

ТЕМА 17. МЕТРИКИ ЯКОСТІ ПРОГРАМНИХ ПРОЕКТІВ

Розглянемо метрики, які слід використовувати у всіх проектах зі створення ПЗ. Їх можна розділити на метрики керування та якості.

Показники якості керування:

- робота і прогрес (робота, що виконана на конкретний момент);
- фінансовий стан (витрати, понесені на конкретний момент);
- динаміка найму персоналу (зміни у складі персоналу, що відбулися на конкретний момент).
- середовище розроблення (Software Engineering Environment);
- використання системних ресурсів (System Resource Utilization). Зазначені метрики можуть використовуватися на етапах планування та контролю проектів та інших завдань керування або використовуватися в якості параметрів керування штатної ERP системи.

Метрика «використання системних ресурсів» – визначає відсоток цільових комп'ютерних ресурсів, що використовуються системою.

«Середовище розроблення» – міра здатності виробника розробляти програмне забезпечення високої якості. Дана метрика може бути виражена в термінах моделі «Software Acquisition Capability Maturity Model» (SA – CMM).

Показники якості розробки:

- інтенсивність змін і стабільність (інтенсивність змін на визначений момент);
- дефекти та коефіцієнт дефективності (середня кількість дефектів на одну зміну на цей момент);
- доробки та адаптованість (середній обсяг доробок на одну зміну на цей момент);
- середній час напрацювання на відмову (Mean time between failures, MTBF) і завершеність (рівень дефектів на цей момент).
- складність інтерфейсів і інтеграції (complexity of interfaces and integration) – метрика, яка вимірює ступінь складності інтерфейсу або додаткового програмування необхідного для інтеграції компоненти в систему, які потрібні для тестування, налагодження та супроводження, яке компенсує втрату якості;

- тестове покриття (test coverage) – метрики «test coverage» вказують ступінь повноти різних типів тестування;
- надійність (reliability) – метрика, що оцінює ймовірність роботи системи без відмов. Дана метрика може бути отримана в рамках традиційного підходу;
- профілі помилок (fault profiles) – метрика, вимірююча кумулятивне число виявлених помилок;
- ступінь задоволення потреб замовника (customer satisfaction) – метрика, що оцінює ступінь відповідності програмного забезпечення очікуванням та вимогам замовника. Дана метрика може бути оцінена перед поставкою на етапі дослідної експлуатації на основі прогнозуючих параметрів.

1. Оцінювання якості показників керування

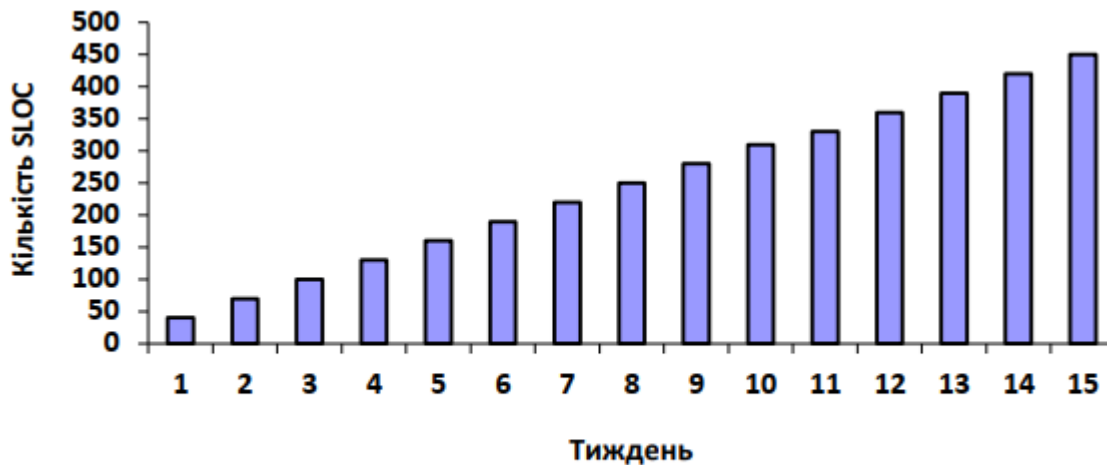
Розглянемо більш доцільно три фундаментальні набори метрик керування: робота та прогрес, фінансовий стан і прогрес у наймі персоналу. Вивчаючи ці аспекти, менеджери проекту можуть оцінити, чи вкладається проект у рамки бюджету та термінів.

