

Tézisek

A dolgozat klasszikus filozófiai megközelítéseket alkalmaz modern empirikus anyagon, a szoftverfejlesztés gyakorlatán. Ez a klasszikus elgondolások újszerű átértelmezését, adaptálását, illetve kritikáját tette szükségessé. Munkám legnagyobb átfogó eredményének éppen ezért a két, egymástól távol fekvő terület közötti kapcsolat kiépítését tartom. Téziseim részletesen az alábbiak.

T1. A szoftverfejlesztés, a szoftverfejlesztők közötti kommunikáció és a szoftver felhasználó általi appropriációja *hermeneutikai tevékenységként (hermeneutic practice)* jellemezhetőek [12.]. Ezt a fogalom a „hermeneutika” kifejezés több különféle elmélet kontextusában betöltött jelentése közül az alábbi jelentéskomponenseket hangsúlyozza ki:

- a.) *A hermeneutikai tevékenység egy olyan értelmezési folyamat, ami nem explicit, nyelvi artikulált módon, hanem készségszerű tevékenységben (skillful coping) bontakozik ki.* Az eredményképpen kialakult megértés nem tézisek vagy fogalmak birtoklását, hanem egyfajta *eligaződési képességet* takar. Ez még a teljesen szimbolikus forráskódok megértésére is igaz.
- b.) *Minden tevékenység magában foglal hermeneutikai összetevőt,* mert a tevékenység folyamán a világ számunkra már eleve célszerűnek mutatkozik, cselekvési és életheletőségek feltárlásaként vagyunk képesek értelmezni.
- c.) *A hermeneutikai tevékenység szituált és megtestesült, azaz az értelmező testi-perceptuális készségeinek, a szituáció tárgyi és társas konkrétumának az összefüggésébe ágyazott, vizsgálata nem szakítható ki ebből az összefüggésből.* Tárgyak, szimbólumok jelentése más tárgyak, társas helyzetek, hagyományos készségek, stb. hermeneutikai tevékenységben játszott *lokális* összefüggésében érthető meg.
- d.) *A megértési folyamat egy egészes jelentés-összefüggés (szituált és megtestesült) életvilágbeli horizontjába, és egy értelmezési tradícióba való előzetes beléhelyezkedést előfeltételez és ennek az elmozdulásával jár együtt.* A programkód egyetlen szimbólumának a megértése további szimbólumok, készségek, elméleti tradíciók, élethelyzetek precízen nem lehatárolható halmazának a megértését előfeltételezi.
- e.) *A megértési folyamat szükségképpen körben mozgó, mert a visszakérdezésen és az ismételt értelmezési próbálkozásokon kívül nincsen más mód a megértés közeledésének a megítélésére.* Ez a körmozgás a beszélgetés folyamán a visszakérdezésben és korábban elmondottak folyamatos újraértékelésében áll, a szövegek esetében a lokális töredéknek a szöveg globális kontextusában való újraértékelését jelenti, míg a szoftverek értelmezésekor a velük való aktív interakció és kísérletezés során épül ki a megértés.

Az így bevezetett fogalom relevanciáját a dolgozat három fő esettanulmánya, ezeken belül pedig több mint hat, mind a nyílt forráskódú, mind pedig a kereskedelmi, illetve haditechnikai célú fejlesztések köréből választott eset részletesen kidolgozott elemzése támasztja alá.

T2. *A szoftver a hermeneutikai tevékenység közegeként meghatározhatja a benne és általa adható gyakorlati értelmezések terét:* a vele interakcióba lépők megértésének, tapasztalásának és cselekvésének a lehetőségterét. A programnyelvek és keretrendszerek esetében ez például azt jelenti, hogy a rendszer meghatározhatja a benne alkalmazott fogalmi absztrakciók lehetséges dimenzióit és irányait, ami kihat a problémák konceptualizálására és megoldására [13.].

T3. *A szoftver és a számítógépes programnyelv egy hatalmi mezőt alkothat azáltal, hogy a fejlesztők egyes gyakorlati értelmezések és interakciós mintázatok iránti preferenciákat építenek a szoftverbe.* A szoftver így hagyományos hatalmi hierarchiák megerősítésére, a felhasználók tevékenységének normalizálására, *kötelező áthaladási pontok (obligatory passage point, Latour)* létrehozására használható fel, amint azt a dolgozatban az Ada programnyelvről és a Netscape/Mozilla projektről szóló elemzések demonstrálják [12.].

T4. *A szoftver a felhasználói appropriáció (user appropriation) során – a szövegekhez és más tárgyakhoz hasonlóan – olyan újraértelmezésen mehet keresztül, amely a fejlesztők által a szoftverbe épített interakciós mintázatoktól eltérő, nem várt használatmódokat eredményez.* A dolgozat harmadik fő tanulmánya a Netscape/Mozilla projekt történetének rekonstrukcióján keresztül demonstrálja ezt a folyamatot, rámutatva arra, hogy a szabad és nyílt forráskódú szoftvereknek a jelentősége ezekben az appropriációs folyamatokban különösen nagy, mert itt nagyobb tere

nyílik az átértelmezésnek, és ez nagyobb nyilvánosság előtt zajlik.