УДК 004.891.3: 004.3

МЕТОД УПРАВЛІННЯ РИЗИКАМИ ПРИ РОЗРОБЛЕННІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

М.В.ОСТРОВИЙ, М.В.КРАСОВСЬКИЙ, В.І.ГРИБИНЧУК
(Науковий керівник – Т.О.Говорущенко)
Хмельницький національний університет

У статті запропоновано метод управління ризиками при розробленні ПЗ, який дає можливість ідентифікувати джерела ризиків та можливі ризики для будь-якого програмного проекту, а також оцінити ризики, визначити їх пріоритет та заходи із зменшення або усунення ризиків. Крім цього, метод дозволяє провести оцінку ризиків після застосування обраних заходів із зменшення або усунення ризиків, що робить можливим підібрати найкращий захід для максимального зменшення величини кожного ризику

The paper proposes a method for risk management in the software developement, that provides: the identification of the risks sources and the risks for any software project; the assessment of risks; their prioritization; the identification of events for reduction or elimination of risks. In addition, the method allows the repeat assessment of risks after the using of selected events for reduction or elimination of risks, that provides the selection of the best events for minimization of the value of each risk.

Ключові слова: програмне забезпечення (ПЗ), ризики при розробленні ПЗ, ймовірність ризику, збиток від настання ризикової події, величина ризику, модель SEI управління ризиками

Вступ. Аналітичні дослідження та огляди, які виконували протягом кількох останніх років провідні зарубіжні аналітики, давали не дуже обнадійливі результати в галузі розроблення програмного забезпечення (ПЗ). Незважаючи на бурхливий розвиток технологій програмування, залишається значна частина програмних проектів, які не можна вважати цілком успішними. Під успішністю реалізації програмного проекту надалі будемо розуміти вчасне виконання програмного проекту в рамках виділеного бюджету та з реалізацією всіх необхідних можливостей та функцій [1] – рис.1.

Статистика успішності реалізації програмних проектів за 1994-2012 роки, представлена у The Standish Group International (CHAOS report) [1], дала можливість побачити приріст кількості успішних проектів та спад кількості провальних проектів у 2010-2012 роках, але в той же час частка проблемних проектів (які мали перевищення термінів, перевитрати або не мали необхідних можливостей та функцій) є досить сталою величиною у 2006-2012 роках і складає 42-46% проектів. Статистика по перевитратах, перевищенню термінів та відсутності необхідної функціональності для проблемних програмних проектів у 2004-2012 роках, за даними [1], представлена на рис.2.

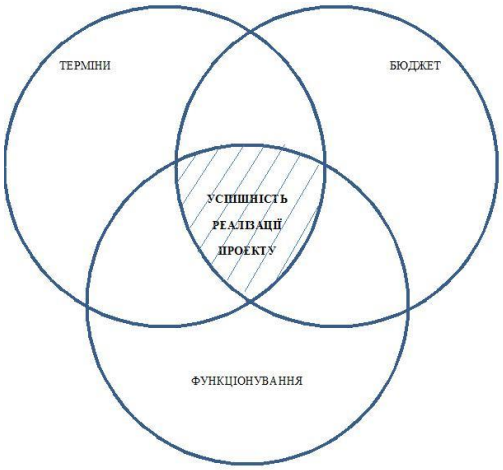

Рис. 1 Візуалізація поняття «успішність реалізації програмного проекту»

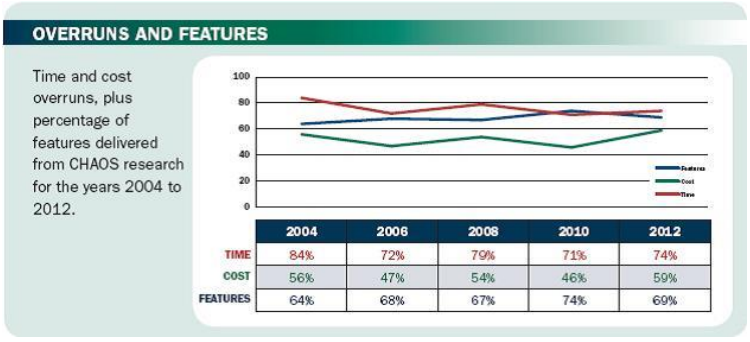

Рис. 2 Статистика по перевитратах, перевищенню термінів та відсутності необхідної функціональності для проблемних програмних проектів

Статистика [1] показує також,що лише 16% програмних проектів успішно завершуються середніми компаніями у визначені термін та бюджет. Ситуація з великими компаніями значно гірша – тільки 9% проектів вкладаються у визначені терміни та бюджет. Проекти, виконані найбільшими американськими компаніями, мають приблизно 42% функціональності, запропонованої на початкових етапах. Менші компанії справляються краще: 78.4% проектів реалізовують 74.2% своєї запланованої функціональності.