1.1. Оцінювання якості роботи та прогресу

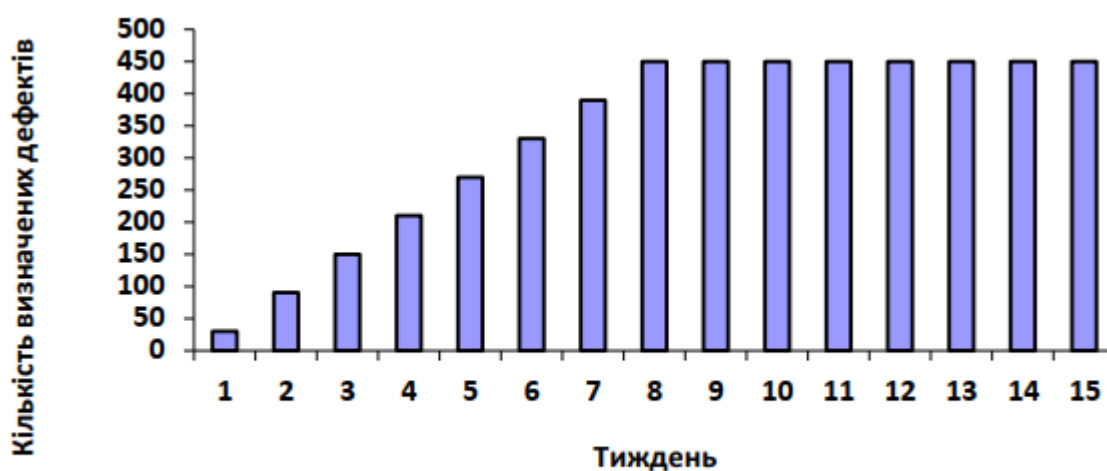
Різні види діяльності при ітераційному розробленні проекту можуть бути виміряні за допомогою визначення обсягу запланованої роботи, вираженій в об'єктивних одиницях виміру, з подальшим відстеженням прогресу (обсяг роботи, що виконана на цей момент) по відношенню до плану. У кожній великій команді в організації повинен існувати принаймні один аспект прогресу, щодо якого проводяться вимірювання.

Наприклад, для архітекторів це кількість розглянутих варіантів використання, для розробників – кількість рядків коду (SLOC) (рис.1), для тестувальників – кількість відкритих SCO, час (в годинах) виконаного тестування та відповідність критеріям оцінки, для менеджера проекту – кількість пройдених контрольних точок.

На рис. 1 а відображені кількість розроблених SLOC за визначений період часу. На рис. 1 б – кількість визначених дефектів. Оскільки ці показники залежать від етапу розробки, то для подальшого аналізу рекомендується використовувати методи експоненціального згладжування.



а)



б)

Рисунок 1 – Очікуваний прогрес типового проекту з трьома основними версіями, а) кількість SLOC, б) кількість дефектів

Також показником керування можуть слугувати функціональні точки (Function Points, FP) та їх похідні і різновиди, які обчислюються на базі не вихідного коду, а вимог користувачів, специфікацій, описів прецедентів (наприклад, точки прецедентів, Use Case Points, UCP) та ін. Відповідно до назви, функціональними точками вважається ряд функцій, що розробляються на основі специфікації вимог. У більшості випадків головне призначення цієї групи метрик, незалежно від способу їх обчислення, полягає в тому, щоб оцінити обсяг робіт за проектом, і, відповідно, бути основою для таких показників, як вартість і тривалість його реалізації.

1.2 Оцінювання фінансового стану проекту

Для контролю над проектом потрібно постійно обчислювати витрати протягом всього життєвого циклу. Використання метрик роботи та прогресу може дати більш об'єктивну оцінку технічного прогресу для її порівняння з фінансовими витратами. Для ітераційного процесу розроблення є важливим детально планувати найближчі дії (зазвичай для тимчасового інтервалу менше шести місяців) і скласти плани подальших дій у вигляді грубих оцінок, які повинні уточнюватися по мірі того, як минає інтервал часу поточної ітерації і стає критичним планування наступної ітерації.

Відстеження фінансового прогресу зазвичай приймає форму, характерну для конкретної організації. Єдиним загальним підходом до вимірювання фінансового стану є застосування системи придбаної вартості, що дозволяє детально визначити витрати часу. Його недоліком при використанні для проектів по створенню ПЗ традиційно є неможливість об'єктивної і точної оцінки технічного прогресу (відсотка завершеної роботи). Це властиво стадії проектування системи проекту. Проте системи придбаної вартості довели свою ефективність на стадії виробництва, для якої характерні правильне відстеження реальних показників порівняно з планами та передбачувані результати. Інші метрики забезпечують основу для отримання докладних і реалістичних кількісних даних по зворотному зв'язку для планування та звірки, особливо на стадії виробництва, де фінансові та часові витрати є найвищими.

Сучасні процеси створення ПЗ легко піддаються вимірюванню фінансового стану за допомогою підходу придбаної вартості. Основними параметрами системи, які звичайно обчислюються в грошових одиницях, є наступні:

- план вартості: вид кривої планованих для проекту витрат щодо планованих термінів. Для більшості проектів зі створення ПЗ (та інших трудомістких проектів) вид цієї кривої повторює вигляд кривої найму персоналу;
- реальний прогрес: технічне виконання щодо планованого прогресу, який лежить в основі кривої фінансових витрат. У проекті, який нормально розвивається, реальний прогрес невідступно слідує за запланованим прогресом;
- реальна вартість: крива реальних витрат на проект для реального графіка виконання. У проекті, що нормально розвивається, ця крива невідступно слідує за планованою кривою;

- придбана вартість: визначається вартістю реального прогресу;
- неузгодженість витрат: різниця між реальними витратами та набутою вартістю. Позитивні значення відносяться до ситуацій, пов'язаних з перевитратою бюджету; негативні значення відповідають ситуаціям, пов'язаним з економією бюджету;
- неузгодженість термінів: різниця між запланованими витратами і придбаною вартістю. Позитивні значення відносяться до ситуацій, пов'язаних з відставанням від графіка; негативні значення відповідають ситуаціям, пов'язаним з випередженням графіка.

