

ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНИХ ПРОДУКТІВ

Т.О.Говорущенко. Забезпечення якості програмних продуктів. Стаття присвячена постановці проблеми в галузі забезпечення якості програмних продуктів. В якості одного з підходів, які дозволяють вирішити поставлену проблему, автор розглядає backcasting-підхід, який допомагає прийняти зважені, фундаментальні, обгрунтовані, легітимні, якісні рішення при виборі програмних проектів, а також може підвищувати спільну відповідальність і усувати конфлікти цілей та інтересів зацікавлених сторін за рахунок проблемної орієнтованості, розвитку майбутнього бачення та активної участі усіх зацікавлених сторін на всіх етапах підходу.

Ключові слова: програмне забезпечення (ПЗ), програмний проект, якість програмного забезпечення, забезпечення якості ПЗ, backcasting-підхід, зацікавлені сторони.

Форм. 0. Табл. 1. Рис. 11. Літ. 6.

Т.А.Говорущенко. Обеспечение качества программных продуктов. Статья посвящена постановке проблемы в отрасли обеспечения качества программных продуктов. В качестве одного из подходов, которые разрешают решить поставленную проблему, автор рассматривает backcasting-подход, который помогает принять взвешенные, фундаментальные, обоснованные, легитимные, качественные решения при выборе программных проектов, а также может повышать совместную ответственность и устранять конфликты целей и интересов заинтересованных сторон благодаря проблемной ориентированности, развитию будущего видения и активному участию всех заинтересованных сторон на всех этапах подхода.

Ключевые слова: программное обеспечение (ПО), программный проект, качество программного обеспечения, обеспечение качества ПО, backcasting-подход, заинтересованные стороны.

Форм. 0. Табл. 1. Рис. 11. Лит. 6.

Т.О.Новоружченко. Software Quality Assurance. The article is devoted to the formulation of the problem of the software quality assurance industry. As one of the approaches, that allow to solve the problem, the author considers the backcasting-approach, that helps to make informed, fundamental, grounded, legitimate, quality decisions at the selecting of software projects, can improves joint responsibility and eliminates the conflicts of stakeholders goals and interests through problem orientation, development of future vision and active participation of all stakeholders at all approach stages.

Key words: software, software project, software quality, software quality assurance, backcasting-approach, stakeholders.

Form. 0. Tab. 1. Fig. 11. Lit. 6.

Вступ. Ключовим фактором забезпечення ефективного застосування програмних продуктів є ретельне оцінювання та досягнення високих значень показників якості. Програмне забезпечення (ПЗ) розробляють вже понад п'ятдесят років, і за цей період задачі, які воно може вирішувати, рівень їх складності та форми представлення отриманих результатів кардинально змінилися. Але й дотепер розробка якісних програмних продуктів не стала нормою. Джерела несправностей сучасного ПЗ вкрай різноманітні, і це лише ускладнює проблему, а також збільшує її масштаб та вартість. Постійне підвищення складності функцій, реалізованих програмами в інформаційних системах, безпосередньо призводить до збільшення їх обсягу та трудомісткості створення.

Звісно, за останні роки програмна індустрія досягла такого рівня розвитку, при якому вимоги до забезпечення якості стали обов'язковим пунктом договорів на предмет розроблення програмних систем. У відповідності до стандартів [1], забезпечення якості - це сукупність планованих та систематичних заходів, необхідних для впевненості в тому, що продукція або процеси задовольняють певні вимоги до якості. Система забезпечення якості стосовно програмних засобів - це сукупність методів та засобів організації керуючих та виконавчих підрозділів підприємства, які беруть участь в проектуванні, розробленні та супроводженні комплексів програм з метою надання їм властивостей, що забезпечують задоволення певних потреб замовників та споживачів при мінімальних або допустимих витратах ресурсів [2]. Гарантування якості ПЗ – проблема, вирішення якої потребує комплексного дослідження за наступними напрямками: 1) розроблення засобів аналізу і оцінювання якості ПЗ на різних етапах його життєвого циклу (ЖЦ); 2) визначення і управління параметрами, які впливають на якість ПЗ на всіх етапах його ЖЦ, із залученням всіх зацікавлених осіб. Потреба у забезпеченні якості ПЗ випливає з двох причин: 1) помилки та відмови критично-важливого ПЗ загрожують катастрофами, які призводять до людських жертв, екологічних катаклізмів, значних економічних та часових втрат; 2) застосування неякісного ПЗ призводить до фінансових збитків.

Постановка проблеми. Наразі істотною і невід'ємною властивістю програмних продуктів є їх складність. Постійне зростання складності функцій ПЗ неминує призводить до збільшення їх обсягу та трудомісткості створення. Щільність помилок, поява яких очікується в ПЗ різного розміру, представлена на рис.1 [3].

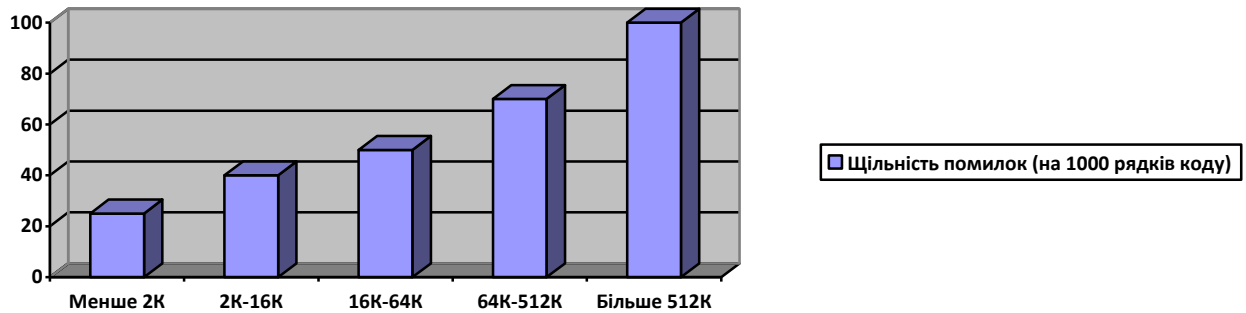


Рис.1. Типова щільність помилок для ПЗ різного розміру

З рис. 1 слідує, що сучасне ПЗ обсягом в мільйони рядків коду в принципі не може бути безпомилковим. Проблема полягає в тому, щоб забезпечити потрібну якість ПЗ з врахуванням того, що деяка невідома кількість помилок та дефектів у ПЗ завжди залишається, а їх негативна дія повинна бути блокована або скорочена до допустимого рівня.

