

ТЕМА 1.1. Складність програмного забезпечення

Складність реальної предметної області

Промислові програмні продукти

Промислові програмні продукти застосовуються для вирішення найрізноманітніших завдань, таких, наприклад, як системи зі зворотним зв'язком, які керують або самі керуються подіями фізичного світу і для яких ресурси часу і пам'яті обмежені; завдання підтримки цілісності інформації об'ємом в сотні тисяч записів при паралельному доступі до неї з оновленнями і запитами; системи управління і контролю за реальними процесами (наприклад, диспетчеризація повітряного або залізничного транспорту). Системи подібного типу зазвичай мають великий час життя, і велика кількість користувачів потрапляють у залежність від їх нормального функціонування.

Суттєва риса промислової програми - рівень складності: один розробник практично не в змозі охопити всі аспекти такої системи.

Чому програмному забезпеченню властива складність?

Складність викликається чотирма основними причинами:

- складністю реальної предметної області, з якої виходить замовлення на розробку;
- труднощами управління процесом розробки;
- необхідністю забезпечити достатню гнучкість програми;
- незадовільними способами опису поведінки великих дискретних систем;

Проблеми, які ми намагаємося вирішити за допомогою програмного забезпечення, часто неминуче містять складні елементи, а до відповідних програм пред'являється безліч різних, часом взаємовиключних вимог.

Зовнішня складність зазвичай виникає через «нестиковки» між користувачами системи і її розробниками: користувачі ледве можуть пояснити в формі, зрозумілій розробникам, що насправді потрібно зробити. Бувають випадки, коли користувач лише смутно уявляє, що йому потрібно від майбутньої програмної системи. Це в основному відбувається не через помилки з тієї чи іншої сто-

рони; просто кожна з груп спеціалізується в своїй області, і їй бракує знань партнера. У користувачів і розробників різні погляди на сутність проблеми, і вони роблять різні висновки про можливі шляхи її вирішення.

Насправді, навіть якщо користувач точно знає, що йому потрібно, ми насилу можемо однозначно зафіксувати всі його вимоги. Зазвичай вони відображені на багатьох сторінках тексту, «розбавлених» небагатьма малюнками. Такі документи важко піддаються розумінню, вони відкриті для різних інтерпретацій і часто містять елементи, що стосуються радше до дизайну, ніж до необхідних вимог розробки.

Додаткові труднощі виникають у результаті змін вимог до програмної системи вже в процесі розробки. В основному вимоги коригуються через те, що саме здійснення програмного проекту часто змінює проблему. Розгляд перших результатів - схем, прототипів, - і використання системи після того, як вона розроблена і встановлена, змушують користувачів краще зрозуміти і чіткіше сформулювати те, що їм дійсно потрібно. У той же час цей процес підвищує кваліфікацію розробників в предметній області і дозволяє їм ставити більш осмислені питання, які проясняють темні місця в проєктованій системі.

Складність управління колективом розробників.

Основне завдання розробників полягає в створенні ілюзії простоти, в захисті користувачів від складності описуваного предмета або процесу. Розмір вихідних текстів програмної системи аж ніяк не входить в число її головних достоїнств, тому ми намагаємося робити вихідні тексти більш компактними, винаходячи хитромудрі і потужні методи, а також використовуючи середовища розробки вже існуючих проєктів і програм. Однак нові вимоги для кожної нової системи неминучі, а вони призводять до необхідності або створювати багато програм «з нуля», або намагатися по-новому використовувати існуючі.

Сьогодні звичайними стали програмні системи, розмір яких обчислюється десятками тисяч або навіть мільйонами рядків на мовах високого рівня. Жодна людина ніколи не зможе повністю зрозуміти таку систему. Навіть якщо ми правильно розкладемо її на складові частини, ми все одно отримаємо сотні, а іноді

і тисячі окремих модулів. Тому такий обсяг робіт зажадає залучення команди розробників. Завжди будуть виникати значні труднощі, пов'язані з організацією колективної розробки. Чим більше розробників, тим складніше зв'язки між ними і тим складніше координація, особливо якщо учасники робіт географічно віддалені один від одного, що типово в разі дуже великих проектів. Таким чином, при колективному виконанні проекту головним завданням керівництва є підтримка єдності і цілісності розробки.