На рис. 2 надається графічна уява цих параметрів і приводиться простий приклад стану проекту.

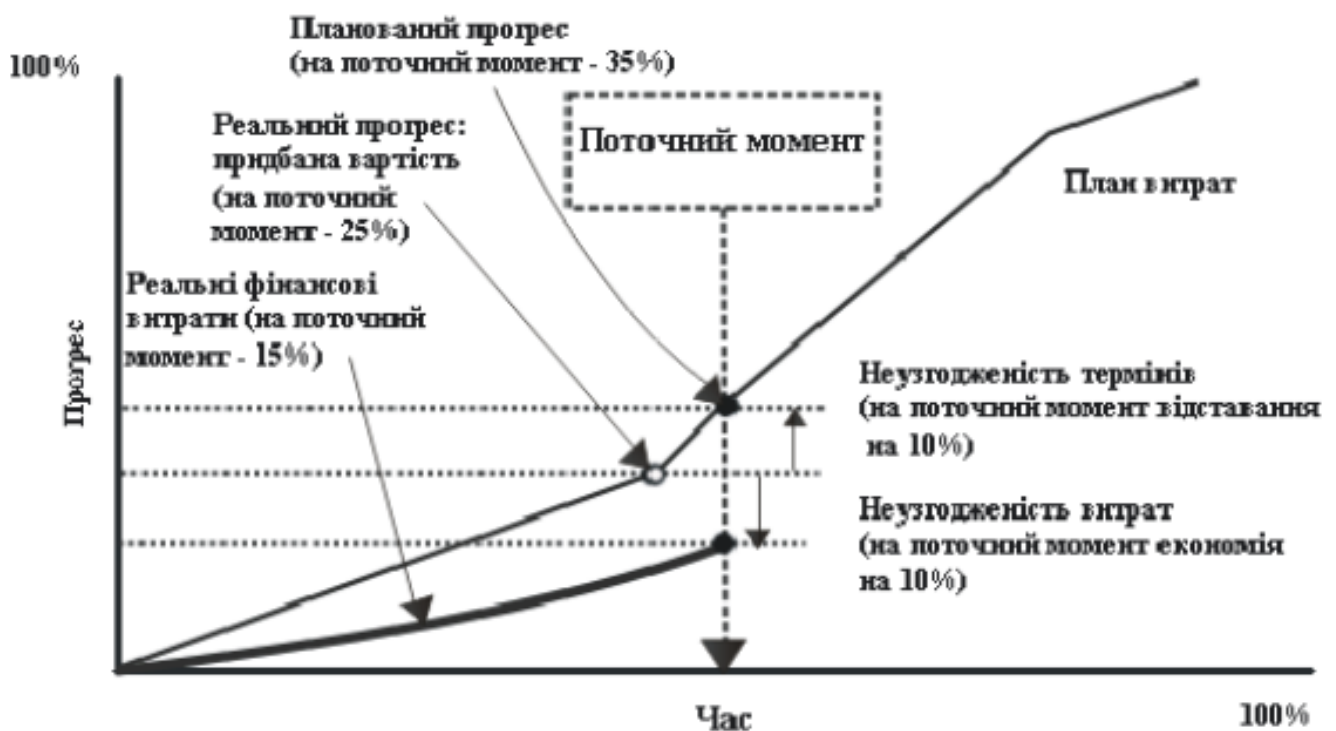


Рисунок 2 - Основні параметри системи придбаної вартості

Основною метою інших основних метрик є забезпечення команд з керування та розроблення більш об'єктивним підходом до оцінки реального прогресу з найбільшою точністю. З усіх параметрів системи придбаної вартості реальний прогрес є найбільш суб'єктивною оцінкою. Оскільки більшість менеджерів точно знають, скільки витрат вони виробили і яка частина часу використана, то розбіжності при виконанні точних оцінок фінансового благополуччя стосуються правильності оцінки реального прогресу.

1.3 Оцінювання найму персоналу

Ітераційна розробка повинна спочатку вестися невеликою командою до тих пір, поки всі ризики у вимогах і архітектурі не будуть дозволені відповідним чином. Залежно від суміщення ітерацій та інших характерних для даного проекту обставин, штатний розклад може змінюватися. Типова для виконання окремих одноразових завдань з розроблення (наприклад, для побудови корпоративної інформаційної системи) зміна чисельності задіяних у проекті співробітників наведена на рис. 3. Розумно очікувати, що група супроводу для такого роду проектів виявиться менше групи розроблення. При створенні комерційного продукту розміри груп підтримки та розроблення можуть виявитися однаковими. Коли довгоживучі продукти, що постійно піддаються змінам в рамках постійного поліпшення якості, супровід є всього лише постійним створенням нових поліпшених версій.

