

УДК 004.732.056

С. Г. Семенов, Кассем Халіфе

## КОМПЛЕКС МАТЕМАТИЧНИХ МОДЕЛЕЙ ПРОЦЕСУ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Національний технічний університет «ХПІ» м. Харків

**Анотація.** У статті представлений комплекс математичних моделей основних етапів процесу розробки програмного забезпечення. При моделюванні були використані основні положення і досвід використання гнучких методологій розробки програмного забезпечення. Основною метою математичного моделювання є дослідження можливостей оптимізації розробки програмного забезпечення з урахуванням факторів і показників безпеки. В результаті розроблені моделі ініціалізації і реалізації функціоналу програмного забезпечення.

**Ключові слова:** програмне забезпечення, математична модель, гнучка методологія розробки програмного забезпечення.

**Аннотация.** В статье представлен комплекс математических моделей основных этапов процесса разработки программного обеспечения. При моделировании были использованы основные положения и опыт использования гибких методологий разработки программного обеспечения. Основной целью математического моделирования является исследования возможностей оптимизации разработки программного обеспечения с учетом факторов и показателей безопасности. В итоге разработаны модели инициализации и реализации функционала программного обеспечения.

**Ключевые слова:** программное обеспечение, математическая модель, гибкая методология разработки программного обеспечения.

**Abstract.** The article presents a complex of mathematical models of the main stages of the software development process. During the simulation, the basic assumptions and experience of using flexible software development methodologies were used. The main goal of mathematical modeling is to investigate the possibilities of optimizing software development taking into account factors and safety indicators. As a result, models for the initialization and implementation of the software functionality have been developed.

**Keywords:** software, mathematical model, flexible methodology of software development.

### Вступ

Проведені дослідження показали, що в даний час існує декілька найбільш популярних методологій розробки ПЗ. Серед них доцільно виділити сімейство «гнучких» методологій Agile (XP, SCRUM) і методології «дбайливого виробництва» (Kanban). Слід зауважити, що в літературі [1-8] і такі методології часто відносять до розряду «гнучких». У кожній з них є свої особливості, які варто враховувати, вибираючи ту чи іншу методологію для управління проектом.

Крім того, для успішного управління процесом розробки програмного забезпечення буває недостатньо вибрати чи іншу методологію і дотримуватися її протягом усього процесу. У разі досить великого проекту, розробка якого ведеться досить довгий період часу, важливою може виявитися здатність швидко адаптувати використовувану в даний момент методологію відповідно до обставин, що змінюються, тобто по суті синтезувати різні варіанти використання «гнучких» і «дбайливих» методологій в один проект.

Таким чином, в даний час питання, пов'язані з оптимізацією процесу розробки ПЗ, незважаючи на різноманіття «гнучких» методологій управління, залишаються актуальними.

Питання оптимізації нерозривно пов'язані з питаннями моделювання та прогнозування. Стосовно процесу розробки програмного забезпечення одним із шляхів вирішення завдання прогнозування є використання підходу, заснованого на оцінці часових витрат на окремі етапи розробки ПЗ, з урахуванням функцій залежно поточного числа активних дефектів додатка від часу, отриманих експериментальним шляхом, а також за допомогою математичного моделювання. Таке комплексне використання апріорних та апостеріорних даних має дозволити врахувати специфіку сучасних методик розробки ПЗ з можливою динамічною зміною (розширенням) рамок проекту за бажанням замовника або з інших причин.

### Актуальність

Саме цього актуальним є питання математичного моделювання процесу розробки програмного забезпечення з урахуванням особливостей гнучких методологій розробки.

### Мета статті

Метою статті є розробка комплексу математичних моделей етапів ініціалізації і реалізації функціоналу ПЗ. При цьому вперше розроблена математична модель етапу ініціалізації процесу розробки ПЗ, заснована на концептуальних положеннях Agile, а удосконалена математична модель етапу реалізації функціоналу ПЗ, відрізняється від відомих урахуванням показників безпечного програмування.

### Розв'язання задач

1. Математична модель етапу ініціалізації процесу розробки ПЗ.