Складність опису поведінки великих дискретних систем.

Всередині великої прикладної програми можуть існувати сотні і навіть тисячі змінних і кілька потоків управління. Повний набір цих змінних, їх поточних значень, поточної адреси і стека виклику для кожного процесу описує стан прикладної програми в кожен момент часу. Так як виконання нашої програми здійснюється на цифровому комп'ютері, ми маємо систему з дискретними станами. Аналогові системи, такі, як рух кинутого м'яча, навпаки, є безперервними. Якщо система описується безперервною функцією, в ній немає прихованих сюрпризів. Невеликі зміни вхідних параметрів завжди викличуть невеликі зміни вихідних.

З іншого боку, дискретні системи за самою своєю природою мають кінцеве число можливих станів, хоча у великих системах це число відповідно до правил комбінаторики дуже велике. Ми намагаємося проектувати системи, розділяючи їх на частини так, щоб одна частина мінімально впливала на іншу. Однак переходи між дискретними станами не можуть моделюватися безперервними функціями. Кожна подія, зовнішня по відношенню до програмної системи, може перевести її в новий стан, і, більш того, перехід з одного стану в інший не завжди детермінований. При несприятливих умовах зовнішня подія може порушити поточний стан системи через те, що її творці не змогли передбачити всі можливі варіанти.

У дискретних системах будь-яка зовнішня подія може вплинути на будь-яку частину внутрішнього стану системи. Це, очевидно, і є головною причиною обов'язкового тестування наших систем; але справа в тому, що за винятком са-

мих тривіальних випадків, всеосяжне тестування таких програм провести неможливо. І поки у нас немає ні математичних інструментів, ні інтелектуальних можливостей для повного моделювання поведінки великих дискретних систем, ми повинні задовольнитися розумним рівнем впевненості в їх правильності.

Проблеми, що виникають при спілкуванні з замовниками програмних систем.

Під час перших зустрічей щодо замовлення на розроблення ПЗ замовник дуже приблизно уявляє, що йому потрібно. Як правило, він надає кілька сторінок тексту завдання і одразу просить оцінити час виконання замовлення та його вартість. Без чіткого визначення процесів, для автоматизації яких буде використовуватись ПЗ, неможливо навіть приблизно оцінити обсяг робіт. Приблизна оцінка з мінімумом вхідної інформації може призвести до помилки в кілька разів, що негативно відіб'ється на точності визначення строків виконання та вартості робіт.

Потреби (needs) – відображають проблеми бізнесу, персоналій або процесу, що повинні співвідноситися з використанням або придбанням системи.

Щоб тримати уявлення про можливі обсяги робіт, потрібно пропонувати замовнику надати або розробити технічне завдання. Завдяки цьому системні аналітики зможуть розібратися в задачі, за допомогою інструментальних засобів виконати декомпозицію системи на компоненти, приблизно визначити обсяги цих компонентів і відповідно час їх реалізації. Потрібно описати поведінку розроблюваної системи. Ця система отримує якісь сигнали з її оточення, тому треба описати поведінку оточення, але оточення само залежить і змінюється під впливом системи, її сигналів, особливо аварійних.

Вирішують це протиріччя ітераційно, поетапно уточнюючи поведінку як системи, так і її оточення. Для відповідальних систем замовник може запропонувати розробити імітаційну модель системи та оточення, що є досить складною. У поставленому завданні замовник визначає вимоги до створюваної си-

стеми, які повинні задовольняти потреби користувачів і бути зрозумілими для розробників. Досвід показує, що оцінка складності системи, що є сумою оцінок її компонентів, отриманих у результаті декомпозиції, значно точніша, ніж первісна оцінка системи в цілому. Використання засобів формалізації результатів аналізу для їх документального оформлення також підвищує якість початкового опису вимог до системи.

