ŠPECIFIKÁ MANAŽMENTU RIZÍK PRI MEDICÍNSKOM SOFTVÉRI

„Lekársky výskum dosiahol takých pokrokov, že v konečnom dôsledku už nie je na svete zdravého človeka.“

Aldus Huxley

Jozef Gajdoš

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
xgajdos@fiit.stuba.sk

Abstrakt. *V dnešnej dobre prenikajú informačné technológie do všetkých odvetví ľudskej spoločnosti a to aj zdravotníctva. Zdravotnícky softvér sa stáva všadeprítomnou súčasťou zdravotníckych zariadení, pričom ide o veľmi špecifickú oblasť softvérových systémov. A to aj z pohľadu samotného softvérového riešenia, ako aj manažmentu rizík pri vývoji. Pri medicínskom softvéri je najväčšie riziko spojené s úmrtím pacienta alebo nevratným poškodením zdravia, a preto sa pri manažmente rizík siaha až po extrémnych opatreniach na to, aby sa tieto riziká eliminovali. V eseji porovnávam konkrétne postupy eliminácie rizika zlyhania v medicínskom softvéri s internetovým obchodom a bankovým systémom. Hodnotím vzájomnú aplikovateľnosť a zmysel riešení v týchto rôznych odvetviach.*

Kľúčové slová: *manažment rizík, vývoj softvéru, medicína, medicínsky softvér, zdravotnícke zariadenia*

Úvod

Riziko v softvérovom projekte je chápané ako neistota, možnosť vzniku určitej škody vyčíslenej peňažne. Manažment rizík sa snaží tieto škody minimalizovať, alebo ich aspoň udržať v akceptovateľnej miere a to tým, že identifikujeme, analyzujeme a eliminujeme riziká [4]. Preto sa naň vynakladá veľké množstvo finančných zdrojov.

Veľmi špecifickým druhom softvéru je medicínsky softvér, na ktorý sú kladené oveľa vyššie požiadavky na bezpečnosť a poruchovosť ako na bežné softvérové systémy.

Medicínsky softvér

Medicínsky softvér ide podľa mňa rozdeliť na základe možného rizika a celkovej funkcionality do troch kategórií:

- Softvér pre jednoduché diagnostické prístojne a prístroje pre ošetrenie
- Softvér pre systémy podpory života
- Informačné a expertné systémy

V každom type prevládajú charakteristické riziká spojené s ich funkcionalitou.

Pri jednoduchom medicínskom softvéri, teda diagnostické a ošetrovacie prístroje (napríklad stomatologické CAD/CAM systémy), väčšinou zlyhanie neohrozí pacientov život. Tento softvér je často algoritmický a využíva prvky umelej inteligencie.

Pri takomto druhu medicínskeho softvéru sa náklady na manažment rizík a elimináciu samotných rizík spotrebuje asi 20 až 30 % z celkových nákladov na projekt [2].

V eseji budem ďalej rozoberať najmä medicínske informačné systémy, pretože sa najviac podobajú svojim manažmentom rizík na bežné softvérové systémy.

Riziko zbytočnej smrti

Popri klasických rizikách pri vývoji softvéru sa pri medicínskom softvéri pridružujú riziká vo forme veľmi prísnej legislatívy, liekovej politiky, ktoré sa navyše ešte aj neustále mení. Ďalej softvér musí spĺňať interné smernice a zodpovedať špecifickým požiadavkám daného zdravotníckeho zariadenia. Najvážnejším rizikom pri prevádzke medicínskeho softvérového systému sú straty ľudských životov, ktoré majú nevyčísliteľnú cenu.

Pre tieto dôvody sa k manažmentu rizík pristupuje inak ako k manažmentu rizík pri klasickom softvéri. Opatrenia, ktoré používa, by sa dali prirovnať k opatreniam NASA alebo letovej prevádzky.

Porovnanie aplikovateľnosti odporúčaní

Odporúčania uvedené v článku od *Johna Burtona Software Risk Management for Medical Devices* [1], a podľa mňa oveľa konkrétnejšie odporúčania uvedené v odbornom článku *Software Risk Management for Medical Devices* od *Billa J. Wood* [2], som sa rozhodol porovnať s tým, ako sa rieši daný problém v bežných aplikáciách. Väčšina odporúčaní sa týka medicínskych informačných systémov, na ktoré som sa zameral. Riešenie rizika

porovnávam s dvomi druhmi bežne používaného softvéru, a to s internetovým obchodom – e-shopom a bankovým informačným systémom.

