

PRENOSITEĽNOSŤ SOFTVÉRU PRE MOBILY A JEJ BUDÚCNOSŤ

Kamkoľvek sa dostanem, fungujem správne.

Jozef Macho

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
jozko.macho[zavináč]gmail[.]com

Abstrakt. *Vlastnosti softvérového produktu predstavujú základ pre proces vytvárania softvéru. Pri vývoji softvéru sa musí brať ohľad na každú vlastnosť softvéru, pretože výsledná kvalita softvéru sa odvíja práve od miery splnenia požadovaných vlastností. Jednou z dôležitých vlastností softvéru je aj jeho prenositeľnosť, teda schopnosť fungovať na viacerých zariadeniach(platformách). Táto vlastnosť je kľúčovou pre schopnosť softvéru presadiť sa na trhu so softvérom v čo najširšom možnom okruhu používateľov. Špeciálnou skupinou softvéru sa stal softvér pre mobilné zariadenia, ktorý v dnešnej dobe dosahuje obrovský rozmach a jeho vývoj napreduje míľovými krokmi. V tejto eseji sa zaoberám prenositeľnosťou softvéru určeného pre mobilné zariadenia. Zameriavam sa na vplyv prenositeľnosti na vývoj softvéru pri práci v tíme a na dôležitosť tejto vlastnosti pri vývoji mobilných aplikácií. V závere eseje sa pokúšam odhadnúť ako sa bude prenositeľnosť vo vývoji softvéru pre mobilné aplikácie meniť a či sa vôbec niečo oproti súčasnosti zmení.*

Kľúčové slová: *prenositeľnosť, vlastnosti softvéru, mobilné zariadenia, software portability*

Úvod

V každodennom živote sa stretávame s problémom výberu rôznych produktov, ktoré majú slúžiť našim potrebám. Ide o produkty, ktoré si vyberáme podľa ich určitej kvality alebo podľa prostriedkov, ktoré sme ochotní a vo väčšom prípade najmä schopní vynaložiť pre získanie danej veci. To platí pre najbežnejšie produkty dennej spotreby ako sú potraviny,

cez rôzne domáce potreby, pomôcky, šaty, až po techniku. Tieto produkty majú svoje vlastnosti, ktorými sa pre nás stávajú zaujímavými, pričom vlastnosti sú aj príčinou, prečo sa pre daný produkt rozhodneme alebo nerozhodneme. Vlastnosti produktov však nie sú dôležité len pre ich používateľom(konzumentov), ale aj pre samotných výrobcov týchto produktov. Pokiaľ by totiž produkt nebol dostatočne atraktívny pre trh, producenti by nedosahovali očakávané zisky.

Jednou z takýchto vlastností je aj schopnosť daného produktu prispôbiť sa, resp. dokázať fungovať v rozličných prostrediach bez toho, aby bolo potrebné výraznejšie zasiahnuť do jeho štruktúry. Ako príklad takéhoto produktu by sa dali uviesť napríklad slúchadlá, pričom väčšina slúchadiel sa dá pripojiť k notebooku, televízoru či k rádiu a svoju úlohu plnia bez toho, aby sme potrebovali meniť ich koncovku.

Podobne ako pri spomínaných produktoch, aj v oblasti vývoja a predaja softvéru zohrávajú vlastnosti softvéru veľmi dôležitú úlohu. Pomocou týchto vlastností vieme softvér do určitej miery ohodnotiť a získať obraz o jeho kvalite. Vlastnosť prenositeľnosti softvéru zohráva kľúčovú úlohu práve pri pokuse o získanie záujmu veľkej časti trhu. Vyžaduje sa zaistenie správnej a jednoduchšej implementácie softvéru v niekoľkých prostrediach. Toto platí ešte väčšmi v prostredí, v ktorom nedominuje jeden softvérový výrobca, ale kde cieľový trh je rozmanitý. Takýmto trhom sa dá nazvať hlavne trh s mobilnými zariadeniami.