Складність оцінки якості програмного забезпечення

Поняття «якість програмного забезпечення».

Кожний програмний продукт повинен виконувати визначені функції, тобто робити те, що задумано розробниками. Якісний програмний продукт повинен мати набір властивостей, які дозволять успішно його використовувати упродовж тривалого періоду, тобто мати визначену якість.

Якість - це сукупність властивостей і характеристик продукції або послуги, які надають їм здатність задовольняти встановлені або передбачувані потреби.

З визначення якості випливає, що, по-перше, якість поняття динамічне: оскільки з часом потреби можуть змінюватися, необхідно періодично переглядати і вимоги до якості. По-друге, оскільки якість - сукупність властивостей, необхідні показники для їх опису та кількісні характеристики для їх оцінки.

Сукупність властивостей ПЗ, що забезпечує задовільну для користувача якість ПЗ, залежить від умов і характеру експлуатації цього ПЗ. Тому при опису якості ПЗ, насамперед, повинні бути визначені критерії оцінки якості ПЗ. В даний час критеріями якості ПЗ прийнято вважати:

Функціональність – здатність ПЗ виконувати набір функцій, які задовольняють потреби користувачів. Набір зазначених функцій визначається в зовнішньому описі ПЗ.

Надійність – це здатність ПЗ безвідмовно виконувати визначені функції при заданих умовах протягом заданого періоду часу з досить великою імовірністю.

Легкість застосування – це характеристика ПЗ, яка дозволяє мінімізувати зусилля користувача по підготовці вхідних даних, застосуванню ПЗ і оцінці от-

риманих результатів. Ефективність – це відношення рівня послуг, які надає ПЗ користувачу при заданих

умовах, до обсягу використовуваних ресурсів.

Супровід – це характеристики ПЗ, що дозволяють мінімізувати зусилля по внесенню змін для усунення в ньому помилок і по його модифікації відповідно до потреби користувача.

Мобільність – здатність ПЗ бути перенесеним з одного середовища (оточення) в інше, зокрема, з одного комп'ютера на інший.

Функціональність і надійність є обов'язковими критеріями якості ПЗ, причому забезпечення надійності червоною ниткою проходить по всіх етапам і процесам розробки ПЗ. Інші критерії використовуються в залежності від потреб користувача у відповідності з вимогами, що пред'являються до ПЗ.

Стандарт ISO9126.

Якість програмного забезпечення визначається в стандарті ISO 9126 як вся сукупність його характеристик, що належать до можливості задовольняти висловлені або ті, що мають на увазі потреби всіх зацікавлених осіб. Той же стандарт ISO 9126 дає наступне представлення якості.

Розрізняються поняття: 1.внутрішньої якості, пов'язаного з характеристиками ПЗ самого по собі, без урахування його поведінки; 2. зовнішньої якості, що характеризує ПЗ із точки зору його поведінки; і 3. якості ПЗ при використанні у різних контекстах - тої якості, що відчувається користувачами при конкретних сценаріях роботи ПЗ. Для всіх цих аспектів якості уведено метрики, що дозволяють оцінити їх. Крім того, для створення добротного ПЗ істотною якістю технологічних процесів його розробки. На рис. 1.показані взаємини між цими аспектами якості за схемою, прийнятою ISO 9126, показане

