

СУЧАСНІ ПРОБЛЕМИ ОЦІНЮВАННЯ ПОЧАТКОВИХ ЕТАПІВ ЖИТТЄВОГО ЦИКЛУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Т. О. Говорущенко¹, О. О. Павлова²

¹Хмельницький національний університет

²Хмельницька гімназія №2

Анотація. Проведено аналіз інформаційних технологій для галузі інженерії програмного забезпечення (ПЗ), дослідження впливу початкових етапів життєвого циклу на успішність ПЗ, аналіз методів та засобів і дослідження онтологій для оцінювання початкових етапів життєвого циклу ПЗ, які показали необхідність розроблення агентно-орієнтованої інформаційної технології оцінювання початкових етапів життєвого циклу ПЗ на основі онтологічного підходу.

Ключові слова: програмне забезпечення (ПЗ), програмний проект, успішність ПЗ, початкові етапи життєвого циклу ПЗ, інформаційна технологія, онтологія, інтелектуальний агент.

Вступ

Практично усі сфери людської діяльності на сьогодні пов'язані з комп'ютерними системами, основою яких є програмне забезпечення (ПЗ). Але, при наявності ряду методів та засобів, залученні кращих фахівців для проектування і розроблення ПЗ, успішність програмних проектів і досі залежить від знань та досвіду розробників.

На сьогодні в світі витрачається більше 250 млрд. USD щорічно на розроблення приблизно 175 тис. програмних проектів. Середня вартість проекту для великої компанії становить 2,322 млн. USD, для середньої компанії – 1,313 млн. USD, а для невеликої компанії – 434 тис. USD [1, 2].

При цьому значна кількість програмних проектів є невдалими. Дослідження The Standish Group International доводить, що 31,1% програмних проектів будуть скасовані до їх завершення, а 52,7% проектів коштуватимуть 189% від їхніх початкових оцінок. Але такі перевитрати при розробленні – це лише вершина айсберга. Альтернативні витрати можуть сягати трильйонів доларів. Так, наприклад, відсутність надійного ПЗ для обробки багажу в аеропорту м. Денвер коштує місту 1,1 млн. доларів на день [1, 2].

В середньому, лише 16-29% програмних проектів виконуються в межах запланованих часу та бюджету (для великих компаній – 9-16% проектів). Крім цього, проекти, виконані найбільшими американськими компаніями, мають лише приблизно 42% від необхідних можливостей та функцій (для малих компаній 78,4% програмних проектів мають принаймні 74,2% початкових можливостей та функцій) [1, 2] – рис. 1 [3].

© Говорущенко Т. О., Павлова О.О., 2018

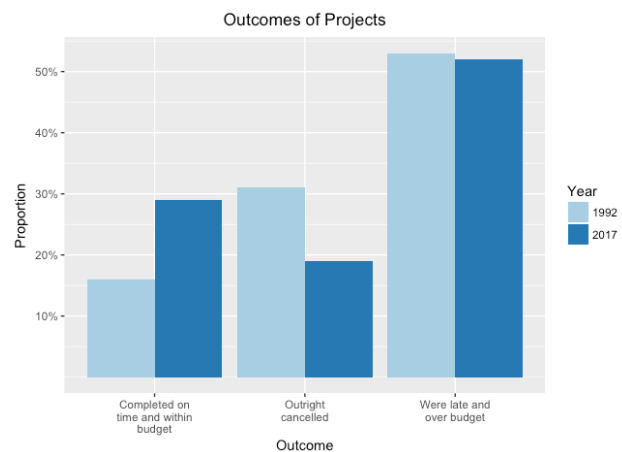


Рис. 1. Порівняльна статистика успішності програмних проектів у 1992 та 2017 роках, за даними The Standish Group International [3]

Аналіз даних рис. 1 дав можливість побачити приріст кількості успішних проектів та спад кількості провальних проектів, але в той же час кількість проблемних проектів є досить сталою величиною та складає 50% і більше проектів.

Значна кількість помилок вноситься у ПЗ на етапі формування та формулювання вимог внаслідок інформаційних втрат через неповне та різне розуміння потреб і контексту інформації. Такі помилки становлять 10-23% всіх помилок, причому чим більший обсяг ПЗ, тим більше помилок вноситься саме на цьому етапі [4, 5]. Переважна більшість аварій, пов'язаних із ПЗ, виникли через помилкові вимоги, а не через помилки кодування. Чим раніше буде виявлено дефект, тим дешевше обійдеться його виправлення. Витрати на виправлення некоректних вимог, виявлених після випуску продукту, майже в 100 разів перевищують витрати на виправлення недоліків специфікації, виявлених на більш ранніх етапах життєвого циклу ПЗ [6].

Отже, успішність реалізації програмного проекту (як вчасне виконання програмного проекту в рамках виділеного бюджету та з реалізацією всіх необхідних можливостей та функцій) суттєво залежить від ранніх етапів життєвого циклу ПЗ. Враховуючи вищевикладене, метою даного дослідження є аналіз сучасних проблем оцінювання початкових етапів життєвого циклу програмного забезпечення.

1. Аналіз сучасного стану інформаційних технологій для галузі інженерії програмного забезпечення

Основною стратегічною метою розвитку інформаційного суспільства в Україні є прискорення розроблення та впровадження новітніх конкурентоспроможних інформаційних технологій в усі сфери людської діяльності, оскільки сучасні інформаційні технології значно розширюють можливості використання інформаційних ресурсів у різних галузях за рахунок зміни характеру праці, зокрема, шляхом зменшення частки фізичної праці та мінімізації впливу людського фактору завдяки автоматизації процесів [7].

На поточному етапі розвитку та впровадження інформаційних технологій у різних галузях людської діяльності відбуваються визначальні зміни, оскільки з'явилися потужні технічні ресурси для накопичення і опрацювання значних масивів інформації. Основними проблемами, які виникають при обробці великих масивів інформації, є відсутність методів аналізу, придатних до застосування через їх різнотипність, потреба у значних людських ресурсах для підтримки процесу аналізу даних, висока обчислювальна складність наявних алгоритмів аналізу та стрімке зростання обсягу зібраних даних [8]. Застосування відомих методів та засобів до опрацювання таких масивів інформації не виправдовує очікувань розробників, призводить до перевитрат ресурсів, втрат значущої інформації та конфліктів між очікуваннями замовників та отриманими результатами [9, 10].