Дослідження міжнародної компанії McKinsey & Company [2] спільно з University of Oxford, проведені у 2010 році, також показали, що половина масштабних програмних проектів з загальним бюджетом понад 15 мільйонів доларів значно перевищують заплановані витрати, зокрема: середнє перевищення витрат на проект складає 66%, середнє перевищення часу реалізації проекту - 33%, а середнє число втрат прибутку - 17% (рис.3, 4 [2]).

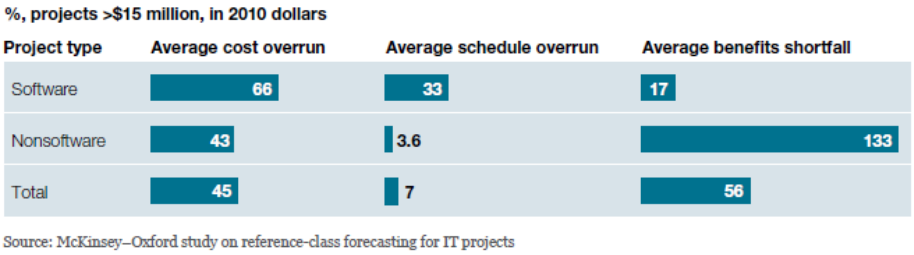

Рис. 3 Продуктивність програмних проектів різних типів

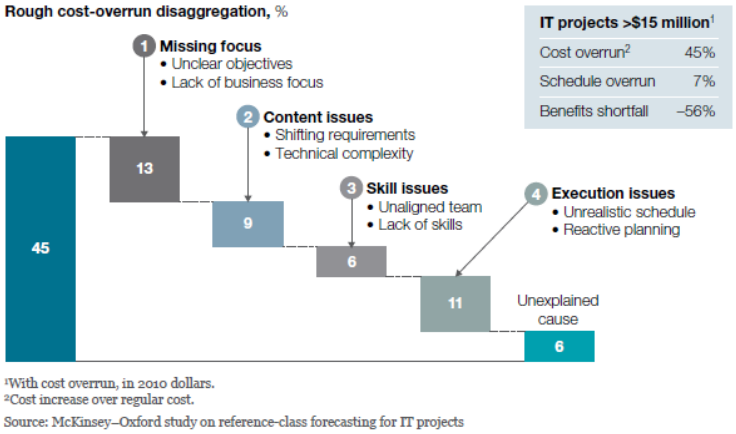

Рис. 4 Чотири групи питань, які викликають більшість невдач програмних проектів

Існує чимало прикладів значних фінансових втрат через невдале управління ризиками під час роботи над програмним проектом [3]: 1) відмова на етапі розроблення від програмного проекту Virtual Case File (VCF), розроблюваного федеральним бюро розслідувань США у 2000 -2005 роках, коли загальна вартість витрат на проект вже досягла близько 170 мільйонів доларів США; 2) відмова від програмного проекту Міністерства оборони США під назвою Enterprise Resource Planning після семи років роботи над проектом у 2005 році, коли на нього було витрачено 1 мільярд доларів США; 3) невдача програмного проекту уряду Великобританії у 2011 році, метою якого було створення електронних медичних записів громадян, з загальними витратами у 18,7 мільярдів доларів США.

Отже, розроблення ПЗ не завжди завершується успішно і часто пов'язане із ризиками недостатньої функційності ПЗ, невиконання термінів виконання проекту або перевищення бюджету.. Тому *актуальною задачею* наразі є покращення управління ризиками при розробленні ПЗ. Ризики характеризують можливі негативні наслідки або збитки при функціонуванні ПЗ. Задача розробників зводиться до скорочення та ліквідації ризиків. Зниження ризиків проекту сприяє підвищенню його якості, ефективності та результативності.

