

Software Estimation Framework for Packaged Products

Shailesh Kumar Shivakumar, Independent Researcher, India

ABSTRACT

Packaged products play a major role in successful implementation of various software projects. Many of the software solutions are built around packaged products. In this paper, the authors propose a novel “software packaged product estimation framework” for an end to end estimation framework for estimating effort for packaged products. The software packaged product estimation framework provides end to end estimation coverage for various project lifecycle stages and supporting activities. The software packaged product estimation framework was used to predict the effort for two projects with MMRE of 0.261 and pred(0.3) of 66.67%.

KEYWORDS

Estimation, Packaged Product Estimation, Product Estimation, Software Engineering

INTRODUCTION

Packaged software products (also referred to as software package or fully packed product) are software tools that provide pre-built functionality to develop the solution application for a particular domain. Specific purpose-built Packaged software products serve as development accelerators reducing overall program cost.

Usage of packaged products such as portal products, CMS products, ecommerce products, digital marketing products is very common in software projects. Packaged products offer pre-built functionality to accelerate the solution development and attain higher productivity by reusing the out of box capabilities. The out of box product functionality can also be extended through configurations and extensions to reuse the product feature in wide variety of scenarios.

In this paper we propose a novel estimation method, “*Software package product estimation framework*” for software projects that employ packaged products for solution development.

The state of the art estimation models do not address the product specific scenarios such as build and unit effort, project lifecycle effort.

The *Software package product estimation framework* provides the estimation guidelines, complexity definition scale and effort calculation formulae for packaged product based software solutions.

Literature Review and Related Work

Generic software estimation models include COCOMO (Boehm, 1981), Function Point/FP (Matson et al., 1994), COCOMO II (Boehm et al., 2000), SEER-SEM (Jenson 1984), SLIM (Putnam & Myers, 1992), PRICE-S (Freiman, 1975), Delphi (Boehm, 1984), Rule-based/Rule of thumb, Use case point (Ochodek, 2011), Work breakdown structure (Jorgensen 2004), Planning poker, Story point estimation, learning (Goldberg 1989), Case based reasoning (Aamodt & Plaza 1994), Analogy based estimation,

DOI: 10.4018/IJPPMPA.2021010102

Select Estimator, top-down estimation, bottom-up estimation, price-to-win, Stepwise ANNOVA (Basha, 2010), Ordinary Least squares (Griffiths et.al.1993). Other software estimation method are machine learning-based estimation (Mair et al., 2000), fuzzy logic based estimation (Gray, 1997), genetic programming (Burgess et al. 2001) and expert based estimating method (Jørgensen, 2004).

For software packaged product estimation, Boehm et al. proposed Constructive Product Line Investment Model (COPLIMO) based on COCOMO 2 model to estimate the effort for broader product lines. In the absence of a specific estimation model, most of the projects use function point or use case point estimation method for packaged product based solutions.

Gaps With State of the Art Techniques

The main gap with state of the art estimations models are as follows:

- State of the art estimation methods do not provide estimation framework for end to end software development lifecycle stages for software projects using packaged products.
- State of the art methods do not provide estimation framework that can be used for wide variety of software products.

METHOD

In this paper we have proposed an estimation framework “*Software packaged product estimation framework*” (SPPEF) that can be used for applications developed using packaged products.

High level steps used in the “*Software packaged product estimation framework*” are as follows:

1. Obtain the historical data for various project lifecycle activities and support activities for software projects using packaged products.
2. Calculate the build effort using the complexity scale factors and effort calculation formula described in this section. We can then calculate the effort estimate for project lifecycle activities and support activities using the guidance values elaborated.
3. The overall effort is the sum total of build effort, lifecycle effort and support activity effort.

In the coming sections we will elaborate the calculation for each of these categories.

Pre-Requisites for Software Testing Estimation Framework

For the software packaged product estimation framework, we need the historical effort data for baselining. We need to get the historical data for previous projects that used the packaged products.

Build and Unit Test Effort Calculation

In a typical packaged product based solution, we can categorize the build effort into these four primary categories:

- *Presentation layer effort*: In this category we calculate the effort needed to develop all the presentation layer components such as page layouts, UI modules, navigation elements, portlets, widgets and such.
- *Configuration and customization effort*: This includes the effort to make the configuration-level changes for the product to fulfil the requirements. If we need to customize or extend any inbuilt product functionality, we will include that effort as well.
- *Business layer effort*: We include all the server side component development effort in this category.

8 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/software-estimation-framework-for-packaged-products/265444

Related Content

Framework for IT Service Value Engineering: Managing Value and IT Service Quality

Claus-Peter Praeg (2011). *Quality Management for IT Services: Perspectives on Business and Process Performance* (pp. 274-297).

www.irma-international.org/chapter/framework-service-value-engineering/46873

Estimating Two-Stage Network Technology Inefficiency: An Application to Cooperative Shinkin Banks in Japan

Hirofumi Fukuyama and William L. Weber (2012). *International Journal of Operations Research and Information Systems* (pp. 1-23).

www.irma-international.org/article/estimating-two-stage-network-technology/65591

A Virtual Learning Tool Design Using Lean Principles

(2018). *Lean Six Sigma for Optimal System Performance in Manufacturing and Service Organizations: Emerging Research and Opportunities* (pp. 66-81).

www.irma-international.org/chapter/a-virtual-learning-tool-design-using-lean-principles/197534

Business Process Modeling in the jABC: The One-Thing Approach

Tiziana Margaria and Bernhard Steffen (2009). *Handbook of Research on Business Process Modeling* (pp. 1-26).

www.irma-international.org/chapter/business-process-modeling-jabc/19685

Stackelberg Game Inventory Model With Progressive Permissible Delay of Payment Scheme

Gede Agus Widyadana, Nita H. Shah and Daniel Suriawidjaja Siek (2018). *Handbook of Research on Promoting Business Process Improvement Through Inventory Control Techniques* (pp. 194-214).

www.irma-international.org/chapter/stackelberg-game-inventory-model-with-progressive-permissible-delay-of-payment-scheme/198692