

# Riziká vývoja softvéru

PETER ARGALÁŠ

*Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava*

**Abstrakt.** Softvérový priemysel je jedným z najrýchlejšie sa rozvíjajúcich odvetví. S týmto rozvojom je spojené aj vysoké riziko neúspechu, ak proces vývoja softvéru nie je riadený a nesústreďuje sa na skutočné riziká, ktoré ho ohrozujú. V tejto eseji sa budem venovať základnému zdroju neúspechu, ktorý vychádza z mylnej predstavy, že softvér je iba produkt. Ďalej sa budem venovať softvéru chápanému ako médium na uchovávaní vedomostí, pričom najprv sa pozriem, aké je postavenie softvéru v rámci ostatných médií na uchovávanie vedomostí a neskôr uvediem, aký vplyv má tento postoj na proces vývoja softvéru. V druhej časti eseje budem rozoberať faktory, ktoré podľa výskumu v rôznych organizáciách najviac ohrozujú tento proces.

## Úvod

V dnešnej dobe je vývoj softvéru zložitý proces, kde je potrebné koordinovať veľké množstvo činností. Na softvérových projektoch sa zúčastňuje množstvo ľudí, pričom ich zameranie nemusí byť vždy rovnaké.

Doba kedy softvér vyvíjali jednotlivci, je už nenávratne preč. Ak chcem, aby sme boli úspešní, je nutné vyvíjať softvér v skupinách, kde má každý presne určenú svoju pozíciu a svoje úlohy. Jednou z týchto jednotiek, ktoré tvoria skupinu, je manažment. Ľudia v tejto jednotke sa sústreďujú na riadenie, plánovanie, organizovanie a vedenie ostatných jednotiek. Okrem týchto činností resp. v rámci týchto činností je manažment zodpovedný aj za odhadovanie rizík, ktoré môžu ohroziť proces vývoja softvéru, v krajnom prípade aj viesť k jeho neúspešnému ukončeniu. V rôznej literatúre sa môžeme dočítať, aké sú tieto riziká a akým spôsobom ich môžeme eliminovať. Aj ja sa budem neskôr v tejto eseji venovať niektorým z nich. Najprv by som sa však rád pristavil pri jednej myšlienke, ktorá bola načrtnutá v článku [1] z ktorého vychádzam.

## Softvér ako médium

Väčšina dnešných softvérových firiem má prepracovanú organizáciu, používajú vyspelé metódy vývoja softvéru, počas procesu vývoja sa snažia identifikovať a odstraňovať riziká. Aj napriek tomu mnohé projekty končia neúspechom.

Hlavná príčinou toho, prečo sa tak deje, vychádza zo samotného prístupu a chápania softvéru. Spoločnosti často vychádzajú z predstavy, že softvér je produkt a tomu prispôsobujú aj svoje správanie. Keď sa pozrieme na softvér len povrchno a z externej perspektívy, vynára sa nám otázka. *Čo iné ako produkt by to mohlo byť?* Ved' predsa softvér má všetko čo má mať klasický produkt. Treba ho navrhnuť, implementovať, otestovať, zabaliť, expedovať a predať. Áno, z tohto hľadiska sa softvér ničím nelíši od produktu. Ak sa však pozrieme na softvér detailnejšie, zistíme, že to nie je celkom tak. Zistíme, že softvér je skôr médium [2], ktoré v sebe obsahuje myšlienky a vedomosti svojich tvorcov, zapísane vo forme programu.

Počas vývoja sveta ako takého sa dá rozlišovať päť etáp, alebo päť období, z ktorých každé je charakteristické vznikom nového média na uchovávanie vedomostí. Skôr ako budem pokračovať v opise jednotlivých médií, bolo by dobré chvíľu sa venovať charakteristikám, ktoré popisujú médium na uchovávanie vedomostí. Jednotlivé média sa od seba odlišujú mierou schopnosti naplniť niektorú z charakteristík. Charakteristiky sú nasledovné:

1. *úroveň perzistencie* – doba zotrvania vedomosti v médiu
2. *rýchlosť zmeny* – ako rýchlo môže byť vedomosť zmenená
3. *zámernosť* – miera zámernosti uloženia a zmeny vedomosti
4. *pôsobnosť* – schopnosť ovplyvňovať vonkajší svet

Softvér je najmladším z piatich médií, ktoré slúžia na uchovanie vedomostí.

