

# HROZBY V TREŤOM SEKTORE PRI INTERNOM VÝVOJI SOFTVÉRU

*Ono je to aj o prístupe zväzu, na jednej strane sa tam  
všetci sťažujú, že to robia zadarmo, na druhej strane  
k tomu nechcú nikoho pustiť.*

*Ondrej Danada*

Slovenská technická univerzita  
Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava  
ondrej[zavináč]danada[.]sk

**Abstrakt.** Organizácie tretieho sektora majú často veľmi obmedzený rozpočet, a preto sú nútené realizovať svoje softvérové projekty vo vlastnej réžii. Tieto typy projektov majú svoje špecifické charakteristiky, a preto sa s ich prípravou a realizáciou spájajú trochu odlišné riziká ako v komerčných projektoch profesionálov. Cieľom tejto eseje je identifikovať a stručne opísať tieto charakteristiky a porovnať riziká s rizikami bežných softvérových projektov. Za najväčšie riziko možno pomerne jednoznačne považovať nedostatok personálu, ktoré hrozí vo zvýšenej miere a jeho dopady sú horšie ako pri iných typoch projektov. Možnosti predchádzania tejto hrozbe sú taktiež obmedzené a treba siahnuť po iných riešeniach. Niektoré riziká sú však vďaka určitým charakteristikám ako napríklad dobrá znalosť domény skôr zanedbateľné a hrozba, že nastanú, je minimálna.

**Kľúčové slová:** riziká, TOP10, tretí sektor, interný vývoj, svojpomocne, vo vlastnej réžii

## Úvod

Riziko by sme mohli definovať ako možnosť výskytu nepredvídanej udalosti, ktorá spôsobí škodu alebo stratu. Pri identifikácii rizík je potrebné určiť pravdepodobnosť, že udalosť nastane, a odhad rozsahu škôd spôsobených touto udalosťou. Riziko sa vo

## 2 Ondrej Danada

všeobecnosti ohodnotí vynásobením pravdepodobnosti odhadom škôd. Možno ich klasifikovať podľa troch dimenzií. Prvá dimenzia je tá, či ide o riziko nákladov, rozvrhu alebo splnenia požiadaviek. Druhá dimenzia nám rozdeľuje riziká na interné a externé podľa toho, či môžeme alebo nemôžeme priamo riadiť. Tretia dimenzia delí riziká na generické a špecifické podľa toho, či sú spoločné pre všetky softvérové projekty alebo vyplývajú zo špecifik konkrétneho projektu.

V tejto eseji rozoberám najmä odlišnosť riadenia generických rizík v špecifickom type projektov. Generické riziká možno identifikovať v takmer každom softvérovom projekte, ale ich pravdepodobnosť výskytu alebo rozsah škôd sa môže vzhľadom na charakteristiky projektu výrazne odlišovať. Taktiež ich riadenie môže byť odlišné. Je totiž dosť veľký rozdiel medzi rizikami hroziacimi vo veľkej softvérovej firme, v školskom tímovom projekte alebo v treťom sektore.

Motiváciou tejto eseje je projekt informačného systému pre Slovenský šachový zväz, ktorý sa rozhodol pre nedostatok finančných zdrojov tento systém neobjednať u veľkej firmy, ktorá má predpripravený produkt pre športové zväzy, ktorý dorába na mieru.

Namiesto toho pracuje na vývoji niekoľko dobrovoľníkov v tíme, ktorí pracujú bez nároku na finančnú odmenu, sú riadne zamestnaní alebo študujú a práca na tomto projekte nie je ich prioritou. Často sú nedostatočne kvalifikovaní, nespoľahliví, majú osobné spory a každé nesprávne rozhodnutie môže znamenať, že niekto odíde. Pre projekt takéhoto druhu je riadenie rizík trochu odlišné ako na komerčných projektoch veľkých firiem.