Prenositeľnosť vo všeobecnosti a jej vplyv na vývoj softvéru

V dnešnej dobe je prenositeľnosť softvéru považovaná za veľmi atraktívnu z pohľadu kvality softvéru najmä kvôli dostupnosti rozmanitých platforiem v oblasti softvéru. Podľa [3] sa však v rámci vývoja softvéru príliš nevynakladajú mimoriadne prostriedky a plánovanie na dosiahnutie prenositeľnosti. Podľa [3] je najväčšou príčinou tohto stavu nedostatok zdokumentovaných stratégií na riešenie prenositeľnosti a problémov spojených s jej riešením. Myslím si, že daný fakt je kritický pre zohľadnenie prenositeľnosti v plánovaní vývoja softvérového produktu. Toto núti tím pracujúci na projekte k iba určitému odhadu či domnienke, ako by mali svoj softvér navrhovať a implementovať, aby dosiahli požadovaný stupeň prenositeľnosti. Spôsobuje to, že projektoví manažéri skepticky pristupujú k riešeniu prenositeľnosti, pretože nemajú istotu, že sa oplatí zamerať sa na tento aspekt kvality a rezervovať mu osobitné prostriedky. Nie je totiž nijako zaručené, že sa im dané prostriedky vynaložené na oblasť prenositeľnosti vrátia, resp. že použitá stratégia bude mať žiadaný efekt a dosiahne sa požadovaný výsledok.

Ako sa uvádza v [1] prenositeľnosť softvéru sa dá charakterizovať nasledujúcou definíciou:

Softvérová jednotka je prenositeľná skrz určitú triedu prostredí do takého stavu, kým je cena jej prenosu a adaptácie na nové prostredie v danej triede menšia, ako je cena za jej prerobenie(vývoj pre toto prostredie).

Za softvérovú jednotku môžeme považovať aplikáciu, program pre systém alebo časť programu. V našom prípade ide hlavne o aplikácie, keďže sa táto esej zaoberá hlavne softvérom mobilných zariadení, teda kde sa mobilná aplikácia prenesie. Za prostredie môžeme považovať všetko, s čím softvérová jednotka spolupracuje, teda najmä nastavenie hardvéru, operačný systém či sieť. Iné pomenovanie pre spomínané je aj platforma.

Dôležitý je i termín *trieda*, pretože zriedka je potrebné zahrnúť všetky prostredia, na ktoré by sa daný produkt dal preniesť.

Pre vývoj softvérového produktu je z hľadiska prenositeľnosti dôležité, aby sme boli schopný odhadnúť, aké veľké úsilie by malo byť vynaložené na vytvorenie návrhu programu, ktorý môže byť v budúcnosti jednoducho nastavený na inú platformu. Je potrebné zanalyzovať náklady a zisky, aby bolo zistený čo najpresnejší stupeň prenositeľnosti aký by mohol byť dosiahnutý predtým, ako náklady na vytvorenie návrhu prevýšia zisky. K tomuto nám podľa [2] poslúžia otázky:

1. Pre akú triedu prostredí by mala byť budúca prenositeľnosť uvažovaná?
2. Aký stupeň prenositeľnosti je vyžadovaný pre rozličné prostredia v týchto triedach?
3. Aké mimoriadne náklady na vývoj(ak nejaké sú) sú akceptovateľné na dosiahnutie týchto cieľov prenositeľnosti?
4. Aké zníženie v kvalite implementácie(ak nejaké je) je akceptovateľné na dosiahnutie vyžadovanej prenositeľnosti?