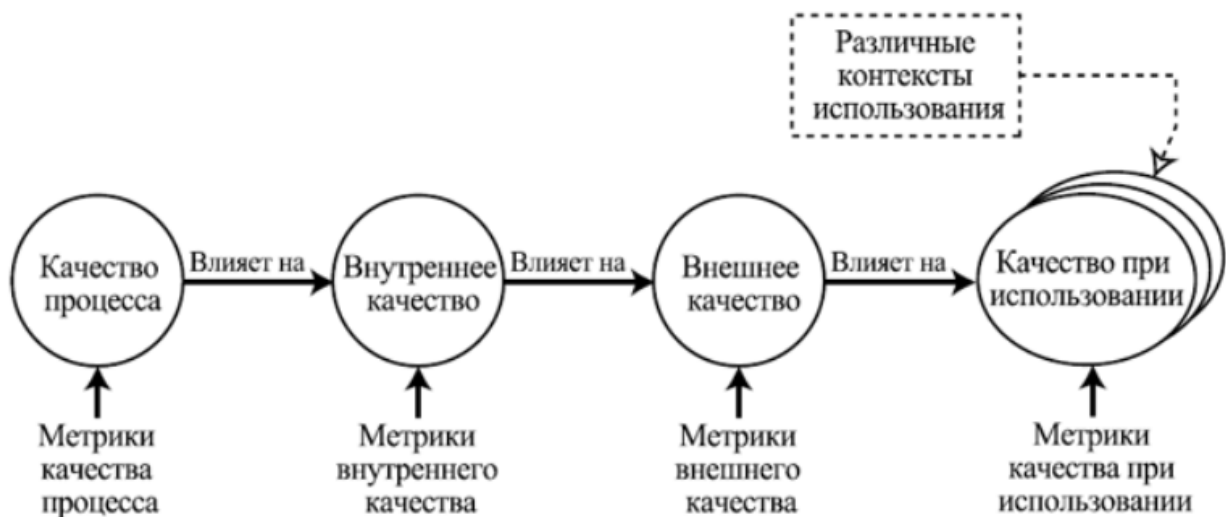


Рис. 1. Основні аспекти якості ПЗ за ISO 9126.



Рис. 2. Характеристики й атрибути якості ПЗ по ISO 9126

Рис. 1. Основні аспекти якості ПЗ за ISO 9126.

Стандарт ISO 9126 пропонує використовувати для опису внутрішнього та зовнішнього якості ПЗ багаторівневу модель. На верхньому рівні виділено 6 основних характеристик якості ПО. Кожна характеристика описується за допомогою кількох вхідних у неї атрибутів. Для кожного атрибута визначається

набір метрик, що дозволяють його оцінити. Множина характеристик і атрибутів якості згідно ISO 9126 показана на рис.2.

Нижче наведені визначення цих характеристик і атрибутів за стандартом ISO 9126:

ФУНКЦІОНАЛЬНІСТЬ (FUNCTIONALITY)

Здатність ПЗ в певних умовах вирішувати задачі, потрібні користувачам. Визначає, що саме робить ПЗ, які задачі воно вирішує

Здатність до взаємодії (interoperability). Здатність взаємодіяти з потрібним набором інших систем

о Функціональна придатність (suitability). Здатність вирішувати потрібний набір задач

о Відповідність стандартам і правилам (compliance). Відповідність ПЗ наявних індустріальних стандартах, нормативним і законодавчим актам, іншим регулюючим нормам

о Захищеність (security). Здатність запобігати неавторизованому, тобто без вказівки особи, що намагається його здійснити, і недозволеному доступу до даних і програм

о Точність (accuracy). Здатність видавати потрібні результати

НАДІЙНІСТЬ (RELIABILITY).

Здатність ПЗ підтримувати визначену працездатність у заданих умовах

Зрілість, завершеність (maturity). Величина, зворотна частоті відмов ПЗ. Звичайно вимірюється середнім часом роботи без збоїв і величиною, зворотною імовірності виникнення відмови за даний період часу

Здатність до відновлення (recoverability). Здатність відновлювати визначений рівень працездатності й цілісність даних після відмови, необхідні для цього час і ресурси

Відповідність стандартам надійності (reliability compliance). Цей атрибут доданий в 2001 році

Стійкість до відмов (fault tolerance). Здатність підтримувати заданий рівень працездатності при відмовах і порушеннях правил взаємодії з оточенням

ЗРУЧНІСТЬ ВИКОРИСТАННЯ (USABILITY) АБО ПРАКТИЧНІСТЬ

Здатність ПЗ бути зручним у навчанні та використанні, а також привабливим для користувачів

- о Зрозумілість (understandability). Показник, зворотний до зусиль, які затрачаються користувачами на сприйняття основних понять ПЗ та усвідомлення їх застосовності для розв'язання своїх задач

- о Зручність навчання (learnability). Показник, зворотний зусиллям, затраченим користувачами на навчання роботі з ПЗ.