### Desať najväčších rizík

Boehm [1] identifikoval 10 najväčších rizík:

- nedostatok personálu,
- nerealistické rozvrhy a rozpočty,
- vývoj nesprávnej funkcionality,
- vývoj nevyhovujúceho používateľského rozhrania,
- pozlátenie systému,
- spojené zmeny požiadaviek,
- nedostatky v externých moduloch,
- nedostatky v outsourcovaných úlohách,
- nedostatky výkonu v reálnom čase,
- precenenie technológií.

Tieto riziká majú však rôzny negatívny dopad a rôznu pravdepodobnosť, že nastanú, vzhľadom na typ projektu. Napríklad vo veľkej firme sa dá momentálny nedostatok personálu riešiť realokáciou ľudských zdrojov na projekty, zatiaľ čo menšia firma môže mať s prijatím nového človeka do tímu väčší problém. V školských projektoch je prijatie nového člena do tímu často neprípustné.

Po charakterizovaní špecifických vlastností vývoja softvéru v treťom sektore sú tieto riziká s ohľadom na tieto charakteristiky podrobnejšie opísané ďalej v texte.

## Charakteristika softvérového projektu organizácie tretieho sektora

Pre softvérové projekty týchto organizácií, ktoré si projekty realizujú samé, sú najvýznamnejšie charakteristiky nasledujúce. Na menšom projekte pracuje menší počet ľudí, tak okolo 4-7 osôb. Títo ľudia majú veľmi dobrú znalosť biznisu, avšak niektorí z nich majú často nízku kvalifikáciu, nepoznajú rovnaké technológie a chýbajú im skúsenosti s prácou v tíme. Komplikáciou je aj to, že sú od seba dosť vzdialení na to, aby sa osobne stretávali. Napriek tomu musia robiť všetky rozhodnutia v rámci tímu, lebo nie je s kým kompetentným konzultovať, keďže neexistuje reálny zákazník. Ide totiž o interný vývoj. Typické sú aj príliš voľne definované termíny, často bez sankcií za ich nedodržanie.

Vzhľadom na tieto charakteristiky býva často problém zostaviť tím a nad výberom správnych ľudí sa nedá ani uvažovať, keďže projektový manažér je vďačný za každého, kto je ochotný niečo robiť. Preto odchod ktoréhokoľvek člena tímu môže predstavovať nenahraditeľnú stratu.

Častým sprievodným javom je aj to, že jednotliví členovia tímu majú medzi sebou osobné spory, nevedia sa dohodnúť, prípadne, že do tímu prenášajú svoje konflikty a problémy politického charakteru, ktoré s vývojom nesúvisia.

## Riziká v treťom sektore

Niektoré riziká od Boehma [1] sa vyskytujú aj v treťom sektore, no väčšinou je ich riadenie trochu odlišné od riadenia rizík vo veľkej softvérovej firme, prípadne, niektoré riziká nehrozia takmer vôbec, alebo v minimálnej miere, resp. ich dopad je občas zanedbateľný. Taktiež existujú aj riziká, ktoré Boehm [1] neidentifikoval alebo ich nepovažoval za dôležité, a pritom sú pre tretí sektor dôležité.

### Nedostatok personálu

Toto riziko je pri silne obmedzenom rozpočte a ľuďoch pracujúcich bez finančného ohodnotenia jednoznačne najvyššie, keďže pravdepodobnosť jeho výskytu je vysoká a negatívne dôsledky môžu viesť až k neúspechu projektu.

Boehm [1] odporúča riešiť tieto riziká výberom vhodných ľudí, budovaním tímu, zazmluvnením kľúčových osôb a vzájomné školenia. Výber vhodných ľudí a zazmluvnenie kľúčových osôb nie je v takomto druhu projektu reálne, keďže záujem o zapojenie sa do projektu je vo všeobecnosti nízky a nie je si z čoho vyberať a ťažko niekoho zazmluvniť bez zdrojov.

Vzájomné školenia sú vhodným riešením nedostatočnej kvalifikácie, nedostatku skúseností a neznalosti technológií. Riešiť takéto školenia je však po organizačnej stránke dosť náročné, keďže členovia tímu sa nemôžu osobne stretnúť a nie je jednoduché stretnúť sa ani prostredníctvom konferenčného hovoru. Preto treba prínos školení dobre zvážiť, či je ten spoločne strávený čas, ktorého je málo, využívaný efektívne.