Загальносвітовим трендом щодо опрацювання великих масивів інформації, що дозволяє вирішувати нові класи задач на основі наявних інформаційних ресурсів, є інтелектуалізація опрацювання інформації і даних (машинне навчання [11], когнітивні обчислення [12], глибинне навчання [13], концепція семантичного вебу [14] – рис. 2 [15]), а також розроблення методів комбінування різних технологій та інструментів для підвищення ефективності використання інформаційних ресурсів.

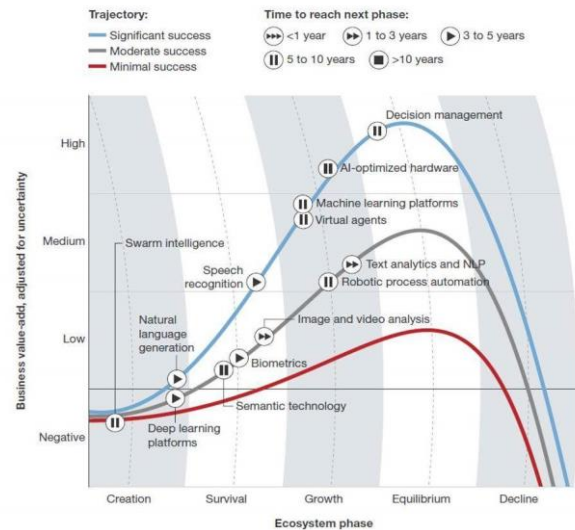


Рис. 2. Топ-10 сучасних технологій штучного інтелекту для опрацювання масивів інформації [15]

Поява потужних технічних ресурсів для накопичення та опрацювання значних масивів інформації, відсутність ефективних методів та засобів для опрацювання таких масивів інформації, розвиток інтелектуалізації опрацювання інформації є передумовами для переходу на новий якісний рівень опрацювання інформації. Значний інтерес становить визначення опорних структур даних і процесів, які можуть бути стабільними властивостями і характеристиками гнучкої інформаційної технології [16].

Однак очікування універсальних підходів до створення ефективних інформаційних технологій для будь-яких галузей діяльності на сьогодні є передчасними, оскільки на зміст та методи опрацювання інформації істотно впливають специфіка та особливості предметних галузей, для яких розробляються інформаційні технології [17]. Виправданим залишається підхід, що базується на дослідженні характеристик та особливостей предметних галузей, а також розроблення інформаційних технологій саме для конкретних галузей. Особливої уваги у напрямку розроблення та впровадження ефективних інформаційних технологій на сьогодні потребує галузь інженерії програмного забезпечення, оскільки саме ПЗ є основою усіх сучасних напрямків господарської діяльності, а помилки та відмови ПЗ загрожують катастрофами, які призводять до людських жертв, екологічних катастроф, значних часових втрат та фінансових збитків.

На сьогодні в галузі інженерії ПЗ реалізована велика кількість програмних проектів, наявна статистика успішних та неуспішних рішень та результатів, але відсутні ефективні інформаційні технології, які б дали можливість отримати всю корисну інформацію з наявної статистики.

З іншого боку, відомі інформаційні технології передбачають людинно-машинну взаємодію на всіх етапах опрацювання інформації, тобто всю інформацію інтерпретує людина, що для галузі інженерії ПЗ часто призводить до втрат істотної інформації і до виникнення помилок на ранніх етапах життєвого циклу. Відтак актуальною задачею є розроблення для галузі інженерії ПЗ інформаційних технологій нової генерації, в яких людина усувається з процесів опрацювання інформації та здобуття знань, які забезпечуватимуть новий якісний рівень опрацювання інформації за рахунок його інтелектуалізації.

Прикладом інформаційної технології нової генерації для галузі інженерії ПЗ є розроблена у [18] інформаційна технологія оцінювання достатності інформації щодо якості у специфікаціях вимог до ПЗ, спрямована на підтримку процесу оцінювання якості ПЗ на ранніх етапах життєвого циклу за рахунок оцінювання достатності об'єму наявної у специфікації вимог інформації щодо якості на основі порівняльного аналізу онтологій.

2. Дослідження впливу початкових етапів життєвого циклу на успішність програмних проєктів

Програмні проєкти часто зазнають невдач через помилки на ранніх етапах життєвого циклу ПЗ, а саме [4]: неадекватне формулювання вимог; невдале проєктування або неефективне планування; невірне розуміння або недостатній аналіз специфікації та проєкту; нереалістичні проєктні плани; некоректно обрана модель життєвого циклу.

Аналіз помилок вбудованого та прикладного ПЗ [19] показав, що некоректність та неповнота специфікації вимог призвели до: неможливості старту «Космосу-419», неможливості відстикування станції «Марс 2», аварії гелікоптера Chippook, «смертельних» сеансів радіаційної терапії із застосуванням Therac-25, вибуху ракети-носія Ariane 5, падіння рейтингу на біржових торгах компанії Dow Jones Industrial Average, збою у системі Нью-Йоркського банку, помилки Y2K (помилка двоцифрового збереження року у даті), аварій на телекомунікаційній компанії AT&T, які призвели до загибелі людей, значних фінансових збитків, втрати репутації компаній.

В роботах [5, 20, 21] підтверджується факт, що причини майже всіх інцидентів та катастроф, пов'язаних з ПЗ, криються у специфікації вимог до програмного забезпечення, тобто переважна більшість аварій, пов'язаних із ПЗ, виникли через помилкові вимоги, а не через помилки кодування.

У [21] описано результати експерименту, проведеного для підтвердження або відкидання

гіпотези про те, що збої та помилки ПЗ, написаного різними розробниками за однією специфікацією, статистично незалежні. Під час експерименту декілька незалежних груп розробників писали свою версію ПЗ за однією специфікацією. В результаті такого експерименту було встановлено, що версії ПЗ, написані різними розробниками за однаковими вимогами, містили ряд спільних помилок, пов'язаних із помилками або неточностями вимог (специфікації).

Аналіз великої кількості програмних проєктів, проведений у [22], також виявив, що головне місце виникнення помилок у ПЗ – це етап формування та формулювання вимог (специфікація).

Робота [23] основними причинами провалу програмних проєктів називає невірні уявлення керівництва та менеджерів проєкту щодо реального часу та коштів, які необхідні для забезпечення функційних вимог користувача, тобто знов-таки підтверджує, що більшість проблем ПЗ пов'язані з проблемами початкових етапів життєвого циклу.