За наближеними оцінками витрати на розроблення ПЗ складають близько 275 мільярдів доларів, але лише 72% програмних проектів досягають етапу впровадження і всього 26% програмних проектів завершуються успіхом [4], тоді існує наступний розподіл витрат – рис.2.

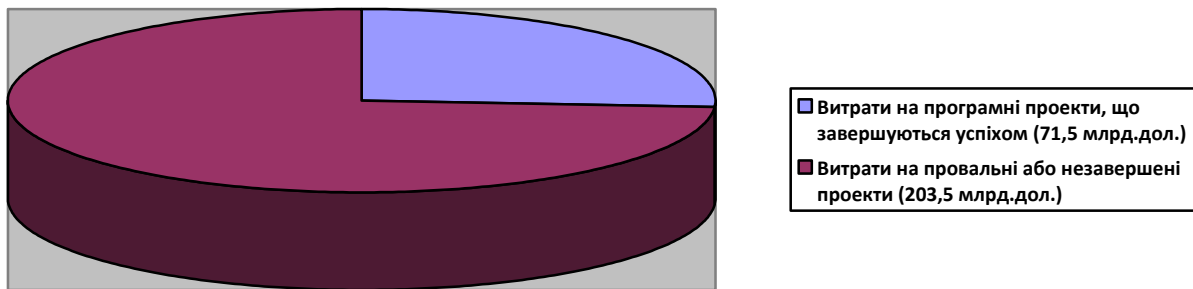


Рис.2. Розподіл витрат на програмні проекти

Програмні проекти часто зазнають невдач через неадекватне формулювання вимог, невдале проектування або неефективне планування, невірне розуміння або недостатній аналіз специфікації та проекту, тобто через помилки на ранніх етапах життєвого циклу ПЗ [3]. Таблиця 1 відображає відсоткову кількість помилок, які виникають на етапах формулювання вимог, проектування та реалізації ПЗ різного обсягу [3].

Таблиця 1. Розподіл помилок, припущених на різних етапах життєвого циклу

Етап ЖЦ	Обсяг ПЗ				
	2К	8К	32К	128К	512К
Формулювання вимог	До 10%	До 15%	До 20%	До 22%	До 23%
Проектування	До 15%	До 19%	До 25%	До 28%	До 32%
Конструювання	До 75%	До 66%	До 55%	До 50%	До 45%

З таблиці 1 видно, що помилки формулювання вимог та проектування складають 25-55% всіх помилок, причому чим більший обсяг ПЗ, тим більше помилок вноситься саме на ранніх етапах.

Отже, забезпечення можливості раннього виявлення помилок та гарантування якості ПЗ на етапі проектування дали б можливість уникнути ряду катастроф та інцидентів, причини яких були внесені на етапах формулювання вимог та проектування.

В основу процесу розроблення ПЗ покладено фундаментальну ідею: проектування ПЗ є формальним процесом, який можна вивчати і вдосконалювати. Завдяки засвоєнню і правильному застосуванню методів і засобів створення ПЗ можна підвищити його якість, забезпечити керованість його процесу проектування і збільшити термін його життя. Але при тому, що складність деяких програмних розробок на сьогодні вже перевищує складність багатьох машинобудівних або інфраструктурних проектів, а наслідки помилок є катастрофічними, процес розроблення ПЗ залишається не забезпеченим фундаментальною теорією та методологією надбання знань і навичок. Всі дослідження у галузі забезпечення якості на ранніх етапах життєвого циклу мають хаотичний, несистематизований характер, хоча, як доведено вище, саме на етапі проектування можна й варто виявляти та усувати до 55% всіх помилок майбутнього ПЗ. Безумовно, є ряд фундаментальних досліджень (роботи Боєма, Дейкстри, Мейєра), але відсутня завершена, протестована та апробована теорія та методологія розроблення якісного ПЗ, а також забезпечення якості ПЗ на ранніх етапах ЖЦ.

Існуючий стан галузі забезпечення якості програмних продуктів представлено на рис.3.

