

Nástroje CASE a ich využitie v reverznom inžinierstve*

Šimon Ukus

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xukus@stuba.sk

6. november 2021

Abstrakt

Článok sa zaoberá problematikou softvérového inžinierstva, konkrétne ako je možné proces vývoja softvéru automatizovať pomocou nástrojov CASE - Computer-Aided Software Engineering (Počítačom podporované softvérové inžinierstvo). V práci sa uvádza klasifikácia týchto nástrojov. Článok ďalej skúma softvérové inžinierstvo a použitie CASE nástrojov z inej perspektívy. Na rozdiel od vnímania vývoja softvéru klasicky, teda smerom vpred (Forward Engineering) sa sústreďí na tzv. spätné inžinierstvo (Reverse Engineering). Tu je skúmaná kompletnosť a presnosť spätne navrhnutých UML diagramov generovaných nástrojmi CASE. Predmetom porovnania bolo celkom 8 nástrojov (z toho 6 open source a 2 komerčné). Tieto nástroje boli hodnotené na základe toho, aké typy vstupov podporujú, aké typy diagramov dokážu rekonštruovať a v akej kvalite.

1 Úvod

Pojem softvérové inžinierstvo môže byť chápaný ako uplatňovanie metód, postupov a nástrojov na riadenie a vývoj počítačových systémov [3]. Ide o komplexný postup, na ktorom sa zúčastňuje mnoho odborníkov z rôznych oblastí, ako napríklad projektový manažér, team líder, softvérový developer, tester, UI dizajnér a mnoho ďalších. Časť ich práce je možno automatizovať či uľahčiť využitím nástrojov CASE. Tieto nástroje sú špeciálne vyvinuté pre podporu vývoja softvéru, automatizujú proces vývoja. Cieľom ich implementácie je ušetrenie času a nákladov pri vývoji softvéru a zvýšenie jeho kvality [1].

Využitie nástrojov CASE a ich klasifikácia sú uvedené v časti 2., kde je popísané delenie podľa toho, ktorú časť tzv. životného cyklu vývoja softvéru (Software Development Life Cycle, ďalej len SDLC) pomáhajú automatizovať. V časti 3 sa čitateľ zoznámí s pojmom *reverzné inžinierstvo*. Nachádza sa tu prehľad, v ktorom je popísaný rozdiel medzi tzv. *forward* (dopredným) a *reverse* (spätným) inžinierstvom. Samotným nástrojom, ich predstaveniu a následnému porovnaniu sa venujú časti 4 a 5. Záverečné hodnotenie týchto nástrojov prináša časť 6.

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22 vedenie: Vladimír Mlynarovič

2 Klasifikácia nástrojov CASE

Nástroje CASE sú odpoveďou na stále narastajúce nároky a zvyšujúcu sa komplexnosť počítačových systémov. Podporujú vývoj softvéru a dajú sa aplikovať v niektorých, niekedy vo všetkých fázach SDLC, ktorého podpora je čoraz žiadúcejšia. Náklady na vývoj softvéru každým rokom vzrastajú, a preto čo i len skromné vylepšenia pri vývoji a automatizácii môžu znamenať veľké úspory. Sú cielené na riešenie ťažkostí pri vývoji vysokokvalitného a komplexného softvéru načas a v súlade s rozpočtom [7].

Ako sa vyššie spomína, nástroje CASE slúžia na podporu rôznych fáz SDLC, prípadne dokážu automatizovať všetky z nich. Zvyknú sa preto klasifikovať podľa toho, ktoré štádium životného cyklu podporujú. Takéto rozdelenie vyzerá nasledovne:

- Upper CASE nástroje
- Lower CASE nástroje
- Integrated CASE nástroje

Upper CASE, niekedy označované aj ako *front end CASE* slúžia na podporu skorých fáz životného cyklu softvéru, napríklad pri analýze a dizajne.

Lower CASE, tiež nazývané aj *back end CASE* zase nachádzajú využitie v neskorších fázach životného cyklu softvéru, najmä pri testovaní a vytváraní kódu.

