

УДК 004.04

П.С. Слободян, Р.М. Небесний

Тернопільський національний технічний університет імені Івана Пулюя

АКТУАЛЬНІСТЬ ПРОБЛЕМ ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

P. Slobodyan, R. Nebesnyy

THE ACTUALITY OF PROBLEMS OF QUALITY ASSESSMENT OF THE SOFTWARE

Створення сайтів на сьогодні стає все більш розповсюдженою сферою інформаційної діяльності, в якій беруть участь як цілі колективи фахівців, так і окремі особи. Через безперервний ріст кількості веб-сайтів та іншого програмного забезпечення для Web, ускладнення їх структури та збільшення розмірів при одночасному збільшенні відповідальності за виконувани функції, розробники, замовники та користувачі все більше стикаються з проблемами, пов'язаними з забезпеченням та оцінкою якості.

Часто виникає типова ситуація, коли розробники, як правило, мають уявлення про те, як зробити сайт взагалі, але не мають у своєму розпорядженні діючий теоретичний, методичний і технологічний інструмент для постановки й досягнення мети цієї діяльності з оптимальними результатами. В свою чергу замовники сайтів через відсутність типових зразків «гарних» сайтів або критеріїв якості не можуть чітко сформулювати свої вимоги до продукту у відповідності зі специфікою своєї предметної області, підприємства або організації, його цілями й завданнями.

Відсутність документів, що регламентують оцінку якості сайтів, також гостро відчувається у розрізі конкурсних ситуацій. Широке проведення конкурсів сайтів різних рівнів (міжнародних, на рівні країни, міжрегіональних, регіональних, міських тощо) стає сьогодні все більш розповсюдженим явищем.

Перед експертами по оцінці сайтів стоять ті ж проблеми оцінки якості, що й перед тими, хто бере участь у їх створенні. Сутність їх полягає у відсутності необхідної теоретичної бази цієї діяльності. Звідси – домінування на практиці суб'єктивних підходів до оцінки сайтів.

У 1997 році Міжнародна Організація по Стандартизації (International Organization for Standardization – ISO) і Міжнародна Електротехнічна Комісія – МЕК (International Electrotechnical Commission – IEC) створили Сумісний Технічний Комітет з Інформаційних Технологій – Joint Technical Committee on Information Technology (JTC1). Зміст робіт JTC1 визначений як “стандартизація в області систем і устаткування інформаційних технологій”. У 1989 році цей комітет ініціював розробку стандарту ISO/IEC 12207, створивши для цього підкомітет SC7 (SuCommittee 7) по програмній інженерії. Відповідний стандарт вперше був опублікований 1-го серпня 1995 року під заголовком “Software Life Cycle Processes” – “Процеси життєвого циклу програмного забезпечення.

Даний стандарт визначає життєвий цикл як структуру декомпозиції робіт. Організація послідовності робіт – модель життєвого циклу.

Згідно стандарту ISO/IEC 12207 життєвий цикл складається з таких етапів:

- аналіз вимог до системи;
- визначення специфікацій;
- проектування;
- кодування;

- тестування;
- експлуатація та супровід.

Труднощі конструювання реальних застосувань обумовлені їх складністю, і критичну роль в подоланні цієї складності грає сам життєвий цикл. Існує декілька різновидів життєвого циклу, і головний з них – це модель водопаду.

Водоспадна модель процесу розробки. Класичною моделлю процесу розробки програм є модель водоспаду [1], в рамках якої процес представляється послідовністю фаз аналізу вимог, проектування, реалізації, інтеграції і тестування.

Аналіз вимог полягає в зборі вимог до продукту. Результатом аналізу, як правило, є деякий текст.

Проектування описує внутрішню структуру продукту. Такий опис дається у формі діаграм і текстів.

Реалізація – це програмування. Результатом реалізації є програмний код всіх рівнів, будь то код, що генерується високорівневою системою програмування, компілятором мови четвертого покоління або який-небудь інший.

Інтеграція – це процес збірки всього продукту з окремих частин.

Насправді перераховані фази не слідуєть строго послідовно одна за одною, а частково перекриваються. На практиці будь-яку з фаз можна починати до того, як буде повністю завершена попередня.

У чистому вигляді процес водоспаду застосовується достатньо рідко. Основною причиною непридатності процесу водоспаду в чистому вигляді є складність більшості застосувань. Проте процес водоспаду є основою для більшості інших різновидів процесу.

Процеси, в яких схема водоспаду застосовується багато разів, називаються ітеративними. В ітеративних процесах не обов'язково всі кроки схеми водопаду повинні виконуватися на кожній ітерації.

Спіральна модель процесу розробки. У разі спірального процесу [1] послідовність аналіз вимог – проектування – реалізація – тестування виконується більше одного разу. Для цього може бути декілька причин. Основна причина зазвичай пов'язана з необхідністю попередження ризиків. Іншою причиною може бути необхідність надати замовникові часткову версію проекту для отримання відгуків і побажань.

Додаткова перевага ітеративних процесів полягає в можливості збирати на кожній ітерації метричні характеристики процесу. Наприклад, маючи в своєму розпорядженні дані про час, який було потрібно для виконання першої ітерації, ми можемо уточнити план-графік подальшої роботи. Така можливість особливо корисна для організацій, що мають невеликий досвід планування розробок.

Хоча спіральна модель відображає типову схему процесу розробки, вона вимагає майстернішого управління, ніж проста модель водопаду. Одна з труднощів полягає в підтримці цілісності документації, яка повинна бути повністю оновлена і доповнена до кінця кожної ітерації. Зокрема, кожна версія програмного коду повинна реалізовувати документований проект і задовольняти вимогам документації. [2]

Література:

1. Брауде Э. Технология разработки программного обеспечения. – СПб.: Питер, 2004. — 655 с: ил.
CrossRef [Electronic Resource]. – Mode of access : URL : www.crossref.org/ Title from the screen.