



DASTURIY TA'MINOTNI ISHLAB CHIQISHNING ZAMONAVIY TAMOYILLARI

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

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

Annotatsiya: Ushbu maqolada dasturiy ta'minot ishlab chiqishda qo'llaniladigan zamonaviy tamoyillar haqida ma'lumotlar keltirilgan.

Kalit so'zlar: Architecture first approach, Iterative life cycle process, Component Based Approach, Change Management system, Round Trip Engineering, Objective Quality Control.

Dasturiy ta'minotni ishlab chiqish uchun ba'zi zamonaviy tamoyillar mavjud. Ushbu zamonaviy tamoyillarga rioya qilish orqali biz mijozlarning barcha ehtiyojlariga javob beradigan samarali dasturiy ta'minotni ishlab chiqishimiz mumkin. To'g'ri dasturiy ta'minotni ishlab chiqish uchun dasturiy ta'minotni ishlab chiqishning quyidagi 10 tamoyiliga amal qilish kerak:

Dasturiy ta'minotni ishlab chiqish tamoyillari:

1. Architecture first approach - Arxitekturaning birinchi yondashuvi
2. Iterative life cycle process - Iterativ hayot tsikli jarayoni
3. Component Based Approach - Komponentga asoslangan yondashuv
4. Change Management system - O'zgarishlarni boshqarish tizimi
5. Round Trip Engineering - Aylanma muhandislik
6. Model Based Evolution - Modelga asoslangan evolyutsiya
7. Objective Quality Control - Maqsadli sifatni nazorat qilish
8. Evolving levels of details - Rivojlanayotgan tafsilotlar darajalari
9. Establish a configurable process - Sozlanadigan jarayonni tashkil qilish
10. Demonstration Based Approach - Namoyishga asoslangan yondashuv

Arxitekturaning birinchi yondashuvi: Ushbu yondashuvda asosiy maqsad bizning dasturiy ta'minotimiz uchun kuchli arxitekturaning yaratishidir. Barcha noaniqliklar va kamchiliklar juda arziyas bosqichda aniqlanmoqda. Shuningdek, biz dasturiy ta'minotning



samaradorligini oshiradigan dasturiy ta'minot dizayni bo'yicha barcha qarorlarni qabul qilishimiz mumkin.

Iterativ hayot tsikli jarayoni: Iterativ hayot tsikli jarayoni, biz xavf omillarini bartaraf etish uchun jarayonni qayta-qayta takrorladik. Iterativ hayot tsikli bizda asosan to'rt bosqichli talablarni yig'ish, loyihalash, amalga oshirish va sinovdan o'tkazish. Ushbu qadamlarning barchasi biz xavf omilini yumshatmagunimizcha qayta-qayta takrorlanadi. Iterativ hayot aylanish jarayoni yuqorida aytib o'tilgan qadamlarni qayta-qayta takrorlash orqali erta bosqichda xavfni engillashtirish uchun muhimdir.

Komponentga asoslangan yondashuv: Komponentga asoslangan yondashuvda keng qo'llaniladigan va muvaffaqiyatli yondashuv bo'lib, unda biz dasturiy ta'minotni ishlab chiqish uchun avval belgilangan funktsiyalardan qayta foydalanamiz. Kod qismini komponentlar shaklida qayta ishlatamiz. Komponentga asoslangan UI ishlab chiqish talablar va dizayn jarayonini optimallashtiradi va shuning uchun zamonaviy dasturiy ta'minotning muhim tamoyillaridan biridir.

O'zgarishlarni boshqarish tizimi: O'zgarishlarni boshqarish - bu barcha o'zgarishlarni boshqarish uchun mas'ul bo'lgan jarayon. O'zgarishlarni boshqarishning asosiy maqsadi kerakli o'zgarishlarni amalga oshirish orqali dasturiy ta'minot sifatini yaxshilashdir. Amalga oshirilgan barcha o'zgarishlar sinovdan o'tkaziladi va sertifikatlanadi.

Aylanma muhandislik: Aylanma muhandislik kodini ishlab chiqarish va teskari muhandislik bir vaqtning o'zida dinamik muhitda amalga oshiriladi. Ikkala komponent ham ishlab chiquvchilar ikkalasida ham oson ishlashi uchun birlashtirilgan. Qaytish muhandisligida asosiy xususiyat artefaktlarni avtomatik yangilashdir.

Modelga asoslangan evolyutsiya: Modelga asoslangan evolyutsiya dasturiy ta'minotni ishlab chiqishning muhim tamoyilidir. Modelga asoslangan yondashuv grafik va matn tushunchalarining evolyutsiyasini qo'llab-quvvatlaydi.

Maqsadli sifatni nazorat qilish: Sifat nazoratining maqsadi dasturiy ta'minot sifatini yaxshilashdir. U Sifat menejmenti rejasini, sifat ko'rsatkichlarini, sifat nazorat ro'yxatini, sifatning asosiy darajasini va sifatni yaxshilash chora-tadbirlarini o'z ichiga oladi.



Rivojlanayotgan tafsilotlar darajalari: Oraliq nashrlarni foydalanish stsenariylari guruhlarida o'zgaruvchan tafsilotlar darajasi bilan rejalashtiring. Biz bosqichma-bosqich amalga oshirishni rejalashtirishimiz kerak, bunda bizda rivojlanish darajasi, arxitektura va tafsilotlar mavjud.

Sozlanadigan jarayonni tashkil qilish: Iqtisodiy jihatdan kengaytirilishi mumkin bo'lgan sozlanishi mumkin bo'lgan jarayonni o'rnatish. Bitta jarayon barcha ishlab chiqishlar uchun mos emas, shuning uchun biz turli xil ilovalar bilan ishlay oladigan sozlanishi mumkin bo'lgan jarayondan foydalanishimiz kerak.

Namoyishga asoslangan yondashuv: Ushbu yondashuvda biz asosan namoyishga e'tibor qaratamiz. Bu muammo sohasi, qo'llaniladigan yondashuvlar va echimlar haqida aniq tavsifni taqdim etish orqali dasturiy ta'minotimizning mahsuldorligi va sifatini oshirishga yordam beradi.

ADABIYOTLAR RO'YXATI.

1. "Software Engineering: Modern Approaches" by Eric J. Braude, Michael E. Bernstein.
2. "Agile Software Development: Principles, Patterns, and Practices" by Robert C. Martin, Micah Martin.
3. "The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations" by Gene Kim, Jez Humble, Patrick Debois, John Willis.
4. "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin.
5. "Software Engineering: A Practitioner's Approach" by Roger S. Pressman, Bruce R. Maxim.