

УДК 004.891.3: 004.3

Т. О. ГОВОРУЩЕНКО, О. В. ПОМОРОВА

Хмельницький національний університет, Україна

МЕТОД ОЦІНКИ ДОСТАТНОСТІ ІНФОРМАЦІЇ ДЛЯ ВИЗНАЧЕННЯ СКЛАДНОСТІ ТА ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ПОРІВНЯЛЬНОГО АНАЛІЗУ ОНТОЛОГІЙ

Наявність взаємної кореляції метрик ПЗ є фактором, що негативно впливає на достовірність визначення характеристик ПЗ та погіршує їх точність. Тому варто проводити аналіз системних специфікацій ПЗ, що розробляється вперше, на предмет достатності інформації для визначення характеристик ПЗ. Пропонований метод оцінки достатності інформації для визначення складності та якості програмного забезпечення базується на порівняльному аналізі фрагментів онтології предметної галузі «Інженерія ПЗ» та онтологій, побудованих на основі системної специфікації розроблюваного ПЗ. Метод забезпечує можливість покращення специфікації вимог до ПЗ на предмет наявності показників, необхідних для визначення характеристик програмного проекту, що проілюстровано на прикладі визначення показників якості та складності ПЗ автоматизованої системи (АС) широкоформатного фотодруку.

Ключові слова: програмне забезпечення (ПЗ), програмний проект, специфікація вимог до ПЗ, складність програмного проекту, якість програмного проекту, складність ПЗ, якість ПЗ, онтологія.

Вступ

Практично усі сфери людської діяльності на сьогодні пов'язані з комп'ютерними та прикладними інформаційними системами, основою яких є програмне забезпечення (ПЗ). Однією із основних вимог користувачів до сучасного ПЗ є якість. Під якістю розуміють таку характеристику ПЗ, що відображає ступінь його відповідності вимогам [1-3]. При цьому вимоги можуть трактуватись досить широко, що породжує цілий ряд незалежних визначень поняття якості. Згідно визначень ISO [1, 4], якість - це ступінь відповідності наявних характеристик вимогам. Згідно [2], якість - це повнота властивостей і характеристик продукту, процесу або послуги, які забезпечують здатність задовольняти оголошеним або передбачуваним потребам. Згідно [3], якість ПЗ - це ступінь, в якій воно володіє потрібною комбінацією властивостей.

Ще однією важливою характеристикою ПЗ є його складність. Існує декілька різних видів складності. Програмна складність характеризується довжиною програми або обсягом пам'яті ЕОМ, необхідної для розміщення ПЗ. Структурна складність програм визначається кількістю взаємодіючих компонент, кількістю зв'язків між компонентами та складністю їх взаємодії. Складність міжмодульних зв'язків в процесі проектування можна характеризувати ймовірністю помилки при їх формалізації та ступенем впливу цієї

помилки на наступне функціонування модулів. Складність програмних модулів характеризується конструктивною складністю створення оформленої компоненти програми і оцінюється з позиції складності внутрішньої структури та перетворення змінних в кожному модулі, а також інтегрально за деякими зовнішніми статичними характеристиками модулів. Для систем реального часу складність ПЗ переважно визначається допустимим часом відгуку, а для інформаційних систем - кількістю типів оброблюваних змінних [5].

На сьогодні існує ряд моделей [6-8], що дозволяють розраховувати якість та складність ПЗ, однак багатозначність трактування цих характеристик ускладнює такі розрахунки. Більшість моделей базуються на використанні різних метрик ПЗ [9, 10].

При цьому актуальною є задача оцінювання достатності інформації щодо ПЗ (наприклад, можливості отримання достовірної інформації щодо показників для обчислення значень метрик), на основі якої здійснюється визначення характеристик якості та складності. Неповнота, неточність та спотворення такої інформації призводять, відповідно, до падіння достовірності оцінок якості та складності.

Постановка задачі. Метою роботи є розроблення методу оцінювання достатності інформації для визначення характеристик програмного забезпечення на прикладі якості та складності ПЗ.

1. Аналіз метрик ПЗ як джерел інформації щодо його характеристик

Найбільш використовуваними джерелами інформації щодо складності ПЗ є метрики [11, 12]: Чепіна, Мак-Клура, Кафура, Холстеда, Маккейба, Джилба, формули для розрахунку яких розглядались у [12].

Щодо якості ПЗ, то найчастіше для її оцінки використовуються метрики зв'язності, зчеплення, очікуваної вартості розроблення, прогнозованої вартості перевірки якості, прогнозованої вартості розроблення, очікуваного загального часу розроблення ПЗ, прогнозованого часу етапу проектування, формули для розрахунку яких розглядались у [12].

Однак усі ці метрики і моделі оцінки якості та складності на їх основі мають ряд недоліків, що не дозволяють досягти бажаної достовірності визначення якості та складності ПЗ [11, 12].

При розробленні ПЗ вже на початкових етапах життєвого циклу, зокрема на етапі проектування, необхідно визначитися з моделями, метриками та показниками, на основі яких у подальшому будуть здійснюватися оцінки якості та складності.

Розглянемо показник «кількість модулів», що використовується у таких метриках оцінки складності як: метрика Чепіна, Джилба (абсолютна), Мак-Клура і Кафура.

Якщо значення цього показника буде неточним, то, відповідно, знизиться ступінь достовірності усіх метрик, що його використовують. Якщо ж значення цього показника буде відсутнім у специфікації вимог до ПЗ, то зникне можливість розрахунку усіх чотирьох метрик складності програмного проекту.

Метрики Чепіна та Кафура мають спільний показник «кількість модулів», тому є взаємно корелюючими.

Метрика Чепіна обчислюється за формулою:

$$I_s = Q_m \cdot (P + 2M + 3C + 0,5T), \quad (1)$$

де I_s - інформаційна міцність, Q_m - кількість модулів, P - середня кількість змінних для розрахунків і виведення в модулі; M - середня кількість модифікованих або створених в модулі змінних; C - середня кількість керуючих змінних модуля; T - середня кількість не використовуваних в модулі ("паразитних") змінних.