## DNA

Historicky prvým médiom, ktoré v sebe uchováva informáciu je *DNA*. Vedomosť, alebo lepšie povedané informácia, ktorú v svojej štruktúre DNA uchováva, predurčuje budúcu tvár života a ukrýva v sebe návod, ako vytvoriť život. Informácia zakódovaná v DNA je stála, aj keď sa dokáže meniť tieto zmeny sú z nášho pohľadu relatívne pomalé. Čo sa týka miery zámernosti, je v prípade DNA nízka, keďže naša možnosť zmeniť záznam DNA je obmedzená. Vplyv vedomostí na okolité prostredie je nepriamy a dá sa realizovať cez objekt, ktorý bol stvorený na základe tejto informácie.

## Rozum

Po DNA sa zhruba pred dva a pol miliónom rokov objavilo druhé médium. Mysel', vedomie alebo *rozum* boli dané našim predkom, aby mohli uchovávať, meniť, pridávať a uberať rôzne informácie. Čo je ale dôležitejšie, ľudia mohli pri používaní rozumu vytvárať nové vedomosti, čo ich odlišovalo od zvierat, ktoré mali tiež rozum, ale ich konanie bolo iba pudové. Uložené vedomosti sú dosť nestále, čo im dodáva možnosť rýchlej zmeny. Sú vo všeobecnosti zámerné a ovplyvňujú okolie cez naše telo.

## Nástroje

Ľudia používali svoj rozum a časom začali vyrábať jednoduché *nástroje*. Čiže človek použil vedomosti, ktoré mal uložené a na základe nich vytvoril nástroje, ktoré mu slúžili. Tým sa tieto nástroje stali tretím médiom na uchovanie vedomostí.

Zoberme si napríklad taký kamenný pazúrik, ktorý slúžil na rezanie. Pazúrik aj keď bol opracovaný, stále to bol iba kameň. To, čo ho v skutočnosti odlišovalo od ostatných kameňov, bolo práve to, že bol systematicky opracovaný, aby slúžil na nejaký účel. Tým, že ho tvorca vytvoril, vložil do neho informáciu, vedomosti človeka o tom, ako opracovať kameň, aby z neho vznikol nástroj. Vedomosti uložené v nástrojoch sú samozrejme stále, na druhej strane ale prakticky nemeniteľné, alebo ich zmena vyžaduje veľké úsilie. Sú zámerné a boli vytvorené na to aby ovplyvňovali okolitý svet.

## Knihy

V poradí štvrté médium sa objavilo okolo roku 7000 p.n.l. v starobylej Číne, ale jeho plný rozkvet zaznamenávame až od polovice 15. storočia, kedy bola vynájdená kníhtlač. Teraz je každému jasné, že reč je o *knihách* ako médiu na uchovanie vedomostí. Hovoriť o tom, že kniha je naozaj médiom, by bolo zrejme zbytočné. Vedomosti v knihách sa udržujú dlhú dobu, tým pádom možnosť ich zmeny je obmedzená. Nevýhoda kníh je v tom, že priamo nemôžu ovplyvniť vonkajší svet.

## Softvér

Cez všetky tieto médiá sme sa konečne dostali, k najmladšiemu z nich, ktorého výskyt sa datuje do polovice minulého storočia. Popri ostatných médiách softvér zaznamenáva dnes svoju renesanciu, na každom kroku môžeme pozorovať jeho rozkvet.

Jeho dôležitosť sa dá demonštrovať skutočnosťou, že množstvo ľudí je zamestnaných iba tým, aby extrahovali vedomosti z iných médií a transformovali ich tak, aby sa dal uložiť práve do softvéru. To, že softvér je naozaj iba médiom na uchovanie vedomostí sa dá demonštrovať na nasledovnom príklade. Hodnota nástroja je v tom s akým umením bolo vytvorené, hodnota knihy spočíva v jej obsahu a nie v papieri alebo atramente, ktorým bolo napísaná, tak aj hodnota softvéru nie je v samotnom kóde, ale v tom aké vedomosti sú v tomto kóde. Nikoho zrejme prvotne nezaujíma, ako je kód naformátovaný, alebo aký je veľký, ale to čo vlastne kód robí a ako to robí. Z pohľadu charakteristík je softvér najlepšie médium. Vedomosti v ňom sú stále, napriek tomu sa dajú ľahko meniť, sú zámerné a samozrejme výdatne vplývajú na okolie.