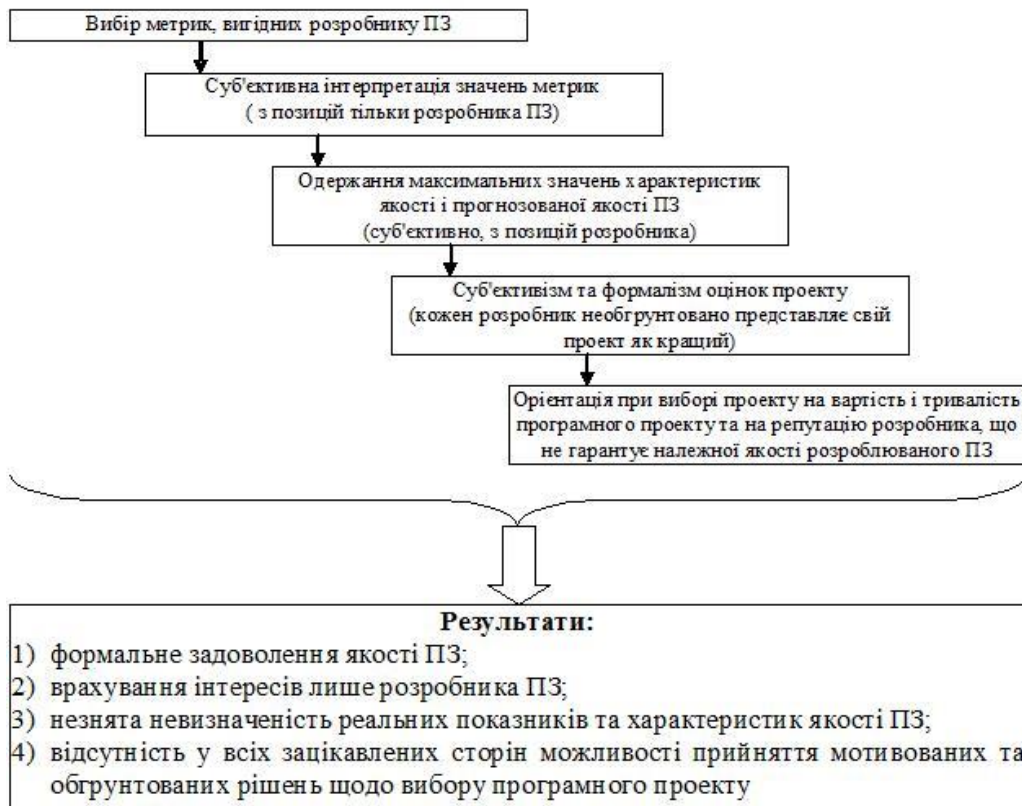


Рис.3. Сучасний стан галузі забезпечення якості ПЗ

Отже, наразі процес забезпечення якості ПЗ є недетермінованим, оскільки відсутні теорія та методологія в галузі оцінювання, прогнозування та забезпечення якості ПЗ, яка б при використанні однакових технологій розроблення із застосуванням однакових стандартів гарантувала створення однаково якісного ПЗ з відповідними витратами та в межах заданого часу. *Актуальною* є проблема створення методів і засобів досягнення необхідного рівня якості для проекту та розроблюваного ПЗ, а *перспективним напрямком досліджень* наразі є забезпечення галузі гарантування якості ПЗ фундаментальними теорією та методологією.

Використання backcasting-підходу для забезпечення якості програмних продуктів. Потіки етапу проектування програмного забезпечення представлені на рис.4.

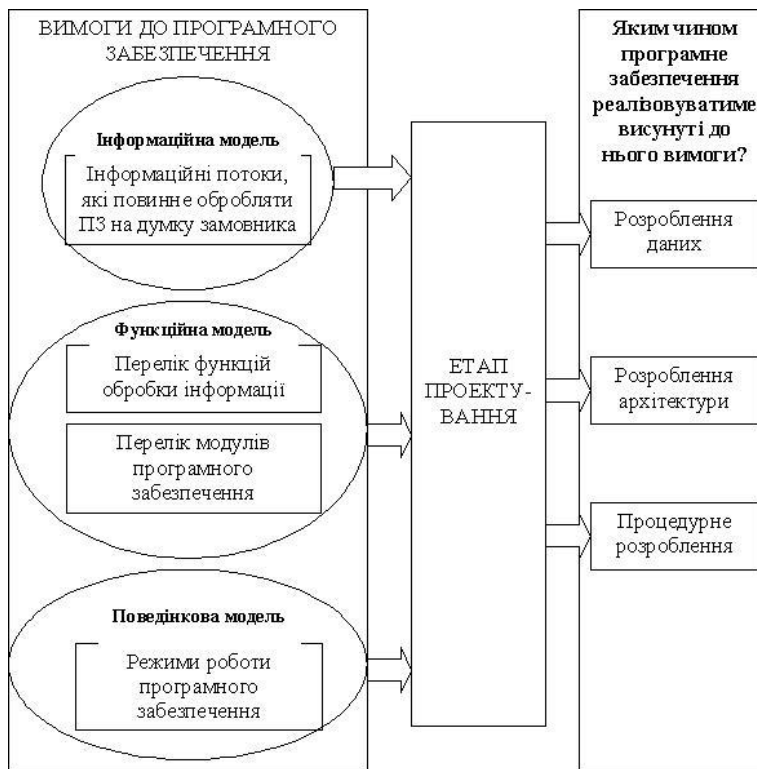


Рис.4. Потоки етапу проектування ПЗ

Аналіз потоків етапу проектування дає можливість визначити основні особливості етапу проектування: значна невизначеність, багатовимірність, велика кількість зацікавлених сторін з різними, часто конфліктними інтересами. При цьому на основі такої інформації необхідно отримати якісні, проблемно-орієнтовані, довготривалі, зважені та ґрунтовні рішення щодо забезпечення якості розроблюваного ПЗ, які враховують цілі та інтереси усіх зацікавлених осіб, а також дозволяють вносити зміни під час наступних етапів життєвого циклу ПЗ, адже великі програмні проекти можуть тривати й декілька років, а вимоги до ПЗ можуть змінюватись під час його проектування, розроблення і навіть супроводу. Одним з підходів, які дозволяють отримати якісні проблемно-орієнтовані рішення на основі невизначених багатовимірних даних із залученням значної кількості зацікавлених сторін, є backcasting-підхід [5, 6].

При використанні backcasting-підходу робота починається із визначення бажаного майбутнього, а потім працює в зворотньому шляху для визначення заходів, які необхідні для досягнення бажаного майбутнього. Основна ідея такого підходу - "якщо ми хочемо досягти певної мети, які дії для цього повинні бути виконані?". Backcasting – це спосіб, в якому передбачаються майбутні бажані умови, а його етапи визначаються з позиції досягнення цих умов. Backcasting-підхід часто є більш ефективним, ніж прогнозування, оскільки прогнозування спирається на відоме сьогодні, але ці знання завжди недосконалі та часто змінюються. Такий підхід наразі посідає центральне місце в стратегічному підході до планування стійкого розвитку та іновацій в багатьох галузях.

В контексті забезпечення якості ПЗ можна уявити велику кількість програмних проектів, призначених для вирішення однієї й тієї ж задачі. Наразі при виборі програмного проекту замовники та розробники орієнтуються на вартість і тривалість програмного проекту, а також на репутацію розробника, що абсолютно не гарантує належного рівня якості. Особливо складною проблема вибору є для проектів, вартість, тривалість та репутація розробника яких є приблизно однаковими. Backcasting-підхід здатен допомогти замовникам та розробникам при виборі програмного проекту прийняти зважені та ґрунтовні рішення завдяки розвитку майбутнього бачення та врахуванню інтересів усіх зацікавлених сторін.

