Szoftvertechnológia alapjai

A szoftver életciklusa

Minden terméknek van élettörténete, úgynevezett életciklusa, amely a rá vonatkozó igény felmerülésétől egészen a termék használatból való kivonásáig (feledésbe merüléséig) tart. Amikor a ciklus elnevezést használjuk, akkor ez főként azt szimbolizálja, amikor egy terméknek rendre újabb, továbbfejlesztett változatait állítják elő. Ilyenkor minden továbbfejlesztési lépést úgy tekinthetünk, mint az élettörténet megismétlődését. Valamennyi életciklustípuson belül megtalálható az a tervező, kísérletező tevékenység, amelyik jellegzetesen az új vagy módosított termék előállítására vonatkozó igény megszületésétől a gyártás megindításáig tart.

Fejlesztésen egy termék életciklusának azt a szakaszát értjük, amelyik a termék előállítására vagy módosítására vonatkozó igény felmerülésétől a gyártás megindításáig tart.

A szoftver maga szintén egy informatika-ipari termék, ezért szintén értelmezhető rá az életciklus fogalma. Míg az informatika kialakulásakor a programozók egyedül dolgoztak minden különösebb tudatosság nélkül, ma a nagyobb cégek kialakulásakor, illetve a termék további fejlesztései során kialakult bonyolítás miatt már a team-munka hatásaként ki kellett alakítani közös gondolkodásmódot és stílust, egy olyan fejlesztési folyamatot, amely fázisai jól definiálhatók és leírhatók. A fejlesztés komplexitása csakis így tud áttekinthetővé válni.

A szoftver élettörténetében az egyes szakaszok súlya jelentősen eltér a hagyományos termékeknél megszokottól. Miután a gyártás – azaz a példányok előállítása – nem okoz különösebb gondot, ezzel szemben a termék maga bonyolult, így az életciklus döntő része a fejlesztés, ezen belül is a termékfejlesztés. Ennek megfelelően a szoftvertechnológia nem a gyártás mikéntjére, hanem a termékfejlesztés mikéntjére koncentrál, erre próbál szisztematikus módszereket adni.

Életciklus jellemzői

A szoftver élettörténetét leíró kezdeti modellek kezdetben szinte kizárólag az első működő példány elkészültéig terjedő szakaszig gondolták át a folyamatot. Az évek során ez jelentősen megváltozott. Hamar kiderült, hogy a szoftverek hajlamosak a folyamatos változásokra, amelyek kezeléséhez olyan életciklus-modellek szükségesek, amelyek ezt a tulajdonságát is tükrözik. Ennek ellenére az életciklus termékfejlesztési szakaszának elsődlegessége (dominanciája) ma is fennáll.

Az, hogy a mai szoftverek gyorsan változnak két tényező okozza. Egyrészt a változtatások technikailag könnyen kivitelezhetők. Ezért a felhasználók elvárják, hogy egyre újabb igényeiket gyorsan kövessék a szoftver újabb változatai. Másrészt a szoftvertermék általában sokkal bonyolultabb, mint a hagyományos anyagi jellegű termékek. Ezért nehéz a “nulláról induló” fejlesztés folyamán már az első példányt úgy elkészíteni, hogy az minden igényt kielégítsen. A szoftver változási hajlama ezért nemcsak azt eredményezi, hogy a nagy példányszámú szoftverek verziói sűrűbben követik egymást, mint ahogyan az anyagi jellegű termékeknél megszoktuk, hanem azt is, hogy életük során az egyedi szoftverek is számos változtatáson, módosításon, bővítésen esnek át.

Megtévesztő lehet a kivitelezés viszonylagos egyszerűsége, amely gyakran csábít a gyakori változtatásra. Azonban óvatosságra intenek a bonyolultságból fakadó nehézségek. A tapasztalatok azt mutatják, hogy egy bonyolult rendszerben minden módosítás rendkívül veszélyes. Ugyanis gyakran

nemcsak a kívánt elsődleges hatást érjük el, hanem olyan kellemetlen mellékhatások is jelentkezhetnek, amelyekre nem gondolunk a módosítás megtervezésekor. A szoftverszállítók gyakorta nem adják át a megrendelőnek a szoftver forrásnyelvű reprezentációját. Ezen ténynek csak egyik oka a szerzői jog védelme.