Po zodpovedaní týchto štyroch otázok by mal byť návrhár schopný vybrať metodológiu, ktorá by mohla pomôcť vytvoriť požadovanú prenositeľnosť a byť pritom stále efektívny vzhľadom na náklady. Prvá otázka odpovedá na ohraničenia prenositeľnosti, teda ohraničením na koľko cieľových platforiem by malo byť možné program nasadiť. Druhá otázka poskytuje pohľad na zisky vyplývajúce z prenositeľnosti a posledné dve otázky naopak zahŕňajú náklady spojené s prenositeľnosťou. S týmito nákladmi je úzko spojený návrh architektúry, keďže prenositeľnosť ovplyvňuje aj iné vlastnosti softvérového produktu. Ide hlavne o spoľahlivosť a výkonnosť („performance“) softvérového produktu. Podľa môjho názoru je tento vplyv v kombinácii so súvisiacimi nákladmi potrebnými na vývoj softvéru dôvodom toho, že členovia vývojového tímu nevenujú prenositeľnosti až toľko úsilia, ako by bolo niekedy potrebné. Myslím si, že pokiaľ sa však vývoj od začiatku nezaobera v dostatočnej miere prenositeľnosťou, je silný predpoklad, že v neskorších fázach vývoja to bude mať vývojový tím ešte náročnejšie, aby zakomponoval aspekt prenositeľnosti do produktu v požadovanej miere. Takisto sa domnievam, že pri pracovaní s hotovým produktom bude oveľa nákladnejšie spraviť vývoj pre novú platformu, ktorá v čase vývoja softvéru nebola dostupná a v tomto prípade bude potrebné spraviť celý vývoj odznova pre túto platformu(prerobenie produktu na danú platformu) miesto dorobenie len určitých komponentov.

Ako píše [3], kľúčovou úlohou pre dosiahnutie požadovanej prenositeľnosti je identifikovanie externých rozhraní, ako je napríklad práca so súbormi, správa pamäte, vstupno-výstupné zariadenia či používateľské rozhrania, a ich štandardizácia, prípadne izolácia. Pokiaľ sa takýto komponent štandardizovať nedá, je na zvolenej architektúre, aby rozhodla ako prenositeľný bude a to tým, ako je ho možné izolovať od zvyšnej programovej funkcionality. Tieto komponenty sú práve tie, ktoré sa líšia medzi platformami a je prirodzené izolovať ich pre dosiahnutie vyššieho stupňa prenositeľnosti.

Prenositelnosť mobilných zariadení

Vzhľadom na trh s mobilmi, prenositeľnosť je považovaná za väčšiu výzvu ako v iných oblastiach. Podľa môjho názoru sa často dostáva aj na prvé miesto v riešení problémov ohľadom vlastností softvérových produktov vyvíjaných pre mobilné zariadenia. Táto skutočnosť má podľa mňa korene v obrovskom množstve rozličných mobilných zariadení, ktoré sú v tejto dobe na trhu, pričom nejde len o značky jednotlivých mobilných zariadení, ale aj o ich rozmanitosť v rámci jednej značky. Ďalším dôvodom je podľa [4] aj dôležitosť výkonnosti softvéru pri vývoji pre mobilné zariadenia, ako jedna z hlavných priorít vlastností mobilov. Ako bolo spomenuté v predošlej kapitole, výkonnosť s prenositeľnosťou úzko súvisia. Prenositeľnosť softvéru však v tejto oblasti nemožno vôbec zanedbávať, keďže v dnešnej dobe existuje množstvo mobilných zariadení, ktoré podporujú spúšťanie „vlastných“ aplikácií a cena vývoja softvéru stále rastie. Preto zostáva prenositeľnosť softvérového produktu v tejto oblasti mimoriadne žiadanou.

Problém veľkého počtu rozmanitých mobilných zariadení je dôvodom, prečo je prenositeľnosť u týchto zariadení dosahovaná ťažšie ako napríklad pri softvéri pre PC. Je to spôsobené hlavne tým, že neexistuje štandard pre veci ako rozlíšenie, rozloženie klávesnice či veľkosť pamäte. S vývojom mobilných zariadení ich rozličnosť ešte rastie, keďže pribúdajú rôzne nové vstupné zariadenia ako virtuálna klávesnica či pero.

Dôležitosť výkonnosti softvéru ako faktor ovplyvňujúci prenositeľnosť softvérového produktu je problém, ktorý sa u týchto zariadení vždy objavoval. Mobil má totižto väčšinou dosť limitované prostriedky, a preto sú vlastnosti ako optimalizovaná správa pamäte a využitie procesora veľmi podstatné a vyžadované. Toto je požadované hlavne kvôli použitiu mobilných zariadení pre rýchly prístup k tomu, po čom používateľ túži. Z tohto dôvodu je rozhodovanie medzi zvyšovaním priority prenositeľnosti a výkonnosti softvérového produktu pri vývoji obzvlášť náročná.

