

УЗГОДЖЕНІСТЬ ТА ОЦІНКА ВИМОГ



Вимоги отримані від користувачів можуть дублюватися або заперечувати один одного. Одні вимоги можуть бути незрозумілими або нереальними, інші можуть залишатися не виясненими.



По цій причині, перед тим, як вимоги попадуть у документ опису вимог, їх необхідно узгодити.



В дійсності узгодження і перевірка обґрунтованості вимог здійснюється паралельно з виявленням вимог. Після того як вимоги виявлені, вони підлягають перевірці. Для всіх сучасних методів виявлення вимог, які пов'язані з так званою *груповою динамікою*, це цілком природньо. Тим більш після виявлення вимог, вони, в будь-якому випадку повинні підлягати до детального обговорення та перевірці.

Узгодження та перевірка вимог не можуть бути відокремлені від процесу підготовки технічного завдання.

Зазвичай *узгодження вимог* базується на чорновому варіанті цього документу. Вимоги, перераховані в чорновому варіанті технічного завдання, обговорюються та при необхідності корегуються. При цьому вилучаються невірні вимоги та доповнюються новими.



Для перевірки *обґрунтованості вимог* необхідно більш повний варіант технічного завдання, в якому всі вимоги чітко ідентифіковані і класифіковані. Учасники проекту вивчають документ та проводять наради по їх формальному перегляду. *Перегляди* часто структуровані у вигляді так званих процедур *наскрізного контролю* або *інспекцій*. Перегляди являються різновидом *тестування*.

ВИМОГИ, ЯКІ ВИХОДЯТЬ ЗА РАМКИ ПРОЕКТУ

Вибір проекту в сфері інформаційних технологій та відповідно створюваній системі визначається в рамках системного планування та бізнес-моделювання. Але деталізовані взаємозв'язки між системами можуть бути розкриті тільки на етапі *аналізу вимог*.

Границі системи слід визначати на етапі аналізу вимог, для того щоб вирішувати проблему «роздуття системи» вже на ранніх етапах процесу розробки.

Для того щоб визначити, чи не виходить конкретна вимога за межі потрібного, необхідна *еталонна модель*, по відношенню до якої повинно прийматися рішення.

Історичну роль подібної еталонної моделі грала *контекстна діаграма* – високорівнева діаграма популярного методу структурного моделювання на основі діаграм потоків даних. Не дивлячись на те, що в мові UML місце цієї діаграми зайняла діаграма прецедентів, контекстна діаграма, як і колись залишається надійним методом встановлення границь системи.

Однак існують і інші причини, по яким вимога може бути розглянута як та, що виходить за рамки проекту.



Наприклад, вимоги можуть представляти велику складність для реалізації в комп'ютеризованій системі і повинно залишатися побажанням людини, яка приймає участь в процесі. Крім того, вимога може мати низький пріоритет і повинна бути виключена із першої версії системи. Деякі вимоги можуть бути також реалізовані за допомогою апаратного забезпечення або зовнішніх пристроїв і, таким чином, виявитися поза контролем програмного забезпечення.

МАТРИЦЯ ЗАЛЕЖНОСТІ ВИМОГ

Коли всі вимоги чітко ідентифіковані та пронумеровані можна сконструювати *матрицю залежностей вимог* (або *матрицю взаємодії*). В цій матриці перераховуються упорядковані вимоги, як показано в табл. 1.

Таблиця 1 Матриця залежностей вимог

Вимога	R1	R2	R3	R4
R1				
R2	Конфлікт			
R3				
R4		Перекриття	Перекриття	

Верхня права частина матриці (над діагоналлю включно) не використовується. Інші комірки вказують на те, чи перекриваються дві будь-які вимоги, суперечать один одному або незалежні один від одного.

Протирічні вимоги необхідно обговорити з замовником та при можливості переформулювати, для того щоб пом'якшити конфлікт.

Перекриваючі вимоги також повинні бути сформульовані знову.

Матриця залежності вимог

представляє собою простий, але ефективний метод знаходження протиріч та перекриттів, коли кількість вимог відносно невелика. В протилежному випадку описаний метод можна застосовувати лише після групування вимог по категоріям та порівняння в межах кожної категорії по окремої.

ВИМОГИ - РИЗИКИ І ПРІОРИТЕТИ

Усунувши суперечності і повтори вимог, необхідно провести аналіз ризиків і призначити пріоритети.

Аналіз ризиків - дозволяє ідентифікувати вимоги, які є потенційними джерелами труднощів.

Призначення пріоритетів - необхідно для того, щоб забезпечити можливість без труднощів змінити рамки проекту в разі виникнення непередбачених затримок.

Здійсненність проекту залежить від його ризикованості.