Постановка задачі. З результатів аналізу сучасного стану галузі розроблення ПЗ слідує, що перспективним напрямком дослідження є розроблення математичного методу управління ризиками при розробленні ПЗ.

1. Управління ризиками. *Ризиками* називають негативні події ймовірнісного характеру, які негативно впливають на результат проекту, відображають втрати або збитки від процесів або продуктів, викликані дефектами при проектуванні вимог, недоліками обгрунтування проектів ПЗ, а також при наступних етапах розроблення, реалізації та всього життєвого циклу ПЗ [4-7]. Важливо розуміти, що ризик – це ймовірна подія, яка може відбутись, а може й не відбутись.

Для успішної реалізації програмних проектів однією з основ управління проектом є управління ризиками, яке охоплює весь життєвий цикл ПЗ. *Управління ризиками* – це процес прийняття та виконання управлінських рішень, спрямованих на зниження ймовірності виникнення несприятливого результату та мінімізацію можливих втрат, викликаних його реалізацією; це систематичні процеси, пов'язані з ідентифікацією, аналізом та прийняттям рішень, які забезпечують мінімізацію негативних наслідків настання ризикових подій, а також максимізацію ймовірності та наслідків настання позитивних подій [4-7]. Управління ризиками включає повне розуміння внутрішніх та зовнішніх причин, які впливають на проект і можуть призвести до його невдачі. Аналіз ризиків виконується після формування плану проекту. Головною метою управління ризиками є ідентифікація та контроль за факторами, що рідко зустрічаються і призводять до варіацій проекту.

Існують різні моделі управління ризиками [4-7], найбільш використовуваною серед яких є модель Інституту програмної інженерії (SEI), що включає в себе як вимоги стандартів [8, 9], так і відомі «кращі

практики» управління ризиками. Модель SEI представлена у вигляді текстових рекомендацій та плану, формалізований метод управління ризиками відсутній, що призводить до вільного використання та трактування наведеної моделі, тому основною задачею авторів є деталізація та формалізація методу управління ризиками при розробленні програмного забезпечення.

Метод управління ризиками при розробленні програмного забезпечення. Деталізуємо та формалізуємо етапи методу управління ризиками при розробленні ПЗ:

1) Ідентифікація ризиків:

1.1) Визначення можливих джерел ризиків. Згідно моделі SEI, є 18 найбільш поширених джерел ризиків, тоді представимо можливі джерела ризиків у вигляді наступної множини:

$RS = \{rs_1,...,rs_{18}\}$, де rs_i - можливі джерела ризиків ($i = \overline{1..18}$), а саме:

$TRS = \{rs_1,...,rs_7\}$, $TRS \subset RS$ - підмножина джерел технічних ризиків, де: rs_1 - функційні характеристики, rs_2 - характеристики якості , rs_3 - характеристики надійності, rs_4 - застосовність, rs_5 - часова продуктивність, rs_6 - супроводжуваність, rs_7 - повторне використання компонентів;

$CRS = \{rs_8,...,rs_{10}\}$, $CRS \subset RS$ - підмножина джерел вартісних ризиків, де: rs_8 - обмеження сумарного бюджету, rs_9 - недоступна вартість проекту, rs_{10} - низький ступінь реалізму при оцінюванні витрат на проект;

$PRS = \{rs_{11},...,rs_{13}\}$, $PRS \subset RS$ - підмножина джерел планових ризиків, де: rs_{11} - властивості та можливості гнучкості зміни планів, rs_{12} - можливості порушення встановлених термінів етапів життєвого циклу, rs_{13} - низький ступінь реалізму планів та етапів життєвого циклу;

$MRS = \{rs_{14},...,rs_{18}\}$, $MRS \subset RS$ - підмножина джерел ризиків процесів та процедур управління проектом, де: rs_{14} - стратегія проекту, rs_{15} - планування проекту, rs_{16} - оцінювання проекту, rs_{17} - документування проекту, rs_{18} - прогнозування проекту;

Аналіз специфікації вимог до ПЗ (SRS) дозволяє визначити можливі джерела ризиків. Введемо наступні правила визначення джерел ризиків:

1. якщо у SRS відсутні або наявні нереалістичні чи неоціненні функційні характеристики, то $rs_1 = 1$, інакше $rs_1 = 0$;
2. якщо відсутні або існують нереалістичні чи неоціненні характеристики якості, то $rs_2 = 1$, інакше $rs_2 = 0$;
3. якщо відсутні або наявні нереалістичні чи неоціненні характеристики надійності, то $rs_3 = 1$, інакше $rs_3 = 0$;
4. якщо у SRS відсутні рекомендації щодо майбутньої застосовності ПЗ, то $rs_4 = 1$, інакше $rs_4 = 0$;
5. якщо відсутні або існують нереалістичні чи неоціненні характеристики часової продуктивності, то $rs_5 = 1$, інакше $rs_5 = 0$;
6. якщо у SRS відсутні рекомендації щодо майбутньої супроводжуваності ПЗ, то $rs_6 = 1$, інакше $rs_6 = 0$;
7. якщо відсутні або існують нереалістичні чи неоціненні пропозиції щодо повторного використання компонентів, то $rs_7 = 1$, інакше $rs_7 = 0$;
8. якщо у SRS присутні обмеження сумарного бюджету, то $rs_8 = 1$, інакше $rs_8 = 0$;
9. якщо у SRS вказано недоступну вартість проекту, то $rs_9 = 1$, інакше $rs_9 = 0$;
10. якщо у SRS присутній низький ступінь реалізму при оцінюванні витрат на проект, то $rs_{10} = 1$, інакше $rs_{10} = 0$;
11. якщо відсутні або наявні нереалістичні чи неоціненні властивості та можливості гнучкості зміни планів, то $rs_{11} = 1$, інакше $rs_{11} = 0$;
12. якщо відсутні або наявні нереалістичні чи неоціненні можливості порушення встановлених термінів етапів життєвого циклу, то $rs_{12} = 1$, інакше $rs_{12} = 0$;
13. якщо у SRS присутній низький ступінь реалізму планів та етапів життєвого циклу етапів життєвого циклу, то $rs_{13} = 1$, інакше $rs_{13} = 0$;
14. якщо відсутня або наявна нереалістична чи неоціненна стратегія проекту, то $rs_{14} = 1$, інакше $rs_{14} = 0$;
15. якщо відсутнє або наявне нереалістичне чи неоціненне планування проекту, то $rs_{15} = 1$, інакше $rs_{15} = 0$;

16. якщо відсутнє або наявне нереалістичне чи неоціненне оцінювання проекту, то $rs_{16}=1$, інакше $rs_{16}=0$;
17. якщо відсутнє документування проекту, то $rs_{17}=1$, інакше $rs_{17}=0$;
18. якщо відсутнє або наявне нереалістичне чи неоціненне прогнозування проекту, то $rs_{18}=1$, інакше $rs_{18}=0$;
19. якщо $(rs_1=1)\cup(rs_2=1)\cup(rs_3=1)\cup(rs_4=1)\cup(rs_5=1)\cup(rs_6=1)\cup(rs_7=1)$, то мають місце технічні ризики;
20. якщо $(rs_8=1)\cup(rs_9=1)\cup(rs_{10}=1)$, то мають місце вартісні ризики;
21. якщо $(rs_{11}=1)\cup(rs_{12}=1)\cup(rs_{13}=1)$, то мають місце планові ризики;
22. якщо $(rs_{14}=1)\cup(rs_{15}=1)\cup(rs_{16}=1)\cup(rs_{17}=1)\cup(rs_{18}=1)$, то мають місце ризики процесів та процедур управління проектом.

1.2) Ідентифікація потенційних ризикових подій. Ідентифікація ризику полягає в фіксації всіх факторів занепокоєння та стурбованості, пов'язаних із проектом, а потім в постійному обмірковуванні інших можливих побоювань. Справжньою проблемою є ризики, які не вдалось ідентифікувати. На основі провідних галузевих публікацій [4-7] сформовано множини потенційних ризиків за вищезазначеними групами:

$TR=\{tr_1,...,tr_{11}\}$ - множина технічних ризиків, де: tr_1 - затримки у постачанні обладнання, необхідного для процесу створення ПЗ; tr_2 - затримки у постачанні програмних засобів, необхідних для підтримки процесу створення ПЗ; tr_3 - небажання розробників використовувати програмні засоби підтримки життєвого циклу; tr_4 - відмова від CASE-засобів; tr_5 - запити на більш потужні інструментальні засоби розроблення; tr_6 - недостатня продуктивність баз(и) даних; tr_7 - програмні компоненти, які використовуються повторно, мають дефекти та обмежені функційні можливості; tr_8 - неефективність програмного коду, генерованого CASE-засобами; tr_9 - неможливість інтеграції CASE-засобів з іншими засобами підтримки проекту; tr_{10} - швидкість виявлення дефектів в системі нижче раніше запланованої; tr_{11} - дефектні системні компоненти;

$CR=\{cr_1,...,cr_6\}$ - множина вартісних ризиків, де: cr_1 - недооцінки витрат на виконання проекту (надмірно низька вартість); cr_2 - переоцінки витрат на виконання проекту (надмірно висока вартість); cr_3 - фінансові ускладнення у компанії-розробника; cr_4 - зменшення бюджету проекта під час його виконання; cr_5 - висока вартість повторних робіт, необхідних через зміну вимог; cr_6 - реорганізація компанії-розробника;

$PR=\{pr_1,...,pr_{10}\}$ - множина планових ризиків, де: pr_1 - зміни графіку робіт; pr_2 - порушення графіку робіт; pr_3 - необхідність зміни багатьох вимог; pr_4 - необхідність великої кількості повторних робіт; pr_5 - недооцінки часу виконання проекту; pr_6 - переоцінки часу виконання проекту; pr_7 - розмір ПЗ перевищує планований розмір; pr_8 - розмір ПЗ значно менший за планований розмір; pr_9 - поява на ринку аналогічного ПЗ до виходу розроблюваного ПЗ; pr_{10} - поява на ринку більш конкурентоздатного ПЗ;

$MR=\{mr_1,...,mr_{16}\}$ - множина ризиків процесів та процедур управління проектом, де: mr_1 - низький моральний стан персоналу; mr_2 - низька взаємодія між членами колективу розробників; mr_3 - пасивність керівника (менеджера) проекту; mr_4 - недостатня компетентність керівника (менеджера) проекту; mr_5 - незадоволеність замовника; mr_6 - недостатня кількість фахівців з необхідним професійним рівнем; mr_7 - хвороба провідного розробника в найкритичніший час; mr_8 - одночасна хвороба декількох розробників; mr_9 - неможливість організації необхідного навчання персоналу; mr_{10} - зміна пріоритетів в управлінні проектом; mr_{11} - недооцінка необхідної кількості розробників; mr_{12} - переоцінка необхідної кількості розробників; mr_{13} - надмірне документування проекту; mr_{14} - недостатнє документування проекту; mr_{15} - нереалістичне прогнозування результатів проекту; mr_{16} - недостатній професійний рівень розробників.

Тоді множина (список) потенційних ризиків має вигляд: $R=\{r_1,...,r_{43}\}$, де $(r_1,...,r_{11})\in TR$, $(r_{12},...,r_{17})\in CR$, $(r_{18},...,r_{27})\in PR$, $(r_{28},...,r_{43})\in MR$, а, в свою чергу, $TR\subset R$, $CR\subset R$, $PR\subset R$, $MR\subset R$.

Множина (список) потенційних ризиків конкретного програмного проекту має вигляд: $R_p=\{r_{1p},...,r_{43p}\}$.

Введемо наступні правила визначення ризиків для конкретного програмного проекту:

1. якщо можливі затримки у постачанні обладнання, необхідного для процесу створення ПЗ, то $r_{1p}=1$, інакше $r_{1p}=0$;
2. якщо можливі затримки у постачанні програмних засобів, необхідних для підтримки процесу створення ПЗ, то $r_{2p}=1$, інакше $r_{2p}=0$;

…

43. якщо у проектний колектив входять розробники з недостатнім професійним рівнем, то $r_{43p}=1$, інакше $r_{43p}=0$.

Сформуємо множину ризиків конкретного програмного проекту $ER_p=\{er_{ip}\}$, де $i=\overline{1..n}$, n - кількість ризиків проекту, за наступними правилами:

1. якщо $r_{1p}=1$, то: $er_{ip}=1$, $i=i+1$;
2. якщо $r_{2p}=1$, то: $er_{ip}=1$, $i=i+1$;

…

43. якщо $r_{43p}=1$, то: $er_{ip}=1$, $i=i+1$.

