A Model for Software Development Cost Estimation with System Dynamic Approach

Mahbubeh Rivadeh
MSc in Information Technology Management; Faculty of Social Sciences and Economics; Alzahra University; Tehran, Iran;
Email: M.rivadeh@gmail.com

Ameneh Khadivar*
PhD in Systems Management; Associate Professor; Department of Management; Faculty of Social Sciences and Economics; Al-Zahra University; Tehran, Iran Email: a.khadivar@alzahra.ac.ir

Iranian Journal of Information Processing and Management

Iranian Research Institute for Information Science and Technology (IranDoc)
ISSN 2251-8223
eISSN 2251-8231
Indexed by SCOPUS, ISC, & LISTA
Vol. 34 | No. 3 | pp. 1343-1370
Spring 2019

Received: 14, Sep. 2017 **Accepted: 15, Sep. 2018**

Abstract: Software development is a complex and intricate process due to the number of factors involved, such as human factors, the complexity of developing products, the diversity of development levels, and the management problems of large projects. In Software development cost estimation, we are faced with many variables that change over time, interact with each other and make prediction difficult. In order to solve this problem, using systems and software dynamic approach, a model for estimation software development costs has been designed and implemented in this research.

In this research, the system dynamics approach and fuzzy logic are used to modeling. This research consists of two statistical populations. The first statistical population in this research includes IT managers and software development project managers. The second population in this research includes experts of Magfa Company. The tool used to collect data is a questionnaire. The data and information related to the project of development of BI-software (Business intelligence software) were at Magfa Company and has been gathered from the people involved in the project. After simulating and testing the model, three scenarios have been defined to reduce software development costs, which include: increasing personnel experience and increasing the experience of project managers, increasing the capabilities and competencies of manpower and changing the lifecycle of the system from cascading to agile methodology scenario. The findings indicate that the company will see a further reduction in software development costs by using the agile life cycle model.

Keywords: Software Cost, Software Development, System Dynamics, Fuzzy Logic

* Corresponding Author

ارائهٔ مدلی برای پیش‌بینی هزینه‌های توسعهٔ نرم‌افزار با رویکرد پویایی‌شناسی سیستم

محبوبه ریواده
کارشناسی ارشد؛ مدیریت فناوری اطلاعات؛
گرایش کسب‌وکار الکترونیک؛ دانشگاه الزهرا؛
m.rivadeh@gmail.com

آمنه خدیور
مدیریت سیستم‌ها؛ دانشیار؛
گروه مدیریت؛ دانشگاه الزهرا؛
پدیدآور رابط a.khadivar@alzahra.ac.ir

پژوهش‌نامهٔ پردازش و مدیریت اطلاعات

دریافت: ۱۳۹۶/۰۶/۲۳ | پذیرش: ۱۳۹۷/۰۶/۲۴ | مقاله برای اصلاح به مدت ۳۶ روز نزد پدیدآوران بوده است.

فصلنامه | علمی پژوهشی
پژوهشگاه علوم و فناوری اطلاعات ایران
(ایرانداک)
شاپا (چاپی) ۸۲۲۳-۲۲۵۱
شاپا (الکترونیکی) ۸۲۳۱-۲۲۵۱
نمایه در LISTA، ISC، و SCOPUS
jipm.irandoc.ac.ir
دورهٔ ۳۴ | شمارهٔ ۳ | ص‌ص ۱۳۴۳-۱۳۷۰
بهار ۱۳۹۸

چکیده: توسعهٔ نرم‌افزار به‌دلیل تعداد عوامل درگیر از جمله عوامل انسانی، پیچیدگی محصول در حال توسعه، تنوع سطوح توسعه و مشکلات مدیریتی پروژه‌های بزرگ فرایندی پیچیده است. در پیش‌بینی هزینهٔ توسعهٔ نرم‌افزار با تعداد زیادی متغیر مواجه هستیم که در طول زمان تغییر کرده و بر هم تأثیر متقابل دارند و عمل پیش‌بینی را مشکل می‌سازند. برای حل این مشکل، در این پژوهش مدلی برای پیش‌بینی هزینه‌های توسعهٔ نرم‌افزار با استفاده از رویکرد پویایی‌شناسی سیستم‌ها و نرم‌افزار «ونسیم» طراحی و ارائه گردیده است.

در این پژوهش از منطق فازی و رویکرد پویایی‌شناسی سیستم‌ها برای مدل‌سازی استفاده شده است. جامعهٔ آماری اول در این پژوهش، مدیران فناوری اطلاعات و مدیران پروژه‌های توسعهٔ نرم‌افزار و جامعهٔ آماری دوم، خبرگان شرکت «مگفا» است. ابزار مورد استفاده برای جمع‌آوری داده‌ها پرسشنامه است و داده‌ها و اطلاعات مربوط به پروژهٔ توسعهٔ نرم‌افزار هوش تجاری در شرکت «مگفا» بوده و اطلاعات آن نیز از افراد دخیل در پروژه جمع‌آوری شده است.

پس از شبیه‌سازی و تست مدل، سه سناریو برای کاهش هزینه‌های توسعهٔ نرم‌افزار تعریف گردید که عبارت‌اند از: افزایش تجربهٔ پرسنل و افزایش تجربهٔ مدیران پروژه، افزایش قابلیت‌ها و شایستگی‌های نیروی انسانی، و تغییر مدل چرخهٔ حیات سیستم از مدل آبشاری به متدولوژی چابک. یافته‌ها حاکی از آن است که شرکت مورد مطالعه با استفاده از مدل چرخهٔ حیات چابک، کاهش بیشتری در هزینه‌های توسعهٔ نرم‌افزار خود شاهد خواهد بود

کلیدواژه‌ها: پیش‌بینی هزینه، توسعهٔ نرم‌افزار، پویایی‌شناسی سیستم‌ها، منطق فازی

۱. مقدمه

از آنجا که توسعهٔ نرم‌افزار به یک سرمایه‌گذاری ضروری برای بسیاری از سازمان‌ها بدل شده است، تخمین نرم‌افزار در حال تبدیل به ضرورتی در حال رشد در مدیریت مؤثر پروژه‌های نرم‌افزاری است. منظور از تخمین نرم‌افزار عبارت است از تخمین هزینه، تخمین ریسک، تجزیه و تحلیل ریسک و ... است. تخمین دقیق هزینه می‌تواند در بهبود تجزیه و تحلیل امکان‌سنجی پروژه و مدیریت مؤثر فرایند توسعهٔ نرم‌افزار به سازمان کمک کند. بنابراین، ریسک به‌شدت کاهش می‌یابد (Huang et al. 2007). تخمین هزینه و تلاش، اهمیت فوق‌العاده‌ای برای موفقیت پروژه‌های توسعهٔ نرم‌افزار دارد. عملکرد روزانه نشان‌دهندهٔ این است که بسیاری از سازمان‌های نرم‌افزاری هزینه‌های غیرواقعی ارائه می‌دهند، با برنامه‌های فشرده کار می‌کنند، پروژه‌های خود را دیرتر از زمان‌بندی و بودجه به پایان رسانده و یا اصلاً موفق به اتمام آن‌ها نمی‌شوند (Trendowicz and Jeffery 2014).

مدل‌های بسیاری برای تخمین هزینهٔ توسعهٔ نرم‌افزار وجود دارد؛ اما به‌دلیل وجود عوامل مشترک تأثیرگذار بر روند توسعهٔ نرم‌افزار، تکامل مداوم محیط توسعه و فقدان سنجه‌ای که واقعاً نشان‌دهندهٔ پیچیدگی سیستم نرم‌افزاری باشد، امروزه تقریباً هیچ مدلی نمی‌تواند هزینهٔ نرم‌افزار را با درجهٔ بالایی از دقت تخمین بزند (Leung and Fan 2002).

تلاش‌های بسیاری در دهه‌های گذشته برای حل دشواری تخمین صحیح نرم‌افزار انجام گرفته، اما ثابت نشده است که هیچ رویکردی بتواند به‌طور مؤثر و مداوم معیارهای خروجی نرم‌افزار را تخمین بزند (Huang et al. 2007). مدل‌های امروزی برای تخمین هزینهٔ نرم‌افزار به توانایی خود برای نمایش پویایی‌های داخلی پروژه‌های نرم‌افزاری محدود هستند (Bohem 2001). در حالی که ابزارهای مدیریت پروژهٔ سنتی پشتیبانی مفیدی ارائه می‌دهند، به نظر می‌رسد که اغلب مدل‌های زیر ساختاری آن‌ها رابطهٔ کمی با پروژهٔ واقعی دارند. رویکردهای سنتی به این فرض متمایل هستند که هرگاه هر عنصر از پروژه فهمیده شود، کل پروژه ممکن است کنترل شود (Ahiaga-Dagbui et al. 2016). با این حال، تجارب نشان می‌دهد که روابط متقابل بین اجزای پروژه‌ها پیچیده‌تر از آن هستند که

توسط ساختارهای کاری سنتی پیشنهاد شوند (Marujo 2009).

چالش‌های اصلی روش‌های ذکرشده عبارت‌اند از: (Huang et al. 2007)

۱. وجود روابط بین معیارهای خروجی نرم‌افزار و عوامل مشارکتی که نشان‌دهندهٔ ویژگی‌های غیرخطی پیچیده هستند؛
۲. وجود معیارهای نرم‌افزار اغلب نامشخص و نامعلوم؛
۳. وجود مشکلات هنگام استفادهٔ همزمان از دانش خبره و داده‌های عددی پروژه در یک مدل.

دیدگاه دیگری توسط رویکرد پویایی‌شناسی سیستم‌ها ارائه می‌شود که بر کل پروژه متمرکز است. این رویکرد بر روابط متقابلی که ممکن است مسئول افراط‌ها و تخطی‌های غیرمنتظره باشند، تأکید می‌کند (Ramírez Cruz 2014). این دیدگاه یک رویکرد امیدوارکننده برای چالش‌های ذکرشده است؛ زیرا رویکرد پویایی‌شناسی سیستم‌ها، رویکردی برای درک رفتار غیرخطی سیستم‌ها در طول زمان با استفاده از ایده‌هایی مانند حلقه‌های بازخور، نرخ و حالت و تأخیرهای زمانی مؤثر بر رفتار کل سیستم است (Blackmore, Ison and Reynolds 2015).

