
CASE

Author(s): Danijel BOLDIN

Source: *Urbani Izziv*, No. 14, INOVATIVNO MESTO (november 1990), pp. 77-78

Published by: Urbanistični inštitut Republike Slovenije

Stable URL: <https://www.jstor.org/stable/44179967>

Accessed: 26-09-2018 10:48 UTC

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <https://about.jstor.org/terms>



This article is licensed under a Attribution 4.0 International (CC BY 4.0). To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>.



JSTOR

Urbanistični inštitut Republike Slovenije is collaborating with JSTOR to digitize, preserve and extend access to *Urbani Izziv*

Danijel BOLDIN CASE

Pri razvoju informacijskih sistemov za prostorsko planiranje sodelujejo v fazi razvoja programske opreme (aplikacije) poleg bodočih uporabnikov planerjev tudi različni računalniški specialisti (sistemski analitiki, programerji, skrbniki baz podatkov ipd.). Za to, da je aplikacija razvita v dogovorjenem roku in z ustrežno kvaliteto, je pri procesu razvijanja aplikacije pomembno sodelovanje med uporabnikom in računalniškimi specialisti. Na osnovi organizacijske in tehnološke zasnove aplikacije je možno izdelati programsko rešitev. Po navadi se za to uporabijo določeni programski jeziki ali pa sistemi četrte generacije programov (sistemi baz podatkov, npr. ORACLE, dBASE III +, ekspertni sistemi itd.)

V fazi zasnove programske opreme in pripravi prvih t. i. beta verzij programske opreme se velikokrat primeri, da je potrebno že pripravljen podatkovni model spremeniti (predvsem po razgovorih z uporabniki planerji, ki po navadi zelo težko opišejo svoje potrebe oz. določijo, katere podatke potrebujejo). V primeru, da osnovni podatkovni model razširimo, dodamo nove relacije ipd., mora skupina programerjev, ki pripravja beta verzijo uporabniškega programa, spremeniti veliko stvari, kar zahteva veliko časa. Ravno pri pripravi podatkovnega modela in prvih verzij programske opreme pa bi bila smiselna uporaba sistemov CASE.

Kaj so sistemi CASE

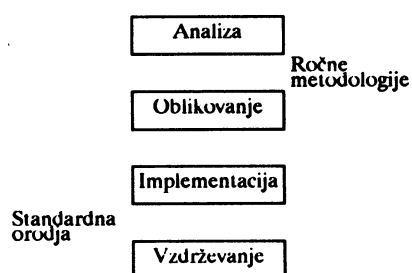
Sistemi CASE so zasnovani tako, da omogočajo avtomatizirano oblikovanje - razvijanje aplikacij s pomočjo računalnika. CASE je kombinacija programskih orodij in strukturiranih programskih metodologij. Orodja avtomatizirajo programske procese, metodologije pa definirajo načine avtomatizacije. Tehnologija CASE je orientirana na povečanje produktivnosti, znanstveni pristop pri raz-

vijanju programov in delo v realnem času.

Osnovna ideja sistemov CASE je omogočiti uporabo dobro integriranih orodij, ki zmanjšajo število napak in potrebnih ur za razvoj programov ter avtomatizirajo celoten proces razvoja programske aplikacije. Ti sistemi nam nudijo podporo pri razvoju in izvedbi aplikacije, vključno z izdelavo dokumentacije in planiranjem celotnega projekta. Sistemi orodij so zasnovani modularno, kar omogoča uporabo samo določenega modula glede na naše zahteve.

Tradicionalno programsko tehnologijo lahko razdelimo na dva tipa: orodja in metodologije. Orodja po navadi vključujejo tretjo, četrto ter v zadnjem času tudi peto generacijo teh orodij. Večina teh orodij je zasnovana na samostojnem delovanju, orientirana na delo na velikih ali miniračunalniških sistemih ter omejena na le en del razvoja programske rešitve. Programske metodologije pa vključujejo ročno uporabo ter metodologije, kot so strukturalna analiza, strukturalno oblikovanje in strukturalno programiranje. Te metodologije so zasnovane tako, da zahtevajo razvoj programa korak za korakom.