Backcasting-підхід для проблеми забезпечення якості розроблюваного ПЗ на етапі проектування представимо в наступному вигляді - рис.5.

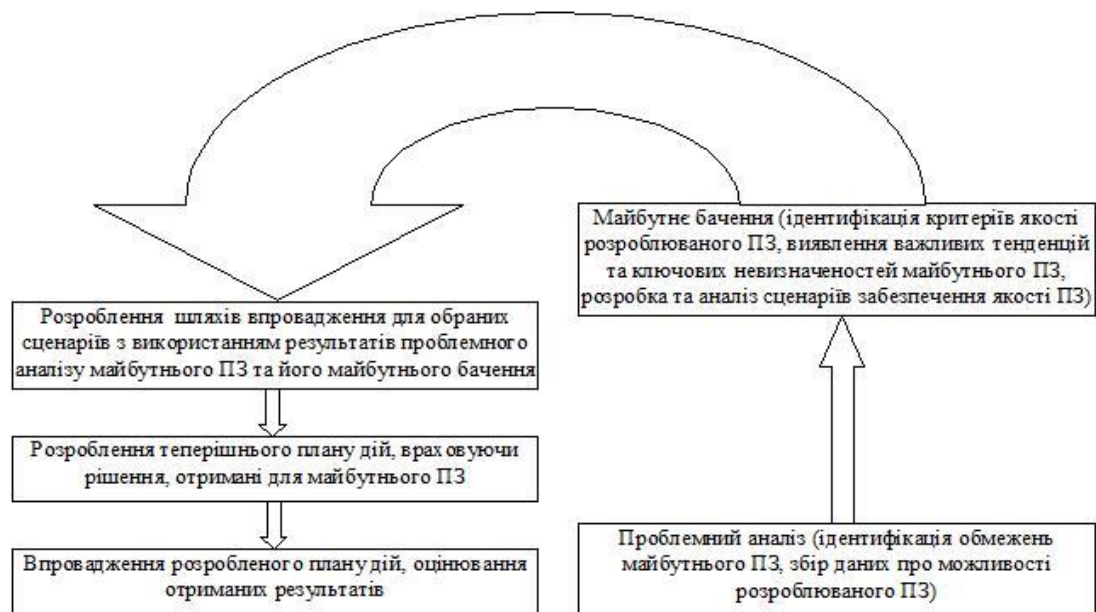


Рис.5. Backcasting-підхід для забезпечення якості ПЗ

Деталізовано етапи backcasting-підходу представлені на рис.6.

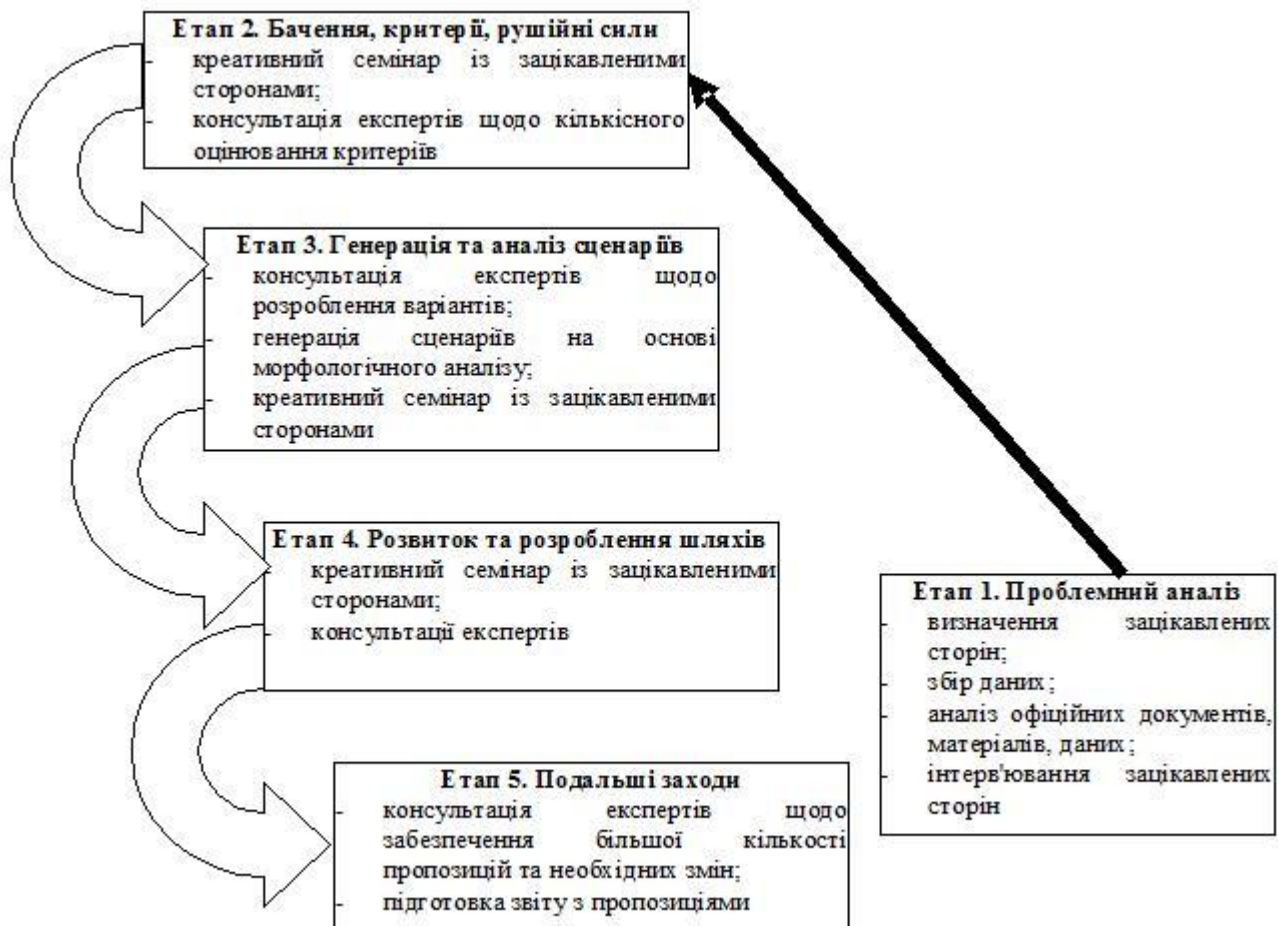


Рис.6. Етапи backcasting-підходу

Роль backcasting-підходу на етапі проектування для забезпечення якості розроблюваного ПЗ представлена на рис.7.