به‌صورت مختصر و با توجه به مطالب و دلایل ذکرشده می‌توان گفت که در پیش‌بینی هزینه‌های توسعهٔ نرم‌افزار با متغیرهای بسیار زیادی مواجه هستیم که در طول زمان تغییر کرده، بر هم تأثیر متقابل دارند و عمل پیش‌بینی را مشکل می‌سازند. در این پژوهش برای حل مشکلات ذکرشده از روش پویایی‌شناسی سیستم‌ها که مناسب این نوع رفتارهای متقابل، غیرخطی و بازخوردی است، استفاده شده است.

۲. ادبیات تحقیق

۲-۱. تخمین هزینهٔ نرم‌افزار

واژهٔ تخمین[1] به معنای پیش‌بینی هوشمندانه از کمیت کاری که باید انجام شود و منابع (انسانی، پولی، تجهیزاتی و زمانی) مورد نیاز برای انجام کاری در زمان آینده در محیطی تعریف‌شده و با استفاده از روش‌های معین است (Chemuturi 2009). تخمین هزینهٔ نرم‌افزار فرایندی است که برای تخمین یا پیش‌بینی منابع، تلاش‌ها و هزینهٔ هر فرایند توسعه در

1. estimation

صنعت توسعهٔ نرم‌افزار استفاده می‌شود. علاوه بر موارد ذکرشده، تخمین هزینهٔ نرم‌افزار برای مدیریت، کنترل و نظارت فرایند در طی فرایند توسعهٔ نرم‌افزار نیز استفاده می‌شود. قبل از اختراع تکنیک‌های تخمین در ابتدای سال ۱۹۷۰، فرایند تخمین عمدتاً وابسته به تجربیات و برخی الگوریتم‌های ساده برای تخمین اندازه، تلاش‌ها و هزینه بوده است (Ziad et al. 2008).

۲-۲. روش‌های تخمین هزینهٔ نرم‌افزار

روش‌های مختلفی برای تخمین هزینهٔ نرم‌افزار وجود دارد. به‌طور اساسی روش‌های تخمین هزینهٔ نرم‌افزار به دو گروه روش‌های الگوریتمی[1] و روش‌های غیرالگوریتمی[2] تقسیم می‌شوند (Leung & Fan 2002; Kumari & Pushkar 2013; Anusha and Mukesh 2013). استفاده از هر دو گروه برای انجام تخمینی دقیق لازم است. هرچه نیازمندی‌ها و الزامات پروژه بهتر شناخته شوند، عملکرد مدل‌ها بهتر خواهد شد (Anusha and Mukesh 2013).

۲-۲-۱. روش‌های تخمین الگوریتمی

روش‌های الگوریتمی بر پایهٔ مدل‌های ریاضی هستند که تخمین هزینه را به‌صورت تابعی از تعدادی متغیر که به نظر می‌رسد عوامل اصلی تخمین هزینه باشند، ایجاد می‌کنند. برخی از مدل‌ها بر پایهٔ فرمول‌های ساده محاسباتی هستند و از روش‌های آماری ساده مانند میانگین و انحراف معیار استفاده می‌کنند. دیگر مدل‌های این گروه بر پایهٔ مدل‌های رگرسیونی و معادلات دیفرانسیلی هستند. هر مدل الگوریتمی فرمی به‌شکل f(x1, x2, …, xn) = تلاش[3] دارد که {x1, x2, …, xn} نشان‌دهندهٔ عوامل هزینه هستند (Sandhu et al. 2009).

مدل‌های الگوریتمی موجود از دو نظر انتخاب عوامل هزینه و شکل تابع F متفاوت‌اند (Sandhu et al. 2009).

انواع مدل‌های الگوریتمی از نظر شکل تابع

این مدل‌ها شامل مدل‌های خطی[4]، مدل‌های ضربی[5] و مدل‌های تابع توانی[6] هستند (Basar 2017).

---

1. algorithmic
2. non-algorithmic
3. effort
4. linear models
5. multiplicative models
6. power function

الف: مدل‌های خطی: در مدل‌های خطی، تخمین هزینه با فرمول $= a_0 + \sum_{i=1}^{n} a_i x_i$ تلاش (i=1,.....,n) محاسبه شده و ضرایب $a_1, \ldots, a_n$ به نحوی انتخاب می‌شوند که بهترین تناسب را با دادهٔ پروژهٔ تکمیل‌شده داشته باشند. در این فرمول $X_1, \ldots, X_n$ عوامل هزینه‌ای و $a_1, \ldots, a_n$ وزن‌های مهم مربوط به معیارها هستند.

ب: مدل‌های ضربی: در مدل‌های ضربی، تخمین هزینه با فرمول $= a_0 \prod_{i=1}^{n} a_i^{x_i}$ تلاش (i=1,.....,n) محاسبه می‌شود و مانند روش‌های خطی، ضرایب $a_1, \ldots, a_n$ به‌نحوی انتخاب می‌شوند که بهترین تناسب را با دادهٔ پروژهٔ تکمیل‌شده داشته باشند.

ج: مدل‌های تابع توانی: مدل‌های تابع توانی فرم کلی $= a \times S^b$ تلاش را دارند که در آن S اندازهٔ کد و a و b تابع سایر عوامل هزینه (معمولاً ساده) هستند. این دسته شامل دو مورد از پرطرفدارترین مدل‌های الگوریتمی مورد استفاده یعنی «کوکومو»[1]، مدل «اسلیم»[2] است.

۲-۲-۲. روش‌های تخمین غیرالگوریتمی

تخمین در مدل‌های غیرالگوریتمی با استفاده از تجارب پروژه‌های انجام‌شده در گذشته که مشابه پروژهٔ فعلی هستند، انجام می‌شود. در حقیقت، این گروه از مدل‌ها با استفاده از داده‌های گذشته که به نیازمندی‌های پروژهٔ ما شباهت داشته باشند، عمل تخمین را انجام می‌دهند (Zahra and Nazir 2012). از جمله روش‌های تخمین غیرالگوریتمی می‌توان به روش‌های هزینه‌یابی مبتنی بر قیاس، روش قضاوت خبره، قانون پارکینسون، روش قیمت‌گذاری برای بردن[3]، روش بالا به پایین، روش پایین به بالا و روش‌های یادگیری ماشین[4] اشاره کرد. از جمله روش‌های یادگیری ماشین که معمولاً برای اندازه‌گیری تلاش استفاده می‌شوند، می‌توان به دو روش شبکهٔ عصبی مصنوعی[5] و مدل فازی[6] اشاره کرد (Tailor, Saini & Rijwani 2014).

۲-۳. عوامل تأثیرگذار بر تلاش و هزینه‌های توسعهٔ نرم‌افزار

بر اساس ادبیات پژوهش عوامل مختلفی وجود دارد که بر هزینه‌های توسعهٔ نرم‌افزار اثرگذار هستند. دقت تخمین هزینه به این عوامل وابسته است. این عوامل باید حین انجام عمل تخمین هزینه توسط تخمین‌زننده در نظر گرفته شوند؛ زیرا امکان

1. COCOMO
2. The Putnam SLIM Model
3. pricing to win
4. machine learning
5. artificial intelligence
6. fuzzy method

دارد کـه تخمیـن نادرسـت هزینـه منجـر بـه خوش‌بینـی بیـش از حـد در مـورد تعهدهـا شـود و سـرریزهای اجتناب‌ناپذیـری در پـی داشـته باشـد (Potda, Ingle & Puri 2014). ادبیـات پژوهـش دسـته‌بندی‌های مختلفـی بـرای عوامـل مؤثـر بـر هزینه‌هـای توسـعهٔ نرم‌افـزار ارائـه کـرده اسـت، امـا بسـیاری از پژوهش‌هـا عوامـل مؤثـر بـر هزینه‌هـای توسـعهٔ نرم‌افـزار را بـه ۴ دسـتهٔ کلـی سـخت‌افزار، پرسـنل، پـروژه و محصـول تقسـیم می‌کننـد (Sharma & Kaushik 2016; Rani & Behel 2014; Madachy & Boehm 2008; Tailor, Saini & Rijwani 2014; Rijwani, Jain & Santani 2014). عوامـل تأثیرگـذار بـر هزینه‌هـای پـروژهٔ نرم‌افـزار در جـدول زیـر آمـده اسـت:

جدول ۱. جمع‌بندی و طبقه‌بندی عوامل مؤثر بر هزینه‌های توسعهٔ نرم‌افزار

	دسته‌بندی	عوامل	نوع متغیر	مراجع
۱	اندازه	اندازهٔ برنامهٔ نرم‌افزاری (تعداد خطوط کد)	درون‌زا	Sharma & Kaushik (2016)
۲	پیچیدگی نرم‌افزار	پیچیدگی معماری	درون‌زا	Trendowicz & Münch 2009; Trendowicz 2013; Trendowicz & Jeffery (2014)
۳		پیچیدگی رابط اتصال به دیگر سیستم‌ها	درون‌زا	Trendowicz & Münch 2009; Trendowicz & Jeffery (2014)
۴		پیچیدگی پایگاه داده	درون‌زا	Trendowicz & Münch (2009); Trendowicz & Jeffery (2014)
۵	ویژگی‌های پرسنل (شایستگی‌ها)	قابلیت تحلیلگر	درون‌زا	Sharma & Kaushik (2016); Madachy & Boehm (2008); Sharma, Bajpai & Litoriya (2012); Rijwani, Jain & Santani (2014); Tailor, Saini & Rijwani (2014); Kumari (2014); Rani & Behel (2014)
۶		تجربه و مهارت‌های مدیر پروژه	درون‌زا	Trendowicz & Münch (2009); Trendowicz (2013); Trendowicz & Jeffery (2014)
۷		قابلیت برنامه‌نویس	درون‌زا	Rani & Behel (2014); Madachy & Boehm (2008); Sharma, Bajpai & Litoriya (2012); Sharma & Kaushik (2016); Rijwani, Jain & Santani (2014); Tailor, Saini & Rijwani (2014)
۸		تجربهٔ پرسنل	درون‌زا	Trendowicz & Münch (2009); Trendowicz & Jeffery (2014)
۹		تعداد پرسنل	درون‌زا	Trendowicz & Münch (2009); Boehm (1981); Trendowicz (2013); Trendowicz & Jeffery (2014); Schweighofer et al. (2016)