Všeobecné odporúčania

Medzi základné opatrenia patrí zahrnutie manažmentu rizík do všetkých častí životného cyklu vývoja softvéru, aj dôsledná analýza rizík už počas zbierania požiadaviek na softvérový systém a dôsledné testovanie spoľahlivosti a bezpečnosti.

Manažment rizík je veľmi dôležitou súčasťou vývoja akéhokoľvek softvérového systému. Preto je podľa mňa je vhodné ho využívať pri e-shopoch a smrteľne dôležité pri bankových systémoch. Takisto aj testovanie je vhodné použiť v oboch typoch projektov, najmä reverzné testovanie môže pomôcť predísť mnohým problémom.

Zahrnutie legislatívy do procesu vývoja

John Burton v článku *Software Risk Management for Medical Devices* [1] navrhol päť-úrovňový model manažmentu rizík zameraného práve na vývoj medicínskeho softvéru. Tento model by som prirovnal k vodopádovému modelu, teda na ďalšiu úroveň je možné prejsť až vtedy, keď je úplne splnená predchádzajúca úroveň. Hovorí len o tom, ako treba postupne zapracovávať legislatívu a normy do manažmentu rizík pri vývoji medicínskeho softvéru.

Podľa mňa je zapracovanie legislatívy a noriem do vývoja medicínskeho, ako aj klasického softvéru síce potrebné, ale absolútne nedostačujúce. Pretože sa spolieha na to, že legislatíva vyrieši všetky problémy a riziká sama.

Zapracovanie legislatívy do softvéru je nutné a potrebné aj pri bankových systémoch, ktoré takisto musia spĺňať v tomto ohľade prísne kritériá. Čo sa týka internetových obchodov, tiež musia spĺňať normy týkajúce sa finančného styku. Ale ani v bankových systémoch ani v e-shopoch nie je podľa mňa dobré diktovať formu softvérových procesov legislatívou.

Predvídateľné zlyhania

Medicínsky softvér by sa mal pri zlyhaní správať predvídateľne. Ja by som ešte toto odporúčanie doplnil tým, aby správanie sa softvéru pri zlyhaní bolo plne zdokumentované.

Tento prístup je podľa mňa vhodný pre softvérové projekty, ktoré obsahujú kritické oblasti, teda skôr bankové systémy. Kritické oblasti sú procesy v rámci softvérového systému, ktorých nepredvídateľné zlyhanie spôsobilo veľké finančné alebo dátové straty. Ak sa niečo pokazí pri e-shope, nepredstavuje to takmer žiaden problém. Zato v bankovom systéme je vhodné zavádzať predvídateľné správanie pri zlyhaní.

Testovanie použiteľnosti

Pri systémoch podpory života, informačných systémov a expertných systémoch je životne dôležité, aby nenastala chyba samotného softvéru, hardvéru a takisto aj personálu, ktorý systém obsluhuje. Zlyhanie alebo chyba systému môžu znamenať pre pacienta smrť, stačí predpísať iný liek, iné dávkovanie, alebo liek vôbec nepodať.

Veľké riziko vzniká aj pri zlom návrhu používateľského rozhrania a sledu obrazoviek, kde treba minimalizovať riziko zadávania nesprávnych údajov alebo omylu personálu. Preto je podľa mňa dôležité už v ranných štádiách prototypovať a testovať používateľské rozhrania priamo s nemocničným personálom.

Tento postup je podľa mňa vhodné uplatňovať aj v bežných softvérových projektoch, a je jedno, či ide o webovú alebo desktopovú aplikáciu. Určite by som prototypovanie a testovanie používateľského rozhrania odporúčal používať aj pri vytváraní e-shopov, kde sa momentálne používa vo veľmi malej miere. Pri bankových systémoch sa používa v menšej miere ako pri medicínskych, ale používa.

Ďalším dôvodom, prečo prototypovať používateľské rozhrania, je to, že zákazník vie lepšie definovať obrazovky ako samotnú funkcionalitu systému. Tým získame rýchlu spätnú väzbu, ktorá je potrebná pri agilnom vývoji.

Databázová bezpečnosť

Medicínske informačné systémy v sebe uchovávajú obrovské množstvo citlivých dát často chránených lekárskym tajomstvom, preto musia rátať s rizikom pokusov o odcudzenie týchto údajov. Pri systémoch podpory života aj medicínskych informačných systémoch tvoria náklady na manažment rizík až 85% celkovej ceny projektu [2], práve kvôli citlivosti týchto dát.