Метрика Кафура розраховується як:

$$I = Q_m \cdot (W \cdot R + W \cdot W_r R_d + W_r R_d \cdot R + W_r R_d \cdot (W_r R_d - 1)), \quad (2)$$

де I - інформаційна складність, Q_m - кількість модулів, W - середня кількість процедур модуля, які оновлюють структуру даних, R - середня кількість процедур модуля, які читають інформацію зі структури даних, $W_r R_d$ - середня кількість процедур модуля, які читають і оновлюють структуру даних.

Кількість модулів програми, виходячи з наведених формул метрик Чепіна та Кафура, можна визначити як на основі метрики Чепіна (формула 3), так і на основі метрики Кафура (формула 4):

$$Q_m = \frac{I_s}{(P + 2M + 3C + 0,5T)}, \quad (3)$$

$$Q_m = \frac{I}{W \cdot R + W \cdot W_r R_d + W_r R_d \cdot R + W_r R_d \cdot (W_r R_d - 1)}. \quad (4)$$

Тоді

$$\frac{I_s}{(P + 2M + 3C + 0,5T)} = \frac{I}{(W \cdot R + W \cdot W_r R_d + W_r R_d \cdot R + W_r R_d \cdot (W_r R_d - 1))}. \quad (5)$$

Отже, метрику Кафура можливо обчислити з використанням метрики Чепіна (і навпаки), що свідчить про їх взаємну кореляцію:

$$I = \frac{I_s \cdot (W \cdot R + W \cdot W_r R_d + W_r R_d \cdot R + W_r R_d \cdot (W_r R_d - 1))}{(P + 2M + 3C + 0,5T)}. \quad (6)$$

Якщо деякі з показників, що входять до складу цих метрик, визначені неточно або відсутні, то використання цих двох метрик одночасно у моделях чи комплексних показниках оцінки складності суттєво вплине на достовірність отриманих оцінок.

За такої ситуації важливою є умова пом'якшення впливу взаємної кореляції таких метрик при використанні їх у моделях чи комплексних показниках оцінки складності та якості. Таке пом'якшення здійснюється шляхом виявлення спільних показників, підвищення точності їх значень, або, за можливості, обмеження одночасного задіявання наборів метрик, що містять однакові показники.

У цьому контексті значну цінність мають знання фахівців, що вже володіють досвідом оцінки якості та складності для різних типів ПЗ. Наприклад, цінними є знання щодо взаємовпливу та кореляції показників, котрі використовуються при обчисленні метрик та характеризують властивості ПЗ, оскільки така кореляція може погіршувати точність отриманих оцінок, частина показників,

важливих для деякого типу ПЗ може упускатись взагалі, достовірність результуючих оцінок може значно знижуватись і т.і.

Інформацію щодо визначення показників якості та складності ПЗ, наприклад, взаємозв'язків метрик за показниками, зручно подавати у вигляді семантичних мереж чи інших структур, що дозволяють відобразити причинно-наслідкові зв'язки між поняттями. Однією з таких структур є онтологія.

2. Метод оцінки достатності інформації для визначення складності та якості програмного забезпечення на основі порівняльного аналізу онтологій

Вперше поняття онтології в галузі інформаційних технологій застосував Том Грубер [13]. Онтологія - це специфікація концептуалізації, де концептуалізація - це опис понять, а також вся інформація, що має відношення до понять і необхідна для опису та вирішення задач предметної галузі. Онтології використовуються для відображення відомих знань, а також надбання, структурування знань і формування нових знань предметної галузі.

Формально онтологія визначається як:

$$O = \langle X, R, F \rangle, \quad (7)$$

де X - скінченна множина понять (концептів) предметної галузі, R - скінченна множина відношень між поняттями, F - скінченна множина функцій інтерпретації, заданих на концептах чи відношеннях.

Для представлення взаємозв'язків онтологій різних рівнів використовуються моделі онтологічних систем:

$$Z = \langle O, P, M \rangle, \quad (8)$$

де O - онтологія верхнього рівня, що містить загальні поняття і відношення, які не залежать від предметної галузі; P - множина предметних онтологій предметної галузі (відповідно до задач, що вирішуються); M - модель машини виведення даної онтологічної системи.

У найпростішому випадку процес побудови онтології складається з двох етапів:

1) виділення концептів - базових понять предметної галузі;

2) побудови зв'язків між концептами - визначення співвідношень і взаємодій базових понять.

Перевагами використання онтологій є системний підхід до вивчення предметної галузі,

можливість цілісного подання відомої інформації предметної галузі, виявлення дублювань та прогалин у знаннях на основі візуалізації відсутніх логічних зв'язків та ін.

Для побудови та візуалізації онтологій на сьогодні розроблено ряд програмних продуктів, зокрема і універсальних, що дають змогу працювати з різними предметними галузями: Ontolingua Server, SMART, Protégé, OntoEdit, WebOnto, ODE (Ontological Design Environment), DOE (Differential Ontology Editor), CONE, OntoEditor+. Автори використовують вільно поширюване програмне забезпечення Protégé 4.2, що розроблене групою медичної інформатики Стенфордського університету (перша версія – у 1987 році) з метою допомоги розробникам ПЗ у створенні та підтримці явних моделей предметної галузі та включення цих моделей безпосередньо у програмний код, але також дає можливість працювати (створювати, редагувати, переглядати та порівнювати) з онтологіями різних предметних галузей [14].

Для оцінки достатності інформації при визначенні складності та якості ПЗ пропонується метод на основі порівняльного аналізу онтологій.

Насамперед, для предметної галузі «Інженерія ПЗ» будується базова онтологія, у котрій є інформація щодо метрик, що застосовуються для визначення складності та якості ПЗ.

