

ЛЕКЦІЯ 8. УПРАВЛІННЯ ВИМОГАМИ ПРИ РЕАЛІЗАЦІЇ ПРОГРАМНИХ ПРОЕКТІВ

План лекції:

1. Основи положення управління вимогами при реалізації програмних проектів.
2. Забезпечення взаємодії із замовником при реалізації програмних проектів.
3. Планування, визначення пріоритетів, оцінка ризику та контроль виконання вимог.
4. Управління вимогами на основі підходу до ітеративного створення версій при реалізації програмних проектів.
5. Інтеграція управління вимогами у загальний процес управління програмними проектами.

Питання, що виносяться на самостійне вивчення студентом (3,5 год.):

1. Управління вимогами на основі UML [8, С. 149-154 243-252, 301-312].
2. Управління вимогами в стандартах SEI SMM/CMMI та ISO 9000 [8, С. 419-425].
3. Метод FAST при визначенні вимог [7, С. 526-527]
4. Метод JAD при визначенні вимог [7, С. 527-532]

8.1. Основи положення управління вимогами при реалізації програмних проектів

Формулювання вимог – найважливіший етап у розробці ПЗ. Неправильно сформульовані вимоги є джерелом найдорожчих помилок у розробці ПЗ.

Важливе завдання на етапі формулювання вимог полягає у тому, щоб зрозуміти бажання і очікування замовника і перетворити їх у конкретні вимоги до продукту, який створюється, з урахуванням обмежень проекту і можливостей розробника.

Згідно з SEI CMM ціль формулювання вимог полягає у тому, щоб «встановити загальне розуміння між замовником та програмним проектом на основі вимог, які висуває замовник до програмного проекту».

Етапи формулювання вимог:

- визначення вимог – розуміння бажань замовника;
- аналіз та обговорення вимог – аналіз вимог та їх обговорення з метою визначення прийняттого рішення;
- специфікація вимог – формулювання однозначних специфікацій рішень;
- системне моделювання;
- атестація вимог – перевірка специфікації;
- управління вимогами.

8.2. Забезпечення взаємодії із замовником при реалізації програмних проектів

Типи вимог:

- свідомі – ті, що явно називаються замовником;
- несвідомі – ті, що не називаються замовником, наприклад, такі, про які він забуває.

Також вимоги бувають:

- функціональні (властивості продукту) – визначають, що саме має виконувати програмний продукт;
- нефункціональні – як правило, мають відношення до якості і визначають, наскільки правильно програмний продукт має виконувати те, що від нього вимагається.

Існують також архітектурні вимоги – ті, які передбачають, яким саме чином мають вирішуватися поставлені перед системою задачі.

Також бувають обмежуючі вимоги, до яких відносяться обмеження при проектуванні систем, відповідність стандартам, юридичні та організаційні питання.

При роботі із замовником важливо максимально точно визначити, що саме вимагає замовник. Інтерпретація слів замовника і переведення їх у конкретні специфікації програмного продукту є досить складною задачею.

Важливо визначати якість вимог. Якісною можна вважати таку вимогу, яку можна перевіряти. Якщо для вимоги можуть бути розроблені конкретні засоби для тестування, то це означає, що подібна вимога визначена однозначно та може бути виміряна. При цьому найважливішим фактором виступає можливість однозначного визначення того моменту, коли вимога задоволена (тобто конкретна необхідна функціональність реалізована).

При формулюванні вимог у процесі взаємодії із замовником розглядаються наступні атрибути вимог, які дозволяють досягти запланованого рівня якості:

- коректність – вимога може бути підтверджена лише замовником;
- придатність – вимога може бути задоволена з урахуванням існуючих проектних обмежень;
- необхідність – у процесі переговорів із замовником та виконавцем визначаються функціональні можливості продукту, які є ключовими для нього;
- пріоритетність – визначається пріоритет одних вимог по відношенню до інших;
- однозначність – наявність лише однієї інтерпретації, визначає простоту читання і розуміння;
- лаконічність – вимога має містити лише ту інформацію, яка необхідна для її виконання, уся інша інформація має міститися окремо;
- можливість перевірки (тестованість, вимірюваність) – відповідність програмного продукту вимозі може бути перевірено програмою чи людиною;
- завершеність – вимога сформульована таким чином, що містить усю необхідну інформацію;

- відповідність – узгодженість усіх документів, які стосуються даної вимоги;
- здатність до змін – можливість внесення змін до вимоги має розглядатися як допустимий і логічний процес;
- трасуємість – можливість відслідковування зв'язків, вимога має явно вказувати на документ, який є джерелом для неї, а її реалізація має бути явно пов'язана із вимогою.

Важливо, щоб при визначенні вимог мова йшла про специфікації і результати, а не про те, яким чином слід їх досягати. Зокрема, бажано формулювати вимоги як відповідь на питання «які?», а не «як?». Наприклад, «Які дані має використовувати та формувати система?», «Які функції має виконувати система?» та ін.

Найважливіша і найскладніша задача – знайти спільну мову із замовником і переконатися, що замовник очікує саме те, що планує реалізувати виконавець.

8.3. Планування, визначення пріоритетів, оцінка ризику та контроль виконання вимог

Методами планування, пріоритизації та визначення вимог є наступні:

- інтерв'ю;
- «мозковий штурм»;
- схеми мислення;
- метод спрощеної специфікації програмного забезпечення (Facilitated Application Specification Technique, FAST);
- сумісна розробка програмного забезпечення (Joint Application Design, JAD);
- сценарії вибору (case scenario).

