

6. UŽIVATELSKÝ MODEL SYSTÉMU

Při návrhu implementačního modelu v Yourdonově pojetí zohledňujeme primárně implementační omezení z hlediska uživatele. Odtud název "uživatelský implementační model systému". V literatuře [Page-Jones] se fáze mezi tvorbou esenciálního modelu a strukturovaným designem označuje jako konfigurační analýza. Konfigurační analýza specifikuje, které implementační charakteristiky je třeba promítnout do esenciálního modelu. Technologický postup konfigurační analýzy zatím vytvořen nebyl. Toť pole neorané.

V dané situaci by bylo nejlepší předat esenciální funkční model (základní) designérům a programátorům, aby vybrali HW, OS, SŘBD, programovací jazyk apod. Ale ve skutečnosti existuje řada dalších omezujících informací.

V rámci konfigurační analýzy se provádějí jako základní tyto skupiny činností:

- sběr statistik o datech, procesech a externích entitách (terminátorech),
- určení hranic částí, které budou implementovány:
 - ručně,
 - dávkově,
 - interaktivně,
 - v režimu reálného času,
- výběr (či zhodnocení) hardware,
- vyhledání existujících programových balíčků,
- výběr stavebních materiálů pro výsledný systém.

Také nelze vše rozhodnout bez ohledu na názor a přání uživatele:

- co bude dělat počítačový systém a co se bude dělat ručně,

- formát uživatelského rozhraní (jak již jsme si řekli, je pro uživatele často důležitější než samotná funkce).

Navíc je třeba uvažovat:

- PC jako součást distribuované sítě, která část systému bude pod správou uživatele, která centrální, a to včetně dat,
- vlastní aplikace uživatelů (4GL, Spreadsheets ap.). Tyto aplikace musejí být rozumně integrovány do vytvářeného systému. Musíme proto řešit napojení jejich vstupů a výstupů. Rovněž je třeba počítat s koexistencí aplikací v 4GL a jazycích 3. generace,
- prototyp části systému (prototyp lze vytvořit v 4GL vysoké úrovně nebo aplikačním generátorem nebo prototypingem jako součástí CASE),
- v případě nakupovaného SW: co bude dělat uživatel, co koupený systém, formáty a pořadí vstupů apod.

Tyto všechny úvahy jsou součástí **implementačního modelu**.

6.1. Úprava esenciálního modelu do uživatelského implementačního modelu

Než začneme esenciální model upravovat, schvejme jej v jedné kopii pro novou implementaci v budoucnu - výchozí bod pro nový systém!

Uživatelský model pokrývá:

1. Rozdělení esenciálního modelu mezi lidi a počítač
2. Podrobný návrh uživatelského rozhraní (člověk-počítač)
3. Další ruční činnosti, které mohou být potřeba
4. Obslužná omezení, která si uživatel přeje do systému začlenit nebo která jsou nutná.

ad1) Rozdělení esenciálního modelu mezi lidi a počítač

Vyjmenujeme extrémní případy rozdělení esenciálního modelu mezi lidi a počítač. Nejčastější reálná dělba zahrnuje v implementačním modelu jak ruční, tak automatizované činnosti.

Extrémní případy:

- uživateli je to jedno (udělat odhad nákladů),
- uživatel chce plně automatizovaný systém (pouze se nahradí původní systém a hranice systému se nezmění),
- ruční činnosti jsou vně systému (skryty "v terminátorech", nemodelují se),
- uživatel chce plně ruční systém - dnes výjimka.

Při rozdělování systému na ruční a automatizované části nesmíme zapomenout na dvě, z hlediska uživatele velmi důležité, věci:

- popsat ruční postupy,
- rozhodnout o umístění vstupních a výstupních zařízení.

Do DFD je vhodné přidat funkce (bubliny) a data story nutné pro inicializaci a shutdown systému, zajišťující zprávy vedení, obsluhu, dále obslužné charakteristiky systému.

ad2) Podrobný návrh uživatelského rozhraní

Určení uživatelského rozhraní zahrnuje:

a) Výběr vstupních a výstupních zařízení

- MP, diskety, terminály a PC, OCR, telefon, fax, hlasitý vstup ...,
- tištěné výstupy, terminál, hlasitý výstup, plotter, MP + MD, mikrofilmy ...

b) Tvar všech vstupů z terminátorů do systému

Určit tvar vstupů znamená navrhnout obrazovky, formuláře a "Control Flow Diagrams" pro obrazovky.

Zásady: - konzistence vstupů a výstupů (př. datum na všech vstupech a výstupech stejně),

- požadovat vstupní a výstupní údaje v logickém pořadí (příjmení, jméno, titul, a ne jméno, titul, příjmení),
- domluvit chybové zprávy a ověřování chyb,
- rozlišovat mezi kontrolou na obrazovce (např. délka údaje) a kontrolou na existenci údaje v datové základně,
- uživateli srozumitelné zprávy (kombinace krátké zprávy s možnostmi vysvětlení),
- možnost zrušit rozpracovanou transakci nebo všechny provedené transakce, možnost návratů a oprav,
- pohodlný a komfortní "help",
- rozlišit "menu" a "příkazovou" komunikaci (výběr nechat na uživateli),
- dlouhé zpracování - zabavit uživatele,
- default hodnoty pro standardní vstupy,
- střídme použití barev a zvuků.

c) *Návrh formulářů*

- jednoduché - 1 papír, stačí domluvit s uživatelem
- složitější, speciální - raději "designér formulářů", který má zkušenost.

d) *Pořadí a časování vstupů a výstupů v on-line systému*

ad3) Další ruční činnosti

V esenciálním modelu se předpokládá, že se systém nikdy "nepoloží" a nikdy neudělá chybu.

Ve skutečnosti je část systému pod kontrolou uživatele (např. PC) a část se provádí ručně.

Možné oblasti chybné technologie a redundance

Nejčastěji chyby nastávají:

- při vstupu dat do systému - chyby lidí,
- při výpočtech - chyba programu, HW ...
- na médiích (vnějších pamětech) - HW, SW (DB programy),
- při výstupu dat.

Jak se proti chybám bráníme: redundantní vstupy a výstupy, redundantní procesorová technologie, vnitřní redundance (ověřování, kontrolní součty, kontrolní čísla ...).

ad4) Specifikace obslužných omezení

Specifikovat obslužná omezení znamená určit:

- hodnoty dat a jejich množství (systémové statistiky, frekvence přístupu, přístupové klíče ...),
- dobu odezvy - do kdy (určitá hodina, určitý den ...),
- strategická omezení a výběr implementace (definuje uživatel - např. značka HW, programovací jazyk, tele komunikační prostředky, "nesmí zabírat víc, než 10 MB na disku" ...),
- omezení prostředí: teplota, vlhkost, elektrické napětí, váha, velikost, vibrace, hluk, radiace,
- bezpečnost a spolehlivost.

Implementační model je vstupem do structured designu (automatizovaná část) a ostatních činností, které spadají do fáze vývoje technologického modelu systému.