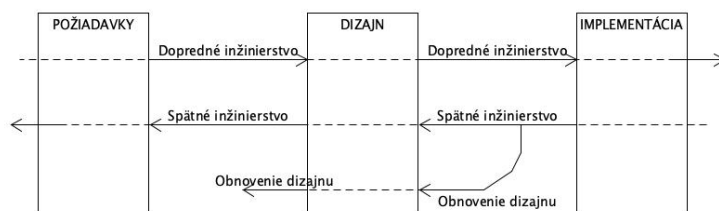
Integrated CASE, sú schopné pokryť obe časti SDLC [4].

Iný pohľad na kategorizáciu poskytuje autor [5], ktorý nespája fázy SDLC do tzv. skorých a neskorších, ale konkrétne fázy jednotlivo vymenúva a podľa toho delí nástroje CASE nasledovne:

- Nástroje na riadenie projektov
- Nástroje na analýzu a návrh
- Nástroje na podporu Objektovo-Orientovaného softvérového inžinierstva
- Nástroje na testovanie
- Nástroje formálnych metód
- Klient/Server nástroje
- Nástroje pre webové inžinierstvo
- Nástroje na opätovné inžinierstvo (Reengineering)

Pri tomto delení sa stretávame s pojmom opätovné inžinierstvo (Reengineering), a keďže v ďalších kapitolách je rozpracovaná téma reverzného inžinierstva a použitie nástrojov CASE v reverznom (spätnom) inžinierstve, je potrebné uviesť, že je rozdiel medzi spätným inžinierstvom (Reverse Engineering) a opätovným inžinierstvom (Reengineering). Síce oba pojmy odkazujú na ďalšie skúmanie alebo vývoj už hotových produktov, metódy a požadované výsledky sa výrazne líšia. Ako [6] ďalej vysvetľuje, reverzné inžinierstvo sa snaží odhaliť, ako daný systém funguje. Na druhej strane, úlohou opätovného inžinierstva je zlepšenie súčasného návrhu skúmaním jeho konkrétnych aspektov.

3 Z dopredného inžinierstva k spätnému



Obr. 1: Vybrané fázy SDLC a znázornené procesy súvisiace s dopredným a spätným inžinierstvom (prevzaté a preložené z [2])

Spätné inžinierstvo, ako už bolo v sekcii 2 spomenuté, sa snaží pochopiť funkciu systému. Obr. 1 ilustruje spôsob vývoja informačných systémov. Pre jednoduchosť boli použité tri fázy životného cyklu, kde prvá - "požiadavky" predstavuje špecifikáciu problému, druhá - "dizajn" zase špecifikáciu riešenia a tretia - "implementácia" predstavuje programovanie a testovanie požadovaného systému.

Pojem dopredné (*forward*) inžinierstvo reprezentuje tradičný proces vývoja softvéru, kedy sa postupuje od prvotných abstraktných modelov, cez dizajn až ku konečnej implementácii systému. Zavedenie názvu pre niečo tak priamočiare sa môže zdať zbytočné, možno aj máťúce. V našom prípade, a tiež v mnohých iných, je to však nevyhnutné pre jeho odlišenie od spätného inžinierstva. Ako teda Obr 1 ukazuje, dopredné inžinierstvo prechádza jednotlivými fázami SDLC z ľava do prava.

// Tu príde ešte popis reverzného inžinierstva spolu s odkazovaním sa na obrázok.