Je nevyhnutné udržiavať motiváciu členov tímu, aby neodišli. Kľúčovú úlohu zohráva komunikačná stratégia. Treba často prosiť, chváliť, ďakovať a prechádzať konfliktom a osobným sporom.

Ako príklad by som uviedol informačný systém Slovenského šachového zväzu, kde niektoré neuvážené príspevky p. prezidenta na diskusnom fóre odradili jedného

#### 4 Ondrej Danada

zanieteneho programatora, ktorý teraz odmieta akúkoľvek ďalšiu spoluprácu. Ako protiklad možno uviesť iného programatora, ktorý nekonečne svoje úlohy odkladá, ale občasná verejná kritika ho dokáže namotivovať k tomu, aby sa posunul ďalej. Je to veľmi individuálne a treba najmä pri tej komunikácii postupovať veľmi citlivo.

#### **Nerealistické rozvrhy a rozpočty**

Toto má relatívne vysokú pravdepodobnosť, že nastane, lebo rozpočet býva často poddimenzovaný a rozvrh príliš optimistický. Častým dôsledkom je nesplnenie termínu, čo nie je až taký problém, keďže termíny väčšinou nebývajú definované striktné a za ich nedodržanie väčšinou nehrozia žiadne sankcie.

Niekedy to však zájde až príďaleko a nekonečným posúvaním termínov môže nastať situácia, že projekt sa nikdy nedokončí. Napríklad projekt automatizovaného spracovania ligových výsledkov je už 4 roky vo fáze návrhu a ešte sa nezačala implementácia.

#### **Vývoj nesprávnej funkcionality**

Toto riziko v takomto type projektu nehrozí takmer vôbec, keďže ľudia majú dobré znalosti biznisu, a preto by nemalo dochádzať k nedorozumeniam. Občas sa však môže stať, že niektorý člen tímu precení svoje doménové znalosti, ktoré získal ako účastník v jednej z rolí, pričom zabúda na tie ostatné. Jednoduchým riešením, ako tomu predchádzať, je intenzívna komunikácia v rámci tímu a keďže ostatní členovia poznajú doménovú oblasť takisto, dokážu včas odhaliť nedostatok.

Ako príklad by som uviedol tréningový portál Šachu na školách (TP-CIS), kde manažment Šachu na školách (rozvojový projekt SŠZ) nedostatočne komunikoval, tak to zobral do rúk jeden analytik, ktorý bol zhodou okolností aj trénerom a trochu videl aj do manažmentu, ale nie dostatočne. Dôsledkom bolo, že sa vychádzalo z nesprávnych biznis procesov, ktoré manažment pravdepodobne bez prečítania odsúhlasil, a tie sa následne nesprávne implementovali. Systém je tak momentálne nepoužiteľný bez zapracovania zmien, len hlavný programátor sa medzi tým odsťahoval do zahraničia. Toto je však jediný prípad, kedy som sa stretol s implementovaním nesprávnej funkcionality, väčšinou sa zlá analýza odhalí najneskôr vo fáze návrhu.

#### **Vývoj nevyhovujúceho používateľského rozhrania**

Toto riziko by zdanlivo nemalo hroziť takmer vôbec vďaka dobrým znalostiam biznisu, prax však ukazuje opak. Kvôli nedostatočným zdrojom, nedostatočnej kvalifikácii vývojárov a mylnému očakávaniu v zručnosti bežného používateľa sa často stáva, že používateľské rozhranie je nevyhovujúce. Boehm [1] odporúča konzultovať grafické návrhy obrazoviek so zákazníkom, čo však podľa môjho názoru postačujúce nie je. Myslím si, že najvhodnejšie by bolo predvádzať prototyp pred koncovými používateľmi, aj keď niektoré ergonomické problémy sa objavia až pri reálnej prevádzke.

Typickým príkladom je turnajový registračný systém, ktorý mal zrýchliť proces registrácie na turnaji s veľkým počtom účastníkov, zamedziť chybám ľudského faktora a prispieť k vyššej spokojnosti účastníkov. Ergonómia systému bola však natoľko zlá, že sa systém použil na jedinom turnaji a od vtedy sa nepoužíva.