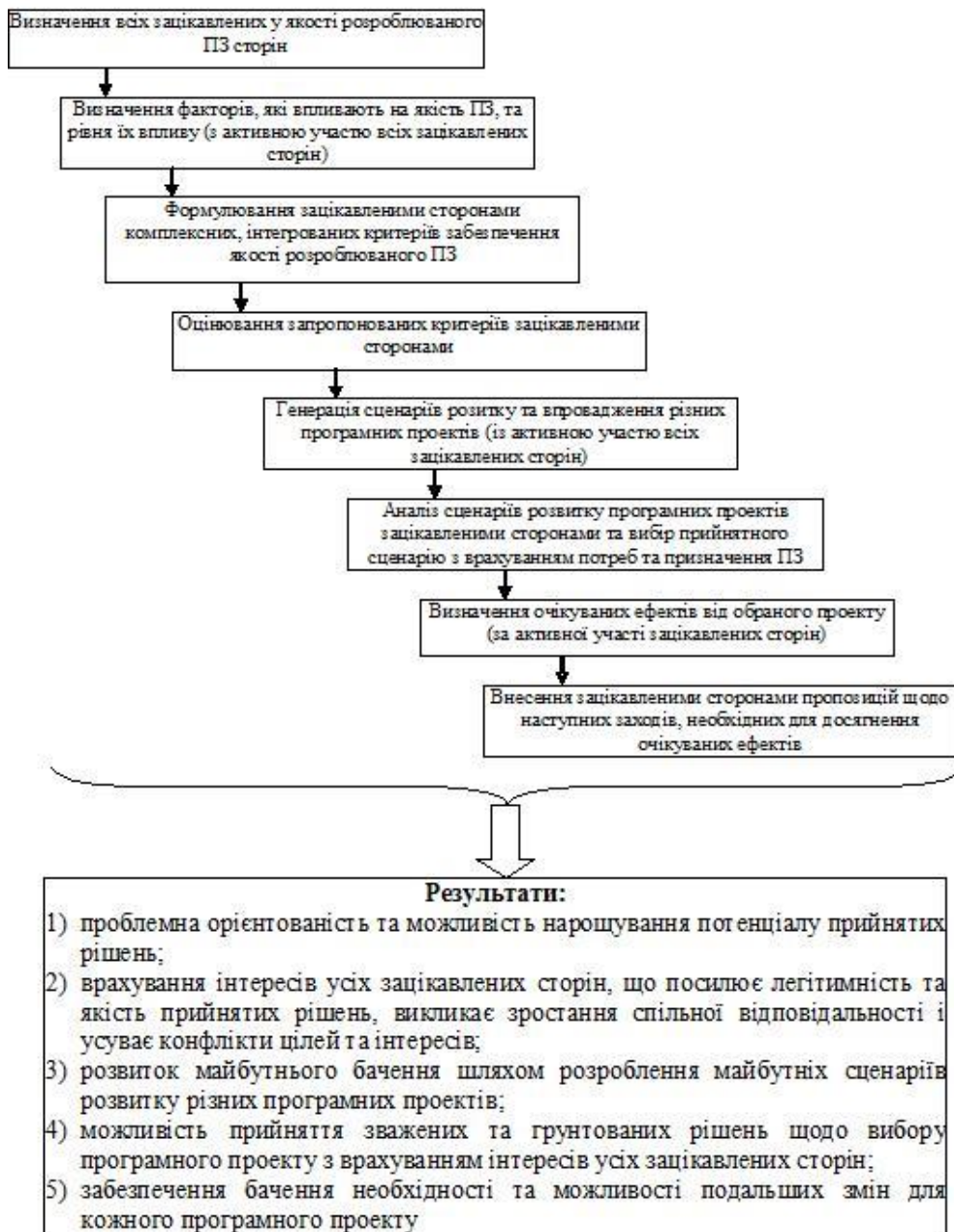


Рис.7. Роль backcasting-підходу для забезпечення якості ПЗ

Етап 1 - проблемний аналіз. Етап проблемного аналізу при забезпеченні якості ПЗ на етапі проектування було розпочато із визначення зацікавлених сторін. Зацікавлені сторони - фізичні та юридичні особи, які мають легітимний інтерес в діяльності організації, тобто залежать від неї або впливають на її діяльність, які можуть впливати на прийняття рішень або на яких можуть впливати прийняті рішення [5, 6]. Участь всіх зацікавлених сторін при прогнозуванні якості ПЗ є важливою, оскільки така спільна робота посилює легітимність прийнятих рішень, надає контекст-залежні рішення та знання, впливає на зростання якості одержаних результатів, а також викликає зростання спільної відповідальності за всі прийняті рішення. Узагальнено зацікавлені сторони, з точки зору проблеми досягнення необхідного рівня якості ПЗ, представлені на рис.8.