Початкова стадія	Прояснення	Конструювання	Введення до дії
Робота: 5% Терміни: 10%	Робота: 20% Терміни: 30%	Робота: 65% Терміни: 50%	Робота: 10% Терміни: 10%

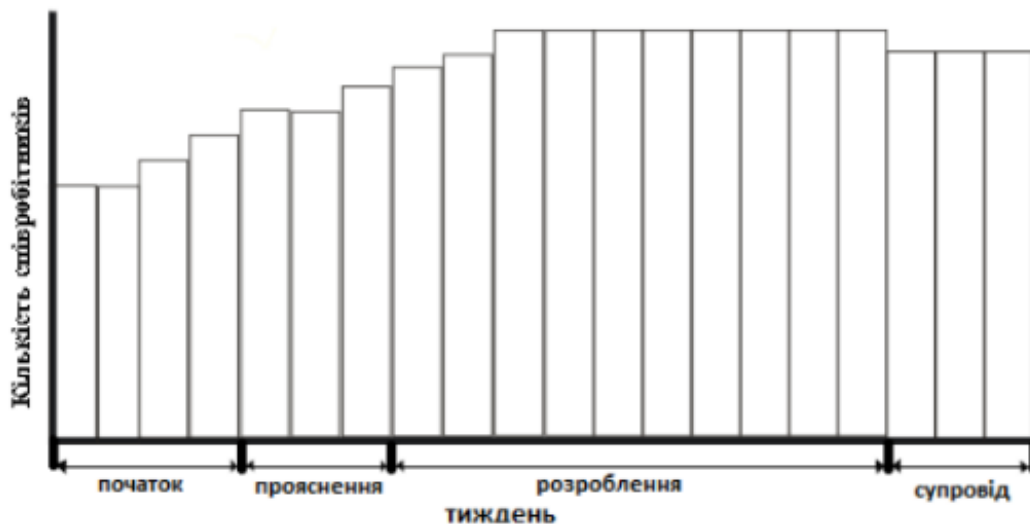


Рисунок 3 – Типова зміна чисельності співробітників

Відстеження реального найму співробітників у порівнянні з запланованим є необхідним параметром керування, що легко розуміється. Існує ще один важливий для керування показник зміни рушійної сили проекту – відношення співробітників, що

звільнилися, до знову прийнятих. Збільшення штату може уповільнити загальний прогрес проекту, оскільки постійно працюючим співробітникам доведеться витратити продуктивний час на введення нових співробітників до курсу справи. Низький відсоток звільнення хороших співробітників є ознакою успіху. Сильною мотивацією для розробників є можливість привести щось у дію; це постійна тема, що лежить в основі ефективного ітераційного процесу розроблення. Якщо така мотивація відсутня, хороші розробники йдуть до іншого джерела. Зростання незапланованого звільнення людей – тобто кількості людей, які передчасно залишають проект, – є одним з найбільш наочних показників того, що цей проект чекають проблеми. Причини передчасного звільнення можуть бути різними, але зазвичай це незадоволеність персоналу методами керування, відсутність командної роботи або висока ймовірність провалу в процесі досягнення поставлених цілей.

2. Оцінювання показників якості розробки

Чотири показника якості засновані насамперед на вимірі змін у ПЗ при зміні базових даних, що мають відношення до розроблення (таких, як проектні моделі та вихідний код).

2.1 Оцінювання інтенсивності змін і стабільності

Загальна інтенсивність змін – це особливий показник прогресу та якості. Інтенсивність змін обчислюється як кількість запитів на внесення змін до ПЗ, відкритих і закритих протягом усього життєвого циклу (рис. 7.4). Цей параметр може визначатися в залежності від типу внесених змін у розрахунок на одну версію, на всі версії, на одну команду, на один компонент, на одну підсистему і т.д. Розглянутий спільно з метриками роботи та прогресу, він дозволяє оцінити стабільність ПЗ і його рух у бік стабільності (чи нестабільності). Стабільність визначається як відношення між відкритими та закритими SCO. Інтенсивність змін, віднесена до графіка версії, дає уявлення про передбачуваність графіка робіт, що є основною цінністю даної метрики та показником того, наскільки добре йде процес.