	دسته‌بندی	عوامل	نوع متغیر	مراجع
۱۰	ویژگی‌های محصول	میزان مستندات مورد نیاز	درون‌زا	Rijwani, Jain & Santani (2014); Madachy & Boehm (2008)
۱۱		اندازهٔ پایگاه داده	درون‌زا	Tailor, Saini & Rijwani (2014); Rijwani, Jain & Santani (2014); Sharma & Kaushik (2016); Sharma, Bajpai & Litoriya (2012); Madachy & Boehm (2008); Rani & Behel (2014); Kumari 2014; Trendowicz & Münch 2009; Trendowicz (2013); Trendowicz & Jeffery (2014);
۱۲		قابلیت اطمینان نرم‌افزار	درون‌زا	Sharma & Kaushik (2016); Sharma, Bajpai & Litoriya (2012); Madachy & Boehm (2008); Rani & Behel (2014); Kumari (2014); Trendowicz (2013)
۱۳		قابلیت استفادهٔ مجدد	درون‌زا	Boehm (1981); Rijwani, Jain & Santani (2014); Madachy & Boehm (2008); Trendowicz & Münch (2009)
۱۴	ویژگی‌های پلتفرم و سخت‌افزار (نیازمندی‌ها)	نوسانات پلتفرم	درون‌زا	Rani & Behel (2014); Rijwani, Jain & Santani (2014); Madachy & Boehm (2008); Tailor, Saini & Rijwani (2014)
۱۵		محدودیت محل ذخیره‌سازی	درون‌زا	Rani & Behel (2014); Rijwani et al. (2014); Madachy & Boehm (2008) ; Tailor, Saini & Rijwani (2014)
۱۷	ویژگی‌های پروژه (محدودیت‌ها)	فشار زمان‌بندی	درون‌زا	Trendowicz & Münch (2009); Rijwani, Jain & Santani (2014); Trendowicz (2013); Trendowicz & Jeffery (2014)
۱۸		نیاز برای توسعهٔ غیرمتمرکز	درون‌زا	Trendowicz & Münch (2009); Madachy & Boehm (2008); Trendowicz (2013); Trendowicz & Jeffery (2014)
۱۹	عوامل زمینه‌ای	نوع زبان برنامه‌نویسی	درون‌زا	Trendowicz & Münch (2009); Trendowicz (2013); Schweighofer et al. (2016); Trendowicz & Jeffery (2014)
۲۰		مدل چرخهٔ حیات	درون‌زا	Trendowicz & Münch (2009); Trendowicz (2013); Schweighofer et al. (2016); Trendowicz & Jeffery (2014)
۲۱		نوع توسعه	درون‌زا	Trendowicz & Münch (2009); Trendowicz (2013); Schweighofer et al. (2016(; Trendowicz & Jeffery (2014)

دسته‌بندی	عوامل	نوع متغیر	مراجع
۲۲ عوامل محیطی	عوامل اقتصادی مانند ضریب افزایش حقوق سالانه	برون‌زا	Sharma, Bajpai & Litoriya (2012); Lagerstorm et al. (2012)
۲۳ عوامل مربوط به مشتری	دخالت مشتری	درون‌زا	Trendowicz (2013); Trendowicz & Jeffery (2014)
۲۴	بلوغ فرایند کسب‌وکار	درون‌زا	Madachy & Boehm (2008); Schweighofer et al. (2016); Trendowicz & Jeffery (2014)

۲-٤. رویکرد پویایی‌شناسی سیستم‌ها

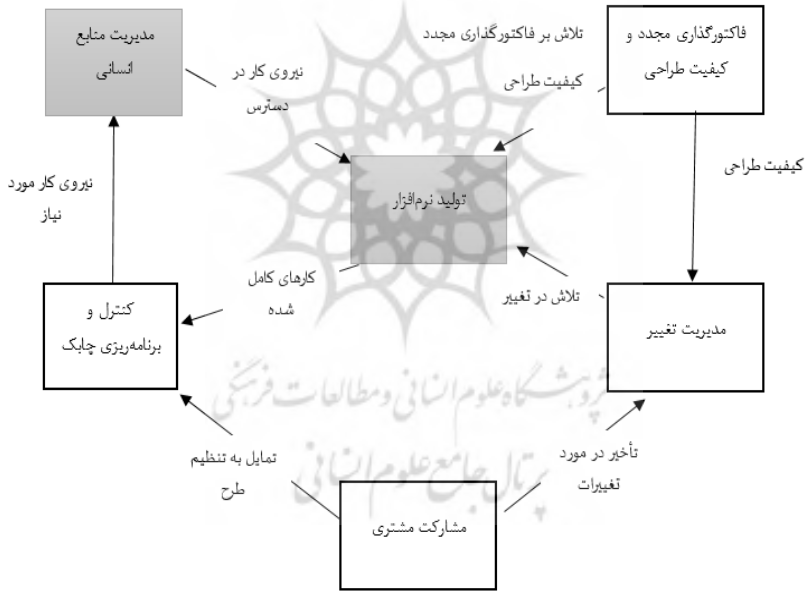
شاخه‌ای از تفکر سیستمی، پویایی‌شناسی سیستم‌ها نامیده می‌شود که در سطح کل سیستم عمل می‌کند؛ ضمن این‌که عمدتاً از توضیحات اضافه و بیهوده اجتناب می‌کند. این روش نظریه، روش‌ها و فلسفهٔ مورد نیاز برای تجزیه و تحلیل رفتار سیستم‌ها را ترکیب می‌کند. روش پویایی‌شناسی سیستم نه فقط در مدیریت، بلکه در دیگر زمینه‌ها مانند تغییر محیطی، سیاست، رفتار اقتصادی، پزشکی و مهندسی کاربرد دارد. پویایی‌شناسی سیستم‌ها، یک روش تفکر و متدولوژی شبیه‌سازی است که به‌طور خاص برای پشتیبانی از مطالعهٔ رفتار پویا در سیستم‌های پیچیده توسعه داده شده است. این متدولوژی توسط «فارستر»[1] توسعه‌یافته و در دهه‌های گذشته بهبود داده شده است. این روش ابتدا در صنعت و مدیریت سیستم کسب‌وکار به‌کار رفته است (Hjorth & Bagheri 2006). رویکرد پویایی‌شناسی سیستم‌ها بر ساختار و رفتارهای سیستم‌هایی که ترکیبی از حلقه‌های بازخوردی و در تعامل با یکدیگر هستند، متمرکز است. هدف این رویکرد، تجسم تعاملات پویا از متغیرهای مختلف سیستم و تجزیه و تحلیل اثر آن‌ها بر سیاست‌های تصمیم‌گیری در افق زمانی بلندمدت است (Sachan, Sahay & Sharma 2005).

۳. پیشینهٔ تحقیق

«کو، رامش و عبدالحمید» به مدل‌سازی پویای توسعهٔ نرم‌افزار چابک پرداخته‌اند. در این پژوهش یک مدل شبیه‌سازی پویایی‌شناسی سیستم توسعه داده شده است که وابستگی‌های پیچیده بین تلاش‌های مختلف استفاده‌شده در توسعهٔ چابک را در نظر

1. Forrester

می‌گیرد. این مدل بر اساس مرور وسیع ادبیات و همچنین جمع‌آوری داده‌های کمّی و کیفی از پروژه‌های واقعی ۹ سازمان توسعه داده شده است. ساختار مدل بر روش‌های اصلی چابک تمرکز دارد. این ۹ سازمان از برنامه‌نویسی «ایکس پی»[1] یا «اسکرام»[2] و یا ترکیبی از این دو مدل استفاده می‌کنند. مدل سیستم پویای توسعه داده‌شده در این پژوهش، فعالیت‌های ضروری در توسعهٔ چابک مانند برنامه‌ریزی چابک، تکرار کوتاه، مشارکت مشتری، فاکتورگذاری مجدد، آزمایش مجدد، آزمایش واحد و برنامه‌نویسی جفتی را ادغام کرده است. این فعالیت‌ها توسط ۴ زیر مدل (شکل ۲) مدل‌سازی شده‌اند: مدیریت تغییر، مشارکت مشتری، کنترل و برنامه‌ریزی چابک، فاکتورگذاری مجدد و کیفیت طراحی. زیرسیستم‌های مدیریت منابع انسانی و تولید نرم‌افزار که برای پشتیبانی توسعهٔ چابک ضروری هستند با اتخاذ از کار Abdel-Hamid and Madnick (1991) ساخته شده‌اند.

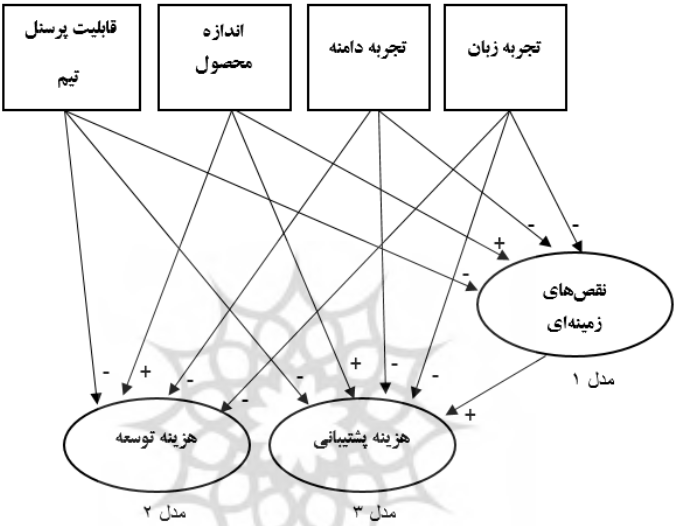


شکل ۱. ساختار مدل در پژوهش Cao, Ramesh and Abdel-Hamid (2010)

«کریشنان» نقش عوامل تیمی در کیفیت و هزینهٔ نرم‌افزار را در مقالهٔ خود بررسی کرده است. این مقاله تأثیر عوامل مختلف مربوط به تیم در توسعهٔ نرم‌افزار را بررسی

1. eXtreme (XP) 2. SCRUM

می‌کند. عواملی مانند تجربهٔ دامنه و تجربهٔ زبان اعضای تیم و قابلیت پرسنل تیم بر روی هزینه‌ها و کیفیت محصولات نرم‌افزاری تولیدشده در این مقاله مورد بررسی قرار گرفته‌اند. کیفیت هدف در این مقاله بر پایهٔ تعداد مشکلات زمینه‌ای گزارش‌شده توسط مشتریان اندازه‌گیری می‌شود. اثر سه مقیاس مرتبط با تیم یعنی قابلیت پرسنل تیم، تجربهٔ دامنهٔ تیم و تجربهٔ زبان تیم در مدل مفهومی آورده شده در شکل ۲، نشان داده شده است.