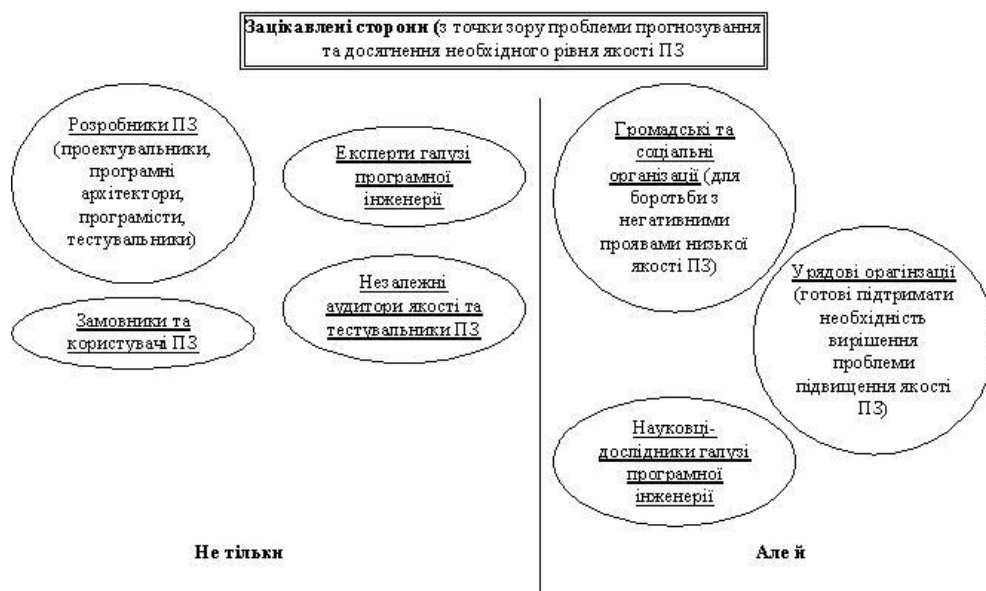


Рис.8. Зацікавлені сторони для проблеми забезпечення якості ПЗ

Далі будувалась сітка зацікавлених сторін, де всі зацікавлені сторони розподілялися згідно рівня своєї зацікавленості та рівня можливого впливу на вирішення проблеми (рис.9).

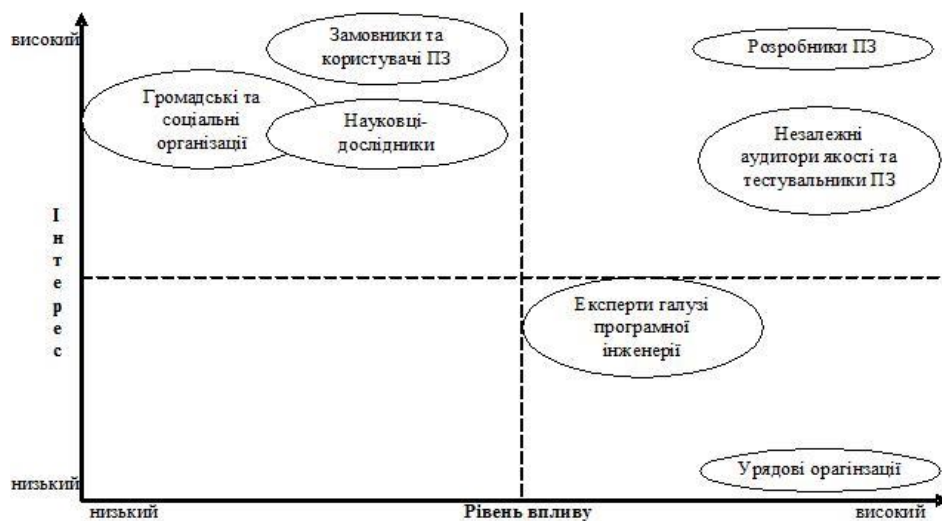


Рис.9. Сітка зацікавлених сторін для проблеми забезпечення якості ПЗ

Після цього зацікавленими сторонами визначались фактори, які впливають на проблему якості ПЗ. Для цього зацікавлені сторони проаналізували всі офіційні та неофіційні документи. Особливу увагу при цьому було приділено галузевим стандартам, тому що при розробленні ПЗ софтверні організації зобов'язані керуватись стандартами як щодо процесів розроблення, так і щодо процесів оцінювання та забезпечення якості. Крім стандартів, зацікавленими сторонами було розглянуто документацію організації-замовника, яка допомогла виявити особливості використання розроблюваного ПЗ, а також специфікацію вимог до ПЗ.

Після визначення множини впливових факторів зацікавлені сторони вказали рівень впливу кожного з факторів, а також розбили множину факторів на підмножини наступним чином:

1) економічні фактори - ефективність (значний рівень впливу), кількість виконавців (середній), розмір і вартість документації для користувачів (незначний), вартість помилки (значний), прогнозовані розмір і вартість розроблюваного ПЗ (значний);

2) технологічні фактори - застосовувані при проектуванні та розробленні ПЗ процеси та технології (значний рівень впливу), функційна придатність (значний), можливість переносу (середній), функційні вимоги (значний), нефункційні вимоги (значний);

3) соціальні фактори - зручність використання (значний), надійність (значний), розмір та склад цільової аудиторії (середній), рівень безпеки програмного забезпечення (значний);

4) організаційні фактори - середовище організації-розробника та зміни в ньому (середній), цілі розробника (середній), розмір та структура організації розробника (незначний), сумісність (середній), супроводжуваність (незначний), термін виконання (значний), залежності та суперечливості вимог (значний).

Етап 2 – формування майбутнього бачення, визначення критеріїв та рушійних сил.

Формування цілісного бачення щодо забезпечення належного рівня якості ПЗ було виконане зацікавленими сторонами після ретельного вивчення ними всіх фактів, які містяться в документації, архівних записах та статистичних вибірках, а також виявлялися під час ретельних спостережень, оглядів, інтерв'ю та опитувань.