Отже, критичний вплив на програмні проєкти та на успішність їх реалізації здійснюють питання, пов'язані із аналізом та оцінюванням початкових етапів життєвого циклу. Таке експериментальне свідчення безпосередньо призводить до необхідності поглиблення аналізу та оцінювання специфікації вимог. Підтверджене статистикою [1-3] зниження кількості проєктів із відсутньою необхідною функціональністю при зростанні кількості проєктів із перевищенням термінів та перевитратами відображає тенденцію, що софтверні організації поглиблюють аналіз та опрацювання вимог за рахунок збільшення часу та коштів. Отже, сьогодні, коли кількість високобюджетних програмних проєктів стрімко зростає, актуальним і дуже важливим є аналіз специфікації вимог до ПЗ та можливість оцінити, наскільки добре відпрацьовані початкові етапи життєвого циклу ПЗ, зокрема, наскільки достатньою є інформація у специфікації вимог до ПЗ.

3. Аналіз методів та засобів оцінювання початкових етапів життєвого циклу програмного забезпечення

Аналіз відомих методів аналізу специфікацій вимог (Using natural language processing technique, Using CASE analysis method, QAW-method, Using global analysis method, O'Brien's approach, Method to discover missing requirement elicitation, Selection of elicitation technique, Comparison and categorization of requirements elicitation techniques, Techniques for ranking and prioritization of software requirements) [24-27], призначених для роботи з вимогами та специфікаціями, показало, що всі вони спрямовані на

контроль за реалізацією вимог, але жоден з них не визначає прогнозованих значень характеристик за специфікацією. Отже, існуючі методи і засоби аналізу специфікацій не прийнятні для кількісного оцінювання рівня відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій (для оцінювання рівня достатності інформації у специфікації вимог).

Аналіз існуючих автоматизованих засобів аналізу специфікацій вимог та оцінювання характеристик ПЗ (IBM Rational RequisitePro, IBM Rational/Telelogic DOORS, Borland Caliber RM, Sybase PowerDesigner, Open Source Requirements Management Tool, пакет засобів компанії LDRA, Sigma Software, DEVPRO, CASE.Аналитик), проведений у [28], показав, що всі вони призначені для роботи з вимогами та специфікаціями, причому деякі з них спрямовані на контроль за реалізацією вимог, максимум для перевірки специфікації на достовірність, повноту та несуперечливість її інформації (CASE.Аналитик), але жоден з них не надає оцінок рівня відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій. Очевидно тоді, що існуючі засоби не прийнятні для кількісного оцінювання рівня відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій (для оцінювання рівня достатності інформації у специфікації вимог).

4. Онтології як перспективний засіб для оцінювання початкових етапів життєвого циклу програмного забезпечення

Для оцінювання початкових етапів життєвого циклу значну цінність мають знання фахівців, що вже володіють досвідом оцінювання рівня відпрацювання початкових етапів для різних типів ПЗ. Інформацію щодо вимог зручно подавати у вигляді онтологій, що дають змогу відобразити причинно-наслідкові зв'язки між вимогами.

Вперше поняття онтології в галузі інформаційних технологій застосував Том Грубер [29]. Онтологія – це специфікація концептуалізації, де концептуалізація – це опис понять, а також вся інформація, що має відношення до понять і необхідна для опису та розв'язання задач предметної галузі. Онтології використовуються для відображення відомих знань, а також надбання, структурування знань і формування нових знань предметної галузі.

Онтології – це формальні концепції конкретних предметних галузей. Вони дозволяють концептуалізувати домен фіксуванням сутностей та зв'язків у домені. Визначення, в яких зв'язках бере участь сутність, частково дозволяє зрозуміти її значення (вміст), оскільки це надає можливість бачити, де дана сутність входить у відношення з іншим доменом [30].

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. Онтології визначаються як ключова технологія для розвитку семантичного вебу і відіграють критично важливу роль в організації опрацювання інформації на базі Web, її спільного використання та обміну нею між додатками [30].

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+, які дають можливість працювати (створювати, редагувати, переглядати та порівнювати) з онтологіями різних предметних галузей.

Ідея використання онтологій вже використовувалась українськими та закордонними вченими. Так, Є. Буров запропонував методи та засоби побудови програмних систем на основі онтологічних моделей задач [31]. І. Шостак та Ю. Бутенко розробили онтологічні моделі та методи формування нормативного профілю при сертифікації ПЗ [32]. Л. Бабенко запропонував онтологічний підхід до специфікування властивостей програмних систем та їх компонентів [33]. Для здійснення трасування вимог до ПЗ вченими використовувались зважені онтології [34-37].

Метою роботи [38] є використання доменної онтології для аналізу ПЗ та засобів реінжинірингу. Робота [39] представляє онтологічну модель для опису та визначення предметних і операційних знань щодо забезпечення якості ПЗ. В роботах [18, 40] розроблено онтології та зважені онтології предметної галузі «Інженерія програмного забезпечення» (частини «Якість ПЗ», «Якість ПЗ. Метричний аналіз», «Специфікація вимог до ПЗ»), які використовуються для формалізації інформації щодо якості за стандартом ISO 25010, формалізації метричної інформації при оцінюванні якості ПЗ, формалізації специфікації вимог до ПЗ з точки зору інформації щодо якості, а також для оцінювання достатності інформації щодо якості у специфікаціях вимог до ПЗ (як наявності у специфікації вимог всіх необхідних атрибутів та показників, необхідних для визначення якості ПЗ) і встановлення пріоритетності доповнення специфікації вимог атрибутами та (або) показниками за умови недостатності у ній інформації щодо якості ПЗ.

Ще однією перевагою використання онтологій та зважених онтологій є можливість доступу, ро-

зуміння та аналізу інформації інтелектуальними агентами (що є дуже актуальним при сучасному переході до ери семантичного вебу (Semantic web), коли ресурси повинні бути зрозумілими не тільки для людини, але й для агентів).

Інтелектуальний агент (англ. Intelligent agent) – система, котра спостерігає за навколишнім середовищем, взаємодіє з ним, а її поведінка раціональна в тому сенсі, що агент розуміє суть власних дій, і ці дії скеровані на досягнення певної мети [41]. Таким агентом може виступати програмна система, бот, сервіс. Інтелектуальний агент використовує під час свого функціонування інформацію, отриману з навколишнього середовища, аналізує її, зіставляючи з уже відомими йому фактами і, на основі результатів аналізу, приймає рішення про подальші дії.