شکل ۲. مدل مفهومی پژوهش (Krishnan 1998)

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی

٤. روش پژوهش

روش مورد استفاده در مطالعهٔ حاضر، رویکرد پویایی‌شناسی سیستم‌هاست که در بخش ادبیات به معرفی آن پرداخته شد. از دید Sterman (2000) همهٔ مدل‌سازان موفق از یک فرایند منظم پیروی می‌کنند که شامل پنج گام ۱) چارچوب‌بندی مسئله (انتخاب مرز)، ۲) فرموله‌کردن فرضیهٔ دینامیکی، ۳) فرموله‌کردن یک مدل شبیه‌سازی، ۴) آزمودن، و ۵) طراحی و ارزیابی سیاست است. در ادامهٔ به شرح هر گام در پژوهش حاضر می‌پردازیم.

٤-۱. تعریف دقیق مسئله و شناسایی مرزهای مدل

مسئلهٔ این پژوهش تخمین هزینه‌های توسعهٔ نرم‌افزار و شبیه‌سازی آن با روش پویایی‌شناسی سیستم‌هاست. برای این کار، ابتدا لازم است عوامل و پویایی‌هایی که بر

هزینه‌های توسعهٔ نرم‌افزار اثر دارند، شناسایی شوند. مرزهای مدل این پژوهش شامل عوامل اثرگذار بر هزینهٔ توسعهٔ نرم‌افزار در یک سازمان توسعه‌دهندهٔ نرم‌افزار است. دیدگاه ما در انتخاب عوامل و شناسایی مرزها، دیدگاه بنگاه‌محور بوده و سایر عوامل غیربنگاهی مانند فعالیت رقبا، کشش، روندهای بازار و ... در نظر گرفته نشده است. با بررسی ادبیات تحقیق و با توجه به نظر خبرگان، عوامل ذکرشده در جدول ۱، به‌عنوان متغیرهای مؤثر بر هزینه‌های توسعهٔ نرم‌افزار شناسایی و دسته‌بندی شده و نوع آن‌ها از نظر درون‌زا یا برون‌زا بودن نیز در جدول ۱، مشخص گردیده است.

٤-۲. توضیح مدل

مدل ارائه‌شده در این پژوهش از چهار زیرسیستم نیروی انسانی، پروژه، سخت‌افزار و نرم‌افزار تشکیل شده است. در ادامه، هر زیرسیستم به‌طور جداگانه بررسی می‌گردد.

٤-۲-۱. زیرسیستم نیروی انسانی

زیرسیستم نیروی انسانی شامل تمامی متغیرهای مرتبط با پرسنل است که در هزینه‌های توسعهٔ نرم‌افزار دخیل هستند. این متغیرها شامل تلاش و عوامل مؤثر بر آن، شایستگی پرسنل و عوامل مؤثر بر آن، تعداد پرسنل و روابط بین آن‌هاست که شرح هر یک از آن‌ها در ادامه آورده شده است. این بخش از بخش‌های مهم تأثیرگذار بر هزینه‌های توسعهٔ نرم‌افزار محسوب می‌شود، زیرا از آنجا که هزینهٔ کارکنان معمولاً قسمت اعظم هزینهٔ پروژه را تشکیل می‌دهد، اصطلاح تخمین هزینه و تخمین تلاش در ادبیات پژوهش به‌صورت متقابل مورد استفاده قرار می‌گیرند (شاه‌محمدی ۱۳۷۹). تلاش لازم برای توسعهٔ نرم‌افزار بستگی به متغیرهای مختلف دارد. این متغیرها شامل دسته‌بندی‌های اندازه، پیچیدگی نرم‌افزار، ویژگی‌های محصول، متغیرهای زمینه‌ای، متغیرهای مربوط به مشتری و شایستگی‌های پرسنل است. بنابراین، عوامل مؤثر بر تلاش شامل متغیرهای اندازهٔ نرم‌افزار، پیچیدگی معماری، پیچیدگی رابط اتصال به سایر سیستم‌ها، پیچیدگی پایگاه داده، نوع توسعه، نوع زبان برنامه‌نویسی، مدل چرخهٔ حیات، اندازهٔ پایگاه داده، میزان مستندات مورد نیاز، قابلیت اطمینان نرم‌افزار، قابلیت استفادهٔ مجدد، میزان دخالت مشتری، بلوغ فرایند کسب‌وکار و شایستگی‌های پرسنل است. با افزایش همهٔ متغیرهای ذکرشده، به‌غیر از متغیر بلوغ فرایند کسب‌وکار و متغیر شایستگی‌های پرسنل، میزان تلاش لازم برای توسعهٔ محصول نرم‌افزاری افزایش می‌یابد. توانایی‌های تحلیلگر، قابلیت برنامه‌نویس،

تجربه و مهارت‌های مدیر پروژه و تجربهٔ پرسنل متغیرهایی هستند که بر شایستگی‌های پرسنل اثر دارند. با افزایش هر یک از این متغیرها، مقدار متغیر شایستگی‌های پرسنل افزایش خواهد یافت.

٤-٢- ٢. زیرسیستم پروژه

این زیرسیستم شامل متغیرهایی است که یا بر کیفیت مدیریت پروژه اثر می‌گذارند و یا محدودیت‌هایی در توسعه و شرایط کاری ایجاد می‌کنند. در حقیقت، این زیرسیستم شامل متغیرهای فشار زمان‌بندی و نیاز برای توسعهٔ غیرمتمرکز است که متغیرهای محدودکنندهٔ پروژه هستند. متغیرهای محدودکنندهٔ پروژه متغیرهایی هستند که بر امکان‌پذیربودن پروژه اثر گذاشته و از این طریق، منجر به محدود شدن پروژه می‌شوند. با افزایش فشار زمان‌بندی و اگر بخواهیم پروژه را در زمان مقرر تحویل دهیم، باید برخی قابلیت‌ها و کارکردهای پروژه را محدود کرده و در غیر این صورت، باید زمان انجام پروژه را بیشتر کنیم. افزایش محدودیت‌های پروژه منجر به افزایش در هزینه‌های پشتیبانی خواهد شد.

٤-٢- ٣. زیرسیستم سخت‌افزار

این زیرسیستم شامل متغیرهای پلتفرمی و سخت‌افزاری است که بر هزینه‌های توسعهٔ نرم‌افزار مؤثر هستند. افزایش در متغیرهای نوسانات پلتفرم، محدودیت محل ذخیره‌سازی و قیمت تکنولوژی‌های سخت‌افزاری مورد نیاز، منجر به افزایش نیازمندی‌های کلی سخت‌افزاری و پلتفرمی برای انجام عمل توسعه می‌شود. افزایش نیازمندی‌های سخت‌افزاری و پلتفرمی منجر به افزایش در هزینه‌های خرید و نگهداری سخت‌افزار خواهد شد.

٤-٢- ٤. زیرسیستم نرم‌افزار

با افزایش متغیرهای تعداد نرم‌افزار مورد نیاز و متوسط قیمت نرم‌افزار، هزینه‌های لازم برای خرید و نگهداری نرم‌افزار افزایش می‌یابد.

علاوه بر زیرسیستم‌های ذکرشده، مدل این پژوهش شامل یک متغیر محیطی تأثیرگذار بر هزینه‌های توسعهٔ نرم‌افزار یعنی متغیر درصد افزایش حقوق سالانهٔ کارمندان پروژه است. این متغیر یک متغیر برون‌زاست، زیرا بر سیستم اثر می‌گذارد، اما از آن اثر نمی‌پذیرد.

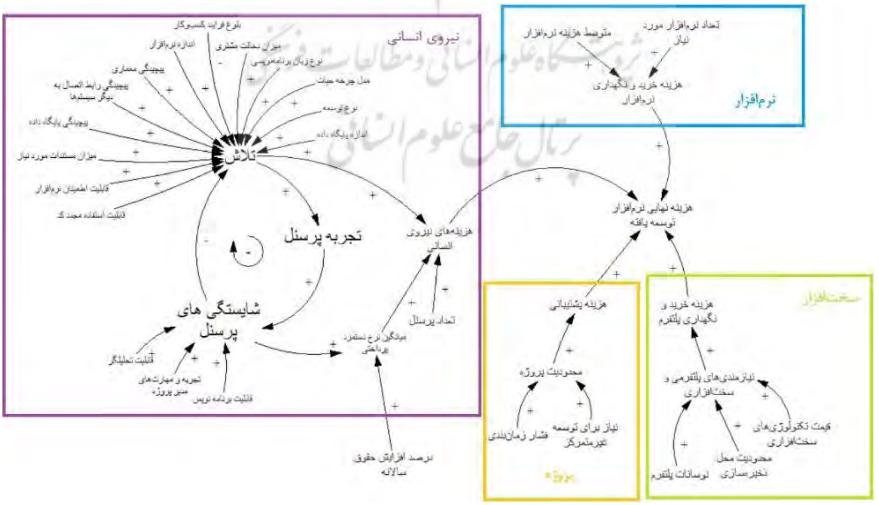
٤- ٣. فرضیات پویا و نمودار علت-معلولی

در تخمین هزینه‌های توسعهٔ نرم‌افزار مهم‌ترین متغیرهای درون‌زا عبارت‌اند از متغیرهای تأثیرگذار بر هزینه‌های توسعهٔ نرم‌افزار که در بخش‌های قبل در جدول و توضیح آن‌ها آورده شده است. از آنجا که تمامی این متغیرهای مؤثر بر هزینه در طول زمان تغییر می‌کنند، رفتار این متغیرها در طول دوره‌های مختلف از پویایی برخوردار است. از طرف دیگر، نحوهٔ تأثیر و تأثر متغیرها بر یکدیگر در شکل ۳، آورده شده است. بر اساس مرور ادبیات فرضیات پذیرفته‌شده و نمودار علّت-معلولی در ادامه آورده خواهد شد.