V tomto sa podobajú bankovým systémom, informačným systémom štátnych a silových zložiek. Preto aj pri akýchkoľvek systémoch uchovávajúcich citlivé dáta je potrebné dbať na bezpečnosť. V takýchto prípadoch sa často používajú konzervatívne modely vývoja. Do normálneho procesu vývoja by som z vývoja medicínskeho softvéru prebral aj masívne deštruktívne testovanie, ktoré by malo odhaliť bezpečnostné riziká.

Zato pri e-shopoch sú z veľkej časti uchovávané verejne prístupné dáta alebo dáta, ktorých zverejnenie nespôsobí žiadne škody. Preto nemá príliš význam investovať veľa času a úsilia na vytvorenie nadštandardnej bezpečnosti.

Údajová bezpečnosť

V medicínskych informačných systémoch sa uchováva veľké množstvo dát, ktorých strata alebo pozmenenie môže ohroziť život pacienta, napríklad strata údajov o alergii pacienta alebo o kontraindikácii v užívaných liekoch. Preto sa využíva pravidelné zálohovanie databázy, ktoré pracuje nezávisle od informačného systému. Ďalším prvkom zvyšujúcim údajovú bezpečnosť, v zmysle zabráneniu straty údajov, je použitie transakcií v databáze.

V e-shopoch nie je taká kritická strata údajov. Strata komentáru alebo príspevku na fóre je prípustná, jednoducho sa napíše znovu a neohrozia sa tým životy ľudí ani nespôsobia obrovské finančné škody. Zato pri bankových systémoch je údajová bezpečnosť a konzistencia databázy nesmierne dôležitá. Napríklad nedostatočné použitie databázových transakcií môže spôsobiť straty miliónov eur. V bankových systémoch podobne ako v medicínskych informačných systémoch sa používa mohutné zálohovanie.

Pri e-shopoch sa nepoužíva automatické zálohovanie databázy, ale podľa mňa by tento krok nestál veľa úsilia ani času, a pritom by pomohol predchádzať riziku straty dát.

Stromy porúch

Podľa *Billa J. Wooda* je vhodné vypracovávať aj takzvané stromy porúch, ktoré analyzujú závislosti a reťazenie možných porúch, tým lepšie predikovať riziká, ich rozsah a pravdepodobnosť. Strom porúch vyjadruje závislosti medzi poruchou, ktorá už hypoteticky nastala a poruchami, ktoré môžu nastať. Uzly v strome reprezentujú poruchy a hrany sú ohodnotené možným rizikom. Možný je aj opačný prístup. Začína sa poruchou, ktorá hypoteticky nastala a uzly sa v strome rozvíjajú k poruchám a príčinám, ktoré prvotnej poruche predchádzali.

Podľa mňa by takéto stromy porúch šlo použiť aj v iných softvérových projektoch, pretože dokážu efektívne vyhodnocovať riziká, a súčasne napomáhajú k tomu, aby sa identifikovali opatrenia na predchádzanie rizík.

Kontrola hardvéru

V medicínskych systémoch podpory života sa odporúča zabezpečiť vzájomnú kontrolu medzi hardvérom a softvérom a podobné opatrenia, ktoré zabránia ešte väčším škodám pri zlyhaní softvéru.

Takéto extrémne opatrenia sú charakteristické iba pre medicínsky softvér. Nemyslím si, že je vhodné ich prenášať do bankových systémov a vôbec nie do internetových obchodov.

Vyčerpanie výpočtových zdrojov

Dosť málo spomínaným rizikom, priamo ohrozujúcim život pacienta, je vyčerpanie výpočtových zdrojov a tým ohroziť systém podpory života. Toto riziko sa najčastejšie rieši predimenzovaním výpočtových prostriedkov a vertikálnou škálovateľnosťou.

Takéto riziko nie je potrebné riešiť pri internetových obchodoch. Pri bankových informačných systémoch sa toto riziko rieši podobne, a to vertikálnym škálovaním serverov poskytujúce biznis služby.

Duplikovanie komponentov

Podľa mňa sa v ňom dá obmedziť riziko zlyhania, spôsobené zmenou alebo úpravou softvérového komponentu, tým, že pristúpime k výraznej redundancii softvérových komponentov. A to tak, keď dva rôzne nezávislé komponenty A,B , ktoré využívajú komponent C , tak sa komponent C naduplikuje, komponent A bude využívať komponent C a komponent B bude využívať komponent C‘ . Teda pri zmene komponentu C nebude ohrozené správne fungovanie komponentu B.