Spomínané dva dôvody zohrávajú hlavnú úlohu pri výbere programovacieho jazyka pre vývoj mobilnej aplikácie. V [5] sú dva príklady najviac používaných programovacích jazykov pre vývoj mobilných aplikácií a to programovanie jazykom Java a C(a C++). Argumentujú, že hlavnou výhodou jazyka Java je práve prenositeľnosť, pričom naopak programovanie v C podporuje viac výkonnosť. Z tohto pohľadu teda usudzujem, že vývojový tím má už v začiatkoch práce ťažkú úlohu hlavne vo výbere programovacieho jazyka. Kým výber napríklad jazyka Java by má dopad na výkonnosť, no podporuje prenositeľnosť, výber jazyka C uprednostňuje výkonnosť. Z hľadiska prenositeľnosti softvéru je teda lepšie zvoliť pre programovanie softvérového produktu jazyk Java. Samozrejme nie je to žiadnou dogmou, no pre zlepšenie prenositeľnosti softvéru je to určite opodstatnené riešenie. Pre takéto riešenia sa v dnešnej dobe používa J2ME(Java Micro Edition), ktorá je dnes najpoužívanejšou platformou vo vývoji pre mobilné zariadenia. Je veľmi zaujímavou hlavne z dôvodu, že umožňuje používateľskému rozhraniu prispôbiť sa, keď sa zmení kontext či rozlíšenie bez toho, aby sa zvýšila komplexnosť kódu.

Ako nezanedbať prenositeľnosť a čo to prinesie

Vývoj softvéru je časovo náročný proces, no ešte náročnejším aspektom vývoja je jeho cena. V dnešnej dobe je tendencia časovú a finančnú stránku softvéru minimalizovať, teda je snaha vytvárať softvér lacno a rýchlo. Toto však nanešťastie príliš nepodporuje prenositeľnosť, keďže robenie softvéru čo najviac prenositeľným, pridáva náklady na začiatku projektu, pričom toto sa prejaví v znížení nákladov vývoja, no až v oveľa neskorších fázach životného cyklu vývoja softvéru. V [3] sa uvádzajú tri dôležité body dosiahnutia prenositeľnosti:

1. *Kontrola nad rozhraniami* – tento bod je veľmi dôležitý, keďže rozhrania sa často veľmi líšia v závislosti od platforiem. Pričom najväčším problémom býva komunikácia medzi rozhraniami a funkcionalitou. Najčastejšie sa to týka hlavne používateľských rozhraní, ktoré sa môžu veľmi líšiť v závislosti od prostredí.
2. *Izolovanie závislostí* – komponenty programu, ktoré budú potrebovať veľa zmien pri pracovaní na rôznych platformách potrebujú byť oddelené. Príkladom takýchto komponentov môže byť práca so súborami alebo správa zariadení(médií).
3. *Myslieť „prenositeľne“* – každý, kto je zahrnutý do návrhu a vývoja musí myslieť na prenositeľnosť v každom momente a nespoliehať sa na to, že sa táto vec bude riešiť až vo fáze implementácie. Prenositeľnosť by mala byť riešená už vo fáze vymenúvania požiadaviek a postupne ďalej cez návrh, implementáciu a testovanie.