Для побудови такої онтології оберемо 4 характеристики ПЗ: складність програмного проекту, складність ПЗ, якість програмного проекту, якість ПЗ. Зазначені характеристики визначаються на основі метрик та моделей, обраних у [10 - 12] – для визначення складності програмного проекту використовуються 4 метрики, для визначення складності ПЗ – 6 метрик, для визначення якості проекту – 5 метрик, для визначення якості ПЗ – 9 метрик. Формули для визначення метрик та моделей, наведені у [10 - 12], дають можливість обрати показники специфікації та ПЗ для визначення відповідних метрик та моделей – для визначення метрик складності проекту необхідно 13 показників, для визначення метрик складності ПЗ – 10 показників, для визначення метрик якості проекту – 14 показників, для визначення метрик якості ПЗ – 13 показників.

Складовими частинами базової онтології є: базова онтологія для визначення складності програмного проекту (рис. 1), базова онтологія для визначення складності ПЗ (рис. 3), базова онтологія для визначення якості проекту (рис. 5), базова онтологія для визначення якості ПЗ (рис. 7).

Побудована базова онтологія дала можливість зробити наступні висновки:

1) всі чотири метрики для визначення

складності програмного проекту використовують показник «кількість модулів»;

2) чотири з шести метрик (метрики Холстеда, Маккейба, Джилба та кількість операторів програми) для визначення складності ПЗ використовують показник «загальна кількість операторів», три (LOC-оцінка, метрика Холстеда та кількість операторів програми) з шести зазначених метрик використовують показник «кількість рядків коду», дві (метрика Холстеда та прогнозована оцінка складності інтерфейсів) з шести метрик використовують показник «загальна кількість операндів» та «кількість унікальних операндів»;

3) метрики для визначення якості проекту не мають спільних показників;

4) три з дев'яти метрик для визначення якості ПЗ (вартість проектування, вартість перевірки якості, вартість розроблення) використовують показник «ціна одного рядка коду», дві (трудовитрати та тривалість проекту за моделлю Боема) з дев'яти зазначених метрик використовують показник «тип проекту», вісім (крім функційного розміру) з дев'яти метрик використовують показник «кількість рядків коду», три (загальний час розроблення ПЗ, час етапу проектування, продуктивність розроблення ПЗ) з дев'яти метрик використовують показник «тривалість проекту»;

5) характеристики «складність проекту», «складність ПЗ» та «якість проекту» використовують показник «кількість модулів»; характеристики «складність проекту» та «якість проекту» використовують показник «кількості модулів, безпосередньо передуючих і безпосередньо слідує за модулем»; характеристики «якість проекту», «складність ПЗ» та «якість ПЗ» використовують показник «кількість рядків коду»; характеристики «якість проекту» та «якість ПЗ» використовують показник «прогнозована тривалість проекту», «частка (доля) етапу проектування у життєвому циклі».

Отже, при оцінюванні складності та якості ПЗ необхідно приділити особливу увагу тим вищевказаним показникам, що входять до складу декількох метрик одночасно.

Метод оцінки достатності інформації для визначення складності та якості програмного забезпечення складається з наступних кроків:

1) аналіз специфікації вимог до ПЗ на предмет наявності показників, необхідних для визначення складності та якості програмного проекту, прогнозованої складності та якості ПЗ;

2) розроблення онтології для визначення складності та якості конкретного програмного забезпечення, складовими частинами якої є: онтологія для визначення складності конкретного

програмного проекту, онтологія для визначення складності конкретного ПЗ, онтологія для визначення якості конкретного проекту, онтологія для визначення якості конкретного ПЗ;

3) порівняння розробленої онтології з базовою онтологією для визначення складності та якості ПЗ, складові частини якої наведені на рис. 1, 3, 5, 7;

4) виявлення показників, які відсутні в онтології для визначення складності та якості конкретного програмного забезпечення;

5) виявлення метрик, які неможливо обчислити на основі наявних показників;

6) виявлення характеристик, значення яких неможливо визначити на основі метрик, які залишається можливим обчислити при наявній інформації;

7) наявність характеристик, значення яких неможливо визначити на основі специфікації вимог до ПЗ, свідчить про вимогу доповнення цієї специфікації необхідними для обчислення цієї чи іншої характеристики показниками (на цьому етапі можливе як додавання необхідної інформації, так і видалення іншої необхідної інформації);

8) повторення етапів 2-7 доти, доки не стане можливим визначити всі необхідні характеристики або доки не буде сформовано висновок, що недостатньо даних для визначення складності та якості ПЗ.

3. Оцінка специфікації вимог до ПЗ автоматизованої системи широкоформатного фотодруку

При проведенні дослідження було проаналізовано специфікацію вимог до ПЗ автоматизованої системи широкоформатного фотодруку. На основі аналізу специфікації визначено показники, необхідні для визначення складності та якості програмного проекту, прогнозованої складності та якості ПЗ. Ці показники дали змогу розробити онтологію для визначення складності та якості даного програмного забезпечення, складовими частинами якої є: онтологія для визначення складності даного програмного проекту (рис. 2), онтологія для визначення складності даного ПЗ (рис. 4), онтологія для визначення якості даного проекту (рис. 6), онтологія для визначення якості даного ПЗ (рис. 8).

Порівняння розробленої онтології для програмного проекту з розроблення автоматизованої системи широкоформатного фотодруку з фрагментами базової онтології для предметної галузі «Інженерія ПЗ» дало можливість виявити, що у розробленій онтології відсутні 9 показників: кількість керуючих змінних, ціна одного рядка коду,

тривалість програмного проекту, тип програмного проекту (за Боемом), кількість рядків коду, кількість зв'язків кожного модуля, кількість модулів, частка (доля) етапу проектування у життєвому циклі, загальна кількість операторів – показники специфікації вимог до ПЗ.

Крім цього, на основі порівняння розробленої онтології для програмного проекту з розроблення автоматизованої системи широкоформатного фотодруку з базовою онтологією було виявлено, що у розробленій онтології недостатньо даних для визначення деяких метрик та моделей через відсутність вищевказаних показників.

Аналіз рис. 1 та рис. 2 дав можливість зробити висновок, що для обчислення метрик Чепіна, Мак-Клура та Кафура недостатньо даних, а для обчислення метрики Джилба (абсолютна складність) взагалі відсутні необхідні показники, отже, жодну з метрик складності проекту не можна обчислити, тому відповідно не можна визначити складність програмного проекту.