- Зручність роботи (operability). Показник, зворотний зусиллям, що вживається користувачами для розв'язання своїх задач за допомогою ПЗ.

- Привабливість (attractiveness). Здатність ПЗ бути привабливим для користувачів. Цей атрибут доданий в 2001 році.

- о Відповідність стандартам зручності використання (usability compliance). Цей атрибут доданий в 2001 році.

ПРОДУКТИВНІСТЬ (EFFICIENCY) АБО ЕФЕКТИВНІСТЬ

Здатність ПЗ при заданих умовах забезпечувати необхідну працездатність стосовно виділюваного для цього ресурсам. Можна визначити її і як відношення одержуваних за допомогою ПЗ результатів до затрачуваних на це ресурсів усіх типів

- о Часова ефективність (time behaviour). Здатність ПЗ видавати очікувані результати, а також забезпечувати передачу необхідного об'єму даних за відведений час

- о Ефективність використання ресурсів (resource utilisation). Здатність вирішувати потрібні задачі з використанням визначених об'ємів ресурсів визначених видів. Маються на увазі такі ресурси, як оперативна й довгострокова пам'ять, мережні з'єднання, пристрої вводу та виводу та ін.

- о Відповідність стандартам продуктивності (efficiency compliance). Цей атрибут доданий в 2001 році

ЗРУЧНІСТЬ СУПРОВОДУ (MAINTAINABILITY).

Зручність проведення всіх видів діяльності, пов'язаних із супроводом програм

- о Аналізованість (analyzability) або зручність проведення аналізу. Зручність проведення аналізу помилок, дефектів і недоліків, а також зручність аналізу необхідності змін і їх можливих наслідків

- о Зручність внесення змін (changeability). Показник, зворотний трудозатратам на виконання необхідних змін

- о Стабільність (stability). Показник, зворотний ризику виникнення несподіваних ефектів при внесенні необхідних змін

- о Зручність перевірки (testability). Показник, зворотний трудозатратам на проведення тестування і інших видів перевірки того, що внесені зміни привели до потрібних результатів

- о Відповідність стандартам зручності супроводу (maintainability compliance). Цей атрибут доданий в 2001 році

ПЕРЕНОСИМІСТЬ (PORTABILITY).

Здатність ПЗ зберігати працездатність при перенесенні з одного оточення в інше, включаючи організаційні, апаратні й програмні аспекти оточення.

Іноді ця характеристика називається у нашій літературі мобільністю. Однак термін "мобільність" варто зарезервувати для перекладу "mobility" - здатності ПЗ й комп'ютерної системи в цілому зберігати працездатність при її фізичному переміщенні в просторі.

- о Адаптованість (adaptability). Здатність ПЗ пристосовуватися різним оточенням без проведення для цього дій, крім заздалегідь передбачених

- о Зручність установки (installability). Здатність ПЗ бути встановленим або розгорнутим у визначеному оточенні

- о Здатність до співіснування (coexistence). Здатність ПЗ співіснувати з іншими програмами у загальному оточенні, ділячи з ними ресурси

- о Зручність заміни (replaceability) іншого ПЗ даним. Можливість застосування даного ПЗ замість інших програмних систем для вирішення тих же задач у певному оточенні

о Відповідність стандартам переносимості (portability compliance). Цей атрибут доданий в 2001 році

Перераховані атрибути належать до внутрішньої та зовнішньої якості ПЗ згідно ISO 9126.