2) Аналіз ризиків:

2.1) Визначення ймовірності ризику (ймовірність настання ризикової події). Для кожного ризику з множини ER_p колектив розробників повинен встановити ймовірність його настання в діапазоні $[0;1]$. Множина ймовірностей ризиків має вигляд: $PRER_p=\{prer_{ip}\}$, де $i=\overline{1..n}$, n - кількість ризиків проекту.

Згідно провідних галузевих публікацій [4-7], сформуємо правила класифікації ризиків за їх ймовірностями:

1. $\forall(prer_{ip}\in PRER_p),\ \forall(er_{ip}\in ER_p)$: якщо $prer_{ip}<0{,}1$, то ймовірність ризику er_{ip} є дуже низькою;
2. $\forall(prer_{ip}\in PRER_p),\ \forall(er_{ip}\in ER_p)$: якщо $(prer_{ip}\geq 0{,}1)\wedge(prer_{ip}<0{,}25)$, то ймовірність ризику er_{ip} є низькою;
3. $\forall(prer_{ip}\in PRER_p),\ \forall(er_{ip}\in ER_p)$: якщо $(prer_{ip}\geq 0{,}25)\wedge(prer_{ip}<0{,}5)$, то ймовірність ризику er_{ip} є середньою;
4. $\forall(prer_{ip}\in PRER_p),\ \forall(er_{ip}\in ER_p)$: якщо $(prer_{ip}\geq 0{,}5)\wedge(prer_{ip}<0{,}75)$, то ймовірність ризику er_{ip} є високою;
5. $\forall(prer_{ip}\in PRER_p),\ \forall(er_{ip}\in ER_p)$: якщо $(prer_{ip}\geq 0{,}75)$, то ймовірність ризику er_{ip} є дуже високою.

2.2) Визначення можливих збитків від ризику (на скільки постраждає проект, якщо ризикова подія відбудеться). Для кожного ризику з множини ER_p колектив розробників повинен встановити розмір можливих збитків від його настання - в діапазоні $[0;1]$. Множина збитків ризиків має вигляд: $LRER_p=\{lrer_{ip}\}$, де $i=\overline{1..n}$, n - кількість ризиків проекту.

2.3) Визначення величини ризику (математичне сподівання збитку). Для кожного ризику з множини ER_p визначимо його величину. Множина величин ризиків має вигляд: $VRER_p=\{vrer_{ip}\}$, де $i=\overline{1..n}$, n - кількість ризиків проекту, $vrer_{ip}=prer_{ip}\cdot lrer_{ip}$.

2.4) Встановлення рівня пріоритету та ранжування ризиків за пріоритетами. Для встановлення рівня пріоритету та ранжування ризиків знайдемо максимальний та мінімальний елемент множини $VRER_p$. Максимальний елемент – це елемент $\max\in VRER_p$, для якого справедливе: $\forall vrer_{ip}\in VRER_p:(\max<vrer_{ip}\Rightarrow\max=vrer_{ip})$, де $i=\overline{1..n}$, n - кількість ризиків проекту. Мінімальний елемент – це елемент $\min\in VRER_p$, для якого справедливе: $\forall vrer_{ip}\in VRER_p:(\min>vrer_{ip}\Rightarrow\min=vrer_{ip})$, де $i=\overline{1..n}$, n - кількість ризиків проекту.

Розіб'ємо далі отриманий інтервал $[\min;\max]$ на три інтервали: $[\min;\min+\frac{\max-\min}{3})$, $[\min+\frac{\max-\min}{3};\min+2\cdot\frac{\max-\min}{3})$, $[\min+2\cdot\frac{\max-\min}{3};\max]$. Тоді правила для встановлення рівня пріоритету ризиків мають наступний вигляд:

1. $\forall(vrer_{i_p}\in VRER_p),\ \forall(er_{i_p}\in ER_p)$: якщо $(vrer_{i_p}\geq\min)\wedge(vrer_{i_p}<(\min+\frac{\max-\min}{3}))$, то: рівень пріоритету ризику er_{i_p} - низький, $ler_{j_p}=er_{i_p}$ ($ler_{j_p}\in LER_p$, $j=\overline{1..m}$, $LER_p=\{ler_{1_p},...,ler_{m_p}\}$ - множина ризиків з низьким пріоритетом, m - кількість ризиків з низьким пріоритетом), $j=j+1$;
2. $\forall(vrer_{i_p}\in VRER_p),\ \forall(er_{i_p}\in ER_p)$: якщо $(vrer_{i_p}\geq(\min+\frac{\max-\min}{3}))\wedge$ $\wedge(vrer_{i_p}<(\min+2\cdot\frac{\max-\min}{3}))$, то: рівень пріоритету ризику er_{i_p} - середній, $mer_{k_p}=er_{i_p}$ ($mer_{k_p}\in MER_p$, $k=\overline{1..v}$, $MER_p=\{mer_{1_p},...,mer_{v_p}\}$ - множина ризиків з середнім пріоритетом, v - кількість ризиків з середнім пріоритетом), $k=k+1$;
3. $\forall(vrer_{i_p}\in VRER_p),\ \forall(er_{i_p}\in ER_p)$: якщо $(vrer_{i_p}\geq(\min+2\cdot\frac{\max-\min}{3}))\wedge(vrer_{i_p}\leq\max)$, то: рівень пріоритету ризику er_{i_p} - високий, $her_{g_p}=er_{i_p}$ ($her_{g_p}\in HER_p$, $g=\overline{1..o}$, $HER_p=\{her_{1_p},...,her_{o_p}\}$ - множина ризиків з високим пріоритетом, o - кількість ризиків з високим пріоритетом), $g=g+1$.

В результаті застосування вищенаведених правил для встановлення рівня пріоритету ризиків до всіх ризиків проекту маємо множину HER_p першочергових ризиків, множину MER_p другочергових ризиків та множину LER_p ризиків останньої, третьої, черги конкретного програмного проекту, які пропонуються членам проектного колективу в якості допомоги при обранні заходів із зменшення або усунення ризиків.

3) Планування ризиків:

3.1) Заходи із зменшення або усунення ризику. На основі провідних галузевих публікацій [4-7] сформуємо множину потенційних заходів із зменшення або усунення ризику: $EV=\{ev_1,...,ev_{19}\}$, де ev_1 - попереднє навчання членів проектного колективу; ev_2 - узгодження детального переліку вимог із замовником; ev_3 - включення узгодженого списку вимог замовника в договір; ev_4 - точне слідування вимогам замовника з узгодженого списку вимог; ev_5 - попередні дослідження ринку; ev_6 - експертна оцінка проекту досвідченим стороннім консультантом; ev_7 - консультації досвідченого стороннього консультанта; ev_8 - тренінг з вивчення необхідних інструментів розроблення; ev_9 - укладання договору страхування; ev_{10} - використання «шаблонних» рішень з вдалих попередніх проектів при управлінні проектом; ev_{11} - підготовка документів, які показують важливість даного проекту для досягнення фінансових цілей компанії-розробника; ev_{12} - реорганізація роботи проектного колективу таким чином, щоб обов'язки та робота членів колективу перекривали один одного; ev_{13} - придбання (замовлення) частини компонентів розроблюваного ПЗ; ev_{14} - заміна потенційно дефектних компонентів розроблюваного ПЗ придбаними компонентами, які гарантують якість роботи; ev_{15} - придбання більш продуктивної бази даних; ev_{16} - використання генератора програмного коду; ev_{17} - реорганізація роботи проектного колективу в залежності від рівня складності задач та професійних рівнів розробників; ev_{18} - повторне використання підходящих компонентів ПЗ, які були розроблені для інших проектів; ev_{19} - аналіз доцільності створення даного ПЗ.

Введемо наступні правила визначення заходів із зменшення або усунення ризиків конкретного програмного проекту (передбачається один, найбільш підходящий, захід для кожного ризику!):

1. $\forall(er_{i_p}\in ER_p)$: якщо ризик er_{i_p} може бути зменшений або усунутий за допомогою заходу $ev_1\in EV$, то $ever_{i_p}=ev_1$;

2. $\forall(er_{i_p} \in ER_p)$: якщо ризик er_{i_p} може бути зменшений або усунутий за допомогою заходу $ev_2 \in EV$, то $ever_{i_p} = ev_2$; …

19. $\forall(er_{i_p} \in ER_p)$: якщо ризик er_{i_p} може бути зменшений або усунутий за допомогою заходу $ev_{19} \in EV$, то $ever_{i_p} = ev_{19}$.