Softvér má ešte jednu výhodu oproti svojim predchodcom. Vedomosti v ňom uložené môžu byť použiteľné a vykonateľné prakticky kýmkoľvek. Vedomosti, ktoré mám v hlave pomáhajú iba mne a ja ich môžem ľahko zabudnúť, naproti tomu kniha dokáže uchovať znalosti na dlhý čas ale to je všetko čo dokáže, ale znalosti v softvéri sa dajú použiť, dajú sa vykonávať a tým pomáhajú mnohým.

Tu sa môže vynoriť otázka, prečo tu opisujem iné médiá, ktoré uchovávajú znalosti, keď by som sa mal venovať iba softvéru. Dôvod prečo je to tak, je aby sme si lepšie uvedomili postavenie a výhody softvéru ako najdokonalejšieho média na uchovávanie znalostí. V tejto myšlienke a v jej prijatí je zakotvený základný zdroj úspechu, alebo neúspechu projektu. Je preto dôležité uvedomiť si, že softvér je naozaj iba médiom a ozajstný produkt predstavujú myšlienky, vedomosti a znalosti, ktoré autor vkladá do tohto média. Ak však vychádzame zo zažitej predstavy, že softvér je produkt, tak potom sa nám môže stať, že vytvoríme síce funkčný softvér, ale ten nebude obsahovať relevantné znalosti. Ak by aj takýto softvér obsahoval vedomosti, ktoré mal obsahovať, tak tým, že sme sa pri procese tvorby nesústredili na ich systematické zbieranie a zabudovanie do softvéru, môžeme naraziť na problém, ak budeme chcieť znovu získať tieto vedomosti z kódu.

Z tohoto prístupu nám potom vychádza, že proces tvorby softvéru nie je procesom tvorby produktu, ale ide tu o proces získavania vedomostí a ich ukladania do softvéru. Uvedomiť si to môžeme aj my, keď sa zamyslíme nad tým, ako tvoríme aj ten najjednoduchší program. Zrejme nikto z vás sa nezaobrá prvotne tým, ako napíše samotný program, ale tým čo bude ten program robiť, ako to bude robiť a prečo to bude tak robiť. No a tento proces sa dá nazvať procesom získavania vedomostí, ktoré sú potrebné na vytvorenie programu.

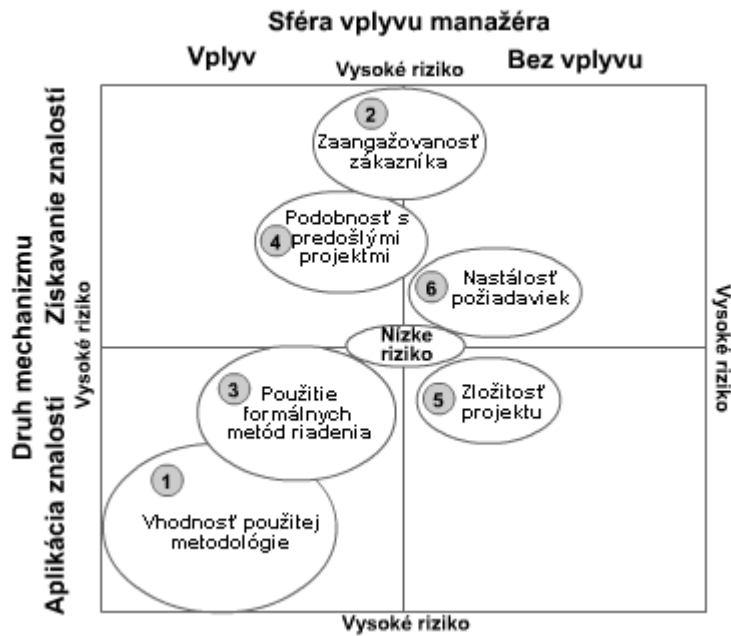
Až keď si uvedomíme, čo je to softvér a aké špecifiká má proces jeho vývoja, tak až potom sa môžeme sústrediť na skutočné riziká spojené s týmto vývojom.

## Šest' ohrození procesu vývoja

Dobrý manažment by sa nemal pri vývoji softvéru sústreďovať na odstraňovanie problémov, ale mal by predchádzať ich vzniku, čiže by sa mal snažiť identifikovať a eliminovať hlavné riziká procesu. Zdrojov rizík, ktoré číhajú na vývojárov počas tvorby softvéru je veľké množstvo. Ja by som sa chcel venovať iba tým, ktoré boli podľa výskumu, autorov článku [1] identifikované ako najvplyvnejšie. Mechanizmy procesu, ktoré môžu byť zdrojom rizika sa dajú rozdeliť do dvoch skupín:

- mechanizmy, ktoré riadia (zabezpečujú) prístup k znalostiam zákazníka a znalostiam o doménovej oblasti, v ktorej sa problém rieši. Sem patrí podobnosť s predchádzajúcimi projektmi, zaangažovanosť zákazníka, premenlivosť požiadaviek na projekt
- mechanizmy, ktoré určujú spôsob, akým sa získané vedomosti aplikujú, zabudujú do projektu. Medzi tieto mechanizmy patrí použitie vhodnej metodológie, používanie formálnych metód riadenia projektu a zložitosť projektu.

Z charakteristík projektov, ktoré boli analyzované sa zistil relatívny vplyv týchto mechanizmov na úspech projektu. Okrem miery rizika, ktoré je s nimi spojené sa dajú mechanizmy deliť aj podľa toho, či sa daný mechanizmus nachádza, alebo nenachádza v oblasti vplyvu manažérov projektu. Na diagrame (Obr. 1) je vidieť mieru vplyvu mechanizmu (čím väčšia je bublina, tým dôležitejší je daný mechanizmus), zaradenie do skupiny a zároveň sa dá vidieť, aký dosah má manažér na daný mechanizmus.



Obr. 1 Zdroje rizík

Ako sa dá vidieť, tak mechanizmy, ktoré sú mimo pôsobnosti manažéra majú najmenší vplyv na výsledok projektu. To je žiadateľné, avšak niekedy sa venuje týmto mechanizmom príliš veľká pozornosť, na úkor zanedbania ostatných mechanizmov, ktoré sú väčším zdrojom rizika.

Treba podotknúť, že výsledky, ktoré dosiahli sa týkajú hlavne projektov, z ktorých vychádzali a ich aplikácia na všetky softvérové projekty by bola príliš odvážna, keďže nezohľadňujú všetky možné riziká spojené s vývojom softvéru. Je to preto, že prvotným úmyslom nebolo identifikovať všetky možné riziká, ale cieľom bol vytvoriť jednoduchý nástroj, ktorý by porovnaním dvoch podobných projektov dokázal určiť mieru rizika pre projekt. Na porovnanie uvádzam zoznam desiatich najväčších rizík, ktoré boli identifikované na základe štúdie z roku 1998.[3]

1. Nedostatok zainteresovanosti vrcholového manažmentu
2. Nezainteresovanosť zákazníka
3. Nepochopenie požiadaviek
4. Nedostatok angažovanosti používateľa
5. Neúspech splniť požiadavky používateľa
6. Zmena cieľu/rozsahu projektu
7. Nedostatok požadovaných znalostí/zručností/skúseností vývojového tímu

8. Nedostatok zmrazených požiadaviek
9. Vovedenie novej technológie
10. Nedostatočné/nevhodné obsadenie pozícií v tíme
11. Nezhody medzi oddeleniami v organizácii zákazníka

Na prvé pozretie si môžeme všimnúť, že chýba *Vhodnosť použitej metodológie*. Tento zdroj rizika prekvapil aj samotných autorov článku[1]. Zásadnejšie však je, že v procese tvorby je tento zdroj často zanedbávaný.

### **Vhodnosť použitej metodológie**

Vedenie projektov často nevenuje pozornosť voľbe vhodnej metodológie a zameriava sa iba na to, aby sa vytvoril softvér, ktorý snád' bude robiť to čo má. Pri tomto zdroji si treba uvedomiť, že neexistuje univerzálna metodológia, ktorá by sa dala aplikovať na každý druh softvéru.