Z týchto bodov(hlavne z 3.) je jasné, že zaoberať sa prenositeľnosťou vyvíjaného softvéru je dôležité počas celého životného cyklu vývoja softvéru. Myslím si, že čím neskôr sa touto otázkou začne vývojový tím zaoberať, tým väčšie náklady(či už časové alebo finančné) bude potrebovať na vyriešenie tejto otázky v neskorších fázach vývoja. Pokiaľ ide o softvér pre mobilné zariadenia, bude prenositeľnosť softvéru vždy mimoriadne dôležitou(ak nie najdôležitejšou) vlastnosťou sledovanou pri vývoji takéhoto softvéru. Technológie totiž napredujú obrovskými krokmi a ako to bolo už pri osobných počítačoch, aj v mobilných zariadeniach výkonnosť a kapacita jednotlivých súčiastok pracujúcich so softvérom rastie. Rastú však aj náklady vyžadované na vývoj softvéru. Preto si myslím, že v budúcnosti sa bude vývoj softvéru pre mobilné zariadenia oveľa viac venovať prenositeľnosti softvéru ako výkonnosti. Veľkou prekážkou v tomto by však mohla byť ešte väčšia rôznorodosť mobilných zariadení. Vývoj tejto skutočnosti je však ťažké odhadnúť. V každom prípade sa však prikláňam k názoru, že investícia do prenositeľnosti je nutná a je potrebné venovať sa jej už na začiatku vývoja softvéru, pretože takýto prístup môže vývojovému tímu ušetriť nemalé množstvo prostriedkov počas ďalšej práce na softvérovom produkte.

Záver

Cieľom tejto eseje bolo zanalyzovať vlastnosť prenositeľnosti softvérového produktu a jej vplyv na vývoj softvéru. Preskúmal som problematiku prenositeľnosti softvéru pri vývoji vo všeobecnosti i konkrétne pre mobilné zariadenia. Dospel som k záveru, že dosiahnuť prenositeľnosť pri vývoji softvéru pre trh s mobilnými zariadeniami je oveľa náročnejšou úlohou, ako je tomu napríklad pri trhu s osobnými počítačmi. Je to spôsobené hlavne

väčšou diverziou mobilných zariadení, pričom softvér musí byť konkurencie schopný. Náročnosť v tejto oblasti je daná hlavne tým, že je tu neustály súboj medzi výkonnosťou a prenositeľnosťou, pričom pri vývoji je uprednostňovaná jedna vlastnosť na úkor druhej.

Pri zvolení cesty vysokej prenositeľnosti však nie je zaručený istý úspech. Je potrebné zodpovedne pristupovať k vývoju a to už zo začiatku životného cyklu softvéru. Investovanie nákladov do prenositeľnosti softvéru na začiatku vývoja sa totiž vráti v neskorších fázach cyklu vývoja. Platí to však i naopak, teda zanedbanie prenositeľnosti v akejkoľvek fáze vývoja softvéru sa premietne do neskorších problémoch pri snahe zakomponovať ju do softvérového produktu.

Pre budúcnosť odhadujem, že sa vývoj softvérových produktov pre trh s mobilnými zariadeniami bude viac orientovať na prenositeľnosť softvéru ako na výkonnosť, keďže sa technologické možnosti stále posúvajú dopredu a podobne ako pri osobných počítačoch už nebude vývoj softvéru až tak limitovaný náročnosťou na výkonnosť.

Použitá literatúra

1. Mooney, J. D. (1990). Strategies for Supporting Application Portability. IEEE Computer, 11(23), 59-70.
2. Mooney, J. D. (1995). Portability and Reusability: Common Issues and Differences. Proceedings of the 1995 ACM 23rd annual conference on Computer science, 150-156.
3. Mooney, J. D. (1997). Bringing Portability to the Software Process. Dept. of Statistics and Comp. Sci., West Virginia Univ., Morgantown WV.
4. Johansson, A., Svensson, J. (2009): Techniques for Software portability in Mobile Development, School of Engineering, Blekinge Institute of Technology, Ronneby.

Annotation

Software portability in mobile devices and its future

Software product metrics constitute basis for the process of software creation. It is important to consider every software metric in software development, because the final quality of software is given by achieving the measure of desired metrics. One of the most important software metric is also its portability, the ability to operate on several devices(platforms). This metric is the key factor of software ability to reach the largest part of software market. Software for mobile devices became a special set of software, which nowadays reaches big expansion and its development aheads rapidly. In this essay, I deal with software portability destined for mobile devices. I survey on effect of portability to team software development and on importance of this metric in mobile applications development. In the conclusion of the essay, I try to forecast the future changes of software portability for mobile applications or whether if something will ever change.