

Vlastnosti a typy CASE nástrojů a jejich význam v analýze a návrhu informačních systémů.

Thursday, May 30, 2013 8:24 AM

CASE - obecně

CASE – Computer Aided Software Engineering – jsou programy určené k tomu, aby **podporovaly vývoj informačních systémů.**

Společný pohled - používání CASE nástrojů umožňuje designerům, programátorům, testerům a manažerům (tedy všem, kteří se na vývoji systému podílejí) mít **společný náhled** na to, jak projekt vypadá jako celek, jak jeho jednotlivé části v detailu a jaký je jeho stav v jednotlivých fázích vývoje.

CASE:

- **pomáhá** zajistit to, že proces vývoje projektu je řízený, říditelný a kontrolovatelný.
- **čelí složitosti** systému, která by bez jejich pomoci byla těžko zvládnutelná.
- **zajišťuje kvalitu procesů** vývoje softwaru (díky použitým metodikám a odhalování chyb při jejich použití).
- **zajišťuje značnou úsporu času** (a tedy nákladů) potřebného k vývoji systému.
- **slouží jako úložiště** projektové dokumentace.

Některé CASE nástroje jsou přímo integrovány do vývojových prostředí.

Odhady hovoří o tom, že použití CASE (přes počáteční zpomalení) nástrojů představuje úspory okolo **50 až 70 procent** v dalších etapách životního cyklu softwaru.

CASE nástroje jsou založeny na dvouvrstvé architektuře.

Základ každého z nich tvoří tzv. „**repository**“, kam se ukládají veškeré informace o navrhovaném systému (jedná se o databázi, která automaticky udržuje data v konzistentním stavu).

Nad společným repository pracuje druhá modelová vrstva, která zpřístupňuje informace uložené v repository.

Každá z modelových vrstev se opírá o jistou metodiku a reprezentuje jistý pohled na informace uložené ve společném repository. Jednotlivé modely jsou díky společnému repository na sebe vzájemně převoditelné (např. z diagramu tříd můžeme vygenerovat fyzický datový model).

Vývoj a typy CASE nástrojů

- CASE systémy vznikly v **sedmdesátých letech dvacátého století**, v situaci, kdy začala prudce narůstat složitost IS.
- CASE se na trhu začaly výrazně prosazovat zhruba v **polovině osmdesátých let**.
- Tyto systémy vznikly v okamžiku dosažení kritické kvality v metodách, organizaci práce a technologiích, potřebných při vývoji informačních systémů. Od podpory čistě vývojových fází životního cyklu IS (analýzy a konstrukce systému) k podpoře strategických rozhodování na počátku projektu a operativních činností souvisejících s provozem systému a řízením jeho změn a rozvoje.
- To, co dalo podnět ke vzniku CASE nástrojů a určuje také směry dalšího vývoje, je **metodikou tvorby informačních systémů – strukturální a objektové metodiky.**

- Postupem doby a s měnícími se požadavky dnes existuje celá řada CASE nástrojů. Je to díky podporovaným metodikám a samozřejmě také tím, v **jaké fázi vývoje je nástroj používán**.
- **Cílem je využít CASE ve všech fázích životního cyklu IS od specifikace požadavků, analýzy, návrh a kódování po údržbu IS.**

Kategorie CASE nástrojů

Podle podporovaných fází životního cyklu systému lze CASE nástroje rozdělit do dvou základních kategorií:

- **Integrované CASE.** Zaměřují se na podporu celého životního cyklu vývoje IS.
- **Specializované CASE.** Tyto nástroje jsou orientované na určité specifické etapy.

Specializované CASE nástroje

Nástroje používané v různých etapách se liší. Většinou pokrývají jen určité činnosti. Podle životního cyklu vývoje lze CASE nástroje rozdělit dle na:

- **Pre CASE** –
Tyto nástroje jsou určeny pro tvorbu celkové strategie IS.
- **Upper CASE** –
Nástroje této kategorie podporují plánování, specifikaci požadavků, modelování organizace podniku a celkovou analýzu IS.

Hlavním úkolem je analýza organizace, zachycení všech procesů, definice klíčových toků a dokumentace zjištěných požadavků. Použití je pro specifikaci cílů a počátečních požadavků a řízení projektu.

