

Úvod do problematiky tvorby tímového MUME projektu

Portál: edu.ukf.sk - Vzdelávací portál - Univerzita
Konštantína Filozofa, Nitra

Kurz: Projektovanie multimedialnych aplikácií

Kniha: Úvod do problematiky tvorby tímového MUME
projektu

Vytlačil(a): Zuzana Pavlendová

Dátum: Streda, 1 december 2021, 17:55

Opis

Úvod do problematiky tvorby tímového MUME projektu

Obsah

1 Úvodné predpoklady pre tvorbu MUME projektu

- 1.1 Konštrukcia
- 1.2 Samotná realizácia multimedialneho projektu – implementácia

2 Jadro - Špecifikácia MUME aplikácie a jej návrh na teoretickej úrovni

3 Záver - Testovanie

- 3.1 Spôsoby a fázy testovania MUME aplikácie
- 3.2 Príprava k zavedeniu – inštalácia

1 Úvodné predpoklady pre tvorbu MUME projektu

Skoro každý softvérový projekt je niečím jedinečný. Poznatky získané z jeho vývoja sa ťažko aplikujú na iné softvérové projekty. Napriek tomu sú určité faktory, ktoré pri tvorbe softvérového projektu zohrávajú kladnú alebo naopak zápornú úlohu.

Pred samotným riešením a realizáciou daného projektu je dôležitá prípravná fáza, v ktorej sa na základe zadania a témy projektu definujú ciele projektu, zohľadňujú požiadavky používateľa, pedagogické a psychologické požiadavky a aspekty a prebieha príprava materiálov, informačných zdrojov a podkladov k projektu.

1.1 Konštrukcia

Vonkajšia - dizajnová

Dizajn je závislý od predstavivosti a kreativity dizajnéra, prakticky mu nie sú kladené žiadne obmedzenia, ale musí samozrejme spĺňať vopred stanovené požiadavky používateľa (ak nejaké existujú), poprípade všeobecné či už pedagogické požiadavky, alebo pri najmenšom musí byť v súlade s cieľmi MUME aplikácie.

Vnútná - dátová

1. *dáta sú vkladané priamo v dizajnovom prevedení:* Toto riešenie je jednorazové a neumožňuje flexibilitu a rýchlu zmenu softvérovej aplikácie, pre flexibilné riešenia sa používajú iné dátové štruktúry.
2. *dáta oddelené v databáze s oddeleným dizajnovým prostredím:* týmto sa zabezpečuje dizajnová univerzálnosť a minimalizuje sa časová náročnosť pri prerábaní poprípade dopĺňaní aplikácie.

1.2 Samotná realizácia multimedialného projektu – implementácia

Na základe preddefinovaných podmienok a cieľov je teraz možné aplikovať tieto poznatky do reálnej situácie resp. ich zahrnutie do programu v ktorom sa bude vytvárať multimedialna aplikácie.

Programátor MUME aplikácie musí brať do úvahy náročnosť aplikácie tvorenej v MUME programe a jej požiadavky na systémové prostriedky.

Zosumarizovaním predchádzajúcich dvoch viet a všetkých dosiaľ absolvovaných etáp tvorby MUME aplikácie je ich zhrnutie a využitie v tzv. *Scenári tvorby softvéru*. Scenár je riadiacim elementom, podľa ktorého všetci členovia tímu vykonávajú činnosť na základe zodpovedností vyplývajúcich z ich pozície v samotnom projekte. Obsahuje *Časový plán* a detailný popis jednotlivých etáp tvorby, súvislosti a náväznosti, prípadne aj grafy a tabuľky, či iné podklady k vnútornej či vonkajšej štruktúre a prevedeniu.

Držaním sa scenára riešiteľa projektu postupne prechádzajú jednotlivými krokmi, či etapami a realizujú ich. Výsledok tejto činnosti je prvotná verzia softvéru, tzv. nultá verzia, ktorá býva označovaná aj ako prototyp. Následne sa celý tím dostáva do fázy testovania

2 Jadro - Špecifikácia MUME aplikácie a jej návrh na teoretickej úrovni

Po dokončení fázy štúdia realizovateľnosti sa tvorcovia MUME projektu dostávajú do fázy samotného hrubého návrhu danej aplikácie. Prvým krokom je teoretický návrh vnútornej a dátovej štruktúry a výber vhodného softvéru, či programovacieho jazyka na tvorbu danej aplikácie, tak aby spĺňala všetky doposiaľ uvádzané predpoklady a zároveň aby mala všetky vlastnosti, ktoré má plnohodnotná MUME aplikácia obsahovať. V tejto fáze je aj riešený návrh a tvorba grafického dizajnu a tvorba grafiky celkového prevedenia produktu, pričom je dôležité brať na ohľad hardvérové nároky a časovú náročnosť tvorby danej aplikácie a zároveň zohľadňovať i hardvérové nároky výsledného produktu.