Ризик - це загроза, що заважає здійснити проект (брак фінансування, часу, ресурсів і т. д.). Ідентифікуючи ризики, менеджер отримує можливість управляти ними.

ВИМОГИ МОЖУТЬ БУТИ «РИЗИКОВАНИМИ» З РІЗНИХ ПРИЧИН. ЦІ ПРИЧИНИ ПОДІЛЯЮТЬСЯ НА ТАКІ КАТЕГОРІЇ:

- **Типовий ризик** означає, що вимога технічно важко реалізувати.
- **Ризик зниження продуктивності** означає, що реалізація вимог може сповільнити реакцію системи.
- **Ризик, пов'язаний з порушенням цілісності баз даних**, означає, що вимога важко перевірити і може виникнути суперечливість даних.
- **Ризик, пов'язаний з процесом розробки**, означає, що для реалізації вимог необхідно використовувати незвичайні методи, незнайомі розробникам (наприклад, методи формальної специфікації).
- **Політичний ризик** означає, що вимога може виявитися нездійсненним через внутрішньополітичні причини.
- **Юридичний ризик** означає, що вимога може призвести до порушення діючих законів або суперечити очікуваних змін закону.
- **Ризик, пов'язаний з мінливістю**, означає, що вимога може потенційно змінюватися або еволюціонувати протягом процесу розробки.

ПРІОРИТЕТИ ВИМОГ

В ідеалі пріоритети вимог призначаються індивідуальними замовниками в процесі їх виявлення. Потім їх погоджують на нарадах і знову змінюють після додавання до них факторів ризику.

Для того щоб виключити невизначеність і полегшити призначення пріоритетів, кількість пріоритетів має бути не великою. Зазвичай достатньо від трьох до п'яти пріоритетів.



Їх можна позначати як **«високий»**, **«середній»**, **«низький»** і **«невизначений»**.

Альтернативний перелік пріоритетів може виглядати, наприклад, так: **«істотне»**, **«корисне»**, **«важко визначити»** і **«відкладене»**.

Наведемо приклад класифікації з описом:

ВИСОКИЙ – критичні для рішення та проект не може бути реалізований без цих вимог;

середній – важливі для рішення, але можна обійтися з втратами функціональності. Можливо це може бути відкладеним до наступного релізу;

НИЗЬКИЙ – було б добре мати, але не впливає на успіх рішення. Може чекати наступного релізу.

РИЗИКИ ПРОГРАМНИХ ПРОЕКТІВ МОЖНА РОЗДІЛИТИ НА ЧОТИРИ ОСНОВНІ КАТЕГОРІЇ:

- 1. Ризики пов'язані з вимогами**
- 2. Технологічні ризики**
- 3. Ризики пов'язані з кваліфікацією персоналу**
- 4. Політичні ризики**

Це тільки основні категорії, однак, у кожному конкретному випадку можуть бути додані інші види, не розглянуті тут.

РИЗИКИ ПОВ'ЯЗАНІ З ВИМОГАМИ

Процес розробки програмного забезпечення починається з визначення вимог і варіантів використання системи. Основна проблема полягає в тому, що деякі ключові вимоги, які потрібні для реалізації системи можуть бути пропущені, оскільки користувачі можуть порахувати їх настільки очевидними, що їх навіть не потрібно згадувати. Інші вимоги не так зрозумілі розробниками. А разом створювана система буде виконувати не те, що хотіли користувачі.

Ще одна група ризиків, пов'язана з вимогами - це реалізація другорядних вимог і відкладання вимог, які можуть принести основний результат користувачам.

Основний спосіб справитися з цією групою ризиків - контролювати процес управління вимогами, створювати UML моделі варіантів використання і залучати замовників для обговорення отриманих моделей. Замовник повинен зрозуміти, навіть якщо він цього не розумів на початку, що він отримає на кожному етапі розробки і як цим можна буде користуватися.

Використання інструментальних засобів для керування вимогами, таких як RequisitePro і ClearQuest і такого інструменту для створення моделей, як, наприклад, Rational Rose.

ТЕХНОЛОГІЧНІ РИЗИКИ

Ця група ризиків об'єднує ризики пов'язані з використовуваними технологіями.

Які технології ви плануєте застосувати?

Тобто, вибрані технології відпрацьовані чи це "щойно спечені пиріжки".

Будь-яке середовище програмування містить помилки, питання в тому чи знаєте ви місця в пропонованій для розробки оболонці, які потрібно обходити і не робите ви ставку на сиру, ще не відпрацьовану технологію.

Найкращий варіант оцінити технологічні ризики - це створити прототип для перевірки нових технологій.