### **Pozlátanie systému**

Pozlátanie systému je v tomto type projektov veľmi zriedkavé, keďže sa často robia minimalistické verzie a nie je implementované ani to, čo by malo byť, ale občas sa stane, že niekto, kto rieši svoju obľúbenú oblasť, sa nechá uniesť a často sa navrhuje funkcionality pre jedného človeka. Boehm [1] navrhuje robiť analýzu prínosov a v tomto prípade naozaj súhlasím a zdôrazňujem, že je to veľmi dôležité, a to nie len kvôli vyhnutiu sa pozlátaniu systému, ale aj kvôli stanoveniu priorít, keďže zdrojov je málo a úloh veľa.

Ako príklad uvádzam automatizovaný výpočet ratingov, ktorý momentálne vykonáva jedna osoba ručne. Automatizácia výpočtu je síce prínosom, ale šetrí čas jedinej osobe, zatiaľ čo iné časti systému dokážu šetriť čas desiatkam ľudí.

### **Spojité zmeny požiadaviek**

Sú oblasti, kde sa požiadavky menia pomerne často, a sú také, kde sa požiadavky menia menej často. Problémom však je, že kvôli nedostatku zdrojov životný cyklus vývoja je príliš dlhý a občas sa stane, že vypracovaná analýza začne byť v čase implementácie neaktuálna. Boehm [1] odporúča agilný vývoj, čo je do istej miery aj realizovateľné.

Jedným z príkladov je aj integrácia na rôzne iné systémy, ktoré nevystavujú žiadne rozhranie určené na integráciu, ale je nutné spracovať html výstupy. Pri akejkoľvek zmene v externom systéme je nutné prerábať HTML Parser.

### **Nedostatky v externých moduloch, nedostatky v outsourcovaných úlohách, nedostatky výkonu v reálnom čase**

Tieto riziká sa často nevyskytujú, keďže outsourcované nie je takmer nič, externé moduly sa príliš nevyužívajú a problémy s výkonom nehrozia.

### **Precenenie technológie**

Nakoľko vývojový tím nepozná takmer žiadne technológie, neviem si celkom dobre predstaviť, ako by sa dali preceniť. Občas sa však stane, že je použitá nevhodná technológia. Boehm [1] odporúča ako jedno z riešení technickú analýzu, s čím súhlasím. Ako príklad uvádzam nevhodné použitie XML v Tréningovom portáli (TPCIS), ktoré podstatne spomalilo a skomplikovalo vývoj.

### **Nedostatok požadovaných znalostí, zručností a skúseností**

Toto riziko sa uvádza v štúdiu [2] a je typické práve pre tretí sektor, dokonca by som povedal, že jedno z najväčších, keďže je vysoká pravdepodobnosť jeho výskytu a zároveň môže mať veľký vplyv na úspešnosť projektu. Trochu súvisí aj rizikom nedostatku personálu od Boehma [1]. Riešením by boli vzájomné školenia a metodiky, problém však je v tom, že je s tým často spojená priveľká réžia. Keď sa aj človek už niečo naučí, občas sa stane, že odíde a potom je problém opätovne treba riešiť.

### **Nedostatok angažovanosti používateľa**

Aj toto riziko je uvedené v štúdiu [2] a tiež som sa s ním často stretol. Po niekoľkých neúspešných projektoch majú používatelia nedôveru v úspešne dokončenie projektu,

## 6 Ondrej Danada

nečítia sa byť kompetentní niečo k tomu povedať alebo sa spoliehajú na to, že sa vyjadrí niekto iný. Tu by som videl ako riešenie častú komunikáciu s používateľmi, predvádzanie prototypov a motivovanie, aby sa zapojili.

Ako príklad by som uviedol opäť Tréningový portál, kde sa každý pýtal, kedy to už bude, ale nikto nebol ochotný podať akékoľvek pripomienky či námety. Každý iba čakal, že sa to urobí samo.