Vlastnosti MUME aplikácie:

celkový vzhľad a grafický dizajn: prehľadné a príjemné používateľské prostredie, poskytujúce stanovenú funkcionality.

výkonnosť a spoľahlivosť: aplikácia musí byť optimalizovaná tak, aby výkonnostne stačila a funkčne bola schopná zvládať požadované zaťaženie, počet používateľov a zároveň si zachovávala kontinuitu činnosti.

efektívnosť: aplikácia by mala byť efektívna na všetkých svojich vrstvách a úrovniach, tak aby jej činnosť bola rýchla a bezchybná a práca s ňou bola pre používateľa príjemná, zbytočne nekomplikovaná a motivujúca.

bezpečnosť údajov: pozostáva z viacerých, vzájomne sa dopĺňajúcich oblastí ochrany údajov, ako napr.: problém poruchy hardvéru, monitorovanie konzistentnosti údajov a stavu systému, archivácia údajov alebo problému neautorizovaného prístupu.

bezpečnosť komunikácie: ak bude MUME aplikácia využívať pri svojej činnosti internet, LAN alebo WAN siete je potrebné zabezpečiť aj ochranu komunikácie a prípadne zvážiť podľa potreby i využitie šifrovaného kanála (SSL) využitím protokolu HTTPS.

3 Záver - Testovanie

Testovanie softvéru je súbor procesov slúžiacich na kontrolu kvality softvérového produktu, ktorých cieľom je dosiahnutie požadovanej kvality softvéru z hľadiska funkčnosti, spoľahlivosti, výkonnosti, použiteľnosti a podporovateľnosti. Kontrola kvality sa môže uskutočniť či už pre jednotlivé časti informačného systému alebo pre systém ako celok.

Testovanie výkonnosti softvéru je testovanie, ktoré sa vykonáva za účelom určenia, ako sa systém správa pod určitou záťažou. Testovanie výkonnosti môže slúžiť na viacero účelov. Môže preukázať, či softvér spĺňa výkonnostné požiadavky. Môže porovnať dva systémy a určiť, ktorý pracuje lepšie alebo preukázať, ktoré časti systému alebo pracovnej záťaže spôsobujú zlé správanie systému.

V diagnostických prípadoch je možné použiť nástroje, profilery, na zmeranie, ktoré časti zariadenia alebo softvéru sa najviac podieľajú na zlej výkonnosti alebo určiť úroveň priepustnosti systému a prahovú úroveň akceptovaného času odozvy. Pri testovaní výkonnosti je často rozhodujúce a súčasne ťažko dosiahnuteľné, aby boli testovacie podmienky podobné, ako predpokladané skutočné podmienky nasadenia.

3.1 Spôsoby a fázy testovania MUME aplikácie

Alfatesting: je testovanie prototypu aplikácie vývojovým tímom. Zistenia sú zaznamenávané a prípadné chyby ihneď riešené.

Betatesting: je testovanie už hotovej aplikácie pred predstavením koncovému používateľovi a vyznačuje sa týmito vlastnosťami:

- konštruktívny feedback,
- testovanie funkčnosť a bezchybnosti produktu,
- zaťažiteľnosť aplikácie v maximálnych prevádzkových požiadavkách,
- prvotný impulz pre prepracovanie jednotlivých vlastností aplikácie,
- potvrdenie spoľahlivosti a ovládania aplikácie a jej následnému vydaniu na trh.

Okrem betatestingu sa vykonávajú aj testy na testovacích vzorkách alebo skupinách ľudí, ktorí budú aplikáciu využívať.

Primárna testovacia skupina: skupina ľudí, ktorá bude modelom postaveným na základe profilu koncových používateľov. Pozorovania tejto skupiny budú pokladané za kľúčové pri možných úpravách a zmenách MUME aplikácie. Po doladení a nasledovnom opätovnom testovaní sa aplikácia otestuje sekundárnou testovacou skupinou.

Sekundárna testovacia skupina: vzorka koncových používateľov, pre ktorých bola aplikácia určená. Postrehy či prípadné chyby sú spätne prezentované vývojovému tímu, čím sa aplikácia dostáva do finálneho štádia – zavedenia aplikácie do praxe.

Testovanie môžeme definovať ako množinu procedúr a činností, ktorých úlohou je demonštrovať správnu činnosť MUME aplikácie v rámci jej zamýšľaného využitia. Účelom testovania je preukázanie existencie chýb, anomálií v správaní sa programu alebo aj demonštrovať, ktoré chyby v programe určite nie sú. Vo všeobecnosti však nemožno s istotou preukázať celkovú správnosť MUME aplikácie; možno len ukázať, že MUME aplikácia sa nespráva tak, ako sme čakali.

Existuje niekoľko postupov použiteľných pre testovanie MUME aplikácií. Jedným z najznámejších je tzv. *testovanie zdola nahor*, pri ktorom sa najprv otestujú základné prvky, potom ich nadradené prvky ... atď. V praxi tvorby MUME aplikácií však možno použiť aj opačný postup - *testovanie zhora nadol*. Toto sa uplatňuje priebežne už v čase tvorby MUME aplikácie tak, že istá hierarchická úroveň riešenia sa testuje s prázdnyimi (zatiaľ nenaplnenými) podúrovňami. Po úspešnom otestovaní sa aplikácia "zjemní" o ďalšiu úroveň a postup sa opakuje. Ak test skončí konštatovaním, že nepreukázal existenciu chýb, aplikácia sa doplní príslušnou dokumentáciou. Je pripravená pre zákazníka.