Schema tradicionalnih programskih tehnologij



Tehnologija CASE ni neka povsem novatehnologija pri razvoju programske opreme, ampak temelji na tehnikah in orodjih, ki so se pokazala uporabna v praksi. Ker so orodja CASE namenjena povečini poklicnim razvijalcem programov in ne končnim uporabnikom, ta niso le posebnost četrte generacije programskih orodij.

Različni sistemi CASE podpirajo različna razvojna orodja. Podpirajo predvsem nekatere znane strukturalne metode za izdelavo programskih diagramov: De Macro, Gane/Sarson, Yourdon, Jacksonovo strukturirano oblikovanje ter Martinov informacijski inženiring.

Generatorji programov so namenjeni oblikovanju specifikacij programskega sistema. Vključujejo oblikovanje ekranov in poročil, simulacije, uporabo prototipov ter funkcije za kontrolo napak.

Oblikovanje podatkovnih baz, pregledovanje in izpis poročil se uporabljajo pri modeliranju podatkovnih baz - podatkovnih shem. Izdelujejo se tudi opisi podatkovnih datotek. Po navadi se uporabljajo naslednje metode: IDEF/Leverge, Chen Toolkit, IDMS/Arhitekt, CASE*Design - Oracle.

Če sistem popravljamo (prestrukturiranje), se avtomatično popravijo tudi že kreirani programi. Tu se po navadi uporabljajo naslednje metode: Adpac CASE Tools, Scan/COBOL, SuperStructure, Via/SmarTest ipd.

Generatorji programov omogočajo programiranje in testiranje posameznih korakov pri programiranju. Uporabljajo se predvsem naslednje metode: P-Source in P-Tools, CoFac ipd. Ta orodja avtomatično kreirajo popolnoma dokumentiran program, ki ga lahko takoj izvajamo. Vključeni so postopki za oblikovanje ekranov, logične kontrole, testiranje in iskanje napak, pisanje posameznih vrstic programov in dokumentiranje programov.

Vzdrževanje sistema zahteva spremljanje izvajanja, kontrolo, usklajevanje in izpise različnih poročil, ki so potrebna za razvijanje programov in usklajevanje projekta. Primer takega orodja je: Project Workbench.

Shema integriranih orodij CASE

Integrirana orodja	Priprava diagramov	Orodja za risanje strukturnih diagramov in grafično oblikovanje specifikacij
	Generatorji ekranov in poročil	Orodja za pripravo uporabniških ekranov in sistemskih specifikacij
	Priprava poročil	Pripomočki za shranjevanje, poročila in ureditev vseh sistemskih informacij
	Kontrola	Orodja za kontrolo konsistence, popolnosti in pravilnosti sistemskih specifikacij
	Generatorji programov	Orodja za generiranje izvajalnih programov na osnovi specifikacij sistema
	Vzdrževanje	Orodja za dokumentiranje in analizo sistema

Problemi pri uporabi CASE

Eden prvih problemov (morda celo največji), s katerim se srečamo pri uporabi CASE, je množica možnosti, ki so nam na voljo. Predvsem se pojavljajo težave pri razumevanju in uporabi. Kaj omogoča posamezno orodje? Katero orodje uporabiti za določen problem?

Ker so to novi sistemi, še ne obstajajo standardi na tem področju. Tako se ti sistemi zelo razlikujejo med seboj, nimajo enotnega koncepta, ni poenostavitve ipd. Nastaja velik razkorak med željami in možnostmi. Zelo velika pričakovanja (nerazumevanje CASE) vodijo v zmote, ker posamezna orodja CASE ne rešijo takoj vseh uporabniških problemov. Drugi velik problem je uporaba orodij CASE za napačne probleme. Težave se pojavljajo tudi pri produktivnosti. Sistemi CASE omogočajo učijo, kako delati bolje in ne delati napačno hitreje.