### **Nedostatočné monitorovanie priebehu projektu**

Toto riziko je uvádzané v analýze [3] a súvisí s nedostatočnou angažovanosťou manažmentu, ktorá sa uvádza v štúdiu [2]. Projektový manažment si tiež vyžaduje svoje zdroje a pokiaľ nefunguje tak, ako by mal, a priebeh projektu sa nemonitoruje dostatočne, často sa zbytočne posúvajú termíny. Aj keď členovia tímu sú obvykle dosť vyťažení inými povinnosťami, ak sa nebudú dostatočne monitorovať, urobí sa ešte menej. Povedať, že treba monitorovať projekt je ľahké, ale niekto to musí tiež robiť a treba zabezpečiť, aby niekto monitoroval aj manažéra monitorovania.

Dovolím si tvrdiť, že v tomto prípade netreba uvádzať konkrétny príklad, keďže kvôli nedostatočnému monitorovaniu sa predĺžili takmer všetky projekty, na ktorých som sa podieľal.

### **Chýbajúci vyšší manažér**

Štúdia [4] ukázala, že častým rizikom je aj chýbajúci vyšší manažér, s čím tiež súhlasím. Ak by sa do softvérových projektov viacej angažovali vyšší manažéri, tempo vývoja by mohlo byť trochu svižnejšie.

Napríklad v SŠZ, ak by na každý projekt dohliadal nejaký člen Výkonného výboru podľa toho, akej oblasti sa projekt týka, pravidelne sa zaujímal o plán a jeho plnenie, verím tomu, že by sa pracovalo trochu efektívnejšie.

### **Záver**

Najväčším rizikom je jednoznačne nedostatok personálu. Nerealistické rozvrhy a rozpočty síce hrozia, ale až tak to nevedí, dokiaľ to nezájde príďaleko. Vývoj nesprávnej funkcionality, vývoj nesprávneho používateľského rozhrania a pozlátanie systému nie je až tak veľké riziko, ale občas nastane a dá sa celkom rozumne riešiť. Spojité zmeny požiadaviek nie sú až tak veľkým problémom a dajú sa čiastočne riešiť agilným vývojom. Nedostatky v externých moduloch, nedostatky v outsourcovaných úlohách a nedostatky výkonu v reálnom čase nehrozia takmer vôbec. Precenenie technológií takisto nehrozí, skôr naopak, veľké riziko je ich neznalosť, resp. celkovo chýbajúce znalosti, zručnosti a skúsenosti. K bezproblémovému priebehu projektu neprispieva ani nedostatočná angažovanosť používateľov. K rizikám rozvrhu prispieva vo väčšej miere aj nedostatočné monitorovanie projektu a chýbajúci vyšší manažér.

## Použitá literatúra

1. Boehm, B.W.: Software risk management: Principles and Practises. *IEEE Software*, Vol. 8, No. 1 (1991) 32-41.
2. Keil, M., Cule, P.E., Lyytinen K., Schmidt, R.C.: A framework for identifying software project risks. *Communications of the ACM*, 76, Vol. 41, No.2 (1998)
3. Han, W.M. and Huang, S.J.: An Empirical Analysis of Risk Components and Performance on Software Projects. *The Journal of Systems and Software*, Vol. 80 No. 1 (2007) 42-50.
4. Addision, T. and Vallabh, S.: Controlling Software Project Risks – an Empirical Study of Method used by Experience Project Managers. *Proceeding of SAICSIT (2002)*, 128-140.

## Annotation

### *Threads of the internal development in the tertiary sector*

*Organizations of the tertiary sector often have limited budget and that is why they have to develop their software projects at their own. This kind of projects has its's own specifics and because of that there are different risks than in commercial projects of professionals. The purpose of this essay is to identify and briefly describe these specifics and compare these risks to the risks of ordinal software projects. The worst risk could be clearly considered the lack of staff, which threads more than in other kinds of projects and its' consequences are bigger, too. Possibilities of avoiding these risks are also limited and it is necessary to look for special solutions. But some risks are because of some specifics, such as good domain knowledge insignificant and the thread is minimal.*