З літератури [4-7], що описує процес управління розробкою ПЗ відомо, що даний час в гнучких методологіях використовується ряд положень, суть яких зафіксована в маніфесті Agile, і які коротко можна

сформулювати таким чином:

- розробка ведеться короткими циклами (ітераціями), тривалістю 1-4 тижні;
- у кінці кожної ітерації замовник отримує цінний для нього додаток (чи його частину), який можна використати у бізнесі;
- команда розробки співпрацює із Замовником в ході усього проекту;
- зміни в проєкті схвалюються і швидко впроваджуються в роботу.

Аналіз прикладів практичної реалізації гнучких методологій дозволив визначити, що етап ініціалізації процесу розробки ПЗ у відповідності представленими положеннями включає низку заходів (мітингів), що мають різні тактичні і стратегічні цілі. Наприклад, Daily meetings (Щоденний контроль). Daily meetings, інакше званий Stand-up Meeting проводиться щодня. На цьому заході кожен член команди повинен відзвітувати про виконану роботу протягом минулої доби, намітити план роботи на день і вирішити, існуючі до інших учасників «мітингу», питання. Тривалість такого заходу повинна бути не більше 15 хвилин. При цьому за регламент «мітингу» стежить Scrum-мастер.

З точки зору тимчасової оптимізації цього процесу суттєвих «виграшів» домогтися складно, оскільки цей захід не займає істотного робочого часу.

В той же час ще одним подібним заходом в гнучких методологіях є «retrospective meeting» (Ретроспективна нарада). Саме ці заходи займають значний час, і можуть бути неефективними при недотриманні регламенту, методологій проведення заходу, низького професійного рівня учасників і інших чинників. Розглянемо детальніше цей процес і математично формалізуємо його. Для математичної формалізації першого етапу пропонується скористатися 4 параметрами оцінки тимчасових витрат: тривалістю первинної колективної оцінки складності проєкту  $T_{оц}$ , час який затрачено на звіт про виконану роботу з моменту попереднього SCRUM-мітингу  $T_{ом}$ , час який затрачено на обговорення проблем, що виникли під час роботи  $T_{об}$ , час на постановку завдань до наступного мітингу  $T_{пост}$ .

Тоді час ініціалізації  $T_u$  можна описати як:

$$T_u = T_{оц} + T_{ом} + T_{об} + T_{пост}. \quad (1)$$

При цьому вказані часові витрати залежать від ряду об'єктивних і суб'єктивних чинників. В цілому, ці показники можна описати таким чином.

Одним з найбільш важливих і детально описаних процесів в Scrum, є процес планування «спринту», проміжку часу впродовж якого виконується робота над продуктом. Тривалість початкової колективної оцінки складності проєкту  $T_{оц}$  залежить значною мірою від методики оцінки, числа учасників SCRUM-мітингу, їх професійної підготовки та комунікативності, що істотно впливає на кількість розбіжностей і спірних моментів при оцінці складності проєкту.

Розглянемо одну з найбільш популярних методик планування і оцінки складності проєкту - покер планування (англ. Planning Poker, а також англ. Scrum poker). Це методика оцінки проєктів при розробці програмного забезпечення, головною метою якої є досягнення домовленості, відносно складності майбутньої роботи або об'єму вирішуваних завдань [6].

Аналіз даної методики показав, що середній час початкової колективної оцінки складності проєкту можна описати як

$$T_{оц} = \sum_{g=1}^{n_{umm_g}} \left( \frac{t_{umm_g}}{n_{umm_g}} + \Delta t_g + \sum_{j=1}^{\hat{n}_g} (a_{g_j} + \Delta t_j) \right) \quad (2)$$

де  $t_{umm_g}$  – час, необхідний на проведення  $g$  – й ітерації оцінки складності проєкту;  $n_{umm_g}$  – число ітерацій, необхідних для досягнення консенсусу;  $\Delta t_g$  – відхилення часу проведення  $g$  – й ітерації оцінки складності проєкту;  $a_{g_j}$  – час, відведений учасникам з високими і низькими оцінками складності проєкту на  $g$  – й ітерації;  $\Delta t_j$  – відхилення часу висловлювання учасників обговорення в  $g$  – й ітерації при обґрунтуванні своєї оцінки;  $\hat{n}_g$  – число учасників з високими і низькими оцінками складності проєкту на  $g$  – й ітерації.