- **Middle CASE** –
Tento druh CASE nástrojů je základem všech komerčně dodávaných nástrojů.

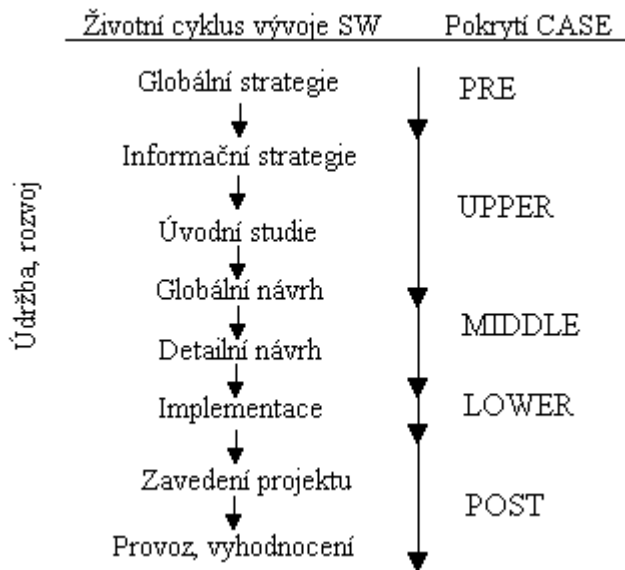
Prostředky této kategorie slouží pro podrobnou specifikaci požadavků analýzu a návrh systému. Používají se také pro dokumentaci a vizualizaci systému.

- **Lower CASE** –
Tyto nástroje slouží především jako podpora kódování, testování a údržby. Jejich součástí jsou i **nástroje forward engineeringu¹**, tedy generátory kódu z modelu (ty mohou generovat kostru nebo podstatnou část výsledného kódu, programátor poté doplňuje jen nutné detaily a algoritmy). Dále pak jde o prostředky pro **reverse engineering²**, které umožňují získat model z již existující aplikace, prostředky pro plánování a zjišťování kvality SW (sběr informací o testování, vyhodnocení testů, řízení testování), pro správu konfigurace, prostředky pro sledování a vyhodnocování práce systému. Funkce CASE nástrojů této kategorie se často překrývají s funkcemi obecných vývojových prostředí.
- **Post CASE** –
Tento druh CASE nástrojů podporuje organizační činnosti jako zavedení, údržbu a rozvoj IS.

Působnost takto rozdělených CASE nástrojů se překrývá, protože jimi podporované činnosti se mohou vyskytovat v různých fázích životního cyklu vývoje IS.

Fáze životního cyklu IS a CASE nástroje

Vztah CASE nástrojů a fází životního cyklu IS je zachycen na následujícím obrázku:



Pravdivé a mylné představy o CASE nástrojích

Pravdivé představy:

- Hlavním přínosem těchto nástrojů je vytváření úplných podkladů pro programování aplikací.
- CASE jsou nástroje, které mohou zlepšit produktivitu práce, efektivita práce vždy závisí na osobních kvalitách jednotlivých pracovníků.
 - Mohou generovat části kódu, ale nenahrazují programovací jazyky.
 - Praxe ukázala, že CASE nástroje často selhávají právě díky nedisciplinovanosti uživatelů.
- Automatizací „chaosu“ vznikne automatizovaný „chaos“.
 - Na počátku práce je nutné vykonat velmi mnoho činností, jejichž výsledek není dlouho vidět.
 - Dostanou-li stejný CASE dva systémoví analytici, dospějí k dvěma naprosto odlišným řešením.

Mylné představy:

- CASE nástroje slouží jako náhrada programovacích jazyků.
- Všechny CASE nástroje pracují podobně (poskytují stejné výstupy).
- Užívání CASE nástrojů zlepší práci manažerů organizace využívající výsledný produkt.
- CASE odstraňuje potřebu disciplíny a přísného vývoje aplikací IT.
- Od CASE nástrojů se často očekává jako výstup tvorba aplikačního programového vybavení.
- Produktivita dosažená pomocí CASE je okamžitě zřejmá.
- Užívání CASE zaručí konzistenci výstupů.

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>