A new approach for estimation of software testing process based on software requirements

Arupratan Santra
Infosys Technologies Ltd, Plot # 129, Nakshatra, Balapur, Keshavgiri PO, Hyderabad 500 005, India

Received 14 January 2010; revised 19 July 2010; accepted 21 July 2010

This study presents a new approach for estimation of software testing process based on software requirements. It is an initial input in software development life cycle (SDLC). This paper proposes a more reliable and simpler approach for estimating software testing process in application software built in object-oriented (OO) technology.

Keywords: Function point, Number of requirements, Requirement approach, Test case, Use case

Introduction
Software product development adheres to schedule of product development and service industry is very inconsistent with schedule. Many service industries tender the project with less billing hours. So it is very difficult to cut short development effort, which is inevitable. Testing effort is major affected activity in software development life cycle (SDLC) for most of the service industries. Therefore, some projects have been delivered to clients without testing to adhere predefined delivery schedule. Software testing team can start testing process at early stages of SDLC to find defects in every stage of SDLC. Software testing is a major phase in SDLC, where 40-60% undetected defects are detected. Generally, software testing phase contains 30-40% of project activity. Software test effort estimation (STEE) is an integral part of project estimation. Stage-wise effort distribution of software projects is described in pie chart. Project estimation is carried out based on either function points (FP) or lines of code (LOC). Software test effort is observed to be 40% of total project effort, but in practice, it lies between 10-20%. Approaches for STEE include Adhoc Approach, FP Approach and Use Case Approach (UCA). This study proposes a requirement approach (RA) for STEE.

Proposed Approach for Software Testing Process Estimation
For present study, an online patient appointment system was deployed for a hospital in United Arab Emirates (UAE). Through this healthcare portal, user can schedule an appointment from home through internet. As this portal is already developed in English, first option was to deploy the same system in this hospital. But during the requirement gathering and gap analysis, users were mostly speaking local language unlike English. So it became a mandatory to provide portal graphical user interface (GUI) in local Arabic language. In order to test this system in English and Arabic languages, manual and automated approaches were used. STEE was carried out with UCA. But effort variation was found very high with respect to industry standard. This system was developed in OO J2EE technologies. UCA limits to do precise estimation because of less inputs towards requirements as use case definitions. In order to overcome this limitation, STEE was carried out directly from software requirements.

Requirement Approach (RA)
Approach to Develop Test Case vs Requirements Relation
There is no easy STEE approach in software industry. All existing approaches are not proved appropriate enough and have limitations. Thus proposed RA is here to estimate testing effort. This study has been done mainly in OO based software (JAVA and C++ application). Test effort data points from 150 application
developments in wide range in number of requirements were used. Software industry receives requirement as a statement of work (SOW) from customer. As day goes on, customer provides additional requirement to the existing one, called as change request. Every time software project plan has to be revised and same time project effort has to be revised. All existing STEE approaches (FP or UCP) are implemented after freezing requirements. In RA, STEE can be started immediately after requirement phase or for every change request. For each change request, revision of estimation is not so easy in existing approaches. All approaches estimate based
on experience or adhoc but there is no formal method of estimation for each change request requirement. In RA, there is a relation between requirements and test cases. In RA, requirement should be collected or divided into granular level.

Number of system test cases (STC) = N * (number of requirements)

where N is a factor to determine number of STC and it is based on requirement granularity.

Every organization has a specific style of requirement gatherings and based on that this factor varies or one can generate this factor from their base line organization process data (OPD). But if one follows standard test requirement tool like Quality center (QC) then N varies from 2.5 to 4. Software testing team has to select this value based on complexity of requirement. This factor range has been generated after considering wide range of software projects in various domains. Generation of N is based on OPD (Table 1). Data was collected in form of 150 samples from 20 to 1621 requirements.

Results and Discussion

Comparison between FP and Requirements Approach (RA)

An example to support derivation of N factor with respect to FP approach is as follows: Base requirement - Create an employee in employee database.

Case 1: Function Point (FP) Approach
Number of unadjusted FP (UFP) = Number of input * Complexity factor
Number of adjusted FP (AFP) = UFP * Adjusted factor

With respect to above requirement, AFP is calculated as UFP = 1 * 4 (Considering average complexity factor is 4) AFP = UFP * 1 (Considering adjusted factor is 1) = 4 So, number of test cases = AFP ^1.2 = (4)^1.2 = 5.3.

Case 2: Requirement Approach (RA)

Now if one breaks base requirement into micro level then requirements are as follows: i) Add employee details to database; and ii) Validate other required data from database to create a employee. So Number of requirements = 2. Using RA, number of STC = N * (number of requirements). Therefore, number of
STC = 2 * 2.5 (Considering minimum complexity & simple business domain) = 5.

Tools Development for Requirements Approach (RA)

From above example (Table 1), numbers of test cases are same for both the approach. In present study, a relation between STC and unit test cases (UTC) was determined. This was achieved by design of experiments from a set of testing projects having OO technology. Ratio of STC/UTC = 0.4.

Testing efficiency depends on organizations maturity or experiences of testing team. One can measure following parameters in testing efficiency: i) Preparation of STC and UTC per person day; and ii) Execution of test cases (STC & UTC) per person day. Based on this rule an excel macro (Table 2) was developed for STEE. The example has been depicted for 68 requirements. A small program as a test effort estimation model was developed (Fig. 1). Software testing efficiency ($T_E$) parameter considered ($N = 2.5$) are as follows: i) Average preparation of STC per person day = 40; ii) Average preparation of UTC per person day = 120; and iii) Average execution of test cases per person day = 80.

All parameters used in this approach are available in every organization. There is no assumption parameter in this model. Any process-based organization will have OPD to calculate N and $T_E$. This approach is also useful to do software test effort estimation for requirement change request. But other approaches cannot be used directly for updating of software test effort estimation. This approach can be used to derive a quantum pricing model. In this model, there is less risk and higher profit (Fig. 2).

Conclusions

Proposed approach (RA) is easier than FP and UCA. Due to shortage of experts in FP methodology, RA can be easily acceptable by any organization. Without deriving requirement to FP and UCA, one can start STEE from requirement. One can conclude that this model is better in all aspects over traditional pricing model (time & material, fixed price).

References