

# HLAVNÉ RIZIKÁ V ŠKOLSKÝCH SOFTVÉROVÝCH PROJEKTOCH

*Najväčšie riziko prichádza s predstavou, že nám  
nehrozia žiadne riziká.*

*Jozef Kukuča*

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

**Abstrakt.** Rovnako ako v každom softvérovom projekte, aj v tom školskom hrozia rôzne riziká. Tie môžu nepriaznivo ovplyvniť jeho výsledok. Vo všeobecnosti sa pri softvérových projektoch ako kritéria úspechu berú kvalita výrobku, vynaložený čas a veľkosť vynaložených zdrojov. V školských projektoch je čas na projekt pevne určený fixným termínom odovzdania a rozpočet nie je väčšinou žiadny. Preto za jediné relevantné kritérium pre určenie úspechu takéhoto projektu považujem kvalitu výrobku alebo inými slovami mieru úspešnosti splnenia požiadaviek. V tejto eseji diskutujem, ktoré známe riziká hrozia práve pri školských softvérových projektoch. Na základe zoznamov známych TOP rizík určím, ktoré riziká považujem za TOP 10 pri školských projektoch a ktoré naopak za kritické nepovažujem. Ďalej uvádzam, aké metódy manažmentu rizík si myslím, že je vhodné použiť pri tomto druhu projektov. Pri ich výbere je dôležitá jednoduchosť a zároveň efektívnosť. Študenti sú začínajúci vývojári, čo má za následok neskúsenosť a tú treba zohľadniť do vhodnej zložitosti používaných postupov a praktík. Takisto majú dosť obmedzený čas na vývoj, na čo treba brať takisto ohľad.

**Kľúčové slová:** manažment rizík, Boehm, softvérový projekt

## Úvod

Školské softvérové projekty sú v mnohých ohľadoch rozdielne od klasických softvérových projektov vo firmách. Vo všeobecnosti by sa dalo povedať, že školské projekty sú

limitované rôznymi obmedzeniami ako menší čas, skúsenosti či rozpočet. Z týchto a iných dôvodov sú riziká späť so školskými softvérovými projektmi odlišné od bežných rizík v softvérových projektoch a ich manažment taktiež vyžaduje iné prístupy.

## Aké riziká hrozia?

Boehm identifikoval nasledujúcich TOP 10 rizík [1]:

1. Nedostatok personálu
2. Nerealistické rozvrhy a rozpočty
3. Vytvorenie nesprávnej funkcionality
4. Vytvorenie nevyhovujúceho používateľského rozhrania
5. Pozlátenie systému
6. Neustále zmeny požiadaviek
7. Nedostatky v externe vytvorených moduloch
8. Nedostatky v externe vykonávaných úlohách
9. Nedostatočný výkon pre beh v reálnom čase
10. Precenenie technológie

Ďalej uvediem o ktorých z týchto rizík si myslím, že sú a naopak nie sú kritické pri projektoch, v ktorých sa vytvára určité softvérové dielo študentmi v rámci školy. Dôležité je uvedomiť si, že úspech týchto projektov nie je meraný finančným ziskom, ale dal by sa merať skôr čo najlepším hodnotením pedagóga na základe splnenie požiadaviek na tento projekt.

## Nepodstatné riziká

Najprv uvediem, ktoré z vymenovaných rizík nepokladám za kritické v školských softvérových projektoch.

Myslím si, že *problémy s nedostatkom personálu* nie sú časté, pretože požiadavky na vytváraný softvér sa často určujú na základe veľkosti vývojového tímu. Tým pádom, ak sa v priebehu vývoja zistí, že vytvorenie daného softvéru je nad sily tímu, tak sa požiadavky zjemnia. *Nedostatky v externe vytvorených moduloch* takisto nepokladám za kritické riziko, pretože študenti majú pomerne veľa času na dôkladnú analýzu takto používaných modulov. *Neustále sa meniace požiadavky* nebývajú pri takýchto projektoch príliš časté. Väčšinou sa príliš nemenia, a teda si nemyslím, že by sa ich bolo treba obávať a zaradiť ich ku kritickým rizikám. Požiadavky sa menia len v prípade, ak sa zistí, že boli stanovené nerealisticky vzhľadom k možnostiam tímu. A nakoniec by som vylúčil *nedostatky v externe vykonávaných úlohách*, pretože využívať externú pomoc je pri podobných projektoch zakázané. Respektíve, ak pod týmto chápem aj situáciu, kedy tím pracuje na doplnení nejakého už existujúceho softvéru o ďalšiu funkcionality, tak v tom prípade záleží od skúsenosti vývojárov daného softvéru. Ale možno predpokladať, že prípadné nedostatky daného softvéru boli zanalyzované, a teda sa na ne prihliada pri stanovovaní požiadaviek.

