

versenyszférához képest lomhábban állnak át a már megszokott technológiákról egy újra a megszokás, valamint az alaposabb kiteszteltség iránti igény miatt.

A fejlesztés során alkalmazott módszertant nem lehet tisztán egyik vagy másik csoportba sorolni, mivel más és más az elvárás a háttérlogikával és a felhasználói felülettel (GUI) szemben. Az előbbihez jobban illik a vízésésmodell alapú, míg utóbbihoz az evolúciós fejlesztési módszertan. A különbség abból adódik, hogy a szoftver meghatározott funkciókat kell, hogy ellásson, ami a követelmények elemzése közben élesen körvonalazódik, miközben a program kinézetéről nem lehet minden esetben előre döntés hozni. A grafikus felületet úgy lehet csak tökéletesre csiszolni, ha egy kezdeti implementációt bocsátunk a leendő felhasználók számára, amit több verzió keresztül tökéletesítünk addig a pontig, míg el nem érjük az optimális működést. A projekt tervezésénél ezt a kettősséget figyelembe kellett venni és a GUI fejlesztésénél körültekintően, kell meghatározni, az erre fordítandó fejlesztési időt megfelelő mennyiségű tartalékot hagyva.

Az alkalmazás teljes megtervezése, túlmutat a megvalósíthatósági tanulmány keretein, viszont a követelmények definiálásánál és a megvalósítható lehetőségeknél a teljességre kell törekedni. A tervezéskor a teljes problémára kell megoldást kínálni, amit a projekt keretében a teljes alkalmazás rendszer fázisokra és modulokra bontott fejlesztésével kell megvalósítani. Az alábbiak a fejlesztés során felmerülő lépéseket foglalják össze.

A fejlesztés következő állomása a Követelmények elemzése és meghatározása. Ebben a kezdő fázisában az igény alap gondolatának kifejtése történik. A szakterületi réteg képviselőivel a projektvezetővel és a szoftvertervezővel közösen meghatározásra kerülnek a rendszerrel szemben támasztott elvárások, amit közösen priorizálva alakul ki a magas szinten meghatározott igény. A fázis alapvető célja hogy a legteljesebb képet alkothassunk megvalósítandó folyamatokról, ami végső soron ahhoz is szükséges, hogy pontosan meg lehessen becsülni az alkalmazás létrehozására fordítandó erőforrások mértékét.

Az implementáció és egységteszt során a részletes tervek alapján ekkor történik az implementáció. Valós körülmények között az esetleges előre nem látott nehézségek (új technológia használata, projekttől független külső hatások) megoldásához szükség lehet az szervezeti döntéshozók, kapcsolattartók bevonására is. Az elkészült programegységek halmaza itt még csak külön-külön kerül tesztelésre. A fázis végén előálló forráskód készen áll az integrációra és a rendszertesztre.

A vízésésmodellől való eltérés itt mutatkozik meg először. A felhasználói felület kialakítása a tervezési szakaszban előállított drótvázakból indul ki, ami több verzió keresztül számos a szakterületi réteg képviselőivel történő iteráció után éri el a végleges formáját. Az evolúciós fejlesztés előnye is ebben rejlik, a testre szabottság így jóval nagyobb mértékű mintha a drótváz egy az egyben került volna leprogramozásra. A felhasználók megtapasztalhatják, hogy milyen lesz az új rendszer (vagy a meglévő rendszeren történt fejlesztés) és az igényeikre szabható.

Az Integráció és rendszerteszt során az elkészült alkalmazás részek itt kerülnek egy egységes egészé alakításra és helyeződnek a majdani alkalmazáskörnyezetbe. A fejlesztés által végzett integrációs tesztek után a rendszertesztet már a felhasználási folyamatok mentén célszerűen a megrendelő végzi. Az alkalmazás akkor helyezhető éles üzembe, ha az előre definiált maximális hibaszámnál kevesebb nyitott hiba van benne.

A tervezett architektúra behatárolja a lehetőségeket, így a korszerű agilis tervezési minták alkalmazásával platform független alkalmazás hozható létre open source eszközökkel:

relációs adatbázis:	mysql.
Back-end fejlesztés:	C#,
ORM:	Hibernate, Dependency injection, SOA/WCF Unit Test.
Front-end fejlesztés:	MVC4, Razor, JQuery, Bootstrap, JS, CSS, HTML5

technológiák kerülnek alkalmazásra.