Слід зауважити, що час  $\Delta t_g$  найчастіше не великий і істотно не впливає на загальний час оцінки складності проекту. Пов'язано це з тим, що це показник багато в чому залежить від кваліфікації тільки одного учасника - модератора, який стежить за часом проведення ітерації. Проте рівень професійної підготовки і комунікативності інших учасників SCRUM- мітингу істотно впливає на показник  $\Delta t_j$ , який у свою чергу може змінюватися у великому діапазоні. Виходячи з цього, цей показник визначимо таким чином.

$$\Delta t_j = v \cdot e^{-k \cdot c \cdot \hat{n}_g}, \quad (3)$$

де  $k$  – коефіцієнт, що характеризує середній рівень професійної підготовки учасників SCRUM- мітингу (варіюється від 0,1 до 0,3);  $c$  – коефіцієнт, що характеризує середній рівень комунікативності учасників SCRUM-мітингу (варіюється від 1 до 3);  $v$  – усереднений коефіцієнт складності розв'язуваної узагальноної задачі (проблеми).

Для врахування часових витрат на звіт про виконану роботу з моменту попереднього SCRUM-мітингу  $T_{om}$  скористаємося думками експертів, які визначають, що найчастіше цей показник обмежується регламентом, що встановлюється модератором. Тому в математичній моделі етапу ініціалізації процесу розробки ПЗ його можна прийняти за константу, при цьому використовуючи практичні дані фірм-розробників ПЗ.

Проведені дослідження показали, що визначення витрат на обговорення проблем, що виникли під час роботи  $T_{ob}$ , можна формалізувати наступним виразом.

$$T_{ob} = \sum_{i=1}^{\bar{N}} \left( a_i + \Delta t_i + \sum_{j=1}^{\bar{n}_i} \left( \frac{b_i}{\bar{n}_i} + \Delta t_{ij} \right) \right), \quad (4)$$

де  $\bar{N}$  – кількість проблем, що виникли під час роботи, і винесених на обговорення в SCRUM- мітингу;  $\bar{n}_i$  – кількість учасників обговорення озвучених проблем.

Нескладно помітити, що скорочення часових витрат на обговорення проблем безпосередньо пов'язане з часовими показниками  $\Delta t_i$  і  $\Delta t_{ij}$ , які відповідно залежать від професійної підготовки учасників команди розробників і їх числа.

Проведені дослідження показали, що ці характеристики можна описати таким чином.

$$\Delta t_i = z1 \times e^{-\left(\frac{k \cdot}{\bar{N}}\right)}, \quad (5)$$

$$\Delta t_{ij} = z2 \times e^{-(k \cdot \bar{n}_i)}, \quad (6)$$

де  $z1$  і  $z2$  усереднені коефіцієнти складності проблем обговорення і управління учасниками обговорення відповідно.

Як зазначено на початку підрозділу ще одним показником, що входять в загальне аналітичне вираження для розрахунку середнього часу ініціалізації процесу розробки ПЗ, є час на постановку завдань до наступного мітингу  $T_{nocm}$ . Для розрахунку цього показника також як і в попередньому випадку скористаємося виразом 1. Тоді час  $T_{nocm}$  дорівнює:

$$T_{nocm} = \sum_{i=1}^{\bar{N}} \left( a_i + \Delta t_i \right) \quad (7)$$

де  $\bar{N}$  – кількість завдань до наступного SCRUM- мітингу,  $\bar{a}_i$  – час, відведений на постановку  $i$  – го завдання до наступного SCRUM- мітингу,  $\Delta t_i = e^{-\left(\frac{\bar{N}}{r}\right)}$  – відхилення часу на постановку  $i$  – го завдання до наступного SCRUM- мітингу,  $r$  – усереднений коефіцієнт складності завдань ( $r = \{2,3, \dots, 45\}$ ).

Проведемо дослідження ступеня впливу, наведених в підрозділі характеристик, на загальний час проведення SCRUM-мітингу.