٤-٣-١. حلقهٔ تأثیر متقابل تلاش و تجربه

حلقهٔ تأثیر متقابل تلاش و تجربه یک حلقهٔ منفی است، زیرا تعداد روابط علّت-معلولی منفی در آن فرد است. حلقه‌های منفی حلقه‌هایی خودکنترلی هستند؛ به این معنا که با حرکت یک متغیر در جهتی خاص، حلقه در جهت مخالف آن حرکت خواهد کرد. این حلقه‌های منفی حلقه‌هایی تعادلی هستند که رفتاری هدف‌جو ایجاد می‌کنند. در این حلقه با افزایش تلاش، تجربهٔ پرسنل افزایش یافته و با افزایش تجربهٔ پرسنل، متغیر شایستگی‌های پرسنل افزایش خواهد یافت. با افزایش شایستگی‌های پرسنل، تلاش لازم برای توسعه کاهش می‌یابد.

٤-٣-٢. نمودار علّت معلولی



شکل ۳. نمودار علت معلولی به همراه زیرسیستم‌های مدل

در نمودار علت-معلولی شکل ۴، موارد زیر قابل مشاهده است:

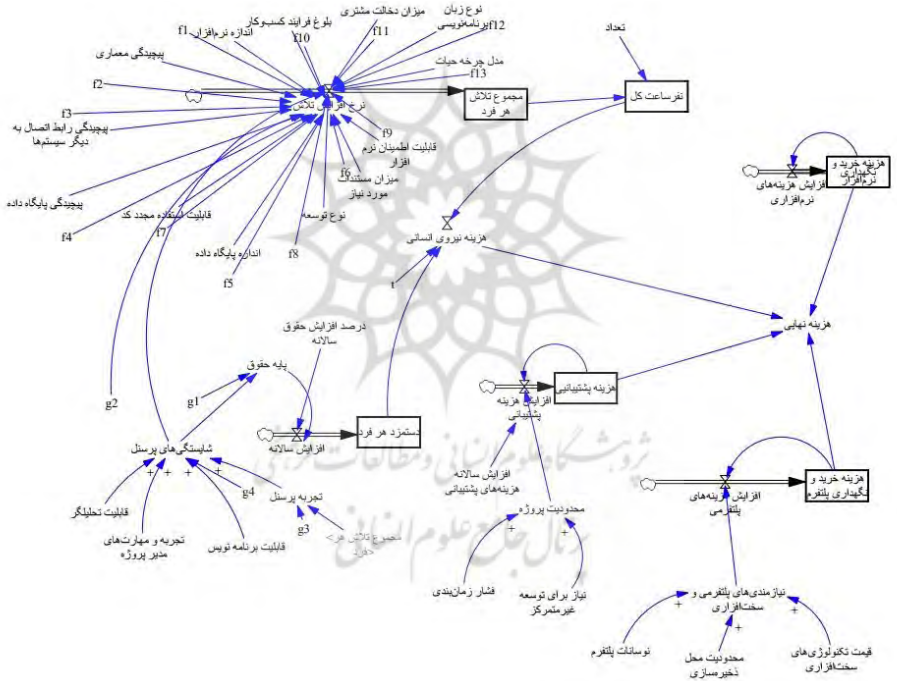
- با افزایش هر یک از متغیرهای میزان دخالت مشتری، نوع زبان برنامه‌نویسی، مدل چرخهٔ حیات، نوع توسعه، اندازهٔ پایگاه داده، قابلیت استفادهٔ مجدد، قابلیت اطمینان، میزان مستندات مورد نیاز، پیچیدگی پایگاه داده، پیچیدگی رابط اتصال به دیگر سیستم‌ها، پیچیدگی معماری و اندازهٔ نرم‌افزار تلاش مورد نیاز برای توسعهٔ محصول نرم‌افزاری افزایش خواهد یافت. این افزایشِ تلاش منجر به افزایش هزینه‌های نیروی انسانی و در نهایت، منجر به افزایش هزینه‌های نهایی توسعهٔ نرم‌افزار می‌شود.
- با افزایش متغیر بلوغ فرایند کسب‌وکار، میزان تلاش لازم برای توسعهٔ نرم‌افزار کاهش می‌یابد. در نتیجهٔ این کاهش تلاش، هزینه‌های نیروی انسانی نیز کاهش می‌یابد. این امر در نهایت، منجر به کاهش هزینهٔ نهایی توسعهٔ نرم‌افزار خواهد شد.
- با افزایش متغیرهای مؤثر بر شایستگی‌های پرسنل (قابلیت تحلیلگر، قابلیت برنامه‌نویس، تجربه و مهارت‌های مدیر پروژه، تجربهٔ پرسنل)، میزان شایستگی افزایش می‌یابد. این افزایش شایستگی موجب افزایش میانگین نرخ حقوق پرداختی و در نتیجه، افزایش هزینه‌های نیروی انسانی و در نهایت، منجر به افزایش هزینه‌های نهایی توسعهٔ نرم‌افزار می‌شود. از طرف دیگر، با افزایش متغیرهای مؤثر بر شایستگی، متغیر تلاش کاهش یافته و این کاهشِ تلاش منجر به کاهش هزینه‌های نیروی انسانی و در نتیجه، کاهش هزینه‌های نهایی خواهد شد.
- با افزایش درصد افزایش حقوق سالانه، میانگین حقوق پرداختی به پرسنل افزایش خواهد یافت. این افزایش در میانگین حقوق پرداختی منجر به افزایش هزینه‌های نیروی انسانی و در نهایت، منجر به افزایشِ هزینه‌های نهایی توسعهٔ نرم‌افزار خواهد شد.
- با افزایش در تعداد پرسنل درگیر در فرایند توسعهٔ نرم‌افزار، هزینه‌های نیروی انسانی و در نهایت، هزینه‌های نهایی توسعهٔ نرم‌افزار افزایش می‌یابد.
- با افزایش فشار زمان‌بندی و افزایش نیاز به توسعهٔ غیرمتمرکز، محدودیت‌های پروژه افزایش یافته و منجر به افزایش هزینه‌های پشتیبانی خواهد شد. این افزایش در هزینه‌های پشتیبانی منجر به افزایش هزینه‌های نهایی توسعهٔ نرم‌افزار می‌شود.
- با افزایش متغیرهای نوسانات پلتفرم، محدودیت محل ذخیره‌سازی، قیمت تکنولوژی‌های سخت‌افزاری، نیازمندی‌های سخت‌افزاری مورد نیاز در فرایند توسعهٔ

نرم‌افـزار افزایـش یافتـه و منجـر بـه افزایـش هزینه‌هـای سـخت‌افزاری می‌شـود. ایـن افزایـش در هزینه‌هـای سـخت‌افزاری منجـر بـه افزایـش هزینه‌هـای نهایـی توسـعهٔ نرم‌افـزار خواهـد شـد.

- بـا افزایـش متغیرهـای هزینه‌هـای نرم‌افـزاری و تعـداد نرم‌افـزار مـورد نیـاز هزینـهٔ خریـد و نگهـداری نرم‌افـزار افزایـش خواهـد یافـت. افزایـش هزینـهٔ خریـد و نگهـداری نرم‌افـزار منجـر بـه افزایـش هزینه‌هـای نهایـی توسـعهٔ نرم‌افـزار خواهـد شـد.

٤-٤. نمودار نرخ و حالت

نمودار نرخ و حالت در تصویر ۴، آورده شده است.



شکل ٤. نمودار نرخ و حالت

در نمـودار نـرخ و حالـت از ۱۷ متغیـر کمکـی (f1 الـی f13 و g1 الـی g4) اسـتفاده شـده اسـت. نحـوهٔ تأثیـر هـر کـدام از ایـن متغیرهـا بـا اسـتفاده از پرسشـنامه صـورت گرفتـه اسـت کـه به‌وسـیلهٔ

آن نقاط مختلف توابع لوک‌آپ[1] از خبرگان اخذ شده است. هر یک از متغیرهای مدل، دارای وزنی است که به‌صورت زیر به‌دست آمده است:

پس از تهیهٔ عوامل مؤثر بر هزینه‌های توسعهٔ نرم‌افزار که در جدول ۱ آورده شده، پرسشنامه‌ای برای گردآوردی نظرات خبرگان استفاده شده است و از خبرگان خواسته شده تا میزان تأثیر هر یک از عوامل را بر هزینه‌های توسعهٔ نرم‌افزار به‌صورت طیف لیکرت از «بسیار کم» تا «بسیار زیاد» امتیاز بدهند و در صورت لزوم، مواردی را نیز به فهرست عرضه‌شده اضافه نمایند و در صورت مخالفت با دسته‌بندی عوامل، توضیحاتی ارائه دهند. از آنجا که در پرسشنامهٔ مذکور هر خبره باید سطحی را برای هر کدام از عوامل تأثیرگذار در نظر بگیرد، این سطوح به‌صورت متغیرهای زبانی) بسیار کم، کم، متوسط، زیاد، بسیار زیاد (در اختیار خبرگان قرار داده شده است. همچنین، با توجه به این‌که اکثر عوامل ذکرشده کیفی هستند و باید برای استفاده در مدل‌سازی به سری صدتایی تبدیل شوند، بنابراین، مقادیر عددی متغیرهای زبانی را در قالب اعداد فازی، به‌صورت اعداد فازی مثلثی بسیار کم (۲۵ و ۰ و ۰)، کم (۵۰ و ۲۵ و ۰)، متوسط (۷۵ و ۵۰ و ۲۵)، زیاد (۱۰۰ و ۷۵ و ۵۰)، بسیار زیاد (۱۰۰ و ۱۰۰ و ۷۵) تعریف کرده‌ایم. پس از جمع‌آوری پاسخ‌های پرسشنامه، میانگین فازی هر پرسشنامه بر طبق فرمول‌های مربوطه محاسبه گردیده است.

٤-٥. آزمون مدل ارائه‌شده

در این بخش به‌عنوان اعتبارسنجی مدل، برخی از مهم‌ترین آزمون‌هایی که روی مدل انجام شده، ارائه می‌شود.

