FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2003. március 21-22.

INTEGRÁLT PROJEKTMENEDZSMENT OPTIMALIZÁLÁS

Dobán Orsolya

Összefoglalás

The key problems in software project management are the proper estimation of the effort needed to implement a specific design, the sufficient and effective allocation of resources and development environment. A proper project plan has to have a good or optimal scheduling of the individual development subtasks. This way our main objective was to provide a methodology, which is able to take into account all the main factors influencing the project efforts and scheduling for UML¹ based design, and to generate the mathematical model of a software process optimization problem based on UML diagrams.

A mindenki számára érthető, egységes szoftver modellező nyelv, valamint a szoftverfejlesztés műszaki folyamatát támogató számos CASE eszköz megjelenése dacára mind a mai napig sok projekt esetén okoz gondot mind a szoftverfejlesztés során felmerülő költségek, mind a szükséges időráfordítás becslése. A nemzetközi trendek szerint ugyanakkor a költséghatékony szoftverfejlesztést egyre inkább komoly időkényszer mellett kell megvalósítani, ezért az erőforrások optimális felhasználása komoly versenyképességi faktor.

A célunk ezért egy olyan UML alapú keretrendszer kidolgozása volt, mely képes az erőforrások optimális allokálásával a fejlesztési időnek az ésszerű minimumra szorítására, így a feladat és a megoldás megjelenése közötti idő (Time to Market) lerövidítésére.

1. Bevezető

Az utóbbi évtizedekben nőtt az igény az egyre megbízhatóbb szolgáltatást nyújtó rendszerek készítése iránt, akár redundancia árán is. Ilyen szolgáltatásbiztos rendszer tervezésekor a stratégiai döntések az alkalmazott redundancia fajtája mellett a becsült költségek és termékminőség együttes kezelését is igénylik. A fentieknek megfelelően intenzív kutatások folynak abban az irányban, hogy a megvalósítandó szoftver funkcionalitása, mérete, a fejlesztési humán és technológiai környezet jellemzői alapján előzetes becslést lehessen adni a teljes folyamat várható költségére.

¹ Unified Modeling Language: a szabványos vizuális objektum-orientált modellező nyelv

2. Költségbecslés

Az utóbbi két évtizedben költségbecslő modellek tucatja született meg. Ezek már a tervezés igen korai fázisától kezdve, az első durva rendszerspecifikáció megalkotása után lehetővé teszik a költségek és a munkaráfordítás becslését. A költségmodellek tapasztalati úton származnak oly módon, hogy

- kiválasztják azokat a faktorokat, amelyeket meghatározónak tartanak a projektköltség kialakulásában. Ezek részben a kialakítandó szoftver termék mennyiségi jellemzőit, (pl. becsült kódhossz), részben a fejlesztési környezet minőségi adottságait (pl. fejlesztők.felkészültsége), félig szubjektíven kvantáló független változókat jelentenek.
- A fenti független változók és a költség összefüggését leíró függvényt valamely adott környezetben kiválasztott és naplózott költségű projekt halmazra mint reprezentatívnak tekintett minta populációra legjobban illeszkedő függvényként határozzák meg.
- A fenti empirikus képletet *általánosítják* a jövőbeni projektek költségének becslésére, feltételezve azt, hogy a mögött elég tapasztalat rejlik az általános érvényű alkalmazáshoz.

Az általunk alkalmazott költségbecslő modell a COCOMO² modell, melynek alkalmazott algoritmusai nyilvánosak [1], az általa figyelembe vett faktorok pedig az 1. táblázatban olvashatóak.

Terméktfaktorok	Személyi faktorok	Platform faktorok	Projektfaktorok
szoftver megbízhatóság	tervező hozzáértése	platform érzékenység	fejlesztői környezet
adatbázis mérete	programozó hozzáértése	végrehajtási idő	Kommunikáció
program komplexitása	alkalmazásbeli tapasztalat	tárkapacitás	Ütemezés
dokumentáció igény	progr. nyelv-beli tapaszt.		
újrahasználhatóság	platform tapasztalat		
	fejl. csoport folytonosság		

1. Táblázat COCOMO faktorok

3. UML bázisú ráfordításbecslés

Az egyre nagyobb méretű, bonyolultabb szoftverek elterjedésével megnőtt az igény az olyan fejlesztői környezetek iránt, amelyek a komplex szoftverrendszerek tervezését könnyen átlátható, mindenki számára egyértelmű módon teszik lehetővé. Mára már nyilvánvaló, hogy ez a cél egy egységesített, vizuális megjelenítésre is képes, ugyanakkor formális nyelvet igényel. Ezen igényre reagálva született meg az Object Management Group (OMG) által ipari szabvánnyá emelt UML (Unified Modelling Language) modellező nyelv.

² COConstructive COst MOdel, „Center for Software Engineering” (Kalifornia) csoport által kidolgozott költségbecslő módszer

A kutatási eredmények bizonyították azt is, hogy az UML nem csak a végtermék fokozatosan finomodó leírására, hanem a választott projektmenedzsment életciklus modellje, valamint a fejlesztési környezet (például háttér infrastruktúra, fejlesztők ismeretei, stb.) szabatos specifikálására is alkalmas, legalábbis a tiszta modellezés erejéig.

