jamoasi tizimning arxitekturasini, modullarini, ma'lumotlar tuzilmalarini, foydalanuvchi interfeysini va boshqa komponentlarini tavsiflovchi keng qamrovli dizayn rejasini yaratadi. Dizayn odatda diagrammalar va hujjatlar orqali ifodalanadi.
3. Amalga oshirish (Implementation): Ushbu bosqichda ishlab chiqish guruhi dizayn spetsifikatsiyalari asosida kodlashni boshlaydi. Amalga oshirish dizayn rejasiga muvofiq amalga oshiriladi va alohida modullar yoki komponentlar uchun kod yozishni o'z ichiga oladi. Kodlash loyiha uchun tanlangan maxsus dasturlash tilida amalga oshiriladi.
4. Sinov (Testing): Amalga oshirish tugallangach, sinov bosqichi boshlanadi. Dasturiy ta'minot nuqsonlar, xatolar va belgilangan talablarga muvofiqligi uchun sinovdan o'tkaziladi. Dasturiy ta'minot sifati va funkcionalligini ta'minlash uchun birlik testi (unit testing) , integratsiya testi (integration testing), tizim testi (system testing) va qabul qilish testi (acceptance testing) kabi testlarning turli turlari amalga oshiriladi.
5. Joylashtirish (Deployment): Muvaffaqiyatli sinovdan so'ng, dasturiy ta'minot ishlab chiqarish muhitiga joylashtiriladi yoki oxirgi foydalanuvchilarga chiqariladi. Ushbu bosqich o'rnatish, sozlash va kerakli ma'lumotlarni ko'chirishni o'z ichiga oladi. Sharshara modelida joylashtirish odatda barcha ishlab chiqish va sinov bosqichlari tugagandan so'ng amalga oshiriladi.
6. Texnik xizmat ko'rsatish (Maintenance): texnik xizmat ko'rsatish bosqichi dasturiy ta'minotni doimiy qo'llab-quvvatlash, xatolarni tuzatish va yangilanishlarni taqdim etishni o'z ichiga oladi. Foydalanuvchilarning fikr-mulohazalari, joylashtirishdan keyin aniqlangan muammolar yoki talablarning o'zgarishi texnik xizmat ko'rsatish faoliyatini boshlashi



mumkin. Dasturiy ta'minotga o'zgartirishlar va takomillashtirishlar alohida xizmat ko'rsatish davrlari orqali boshqariladi.

Sharshara modeli barcha talablarni oldindan to'plash va aniqlash mumkinligini va ishlab chiqish jarayonida minimal o'zgarishlar bo'lishini nazarda tutadi. Har bir faza avvalgisining tugallanishiga bog'liq bo'lgan ketma-ket oqimga amal qiladi. Ushbu model ko'pincha o'zgarishlar minimal bo'lishi kutilayotgan aniq va barqaror talablarga ega bo'lgan loyihalar uchun foydali hisoblanadi. Sharshara modelida ba'zi cheklovlar mavjud. Uning chiziqli tabiati uni o'zgarishlarga moslashish yoki rivojlanayotgan talablarni qondirish uchun kamroq moslashuvchan qiladi. Bundan tashqari, agar sinov yoki o'rnatish bosqichida biron bir muammo yoki xatolik aniqlansa, bu ko'p vaqt talab qiladigan va qimmatga tushishi mumkin bo'lgan oldingi bosqichlarga qaytishni talab qilishi mumkin. Ushbu cheklovlar tufayli ko'plab dasturiy ta'minotni ishlab chiqish guruhlariga Agile kabi ko'proq iterativ va moslashuvchan metodologiyalarga o'tishdi, ular ishlab chiqish jarayonida ko'proq moslashuvchanlik va mijozlar bilan hamkorlikni taklif qiladi.

ADABIYOTLAR RO'YXATI.

1. "Software Requirements: Practical Techniques for Gathering and Managing Requirements Throughout the Product Development Cycle" by Karl Wieggers
2. "Software Engineering: A Practitioner's Approach" by Roger S. Pressman
3. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development" by Craig Larman
4. "Software Engineering: Theory and Practice" by Shari Lawrence Pfleeger and Joanne M. Atlee
5. "Managing Software Projects: The Open University Guide to Project Management" by Steve McConnell