Аналіз рис. 3 та рис. 4 показав, що для обчислення метрик Холстеда, Маккейба, Джилба (відносна складність) та прогнозованої оцінки складності інтерфейсів ПЗ наявних показників недостатньо, а для обчислення LOC-оцінки та прогнозованої кількості операторів програми необхідні показники взагалі відсутні, отже, жодну з метрик прогнозування складності ПЗ не можна обчислити, тому не можна визначити й прогнозовану складність розроблюваного ПЗ.

Аналіз рис. 5 та рис. 6 дав можливість зробити висновок, що для обчислення часу модифікації моделей необхідні показники відсутні, для обчислення кількості знайдених помилок при інспектуванні моделей наявної інформації недостатньо, а метрики зв'язності, зчеплення та звертання до глобальних змінних можуть бути обчислені. Отже, якість програмного проекту можна визначити лише на основі метрик зв'язності, зчеплення та звертання до глобальних змінних (три метрики з п'яти – недостатність інформації).

Аналіз рис. 7 та рис. 8 показав, що для обчислення загального часу розроблення ПЗ, часу виконання робіт процесу проектування, очікуваної вартості розроблення ПЗ, продуктивності розроблення ПЗ, оцінок трудовитрат та тривалості проекту за моделлю Боема необхідні показники відсутні, а для обчислення вартості перевірки якості та витрат на реалізацію програмного коду наявної інформації недостатньо. Обчислена може бути лише одна метрика – прогнозований функційний розмір, тому прогнозовану якість ПЗ можна визначити лише на основі однієї метрики з дев'яти (недостатність інформації).

Тоді відсутність у специфікації вимог до ПЗ дев'яти показників призвела до неможливості обчислення 20 метрик (з 24) та відповідно до неможливості визначення складності проекту та розроблюваного ПЗ і до неповноти визначення якості проекту та розроблюваного ПЗ.

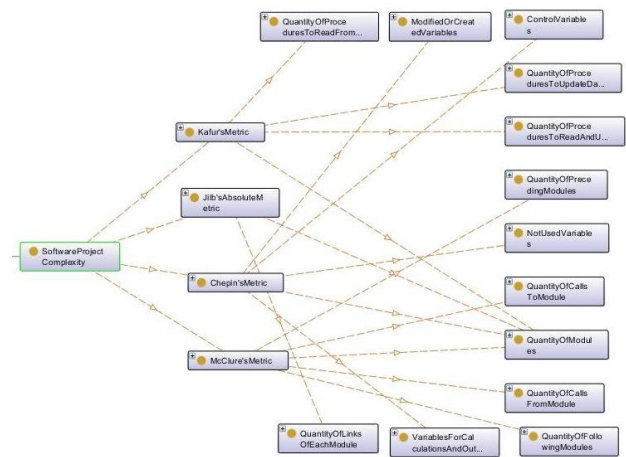


Рис. 1. Базова онтологія для визначення складності програмного проекту

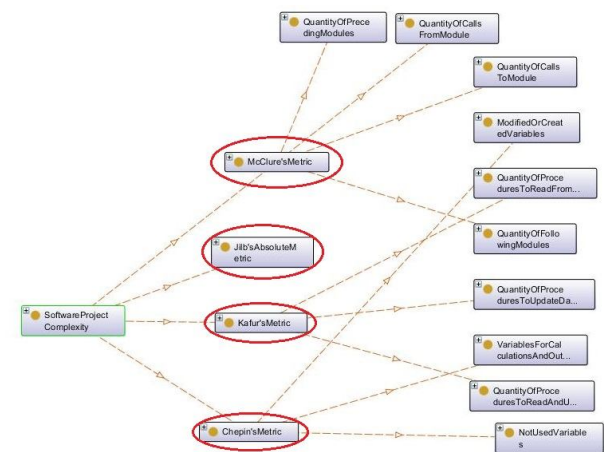


Рис. 2. Онтологія для визначення складності програмного проекту з розроблення автоматизованої системи (АС) широкоформатного фотодруку