Tento prístup je absolútne nevhodný pre bežné softvérové systémy, pretože ide v podstate o antipatern *Duplicitný kód*. Preto ja vhodný iba pre systémy podpory života, a aj to vo veľmi malej miere.

Eliminácia logických chýb v programoch

Logické chyby v programoch môžu byť problémom, keď ich nezachytí testovanie. Takáto situácia môže nastať najmä, ak sa jedná o zložité algoritmy. Podľa mňa je istým riešením použiť spôsob programovania, kde jednotlivé moduly sa môžu tvoriť v odlišných

jazykoch. Jazyk pre modul by sa volil podľa požiadaviek na systém a vhodnosti daného jazyka. Už len toto môže znížiť chyby v jednotlivých komponentoch, pretože zvýši komfort programovania.

Ďalším prístupom, ktorý by mohol znížiť počet logických chýb ja podľa mňa použitie prísne typového funkcionálneho jazyka. Funkcionálne jazyky sú vhodné na písanie aplikácii, ktoré obsahujú veľa zložitých matematických výpočtov a algoritmov. Takéto jazyky eliminujú logické chyby tým, že nútia programátora robiť malé funkcie bez vedľajších efektov. Napríklad veľké banky často využívajú funkcionálne jazyky pre svoje systémy a z tohto prístupu by sa mohli poučiť aj vývojári medicínskeho softvéru. Podľa mňa by im to prinieslo spomínané výhody a znížilo riziko výskytu logických chýb v programe.

Pri vývoji e-shopov nie je potrebné používať funkcionálne programovanie kvôli zvýšeniu bezpečnosti kódu. No jestvujú webové aplikácie, ktoré sú naprogramované deklaratívne.

Záver

Zdravotnícky softvér patrí v dnešnej dobe medzi neodmysliteľnú súčasť zdravotníckych zariadení a pomáha zachraňovať denne mnoho ľudských životov. Pri jeho vývoji a prevádzke treba zabezpečiť, aby nerobil presne opačnú činnosť, teda spôsoboval úmrtia pacientov. Práve úmrtie považujem za najväčšie riziko spojené s týmto typom softvéru a preto predchádzať zbytočným úmrtiam v zdravotníctve je potrebné za každú cenu, vrátane netradičných postupov v manažmente rizík softvéru.

V eseji som posudzoval aplikovateľnosť odporúčaní manažmentu rizík medicínskeho softvéru na iné typy softvérových systémov. Niektoré odporúčania týkajúce sa údajovej, databázovej bezpečnosti a vývoja používateľského rozhrania považujem za vhodné použiť v bankových systémoch a aj v internetových obchodoch, zatiaľ čo odporúčania týkajúce sa legislatívy a kontroly hardvéru sú až príliš špecifické pre medicínsku doménu. Navrhol som použitie prísne typových funkcionálnych jazykov na zníženie počtu logických chýb v kritických častiach softvérových systémov.

Použitá literatúra

1. Burton John: *Software Risk Management for Medical Devices*, ACM New York, USA 2006, ISBN:1-59593-399-9
2. Bill J. Wood: *Software Risk Management for Medical Devices*, An MD&DI January 1999 Column, 1999
3. Kuracina R. Ing., Ferjenčík M. Phd.: *Nástroje pre oceňovanie rizika a vyšetrovanie havárií*, Recenzovaný zborník, Aktuálne otázky bezpečnosti práce, ISBN 80-8073-649-9, 25.-27.10.2006 Stará Lesná,
4. Barry W. Boehm: *Software risk management, Principles and Practices*, IEEE Software, Vol. 8, 1991.

Annotation

Specifics of risk management for medical software

Nowadays, the Information Technology penetrates into all sectors of human society and even healtcare. In the medical softwarethere is the greatest a ubiquitous part of the medical devices with a very specific area of software systems. And even in terms of the actual software solution as well as in the development of risk management. In medical software is the greatest risk associated with patient deaths or irreversible health damage and therefore the risk management goes to extreme measures to eliminate these risks. In this essay, I compare the specific procedures to eliminate the risk of failure in the medical software with e-shop and the banking system. Are mutual applicability and meaning of these solutions in various industries.