č.	nástroj CASE	informácie	Predajca	Typ Licencie
1	Visual Paradigm 8.1	http://www.visual-paradigm.com/	Visual Paradigm	Testovanie
2	MagicDraw 17.0	http://www.magicdraw.com/	No Magic	Testovanie (akademický)
3	Altova Umodel 2011	http://www.altova.com/	Altova	Testovanie
4	Enterprise Architect 8.0	http://www.sparxsystems.com.au/	Sprax System	Testovanie
5	Rational Software Architect 8.0.1	https://www.ibm.com/products/rational-software-architect-designer	IBM	Testovanie
6	MyEclipse 8.6	https://www.genuitec.com/products/myeclipse/features/	Genuitec	Testovanie (akademický)
7	StarUML 5	http://staruml.sourceforge.net/	StarUML	Open Source
8	ArgoUML	https://argouml-tigris.org.github.io/tigris/argouml/tours/	Tigris.org	Open Source

Obr. 2: Zoznam posudzovaných nástrojov (prevzaté a preložené z [8])

č	nástroj CASE	informácie	Predajca	Typ Licencie
1	Visual Paradigm 8.1	http://www.visual-paradigm.com/	Visual Paradigm	Komerčné
2	MagicDraw 17.0	http://www.magicdraw.com/	No Magic	Komerčné
3	Altova Umodel 2011	http://www.altova.com/	Altova	Komerčné
4	Enterprise Architect 8.0	http://www.sparxsystems.com.au/	Sprax System	Komerčné
5	Rational Software Architect 8.0.1	https://www.ibm.com/products/rational-software-architect-designer	IBM	Komerčné
6	MyEclipse 8.6	https://www.genuitec.com/products/myeclipse/features/	Genuitec	Komerčné
7	StarUML 5	http://staruml.sourceforge.net/	StarUML	Open Source
8	ArgoUML	https://argouml-tigris-org.github.io/tigris/argouml/tours/	Tigris.org	Open Source

Tabuľka 1: Zoznam posudzovaných nástrojov (prevzaté a preložené z [8])

4 Predstavenie nástrojov

Táto sekcia popisuje nástroje, ktoré boli predmetom skúmania, realizovaného autorom [8]. Všetkých 8 posudzovaných nástrojov je uvedených v tabuľke 2. Pri evaluácii komerčných nástrojov bola použitá verzia pre účely testovania, prípadne verzia na vzdelávacie účely.

5 Porovnanie nástrojov

6 Zhrnutie

Tento článok skúmal automatizáciu SDLC pomocou nástrojov CASE. Boli uvedené dve rôzne klasifikácie, na základe ktorých možno nástroje CASE rozdeliť. Obe sice vychádzajú z jednotlivých fáz životného cyklu, no jedno delí nástroje podľa konkrétnych fáz a druhé ich zase stručnejšie delí do ranných a neskorých fáz. V práci je ďalej opísané tzv. *dopredné* inžinierstvo a potreba zavedenia tohto prídavného mena. Neskôr je v článku popísané aj reverzné inžinierstvo a stručne uvedená potreba jeho využitia. V posledných častiach sa článok zameriava na ohodnotenie nástrojov CASE v reverznom inžinierstve. Nasledujúce závery, prevzaté od [8], boli vyvedené na základe toho, aké schopnosti a funkcie porovnávané nástroje mali. // *tu prídadne zbytok zhodnotenia práce.*

Literatúra

- [1] O. I. A. Ashour and T. Pusatli. Adoption of case tools & UML. ACM, Nov. 2020.

- [2] E. Buss and J. Henshaw. A software reverse engineering experience. ACM Press, 2010.
- [3] A. F. Case. Computer-aided software engineering (CASE). 17(1):35–43, Sept. 1985.
- [4] N. L. Chervany and D. Lending. CASE tools. 19(2):13–26, Apr. 1998.
- [5] G. Dias. Evolvment of computer aided software engineering (case) tools: A user experience. *International journal of computer science and software engineering*, 6(3):55, 2017.
- [6] L. J. Fagleman. What is the difference between reverse engineering and re-engineering?, Mar 2019.
- [7] L. Fowler, J. Armarego, and M. Allen. CASE tools: Constructivism and its application to learning and usability of software engineering tools. 11(3):261–272, Sept. 2001.
- [8] M. H. Osman and M. Chaudron. Correctness and completeness of case tools in reverse engineering source code into uml model. *The GSTF Journal on Computing (JoC)*, 1, 01 2012.