Тоді в результаті застосування вищенаведених правил визначення заходів із зменшення або усунення ризиків матимемо множину заходів для конкретного програмного проекту: $EVER_p = \{ever_{i_p}\}$, де $i = \overline{1..n}$, n - кількість ризиків проекту.

4) Моніторинг ризиків:

4.1) Оцінки ризиків. Дуже важливо відзначити, що всі величини, пов'язані із ризиком, не є постійними в проекті. Ймовірність ризикової події та можливі збитки можуть збільшуватись та зменшуватись в результаті застосування заходів із зменшення або усунення ризиків. Тому потрібні оцінки ймовірності, збитку та величини ризику після застосування таких заходів. Для кожного ризику з множини ER_p колектив розробників повинен встановити ймовірність (в діапазоні $[0;1]$) його настання після застосування обраного заходу із зменшення або усунення ризиків. Множина ймовірностей ризиків після застосування заходів має вигляд: $EPRER_p = \{eprer_{i_p}\}$, де $i = \overline{1..n}$, n - кількість ризиків проекту. Для кожного ризику з множини ER_p колектив розробників повинен встановити розмір можливих збитків (в діапазоні $[0;1]$) від його настання після застосування обраного заходу із зменшення або усунення ризиків. Множина збитків ризиків після застосування заходів має вигляд: $ELRER_p = \{elrer_{i_p}\}$, де $i = \overline{1..n}$, n - кількість ризиків проекту. Для кожного ризику з множини ER_p визначимо його величину після застосування обраного заходу із зменшення або усунення ризиків. Множина величин ризиків після заходів має вигляд: $EVRER_p = \{evrer_{i_p}\}$, де $i = \overline{1..n}$, n - кількість ризиків проекту, $evrer_{i_p} = eprer_{i_p} \cdot elrer_{i_p}$. Можливе повторне встановлення рівня пріоритету ризиків після застосування заходів із зменшення або усунення ризиків – за правилами етапу 6.

Висновки. Результат будь-якого проекту залежить від кількості та величини ризиків недостатньої функційності ПЗ, невиконання термінів виконання проекту, перевищення бюджету. Тому управління ризиками повинне бути однією з основ управління проектами. Авторами запропоновано метод управління ризиками при розробленні ПЗ, який дає можливість ідентифікувати джерела ризиків та можливі ризики для будь-якого програмного проекту, а також оцінити ризики, визначити їх пріоритет та заходи із зменшення або усунення ризиків. Крім цього, метод дозволяє провести оцінку ризиків після застосування обраних заходів із зменшення або усунення ризиків, що робить можливим підібрати найкращий захід для максимального зменшення величини кожного ризику. Представлений метод забезпечує процес управління ризиками математичним підґрунтям, внаслідок чого зменшується складність та зростає ефективність управління ризиками. Перспективою для подальших досліджень авторів є розроблення системи управління ризиками при розробленні ПЗ, в основу якої буде покладено запропонований у статті метод.

Перелік посилань:

1. CHAOS Manifesto: Think big, act small, 2013 // [Electronic resource] – Access mode: http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf
2. M.Bloch. Delivering large-scale IT projects on time, on budget, and on value / M.Bloch, S.Blumberg, J.Laartz – October, 2012 // [Electronic resource] – Access mode: http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value
3. П.Алферов. Роль бизнес-заказчика в ИТ-проекте // Управление проектами, №4, 2008 // [Електронний ресурс] – Режим доступу: http://www.pmmagazine.ru/document.asp?ob_no=771
4. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами: достижение оптимального качества при минимуме затрат.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1136 с.
5. Эрик Дж. Брауде. Технология разработки программного обеспечения. – СПб: Питер, 2004. – 655 с.
6. С.Зыль. Проектирование, разработка и анализ програмного обеспечения систем реального времени – СПб: БХВ-Петербург, 2010. – 336 с.
7. И.Соммервилл. Инженерия программного обеспечения. - Издательский дом “Вильямс”, 2002 – 624 с.
8. ISO/IEC 12207:2008. Systems and software engineering – Software life cycle processes
9. ISO/IEC 33001:2015. Information technology – Process assessment

Рецензенти: Гнатчук Є.Г., к.т.н., доц. каф. СПр ХНУ; Тітова В.Ю., к.т.н., доц. каф. СПр ХНУ