Na tomto mieste sa hodnotí hlavne výber metodológie a miera vhodnosti použitej metodológie na konkrétny projekt. V kontexte „softvér ako médium“ ide o to, aby sa vybrala metóda, ktorá by pomohla riadeným spôsobom zachytiť, čo najviac vedomostí od zákazníka a spojiť ich z vedomosťami tvorcov softvéru s cieľom vytvorenia komplexného riešenia.

Napríklad, ak vyvíjame softvér v oblasti, v ktorej sme nováčikovia, je výhodné použiť evolučný model v spojení s prototypovaním. Pri tejto metóde sa totiž znižuje riziko pre nové aplikácie, keďže špecifikácia a implementácia sú stále v súlade.

Na druhej strane ak máme za úlohu vytvoriť systém, ktorý je do veľkej miery podobný systému, ktorý sme už vyvíjali je dobré použiť napríklad vodopádový model. Tento znižuje riziko pri vývoji dobre zvládnutých aplikácií, kde sa používajú známe postupy riešenia. Je veľká pravdepodobnosť, že pri oboch projektoch sa budeme počas vývoja stretávať s rovnakými problémami a budeme musieť prekonávať podobné úskalia. Preto je dobré pri vývoji akéhokoľvek projektu dôkladne dokumentovať celý proces. Táto dokumentácia by nemala byť iba akýmsi denníkom toho, čo sa nám podarilo, ale mali by tu byť zachytené aj problémy, ktoré sa počas vývoja vyskytli. Lebo aj tieto komplikácie a ich možné riešenie sú novými znalosťami, ktoré nám neskôr môžu pomôcť.

### **Nedostatočná zaangažovanosť zákazníka**

Druhým zdrojom rizika je laxný prístup zákazníka k projektu. Nemôže však chcieť, aby zákazník chodil k nám a hovoril nám, čo chce. To je naša úloha, aby sme od neho dostali čo najviac vedomostí.

V tejto fáze je najväčším problémom, že zákazník dokáže vyjadriť oveľa menej ako v skutočnosti naozaj vie. Dalším problémom je, že zákazník často ani nevie čo presne chce. Tu sa dá s výhodou použiť prototypovanie, kedy sa vytvorí časť systému, napríklad obrazovky, zákazník si to prejde a povie nám, čo by chcel zmeniť, čo by chcel pridať a takto spolu sním vyvíjame systém. Ak však túto fázu zanedbáme môže sa nám stať, že aj keď vytvoríme funkčný systém, tak naše riešenie nemusí zodpovedať

požiadavkám zákazníka. Ak však budeme výdatne spolupracovať so zákazníkom, môžeme si ušetriť naozaj veľké množstvo roboty. To sa netýka len časti špecifikácie, ale aj analýzy a časti návrhu.

### **Nedostatočné používanie formálnych metód riadenia**

Riziko neúspechu, môžeme znížiť aj používaním smerníc a definovaných modelov riadenia. Základom úspechu je, aby sme v okamihu, keď ideme niečo robiť, vedeli presne, čo ideme robiť, načo to ideme robiť a dokedy to musíme spraviť. K tomuto nám pomáhajú plány a rozvrhy, ktoré vytvárajú vedúci projektov. Tieto plány môžu byť ďalej zjemňované na úrovni pracovných skupín. Takto si môžeme definovať, čo by mal kto a kedy vykonať. Na druhej strane ak chceme skontrolovať, či sa dané činnosti naozaj vykonali, alebo aj ako sa vykonali, môžeme použiť napríklad míľniky, čiže v nejakom čase skontrolujeme stav prác na projekte. Pomocou nich môžeme odhaliť sklzy, alebo prípadné nezrovnalosti v procese vývoja a vykonať tak nápravu. Takéto parciálne kontroly sú určite vhodnejšie ako keby sme išli zisťovať nezrovnalosti až na konci projektu.