تست کفایت مرزهای مدل: این آزمون بررسی می‌کند که «آیا ساختار مدل برای هدف مدل مناسب است یا خیر؟ آیا تجمیع مدل مناسب است و آیا شامل تمام ساختارهای مرتبط حاوی متغیرها و بازخوردهای ضروری برای بیان مشکل است و آیا با هدف پژوهش تطبیق دارد؟ آیا تغییر مرزهای مدل می‌تواند سیاست‌های پیشنهادی مدل را تغییر دهد؟». در مدل طراحی‌شده تا حد امکان سعی شده است که متغیرها درون‌زا باشند و مهم‌ترین متغیر برون‌زا متغیر درصد افزایش حقوق سالیانه است که تغییرات آن در اختیار مدل‌ساز نیست و باید برون‌زا در نظر گرفته شود.

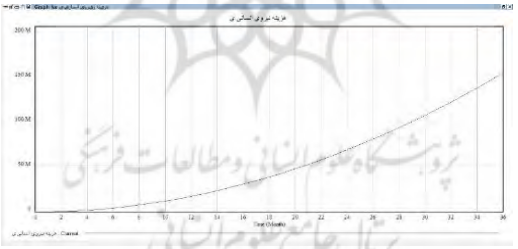
1. lookup

تست ساختار: این آزمون پاسخ به این سؤال است که «آیا ساختار مدل با علمی که ما از ساختار سیستم واقعی آن داشتیم، در تضاد نیست؟ و آیا مرتبط‌ترین ساختار با سیستم واقعی مدل شده است؟».

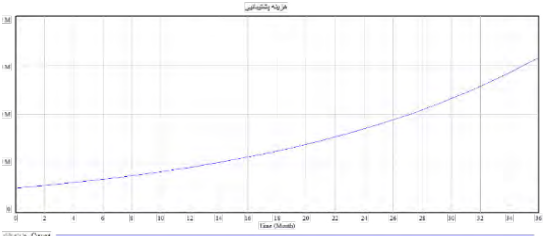
در این بخش بررسی تطابق مدل با واقعیت در مراحل تصمیم‌گیری و قوانین موجود فیزیکی بررسی شده است. رفتار متغیرهای تعریف‌شده برای تصمیم‌گیری مدیران و تأثیر آن‌ها روی رفتار مدل کاملاً در مراحل بحرانی با واقعیت تطبیق داشته و این موضوع از رفتار مدل در این مواقع و همین‌طور با نظرسنجی از خبرگان شرکت «مگفا» که در پروژهٔ مورد مطالعه فعال هستند، به تأیید رسیده است.

٤-٦. شبیه‌سازی

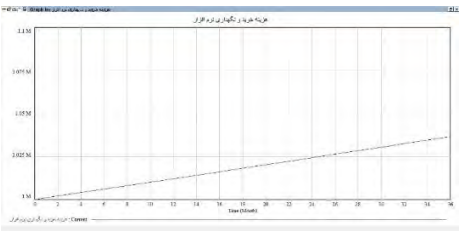
پس از انجام آزمون‌های لازم و اطمینان از اعتبار مدل، در این مرحله شبیه‌سازی مدل‌ها انجام می‌شود. از آنجا که طول مدت پروژه در شرکت پیش‌بینی می‌شود سه سال باشد، شبیه‌سازی برای یک دورهٔ سه‌ساله، معادل ۳۶ ماه انجام شده و ماه، واحد زمانی شبیه‌سازی است. نتایج شبیه‌سازی برای متغیرهای مختلف مدل در نمودارهای شکل ۵، تا شکل ۸، در ادامه ارائه شده است.



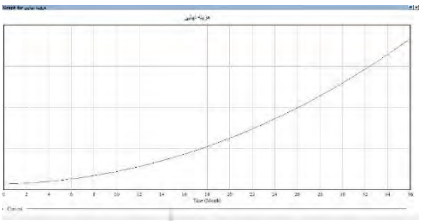
شکل ۵. رفتار متغیر هزینهٔ نیروی انسانی



شکل ٦. رفتار متغیر هزینهٔ پشتیبانی



شکل ۷. رفتار متغیر هزینهٔ خرید و نگهداری نرم‌افزار

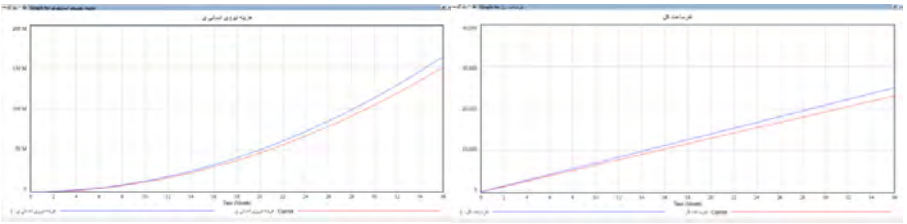


شکل ۸. رفتار متغیر هزینهٔ نهایی

٤-۷. آنالیز حساسیت

در این بخش برای اطمینان از درست بودن مدل و مطابقت رفتار آن با حالت واقعی، آنالیز حساسیت انجام خواهد شد و پس از آن به تشریح سناریوهایی خواهیم پرداخت که موجب بهبود سیستم می‌شوند. به سه شکل می‌توان آنالیز حساسیت را انجام داد: آنالیز حساسیت عددی، که بر اثر تغییر در برخی مفروضات اعداد متغیرهایی که به‌عنوان هدف و نتیجهٔ آنالیز تعیین شده‌اند، تغییر می‌کند؛ آنالیز حساسیت رفتار، که در اثر تغییر در مفروضات الگوهای رفتاری متغیرهای هدف تغییر می‌کند؛ به‌عنوان مثال، الگوی رفتاری به‌شکل حرف S به الگوی رفتاری جهش و فروپاشی[1] تبدیل می‌شود؛ و آنالیز حساسیت تغییر سیاست، که در این حالت با تغییر در برخی مفروضات و اعمال برخی سیاست‌ها حالت‌های مطلوب در سیستم به‌دست می‌آید. در این پژوهش، ابتدا برخی از متغیرهای اصلی و مهم را در بخش آنالیز حساسیت مقداری و رفتاری مورد بررسی قرار خواهیم داد.

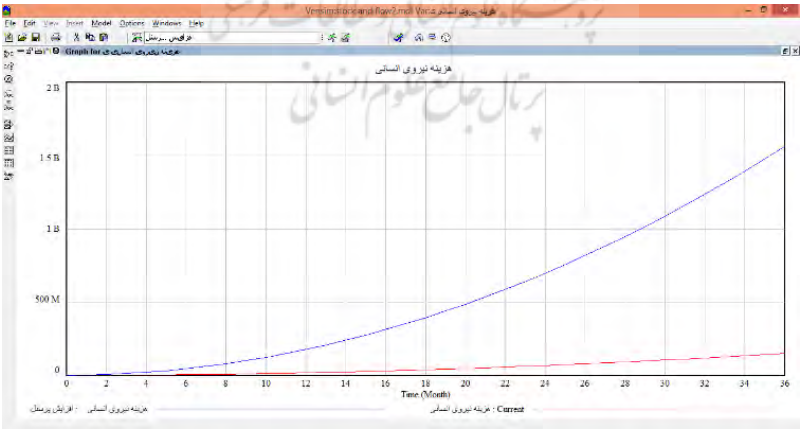
1. overshoot & collapse



شکل ۹. نتایج تحلیل حساسیت ۱

تغییر در متغیر اندازهٔ پایگاه داده: با توجه به مفروضات و مدل پژوهش می‌دانیم که تأثیر متغیر «اندازهٔ پایگاه داده» بر تلاش و هزینهٔ نیروی انسانی و نفرساعت کل مثبت است. بنابراین، انتظار داریم که هزینهٔ این متغیرهای اصلی با افزایش مقدار متغیر «اندازهٔ پایگاه داده» افزایش یابد. نتیجهٔ تحلیل حساسیت در مورد متغیر هزینه‌های نیروی انسانی و نفرساعت کل در شکل ۹ قابل مشاهده است. نتایج نشان‌دهندهٔ حالت افزایشی و تأییدکنندهٔ حساسیت مدل نسبت به تغییر اندازهٔ پایگاه داده است.

تغییر در تعداد نیروی انسانی: در یک آزمون تحلیل حساسیت دیگر، تعداد نیروی انسانی پروژه از ۱۰ نفر به ۱۰۰ نفر افزایش یافته است که نتیجهٔ این افزایش در رفتار متغیر هزینه‌های نیروی انسانی در شکل ۱۰، قابل ملاحظه است. با توجه به مفروضات و مدل پژوهش می‌دانیم که به‌صورت منطقی، با افزایش تعداد نیروی انسانی درگیر در پروژه، نمودار هزینهٔ نیروی انسانی باید افزایشی باشد. در شکل ۱۰، مشاهده می‌شود که طبق انتظارات، نمودار هزینهٔ نیروی انسانی روند افزایشی داشته و شیب آن از نمودار فعلی بالاتر است.