Аналіз множини факторів, які впливають на проблему виявлення можливого рівня якості та досягнення необхідного рівня якості ПЗ, а також множини одержаних фактів дав можливість зацікавленим сторонам сформулювати основні критерії забезпечення якості ПЗ. При формуванні критеріїв кожна зацікавлена сторона висувала та захищала саме важливі для неї критерії. Наступним кроком стало визначення в даному випадку п'яти найважливіших інтегрованих критеріїв. З точки зору зацікавлених сторін для забезпечення якості ПЗ такими критеріями є: 1) безпека та надійність ПЗ - даний критерій відстоюють замовники та користувачі ПЗ, громадські та соціальні організації, незалежні аудитори якості та тестувальники, експерти галузі програмної інженерії, науковці-дослідники; 2) функційна придатність ПЗ - відстоюють розробники ПЗ, замовники та користувачі ПЗ, науковці-дослідники; 3) зручність використання ПЗ - відстоюють замовники та користувачі ПЗ; 4) економічна ефективність ПЗ - відстоюють розробники ПЗ, замовники та користувачі ПЗ, урядові організації; 5) сумісність ПЗ - відстоюють замовники та користувачі ПЗ.

Після формулювання основних критеріїв зацікавлені сторони виконали їх узгоджене оцінювання за 5-бальною шкалою, при якому кожна зацікавлена сторона відстоювала свою думку та свою оцінку і намагалась переконати інші сторони в правильності своїх міркувань. В жодному разі оцінка критерію не є середнім арифметичним оцінок даного критерію, наданих різними зацікавленими сторонами, - вона відображає досягнутий компроміс між зацікавленими сторонами.

Наступним кроком було розташування визначених комплексних критеріїв в сітці, яка відображає величину рушійної сили та рівень впливу даного критерію на проблему (рис. 10).

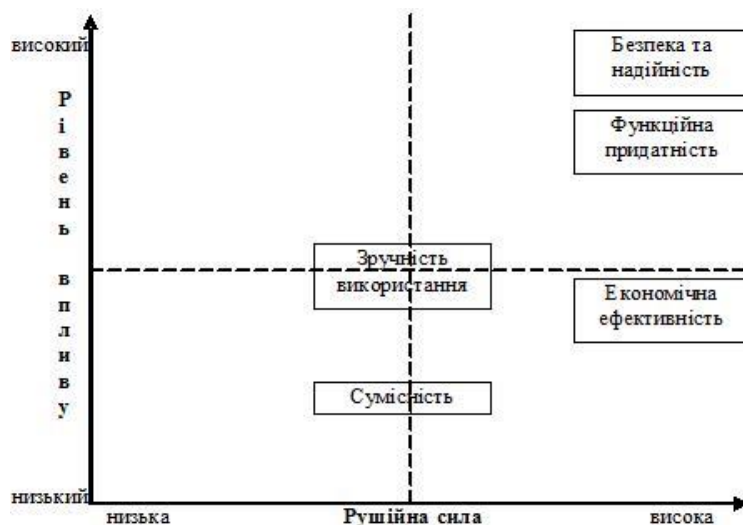


Рис.10. Сітка основних критеріїв для проблеми забезпечення якості ПЗ

Етап 3 – генерація та аналіз сценаріїв. На третьому етапі відбулась генерація сценаріїв розвитку та впровадження різних програмних проектів. Розробниками було запропоновано наступні програмні проекти та сценарії їх розвитку:

1) більша частина проектного часу та коштів (80%) витрачається на забезпечення можливості програми виконувати задані функції в заданих умовах роботи без збоїв та відмов, а решта часу та коштів - на естетичність, доступність, функційну повноту, можливість співіснування з іншим програмним забезпеченням на різних платформах, продуктивність та повернення інвестицій;

2) більша частина проектного часу та коштів витрачається на функційну повноту та доцільність (приблизно 50%) та на можливість вивчення, естетичність та доступність (приблизно 30%), а на забезпечення безперебійної роботи ПЗ, можливість співіснування з іншим програмним забезпеченням на різних платформах, продуктивність та окупність доводиться решта часу та коштів;

3) більша частина проектного часу та коштів доводиться на можливість співіснування та взаємодії з іншим програмним забезпеченням на різних платформах (40%) та продуктивність, окупність і повернення інвестицій (40%), а решта проектного часу та коштів витрачається на функційну повноту, естетичність, доступність і забезпечення безперебійної роботи ПЗ.

Для здійснення обґрунтованого та мотивованого вибору програмного проекту зацікавленими сторонами було виконано аналіз сценаріїв розвитку запропонованих проектів - всі зацікавлені сторони оцінили за п'ятибальною шкалою розроблені сценарії за кожним з обраних критеріїв, виходячи з потреб та призначення ПЗ, та знайшли середні оцінки. Після чого було пораховано сумарні бали для кожного сценарію. Так, наприклад, для критичного ПЗ сценарій (проект) №1 одержав 17 балів, сценарій (проект) №2 - 14 балів, сценарій (проект) №3 - 13 балів. Отже, для забезпечення необхідного рівня якості критично важливого ПЗ слід обирати сценарій (проект) №1.

Етапи 4 і 5 – розроблення шляхів та подальших заходів для обраного сценарію. На четвертому етапі зацікавлені сторони для обраного сценарію надавали відповіді на наступні запитання: 1) Які зміни (методологічні - М, структурні - С, технологічні - Т) необхідні?; 2) Які зацікавлені сторони необхідні для виконання даних змін?; 3) Терміни, доречні для запропонованих змін? - при цьому процес забезпечення якості ПЗ розглядався в трьох часових перспективах ЖЦ: етапи реалізації, тестування та супроводу. Відповіді на ці питання представлено на графіку - рис. 11.