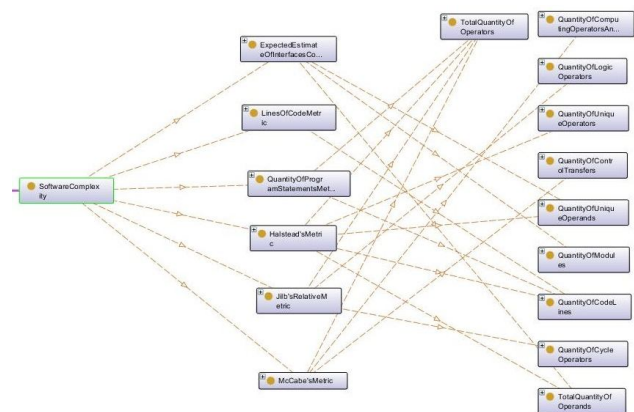


Рис. 3. Базова онтологія для прогнозування складності ПЗ

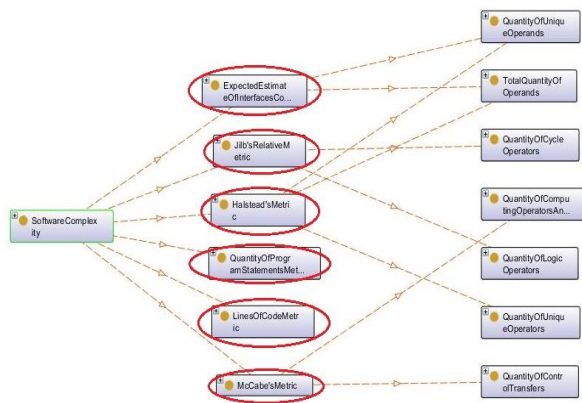


Рис. 4. Онтологія для прогнозування складності ПЗ АС широкоформатного фотодруку

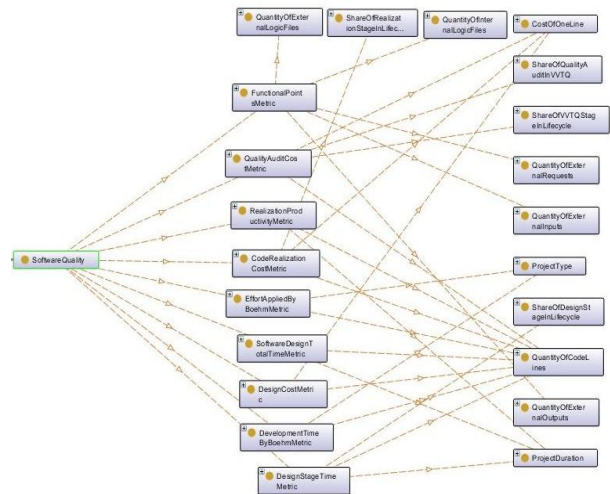


Рис. 7. Базова онтологія для прогнозування якості програмного забезпечення

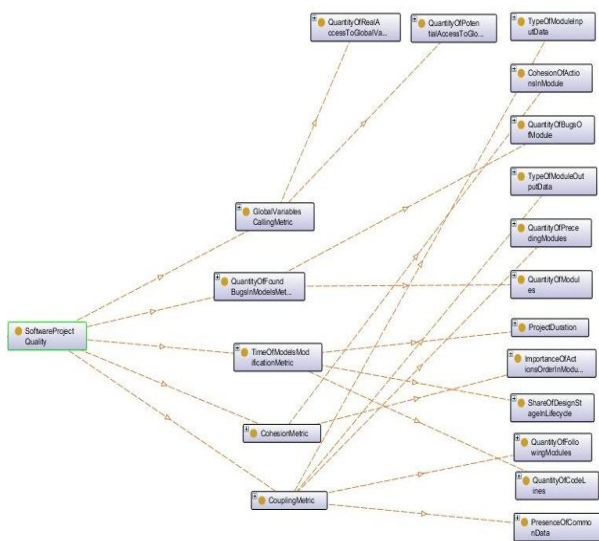


Рис. 5. Базова онтологія для визначення якості програмного проекту

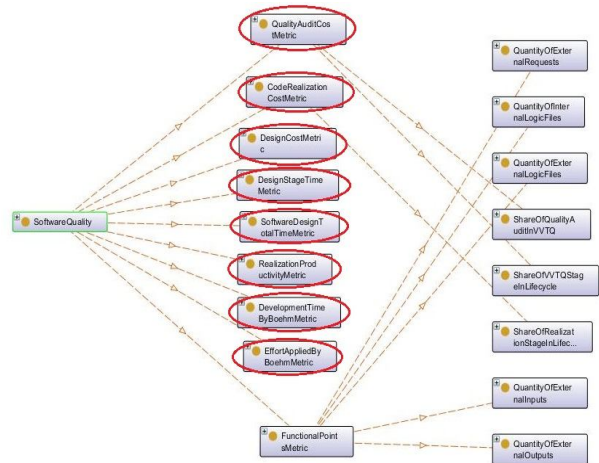


Рис. 8. Онтологія для прогнозування якості ПЗ АС широкоформатного фотодруку

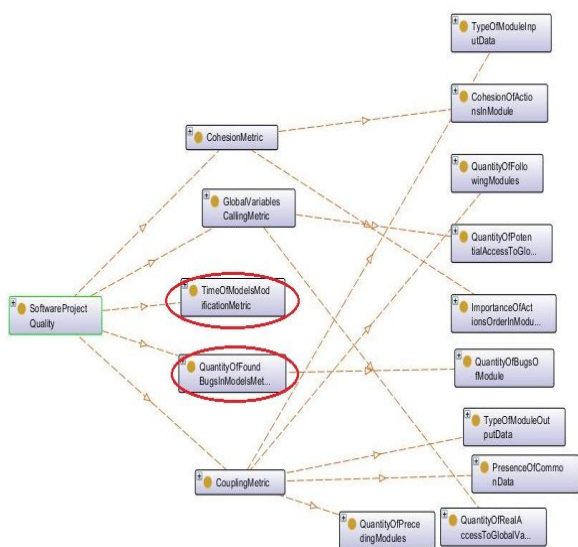


Рис. 6. Онтологія для визначення якості програмного проекту з розроблення АС широкоформатного фотодруку

Оскільки для даного програмного проекту є характеристики, які при наявній в специфікації вимог інформації взагалі неможливо визначити або можна визначити в недостатній мірі, то відбулось доповнення специфікації вимог до ПЗ, після чого було створено нову версію онтології для визначення складності та якості конкретного програмного забезпечення.

Порівняльний аналіз нової версії онтології із базовою онтологією показав, що зміни відбулись при визначенні складності конкретного програмного проекту (рис. 9), при прогнозуванні складності конкретного ПЗ (рис. 10) та при визначенні якості проекту (рис. 11).

Порівняння нової версії онтології для визначення складності та якості конкретного програмного забезпечення та базової онтології показало, що у розробленій онтології відсутні 14 показників, тобто при доповненні специфікації вимог до ПЗ було додано лише 2 показники:

кількість модулів та загальна кількість операторів. Але після додавання цих показників з'явилась можливість обчислити: метрики Мак-Клура і Кафура, та визначити складність програмного проекту на основі двох метрик (з чотирьох) – рис. 9; метрики Маккейба, Джилба (відносна складність), прогнозовану оцінку складності інтерфейсів, що дає можливість визначити прогнозовану складність ПЗ на основі трьох метрик (з шести) – рис. 10; кількість знайдених помилок при інспектуванні моделей, що дає можливість визначити якість проекту на основі вже чотирьох з п'яти метрик – рис. 11. Отже, доповнення специфікації вимог до ПЗ двома показниками збільшило кількість інформації для визначення якості проекту та дало можливість визначити складність проекту та розроблюваного ПЗ (все ще недостатність інформації).