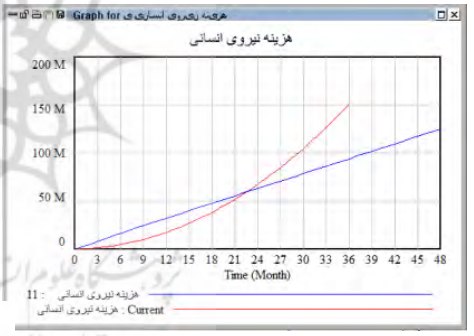


شکل ۱۰. رفتار متغیر نیروی انسانی برای تحلیل حساسیت ۲

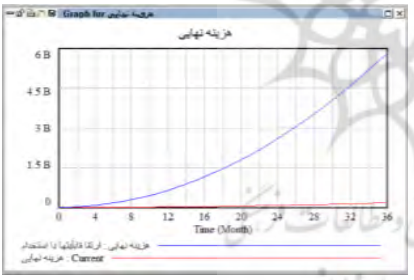
٤-٨. سناریوسازی

سناریوی ۱: افزایش تجربهٔ پرسنل و افزایش تجربهٔ مدیران پروژه

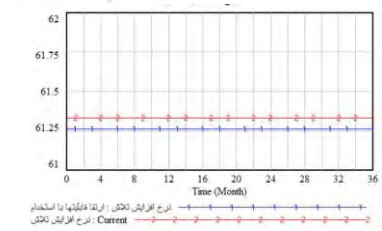
در این سناریو می‌خواهیم بدانیم تأثیر افزایش تجربهٔ پرسنل که شامل همهٔ نیروهای فعال در پروژه و مدیران پروژه هستند، چگونه شبیه‌سازی خواهد شد. افزایش تجربه از یک طرف تأثیری مستقیم بر افزایش دستمزد و در نهایت، هزینه‌های نیروی انسانی و هزینه‌های کل دارد و از طرف دیگر، از طریق یک ساختار بازخوردی منفی، منجر به کاهش تلاش مورد نیاز برای اجرای پروژه خواهد شد. لذا، پیش‌بینی رفتار نهایی هزینه به‌راحتی امکان‌پذیر نیست. لازم به ذکر است که برای مشخص‌شدن نتیجه، لازم بود که مدت اجرا از ۳ سال به ۴ سال افزایش یابد. با توجه به نتایج و نمودارها در بازهٔ زمانی ۴ ساله مشاهده می‌شود که با وجود روند افزایشی هزینه‌ها، نمودار هزینه‌های نیروی انسانی شیب کمتری نسبت به قبل از خود نشان داده است. تأثیر اجرای این سناریو در مقایسه با وضع موجود در شکل‌های ۱۱ و ۱۲ نشان داده شده است:



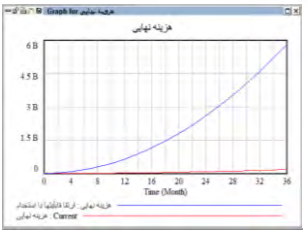
شکل ۱۱. تأثیر سناریوی ۱ بر هزینهٔ نهایی



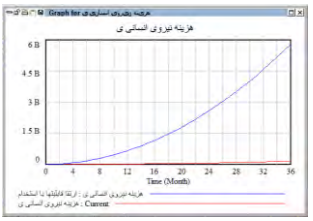
شکل ۱۲. تأثیر سناریوی ۱ بر هزینهٔ نیروی انسانی



شکل ۱۳. رفتار نرخ افزایش تلاش در سناریوی ۲



شکل ١٤. رفتار هزینهٔ نهایی در سناریوی ۲



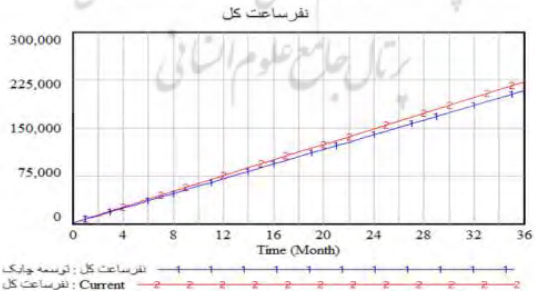
شکل ۱۵. رفتار هزینه نیروی انسانی در سناریوی ۲

سناریوی ۲: افزایش قابلیت‌ها و شایستگی‌های نیروی انسانی

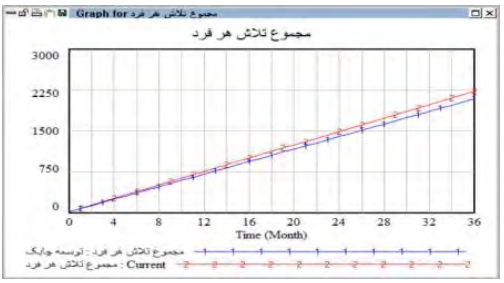
هدف از این سناریو، افزایش قابلیت‌ها و شایستگی‌های نیروی انسانی با تمرکز بر استخدام نیروی انسانی شایسته‌تر است. بنابراین، در این سناریو متغیرهای توانایی‌های تحلیلگر، قابلیت برنامه‌نویس و تجربه و مهارت‌های مدیر پروژه تغییر کرده‌اند. نتایج حاصل از این سناریو بر رفتار متغیرهای اصلی در شکل‌های ۱۳ الی ۱۵ مشاهده می‌شود.

سناریوی ۳: تغییر مدل چرخهٔ حیات سیستم از آبشاری به متدولوژی چابک

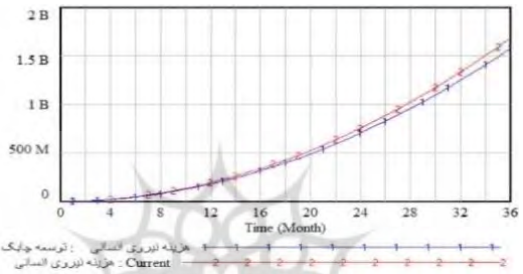
در این سناریو با فرض ثابت‌بودن نیروی انسانی در دسترس از نظر کیفیت و کمیت، فرض می‌شود که شیوهٔ اجرای پروژه تغییر کند. بنابراین، متغیر مدل چرخهٔ حیات از کد ۱ به کد ۴ تغییر کرده است و نتیجهٔ اعمال آن در متغیر هزینه‌ها در شکل‌های ۱۶ الی ۱۸ نمایش داده شده است. لازم به ذکر است که متغیر مدل چرخهٔ حیات با یک تابع لوک‌آپ شرطی و بر اساس نظر خبرگان در چهار حالت آبشاری، حلزونی، چابک و ۷ شکل که چهار شیوهٔ مورد استفاده در شرکت هستند، مدل‌سازی شده بود.



شکل ۱۶. تأثیر سناریوی ۳ بر رفتار متغیر نفرساعت کل



شکل ۱۷. تأثیر سناریوی ۳ بر رفتار متغیر مجموع تلاش هر فرد



شکل ۱۸. تأثیر سناریوی ۳ بر هزینهٔ نیروی انسانی

۵. محدودیت‌های مدل

به‌طور خلاصه می‌توان موانع زیر را از جمله موانعی ذکر کرد که انجام این پژوهش را با محدودیت‌هایی مواجه نمود:

۱. داده‌های مرجع برای بعضی از متغیرها وجود نداشته و محقق مجبور بوده است که به نظرسنجی از خبرگان برای مدل‌سازی توابع لوک‌آپ اکتفا کند؛

۲. مقادیر اولیهٔ متغیرها در واقعیت، بر اساس اطلاعات موجود در سازمان مورد مطالعه در ابتدای سال ۱۳۹۴ تعیین شده است و رفتار متغیرها نیز بر اساس اطلاعات موجود در پایگاه‌های دادهٔ سازمان طی همین سال به‌دست آمده است؛

۳. انجام پروژه با محدودیت‌های زمانی و اجرایی مواجه بوده است؛

۴. رویکرد پژوهشگر در این پژوهش، رویکردی درون‌بنگاهی بوده و عوامل خارج از بنگاه مانند بازار، مصرف‌کننده و ... در نظر گرفته نشده است.

٦. نتیجه‌گیری

با توجه به دستاوردهای این پژوهش و با در نظر گرفتن بازار رقابتی امروز در صنایع نرم‌افزاری و محدودیت‌های منابع و لزوم به‌کارگیری کارایی و اثربخشی، موضوع هزینه‌یابی نرم‌افزار باید در رأس کار مدیران پروژه‌های نرم‌افزاری و سازمان‌های تولیدکنندهٔ نرم‌افزار قرار گیرد تا سیستم قادر به صرفه‌جویی در زمینهٔ هزینه‌های زاید باشد. در این پژوهش سعی شده است که یک مدل پویا برای تخمین هزینهٔ توسعهٔ نرم‌افزار ارائه شود. پس از تهیهٔ مدل، که با رویکرد پویایی سیستم تهیه گردیده و توسط نرم‌افزار «ونسیم» شبیه‌سازی کامپیوتری شده، این مدل مورد آزمون قرار گرفت. در ادامه، پژوهش سناریوهایی برای بهبود رفتار سیستم پیشنهاد شد.

با توجه به نهایی‌شدن مدل این پژوهش، به شرکت‌های نرم‌افزاری پیشنهاد می‌شود که از این مدل برای تخمین هزینه‌های توسعهٔ نرم‌افزار خود استفاده کرده و با مشورت خبرگان داخلی، سناریوهای مناسب سازمان خود را توسعه دهند. همچنین، مدل این پژوهش به‌عنوان یک مدل پیشنهادی برای تصمیم‌گیری در اختیار شرکت‌های نرم‌افزاری قرار داده می‌شود. بر اساس نتایج حاصل از مقایسهٔ رفتار متغیرهای کلیدی برای هر سه سناریوی پیشنهادی، به شرکت «مگفا» پیشنهاد می‌شود که اگر خواهان صرفه‌جویی در هزینه‌های توسعهٔ نرم‌افزار است، برای کاهش هزینه‌های نهایی توسعهٔ نرم‌افزار و بهبود عملکرد خود از مدل چرخهٔ حیات چابک به‌جای مدل چرخهٔ حیات آبشاری استفاده کند. این سازمان همچنین می‌تواند از اقدامات مؤثر مربوط به مدیریت منابع انسانی در جهت حفظ کارکنان با تجربهٔ خود برای کاهش هزینه‌ها در بازهٔ زمانی طولانی‌مدت نیز بهره ببرد.

مطالعات آتی می‌تواند شامل یک سیستم پشتیبان تصمیم برای محاسبهٔ هزینهٔ نهایی توسعهٔ نرم‌افزار با استفاده از رویکرد پویایی‌شناسی سیستم‌ها، ترکیب روش‌های هوش مصنوعی و داده‌کاوی با پیش‌بینی سیستم پویا برای تخمین هزینهٔ توسعهٔ نرم‌افزار باشد.