Для опису якості ПЗ при використанні стандарту ISO 9126-4 пропонує інший, більш вузький набір характеристик

Ефективність (effectiveness). Здатність ПЗ надавати користувачам можливість вирішувати їх задачі з необхідною точністю при використанні в заданому контексті

- Продуктивність (productivity). Здатність ПЗ надавати користувачам визначені результати в рамках очікуваних витрат ресурсів

- Безпека (safety). Здатність ПЗ забезпечувати необхідно низький рівень ризику завдання втрат життя й здоров'ю людей, бізнесу, власності або навколишньому середовищу

- Задоволення користувачів (satisfaction). Здатність ПЗ приносити задоволення користувачам при використанні в заданому контексті

Крім перерахованих характеристик і атрибутів якості, стандарт ISO 9126:2001 визначає набори метрик для оцінки кожного атрибута. Наведемо наступні приклади таких метрик

- Повнота реалізації функцій - відсоток реалізованих функцій по відношенню до перерахованого у вимогах. Використовується для виміру функціональної придатності

- Коректність реалізації функцій - правильність їх реалізації по відношенню до вимог. Використовується для виміру функціональної придатності

- Відношення числа виявлених дефектів до прогнозованого. Використовується для визначення зрілості

- Відношення числа проведених тестів до загального їх числа. Використовується для визначення зрілості

- Відношення числа доступних проектних документів до зазначеного в їх списку. Використовується для виміру зручності проведення аналізу

- Наочність і повнота документації Використовується для оцінки зрозумілості

Перераховані характеристики та атрибути якості ПЗ дозволяють систематично описувати вимоги до нього, визначаючи, які властивості ПЗ за даною характеристикою хочуть бачити зацікавлені сторони. Таким чином, вимоги повинні визначати наступне.

- ПЗ має , наприклад:

- о дозволяти клієнтові оформити замовлення й забезпечити їхню доставку;

- о забезпечувати контроль якості будівництва й відслідковувати проблемні місця;

- о підтримувати потрібні характеристики автоматизованого процесу

- виробництва, запобігаючи аварії й оптимальним способом використовуючи наявні ресурси

- Бути надійним, наприклад:

- о працювати 7 днів у тиждень і 24 години на добу;

- о допускається непрацездатність протягом не більше 3 годин у рік; о ніякі введені користувачами дані при відмові не повинні губитися

- Бути зручним для користування, наприклад:

- о користувач повинен, знаючи назву товару й маючи середні навички роботи в Інтернет, знаходити потрібний йому товар за не більш ніж 2 хв.;

- інженер не професіонал з комп'ютерних технологій повинен протягом одного дня вміти розібратися в 80% функцій системи

- Бути ефективним, наприклад:

- о підтримувати обслуговування до 10000 запитів у секунду;

- о час відгуку на запит при максимальному завантаженні не повинне перевищувати 3 с;

- о час реакції на зміну параметрів процесу виробництва не повинне

перевищувати 0.1 с;

о на обробку одного запиту не повинне витрачатися більше 1 МВ оперативної

пам'яті

- Зручним для супроводу, наприклад:

о додавання в систему нового виду запитів не повинне вимагати більше 3 людино-днів;

о додавання підтримки нового етапу процесу виробництва не повинне коштувати більше \$20000.

- Бути пристосованим, наприклад:

о ПЗ повинне працювати на операційних системах Linux, Windows XP і MacOSX;

о ПЗ повинне працювати з документами у форматах MS Word і HTML;

о ПЗ повинне зберігати файли звітів у форматах MS Word 20**, MS Excel 20**, HTML, RTF та у вигляді звичайного тексту;

о ПЗ повинне сполучатися з існуючою системою запису даних про замовлення.

Наведені атрибути якості закріплені в стандартах, але це не означає, що вони цілком вичерпують поняття якості ПЗ. Так, у стандарті ISO 9126 відсутні характеристики, пов'язані з мобільністю ПЗ (mobility), тобто здатністю програми працювати при фізичних переміщеннях машини, на якій вона працює. Замість надійності багато дослідників воліють розглядати більш загальне поняття добротності (dependability), що описує здатність ПЗ підтримувати визначені показники якості за основними характеристиками (функціональності, продуктивності, зручності використання) із заданими ймовірностями виходу за їх рамки та визначеним максимальним збитком від можливих порушень. Крім того, активно досліджуються поняття зручності використання, безпеки й захищеності ПЗ, - вони здаються більшості фахівців набагато більш складними, ніж це описується даним стандартом.