РИЗИКИ ПОВ'ЯЗАНІ З КВАЛІФІКАЦІЄЮ ПЕРСОНАЛУ

Хоча цій групі ризиків зазвичай приділяють мало уваги, але вона не менш важлива, ніж дві попередні. Наскільки співробітники, які беруть участь у проекті досвідчені у роботі з вибраними технологіями? Чи є досвід ведення аналогічних проектів, чи достатній досвід менеджера проекту?

Проект не можуть врятувати *генії-одинаки*, якщо основна маса учасників проекту - *дилетанти*. Якщо *програмісти* не розбираються в UML діаграмах, а *аналітики* тільки їх і створюють, то проект можна навіть не починати.

Керівник проекту повинен оцінити ризики та організувати навчання ще до початку проекту, щоб не втрачати час на помилки в майбутньому. *Досвідчені наставники* вже знають, як обійти більшість помилок, які зустрічаються на шляху розробники, тому їх досвід дозволить заощадити значні кошти, навіть якщо це не очевидно.

Для вирішення цієї групи ризиків
необхідна в першу чергу *підтримка*
керівництва. Потім, якщо учасники
проекту частково зайняті своїми
основними обов'язками і у них є
власний керівник, то щодо проведення
робіт по проекту *необхідно*
вирішувати питання саме з
керівником.

Менеджер проекту повинен вирішити, що і до якого терміну потрібно зробити, а керівник підрозділу має вирішити, хто це буде робити і коли.

ПОЛІТИЧНІ РИЗИКИ

Це дуже небезпечна група ризиків, яка з високою ймовірністю може привести до краху проекту навіть якщо всі інші "підводні камені" будуть обійдені.

Кожна людина має свої цілі діяльності, які не завжди збігаються з цілями керівника проекту. Якщо керівник структурного підрозділу у безпосередньому підпорядкуванні якого знаходиться учасник проекту не вважає за потрібне виділяти час свого підлеглого на конкретні роботи в проекті, то це дуже великий ризик.

ПЛАНУВАННЯ РЕАГУВАННЯ НА РИЗИКИ

Планування реагування на ризику - це процес розробки шляхів і визначення дій по збільшенню можливостей та зниження загроз для цілей проекту. Даний процес починається після проведення якісного та кількісного аналізу ризиків.

МОЖЛИВІ ЧОТИРИ МЕТОДИ РЕАГУВАННЯ НА РИЗИКИ:

- 1.** Ухилення від ризику (risk avoidance).
- 2.** Передача ризику (risk transference).
- 3.** Зниження ризиків (risk mitigation).
- 4.** Прийняття ризику (risk acceptance).

УХИЛЕННЯ ВІД РИЗИКУ

Ухилення від ризику передбачає зміну плану управління проектом таким чином, щоб виключити загрозу, викликану негативним ризиком, захистити цілі проекту від наслідків ризику або послабити мети, що перебувають під загрозою (наприклад, зменшити зміст проекту). Деякі ризики, що виникають на ранніх стадіях проекту, можна уникнути за допомогою уточнення вимог, отримання додаткової інформації або проведення експертизи.

ПЕРЕДАЧА РИЗИКУ

Передача ризику передбачає перекладення негативних наслідків загрози з відповідальністю за реагування на ризик на третю сторону. Передача ризику просто переносить відповідальність за його управління іншій стороні, але ризик при цьому нікуди не дівається. Передача ризику практично завжди передбачає виплату премії за ризик стороні, що приймає на себе ризик. Наприклад, замовлення на стороні розробки ризикованого компонента за фіксованою ціною.

ЗНИЖЕННЯ РИЗИКІВ

Зниження ризиків передбачає зниження ймовірності та / або наслідків негативного ризикованого події до прийнятних меж. Прийняття запобіжних заходів щодо зниження ймовірності настання ризику або його наслідків часто виявляються більш ефективними, ніж зусилля з усунення негативних наслідків, що вживаються після настання події ризику.

ПРИЙНЯТТЯ РИЗИКУ

Прийняття ризику означає, що команда проекту усвідомлено прийняла рішення не змінювати план управління проектом у зв'язку з ризиком або не знайшла відповідної стратегії реагування. Ми змушені приймати всі «невідомі ризики».