Zelo velik pomen pri uporabi CASE ima izobraževanje. Brez predhodne organizacijske priprave in ustreznega izobraževanja uporabnikov ni moč pričakovati uspehov. Na vsakem koraku pri implementaciji metodologije CASE se pojavljajo nova vprašanja in novi problemi. To zahteva večje razumevanje problematike, več informacij in boljše

organiziranje. Samo zato, ker imamo nova orodja, še ne pomeni, da smo tudi ustrezno usposobljeni za njihovo uporabo.

Razvijalci programov po navadi nimajo metodološkega pristopa pri svojem delu, kar pa zahtevajo orodja CASE. Po navadi razvijajo programe po trenutnem navdihu, kjer ima vsak svojo knjižnico že izdelanih postopkov - orodij. Mnogi izmed njih nimajo ustreznih znanj o formalnih metodologijah, ter jih uporabljajo bolj naključno.

V tuji literaturi lahko zasledimo veliko prispevkov o orodjih CASE. Mnenja o uporabnosti teh orodij so si še zelo deljena. Na osnovi ankete, ki je bila objavljena v časopisu Computerworld, se je izkazalo, da kakšnih 50% uporabnikov sistemov CASE (razgovor s 152 uporabniki) meni, da se bodo prednosti CASE orodij šele pokazale. Nekako 38% jih smatra, da je preveč orodij in premalo standardov. Celo 54% uporabnikov pa je prepričanih, da ni dovolj informacij o samem sistemu CASE. Samo 24% vprašanih je potrdilo uporabo teh orodij v svojih organizacijah, dvakrat toliko 47% pa jih predvideva, da bodo v naslednjem letu pričeli z uvajanjem orodij CASE.

Zelo veliko je domnev okoli IBM-

ovega odgovora orodjem CASE. To, da IBM ni objavil svojega konkretnega odgovora tem sistemom, kaže, da bo "okolje za razvoj aplikacij" ADE (Application Development Environment) IBM-ov odgovor orodjem CASE. Pričakuje se, da ADE ne bo tipičen izdelek CASE, ampak bo integrirano okolje, ki bo omogočalo različnim produktom CASE delovati skupaj, kot če bi bili integrirani v celoto. To pomeni, da morajo biti posamezne komponente, ki delujejo znotraj sistema CASE najprej povezane med seboj, nato pa mora biti celoten sistem CASE integriran v drug sistem.

Zaključek

Vsekakor je potrebno precej časa za uvajanje orodij CASE. Več ko je splošnega znanja o sistemih CASE, seminarjev, kontaktov med uporabniki, več je pravih odgovorov na zastavljena vprašanja.

Velik problem ima standardizacija. Nobeden izmed razvitih sistemov nima popolne izbire možnosti, ki jih nudi CASE. Če pogledamo sisteme CASE, za katere razvijalci trdijo, da nudijo optimalne možnosti, so to le ugotovitve razvijalcev orodij CASE in ne uporabnikov.

Ne glede na vse omenjene težave, ki se pojavljajo pri uvajanju tehnologije CASE, pa se je pokazalo, da je potreben še nadaljnji razvoj na tem področju. Prednosti, ki jih nudijo orodja CASE, bodo v prihodnosti prišle še bolj do veljave.

Literatura:

1. GRIČAR Jože: Metodološki pristop k oblikovanju računalniško zasnovanega informacijskega sistema, Organizacija in kadri, 7-8/1987.
2. BOBEK Samo: Metode snovanja računalniško podprtega poslovnega informacijskega sistema in njihova računalniška podpora, Organizacija in kadri, 7-8/1987.
3. MERLYN Vaughan: CASE, Computerworld, 11/1988.
4. LUCAS GIBSON Michael: The CASE Philosophy, BYTE, 4/1989.
5. McCLURE Carma: CASE is software automation, Prentice Hall, 1989.
6. TOWER Larry: CASE Concepts and implementation, McGraw-Hill, 1990.

Danijel Boldin, inž. geod.