3.2 Príprava k zavedeniu – inštalácia

Po úspešnom ukončení testovacej fázy projektový tím vytvára a ladí inštalačný balík danej MUME aplikácie, tvorí podporné systémy aplikácie – helpy, informačné registre, slovné registre alebo slovníky. Dôležitou súčasťou MUME aplikácie je softvérová dokumentácia.

Softvérová dokumentácia

Táto na prvý pohľad banálna záležitosť však v sebe skrýva viac problémov, ako by sa na prvý pohľad zdalo. Softvér, ktorý neobsahuje kvalitnú obsažnú a vyčerpávajúcu dokumentáciu je už dopredu odsúdený na zánik. Základným faktorom, ktorý je treba mať na zreteli, je cieľová skupina. Odlišný typ informácií je zaujímavý pre bežného používateľa softvéru, iný pre vývojára aplikácie. Tomu treba následne prispôbiť formu aj obsah dokumentácie. Preto odlišujeme dva rôzne typy dokumentácie:

Vývojárska dokumentácia

Príslušný vývojár pracujúci na projekte potrebuje mať zdokumentované všetky vnútorné technické záležitosti softvéru, ako sú napr. hierarchia dedičnosti tried, popis dátových štruktúr a ich atribútov, zoznam zdrojových súborov, zoznam globálnych identifikátorov a pod. Medzi informácie takéhoto charakteru patrí tiež opis správneho použitia príslušných vývojových nástrojov, slúžiacich k zostavovaniu programu. Základným účelom existencie vývojárskej dokumentácie je údržba a rozšíriteľnosť softvérového produktu. Nutné je zdokumentovať najmä všetky neštandardné programovacie konštrukcie a taktiež podrobný popis postupu ladenia programu.

Používateľská dokumentácia

Bežného používateľa spravidla nezaujíma štruktúra softvéru ani jeho vnútorná implementácia. Nie je preňho dôležité ako sú veci urobené, aké prostriedky a programovacie jazyky boli pri vývoji použité, ale ako v skutočnosti program funguje, ako sa používa a taktiež v akom prostredí a pod akým operačným systémom je spustiteľný. Preto by sa mal dôraz klásť na podrobné, ale hlavne zrozumiteľné vysvetlenie všetkých funkcií a možností programu. Je lepšie, keď je používateľská dokumentácia písaná niekým iným ako samotným autorom programu. Ten totiž považuje mnoho vecí za úplne jasné a samozrejmé, ktoré však už pre bežného používateľa také jasné a samozrejmé nie sú. Vhodné je, keď používateľská dokumentácia obsahuje množstvo príkladov a keď je celkovo písaná s vedomím, že ju nebude citáť osoba rozumejúca programovaniu.

Z prechádzajúcich definícií je teda zrejmé, že tieto dva typy dokumentácie sú úplne odlišné. Nie je možné, aby dokumentácia prvého typu nahrádzala druhú alebo opačne. U vývojárskej dokumentácie platí pravidlo písať ju priamo za behu vytvárania programu. Ak sa vnútorná implementácia alebo štruktúra zmení, zmení sa aj príslušná časť dokumentácie. Používateľskú dokumentáciu je vhodné taktiež začať písať čo najskôr, ale až po dosiahnutí aspoň základnej funkcionality programu. Oba typy dokumentácie píše iní ľudia a tomu sú aj prispôbené nástroje na ich tvorbu a údržbu. Ich použitie bude opísané v nasledujúcich kapitolách tohto dokumentu.

V prípade, že MUME aplikácia je úzko špecifická alebo veľmi komplexná je pre tím tvorcov nevyhnutné vyškolenie a zaučenie koncových používateľov, alebo len ich vedúcich pracovníkov, ktorí budú školiť cieľové auditórium.

Zavedenie mume aplikácie

K samostatnému zavedeniu MUME aplikácie predchádzajú dve hlavné fázy. Inštalačná fáza a fáza skúšobnej prevádzky.

Pod pojmom inštalačná fáza rozumieme samostatnú inštaláciu danej aplikácie u používateľa, nastavenie hardvérových prostriedkov a pripojenie prídavných zariadení ktoré zabezpečuje tím technikov. Pokiaľ projekt využíva internetové technológie prebieha spúšťanie serverov a databázových systémov a zabezpečuje sa bezpečnosť ich prevádzky.

Fáze skúšobnej prevádzky zodpovedá čas určený vývojármi a prevádzkovateľom aplikácie po vzájomnej dohode. Tento čas je využitý na testovanie stability a na prípadné jemné doladenie aplikácie. Po uplynutí tohto času je aplikácia kvalifikovaná ako plne funkčná.