ГОЛОВНІ РИЗИКИ ПРОГРАМНИХ ПРОЕКТІВ ТА СПОСОБИ РЕАГУВАННЯ

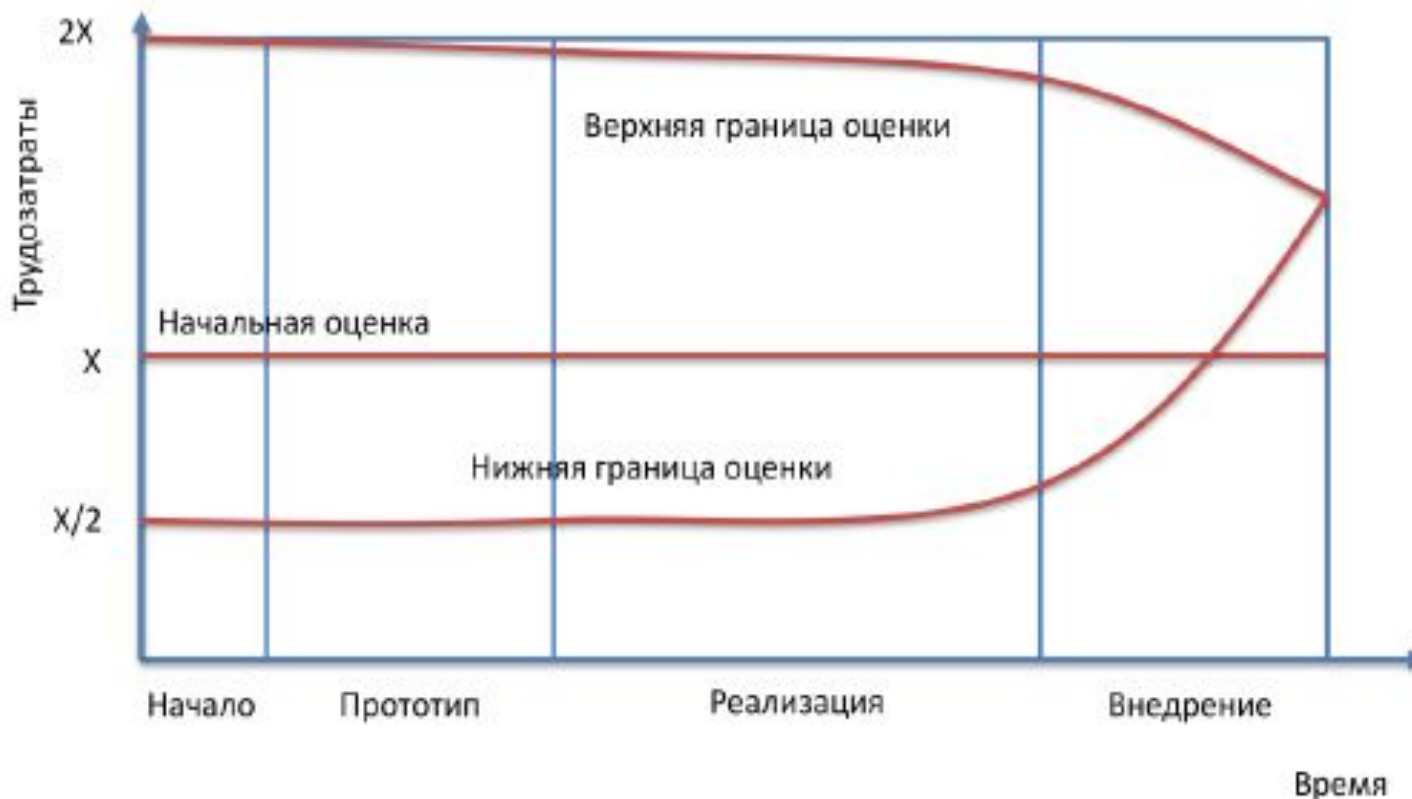
Список з п'яти головних причин провалу програмних проектів - наступний:

- Вимоги замовника відсутні / не повні / схильні до частих змін.
- Відсутність необхідних ресурсів та досвіду.
- Відсутність робочого взаємодії з замовником.
- Неповнота планування. «Забуті роботи».
- Помилки в оцінках трудоемкостей і термінів робіт

ДО ЧАСТО УПУСКАЄМОГО У ВИМОГАХ МОЖНА ВІДНЕСТИ:

- Функціональні.
- Програми установки, настройки, конфігурації.
- Міграція даних.
- Інтерфейси з зовнішніми системами.
- Довідкова система.
- Загальносистемні.
- Продуктивність.
- Надійність.
- Відкритість.
- Масштабованість.
- Безпека.
- Міжплатформеність.
- Ергономічність.

НЕВИЗНАЧЕНІСТЬ НЕ ЗМЕНШУЄТЬСЯ, ЯКЩО УПРАВЛІННЯ НЕ СПРЯМОВАНЕ НА РАННІЙ ДОЗВІЛ РИЗИКІВ



ВИЗНАЧЕННЯ ПРІОРИТЕТІВ ВИМОГ НА ПЕРШІ ІТЕРАЦІЇ ПРОЕКТУ



УПРАВЛІННЯ, НАЦІЛЕНЕ НА ЗНИЖЕННЯ РИЗИКІВ, ДОЗВОЛЯЄ ЗМЕНШУВАТИ НЕВИЗНАЧЕНІСТЬ