Розв'язанню задачі розроблення інтелектуальних агентів на основі онтологічного підходу присвячено ряд досліджень. Так, у статті [42] розроблено уніфіковані методи побудови інтелектуальних агентів планування діяльності з використанням онтологічного підходу з метою підвищення ефективності процесів функціонування таких систем. Робота [43] присвячена розробленню багатоагентних систем, в яких агенти приймають рішення на основі знань, взятих з онтологій., що дозволяє інтегрувати платформи агентів з семантичними веб-даними та онтологіями. Автори [44] досліджують застосування онтологій для агентно-орієнтованої програмної інженерії, пропонують інструмент, який використовує існуючі онтологічні конструкції для створення програмного коду, а також експериментально підтверджують переваги застосування агентів на основі онтологій для галузі інженерії ПЗ. Робота [45] присвячена усуненню невизначеності у вимогах до ПЗ та покращенню спілкування зацікавлених сторін шляхом впровадження інтелектуальних агентів на основі онтологічного підходу. Автори [46] пропонують розроблення інтелектуального агента на основі онтології для мінімізації існуючої семантичної невизначеності при розробленні специфікації вимог до ПЗ природньою (іспанською) мовою, для автоматичного отримання основних елементів специфікації та для автоматичної побудови діаграми цілей. У роботі [47] запропоновано методологічний підхід до інженерних систем, які базуються на інтелектуальних агентах на основі онтологічного підходу. Автори [48] використовують онтологічні агентно-орієнтовані моделі для формалізації первинних вимог до ПЗ з метою зниження витрат на прикладі розроблення додатків для Ambient Assisted Living для пацієнтів із хворобою Паркінсона. У статті [49] розроблено завдання

орієнтовану архітектуру на основі агентно-орієнтованої парадигми та онтологічного дизайну для систем підтримки прийняття рішень (на прикладі клінічної системи для відділення невідкладної допомоги), яка дозволяє ітеративно переносити функційні вимоги в архітектурні компоненти. Автори [50] надають основу для формального подання та перевірки вимог та забезпечення функційної правильності обмежених ресурсами контекстних систем критичного застосування у вигляді інтелектуальних агентів на основі онтологічного підходу.

Проведений аналіз онтологічних моделей, методів та засобів для оцінювання початкових етапів життєвого циклу ПЗ показав, що вони не розв'язують задачу кількісного оцінювання рівня відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій (зокрема, оцінювання достатності інформації у специфікації вимог до ПЗ). Крім цього, всі вони належать до різних методологічних підходів і не інтегруються між собою, тобто наразі відсутня агентно-орієнтована інформаційна технологія оцінювання початкових етапів життєвого циклу ПЗ на основі онтологічного підходу. Єдиним рішенням в галузі оцінювання достатності інформації у специфікаціях вимог до ПЗ є роботи [18, 40], в яких запропоновано теоретичні та прикладні засади оцінювання достатності інформації щодо якості у специфікаціях вимог до ПЗ та розроблено онтології предметної галузі «Інженерія програмного забезпечення» (частини «Якість ПЗ», «Якість ПЗ. Метричний аналіз», «Специфікація вимог до ПЗ»), які й будуть взяті за основу при розробленні інтелектуальних агентів на базі онтологічного підходу для інформаційної технології оцінювання початкових етапів життєвого циклу ПЗ.

Висновки

Аналіз сучасного стану інформаційних технологій для галузі інженерії програмного забезпечення показав необхідність розроблення для цієї галузі інформаційних технологій нової генерації, в яких людина усувається з процесів опрацювання інформації та здобуття знань, які забезпечуватимуть новий якісний рівень опрацювання інформації за рахунок його інтелектуалізації.

Наразі людство все частіше покладається на програмне забезпечення при вирішенні складних задач, стрімко зростає кількість програмних проєктів з високою вартістю. Але, як показує статистика, частка проблемних програмних проєктів складає порядку половини всіх програмних проєктів, а частка провальних програмних проєктів складає порядку 1/5 всіх програмних проєктів, тобто успішними програмними проєктами, на які

можна покладатись і на які варто витратити кошти, є лише 1/3 всіх програмних проєктів. Значна кількість помилок вноситься у ПЗ на початкових етапах життєвого циклу ПЗ. Переважна більшість аварій, пов'язаних із ПЗ, виникли через помилки у специфікації вимог. Тому актуальним і дуже важливим є вміння автоматизовано оцінити рівень відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій.

Аналіз відомих методів та автоматизованих засобів аналізу специфікацій вимог показав, що всі вони не прийнятні кількісного оцінювання рівня відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій.

Дослідження онтологій як перспективного засобу для оцінювання початкових етапів життєвого циклу ПЗ показало, що вчені приділяють значну увагу онтологіям як ключовій технології семантичного вебу та інтелектуальним агентам на основі онтологічного підходу як ключовим засобам для усунення людини з процесів опрацювання інформації. На сьогодні розроблено ряд онтологічних моделей, методів та засобів, які належать до різних методологічних підходів і не інтегруються між собою, тобто наразі відсутня агентно-орієнтована інформаційна технологія оцінювання початкових етапів життєвого циклу ПЗ на основі онтологічного підходу.

Актуальність задачі автоматизованого оцінювання рівня відпрацювання початкових етапів життєвого циклу ПЗ на основі аналізу специфікацій (зокрема, автоматизованого оцінювання достатності інформації у специфікації вимог до ПЗ), а також відсутність інформаційної технології оцінювання початкових етапів життєвого циклу ПЗ обумовлює необхідність розроблення агентно-орієнтованої інформаційної технології оцінювання початкових етапів життєвого циклу ПЗ на основі онтологічного підходу. Вирішення окресленої науково-прикладної задачі потребує комплексних досліджень за наступними напрямками: 1) моделювання спеціалізованих інтелектуальних агентів на основі онтологічного підходу; 2) розроблення методів та алгоритмів планування дій спеціалізованих інтелектуальних агентів на основі онтологічного підходу; 3) розроблення мультиагентної інтелектуальної системи та інформаційної технології оцінювання початкових етапів життєвого циклу ПЗ. На вирішення цих задач і спрямовуватимуться подальші зусилля авторів.