Загальні рекомендації:

- в процесі визначення вимог мають приймати участь фактичні користувачі, а не сторонні люди;

- необхідно враховувати вимоги замовників і спонсорів, хоча вони не є кінцевими користувачами, однак несуть витрати по розробці системи;
- усі організатори проекту мають бути ідентифіковані, причому в процесі збору вимог мають приймати участь представники кожного типу;
- неоднозначні вимоги мають бути ідентифіковані з метою прототипування;
- до етапів розробки необхідно ідентифікувати обмеження предметної області, що відобразиться на обмеженості функціональних властивостей чи продуктивності програмного продукту;
- кожен організатор проекту є дещо необ'єктивним при оцінці корисності проекту для своєї організації;
- якщо розробляється лише програмний продукт, то має бути визначене середовище, в якому він буде експлуатуватися (архітектура комп'ютерів, операційні системи, середовище комунікацій);
- якщо розробляється система в цілому (апаратна і програмна система), то вимоги до системи в цілому мають бути визначені до визначення вимог до програмного забезпечення;
- вхідні дані для методів визначення вимог включають визначення понять, бізнес-цілі високого рівня, визначення потреб і придатності, визначення обсягу робіт та проектний план;
- результуючі дані для методів визначення вимог включають сформований згідно з пріоритетами і функціями перелік вимог, обмеження предметної області, набір сценаріїв вибору та прототипи;
- для кожної вимоги має даватися обґрунтування;
- методи визначення вимог використовуються лише за наявності усіх необхідних учасників.

Визначення пріоритетів здійснюється, починаючи з того моменту, як зібрані і зафіксовані усі вимоги до системи.

Визначення пріоритетів важливо з наступних позицій:

- можливість «пожертвувати» окремими функціями для того, щоб уникнути ризику зриву проекту в цілому;
- можливість скоротити вартість системи і строк її розробки, відмовившись від найменш важливих функцій;
- можливість якомога раніше ввести систему в експлуатацію за рахунок реалізації в першу чергу найважливіших функцій і зменшити таким чином ризик невідповідності очікувань замовника і результатів роботи системи.

Процес визначення пріоритетів вимог здійснюється наступним чином:

- визначаються критерії, що дозволяють оцінювати цінність вимог, як правило, це співвідношення витрати/вигода;
- функціональність системи групується в певні набори властивостей, які містять логічно пов'язані властивості;
- потім кожна вимога всередині кожного набору властивостей оцінюється на предмет її значимості (дуже часто виконуються проста система бальної оцінки: 3 – важлива, абсолютно необхідна вимога, 2 – важлива, але не абсолютно необхідна вимога, 1 – бажана, але не обов'язкова);
- на завершальному етапі для кожної вимоги розробники проекту та замовники вирішують для кожної вимоги можливість її негайної реалізації, можливість відстрочки та можливість відмови від розгляду вимоги взагалі.

В процесі визначення вимог здійснюється оцінка їх ризиків, найбільш ризиковані вимоги мають бути реалізовані на якомога ранніх стадіях розробки проекту.

Контроль за виконанням вимог здійснюється, як правило, за допомогою спеціального програмного інструментарію, який дозволяє

пов'язати саму вимогу з програмним кодом, який її реалізує та системою тестування, що дозволяє переконатися у тому, що вимога реалізована.

8.4. Управління вимогами на основі підходу до ітеративного створення версій при реалізації програмних проектів

Важливе завдання пріоритизації і оцінки ризику за вимогами полягає у тому, щоб підготувати вимоги до поетапної реалізації у відповідності із ітеративною моделлю життєвого циклу ПЗ.

Перша версія програмного продукту має містити ті вимоги, пріоритет яких відповідає найвищому рівню, тобто абсолютно необхідні функції. Причому до першої версії можлива розробка декількох ітерацій, починаючи з найбільш ризикованих функцій.

Якщо при розробці певної ризикованої функції були встановлені значні проблеми, які вимагають перегляду проекту, то на ранніх стадіях його виконання така процедура може врятувати проект в цілому, незначним чином вплинувши на витрати і строки виконання. У протилежному разі можливий повний провал проекту зі значними фінансовими витратами.

Пріоритет і ризикованість вимог мають переглядатися перед початком кожної нової ітерації розробки, крім того, між ітераціями можлива модифікація вимог, додавання нових та видалення існуючих.

8.5. Інтеграція управління вимогами у загальний процес управління програмними проектами

Як зазначалося раніше, етап формування вимог є найважливішим у розробці програмного проекту. З цієї причини він безпосередньо пов'язаний з іншими етапами роботи над проектом, які цілком і повністю визначаються вимогами.

Зокрема, найбільш тісними є зв'язки з наступними етапами:

- проектування – проект ПЗ цілком і повністю має бути розроблений таким чином, щоб відповідати поставленим вимогам;

- кодування – даний процес виконується з метою задоволення вимог, якщо на даному етапі встановлюється невідповідність проекту вимогам, то проект переглядається;
- тестування – головне завдання даного етапу полягає у тому, щоб пересвідчитися в повній реалізації вимоги. Для деяких вимог цей процес може бути цілком об'єктивним і виконуватися повністю автоматизовано, для інших – лише за участі людини і може мати значну долю суб'єктивізму.

Важлива задача – забезпечення трасування вимог. Кожна строчка програмного коду чи дія виконавця має бути безпосередньо пов'язана з реалізацією конкретної функціональної чи нефункціональної вимоги, якщо це не так, то вимоги слід переглянути, чи від дії необхідно відмовитися. Зв'язок має бути задокументованим і чітко відстежуватися.