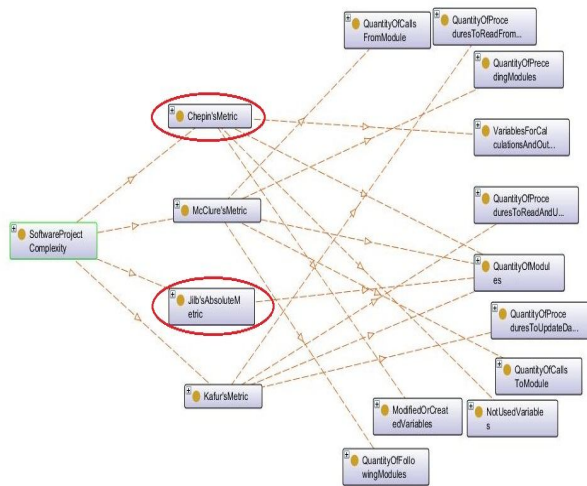


Рис. 9. Онтологія (версія 2) для визначення складності програмного проекту з розроблення АС широкоформатного фотодруку

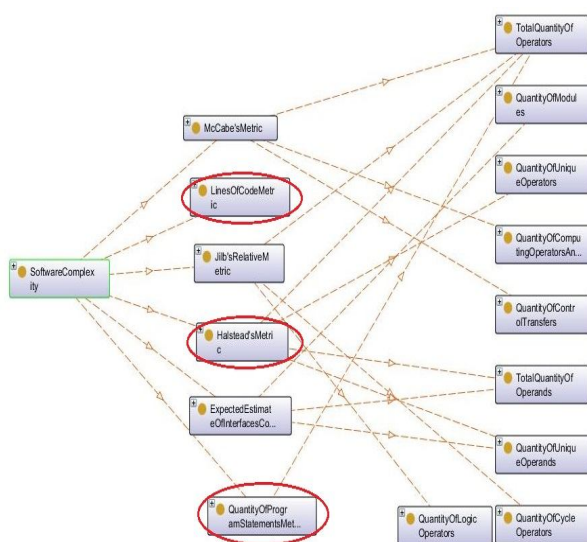


Рис. 10. Онтологія (версія 2) для прогнозування складності ПЗ АС широкоформатного фотодруку

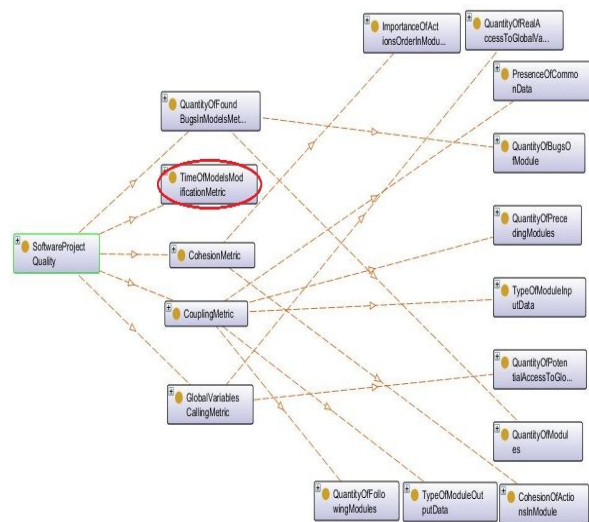


Рис. 11. Онтологія (версія 2) для визначення якості програмного проекту з розроблення автоматизованої системи широкоформатного фотодруку

На основі методу оцінки достатності інформації для визначення складності та якості ПЗ і аналізу базових онтологій можна зробити наступні рекомендації щодо формування специфікації вимог до ПЗ (з точки зору можливості оцінювання зазначених характеристик):

1) за потреби визначити складність проекту за чотирма метриками, специфікація вимог до ПЗ повинна містити наступні 13 показників:

- кількість змінних для розрахунків і виведення,
- кількість модифікованих або створених в модулі змінних,
- кількість керуючих змінних,
- кількість невикористовуваних в модулі змінних,
- кількість модулів,
- кількість зв'язків кожного модуля,
- кількості модулів, безпосередньо передуючих і безпосередньо слідуючих за модулем,
- кількість звертань до модуля,
- кількість викликів з модуля,
- кількість процедур, які оновлюють структуру даних,
- кількість процедур, які читають інформацію зі структури даних,
- кількість процедур, які читають і оновлюють структуру даних;

2) за потреби прогнозувати складність ПЗ за шістьма метриками, специфікація вимог до ПЗ повинна містити наступні 10 показників:

- прогнозована кількість рядків коду,
- прогнозована кількість унікальних операторів,
- прогнозована кількість унікальних операндів,

- прогнозована загальна кількість операторів,
- прогнозована загальна кількість операндів,
- прогнозована кількість обчислювальних операторів та виразів,
- прогнозована кількість передач керування,
- прогнозована кількість логічних операторів,
- прогнозована кількість циклічних операторів,
- кількість модулів;

3) за потреби визначити якість проекту за п'ятьма метриками, специфікація вимог до ПЗ повинна містити наступні 14 показників:

- зв'язність дій всередині модуля,
- важливість порядку дій всередині модуля,
- типи вхідних та вихідних параметрів модуля,
- наявність спільних даних,
- кількості модулів, безпосередньо передуючих і безпосередньо слідуючих за модулем,
- кількість реальних викликів глобальної змінної,
- кількість можливих викликів глобальної змінної,
- прогнозована кількість рядків коду,
- прогнозована тривалість проекту,
- частка (доля) етапу проектування у життєвому циклі,
- кількість помилок в модулі,
- кількість модулів;

4) за потреби прогнозувати якість ПЗ за дев'ятьма метриками, специфікація вимог до ПЗ повинна містити наступні 13 показників:

- прогнозована кількість рядків коду,
- прогнозована тривалість проекту,
- частка (доля) етапу проектування у життєвому циклі,
- ціна одного рядка коду,
- частка (доля) етапу тестування, верифікації, валідації та аудиту якості у життєвому циклі,
- частка (доля) аудиту якості у етапі тестування, верифікації, валідації та аудиту якості,
- частка (доля) етапу реалізації у життєвому циклі,
- кількість зовнішніх входів,
- кількість зовнішніх виходів,
- кількість зовнішніх запитів,
- кількість внутрішніх логічних файлів,
- кількість зовнішніх логічних файлів,
- тип проекту (за Боемом).

Висновки

В процесі аналізу метрик ПЗ, як джерел інформації щодо його характеристик, було виявлено наявність взаємної кореляції таких метрик, оскільки вони мають ряд спільних показників. Якщо деякі з показників, що входять до складу цих метрик,

визначені неточно або відсутні, то використання цих двох метрик одночасно у моделях чи комплексних показниках оцінки складності суттєво вплине на достовірність отриманих оцінок, тобто кореляція метрик може погіршувати точність та достовірність визначення характеристик ПЗ.

За такої ситуації важливою є умова пом'якшення впливу взаємної кореляції таких метрик при використанні їх у моделях чи комплексних показниках оцінки складності та якості. Таке пом'якшення здійснюється шляхом виявлення спільних показників, підвищення точності їх значень, або, за можливості, обмеження одночасного задіювання наборів метрик, що містять однакові показники.

Цінними є знання досвідчених фахівців щодо взаємовпливу та кореляції метрик, котрі характеризують властивості ПЗ. Такі знання варто зберігати та використовувати при оцінюванні системних специфікацій ПЗ, що розробляється вперше, на предмет достатності інформації для визначення характеристик ПЗ.

У якості засобу відображення цих знань обрано онтології, які стали основою методу оцінки достатності інформації для визначення складності та якості програмного забезпечення.

Запропонований метод оцінки достатності інформації для визначення складності та якості програмного забезпечення на основі порівняльного аналізу онтологій дав можливість розробити рекомендації щодо покращення специфікації вимог до ПЗ, що проілюстровано на прикладі визначення показників якості та складності ПЗ АС широкоформатного фотодруку.

Література

1. ISO 9001:2015. *Quality management systems – Requirements [Text]. – Revises ISO 9001:2008 : introduced 2015-09-15 – ISO/IEC, 2015. – 29 p.*
2. Bourque, P. *Guide to the software engineering body of knowledge (SWEBOK). Version 3.0 [Text] / P. Bourque, R. E. Fairley – A project of the IEEE Computer Society, 2014. – 335 p.*
3. ISO 9000:2015 *Quality management systems – Fundamentals and vocabulary [Text] – Revises ISO 9000:2005: introduced 2015-09-15 – ISO/IEC, 2015. – 51 p.*
4. ISO/IEC 25010:2011 *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuARE) - System and software quality models [Text] – Revises ISO/IEC 9126-1:2001: introduced 2011-03-01 – ISO/IEC, 2011. – 34 p.*
5. Ковалевская, Е. В. *Метрология, качество и сертификация ПО [Текст] : материалы к курсу /*

Е. В. Ковалевская ; Московский государственный университет экономики, статистики и информатики. – М. : МГУЭСИ, 2002. - 38 с.

6. Miguel, J. P. *A review of software quality models for the evaluation of software products [Text]* / J. P. Miguel, D. Mauricio, G. Rodriguez // *International Journal of Software Engineering & Applications (IJSEA)*. – 2014. – Vol. 5, No. 6. – P. 31-54.

7. Klas, M. *Adapting software quality models: practical challenges, approach, and first empirical results [Text]* / M. Klas, C. Lampasona, J. Munch // *37th EUROMICRO Conference on Software Engineering and Advanced Applications, Oulu (Finland), 30 Aug – 02 Sep 2011 – IEEE, 2011.* – P. 341-348.

8. Ayala, C. *Selection of third party software in Off-The-Shelf-based software development - an interview study with industrial practitioners [Text]* / C. Ayala, Ø. Hauge, R. Conradi, X. Franch, J. Li // *The Journal of Systems and Software*. – 2010. – P. 24-36.

9. Hovorushchenko, T. *The software emergent properties and them reflection in the non-functional requirements and quality models [Text]* / T. Hovorushchenko // *Комп'ютерні науки та інформаційні технології : матеріали X Міжнародної науково-технічної конференції, Львів, 14-17 вересня 2015 р. - Львів, 2015.* – С. 146-153.

10. Говорущенко, Т. О. *Дослідження відомих моделей оцінювання характеристик програмного забезпечення [Текст]* / Т. О. Говорущенко // *Вісник Хмельницького національного університету.* – 2013. – № 1. – С. 117-121.

11. Поморова, О. В. *Аналіз методів та засобів оцінки якості програмних систем [Текст]* / О. В. Поморова, Т. О. Говорущенко // *Радіоелектронні і комп'ютерні системи.* – 2009. – № 6. – С. 148-158.

12. Поморова, О. В. *Інтелектуальний метод оцінювання результатів проектування та прогнозування характеристик якості програмного забезпечення [Текст]* / О. В. Поморова, Т. О. Говорущенко // *Радіоелектронні і комп'ютерні системи* – 2010. – № 6. – С. 211-218.

13. Gruber, T. *A translation approach to portable ontologies [Text]* / T. Gruber // *Knowledge Acquisition.* – 1993. – № 5(2). – P. 199-220.

14. *A free, open-source ontology editor and framework for building intelligent systems [Electronic resource].* – Access mode: <http://protege.stanford.edu/>. – 11.03.2016.

2. Bourque, P., Fairley, R. E. *Guide to the software engineering body of knowledge (SWEBOK).* Version 3.0. *IEEE Computer Society*, 2014. 335 p.

3. *ISO 9000:2015 Quality management systems – Fundamentals and vocabulary.* Revises ISO 9000:2005: introduced 2015-09-15. ISO/IEC, 2015. 51 p.

4. *ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.* Revises ISO/IEC 9126-1:2001: introduced 2011-03-01. ISO/IEC, 2011. 34 p.

5. Kovalevskaya, Ye. V. *Metrologiya, kachestvo I sertifikaciya PO: materialy k kursu [Metrology, quality and certification of software: Tutorials].* Moscow State University of Economic, Statistic and Infromatic. Moscow, MSUESI Publ., 2002. 38 p.

6. Miguel, J. P., Mauricio, D., Rodriguez, G. *A review of software quality models for the evaluation of software products.* *International Journal of Software Engineering & Applications*, 2014, vol. 5, no. 6, pp.31-54.

7. Klas, M., Lampasona, C., Munch, J. *Adapting software quality models: practical challenges, approach, and first empirical results.* *37th EUROMICRO Conference on Software Engineering and Advanced Applications*, Oulu (Finland), 30 Aug – 02 Sep 2011, pp. 341-348.

8. Ayala, C., Hauge, Øyvind, Conradi, R., Franch, X., Li Jingyue. *Selection of third party software in Off-The-Shelf-based software development - an interview study with industrial practitioners.* *The Journal of Systems and Software*, 2010, pp. 24-36.

9. Hovorushchenko, T. *The software emergent properties and them reflection in the non-functional requirements and quality models.* *Computer Science and Information Technologies: Proceedings of X International Conference*, Lviv, 14-17 Sep 2015, pp.146-153.

10. Hovorushchenko, T. O. *Doslidjennya vidomyh modelei otsinuvannya harakteristik programnogo zabezpechennya [Research of known models of evaluation of software characteristics].* *Transactions of Khmelnytsky National University*, 2013, no. 1, pp. 117-121.

11. Pomorova, O. V., Hovorushchenko, T. O. *Analiz metodiv ta zasobiv otsinky yakosti programnyh system [Analysis of methods and tools of evaluation of software systems quality].* *Radioelectronic and computer systems*, 2009, no. 6, pp. 148-158.

12. Pomorova, O. V., Hovorushchenko, T. O. *Intelektualniy metod otsinuvannya rezultativ proektuvannya ta prognuzuvannya harakteristik yakosti programnogo zabezpechennya [Intelligent method of evaluation of design results and prediction of software quality characteristics].* *Radioelectronic and computer systems*, 2010, no. 6, pp. 211-218.

References

1. *ISO 9001:2015. Quality management systems – Requirements.* Revises ISO 9001:2008 : introduced 2015-09-15. ISO/IEC, 2015, 29 p.

13. Gruber, T. A translation approach to portable ontologies. *Knowledge Acquisition*, 1993, no. 5(2), pp. 199-220.

14. *A free, open-source ontology editor and framework for building intelligent systems*. Available at: <http://protege.stanford.edu/> (accessed 11.03.2016)

Надійшла до редакції 11.03.2016, розглянута на редколегії 14.04.2016

МЕТОД ОЦЕНКИ ДОСТАТОЧНОСТИ ИНФОРМАЦИИ ДЛЯ ОПРЕДЕЛЕНИЯ СЛОЖНОСТИ И КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ СРАВНИТЕЛЬНОГО АНАЛИЗА ОНТОЛОГИЙ

Т. А. Говорущенко, О. В. Поморова

Наличие взаимной корреляции метрик программного обеспечения (ПО) является фактором, который отрицательно влияет на достоверность определения характеристик ПО и ухудшает их точность. Поэтому стоит проводить анализ системных спецификаций ПО, которое разрабатывается впервые, на предмет достаточности информации для определения характеристик ПО. Предлагаемый метод оценки достаточности информации для определения сложности и качества ПО базируется на сравнительном анализе фрагментов онтологии предметной области «Инженерия ПО» и онтологий, построенных на основе системной спецификации разрабатываемого ПО. Метод обеспечивает возможность улучшения спецификации требований к ПО на предмет наличия показателей, необходимых для определения характеристик программного проекта, что проиллюстрировано на примере определения показателей сложности и качества ПО автоматизированной системы широкоформатной фотопечати.

Ключевые слова: программное обеспечение (ПО), программный проект, спецификация требований к ПО, сложность программного проекта, качество программного проекта, сложность ПО, качество ПО, онтология.

METHOD OF ASSESSMENT OF INFORMATION SUFFICIENCY FOR DETERMINATION OF SOFTWARE COMPLEXITY AND QUALITY BASED ON THE COMPARATIVE ANALYSIS OF ONTOLOGIES

T. A. Hovorushchenko, O. V. Pomorova

The presence of the cross-correlation of the software metrics is a factor that negatively affects to the accuracy of the determination of the software characteristics and degrades their accuracy. Therefore it is necessary to analyze the system specifications of software, that is developed for the first time, on the subject of information sufficiency for determination of the software characteristics. The proposed method of assessment of information sufficiency for determination of software complexity and quality is based on the comparative analysis of the fragments of domain ontology "Software Engineering" and ontologies, that are based on system specifications of developed software. The method provides the possible of the improvement of the software requirements specification on the subject of the presence of indicators for determining the software project characteristics, as illustrated by the example of the determination of indicators of the software complexity and quality of the automated systems of large-format fotoprinting.

Key words: Software, software project, software requirement specification, software project complexity, software project quality, software complexity, software quality, ontology.

Говорущенко Тетяна Олександрівна – канд. техн. наук, ст. наук. співроб., доцент кафедри системного програмування, Хмельницький національний університет, Хмельницький, Україна, e-mail: tat_yana@ukr.net.

Поморова Оксана Вікторівна – д-р техн. наук, проф., завідувач кафедри системного програмування, Хмельницький національний університет, Хмельницький, Україна, e-mail: o.pomorova@gmail.com.

Hovorushchenko Tetiana Oleksandrivna – PhD (Candidate of Technical Science), Senior Researcher, Associate Professor of System Programming Department, Khmelnytsky National University, Khmelnytsky, Ukraine; e-mail: tat_yana@ukr.net.

Pomorova Oksana Viktorivna – Full Doctor (Doctor of Technical Science), Professor, Head of System Programming Department, Khmelnytsky National University, Khmelnytsky, Ukraine, e-mail: o.pomorova@gmail.com.