### Kritické riziká

Teraz uvediem, ktoré riziká z uvedeného zoznamu pokladám v kontexte školských projektov za kritické.

V prvom rade by som spomenul *nerrealistické rozvrhy*. Študenti často nebývajú veľmi skúsení v zostavovaní rozvrhov pre projekt, a preto si myslím, že častou príčinou neúspechu splnenia požiadaviek je práve zle vytvorený rozvrh. O rozpočte nemá zmysel hovoriť, keďže študenti žiadny rozpočet k dispozícii nemajú. *Vytvorenie nesprávnej funkcionality* alebo *používateľského rozhrania* je častý problém spôsobený hlavne zlou komunikáciou so zadávateľom úlohy. Takisto *pozlátenie* pokladám za možné riziko, pretože študenti niekedy v dobrej viere pridávajú do aplikácie určitú funkcionality, ktorá tam ale vzhľadom na požiadavky nemá čo robiť. Môžu ňou chcieť napríklad predviesť svoje zručnosti alebo jednoducho chcú, aby toho daný softvér robil viac a bolo vidno, že toho viac spravili. Môže to však mať negatívny dopad, lebo to môže odvieť pozornosť tímu od podstatnej funkcionality, ktorá vo výstupe projektu môže ostať chybová, nedostatočná alebo nestihnutá. Riziko *nedostatočného výkonu softvéru v reálnom čase* a *precenenie technológie* sú riziká, ktoré sú v školských projektoch reálne hlavne z dôvodu neskúsenosti študentov a ich nedostatočných vedomostiach ako týmto rizikám predísť.

Z TOP 10 Boehmových rizík som vybral šesť o ktorých si myslím, že sú relevantné pre školské softvérové projekty. Skúsím teraz doplniť ďalšie štyri riziká o ktorých si myslím, že by v súčte so šiestimi menovanými mohli vytvoriť TOP 10 rizík v školských softvérových projektoch. Čerpám pritom zo zoznamu rizík získaných v štúdiu [2]:

1. Nedostatočné / nevhodné obsadenie pozícií v tíme
2. Nezainteresovanosť zákazníka
3. Nedostatok požadovaných znalostí a zručností vývojového tímu
4. Nepochopenie požiadaviek

Myslím, že tieto položky predstavujú relevantné riziká v školských softvérových projektoch. *Nedostatočné alebo nevhodné obsadenie pozícií v tíme* môže mať za následok pomalý vývoj časti softvéru, ktorá by mala byť vyvíjaná expertom na danú oblasť. Príčinou sú hlavne osobné preferencie namiesto odborných pri zostavovaní tímu. Avšak aj pri vhodnom zostavení tímu môže vzniknúť riziko *nedostatku požadovaných znalostí a zručností*. Je to kvôli ich nie veľkým skúsenostiam, veď aj to je cieľom školských projektov – aby ich získali. *Nezainteresovanosť zákazníka*, v tomto prípade lepšie povedané zadávateľa projektu môže nepriaznivo ovplyvniť proces identifikácie požiadaviek na projekt a tým výrazne sťažiť celý priebeh vývoja. S týmto úzko súvisí aj *nepochopenie požiadaviek*. Avšak myslím, že podstatná časť neúspechu pri procese porozumenia požiadaviek je zakotvená v neskúsenosti študentov v jednaní so zákazníkom (zadávateľom).

### Ako si poradiť s rizikami?

Prevažná väčšina rizík v školských softvérových projektoch je podľa môjho názoru zapríčinených neskúsenosťou študentov. Jedna vec je výber študentov pri skladaní tímu, avšak ak je tím raz zložený, už nemožno vymeniť členov a treba vzniknutú situáciu riešiť.

Prvý krok je identifikácia rizík a priradenie určitej váhy každému z nich. Na túto úlohu podľa môjho názoru skvele poslúži vytvorenie zoznamu rizík a určenie veľkosti rizika maticou pre kvantifikáciu rizika uvedenou v [3]. Na tomto prístupe považujem za hlavnú výhodu jeho jednoduchosť a účinnosť. Ako zoznam rizík môže poslúžiť dekompozícia projektu na úlohy a určenie pravdepodobnosti ich neúspechu a vplyv tohto neúspechu na splnenie požiadaviek projektu. Na riziká s väčšou váhou je potom nutné sa viac sústrediť ako na tie s menšou. Dobrá komunikácia v tíme môže výrazne znížiť pravdepodobnosť rizika, preto je podľa môjho názoru treba sa dohodnúť na pravidelných stretnutiach a diskutovať na nich riziká.

Najvhodnejším spôsobom ako riešiť riziká spojené s neskúsenosťou vývojárov je podľa mňa vytváranie modelov, prototypov, simulácií a diskutovanie o nich v rámci tímu a prípadne s nejakou skúsenou osobou, ktorá pri školských projektoch môže byť zadávajúci pedagóg. Keď je problém správne pochopený členmi tímu, je potom omnoho ľahšie prísť s nápadom na jeho úspešné riešenie. A práve spomenuté metódy sú veľmi účinné pre správne pochopenie problému.

Pre účel monitorovania rizík je dobré vytvoriť zoznam momentálne najrizikovejších položiek a ten pri každom stretnutí obmieňať na základe zmeny váhy rizika jednotlivých položiek. Tá sa môže často meniť, preto považujem za vhodné nevynechávať tieto stretnutia a zaistiť aby boli skutočne pravidelné.

Krok učenie, v ktorom sa vytvorí databáza s rizikami môže byť užitočný pre komunikovanie rizík z minulých projektov a pomoc s identifikovaním rizík v súčasných projektoch na základe minulých skúseností. Avšak pri tomto druhu projektu to môže byť kontraproduktívne, ak je len malá šanca, že by tento istý tím v budúcnosti pracoval opäť v podobnom zložení na projekte podobného charakteru. A takisto vytvorenie takejto databázy a jej udržovanie je podľa mňa pri takomto druhu projektu zbytočne nákladné na čas.

## Záver

Aj v školských softvérových projektoch hrozí veľa rizík, ktoré môžu zapríčiniť ich neúspech. Keďže manažment rizík je veľmi dôležitá súčasť v procese vývoja softvéru, je dôležité, aby študenti pochopili jeho význam a osvojili si účinné praktiky pre jeho riešenie. Toto sa im neskôr môže vrátiť v podobe efektívnejšieho a rýchlejšieho vývoja softvérového diela a takisto im to dá základ pre riešenie rizík v praxi do ktorej sa dostanú po ukončení štúdia. V minulosti bola manažmentu rizík venovaná malá pozornosť, ale v dnešnej dobe ak má byť softvérový projekt úspešný, tak sa bez neho neobíde. Preto si myslím, že je veľmi vhodné tieto princípy používať v školských softvérových projektoch.

## Použitá literatúra

1. Boehm, B.W.: Software risk management: principles and practices. In: *Software*, IEEE, vol. 8, 1991, 32 – 41.
2. Keil, M. et al.: A framework for identifying software project risks. In: *Communications of the ACM*, vol. 41, 1998, 76 – 83.

3. Zardari, S.: Software Risk Management. In: *Information Management and Engineering*, 2009. ICIME '09, 375 – 379.

## **Annotation**

### *Main risks in school software projects*

*As well as any software project even the school one is threatened with various risks. These can affect its outcome. In general, main success factors in software projects are cost, time and quality. In school projects time is determined by fixed deadline and cost is not relevant because usually there is no budget. Therefore I assume that the only criterion for determination whether or not a school project is successful is by measuring its quality. In this essay I discuss which of the known risks are relevant for school software projects. Based on lists of TOP risks I determine which of them I assume to be the most crucial ones. Next I present which methods of risk management I think are suitable for use in this kind of software projects. They should be simple and effective. Students have often lack of needed skills and this should be considered in choosing techniques and practices.*