На рис. 1. а наведені криві графіків залежності відхилення  $\Delta t_j$  часу висловлювання учасників обговорення в  $g$ -й ітерації при обґрунтуванні своєї оцінки від коефіцієнта  $k$ , що характеризує середній рівень професійної підготовки учасників SCRUM-мітингу і коефіцієнта  $c$ , що характеризує середній рівень комунікативності учасників SCRUM-мітингу, в умовах коли  $v = 0,6$ ,  $\hat{n}_g = 2$ .

Як видно з цих графіків коефіцієнти істотно (до 2,5 разів) збільшують час  $\Delta t_j$ .

На рис. 1. б наведені криві графіків залежності тривалості  $T_{оц}$  початкової колективної оцінки складності проекту від коефіцієнта  $k$ , що характеризує середній рівень професійної підготовки учасників SCRUM-мітингу і коефіцієнта  $c$ , що характеризує середній рівень комунікативності учасників SCRUM-мітингу, в умовах коли  $v = 0,6$ ,  $\hat{n}_g = 2$ ,  $t_{umm_g} = \{0,5 \dots 1\}$  хв.,  $n_{umm_g} = 6$ ,  $\Delta t_g = 0,1$  хв.,  $a_{g_j} = \{0,1 \dots 0,22\}$  хв.

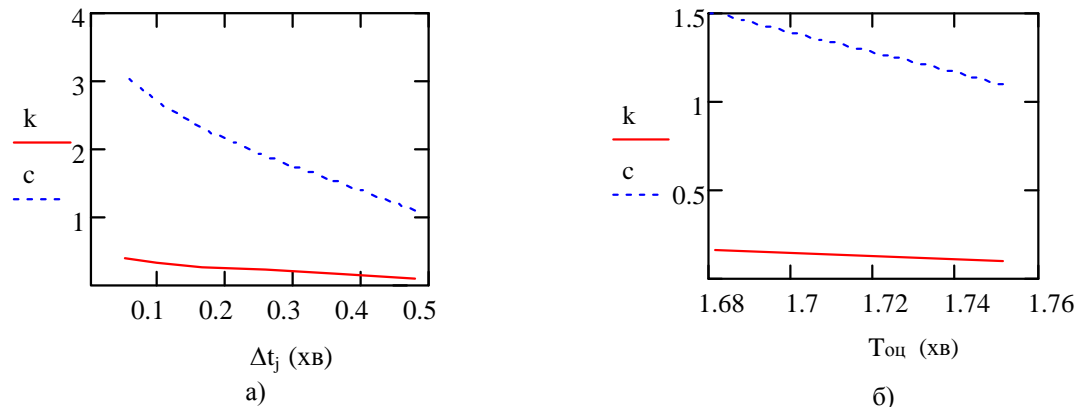


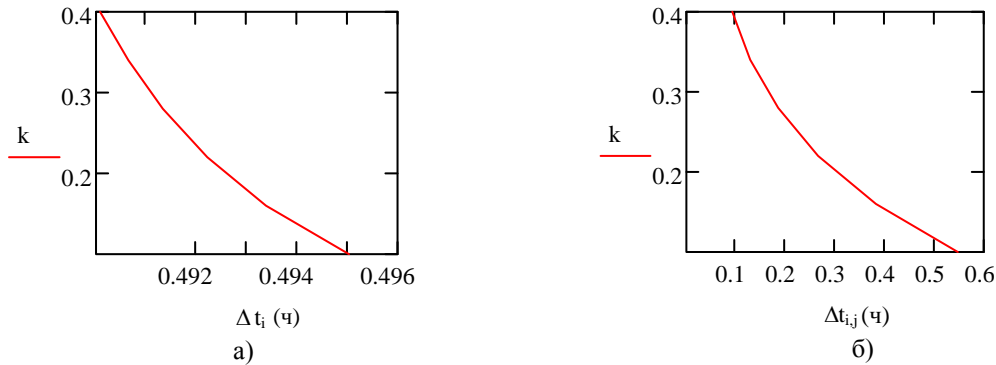
Рисунок. 1 – Графіки залежності відхилення  $\Delta t_j$  від коефіцієнтів  $k$  і  $c$

Як видно з наведених на рис. 1.б графіків поліпшення показника середнього рівня комунікативності учасників SCRUM-мітингу в 1,1 рази (до 10 хвилин за годину) зменшить час початкової колективної оцінки складності проекту.