Список використаної літератури

1. Hastie, S. Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch / S. Hastie, S. Wójcik. [Electronic resource]. – Access mode: <http://www.infoq.com/articles/standish-chaos-2015>.
2. The Standish Group Report CHAOS. [Electronic resource]. – Access mode: <https://www.projectsart.co.uk/white-papers/chaos-report.pdf>.
3. A Look at 25 Years of Software Projects. What Can We Learn? [Electronic resource]. – Access mode: <https://speedandfunction.com/look-25-years-software-projects-can-learn/>.
4. McConnell, S. Code complete [Text] / S. McConnell. – Microsoft Press, 2013. – 896 p.
5. Levenson, N. G. Engineering a safer world: systems thinking applied to safety [Text] / N. G. Levenson. – MIT Press, 2012. – 560 pp.
6. Shamieh, C. Systems engineering for dummies [Text] / C. Shamieh. – New York: John Wiley & Sons, 2014. – 76 p.
7. Пиріг, С. О. Інформаційні технології та їх використання на підприємствах України [Текст] / С. О. Пиріг, О. А. Нужна // Економічний форум. – 2014. – № 3. – С. 190–195.
8. Шаховська, Н. Великі дані як інформаційна технологія / Н. Шаховська, В. Москалюк [Текст] // Інформація, комунікація, суспільство: 3-я Міжнародна наукова конференція, 21–24 травня 2014 р. : матеріали – Львів, 2014. – С. 188–189.
9. Sivarajah, U. Critical analysis of Big Data challenges and analytical methods [Text] / U. Sivarajah, M. M. Kamal, Z. Irani, V. Weerakkody // Journal of Business Research. – 2017. – Vol. 70. – Pp. 263–286.
10. Mauerhoefer, T. The impact of information technology on new product development performance [Text] / T. Mauerhoefer, S. Strese, M. Brettel // Journal of Product Innovation Management. – 2017. – Vol. 34. – Issue 6. – Pp. 719–738.
11. Kinch, M. W. Reviewing the current state of machine learning for artificial intelligence with regards to the use of contextual information [Text] / M. W. Kinch, W. J. C. Melis, S. Keates // The Second Medway Engineering Conference on Systems: Efficiency, Sustainability and Modelling, June 6, 2017: Proceedings. – University of Greenwich (UK), 2017.
12. Noor, A. K. Potential of cognitive computing and cognitive systems [Text] / A. K. Noor // Open Engineering. – 2015. – Vol. 5. – Issue 1. – Pp. 75–88.
13. LeCun, Y. Deep learning [Text] / Y. LeCun, Y. Bengio, G. Hinton // Nature. – 2015. – Vol. 521. – Issue 7553. – Pp. 436–444.
14. Ristoski, P. Semantic Web in data mining and knowledge discovery: A comprehensive survey [Text] / P. Ristoski, H. Paulheim // Journal of WEB Semantics. – 2016. – Vol. 36. – Pp. 1–22.
15. Press, G. Top 10 hot artificial intelligence (AI) technologies / G. Press. [Electronic resource] Access mode: <https://www.forbes.com/sites>

- /gilpress/2017/01/23/top-10-hot-artificial-intelligence-ai-technologies/#1d5da9561928.
16. Минухин, С. В. Формирование информационного обеспечения системы управления бизнес-процессами предприятия [Текст] / С. В. Минухин // Актуальні проблеми економіки. – 2006. – № 10. – С. 170–178.
 17. Golub, K. Subject access to information: An interdisciplinary approach [Text] / K. Golub. – Libraries Unlimited, 2015. – 165 pp.
 18. Hovorushchenko, T. Information Technology for Assurance of Veracity of Quality Information in the Software Requirements Specification [Text] / T. Hovorushchenko // Advances in Intelligent Systems and Computing. – 2018. – Vol. 689. – Pp. 166–185.
 19. Pomorova, O. Intelligent assessment and prediction of software characteristics at the design stage [Text] / O. Pomorova, T. Hovorushchenko // American Journal of Software Engineering and Applications. – 2013. – Vol. 2. – Issue 2. – Pp. 25–31.
 20. Ishimatsu, T. Hazard analysis of complex spacecraft using systems-theoretic process analysis [Text] / T. Ishimatsu, N. G. Levenson, J. P. Thomas, C. H. Fleming, M. Katahira, Yu. Miyamoto, R. Ujiie, H. Nakao, N. Hoshino // Journal of Spacecraft and Rockets. – 2014. – Vol. 51. – No. 2. – Pp. 509–522.
 21. Levenson, N. Software challenges in achieving space safety [Text] / N. Levenson // Journal of the British Interplanetary Society. – 2009. – Vol. 62. – Pp. 265–272.
 22. Jones, C. The economics of software quality [Text] / C. Jones, O. Bonsignour. – Boston: Pearson Education, 2012. – 588 p.
 23. Yourdon, E. Death march: The complete software developer's guide to surviving "Mission impossible" projects [Text] : 2nd edition / E. Yourdon. – Prentice Hall, 2003. – 256 p.
 24. Chen, A. Visual models for software requirements [Text] / A. Chen, J. Beatty. – Washington: MS Press, 2012. – 444 p.
 25. Fatwanto, A. Software requirements specification analysis using natural language processing technique [Text] / A. Fatwanto // The 13-th IEEE International Conference on Quality in Research, June 25-28, 2013 : Proceedings. – Yogyakarta (Indonesia), 2013. – Pp. 105–110.
 26. Rehman, T. Analysis of requirement engineering processes, tools/techniques and methodologies [Text] / T. Rehman, M. N. A. Khan, N. Riaz // International Journal of Information Technology and Computer Science. – 2013. – Vol. 5. – No. 3. – Pp. 40–48.
 27. Bass, L. A comparison of requirements specification methods from a software architecture perspective / L. Bass, J. Bergey, P. Clements, P. Merson, I. Ozkaya, R. Sangwan. [Electronic resource]. – Access mode: http://resources.sei.cmu.edu/asset_files/TechnicalReport/2006_005_001_14786.pdf.
 28. Hovorushchenko, T. Method of evaluating the success of software project implementation based on analysis of specification using neuronet information technologies [Text] / T. Hovorushchenko, A. Krasiiy // CEUR-WS. – 2015. – Vol. 1356. – Pp. 100–107.
 29. Gruber T. A translation approach to portable ontologies [Text] / T. Gruber // Knowledge Acquisition. – 1993. – No. 5(2) – Pp. 199–220.
 30. Андон, Ф. И. Semantic Web как новая модель информационного пространства Интернет [Текст] / Ф. И. Андон, И. Ю. Гришанова, В. А. Резниченко // Проблемы програмування. – 2008. – №2-3. Спеціальний випуск. – С. 417–430.
 31. Burov, E. Complex ontology management using task models [Text] / E. Burov // International Journal of Knowledge-Based and Intelligent Engineering Systems. – 2014. – Vol. 18. – No. 2. – Pp. 111–120.
 32. Shostak, I. Ontology approach to realization of information technology for normative profile forming at critical software certification [Text] / I. Shostak, I. Butenko // Збірник наукових праць Військового інституту КНУ ім. Т. Г. Шевченка. – 2012. – № 38. – С. 250–253.
 33. Бабенко, Л. Онтологический подход к спецификации свойств программных систем и их компонентов [Текст] / Л. Бабенко // Кибернетика и системный анализ. – 2009. – № 1. – С. 180–187.
 34. Li, Y. Ontology-based trace retrieval [Text] / Y. Li, J. Cleland-Huang // The 7-th International Workshop on Traceability in Emerging Forms of Software Engineering, May 19, 2013 : Proceedings. – Passau (Germany), 2013. – Pp. 30–36.
 35. Assawamekin, N. Ontology-based multiperspective requirements traceability framework [Text] / N. Assawamekin, A. Namfon, T. Sunetnanta, C. Pluempitiwiriawej // Knowledge Information Systems. – 2010. – No. 3 – Pp. 493–522.
 36. Zhang, Y. G. Ontological approach for the semantic recovery of traceability links between software artefacts [Text] / Y. G. Zhang, R. Witte, J. Rilling, V. Haarslev // IET Software. – 2008. – Vol. 2. – Issue 3. – Pp. 185–203.
 37. Leonid, K. Ontology and model alignment as a means for requirements validation [Text] / K. Leonid, R. Gacitua, M. Rouncefield, P. Sawyer // The 4-th IEEE International Conference on Semantic Computing, September 22-24, 2010 : Proceedings. – Pittsburgh (USA), 2010. – Pp. 46–51.
 38. Jin, D. Ontology-based software analysis and reengineering tool integration: The OASIS service-sharing methodology [Text] / D. Jin, J. R. Cordy // The 21-st IEEE International Conference on Software Maintenance, September 25-30, 2005 : Proceedings. – Budapest (Hungary), 2005. – Pp. 613–616.

39. Bajnaid, N. O. An ontological approach to model software quality assurance knowledge domain [Text] / N. O. Bajnaid, R. Benlamri, A. Pakstas, Sh. Salekzamanikhani // *Lecture Notes on Software Engineering*. – 2016. – Vol. 4. – No. 3. – Pp. 193–198.
40. Hovorushchenko, T. Ontological approach to the assessment of information sufficiency for software quality determination [Text] / T. Hovorushchenko, O. Pomorova // *CEUR-WS*. – 2016. – Vol. 1614. – Pp. 332–348.
41. Wooldridge, M. Intelligent agents – theory and practice [Text] / M. Wooldridge, N. R. Jennings // *Knowledge Engineering Review*. – 1995. – Vol. 10. – Issue 2. – Pp. 115–152.
42. Lytvyn, V. Approach to decision support intelligent systems development based on ontologies [Text] / V. Lytvyn, O. Oborska, R. Vovnjanka // *Econ-techmod : Polish Academy of Sciences, Branch in Lublin*. – Poland, 2015. – Vol. 4. – No 4. – Pp. 29–35.
43. Freitas, A. Applying ontologies to the development and execution of Multi-Agent Systems / A. Freitas, A. R. Panisson, L. Hilgert, F. Meneguzzi, R. Vieira, R. H. Bordini // *Web Intelligence*. – 2017. – Vol. 15. – Issue 4. – Pp. 291–302.
44. Freitas, A. Model-driven engineering of multi-agent systems based on ontologies [Text] / A. Freitas, R. H. Bordini, R. Vieira // *Applied Ontology*. – 2017. – Vol. 12. – Issue 2. – Pp. 157–188.
45. Ossowska, K. Exploring an Ontological Approach for User Requirements Elicitation in the Design of Online Virtual Agents [Text] / K. Ossowska, L. Szewc, P. Weichbroth, I. Garnik, M. Sikorski // *Information Systems: Development, Research, Applications, Education*. – 2017. – Vol. 264. – Pp. 40–55.
46. Lezcano-Rodriguez, L. A. Ontological characterization of basics of KAOS chart from natural language [Text] / L. A. Lezcano-Rodriguez, J. A. Guzman-Luna // *Revista Iteckne*. – 2016. – Vol. 13. – Issue 2. – Pp. 157–168.
47. Hilaire, V. An approach for the integration of swarm intelligence in MAS: An engineering perspective [Text] / V. Hilaire, M. Cossentino, F. Gechter, S. Rodriguez, A. Koukam // *Expert Systems with Applications*. – 2013. – Vol. 40. – Issue 4. – Pp. 1323–1332.
48. Garcia-Magarino, I. An ontological and agent-oriented modeling approach for the specification of intelligent Ambient Assisted Living systems for Parkinson patients [Text] / I. Garcia-Magarino, J. J. Gomez-Sanz // *Hybrid Artificial Intelligent Systems*. – 2013. – Vol. 8073. – Pp. 11–20.
49. Wilk, S. A task-based support architecture for developing point-of-care clinical decision support systems for the emergency department [Text] / S. Wilk, W. Michalowski, D. O'Sullivan, K. Farion, J. Sayyad-Shirabad, C. Kuziemsy, B. Kukawka // *Methods of Information in Medicine*. – 2013. – Vol. 52. – Issue 1. – Pp. 18–32.
50. Rakib, A. A formal approach to modelling and verifying resource-bounded context-aware agents [Text] / A. Rakib, R. U. Faruqui // *Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering*. – 2013. – Vol. 109. – Pp. 86–96.

References

1. Hastie, S. and Wojewoda S. (2015), “Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch”, available at: <http://www.infoq.com/articles/standish-chaos-2015>
2. Project Smart (2014), “The Standish Group Report CHAOS”, available at: <https://www.projectsmaart.co.uk/white-papers/chaos-report.pdf>
3. Speed&Function (2017), “A Look at 25 Years of Software Projects. What Can We Learn?”, available at: <https://speedandfunction.com/look-25-years-software-projects-can-learn/>
4. McConnell, S. (2013), *Code complete*, Microsoft Press, Redmond.
5. Levenson, N. G. (2012), *Engineering a safer world: systems thinking applied to safety*, MIT Press, Massachusetts.
6. Shamieh, C.(2014), *Systems engineering for dummies*, John Wiley & Sons, New York.
7. Pyrig, S. O. and Nuzhna, O. A. (2014), “Information technologies and their use at enterprises of Ukraine [Informaciyni tekhnologiyi ta yih vykorystannya na pidpriemstvakh Ukrainy], *Economical Forum*, No. 3, pp. 190–195.
8. Shakhovska, N. and Moskalyuk V. (2014), “Big data as information technology” [Velyki dani yak informaciyna tekhnologiya], *Proceedings of the 3-rd International Conference of Information, Communication, Society, May 21–24, 2014, Lviv*, pp. 188–189.
9. Sivarajah, U., Kamal, M. M., Irani, Z. and Weerakkody, V. (2017), “Critical analysis of Big Data challenges and analytical methods”, *Journal of Business Research*, Vol. 70, pp. 263–286.
10. Mauerhoefer, T., Strese, S. and Brettel, M. (2017), “The impact of information technology on new product development performance”, *Journal of Product Innovation Management*, Vol. 34 Issue 6, pp. 719–738.
11. Kinch, M. W., Melis, W. J. C. and Keates S. (2017), “Reviewing the current state of machine learning for artificial intelligence with regards to the use of contextual information”, paper presented at The Second Medway Engineering Conference on Systems: Efficiency, Sustainability and Modelling, University of Greenwich, Greenwich.
12. Noor, A. K. (2015), “Potential of cognitive computing and cognitive systems”, *Open Engineering*, Vol. 5 Issue 1, pp. 75–88.

13. LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, Vol. 521 Issue 7553, pp. 436–444.
14. Ristoski and P. Paulheim, H. (2016), "Semantic Web in data mining and knowledge discovery: A comprehensive survey", *Journal of WEB Semantics*, Vol. 36, pp. 1–22.
15. Press, G. (2017), "Top 10 hot artificial intelligence (AI) technologies", available at: <https://www.forbes.com/sites/gilpress/2017/01/23/top-10-hot-artificial-intelligence-ai-technologies/#1d5da9561928>.
16. Minukhin, S. V. (2006), "Formation of information support for the management system of the enterprise business process" [Formirovaniye informacionnogo obespecheniya sistemi upravleniya biznes-protseami predpriyatiya], *Actual Problems of Economics*, No. 10, pp. 170–178.
17. Golub, K. (2015), *Subject access to information: An interdisciplinary approach*, Libraries Unlimited, Westport.
18. Hovorushchenko, T. (2018), "Information Technology for Assurance of Veracity of Quality Information in the Software Requirements Specification", *Advances in Intelligent Systems and Computing*, Vol. 689, pp. 166–185.
19. Pomorova, O. and Hovorushchenko, T. (2013), "Intelligent assessment and prediction of software characteristics at the design stage", *American Journal of Software Engineering and Applications*, Vol. 2 Issue 2, pp. 25–31.
20. Ishimatsu, T., Levenson, N. G., Thomas, J. P., Fleming, C. H., Katahira, M., Miyamoto, Yu., Ujiie, R., Nakao, H. and Hoshino, N. (2014), "Hazard analysis of complex spacecraft using systems-theoretic process analysis", *Journal of Spacecraft and Rockets*, Vol. 51 No. 2, pp. 509–522.
21. Levenson, N. (2009), "Software challenges in achieving space safety", *Journal of the British Interplanetary Society*, Vol. 62, pp. 265–272.
22. Jones, C. and Bonsignour, O. (2012), *The economics of software quality*, Pearson Education, Boston.
23. Yourdon, E. (2003), *Death march: The complete software developer's guide to surviving "Mission impossible" projects*, 2nd ed., Prentice Hall, New Jersey.
24. Chen, A. and Beatty, J. (2012), *Visual models for software requirements*, Microsoft Press, Redmond.
25. Fatwanto, A. (2013), "Software requirements specification analysis using natural language processing technique", *Proceedings of The 13-th IEEE International Conference on Quality in Research, June 25-28, 2013, Yogyakarta*, pp. 105–110.
26. Rehman, T., Khan, M. N. A. and Riaz, N. (2013), "Analysis of requirement engineering processes, tools/techniques and methodologies", *International Journal of Information Technology and Computer Science*, Vol. 5 No. 3, pp. 40–48.
27. Bass, L., Bergey, J., Clements, P., Merson, P., Ozkaya, I. and Sangwan, R. (2006), "A comparison of requirements specification methods from a software architecture perspective", available at: http://resources.sei.cmu.edu/asset_files/TechnicalReport/2006_005_001_14786.pdf.
28. Hovorushchenko, T. and Krasiy, A. (2015), "Method of evaluating the success of software project implementation based on analysis of specification using neuronet information technologies", *CEUR-WS*, Vol. 1356, pp. 100–107.
29. Gruber, T. (1993), "A translation approach to portable ontologies", *Knowledge Acquisition*, No. 5 (2), pp. 199–220.
30. Andon, F. I., Grishanova, I. Yu. and Reznichanko, V. A. (2008), "Semantic Web as the new model of the Internet information space" [Semantic Web kak novaya model informacionnogo prostranstva Internet], *Problems of Programming*, No. 2-3, pp. 417–430.
31. Burov, E. (2014), "Complex ontology management using task models", *International Journal of Knowledge-Based and Intelligent Engineering Systems*, Vol. 18 No. 2, pp. 111–120.
32. Shostak, I. and Butenko, I. (2012), "Ontology approach to realization of information technology for normative profile forming at critical software certification", *Collection of Scientific Works of the Military Institute of the KNU named after T. G. Shevchenko*, No. 38, pp. 250–253.
33. Babenko, L. (2009), "Ontological approach to the specification of the properties of software systems and their components" [Ontologicheskii podhod k specifikacii svoistv programmnih sistem i ih komponentov], *Cybernetics and System Analysis*, No. 1, pp. 180–187.
34. Li, Y. and Cleland-Huang, J. (2013), "Ontology-based trace retrieval", *Proceedings of The 7-th International Workshop on Traceability in Emerging Forms of Software Engineering, May 19, 2013, Passau*, pp. 30–36.
35. Assawamekin, N., Namfon, A., Sunetnanta, T. and Pluempitiwiriawej, C. (2010), "Ontology-based multiperspective requirements traceability framework", *Knowledge Information Systems*, No. 3, pp. 493–522.
36. Zhang, Y. G., Witte, R., Rilling, J. and Haarslev, V. (2008), "Ontological approach for the semantic recovery of traceability links between software artefacts", *IET Software*, Vol. 2 Issue 3, pp. 185–203.
37. Leonid, K., Gacitua, R., Rouncefield, M. and Sawyer, P. (2010), "Ontology and model alignment as a means for requirements validation", *Proceedings of The 4-th IEEE International Conference on Semantic Computing, September 22-24, 2010, Pittsburgh*, pp. 46–51.

38. Jin, D. and Cordy, J. R. (2005), "Ontology-based software analysis and reengineering tool integration: The OASIS service-sharing methodology", *Proceedings of The 21-st IEEE International Conference on Software Maintenance, September 25-30, 2005, Budapest*, pp. 613–616.
39. Bajnaid, N. O., Benlamri, R., Pakstas, A. and Salekzamankhani, Sh. (2016), "An ontological approach to model software quality assurance knowledge domain", *Lecture Notes on Software Engineering*, Vol. 4 No. 3, pp. 193–198.
40. Hovorushchenko, T. and Pomorova, O. (2016), "Ontological approach to the assessment of information sufficiency for software quality determination", *CEUR-WS*, Vol. 1614, pp. 332–348.
41. Wooldridge, M. and Jennings, N. R. (1995), "Intelligent agents – theory and practice", *Knowledge Engineering Review*, Vol. 10 Issue 2, pp. 115–152.
42. Lytvyn, V., Oborska, O. and Vovnjanka R. (2015), "Approach to decision support intelligent systems development based on ontologies", *Econ-techmod*, Vol. 4 No 4, pp. 29–35.
43. Freitas, A., Panisson, A. R., Hilgert, L., Meneguzzi, F., Vieira, R. and Bordini, R. H. (2017), "Applying ontologies to the development and execution of Multi-Agent Systems", *Web Intelligence*, Vol. 15 Issue 4, pp. 291–302.
44. Freitas, A., Bordini, R. H. and Vieira, R. (2017), "Model-driven engineering of multi-agent systems based on ontologies", *Applied Ontology*, Vol. 12, Issue 2, pp. 157–188.
45. Ossowska, K., Szewc, L., Weichbroth, P., Garnik, I. and Sikorski M. (2017), "Exploring an Ontological Approach for User Requirements Elicitation in the Design of Online Virtual Agents", *Information Systems: Development, Research, Applications, Education*, Vol. 264, pp. 40–55.
46. Lezcano-Rodriguez, L. A. and Guzman-Luna, J. A. (2016), "Ontological characterization of basics of KAOS chart from natural language", *Revista Iteckne*, Vol. 13 Issue 2, pp. 157–168.
47. Hilaire, V., Cossentino, M., Gechter, F., Rodriguez, S. and Koukam, A. (2013), "An approach for the integration of swarm intelligence in MAS: An engineering perspective", *Expert Systems with Applications*, Vol. 40 Issue 4, pp. 1323–1332.
48. Garcia-Magarino, I. and Gomez-Sanz, J. J. (2013), "An ontological and agent-oriented modeling approach for the specification of intelligent Ambient Assisted Living systems for Parkinson patients", *Hybrid Artificial Intelligent Systems*, Vol. 8073, pp. 11–20.
49. Wilk, S., Michalowski, W., O'Sullivan, D., Farion, K., Sayyad-Shirabad, J., Kuziemy, C. and Kukawka, B. (2013), "A task-based support architecture for developing point-of-care clinical decision support systems for the emergency department", *Methods of Information in Medicine*, Vol. 52 Issue 1, pp. 18–32.
50. Rakib, A. and Faruqui, R. U. (2013), "A formal approach to modelling and verifying resource-bounded context-aware agents", *Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering*, Vol. 109, pp. 86–96.

CURRENT PROBLEMS OF EVALUATING THE INITIAL STAGES OF THE SOFTWARE LIFECYCLE

T. O. Hovorushchenko¹, O. O. Pavlova²

¹ Khmelnytskyi National University

² Khmelnytskyi Gymnasium No. 2

Abstract. The conducted analysis of information technologies for the software engineering domain has shown the need for the development of information technologies of the new generation, in which the person is eliminated from the process of processing the information and gaining the knowledge. The conducted study of the impact of the initial stages of the lifecycle on the success of the software projects revealed that the ability to automate the evaluation of the level of elaboration of the initial stages of the software lifecycle based on the analysis of the specifications (in particular, the automated evaluation of the sufficiency of information in the software requirements specifications) is actual and very important. The conducted analysis of known methods and automated tools of analysis of the software requirements specifications showed that they are not acceptable for the evaluation of the level of elaboration of the initial stages of the software lifecycle based on the analysis of specifications. The conducted study of ontologies for evaluating the initial stages of the software lifecycle revealed the interest of researchers to intelligent agents on the basis of the ontological approach as the key tools for eliminating the person from the processes of information processing and knowledge gaining. Nowadays, the numbers of ontological models, methods and tools (agents) have been developed, but they belong to different methodological approaches and don't integrate with each other, that is, there is currently no agent-oriented information technology for evaluating the initial stages of the software lifecycle based on the ontologi-

cal approach. The actuality of the task of automated evaluating the level of elaboration of the initial stages of the software lifecycle on the basis of the analysis of specifications (in particular, the automated evaluating the sufficiency of information in the software requirements specification), and the lack of information technology of evaluating the initial stages of the software lifecycle causes the need of development of the agent-oriented information technology for evaluating the initial stages of the software lifecycle based on the ontological approach.

Keywords: software, software project, success of software project, initial stages of software lifecycle, information technology, ontology, intelligent agent.

СОВРЕМЕННЫЕ ПРОБЛЕМЫ ОЦЕНИВАНИЯ НАЧАЛЬНЫХ ЭТАПОВ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Т. А. Говорущенко¹, О. А. Павлова²

¹Хмельницький національний університет

²Хмельницька гімназія №2

Аннотация. Проведены анализ информационных технологий для области инженерии программного обеспечения (ПО), исследование влияния начальных этапов жизненного цикла на успешность ПО, анализ методов и средств и исследование онтологий для оценивания начальных этапов жизненного цикла ПО, которые показали необходимость разработки агентно-ориентированной информационной технологии оценивания начальных этапов жизненного цикла ПО на основе онтологического подхода.

Ключевые слова: программное обеспечение (ПО), программный проект, успешность ПО, начальные этапы жизненного цикла ПО, информационная технология, онтология, интеллектуальный агент.

Отримано 09.03.2018



Говорущенко Тетяна Олександрівна, кандидат технічних наук, старший науковий співробітник, доцент, завідувач кафедри комп'ютерної інженерії та системного програмування Хмельницького національного університету. Вул. Інститутська, 11, Хмельницький, Україна, E-mail: tat_yana@ukr.net, t.hovorushchenko@khnu.km.ua, тел. +38-095-11-22-544

Tetiana Hovorushchenko, PhD, Senior Researcher, Associate Professor, Head of Computer Engineering and System Programming Department, Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, Ukraine

ORCID ID: 0000-0002-7942-1857



Павлова Ольга Олександрівна, вчитель інформатики Хмельницької гімназії №2. Пр. Миру, 84/2, Хмельницький, Україна, E-mail: olya1607pavlova@gmail.com, тел. +38-096-847-04-51

Olga Pavlova, Teacher of Informatics of Khmelnytskyi Gymnasium No. 2, Myru avenue, 84/2, Khmelnytskyi, Ukraine

ORCID ID: 0000-0003-2905-0215