

# Nástroje CASE a ich využitie v reverznom inžinierstve\*

Šimon Ukus

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

3. 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 popisuje či už delenie týchto nástrojov, tak aj sféry ich využitia. 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 venuje tzv. spätnému inžinierstvu (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 [2]. 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. V tejto časti 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

---

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

porovnaniu sa venujú časti 4 a 5. Záverečné hodnotenie týchto nástrojov prináša časť 6.

## 2 klasifikácia nástrojov CASE a sféry použitia

Nástroje CASE sú odpoveďou na stále sa 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 [6].

### 2.1 Klasifikácia nástrojov CASE

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 pri neskorších fázach životného cyklu softvéru, najmä pri testovaní a vytváraní kódu.

**Integrated CASE**, sú schopné pokryť obe časti SDLC [3].

Iný pohľad na kategorizáciu poskytuje autor [4], ktorý nespája fázy SDLC do tzv. skorých a neskorších, ale konkrétne fázy jednotlivito vymenúva a podľa toho delí nástroje CASE delí 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 [5] ďalej vysvetľuje, reverzné inžinierstvo sa snaží odhaliť, ako daný systém funguje. Kdežto na druhej strane úlohou opätovného inžinierstva je zlepšenie súčasného návrhu skúmaním jeho konkrétnych aspektov.

## 2.2 Sféry použitia nástrojov CASE

//rozpracovaná kapitola, zatiaľ prázdna, neskôr pridám text.

## 3 Z dopredného inžinierstva k spätnému

## 4 Predstavenie nástrojov

## 5 Porovnanie nástrojov

## 6 Zhrnutie

## 7 Nejaká časť

Z obr. 1 je všetko jasné.

Aj text môže byť prezentovaný ako obrázok. Stane sa z neho označný plávajúci objekt. Po vytvorení diagramu zrušte znak % pred príkazom `\includegraphics` označte tento riadok ako komentár (tiež pomocou znaku %).

Obr. 1: Rozhodujúci argument.

## 8 Iná časť

Základným problémom je teda... Najprv sa pozrieme na nejaké vysvetlenie (časť 8.1), a potom na ešte nejaké (časť 8.1).<sup>1</sup>

Môže sa zdať, že problém vlastne nejestvuje [?], ale bolo dokázané, že to tak nie je [?,?]. Napriek tomu, aj dnes na webe narazíme na všelijaké pochybné názory [?]. Dôležité veci možno *zdôrazniť kurzívou*. [7] skúsím pridať citáciu *toto bude zdôraznené* Niekedy budem chcieť citovať toto [1]. OK?

### 8.1 Nejaké vysvetlenie

Niekedy treba uviesť zoznam:

---

<sup>1</sup>Niekedy môžete potrebovať aj poznámku pod čiarou.

- jedna vec
- druhá vec
  - x
  - y

Ten istý zoznam, len číslovaný:

1. jedna vec
2. druhá vec
  - (a) x
  - (b) y

## 8.2 Ešte nejaké vysvetlenie

**Veľmi dôležitá poznámka.** Niekedy je potrebné nadpisom označiť odsek. Text pokračuje hneď za nadpisom.

## 9 Dôležitá časť

## 10 Ešte dôležitejšia časť

## 11 Záver

## Literatúra

- [1] O. I. A. Ashour and T. Pusatli. Adoption of case tools & UML. ACM, Nov. 2020.
- [2] A. F. Case. Computer-aided software engineering (CASE). 17(1):35–43, Sept. 1985.
- [3] N. L. Chervany and D. Lending. CASE tools. 19(2):13–26, Apr. 1998.
- [4] 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.
- [5] L. J. Fagleman. What is the difference between reverse engineering and re-engineering?, Mar 2019.
- [6] 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.
- [7] 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.