در مقایسه با پژوهش‌هایی که از رویکرد پویایی‌شناسی سیستم‌ها استفاده نکرده‌اند، می‌توان به موارد زیر اشاره کرد:

«کلانتر هرمزی و عزیز مزرعه» در پژوهش خود فقط عوامل ذکرشده در مدل تخمین هزینهٔ «کوکومو» را در نظر گرفته‌اند. به‌علاوه مدل ارائه‌شده توسط آن‌ها با دید پویا به مسئلهٔ تخمین نگاه نکرده است (۱۳۹۵). پژوهش حاضر، عوامل مؤثر بر هزینه‌های توسعهٔ

نرم‌افزار را از مدل‌های مختلف استخراج کرده و همچنین، با دید پویا به مسئلهٔ تخمین هزینهٔ توسعهٔ نرم‌افزار پرداخته است. «زارع» در پایان‌نامهٔ خود فقط عوامل مؤثر در دو روش «کوکومو» و تحلیل کارکردها را در نظر گرفته است (۱۳۹۱). همچنین، مدل ارائه‌شده توسط «زارع» بر اساس مدل چرخهٔ حیات آبشاری طراحی گردیده است. پژوهش حاضر، عوامل مؤثر بر هزینه‌های توسعهٔ نرم‌افزار را از مدل‌های مختلف استخراج کرده و مدل ارائه‌شده نیز با انواع مدل‌های چرخهٔ حیات توسعهٔ سیستم مانند مدل آبشاری، مدل V شکل، مدل تکراری و مدل چابک مناسب است. «شقوفر» و همکاران فقط چگونگی تخمین تلاش در پروژه‌های چابک را در نظر گرفته‌اند (Schweighofer 2016)، اما در مدل ارائه‌شده در پژوهش حاضر، تخمین تلاش و هزینه برای سایر مدل‌های چرخهٔ حیات توسعهٔ سیستم مانند مدل آبشاری، مدل V شکل، مدل تکراری نیز در نظر گرفته شده است. «کریشنان» تنها به بررسی نقش عوامل تیمی در هزینه‌های توسعهٔ نرم‌افزار پرداخته است (Krishnan 1998)، اما در مدل ارائه‌شده در این پژوهش علاوه بر نقش عوامل تیمی، سایر عوامل پویای تأثیرگذار بر هزینه‌های توسعهٔ نرم‌افزار نیز دیده شده‌اند.

برای مقایسه با پژوهشی که از رویکرد پویایی‌شناسی سیستم‌ها استفاده کرده است، می‌توان به مدل Abdel-Hamid & Madnick (1989) اشاره کرد که بیشتر بر پیش‌بینی تغییرات در هزینه و نیازمندی‌های نیروی انسانی و زمان‌بندی پروژه تمرکز داشته و به عوامل تأثیرگذار بر هزینه و پیش‌بینی هزینهٔ نهایی توجه نکرده است.

فهرست منابع

زارع، فاطمه. ۱۳۹۱. توسعه یک مدل ترکیبی برای تخمین زمان و قیمت تمام‌شدهٔ نرم‌افزار در شرایط عدم اطمینان. پایان‌نامهٔ کارشناسی ارشد ،دانشکده مهندسی صنایع. دانشگاه یزد.

شاه‌محمدی، غلامرضا. ۱۳۷۹. *طراحی مدل تخمین هزینه و زمان برای ساخت سیستم‌های شیءگرا*. پایان‌نامهٔ کارشناسی ارشد. دانشکدهٔ فنی‌مهندسی. دانشگاه تربیت مدرس.

کلانتر هرمزی، شهرزاد، و آرش عزیز مزرعه. ۱۳۹۵. یک روش مبتنی بر منطق فازی برای تخمین تلاش توسعهٔ نرم‌افزار. اولین کنفرانس ملی مهندسی کامپیوتر، علوم کامپیوتر و فناوری اطلاعات، قم، دانشگاه جامع علمی کاربردی استانداری قم، https://www.civilica.com/Paper-NCRC01-NCRC01_128.html

References

Abdel-Hamid, T. K., & S. E. Madnick. 1989. Lessons learned from modeling the dynamics of software development. *Communications of the ACM* 32 (12): 1426-1438.

Ahiaga-Dagbui, D. D., P. E. D. Love, S. D. Smith, & F. Ackermann. 2017. Toward a Systemic View to Cost Overrun Causation in Infrastructure Projects: A Review and Implications for Research. *Project Management Journal* 48 (2): 88–98. https://doi.org/10.1177/875697281704800207

Anusha, J., & R. Mukesh. 2013. A Study On Software Cost Estimation. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS*)-SpecialIssue,ISSN2278-6856

BAŞAR, A. 2017. Hesitant fuzzy pairwise comparison for software cost estimation: a case study in Turkey. *Turkish Journal of Electrical Engineering & Computer Sciences* 25 (4): 2897-2909

Blackmore, C., R. Ison, & M. Reynolds. 2015. Thinking Differently About Sustainability: Experiences from the UK Open University BT - Integrating Sustainability Thinking in Science and Engineering Curricula: Innovative Approaches, Methods and Tools. In W. Leal Filho, U. M. Azeiteiro, S. Caeiro, & F. Alves (Eds.) (pp. 613–630). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-09474-8_43 (accessed Sept. 14, 2014).

Boehm, B. W. 2001. Software engineering economics. In Pioneers and Their Contributions to Software Engineering (pp. 99-150). Berlin Heidelberg: Springer.

Cao, L., B. Ramesh, & T. Abdel-Hamid. 2010. Modeling dynamics in agile software development. *ACM Transactions on Management Information Systems (TMIS)* 1 (1): 5.

Chemuturi, M. 2009. *Software estimation best practices, tools & techniques: A complete guide for software project estimators*.: J. Ross Publishing.

Huang, X., D. Ho, J. Ren, & L. F. Capretz. 2007. Improving the COCOMO model using a neuro-fuzzy approach. *Applied Soft Computing* 7 (1): 29-40.

Hjorth, P., & A. Bagheri. 2006. Navigating towards sustainable development: A system dynamics approach. *Futures* 38 (1): 74-92.

Krishnan, M. S. 1998. The role of team factors in software cost and quality: An empirical analysis. *Information Technology & People* 11 (1): 20-35.

Kumari, K. 2014. Software cost estimation techniques. *International Journal of Emerging Research in Management & Technology* 3 (4): 104-108.

Kumari, S., & S. Pushkar. 2013. Performance analysis of the software cost estimation methods: a review. *International Journal of Advanced Research in Computer Science and Software Engineering* 3 (7):.

Lagerström, R., L. M. von Würtemberg, H. Holm, & O. Luczak. 2012. Identifying factors affecting software development cost and productivity. *Software Quality Journal* 20 (2): 395-417.

Leung, H., & Z. Fan. 2002. Software cost estimation. *Handbook of Software Engineerin.* Hong Kong: Polytechnic University.

Madachy, R., & B. Boehm. 2008. *Comparative analysis of cocomo ii, seer-sem and true-s software cost models*. USC-CSSE-2008-816. Los Angeles: University of Southern California Center for Systems and Software Engineering.

Marujo, Lino Guimarães. 2009. Rework Impacts Evaluation through System Dynamics Approach in Overlapped Product Development Schedule. *Journal of technology management & innovation* 4 (2): 90-101.

Potda, M. S. M., A. Ingle, & M. Puri. 2014. Factors Influencing on Cost Estimation for Software Development. *Global Journal of Advanced Engineering Technologie* 3 (2): 119-123.

Ramírez Cruz, N. E. 2014. Application of System Dynamics to model rework in construction projects. Master's thesis, Universitat Politècnica de Catalunya.

Rani, J., & R. Behel. 2014. Comparison of Cost Estimation Technique. *International Journal of Advanced Research in Computer Science and Software Engineering.* 4 (5): 1005-1009.

Rijwani, P., S. Jain, & D. Santani. 2014. Software Effort Estimation: A comparison based Perspective.

International Journal of Application or Innovation in Engineering and Management (IJAIEM) 3 (12): 18-29.

Sachan, A., B. S. Sahay, & D. Sharma. 2005. Developing Indian grain supply chain cost model: a system dynamics approach. *International Journal of Productivity and Performance Management* 54 (3): 187-205.

Sandhu, P. S., M. Prashar, P. Bassi, & A. Bisht. 2009. A model for estimation of efforts in development of software systems. *World Academy of Science, Engineering and Technology* 56: 148-152.

Schweighofer, T., A. Kline, L. Pavlic, & M. Hericko. 2016. How is effort estimated in agile software development projects? Fifth workshop on software quality analysis, monitoring, improvement, and applications SQAMIA (pp. 73-80).

Sharma, N., A. Bajpai, & M. R. Litoriya. 2012. A Comparison of software cost estimation methods: A Survey. *The International Journal of Computer Science and Applications (TIJCSA)* 1 (3): 121-127.

Sharma, S., & A. Kaushik. 2016. A Review on Recent Trends in Software Cost Estimation 3 (1): 233–238.

Sterman, J. D. 2000 Business dynamics, system thinking and modeling for a complex world. Irwin: McGraw-Hill, ISBN:978-0072389159.

Tailor, O., J. Saini, & M. P. Rijwani. 2014. Comparative Analysis of Software Cost and Effort Estimation Methods: A Review. *International Journal of Computer Science and Mobile Computing (IJCSMC)* 3 (4): 1364–1374.

Trendowicz, A. 2013. *Software Cost Estimation, Benchmarking, and Risk Assessment: The Software Decision-Makers' Guide to Predictable Software Development*. Switzerland: Springer Science & Business Media.

Trendowicz, A., & J. Münch. 2009. Factors Influencing Software Development Productivity—State-of-the-Art and Industrial Experiences. *Advances in computers* 77: 185-241.

Trendowicz, A., & R. Jeffery. 2014. *Software project effort estimation: Foundations and best practice guidelines for success*. Switzerland: Springer.

Zaid, A., M. Hasan Selamat, A. Abd Ghani, R. Atan, & T. Wei Koh. 2008. Issues in Software Cost Estimation. *IJCSNS International Journal of Computer Science and Network Security* 8 (11): 350–356.

Zahra, S. B., & M. Nazir. 2012. A Review of Comparison among Software Estimation Techniques. *Journal of Information & Communication Technology* 5 (1): 39-45.

محبوبه ریواده

دارای مدرک تحصیلی کارشناسی ارشد مدیریت فناوری اطلاعات گرایش کسب‌وکار الکترونیک از دانشگاه الزهراست.

مباحث موجود در زمینهٔ مدیریت فرایندهای کسب‌وکار، مدیریت سیستم‌های اطلاعاتی و فناوری اطلاعات از جمله علایق پژوهشی وی است.

آمنه خدیور

متولد سال ۱۳۶۰، دارای مدرک تحصیلی دکتری در رشتهٔ مدیریت سیستم از دانشگاه تربیت مدرس است. ایشان هم‌اکنون دانشیار گروه مدیریت در دانشگاه الزهراست. مباحث موجود در زمینهٔ مدیریت سیستم‌های اطلاعاتی و فناوری اطلاعات، مدیریت دانش، سیستم‌های خبره و هوش تجاری از جمله علایق پژوهشی وی است.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی