

ДОСЛІДЖЕННЯ ВІДОМИХ МОДЕЛЕЙ ОЦІНЮВАННЯ ХАРАКТЕРИСТИК ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Автором статті проведено дослідження моделей надійності, складності та якості програмного забезпечення, а також критеріїв вибору моделей для конкретного програмного забезпечення.

Ключові слова: програмне забезпечення (ПЗ), моделі якості програмного забезпечення, моделі складності програмного забезпечення, моделі надійності програмного забезпечення, критерії вибору моделей.

Author conducted the research of software reliability, complexity and quality models, and the criteria of selection of the models for a particular software.

Keywords: software (SW), model of software quality, model of software complexity, model of software reliability, selection criteria model.

Вступ

До характеристик ПЗ належать: вартість ПЗ, захист ПЗ, повнота реалізації вимог, обсяг файлів ПЗ, вимоги до системного програмного забезпечення та технічних засобів, обсяг потрібної оперативної та дискової пам'яті. Але безумовно наважливішими характеристиками ПЗ є його надійність, складність та якість.

Надійність ПЗ - це властивість програми виконувати задані функції в заданих умовах роботи і на заданій ЕОМ [1]. За імовірнісним підходом під надійністю розуміють імовірність того, що при функціонуванні системи протягом певного часу не буде виявлено помилок. Проте тут не враховано відмінність між помилками різних типів. Надійність ПЗ краще розглядати з точки зору впливу помилок на користувача системи. Усе залежить від того, на якому етапі (розроблення чи експлуатація) і в якому компоненті припущено помилку. Згідно стандарту [2], *надійність ПЗ* рекомендується характеризувати рівнем завершеності, стійкістю до помилок та можливістю перезапуску. Згідно стандарту [3], *надійність ПЗ* - сукупність властивостей, яка характеризує здатність програмного засобу зберігати заданий рівень придатності в заданих умовах протягом заданого інтервалу часу. Отже, надійність ПЗ є показником якості, який характеризує властивість ПЗ виявляти в процесі експлуатації помилки, що залишились в ньому, за певної сукупності початкових даних [1].

Складність ПЗ включає в себе численні властивості програмного забезпечення, які впливають на внутрішні взаємодії. *Програмна складність* характеризується довжиною програми або обсягом пам'яті ЕОМ, необхідної для розміщення ПЗ. Можна оцінити *складність* часом виконання програми для великого вхідного аргумента [4]. Складність ПЗ визначається: складністю реального світу; складністю керування процесом розроблення; гнучкістю програмного забезпечення; складністю опису поведінки систем.

Якість ПЗ - це характеристика ПЗ, яка відображає ступінь його відповідності вимогам. При цьому вимоги можуть трактуватись досить широко, що породжує цілий ряд незалежних визначень поняття якості. Згідно визначення ISO [5], *якість* - це ступінь відповідності присутніх характеристик вимогам. Згідно [6], *якість* - це повнота властивостей і характеристик продукту, процесу або послуги, які забезпечують здатність задовольняти оголошеним або передбачуваним потребам. Згідно [7], *якість ПЗ* - це ступінь, в якій воно володіє потрібною комбінацією властивостей. Згідно [3], *якість програмного засобу* - це сукупність властивостей програмного засобу, які обумовлюють його придатність задовольняти задані або передбачувані потреби у відповідності до його призначення. Стандарт [8] дає наступне визначення якості ПЗ - весь обсяг ознак та характеристик програмної продукції, який належить до її здатності задовольняти встановлені потреби.

Саме моделюванню та визначенню надійності, складності та якості ПЗ і присвячено дану статтю.

1. Моделі надійності програмного забезпечення

Модель надійності ПЗ - це математична модель, побудована для оцінювання залежності надійності ПЗ від деяких певних параметрів [9]. Значення таких параметрів або відомі, або можуть бути виміряні під час спостережень або експериментального дослідження процесу функціонування ПЗ. Термін "модель надійності ПЗ" може бути також використаний застосовно до математичної залежності між певними параметрами, які хоча й мають відношення до оцінки надійності ПЗ, але не містять її характеристик в явному вигляді. Наприклад, поведінка деякої гілки ПЗ на підмножині наборів вхідних даних, за допомогою яких ця гілка контролюється, істотно пов'язана з надійністю ПЗ, однак характеристики цієї поведінки можуть бути оцінені незалежно від оцінки самої надійності. Іншим параметром є частота помилок, яка дозволяє оцінити *якість* систем реального часу і в той же час одержувати лише непряму інформацію відносно надійності ПЗ.

Одним з видів моделі надійності ПЗ, яка заслуговує особливої уваги є так звана феноменологічна або емпірична модель [9]. При розробленні таких моделей вважається, що зв'язок між надійністю та іншими параметрами - статичний. За допомогою таких моделей слід намагатись оцінити кількісно ті характеристики

ПЗ, які свідчать про високу або низьку його надійність. Наприклад, параметр "складність ПЗ" характеризує ступінь зменшення рівня його надійності, адже складність ПЗ завжди призводить до помилок.

Моделі надійності програмного забезпечення (МНПЗ) поділяються на аналітичні та емпіричні [9]. Аналітичні моделі дають змогу розрахувати кількісні показники надійності, базуючись на даних поведінки ПЗ в процесі тестування (моделі вимірювання та оцінювання). Емпіричні моделі базуються на аналізі структурних особливостей програм. Розвиток емпіричних моделей дозволяє виявляти взаємозв'язок між складністю ПЗ та його надійністю. Ці моделі можна використовувати на етапі проектування ПЗ, коли виконане розбиття на модулі та відома структура ПЗ. Класифікація моделей надійності ПЗ наведена на рис. 1.

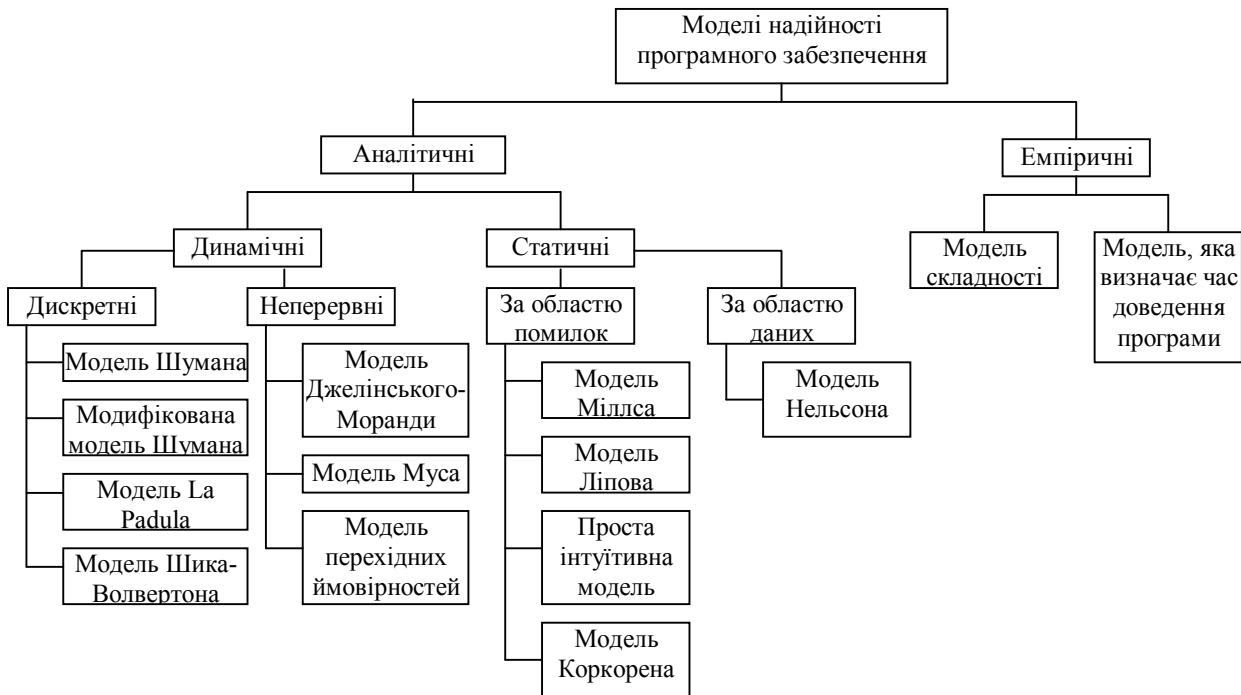


Рис.1. Класифікація моделей надійності програмного забезпечення

Проблема вибору моделей надійності для використання на етапі експлуатації і супроводу ПЗ практично не досліджена. Причинами цього є велика кількість моделей та недостатній рівень вивчення особливостей їх використання.

2. Моделі складності програмного забезпечення

Основні види складності ПЗ наведено на рис.2.

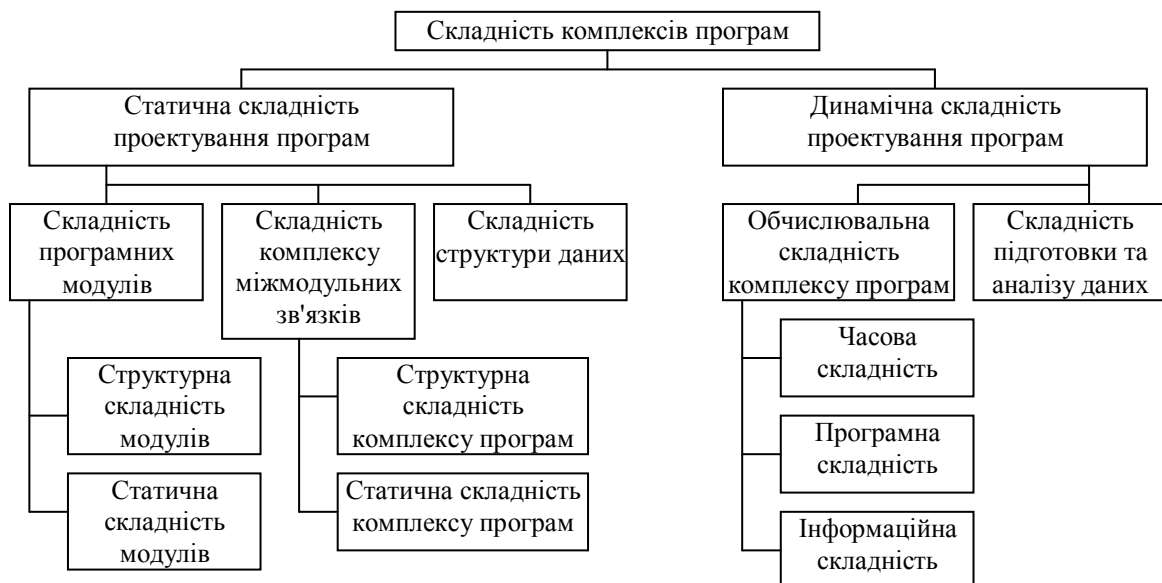


Рис. 2. Основні види складності програмного забезпечення

Емпіричні правила для оцінки складності алгоритму [4]: асимптотична складність - описується час

виконання алгоритму з точки зору кількості основних кроків; якщо час виконання є сумою мультиплікативних факторів, то враховується один з більшою швидкістю зростання; якщо останні терми є добутком, то будь-які мультиплікативні фактори не враховуються; використовується нотація $\text{big } O$ для верхньої границі асимптотичного зростання функції.

Класи складності алгоритмів (в порядку зростання складності) [4]: $O(1)$ - постійна складність; $O(\log n)$ - логарифмічна складність; $O(n)$ - лінійна складність; $O(n \log n)$ - логарифмічно-лінійна складність; $O(n^c)$, де $c = \text{const}$ - поліноміальна складність; $O(c^n)$, де $c = \text{const}$, яка зростає як степінь на основі розміру входів - експоненційна складність.

Моделі складності ПЗ засновані на гіпотезі, що рівень безпомилковості ПЗ може бути спрогнозований за допомогою показників (метрик) складності ПЗ [7]. Детально метрики складності та їх ефективність розглядалися автором у [10]. В якості аргументів моделей використовуються метрики складності ПЗ. Класифікацію моделей визначення складності ПЗ можна представити у наступному вигляді (рис.3).

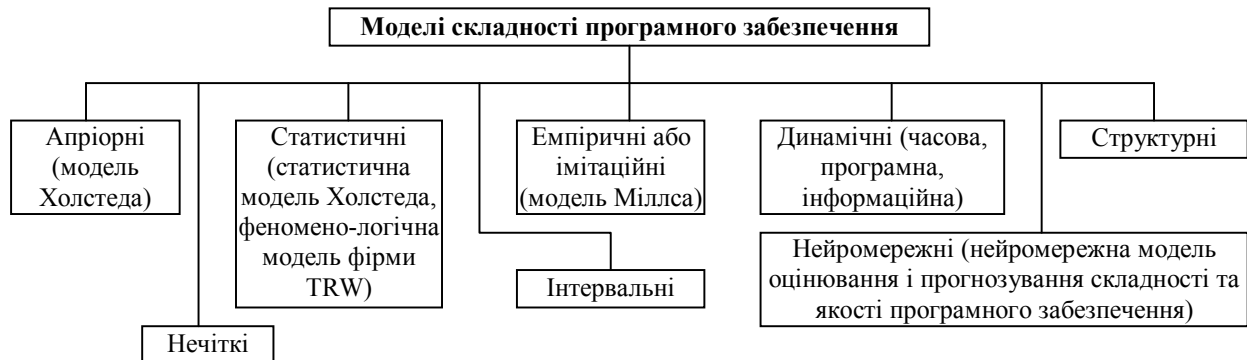


Рис.3. Класифікація моделей складності програмного забезпечення

Найбільш відомою *априорною моделлю* складності ПЗ є модель *Холстеда* (Halstead) [11]. До простих *статистичних моделей* складності належить *феноменологічна модель фірми TRW* [12]. До недоліків багатофакторних моделей належить можливість одержання ефекту "прокляття розмірності" [11] при великій кількості факторів або нелінійному вигляді апроксимуючого полінома. *Емпіричні (імітаційні) моделі* [13] базуються на аналізі структурних особливостей програмного забезпечення. Такі моделі часто не дають кінцевих результатів показників складності, якості та надійності, але їх використання на етапі проектування ПЗ корисне для прогнозування потрібних ресурсів тестування, уточнення планових термінів завершення проекту і т.і. Для складних модулів та для великих багатомодульних програм складається *імітаційна модель*, програма якої "засівається" помилками і тестується за випадковими входами. Недоліками такої моделі є висока вартість методу, оскільки він вимагає додаткових витрат на складання імітаційної моделі, і наближений характер одержаних показників. *Динамічна (або обчислювальна) складність* характеризує процес виконання програми і має три складових [14]: 1) часова - визначається часом виконання програми або часом її реакції на запит користувача; 2) програмна - визначається складом та способом взаємодії процедур або модулів, які утворюють програму, а також можливістю їх розташування в кеш-пам'яті, основній пам'яті або на диску, а у випадку розподілених додатків - розташуванням програми на комп'ютерах мережі; 3) інформаційна - визначається складністю організації даних та доступу до них, а також особливостями їх розташування в кеш-пам'яті, основній пам'яті, на диску або на мережевому сервері. Крім розглянутих моделей існують також структурні моделі, нечіткі моделі, інтервальні моделі складності ПЗ, а також нейронні мережі, які застосовуються для розв'язку задачі оцінювання складності ПЗ в окремих випадках [15].

Для вибору потрібної моделі використовуються якісні та кількісні критерії [11]. *Якісні критерії*: 1) простота використання; 2) достовірність; 3) застосовність для розв'язку різних задач; 4) простота реалізації.

Кількісні критерії: 1) показники точності оцінювання; 2) показники якості прогнозуючих моделей (збіжність, стійкість, точність прогнозу, узгодженість); 3) інформаційні критерії якості прогнозуючих моделей (розмірність, критерії BIC/AIC); 4) комбіновані та інтегральні показники.

Дослідження показало, що існує велика кількість математичних моделей складності ПЗ, які дозволяють одержати оцінки показників технологічної безпеки ПЗ на різних етапах життєвого циклу, що важливо при плануванні витрат на інформаційну безпеку. Розглянута класифікація моделей дозволяє на практиці зорієнтуватись при виборі моделей в залежності від одержаної статистики.

3. Моделі якості програмного забезпечення

Основним стандартом якості в області інженерії ПЗ є стандарт [16], який визначає номенклатуру, атрибути і метрики вимог якості ПЗ. Цей стандарт є одним з визначальних факторів при моделюванні якості ПЗ. На додаток до нього випущено набір стандартів [17], які регламентують способи оцінки цих характеристик. В сукупності вони утворюють *модель якості*, відому під назвою *SQuaRE* (Software Quality Requirements and Evaluation). Загальний підхід до моделювання якості програмного забезпечення полягає у

тому, щоб спочатку ідентифікувати невеликий набір атрибутів якості найвищого рівня абстракції та потім у напрямку "згори-донизу" розбити ці атрибути на набори підлеглих атрибутів. Нижній рівень ієрархії представляє собою безпосередньо атрибути ПЗ, які підлягають точному опису та вимірюванню. Вимоги якості в свою чергу можуть бути представлені як обмеження на модель якості.

Модель якості SQuaRE та її основні характеристики якості за стандартом [16] представлена на рис.4.

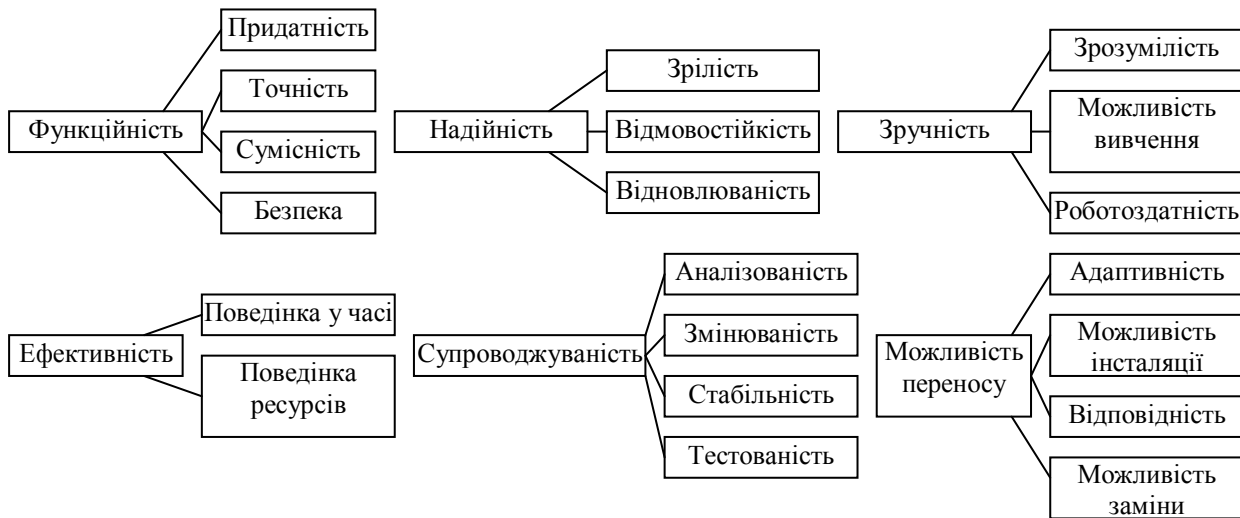


Рис.4. Модель якості ПЗ за стандартом ISO/IEC 25010:2011

На сьогодні в галузі забезпечення якості ПЗ створено цілу низку нових стандартів та методологій (Capability Maturity Model (CMM), ISO/IEC 15504 (SPICE), Bootstrap, Trillium, ISO 12207).

Для визначення загального рівня розвитку технологічних процесів в програмних організаціях розробили спеціальну систему оцінки зрілості технологічних процесів в софтверних організаціях - *модель Capability Maturity Model (CMM)*, засновану на так званих рівнях зрілості (maturity levels) [18]. Модель CMM має п'ять рівнів зрілості, і кожен з них характеризує певний ступінь якості програмних продуктів [18]: 1) початковий - відсутнє стабільне середовище розробки і супроводження; не витримуються терміни випуску продуктів; всі сили спрямовано на кодування і тестування програми (75% софтверних організацій); 2) повторюваний - жорстке керування, планування і контроль; акцент робиться на початкові вимоги, методи оцінки і конфігураційний менеджмент (15%); 3) фіксований - процеси повністю документовані, стандартизовані і інтегровані в єдиний технологічний потік (8%); 4) керований - намагання оцінити якість процесів і готового продукту кількісно; для контролю над процесами використовуються метрики (1.5%); 5) оптимізований - намагання покращення роботи, керуючись кількісними критеріями якості; основна мета - випуск бездефектних продуктів, в яких помилки усунені ще на стадії внутрішнього тестування (0.5%).

Модель SPICE (Software Process Improvement and Capability dEtermination) [19] створено ISO як єдиний стандарт оцінки програмних процесів. SPICE нагадує CMM. Основною задачею софтверної організації теж є постійне покращення процесу розробки ПЗ, також використовується схема з різними рівнями можливостей (в SPICE визначено 6 різних рівнів), але ці рівні застосовуються не лише до організації в цілому, але й до окремо взятих процесів. Однією з переваг SPICE є його вільне поширення на офіційному сайті SPICE - [20].

В останні роки запропонований та розвивається так званий "*мовно-орієнтовний підхід (МОП) до вимірювань та оцінки ПЗ*", який дозволяє створювати автоматичні, засновані на знаннях, програмні технології оцінки ПЗ, що базуються на однозначному, "прозорому" та семантично коректному описі понять прикладної області "якість програмного забезпечення" [21]. МОП підтримує всі відомі стандарти, моделі і метрики програмного забезпечення.

Висновки

В ході дослідження моделей надійності, складності та якості програмного забезпечення автором виявлено наступні проблеми: 1) недослідженість моделей надійності ПЗ для використання на етапі проектування ПЗ; 2) недостатній рівень вивчення особливостей використання моделей надійності ПЗ; 3) відсутність критеріїв вибору моделей складності ПЗ на етапі проектування; 4) неможливість висування до моделей складності ПЗ високих вимог щодо точності через їх динамічність, складність та різноманітність; 5) інтуїтивний характер моделей складності ПЗ при прийнятті рішень по плануванню тестування ПЗ на всій множині вхідних даних; 6) ґрунтування моделей складності ПЗ на гіпотезі, що рівень безпомилковості ПЗ може бути спрогнозований за допомогою метрик складності ПЗ, для яких: відсутні єдині стандарти, існує проблема складності інтерпретації величин, рівень автоматизації аналізу та опрацювання на етапі проектування залишається низьким; 7) технологія вимірювання якості ще не досягла зрілості - лише 1.5% софтверних організацій намагаються оцінити якість процесів і готового продукту кількісно, за допомогою метрик, тобто знаходяться на керованому рівні моделі CMM, і лише 0.5% софтверних організацій намагаються

покращити роботу, керуючись кількісними критеріями якості з метою випуску бездефектних продуктів, тобто знаходяться на оптимізованому рівні моделі CMM; 8) недостатня деталізованість стандартів серії ISO 9000 та моделі SQuaRE, можливість різних трактувань стандарту та моделі в залежності від уявлень аудитора; 9) неточність оцінки якості процесів за моделлю SQuaRE, задіяних при створенні та впровадженні ПЗ; 10) відсутність у моделі SQuaRE механізмів, які сприяють покращенню існуючих процесів; 11) модель CMM є власністю Software Engineering Institute (SEI) і не є загальнодоступною, тому подальша розробка моделі ведеться без залучення програмістської спільноти, причому оцінка якості процесів організацій за моделлю CMM може проводитись лише спеціалістами, які пройшли спеціальне навчання та акредитовані SEI; 12) модель CMM орієнтована на застосування у відносно великих софтверних компаніях.

Саме через невирішеність цих питань поки що неможливо бути впевненими у вірному виборі моделей надійності, складності та якості програмного забезпечення на етапі проектування, тобто одержати точні оцінки надійності, складності та якості проекту та розроблюваного за проектом програмного забезпечення.

Перспективним напрямком подальших досліджень у галузі моделювання оцінювання характеристик програмного забезпечення є розроблення критеріїв вибору моделей надійності, складності та якості програмного забезпечення на етапі проектування.

Література

1. Говорушенко Тетяна Олександрівна. Підвищення достовірності процесу тестування програмних продуктів на основі нейромережних інформаційних технологій: дис. ... канд. техн. наук: 05.13.06 / Говорушенко Тетяна Олександрівна. - Хмельницький, 2007. - 186 с.
2. DTS-пакеты // [Електронний ресурс] - Режим доступу: <http://cadmium.ru/content/view/1514/50>
3. ГОСТ 28806-90. Качество программных средств. Термины и определения. // [Електронний ресурс] - Режим доступу: <http://vsegost.com/Catalog/10/10605.shtml>
4. MITx: 6.00x Introduction to Computer Science and Programming: A Simplistic Introduction to Algorithmic Complexity // [Електронний ресурс] - Режим доступу: https://www.edx.org/courses/MITx/6.00x/2012_Fall/book/0
5. ISO 9001:1994 Quality systems - Model for quality assurance in design, development, production, installation and servicing
6. ISO 8402:1994 Quality management and quality assurance
7. IEEE Std. 1061-1998 IEEE Computer Society: Standard for Software Quality Metrics Methodology, 1998
8. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению
9. Благодатских В.А. Стандартизация разработки программных средств: Учебное пособие / В.А.Благодатских, В.А.Волнин, К.Ф.Посакалов. - М.: Финансы и статистика, 2005. - 288 с.
10. Говорушенко Т.О. Оцінювання ефективності метрик складності програмного забезпечення / Т.О. Говорушенко, А.В. Бачинський // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2012 - №2, с.171-179
11. Марков А.С. Модели оценки и планирования испытаний программных средств по требованиям безопасности информации / А.С. Марков // Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение», 2011. Специальный выпуск «Технические средства и системы защиты информации» - с. 90-103
12. Тейер Т. Надежность программного обеспечения. / Т. Тейер, М. Липов, Э. Нельсон - М.: Мир, 1981 - 326 с.
13. Эмпирические модели надежности программного обеспечения // [Електронний ресурс] - Режим доступу: http://info-tehnologii.ru/kac_sr/Mod_nad/Emp_mod/index.html
14. Метрология программного обеспечения // [Електронний ресурс] - Режим доступу: <http://www.studfiles.ru/dir/cat32/subj118/file1716/view2581.html>
15. В.О. Мищенко CASE-оценка критических программных систем. В 3-х томах. Том 1. Качество / Мищенко В.О., Поморова О.В., Говорушенко Т.А. / Под ред. Харченко В.С. - Харьков: НАУ "ХАИ", 2012. - 201 с.
16. ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models
17. ISO/IEC 14598 Information technology - Software product evaluation
18. Стандарты качества и сложности программного обеспечения // [Електронний ресурс] - Режим доступу: http://www.rol.ru/news/it/press/cwm/25_96/teh.htm
19. ISO/IEC 15504: Information Technology - Software Process Assessment
20. Software Process Improvement and Capability Determination (SPICE) // [Електронний ресурс] - Режим доступу: <http://www.sqi.gu.edu.au/spice>
21. Коган Б.И. Автоматизация оценивания качества программного обеспечения // [Електронний ресурс] - Режим доступу: <http://www.febras.ru/~conf/seminar/kogan.html>

Надійшла 24.1.2013 р.

Рецензент: д.т.н. Поморова О.В.