### **Podobnosť s predchádzajúcimi projektmi**

Asi každému je jasné, že ak robím niečo podobné ako som už robil je väčšia šanca že to zvládnem, ako keď idem robiť novú vec. Aby som si mohol proces z predošlého projektu využiť, musí byť kvalitne zdokumentovaný. Znova keď sa na proces tvorby softvéru pozrieme, ako na proces získavania vedomostí, tak aj podobné predchádzajúce projekty môžeme brať ako časť technických znalostí, ktoré ak sú a sú kvalitné, určite pomôžu v procese vývoja.

### **Zložitosť projektu**

Na zložitosť projektu ako na zdroj rizika sa dá pozeráť z dvoch pohľadov. Zložitosť môžeme chápať z technického hľadiska a potom riziko sa vyskytuje, hlavne v prípadoch keď vyvíjaný softvér je súčasťou iného systému. Zaujímavejšou z hľadiska manažmentu je však zložitosť organizačná, ktorá vychádza z organizačnej štruktúry celého tímu, ktorý sa podieľa na procese. Nie vždy väčší počet pracovníkov znamená väčší výkon. Treba voliť zdroje tak, aby nevzniklo presýtenie, a treba sa sústrediť aj na odstraňovanie problémov, pri komunikácii medzi organizačnými jednotkami.

### **Nestálosť požiadaviek**

Prekvapením výskumu bolo, že na posledný zdroj rizika sa často sústredila priveľká pozornosť na úkor iných zdrojov. Na druhej strane cesta úplnej ignorácie tiež nevedie k úspechu. Túto situáciu presne vyjadruje myšlienka, ktorá hovorí, že „Stavať softvér na premenlivých požiadavkách je ako stavať dom na základoch z piesku“ [1]. Na druhej strane, predstava, že požiadavky sa nebudú meniť je naivná a takáto situácia zrejme ani nemôže nastať. Preto sa netreba snažiť o to aby sa požiadavky vôbec nemenili, ale skôr treba predvídať možné zmeny a byť na ne pripravený.



## Záver

Zdroje rizík, alebo mechanizmy procesu ktoré, v ktorých sa môžu vyskytnúť problémy ohrozujúce úspech projektu, tvoria iba podmnožinu všetkých zdrojov rizika.

Ešte raz by som však chcel poukázať na základný zdroj, ktorý vedie k neúspechu projektov. Áno, znova rozprávam o chápaní softvéru. Ak sa budeme k procesu vývoja stavať, tak ako keby cieľom bolo vytvoriť produkt a tomu budeme podriaďovať všetku našu námahu, nemusí byť naša námaha vždy korunovaná úspechom. Aj keď zmeniť myslenie v dnešných podmienkach je veľmi ťažké, keďže tímy sú pod silným tlakom zo stany trhu, ktorý ich núti vyvíjať stále rýchlejšie a s nižšími nákladmi. Ale aj tak si myslím, že by stálo za pokus, trochu spomaliť a sústrediť sa skutočnú podstatu vytvárania softvéru, ktorá nie je v generovaní kódov, ale skutočne je to činnosť, pri ktorej sa snažíme naše vedomosti a vedomosti ostatných účastníkov procesu získať, transformovať a uložiť do softvéru. Ak sa nám podarí orientovať myslenie a chápanie softvéru a procesu jeho tvorby týmto smerom, určite sa to odrazí aj na kvalite nášho výsledného riešenia.

## Použitá literatúra

1. Tiwana A., Keil M.: The One-Minute Risk Assessment Tool. *Communications of the ACM*, Vol. 47, No. 11 (November 2004), 73-77.
2. Armour, P.: A case for a new business model: Is software a product or a medium? *Communications of the ACM*, Vol. 43, No. 8 (August. 2000), 19–22.
3. Bieliková M.: *Softvérové inžinierstvo - Princípy a manažment.*, STU, Bratislava, 2000

## Annotation

### *Software development risks*

Software business is one of the most expansive business segments. This expansion brings a big risk to the project, if software development process is not managed and doesn't focus attention on the true risk associated with this process. First part of my essay is dedicated to the basic source of software failures that comes out from bad thinking of software as a product rather than a medium in which knowledge is embodied. Further I will present role of this medium within the scope of other knowledge mediums and results of this thinking of software to its development process. Second part is then dedicated to the factors or risk resource that can attack this process.