Слід зауважити, що приблизно аналогічних результатів можна досягти при підвищенні рівня професійної підготовки учасників SCRUM-мітингу.

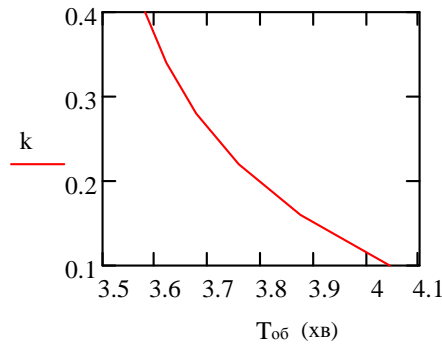
Проведемо дослідження моделі процесу обговорення проблем, що виникли під час роботи. На графіках рис. 2. наведені криві залежності часових показників  $\Delta t_i$  та  $\Delta t_{ij}$  від коефіцієнта, що характеризує середній рівень професійної підготовки учасників SCRUM-мітингу, в умовах, коли  $a_i = \{20,30, \dots, 70\}$  с,  $b_i = \{10,20, \dots, 60\}$  с,  $\tilde{n}_i = 6$ .

Як видно з цих графіків підвищення коефіцієнта  $k$  професійної підготовки в 4 рази приблизно в 1,013 рази (приблизно 36 с.) зменшує час  $\Delta t_i$ , і в 6 разів зменшує час  $\Delta t_{i,j}$  (приблизно 3,5 хвилини).

Рисунок. 2. – Графіки залежності часових показників  $\Delta t_i$  та  $\Delta t_{ij}$  від коефіцієнта  $k$ 

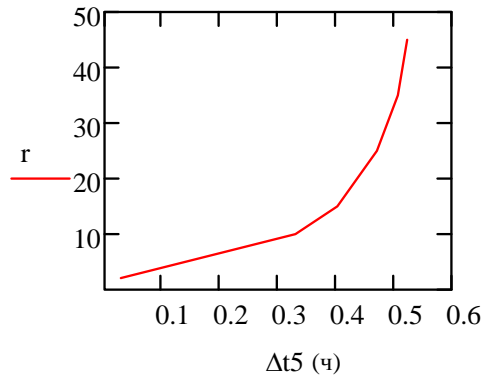
Зауважимо, що в цілому підвищення рівня професійної підготовки учасників проекту зменшує час, необхідний на обговорення проблем, до 4 хвилин (в залежності від вхідних даних), що наочно ілюструє графік рис. 3.

Так на рис. 3. приведена крива графіка залежності часу обговорення проблем, що виникли під час роботи  $T_{об}$  від коефіцієнта  $k$  в умовах, коли  $\sum_{i=1}^{\bar{N}} (a_i) = 2$  хв., та  $\sum_{i=1}^{\bar{N}} \frac{b_i}{n_i} = 1$  хв.

Рисунок. 3. – Графік залежності  $T_{об}$  від коефіцієнта  $k$ 

Досліджуємо взаємовплив показників, використовуваних в аналітичному вираженні для розрахунку часу на постановку завдань до наступного мітингу  $T_{пост}$ .

На рис. 4. приведен графік залежності відхилення часу  $\Delta t_i$  на постановку  $i$ -го завдання до наступного SCRUM-мітингу від усередненого коефіцієнта складності завдань  $r$ , виконаний за умови, що  $\bar{N}$  – кількість завдань до наступного SCRUM-мітингу дорівнює 6.

Рисунок. 4. – Графік залежності відхилення часу  $\Delta t_i$  від усередненого коефіцієнта складності завдань  $r$ 

Даний графік наочно ілюструє, що збільшення усередненого коефіцієнта складності завдань в 4 рази призводить до збільшення відхилення часу в 1,5 рази. Слід зауважити, що відповідно до виразу 3.8. мож-

на спостерігати аналогічну залежність часу на постановку завдань до наступного мітингу  $T_{пост}$  від коефіцієнта складності завдань.

Оцінимо ступінь впливу коефіцієнтів  $k$  і  $c$  на загальний час SCRUM-мітингу.

На рис. 5. представлений графік залежності часу ініціації розробки ПЗ  $T_u$  від показників  $k$  і  $c$  в умовах коли тимчасові витрати на звіт про виконану роботу з моменту попереднього SCRUM-мітингу  $T_{om}=50$ хв,  $a_i = \{20,30,\dots,70\}$  с,  $b_i = \{10,20,\dots,60\}$  с,  $\bar{n}_i = 6$ ,  $\bar{N} = 6$ ,  $v = 0,6$ ,  $\hat{n}_g = 2$ .

Як видно з цього графіка підвищення коефіцієнтів  $k$  і  $c$  в зазначених вище розмірах зменшує час проведення етапу ініціалізації процесу розробки ПО (за графіком на хвилину).

Слід зауважити, що представлені результати мають більше якісний характер. Це викликано тим, що величини  $\Delta t_j$ ,  $\Delta t_i$ ,  $\Delta t_i$ , а також  $\Delta t_{i,j}$  можуть мати як додатне, так і від'ємне значення. Крім того обрані емпіричним шляхом значення коефіцієнтів мають скоріше якісний ніж кількісний характер, і можуть варіюватися в залежності від певної експертної оцінки. Крім того, при моделюванні не враховувалося, що в командах на даний час не задіюються фахівці безпечного програмування і тестування безпеки. А це в свою чергу суттєво знижує середній рівень підготовки учасників команди, і відповідно коефіцієнт  $k$  може бути істотно нижче (менше 0,01) зазначеного в роботі діапазону значень.

Наступним етапом після ініціалізації є етап реалізації функціонала ПЗ. Розробимо модель, що формалізує цей етап.

## 2. Математична модель етапу реалізації функціонала ПЗ

Проведені аналіз літератури [3-7] і дослідження показали, що в даний час існує ряд підходів до математичної формалізації етапу реалізації функціонала ПЗ. В абсолютній більшості ці моделі мають на увазі найпростіший варіант - лінійну оцінку часу, необхідну для реалізації проекту. З одного боку, це спрощує модель, а з іншого боку дає можливість внесення змін (уточнень) у разі виникнення додаткових чинників, що впливають на точність результатів моделювання.

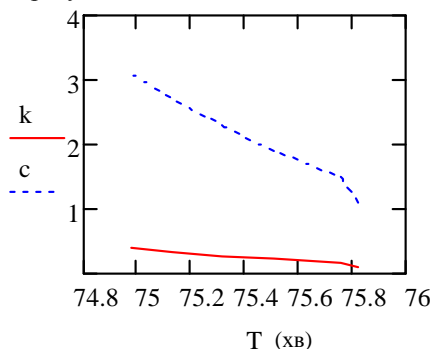


Рисунок 5. – Графік залежності часу  $T_u$  від показників  $k$  і  $c$

Скористаємося підходом, описаним в роботі [3], де модель оцінки часових витрат на реалізацію функціонала ПЗ пропонується обмежити трьома параметрами: складністю виконання, важливістю точності обчислень і новизною. Однак, як було зазначено в попередньому розділі нехтування чинником безпеки ПЗ в значній мірі погіршує його якість. Тому, більшість фірм-розробників ПЗ для врахування фактора безпеки виділяють додаткові ресурси та сили. Виходячи з цього вдосконалимо математичну модель етапу реалізації функціонала ПЗ, шляхом врахування фактору безпечного кодування ПЗ.

В цьому випадку час, необхідний на реалізацію функціонала ПЗ можна представити у вигляді виразу:

$$T_{реал} = K_{рез} \sum_{\ell} (K_{скл} D + K_{важл} \Pi_{важл} + K_{нов} \Pi_{нов} + K_{без} G), \quad (8)$$

де  $K_{рез}$  – коефіцієнт резервного часу, пов'язаний з урахуванням різних ризиків проекту, середнього часу затримки виконання завдань в команді. В ідеальному випадку він може бути прийнятий за 1, але на практиці змінюється в межах від 1,3 до 1,5;  $K_{скл}$  – коефіцієнт складності завдання;  $D$  – складність завдання, виражена в людино-годинах;  $K_{важл} \Pi_{важл}$  – коефіцієнт та параметр важливості точності обчислень. У будь-яких програмах, зокрема, орієнтованих на фінансові операції, надзвичайну важливість набуває

точність обчислень. Відомий випадок, коли через можливої помилки в одному з молодших розрядів при обчисленнях з рухомою крапкою компанія Intel проводила кампанію з вилучення своїх процесорів;  $K_{нов} \Pi_{нов}$  – коефіцієнт та параметр новизни рішення;  $K_{без} G$  – коефіцієнт та параметр безпеки ПЗ. Коефіцієнт безпеки визначається уповноваженою організацією, яка проводить тестування ПЗ на безпеку, а параметр безпеки визначається часом на реалізацію функціональності, відповідним вимогам грифа таємності ПЗ;  $\ell$  – число вимог до системи.

Слід зауважити, що число факторів та ступінь їх впливу на проект не є жорстко заданими, однак для конкретного проекту можуть бути виведені аналітично. Дані фактори можна як апіорно, так і апостеріорно закладати в параметри складності завдання або новизни рішення. Наприклад, час додавання в проект додаткових ідентичних SQL-запитів лінійно залежить від числа запитів (підвищується складність), а необхідність вивчення раніше не застосовуваної технології в ході виконання проекту (в тому числі і правил безпечного кодування) може привести до експоненціального зростання часу виконання поставленого завдання.

Оцінимо показник часу необхідного на реалізацію функціонала ПЗ відповідно до виразу 8.

На рис. 6. представлений графік залежності часу реалізації функціонала ПЗ  $T_{реал}$  від коефіцієнта резервного часу  $K_{рез}$ , отриманий за умови, що  $K_{скл}$  варіюється від 2 до 45,  $D$  – складність завдання дорівнює  $D = \{40, 45, \dots, 65\}$  люд-год,  $K_{важл} = \{1, 2, \dots, 6\}$ ,  $\Pi_{важл} = \{2, 4, \dots, 12\}$  люд-год,  $K_{нов} = \{1, 2, \dots, 6\}$ ,  $\Pi_{нов} = \{2, 3, \dots, 7\}$  люд-год,  $K_{без} = \{1, 2, \dots, 6\}$ ,  $G = \{6, 8, \dots, 16\}$  люд-год.

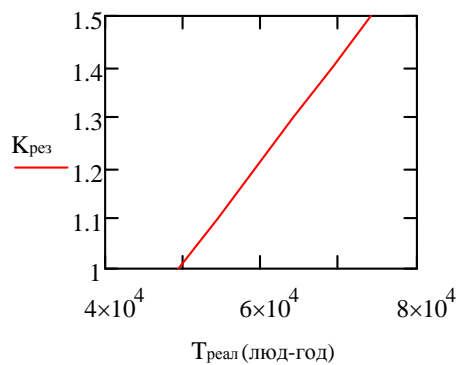


Рисунок. 6 – Графік залежності часу реалізації функціонала ПЗ  $T_{реал}$  від коефіцієнта резервного часу  $K_{рез}$

Як видно з графіка залежність часу реалізації функціонала ПЗ  $T_{реал}$  від коефіцієнта резервного часу  $K_{рез}$  досить велика. Так, при  $K_{рез} = 1$ ,  $T_{реал} = 4,944 \times 10^4$  люд-год. А при  $K_{рез} = 1,3$ ,  $T_{реал} = 6,427 \times 10^4$  люд-год, що в 1,3 раз більше попереднього результату. В цілому, отримані результати часу реалізації функціонала ПЗ представлені в табл. 1.

Слід зауважити, що в цій же таблиці (стовпець 3) наведені значення часу необхідного на реалізацію функціонала ПЗ без урахування показників безпечного програмування ( $K_{без}$ ,  $G$ ) –  $T_{реал}^*$ .

Табл. 1.- Порівняльні результати часу реалізації функціонала ПЗ

$K_{рез}$	$T_{реал}$	$T_{реал}^*$
1	4.944*10 <sup>4</sup>	4.784*10 <sup>4</sup>
1.1	5.438*10 <sup>4</sup>	5.263*10 <sup>4</sup>
1.2	5.933*10 <sup>4</sup>	5.741*10 <sup>4</sup>
1.3	6.427*10 <sup>4</sup>	6.22*10 <sup>4</sup>
1.4	6.922*10 <sup>4</sup>	6.698*10 <sup>4</sup>
1.5	7.416*10 <sup>4</sup>	7.177*10 <sup>4</sup>

Як видно з цієї таблиці нехтування урахуванням показників безпечного програмування занижує прогнозований час в 1,033 рази, і це з урахуванням того, що для вхідних даних показників безпечного програмування ( $K_{без}$ ,  $G$ ) взяті досить низькі значення. На практиці ці значення можуть бути більше і, відповідно, прогнозований час так само збільшується.

### Висновки

1. Розроблено комплекс математичних моделей процесу розробки програмного забезпечення, що включає моделі ініціалізації та реалізації функціоналу ПО.
2. Розроблено математичну модель етапу ініціалізації процесу розробки ПО, що заснована на концептуальних положеннях Agile. Це дозволило виділити ряд найбільш важливих параметрів оцінки часових витрат ініціалізації і визначити їх залежності від якісних характеристик учасників проекту.
3. Удосконалено математичну модель етапу реалізації функціоналу ПЗ, що відрізняється від відомих урахуванням показників безпечного програмування. Це дозволило підвищити точність результатів моделювання на 3%.
4. Подальший розвиток дані моделі отримують в способі масштабування існуючої методології розробки з урахуванням вимог безпеки програмного забезпечення. Це дозволить підвищити безпеку проекту і забезпечувати як швидке зростання функціоналу, так і прийнятний рівень якості сервісу.

### Список літератури

1. Veyder Maykl Tomas Instrumenty berezhlyvoho vyrobnytstva. Mini-posibnyk z vprovadzhennya metodyk berezhlyvoho vyrobnytstva / Maykl Tomas Veyder - Pabliher, 2012 125 s.
  2. Hmurman V.YE. Teoriya umovirnostey i matematychna statystyka / V.YE. Hmurman. - M .: Vyshcha shkola, 2003. - 479 s.
  3. Kaner Sem Testuvannya prohramnoho zabezpechennya. Fundamental'ni kontseptsiiy menedzhmentu biznes-dodatkov / S.Kaner. - K.: DyaSoft, 2001. - 544 s.
  4. Kniberg Henrik Scrum and XP from the Trenches - 2nd Edition / Henrik Kniberg - InfoQ 2015 - 94 c.
  5. Makkonnell S. Skil'ky koshtuye prohramnyy proekt / S. Makkonnell - Piter, Rosiys'ka Redaktsiya, 2007 - 304 s.
  6. Politsyn S. A. Pidkhody do obchyslennya chasovoyi zatpat na ppoektov v sfepe pazpobotky prohrammnoho zabezpechennya na osnovi vykorystannya ppetsedentov / S.A. Politsyn // Prohramna inzheneriya №7 2011 S.9-14
  7. Semenov. S.G. Zashchita danyh v kompyuterizirovanyh systemah upravleniya (monographiya) [Data protection in computerized control systems (monograph)] / S.G. Semenov, V.V. Davydov, S.Yu. Gavrilenko. – LAP LAMBERT ACADEMIC PUBLISHING GmbH & Co. KG, Germany, 2014.– 236 p.
  8. Ruby Sam Agile Web Development with Rails / Sam Ruby, Dave Thomas, David Heinemeier Hansson - The Pragmatic Bookshelf Rik: 2011, 2011 - 480 s.
- Статья надійшла: 13.11.2017.

### References

- Семенов Сергій Геннадійович**, доктор технічних наук, Національний технічний університет «Харківський політехнічний інститут», вулиця Кирпичова, 2, Харків, Україна, 61002, Завідувач кафедри обчислювальної техніки та програмування; ORCID ID: 0000-0003-4472-9234
- Кассем Халіфе**, аспірант, Національний технічний університет «Харківський політехнічний інститут», вулиця Кирпичова, 2, Харків, Україна, 61002.