<p>Т</p> <p>С</p> <p>М</p>	<p>Підвищення зрілості технологічних процесів в софтверній організації</p> <p><u>Зацікавлені сторони:</u> розробники ПЗ, незалежні аудитори, науковці, замовники та користувачі</p>	<p>Зосередженість на функціях самого ПЗ, які допомагають виявляти помилки</p> <p><u>Зацікавлені сторони:</u> розробники ПЗ, замовники та користувачі, аудитори, науковці, експерти</p>	<p>Приділення значної уваги питанням супроводу після передачі ПЗ в експлуатацію</p> <p><u>Зацікавлені сторони:</u> замовники та користувачі, громадські та урядові організації, експерти</p>
	<p>Забезпечення можливості кількісного вимірювання результатів етапу реалізації - аудит коду</p> <p><u>Зацікавлені сторони:</u> замовники та користувачі, аудитори, експерти, науковці</p>	<p>Тестування ПЗ з врахуванням результатів аудиту коду та структурних особливостей ПЗ</p> <p><u>Зацікавлені сторони:</u> науковці, замовники та користувачі, аудитори, експерти</p>	<p>Забезпечення задоволення вимог користувачів програмним забезпеченням протягом всієї експлуатації</p> <p><u>Зацікавлені сторони:</u> замовники та користувачі, аудитори, експерти, науковці, громадські організації</p>
	<p>Розроблення методологій розроблення ПЗ з метою концентрування уваги на програмному коді та на детальних моделях ПЗ</p> <p><u>Зацікавлені сторони:</u> науковці, експерти, громадські організації, урядові організації</p>	<p>Розроблення методологій тестування ПЗ на основі аналізу специфікації вимог та з орієнтацією на код і дефекти</p> <p><u>Зацікавлені сторони:</u> науковці, експерти, аудитори, громадські організації, урядові організації</p>	<p>Розроблення методологій супроводу ПЗ з метою усунення збоїв та покращення показників продуктивності</p> <p><u>Зацікавлені сторони:</u> науковці, експерти, аудитори, громадські організації, урядові організації</p>
<p>Етап проектування</p>	<p>Етап реалізації</p>	<p>Етап тестування</p>	<p>Етап супроводу</p>

Рис.11. Необхідні методологічні, структурні та технологічні зміни для ПЗ

На п'ятому етапі зацікавленими сторонами було визначено очікувані ефекти від обраного сценарію. Так, для обраного сценарію №1 очікуваними ефектами зацікавлені сторони назвали: підвищення надійності та безпеки розроблюваного ПЗ; підвищення зрілості технологічних процесів в організації; забезпечення можливості кількісного вимірювання програмного коду; розроблення відповідних методологій для етапів ЖЦ, які можуть бути використані у наступних проектах.

Після визначення очікуваних ефектів зацікавлені сторони сформулювали наступні заходи, необхідні для досягнення ефектів: 1) науковці, експерти та аудитори вимагають стандартизувати метрики для аудиту програмного коду, а також вирішити проблему складності інтерпретації величин метрик; 2) користувачі вимагають розробити програмні засоби, які б були орієнтовані не тільки на розрахунок метрик на етапах проектування та реалізації, але й на трактування одержаних величин метрик для більш ґрунтовного та мотивованого вибору програмного проекту; 3) замовники ПЗ та урядові організації вимагають розробити методологію оцінювання якості проекту та прогнозування якості розроблюваного за проектом програмного забезпечення на етапі проектування, яка надаватиме можливість раннього виявлення помилок, що забезпечить підвищення якості ПЗ та зменшення витрат на його розроблення; 4) розробники ПЗ вимагають розробити теорію та методологію галузі програмної інженерії, яка б при використанні однакових технологій розроблення із застосуванням однакових стандартів гарантувала створення однаково якісного ПЗ з відповідними витратами та в межах заданого часу; 5) громадські організації вимагають розробити фундаментальну теорію та методологію оцінювання і забезпечення (гарантування) якості ПЗ, яка дозволить зменшити кількість та вплив помилок і відмов ПЗ, що загрожують катастрофами, людськими жертвами, екологічними катаклізмами, значними економічними та часовими втратами.

Висновки. Автором доведено можливість застосування backcasting-підходу на етапі проектування для забезпечення необхідного (з точки зору призначення ПЗ) рівня якості ПЗ.

Перевагами використання такого підходу при забезпеченні якості ПЗ є:

- 1) проблемна орієнтованість підходу;
- 2) можливість розвитку майбутнього бачення, розроблення майбутніх альтернатив та сценаріїв;
- 3) активна участь та врахування інтересів усіх зацікавлених сторін на всіх етапах підходу (при бажанні зацікавлених сторін активно співпрацювати та йти на компроміс), що посилює легітимність прийнятих рішень, впливає на зростання якості одержаних результатів, а також викликає зростання спільної відповідальності та дозволяє усунути конфлікти цілей та інтересів;
- 4) можливість прийняття зважених та ґрунтовних рішень щодо вибору програмних проектів;
- 5) можливість бачення необхідності та здійсненності подальших змін для різних проектів.

Подальшими перспективами використання backcasting- підходу для забезпечення якості ПЗ є:

- 1) розроблення математичного підґрунтя для використання даного підходу в галузі;
- 2) розроблення фундаментальної теорії та методології забезпечення якості ПЗ;
- 3) забезпечення необхідного ступеня довіри до даного підходу та одержаних рішень.

Література

1. Сборник действующих международных стандартов ИСО серии 9000. Т.1, 2, 3. - М.: ВНИИКИ, 1998.
2. Липаев В.В. Программная инженерия. Методологические основы. - М.: ТЕИС, 2006. - 608 с.
3. Макконнелл С. Совершенный код - М.: Издательство "Русская редакция", 2013 - 896 с.
4. Мищенко В.О., Поморова О.В., Говорушенко Т.А. CASE-оценка критических программных систем. В 3-х томах. Том 1. Качество / Под ред. Харченко В.С. - Харьков: НАУ "ХАИ", 2012. - 201 с.
5. Ebert J. E. J., Gilbert D. T., Wilson T. D. Forecasting and back casting: Predicting the impact of events on the future // Journal of Consumer Research, 2009 - Volume: 36, Issue: 3, pp.353-366
6. Höjer M., Mattsson L-G. Determinism and backcasting in future studies // Futures, 2010 - Volume 32, pp. 613-634

Рецензенти: Остапов Сергій Едуардович, доктор фізико-математичних наук, професор, завідувач кафедри програмного забезпечення комп'ютерних систем Чернівецького національного університету ім. Ю.Федьковича; Шалапко Юрій Іванович, доктор технічних наук, професор, завідувач кафедри інженерної механіки Хмельницького національного університету.