A másik ok az, hogy a szállítók el akarják kerülni a szoftver átgondolatlan megváltoztatását. Ugyanakkor a fejlesztők elemi érdeke, hogy a felmerülő módosítási igényeket minél gyorsabban és minél biztonságosabban ki tudják elégíteni, egy-egy új igény kielégítésére pedig minél gyorsabban tudjanak – lehetőleg már meglévő komponensekből – új rendszert készíteni.

Összefoglalva tehát ezért lett a szoftvertechnológia két kulcsszava a **módosíthatóság** és az **újrafelhasználhatóság**.

A szoftverfolyamat modelljei

A szofverfejlesztések számának és legfőképpen az egy-egy fejlesztéshez szükséges erőforrások volumenének növekedésével természetes módon jelent meg az igény a fejlesztési folyamat racionalizálására, ami többek között az ütemezési és pénzügyi tervezhetőséget, az eredmények megvalósíthatóságának illetve tényleges megvalósulásának kontrollját, és a biztonsági problémák tervezhetőségét érinti. Ezeknek megfelelően alakultak ki a különböző életciklus modellek, melyek célja a fejlesztési folyamat modellezése. A szoftverfejlesztés életciklusában megjelenő általános feladatok:

1. Követelmények megfogalmazása - funkcionális specifikáció

2. Rendszertervezés (design) - rendszerterv

3. Kódolás, testreszabás és tesztelés

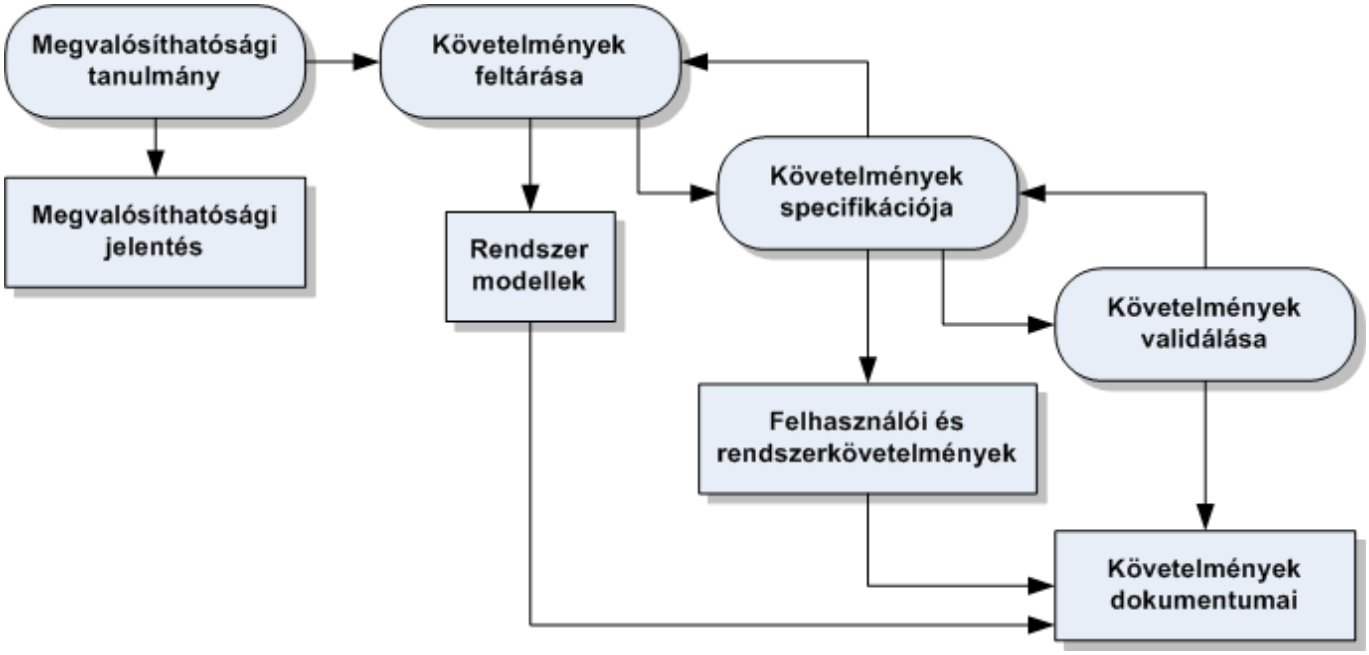
4. A szoftver bevezetése

Szoftverspecifikáció

A **szoftverspecifikáció**, avagy a **követelménytervezés** az a folyamat, ahol megértjük és definiáljuk, hogy a rendszernek milyen szolgáltatásokat kell biztosítania, és azonosítjuk a rendszer üzemeltetésének és fejlesztésének megszorításait. A követelmények tervezése a szoftverfolyamat különösen kritikus szakasza. Az ebben a szakaszban vétett hibák elkerülhetetlenül problémákhoz vezetnek majd a rendszertervezés későbbi szakaszában és az implementációban.

A folyamat eredménye a követelmény dokumentum előállítása, amely a rendszer specifikációja. A követelmények két különböző szinten kerülnek kifejtésre a dokumentumban: a végfelhasználók és ügyfelek számára leírt követelmények, amelyek megfogalmazása absztraktabb, mélyebb technoglógiai ismertek nélkül is megérthető. Ezen felül pedig a fejlesztőknek szánt sokkal részletesebb rendszerspecifikáció, amely alacsony szinten fogalmazza meg a rendszer funkcióit és megszorításait.

A követelmények tervezési folyamatát a következő ábra mutatja:



Szoftverspecifikáció fázisai

A követelmények tervezésének négy nagy fázisát különböztethetjük meg. Ezeket találjuk az ábra felső vonalában:

1. **Megvalósíthatósági tanulmány**: Gyakorlatilag ez az első fázis a szoftver életében. Megléte rendkívül fontos, mert ebben a fázisban meg kell becsülni és meghatározni, hogy a megrendelő kívánságai kielégíthetők-e az adott szoftver- és hardvertechnológia mellett. A vizsgálatoknak el kell dönteniük, hogy a rendszer költséghatékony-e, és hogy az kivitelezhető-e és azt sem szabad figyelmen kívül hagyni, hogy a tervezett szoftver megfelel-e a vállalat stratégiai céljainak. A megvalósíthatóság elemzésének relatíve olcsónak és gyorsnak kell lennie. Eredménye a megvalósíthatósági jelentés.

2. **Követelmények feltárása és elemzése**: Ez a folyamat a rendszerkövetelmények meglévő rendszereken történő megfigyelésén, a potenciális felhasználókkal és beszerzőkkel folytatott megbeszéléseken, tevékenységelemzéseken alapszik. Akár egy vagy több különböző rendszermodell, illetve prototípus elkészítését is magában foglalhatja. A feltárás egy ciklikus folyamat, amely során addig iteráljuk a gyűtési folyamatot, amíg minden részlet a helyére nem kerül. Ebben a fázisban olyan további feladatokat végzünk el, mint a

- szakterület megismerése,
- követelmények összegűjtése,
- követelmények osztályozása,
- ellentmondások feloldása,
- fontossági sorrendbe állítás.

3. **Követelmény specifikáció**: A követelményspecifikáció az elemzési tevékenységek során összegyűjtött információk egységes dokumentummá alakítása. A dokumentumnak a követelmények két típusát kell tartalmaznia:

- A felhasználói követelmények a rendszerkövetelmények absztrakt leírása, amelyek a végfelhasználóknak, illetve a megrendelőknek szólnak.
- A konkrét rendszerkövetelmények, amelyek részletezik az elkészítendő rendszer által nyújtandó

funkciókat.

4. **Követelmény-validáció**: A tevékenység ellenőrzi, hogy mennyire valószerűek, konzisztensek és teljesek a követelmények. A folyamat során fel kell tárni a követelmények dokumentumában található hibákat, és kijavítani.

Nagyon fontos megjegyeznünk, hogy nem mindig célszerű a követelménytervezés különböző tevékenységeit szigorú sorrendben végrehajtani. Például a követelmények elemzése folytatható a meghatározásuk és specifikálásuk alatt, továbbá a folyamat során bármikor napvilágra kerülhetnek új követelmények is. Bizonyos esetekben ilyenkor az elemzés, a meghatározás és a specifikáció tevékenységei összefésülhetők, és egymást átfedhetik a folyamatban.

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:infrendalapjai_architekturak:szoftvertechnologia:szoftvertechnologia_alapjai

Last update: **2024/11/13 19:21**