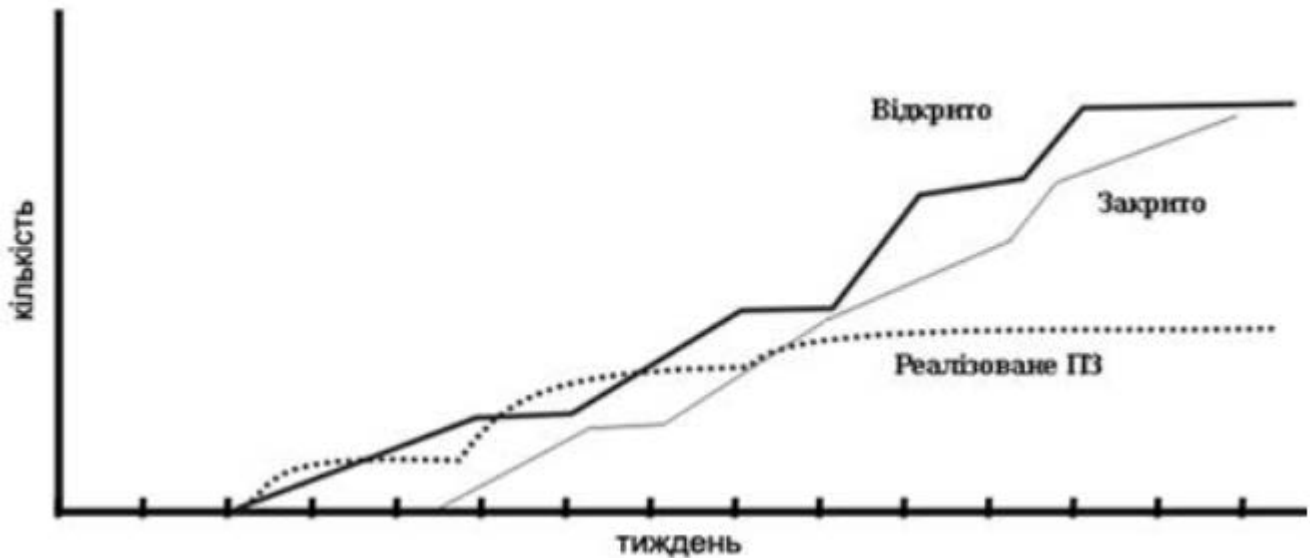


Рисунок 4 – Очікувана стабільність протягом життєвого циклу благополучного проекту

Наступні три метрики якості мають більше відношення до якості продукту.

2.2 Дефекти та коефіцієнт дефективності

Дефективність визначається як середня міра змін, що являє собою обсяг базового ПЗ, що вимагає доробки (може виражатися в SLOC, функціональних точках, компонентах, підсистемах, файлах і т.п.). Коефіцієнт дефективності – це тенденція зміни середньої кількості дефектів зі зміною часу. Для успішного проекту очікувана тенденція – зменшення або стабільність (рис. 5).

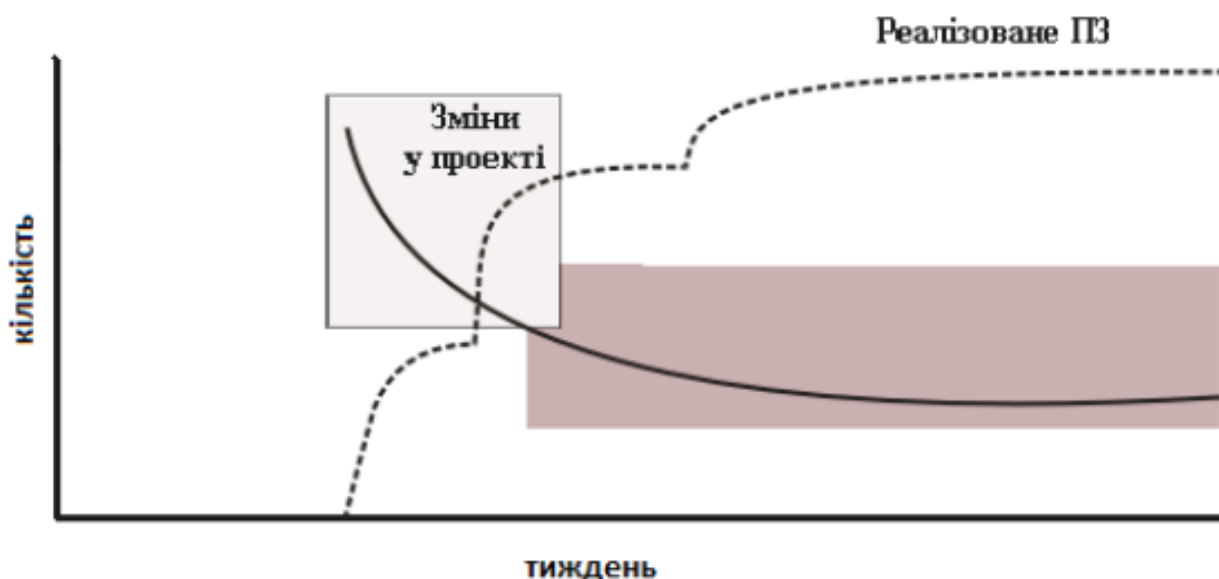


Рисунок 5 – Очікуваний коефіцієнт дефективності протягом життєвого циклу благополучного проекту

Цей показник дозволяє зрозуміти, який характер носять зміни ПЗ – доброякісний або злоякісний. У разі зрілого процесу ітераційного розроблення зміни на ранніх етапах зазвичай призводять до більшого обсягу відбракування, ніж на пізніших. Тенденція до збільшення дефектів з часом ясно вказує на те, що супровідність продукту викликає підозри.