A jelen projekt alapgondolata az, hogy összekapcsolja a szoftverfejlesztési projekt költségét meghatározó valamennyi tényezőt (céltermék, technológia, környezet, stb.) egyetlen egységes UML modellé, azokat a modell különböző moduljaiként implementálva.

- A *termékmodell* elemi komponenseihez a költségbecslési faktorok értékét (pl. funkcionális bonyolultság vagy kódhossz) hozzárendelve már automatikusan felépíthető a megvalósítandó célrendszer műszaki szempontból teljes költségmodellje. A korábbiak szerint önálló UML modulként lehet megfogalmazni az adott fejlesztői környezet és folyamat költségbecslés szempontjából releváns modelljét, kibővítve az adott költségbecslési metodika által alkalmazott elemi költségfaktorokkal.
- Ez a kompozit költségmodell alkalmas mindazon elemzések elvégzésére, amelyekre a klasszikus projektvezetési eszközök képesek, mi több azok számára képes automatikus konverzióval a teljes bemenő leíró modelljüket származtatni.
- A rendszertervezés korai fázisaiban az UML segíti a tervezőt különböző megoldási alternatívák jól dokumentált felvázolásában. Az integrált rendszer ehhez automatikusan képes generálni a költségbecsléseket.

4. Integrált környezet

Kihasnálva azt, hogy az UML modell formális jellegű szabatosága miatt mód van abból közvetlenül matematikai modell származtatására („matematikai kódgenerálás”) több tervezési alternatíva esetén mód van azok költségmodelljének automatikus generálására. Ezen matematikai problématerben az optimumot ismert algoritmusokkal megkeresve támogatható a költség szempontjából optimális műszaki megoldás kiválasztása.

Az egyes, UML környezetben ábrázolt elemi döntési alternatívák költségfüggvényének meghatározásából egy olyan *kombinatorikus optimalizálást* megvalósító vegyes egész programozási feladat származtatható, melynek megoldása megadja az adott célfüggvény (például legrövidebb fejlesztési idő, költségminimum, vagy a team egyenletes terhelése, stb.) meghatározására szolgáló matematikai feladatot. E feladat megoldása ugyanakkor az optimális megoldáson túl még számos egyéb, a projektmenedzsment szempontjából kritikus információt is szolgáltathat:

- Az optimális megoldás mellett jelzi, hogy mik a szűk keresztmetszetek, illetve milyen szabad kapacitások maradtak fenn.

- Az érzékenységvizsgálat módot ad arra, hogy megvizsgálja az egyes paraméterek kihatását az optimumra. Ez két szempontból is kitüntetett:
 - egyrészt módot ad a leginkább *költségérzékeny faktorok* meghatározására,
 - másrészt megmutatja az optimális megoldás szempontjából *kritikus faktorokat* (azokat, amelyek esetében a optimalizálandó célfüggvény maximálisan változik). Az ilyen faktorok esetében ugyanis mód van a költségbecslés egyes tényezőiben elkövetett hibák hatásának a felmérésére, azaz azon tényezők meghatározására, melyek becslésénél a tévedés a projekt egésze számára nagy kockázattal jár.

5. Összegzés

Az általunk alkalmazott megközelítésnek előnye az, hogy a tényleges műszaki tervezéssel integrált környezetbe helyezi el a korábban teljesen elkülönült, durva külső becsléseken alapuló projektmenedzsment aspektust. Ez a környezet a ma legkorszerűbbnek tekintett UML nyelvre épül, ahol akár külön adatgyűjtés nélkül is, csak kis, lokális adatok folyamatos megadásával, illetve karbantartásával a különböző vezetői szintek információellátása megoldhatóvá válik.

Hivatkozások

- [1] Barry W. Boehm: Software Cost Estimation with COCOMO II., Prentice Hall PTR, 2000, New Jersey
- [2] O. Dobán, A. Pataricza, Cost Estimation Driven Software Development Process, Proceedings of the 27th EUROMICRO Conference, ISBN 0-7695-1236-4, pp. 208., Warsaw, Poland, 4-6 September 2001
- [3] Dobán O, Pataricza A., Pintér G., Szécsi G., UML alapú költségbecslés, IKTA 00194/2000 - 3.
- [4] Object Management Group, *Object Constraint Language*, 2002.
- [5] A. Pataricza, Gy. Csértán, O. Dobán, A. Gábor, J. Sziray: Process Modeling and Optimization in UML, IEEE International Conference on Intelligent Engineering Systems, INES-2001, Proceedings, pp. 457-461, Helsinki, September 16-18, 2001.

Köszönetnyilvánítás

Köszönettel tartozom témavezetőmnek, Dr. Pataricza Andrásnak, aki tanácsaival segítette munkám előrehaladását.

Dobán Orsolya doktorandusz hallgató

Budapesti Műszaki és Gazdaságtudományi Egyetem,

H-1117 Budapest, Magyar tudósok körútja 2.

Tel.: +36-1-4633579, Fax: +36-1-4632667, E-mail: doban@mit.bme.hu