2.3 Оцінювання вартості доробки і адаптованості

Доробка визначається як середня вартість внесення змін, до якої відносять витрати на аналіз, ухвалення рішення, повторне тестування всіх змін в основі ПЗ. Адаптованість визначається як тенденція зміни кількості доробок в залежності від часу. Для якісного проекту очікувана тенденція – зменшення або стабільність (рис.6).

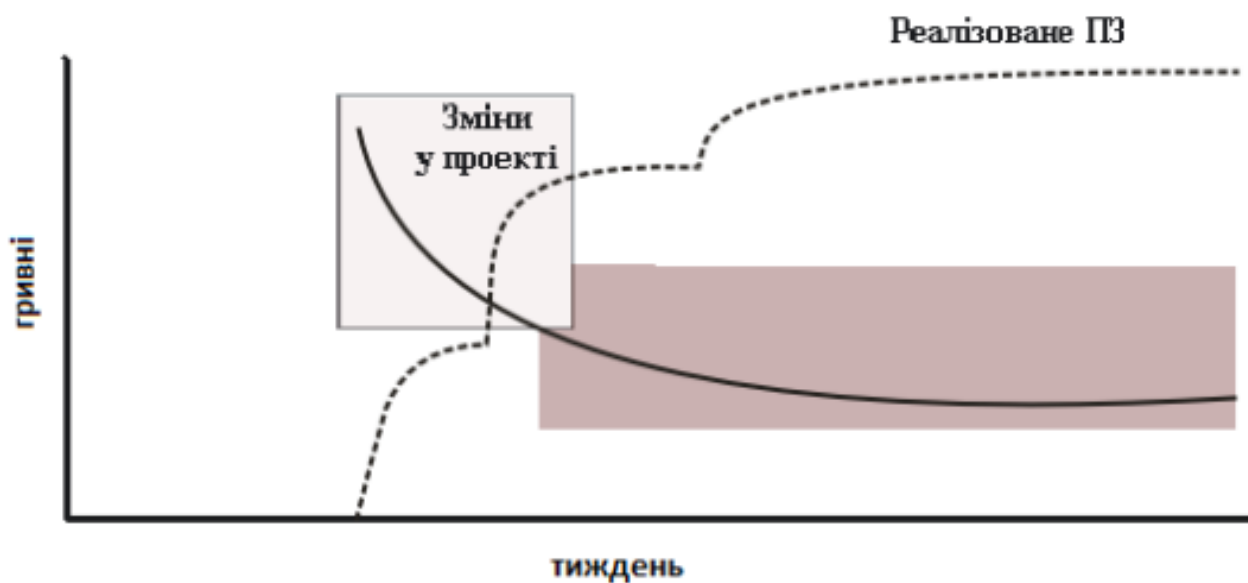


Рисунок 6 – Очікувана адаптованість протягом життєвого циклу благополучного проекту

Не всі зміни однакові. Деякі з них можуть бути внесені за одну людино-годину, інші вимагають декількох людино-тижнів. Ця метрика дозволяє вимірювати обсяг доробок. У сформованому ітераційному процесі зміни на ранніх етапах (зміни в архітектурі, які впливають на велику кількість компонентів і людей), як правило, вимагають великих доробок, ніж на пізніх (зміни при розробці, які зазвичай обмежуються одним компонентом або людиною). Тенденції до збільшення доробок з часом ясно вказують на те, що супровід продукту викликає підозри.

2.4 Оцінювання середнього часу до відмови

MTBF – це середній час використання ПЗ між двома відмовами. У загальному випадку MTBF можна обчислити, поділивши час тестування на кількість SCO типів 0 і 1. Завершеність визначається як тенденція зміни MTBF з плином часу (рис. 7).

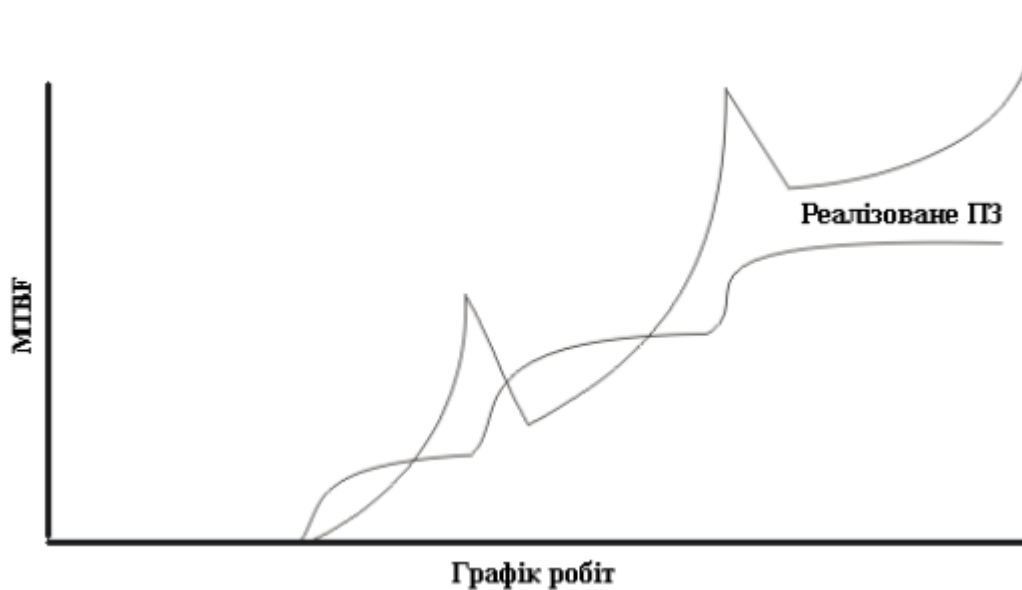


Рисунок 7 – Очікувана завершеність протягом життєвого циклу благополучного проекту

Раннє визначення завершеності вимагає створення ефективної інфраструктури для тестування. Традиційні підходи до тестування монолітних комп'ютерних програм приділяють основну увагу досягненню повного тестового покриття кожного рядка коду, кожної гілки програми і т.д. У сучасних розподілених і розбитих на компоненти програмних системах таке вичерпне тестове покриття можливо досягти тільки для окремих компонентів. Системи компонентів більш ефективно тестуються з використанням статистичних методів. Відповідно параметр «завершеність» визначає статистику за час використання, а не охоплює весь продукт.

Помилки в ПЗ можуть бути розбиті на дві категорії: детерміновані та недетерміновані. Перші – це клас помилок, які завжди виявляються, коли ПЗ використовується певним чином. Такі помилки в більшості випадків є наслідком помилок кодування, а необхідні зміни зазвичай обмежуються одним компонентом. Помилки Інший клас – це відмови ПЗ, що відбуваються випадково з деякою ймовірністю при настанні певної ситуації. Ці помилки майже завжди виникають через помилки, допущені при проектуванні (і часто потребують внесення змін до безлічі компонентів), і звичайно

не відтворюються навіть у тих випадках, коли ПЗ використовується одним і тим же способом. Для забезпечення адекватного тестового покриття та виправлення статистично значущих помилок другого класу потрібно екстенсивне статистичне тестування за реальними та випадковими сценаріями використання.

Таблиця 7.1 – Стандартні зразки зміни метрик протягом життєвого циклу для RUP

	Початок	Проектування	Розробка	Впровадження
Прогрес	5%	25%	90%	100%
Архітектура	30%	90%	100%	100%
Програми	<5%	20%	85%	100%
Фінальний стан	Низькі	Помірні	Високі	Високі
Робота	5%	25%	90%	100%
Час	10%	40%	90%	100%
Динаміка у наймі персоналу	Невелика команда	Зростаюче	Стале	Змінне
Стабільність	Змінна	Помірна	Помірна	Висока
Архітектура	Змінна	Помірна	Стабільна	Стабільна
Програми		Змінна	Помірна	Стабільна
Коефіцієнт дефективності	50% – 100%	25% – 50%	<25%	5% – 10%
Архітектура	>50%	>50%	<15%	<5%
Програми	>80%	>80%	<25%	<10%
Адаптованість	Різна	Різна	Якісна	Якісна
Архітектура	Різна	Помірна	Якісна	Якісна
Програми		Різна	Помірна	Якісна
Завершеність	Прототип	Незначна	Застосовна	Повна
Архітектура	Прототип	Застосовна	Повна	Повна
Програми	Прототип	Незначна	Застосовна	Повна

Традиційні програмні продукти, в яких єдина програма виконується на єдиному процесорі, як правило, містять тільки помилки 1 класу. Сучасні розподілені системи з безліччю взаємодіючих компонентів, що виконуються в мережі процесорів, виявляються уразливими до помилок 2 класу, які складно виявляти, аналізувати та виправляти. Кращий спосіб зробити програмний продукт досконалим полягає у створенні з самого початку такої інфраструктури тестування, яка допускає виконання випадкових сценаріїв використання на ранніх стадіях життєвого циклу і постійно збільшує ширину і глибину сценаріїв для охоплення критичних по надійності компонентів.

По мірі свого створення базові версії ПЗ повинні постійно перевірятися за допомогою сценаріїв тестування. На основі цього тестування можна визначити

параметри надійності. Осмислена оцінка завершеності продукту може бути виконана за рахунок максимізації часу тестування (з використанням незалежного середовища тестування, автоматизованих регресійних тестів, випадкового статистичного тестування, тестування в години, які слідують за великими навантаженнями, і т.д.). Такий підхід надає потужний механізм для автоматизації тестування на самих ранніх стадіях життєвого циклу. Цей метод може бути застосований також для моніторингу продуктивності та вимірювання надійності.

3. Використання метрик ПЗ

Будь які вимірювання корисні, але висновки робить особа приймаюча рішення (менеджер проекту, керівник, команди). Вимірювання тільки дають дані, які допомагають правильно ставити питання, розуміти контекст і приймати об'єктивні рішення. Оскільки природа проектів зі створення ПЗ надзвичайно динамічна, то можливість виконувати такі вимірювання повинна існувати в будь-який час і бути застосовна до різних частин мінливого продукту (версія, компонент, клас). Виміри повинні проводитися таким чином, щоб можна було оцінити тенденції замість показників з часом. На практиці така ситуація досяжна тільки в тих проектах, де підтримується доступ в онлайн-режимі до метрик, що автоматично визначаються, як до побічного продукту середовища розроблення/інтеграції.

У таблиці 2 зведені розглянуті метрики. Кожна з них має два виміри: статичне значення, що використовується в якості мети, і динамічна тенденція, що застосовується для досягнення мети. Значення метрик дають уявлення тільки в одному вимірі; з точки зору керування процесом важливішими виявляються тенденції їх зміни. Тенденції зміни метрик у часі дозволяють зрозуміти, в яких напрямках змінюються продукт і процес. Ітераційному процесу притаманне внесення змін, а вимір цих змін є важливим аспектом параметрів програми. Абсолютні значення продуктивності і якісних поліпшень вторинні до тих пір, поки не буде вирішена фундаментальна задача керування: передбачуваність витрат коштів і часу для заданого рівня якості.

Таблиця 7.2 – Сім основних метрик керування проектами

Метрика	Мета	Способи визначення
Робота і прогрес	Планування ітерацій, порівняння плану і реальних досягнень, показник керування	<i>SLOC</i> , функціональні точки, об'єктні точки, сценарії, варіанти тестування, <i>SCO</i>
Фінансовий стан	Розуміння фінансових питань, порівняння плану і реальних значень, показник керування	Щомісячні витрати, кількість використовуваних протягом місяця співробітників з повним робочим днем, відсоток витраченого бюджету
Динаміка найму персоналу	План споживання ресурсів у порівнянні з реальністю, рівень найму, рівень скорочення чисельності персоналу	Кількість нових співробітників за місяць, кількість що пішли співробітників за місяць
Інтенсивність змін і стабільність	Планування ітерацій, показник дотримання графіка	Кількість поданих <i>SCO</i> в порівнянні з кількістю закритих <i>SCO</i> за типами (0,1,2,3,4), що припадають на одну версію/компонент/підсистему
Дефекти і коефіцієнт дефективності	Досягнення мети, доробка ПЗ, показник якості	Кількість перероблених <i>SLOC</i> на одну зміну за типами (0,1,2,3,4), що припадають на одну версію/компонент/ підсистему
Доробки і адаптованість	Досягнення мети, переробка ПЗ, показник якості	Середня кількість годин, витрачених на одну зміну за типами (0,1,2,3,4), що припадають на одну версію/компонент/підсистему
<i>MTBF</i> і завершеність	Покриття/ адекватність тестування, простота використання, показник якості	Число відмов, кількість годин між відмовами при тестуванні, що припадають на одну версію/компонент/підсистему

Після того як сімейство метрик визначено, вони можуть бути використані проектом або організацією для передбачення витрат, термінів і якості виконання різних робіт у майбутньому.

Коли метрики вказують на деяку проблему, важливо вникнути в усі симптоми, щоб поставити діагноз. Метрики зазвичай відображають ефекти; для розуміння причин потрібен синтез багатьох точок зору та висновків. Наприклад, умови, необхідні для правильної інтерпретації таких ситуацій:

- мала кількість запитів на внесення змін може означати, що ПЗ досконало і вільно від помилок, а може означати, що команда, відповідальна за тестування, перебуває у відпустці;

– запит на внесення змін, що залишається відкритим протягом тривалого часу, може означати, що помилка була легко діагностована, але її виправлення потребувало значних доробок, або може означати, що її діагностика зайняла дуже багато часу, а виправлення полягало у внесенні змін в один єдиний рядок коду;

– велике зростання персоналу в поточному місяці може призвести до пропорційного збільшення прогресу, якщо новий персонал складається з навчених людей, які відразу ж приступлять до продуктивної праці. І він же може викликати уповільнення прогресу, якщо найняті ненавчені люди, яким потрібно найрізноманітніша підтримка з боку тих, хто зайнятий продуктивною працею, для введення в курс справи;

– поточні значення та поточні тенденції зміни метрик використовуються в якості параметрів керування проектом;

– реальні значення наведених метрик можуть змінюватися в широких межах в рамках проектів, організацій та областей застосування. Однак відносні тенденції зміни протягом стадій проекту повинні слідувати загальним прикладам, наведеним у таблиці 1. Стабільна організація-розробник повинна вміти описувати ті значення метрик, до яких вона прагнутиме, що є більш інформативним і точним відносно її основної діяльності і конкретних процесів.

Якісні оцінки не можуть виконуватися самими метриками, для цього потрібні інтелектуальні зусилля з боку, наприклад, менеджерів проектів по створенню ПЗ.