مجله حسابداری مدیریت
سال دوم / پیش‌شماره سوم / بهار ۱۳۸۸

نقد و بررسی روش های برآورد بهای تمام شده نرم افزار
مطالعه موردی شرکت همکاران سیستم

دکترایرج نوروش[۱]
دکتر بیتا مشایخی[۲]
محمد نوری جاوید[۳]

تاریخ دریافت: ۸۸/۰۱/۲۳ تاریخ پذیرش: ۸۸/۰۳/۰۶

چکیده:

این مقاله به معرفی موضوع برآورد بهای تمام شده نرم افزار می پردازد. بدین منظور درباره انواع روش های برآورد و نقاط قوت و ضعف آنها صحبت شده و در نهایت برای آشنایی بیشتر با موضوع برآورد نرم افزار، یکی از جذاب ترین مدل های برآورد بهای تمام شده نرم افزار، مدل COCOMO، تشریح شده است.

واژه‌های کلیدی: برآورد بهای تمام شده نرم افزار، مدل های برآورد نرم افزار، مدل COCOMO

[1] – دانشیار دانشکده مدیریت دانشگاه تهران
[2] – استادیار دانشکده مدیریت دانشگاه تهران– **نویسنده اول و مسئول مکاتبات**.
آدرس: بزرگراه جلال آل احمد، پل نصر، دانشکده مدیریت دانشگاه تهران Email: mashykhi@ut.ac.ir
[3] – کارشناس ارشد حسابداری دانشگاه تهران

۱- مقدمه

روز به روز بر تعداد شرکت هایی که از فناوری اطلاعات استفاده می کنند افزوده می شود. بکارگیری کامپیوتر در سازمان ها ابزار قدرتمندی برای تولید و مدیریت فراهم آورده و منجر به افزایش بهره وری شده است. طی ۱۰ سال گذشته تمایل شدیدی به استفاده از سیستم های اطلاعاتی تجاری موسوم به سیستم های برنامه ریزی منابع سازمان (ERP[1]) به وجود آمده است.

عموما شرکت های تولید کننده نرم افزار دچار معضل افزایش هزینه ها بیش از میزان بودجه شده می باشند. طبق تحقیقی که توسط موسسه استندیش[2] انجام شده، بطور متوسط هزینه های واقعی یک پروژه نرم افزاری ۱۸۹٪ بیشتر از میزان بودجه شده است و تنها ۱۷٪ پروژه ها به موقع، طبق هزینه بودجه شده و با تمام قابلیت ها و ویژگی هایی که از اول مشخص شده بودند به اتمام می رسند. به دنبال این قضیه، نارضایتی مشتری؛ بی کیفیتی نرم افزارها و ناامیدی تولید کنندگان بوجود می آید.

یک امر ضروری در مراحل اولیه پروژه، برآورد زمان و فعالیت لازم برای تکمیل پروژه است. متاسفانه این موضوع یکی از مشکل ترین کارها در حوزه فناوری اطلاعات است. اغلب پروژه های نرم افزاری دچار مشکل کمبود مالی و زمان می شوند که یکی از دلایل آن برآوردهای اولیه اشتباه می باشد.

برآورد بهای تمام شده نرم افزار فرآیند پیش بینی میزان فعالیت لازم برای ایجاد یک سیستم نرم افزاریست (Johnson-1998) . بررسی داده های تاریخی مربوط به پروژه های مختلف نشان داده است که روند بهای تمام شده با برخی پارامترهای قابل اندازه گیری همبستگی دارند(Johnson-1998). این مشاهدات به ارائه مدل های متعددی منتهی شده است که برای ارزیابی، پیش بینی و کنترل بهای تمام شده نرم افزار می توانند مورد استفاده قرار بگیرند.

در ادامه این مقاله درباره انواع روش های برآورد و نقاط قوت و ضعف آنها صحبت شده و در نهایت برای آشنایی بیشتر با موضوع برآورد نرم افزار، یکی از جذاب ترین مدل های برآورد بهای تمام شده نرم افزار، مدل COCOMO، تشریح شده است.

۲- ادبیات موضوع

به منظور جلوگیری از افزایش هزینه ها و زمان بیش از میزان بودجه شده یک پروژه نرم افزاری، مدل های برآورد بهای تمام شده متعددی به وجود آمده است. به دلیل وجود تحولات شدید در تولید نرم افزار ایجاد مدلی که برآوردهای دقیق از پروژه در اختیار استفاده کننده قرار دهد بسیار مشکل است. لذا یکی از مهمترین اهداف صنعت نرم افزار ایجاد مدل های مفیدی است که منطبق بر چرخه عمر تولید نرم افزار[3] باشد و هزینه تولید یک محصول نرم افزاری را به دقت برآورد کند.

سه عامل در تعیین کل هزینه یک پروژه نقش دارد: بهای نرم افزار و سخت افزار به علاوه هزینه نگهداری؛ هزینه ایاب و ذهاب و آموزش؛ هزینه نیروی انسانی[4]. برای اغلب پروژه ها عمده ترین هزینه، هزینه نیروی انسانی است. رایانه های پرقدرت مناسب برنامه نویسی نرم افزارها نسبتا ارزان هستند. اگرچه ممکن است به واسطه اینکه مراحل تولید نرم افزار در ایستگاه های کاری متعدد انجام می شود، هزینه ایاب و ذهاب زیادی نیاز باشد اما این هزینه ها به نسبت هزینه نیروی انسانی بسیار ناچیز است. به علاوه بکارگیری سیستم های مخابره الکترونیکی از قبیل ایمیل، وب سایت و ویدئو کنفرانس می تواند این هزینه را کاهش دهد.

بررسی مطالعات و تحقیقات انجام شده در زمینه برآورد بهای تمام شده طی سالهای ۱۹۸۹ تا ۲۰۰۴

نشان می دهد که رایـج تـرین موضـوع ایـن تحقیقـات معرفی و ارزیابی مدل هـای بـرآورد بـوده اسـت. ایـن بررسی نشان می دهد که اکثر مقالات به بررسـی مـدل های برآورد از نقطه نظر فنی پرداخته انـد (جـدول ۱). همچنین این تحقیقات حاکی از آن اسـت کـه مقـالات ارائه شده غالبا به مطالعه رویکردهای برآورد مبتنـی بـر رگرسیون پرداخته اند. باید به این نکته توجـه کـرد کـه بیشتر مدل های پـارامتری رایـج، مثـل COCOMO، جزو این گروه قرار می گیرند. تقریبا نیمـی از مقـالات به ساخت، بهبود یا مقایسه مدل ها با مدل هـای مبتنـی بـر رگرسـیون پرداختـه انـد . (Jørgensen, et. al., 2007)

همچنین این تحقیقات حاکی از آن است که مقالات ارائه شده غالبا به مطالعه رویکردهای برآورد مبتنـی بـر رگرسـیون پرداختـه انـد. تقریبـا نیمـی از مقـالات بـه ساخت، بهبود یا مقایسه مدل ها با مدل های مبتنـی بـر رگرسیون پرداخته اند (جدول ۲). (Jørgensen, et. al., 2007)

جدول ۱: طبقه بندی موضوعی مطالعات برآورد بهای تمام شده نرم افزار

موضوع تحقیق	-۱۹۸۹	۱۹۹۰-۱۹۹۹	۲۰۰۰-۲۰۰٤	جمع
روش برآورد	۳۰ (٪۷۳)	۹٦ (٪۵۹)	۵۸ (٪۵۸)	۱۸٤ (٪٦۱)
تابع برآورد	۸ (٪۲۰)	۷ (٪٤)	۳ (٪۳)	۱۸ (٪٦)
بهینه سازی مدل ها	۳ (٪۷)	۱۳ (٪۸)	٤ (٪٤)	۲۰ (٪۷)
تعیین اندازه سیستم	۵ (٪۱۲)	۳۹ (٪۲٤)	۱٦ (٪۱٦)	٦۰ (٪۲۰)
مسائل سازمانی	۹ (٪۲۲)	۲۵ (٪۱۵)	۱٤ (٪۱٤)	٤۸ (٪۱٦)
ارزیابی عدم اطمینان مدل ها	۲ (٪۵)	۱۰ (٪٦)	۱۳ (٪۱۳)	۲۵ (٪۸)
ارزیابی عملکرد برآورد	۲ (٪۵)	۸ (٪۵)	٦ (٪٦)	۱٦ (٪۵)
ویژگی های مجموعه داده های مدل	۰ (٪۰)	۱ (٪۱)	۲ (٪۲)	۳ (٪۱)
سایر	۰ (٪۰)	۳ (٪۲)	۱ (٪۱)	٤ (٪۱)

هر مقاله ممکن است در مورد بیشتر از یک موضوع بحث کند.

جدول ۲: سیر تحقیقات حول موضوع روش های برآورد بهای نرم افزار

موضوع تحقیق	-۱۹۸۹	۱۹۹۰-۱۹۹۹	۲۰۰۰-۲۰۰٤	جمع
رگرسیون	21 (٪51)	76 (٪47)	51 (٪51)	148 (٪49)
برآورد بر اساس مقایسه	1 (٪۲)	15 (٪9)	15 (٪15)	31 (٪10)
قضاوت تجربی	7 (٪۷)	13 (٪۸)	21 (٪٤)	15 (٪۷)
تجزیه کار[٦]	3 (٪7)	5 (٪3)	4 (٪4)	12 (٪4)
Function Point	7 (٪17)	47 (٪29)	١٤ (٪١٤)	68 (٪22)
طبقه بندی و رگرسیون	0 (٪0)	5 (٪7)	9 (٪11)	14 (٪7)
شبیه سازی	۲ (٪۵)	4 (٪2)	4 (٪4)	10 (٪3)
شبکه های عصبی	0 (٪0)	11 (٪7)	11 (٪11)	22 (٪7)
تئوری	۲0 (٪49)	14 (٪9)	5 (٪5)	39 (٪13)
روش بیز[۷]	0 (٪0)	1 (٪1)	6 (٪6)	7 (٪2)
ترکیب روشهای مختلف	۰ (٪۰)	3 (٪2)	۲ (٪۲)	5 (٪2)
سایر	2 (٪5)	7 (٪4)	16 (٪16)	25 (٪8)

هر مقاله ممکن است در مورد بیشتر از یک موضوع بحث کند.

۲-۱- روش های برآورد

سازمان ها به برآورد هزینه و نیروی انسانی نیازمندند. برای این منظور یکی از روش های موجود در جدول ۳ استفاده می شود (Sommerville, 2004).

هر تکنیک برآورد نقاط قوت و ضعف خودش را دارد. هر کدام نیاز به اطلاعات مختلفی در مورد پروژه دارند لذا اگر اطلاعات مورد نیاز مدل دقیق نباشد برآورد حاصل درست نخواهد بود. هیچ مدلی نتوانسته است بطور مداوم و در همه شرایط برآوردهای دقیقی ارائه دهد. این بدان علت است که برخی از اطلاعات مهم پروژه در مراحل اولیه کار بسیار مبهم و ناقص هستند.

جدول ۳ : روش های برآورد

روش	شرح
مدل سازی الگوریتمی بهای تمام شده[8]	بر اساس اطلاعات موجود درباره بهای تمام شده تاریخی مدلی طرح می شود که برخی مشخصات نرم افزار (معمولا اندازه آن را) به بهای تمام شده پروژه مربوط می کند.
قضاوت حرفه یی[9]	چندین فرد خبره در زمینه مربوط به حوزه پروژه اقدام به برآورد بهای تمام شده آن می کنند و در نهایت با مقایسه برآوردها و بحث و تبادل نظر به یک اتفاق نظر درباره برآوردها می رسند.
برآورد بر اساس مقایسه[10]	این تکنیک زمانی کاربرد دارد که پروژه های دیگری مشابه با نرم افزار کنونی قبلا اجرا شده باشند. بهای پروژه جدید از طریق مقایسه با پروژه های قبلی برآورد می شود.
قانون پارکینسون[11]	قانون پارکینسون چنین بیان می کند که کار تا هنگامی توسعه می یابد که همه زمان موجود به اتمام برسد. در این روش بهای تمام شده با توجه به منابع موجود تعیین می شود نه بر اساس ارزیابی اهداف پروژه.
Price to win	بهای تمام شده نرم افزار تا آن میزانی که مشتری قادر به خرج کردن برای پروژه می باشد برآورد می شود. میزان برآورد به بودجه مشتری بستگی دارد نه به قابلیت های سیستم.

۲-۲- روش قضاوت حرفه یی

قضاوت حرفه یی که رایج ترین شیوه می باشد بر تجربه جمعی یک تیم از افراد متخصص استوار است. مدیران سیستم های اطلاعاتی عمدتا به روش قضاوت حرفه یی اعتماد می کنند چون برآوردها به سادگی و بدون استفاده از ابزارها یا تکنیک های پیچیده بدست می آیند. برآوردهای حاصل از روش قضاوت حرفه یی معمولا خیلی کمتر از میزان واقعی هستند (Jørgensen, et. al., 2004).

۲-۳- روش برآورد بر اساس مقایسه

روش برآورد براساس مقایسه را می توان بطور مختصر این گونه توضیح داد که این روش سه مرحله دارد: در مرحله اول، مشخصه های پروژه ای که قرار است برآورد شود را بر اساس مشخصات عمومی حاصل از اطلاعات تاریخی مجموعه ای از پروژه های تکمیل شده قبلی بدست می آورند. در مرحله دوم، یک یا چند پروژه مشابه (همسایه[5] یا تمثیل[6]) از میان مجموعه با توجه به معیارهای تشابه که قبلا تعریف شده اند، انتخاب می شوند. در مرحله نهایی، هزینه پروژه های همسایه با هم ترکیب می شوند تا برآورد هزینه پروژه مورد نظر را شکل دهند (معمولا به شکل میانگین موزون یا میانگین ساده). برخی مطالعات نشان داده اند که نتایج این روش با روش های رگرسیونی قابل مقایسه و حتی در بعضی مواقع بهتر از آنها هستند (Mittas, 2007).

۲-٤- مدل سازی الگوریتمی بهای تمام شده

این شیوه از یک فرمول ریاضی برای پیش بینی هزینه پروژه، از طریق برآورد های اندازه و تعداد پرسنل متخصص لازم و سایر عوامل، استفاده می کند. یک مدل الگوریتمی از طریق بررسی و تحلیل هزینه و مشخصات پروژه های تکمیل شده به منظور یافتن یک فرمول که با اطلاعات واقعی بهترین تناسب را داشته باشد، بدست می آید.

متاسفانه همه مدل های الگوریتمی از مشکلات بنیادی مشابهی بدین شرح رنج می برند:

۱. برآورد اندازه در مراحل اولیه پروژه اغلب کار مشکلی است.
۲. برآورد فاکتورهای تعدیل کننده مدل بسیار ذهنی هستند. برآورد دو نفر با هم تفاوت بسیاری دارد که به علت زمینه کاری و تجربه آنها در کار با نوع سیستمی است که تولید می کنند.

تعداد خطوط کد برنامه[۷]، واحد اصلی اندازه گیری نرم افزار در خیلی از مدل های الگوریتمی می باشد. همچنین برآورد اندازه نرم افزار می تواند به روش مقایسه با پروژه های تکمیل شده دیگر، برآورد به وسیله تبدیل Function Point به تعداد خطوط کد برنامه، برآورد بر اساس رتبه بندی اجزاء برنامه و استفاده از یک مرجع معتبر برای تعیین اندازه اجزاء و یا می تواند از طریق قضاوت حرفه یی مهندسین نرم افزار باشد. برآورد دقیق اندازه کد برنامه در مراحل اولیه پروژه بسیار مشکل است زیرا اندازه کد برنامه به تصمیماتی که برای طراحی نرم افزار مربوط می شود بستگی دارد که در مراحل اولیه این تصمیمات هنوز اتخاذ نشده اند.

Function Point یک واحد اندازه گیری نرم افزار است؛ درست مثل ساعت برای اندازه گیری زمان، مایل برای اندازه گیری مسافت و سلسیوس برای اندازه گیری دما. این روش، نرم افزار را از طریق شمارش قابلیت هایی که بر اساس طراحی منطقی برای کاربر فراهم می شود، اندازه گیری می کند.

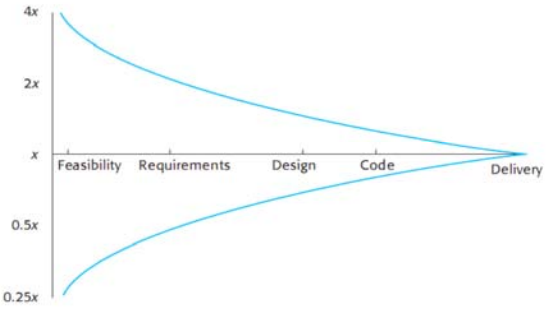
فرآیندهای قابل شمارش دو نوعند: فرآیندهای مربوط به داده ها[۸]، فرآیندهای مربوط به نقل و انتقال داده ها[۹]. نوع اول وظیفه ذخیره و نگهداری داده ها و نوع دوم قابلیت دریافت، تغییر و یا ارسال داده ها به کاربر یا برنامه های دیگر را دربر می گیرند. بعد از تعیین فرآیندها با در نظر گرفتن ضریب پیچیدگی برای هر کدام از فرآیندها، تعداد Function Point تعدیل نشده[۱۰] از طریق جمع نتایج حاصل بدست می آید. (Longstreet, 2004)

دقت برآوردهای یک مدل برآورد الگوریتمی به میزان اطلاعات در دسترس از پروژه نرم افزاری بستگی دارد. با پیشرفت فرآیند تولید نرم افزار اطلاعات بیشتری از پروژه بدست می آید که باعث می شود برآوردهای پروژه دقیق تر انجام شود. اگر برآورد اولیه از میزان فعالیت لازم برای پروژه x ماه باشد، مقدار واقعی میزان فعالیت در فاصله بین 0.25x و 4x قرار دارد. این فاصله با پیشرفت تولید، همانطور که در شکل ۱ نشان داده شده، کمتر می شود (Boehm, et al., 1995).

۲-۵- مشکلات مدل های الگوریتمی موجود

اینکه آیا مدل های الگوریتمی موجود برای استفاده در حیطه وسیعی از پروژه های نرم افزاری معتبر هستند یا نه جای سوال است؟ کمرر[۱۱] در مطالعه مستقلی که روی مدل COCOMO انجام داد به این نتیجه رسید که اختلاف بین مقادیر پیش بینی شده و مقادیر واقعی بیش از ۶۰۰ درصد می باشد (Johnson, 1998). دلایل این امر می تواند یکی از موارد زیر باشد: ساختار مدل ها، پیچیدگی پروژه ها و برآورده اندازه پروژه.

شکل ۱ : فاصله اطمینان برآوردهای الگوریتمی در مراحل مختلف تولید نرم افزار



۱-۵-۲- ساختار مدل ها

اگرچه اکثر محققین و متخصصین با این موضوع موافقند که اندازه پروژه اصلی ترین عامل تعیین کننده میزان فعایت لازم برای تکمیل پروژه است؛ اما رابطه بین اندازه و میزان فعالیت به روشنی مشخص نیست (Johnson, 1998). بیشتر مدل ها میزان فعالیت را بر حسب اندازه تعیین می کنند و برای نشان دادن این موضوع که پروژه های بزرگتر فعالیت بیشتری لازم دارد، اندازه را به توان می رسانند ($Effort = Size^{b}$). این موضوع بطور تجربی تائید شده اما شواهد کمی برای تائید آن وجود دارد (Johnson, 1998).

غالب مدل ها تحت شرایطی که در آن ایجاد شده است خوب عمل می کنند اما وقتی که تعمیم داده می شود ضعیف هستند. مدل COCOMO81 از تحلیل ۶۳ پروژه بدست آمده . مدل COCOMO II حاصل بررسی ۸۳ پروژه است. مدل هایی که بر پایه مجموعه اطلاعات محدودی استوارند بیشتر به ویژگی های خاص آن مجموعه متمایل هستند. این موضوع باعث افزایش دقت مدل در پروژه های مشابه می شود اما دامنه کاربرد مدل را محدود می کند.

۲-۵-۲- پیچیدگی مدل ها

شرایط خاصی که در هر سازمان وجود دارد در بهره وری آنها موثر است .(Johnson, 1998). خیلی از مدل ها (مثل COCOMO و SLIM) عامل تعدیلی دارند که این تفاوت ها را به حساب می آورند. تخمینگر برای لحاظ کردن تفاوت بین پروژه خود با مجموعه اطلاعاتی که مدل بر اساس آنها شکل گرفته بر این عامل تعدیل اطمینان می کند. اما این گونه تعمیم ها معمولا کافی نیستند.

کمرر[۱۲] اظهار می کنند که استفاده از محرک های هزینه مدل COCOMO همیشه باعث بهبود دقت برآورد نمی شود. مدل COCOMO فرض می کند که این محرک های هزینه از هم مستقلند اما در عمل واقعا اینطور نیست. خیلی از این عوامل بر یکدیگر اثر می گذارند و این باعث می شود که روی برخی ویژگی ها تاکید بیشتری بشود. از طرف دیگر این محرک های هزینه به شدت ذهنی هستند. به علاوه محاسبات عامل تعدیل معمولا بسیار پیچیده است. مدل SLIM روی عامل فناوری بسیار حساس است، اما این عامل به راحتی محاسبه نمی شود.

۳-۵-۲- برآورد اندازه پروژه

بیشتر مدل ها برآوردی از اندازه پروژه را برای محاسبات خود لازم دارند. با این حال، برآورد اندازه پروژه در آغاز آن مشکل است. خیلی از مدل ها از تعداد خطوط کد برنامه برای تعیین اندازه پروژه استفاده می کنند که در مراحل اولیه کار ، قبل از کد نویسی، قابل تعیین نیست. اگرچه می توان برای این منظور از روش Function Points و یا Object Points برای تعین اندازه استفده کرد اما این روش ها بسیار ذهنی هستند.

برآوردهای اندازه ممکن است بسیار نادقیق باشند. برای حصول اطمینان از انجام دقیق پیش بینی اندازه باید روش های برآورد و مجموعه داده ها با هم سازگار باشند. اگر روش اندازه گیری مورد استفاده مدل با روشی که در عمل بکار می رود یکسان نباشند

مدل به نتایج دقیق دست نخواهد یافت (Johnson, 1998).

۲-۶- مدل COCOMO

مدل COCOMO یک مدل تجربی است که از جمع آوری داده های تعداد زیادی پروژه نرم افزاری بدست آمده. این داده ها تجزیه و تحلیل شدند تا بهترین فرمولی که با داده های واقعی تناسب دارد حاصل شود. مدل اولیه در سال ۱۹۸۱ منتشر شد. بعدها بوهم و همکارانش مدل مذکور را بهبود داده و مدل COCOMO II را ارائه کردند که باعث تحولات زیادی در مهندسی نرم افزار شد.

مدل COCOMO به دلایل متعدد نسبت به مدل های دیگر برتری دارد:

۱. مستندات کافی از این مدل در اختیار عموم قرار دارد و ابزارهای تجاری متعددی برای استفاده از آن در دسترس می باشد.
۲. این مدل بطور گسترده ای در سازمان های مختلف مورد ارزیابی و استفاده قرار گرفته است.[۱۳]

۲-۷- مدل COCOMO II

کاهش چشمگیر هزینه سخت افزار رایانه و رواج استفاده از بسته های نرم افزاری آماده در پروژه های نرم افزاری باعث کاهش هزینه تولید نرم افزار شده است. همزمان با این موضوع نسل جدید فرآیند های تولید نرم افزار و محصولات نرم افزاری به میدان آمده اند و در حال تغییر نحوه تولید نرم افزار در سازمان ها هستند. این رویکردهای جدید – مانند فرآیندهای نرم افزاری پویا و ریسک محور، زبان های برنامه نویسی نسل چهارم، برنامه های رایانه ای مولد نرم افزار[۱۴]، نرم افزارهای تجاری آماده برای استفاده[۱۵]، روش های میانبر تولید نرم افزار[۱۶] – باعث افزایش کیفیت محصولات نرم افزاری، کاهش هزینه تولید و کاهش ریسک و چرخه عمر آنها می شود.

با این حال، مدل های موجود برآورد بهای تولید نرم افزار این روش های جدید را بطور کامل پوشش نمی دهند. این مباحث منجر به ارائه نسخه جدید از مدل COCOMO شد (Boehm, 1995). مدل اولیه COCOMO (Boehm, 1981)، برای پروژه های نرم افزاری زمان خود بسیار مناسب بود. اما در مواجه با روش های جدید تولید نرم افزار با مشکلاتی مواجه شده بود.

مدل COCOMO II شامل مدلهای فرعی زیر می باشد:

الف) مدل Application Composition که شامل روش تهیه یک نمونه اولیه از نرم افزار، که در اصطلاح به آن پروتو تایپ[۱۷]می گویند، برای کار روی مسائل پر ریسک مثل رابط کاربر، عملکرد نرم افزار و فناوری مورد استفاده می باشد.

ب) مدل Early Design که برای اکتشاف گزینه های مختلف معماری نرم افزار و مفاهیم عملیات مناسب است. در این مرحله تولید اطلاعات کافی برای انجام برآوردهای دقیق وجود ندارد. این مدل زمانی بکار می رود که نیازسنجی مشتری کاملا انجام شده و مراحل اولیه طراحی سیستم در حال اجراست. در این مرحله هدف، انجام برآورد بدون مشقت زیاد است. این مدل از ۷ ضریب تعدیل برای برآورد تشکیل شده است.

ج) مدل Post-Architecture مفصل ترین و دقیق ترین مدل COCOMO II است که مراحل تولید و نگهداری محصول نرم افزاری را در بر می گیرد. در این مرحله اطلاعات بیشتری از نرم افزار و فرآیند تولید آن در دسترس می باشد. این مدل از مجموعه ۱۷ ضریب تعدیل برای برآورد دقیق تر نرم افزار استفاده می کند.

۳- مدل سازی بهای تمام شده در COCOMO II

با دانستن اندازه پروژه، میزان نیروی انسانی لازم برای تکمیل پروژه بر مبنای نفر- ماه[۱۸] از فرمول زیر بدست می آید:

$$PM = A \times (Size)^{B} \quad \text{(فرمول ۱)}$$

برای تعیین مقدار B مقادیر عددی درجه بندی فاکتورهای تعدیل اندازه با هم جمع شده و در فرمول زیر قرار می گیرند:

$$B = 0.91 + 0.01\sum W_i \quad \text{(فرمول ۲)}$$

در مدل COCOMO II پنج فاکتور برای تعدیل اندازه به قرار زیر تعریف شده است:

۱. سابقه اجرایی[۱۹]: میزان تجربه قبلی شرکت را در کار بر روی این نوع نرم افزار نشان می دهد. خیلی کم یعنی شرکت هیچگونه تجربه قبلی در این زمینه ندارد. فوق العاده زیاد به معنی این است که شرکت کاملا با موضوع آشناست.

۲. قابلیت انعطاف در تولید[۲۰]: نشان دهنده میزان انعطاف در تولید است. خیلی کم یعنی فرآیندهای تولید از قبل تعیین شده هستند. فوق العاده زیاد به معنی این است که مشتری فقط اهداف کلی را معین کرده.

۳. ریسک معماری نرم افزار[۲۱] : میزان تحلیل و بررسی ریسک انجام شده را اندازه می گیرد. خیلی کم یعنی بررسی ناچیزی صورت گرفته و فوق العاده زیاد یعنی بررسی به صورت کامل و مفصل انجام شده است. این فاکتور به دنبال پاسخ به این سوال است که معماری نرم افزار تا چه اندازه دقیق معین شده است.

۴. همکاری تیمی[۲۲]: میزان همکاری اعضای تیم تولید را با همدیگر نشان می دهد. خیلی کم یعنی اعضا تیم در همکاری و ارتباط با هم دچار مشکل هستند. فوق العاده زیاد یعنی اعضا همکاری بی وقفه ای با هم دارند.

۵. بلوغ فرآیندهای تولید[۲۳]: میزان بهبود انجام گرفته در فرآیندهای تولید را نشان می دهد. این فاکتور یک درجه بندی ۵ سطحی دارد. در سطح اول، شرکت دارای فرآیندهای تثبیت شده ای برای تولید نیست. این گونه شرکت ها شرایط با ثباتی ندارند و قادر به تکرار موفقیت های قبلی نیستند. در سطح دو، شرکت ها قادر به تکرار فرآیندهای موفق قبلی در پروژه های بعدی هستند. شرکت ها در این سطح از اندکی مدیریت پروژه بهره مند هستند اما هنوز خطر سرریز شدن هزینه ها از میزان بودجه وجود دارد. در سطح سه، شرکت فرآیندهای استانداردی را برای تولید نرم افزار تعریف کرده و در طول زمان آنها را بهبود می دهد. شرکت ها از یک ثبات رویه در بین پروژه های مختلف خود برخوردارند. در سطح چهار، مدیریت با استفاده از اندازه گیری های دقیق، فرآیند تولید را به طور موثری کنترل می کند. مدیریت می تواند راه های تعدیل و تطبیق فرآیندها با ویژگی های پروژه های مختلف را با کمترین میزان افت کیفیت شناسایی کند. در سطح پنج، شرکت ها به بهبود مستمر فرآیندها از طریق بهبود نوآوری های فناوری متمرکز هستند.

۳-۱- ضرایب تعدیل کننده برآورد فعالیت[۲۴]

در مدل Post-Architecture هفده ضریب تعدیل برآورد برای نشان دادن اثر شرایط حاکم بر فرآیند تولید در برآورد نرم افزار تعریف شده است. ارزش اسمی تعیین شده برای هر کدام از این ضرایب برابر ۱ است. اگر ضریب، اثر افزاینده روی برآورد اولیه داشته باشد مقدار آن بزرگ تر از ۱ است. بالعکس اگر ضریب، عامل نیروی انسانی را کاهش دهد مقدار آن کوچک تر از ۱ می شود. بنابراین، مقدار ضریب A در فرمول ۱ به وسیله فرمول زیر محاسبه می شود:

$$A = 2.94 \times \prod EM_i \quad \text{(فرمول ۳)}$$

مجموعه ۱۷ ضریب تعدیل مدل Post-Architecture در ۴ گروه مختلف دسته بندی شده اند:

۱. فاکتورهای محصول که ویژگی های لازم برای تولید محصول نرم افزاری هستند. شامل:

a. Required Software Reliability (RELY)
b. Database Size (DATA)
c. Product Complexity (CPLX)
d. Required Reusability (RUSE)
e. Document Match to Life-Cycle Needs (DOCU)

۲. فاکتورهای پلت فورم که به محدودیت های سخت افزاری موجود در نرم افزار دلالت دارند. شامل:

a. Execution Time Constraint (TIME)
b. Main Storage Constraint (STOR)
c. Platform Volatility (PVOL)

۳. فاکتورهای نیروی انسانی که به قابلیت ها و تجربیات تیم تولید نرم افزار می پردازند. شامل:

a. Analyst Capability (ACAP)
b. Programmer Capability (PCAP)
c. Application Experience (AEXP)
d. Platform Experience (PEXP)
e. Language and Tool Experience (LTEX)
f. Personnel Continuity (PCON)

۴. فاکتورهای پروژه که شرایط خاص پروژه تولید نرم افزار را مورد توجه قرار می دهند. شامل:

a. Use of Software Tools (TOOL)
b. Multisite Development (SITE)
c. Required Development Schedule (SCED)

۲-۳- برآورد زمان تکمیل پروژه

مدیران پروژه همانطور که میزان فعالیت لازم برای تکمیل پروژه و هزینه کل پروژه را برآورد می کنند به مقدار زمان لازم برای اجرای پروژه هم نیازمندند. مدت زمان لازم برای تکمیل پروژه جدول زمان بندی پروژه[۲۵] خوانده می شود. مدل COCOMO حاوی فرمولی برای محاسبه مدت زمان لازم برای تکمیل پروژه ($TDEV$) است که برای تمام مدل های فرعی آن بکار می رود:

(فرمول ۴)

$$TDEV = 3.67 \times PM^{(0.28+0.2\times(B-0.91))} \times \frac{\%SCED}{100}$$

$TDEV$ مدت زمان برآوردی بر حسب ماه از زمان شناسایی نیازمندی های سیستم تا هنگامی که تائید می شود که نرم افزار همه نیازمندی های مشخص شده را پاسخ می دهد را دربر می گیرد. PM مقدار نیروی انسانی محاسبه شده از فرمول ۱ می باشد. $\%SCED$ درصد افزایش یا کاهش در زمان بندی پروژه است.

۳-۳- محدوده نتایج خروجی

برخی کاربران مدل COCOMO ترجیح می دهند به جای استفاده از برآورد نقطه ای از برآورد فاصله ای استفاده کنند. برای این منظور وقتی که مقدار فعالیت از طریق یکی از سه مدل COCOMO II برآورد شد مقادیر خوشبینانه و بدبینانه مقدار فعالیت با در نظر گرفتن یک واحد انحراف استاندارد حول مقدار محاسبه شده از فرمول بدست می آید (جدول ۴). این مقادیر را می توان با کمک فرمول ۳ برای محاسبه محدوده زمانی لازم برای تکمیل پروژه بکار برد.

جدول ۴ : برآورد فاصله ای مدل COCOMO II

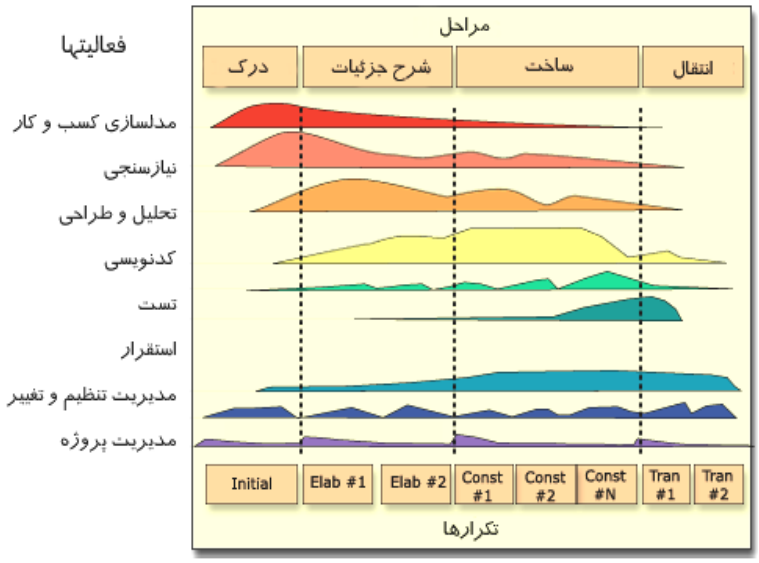
مدل	برآورد خوش بینانه	برآورد بدبینانه
Application Composition	0.50	2.0
Early Design	0.67	1.5
Post-Architecture	0.80	1.25

٤- پروژه برآورد نرم افزار لیزینگ در شرکت همکاران سیستم

شرکت همکاران سیستم یکی از شرکت های خصوصی فعال در صنعت نرم افزار کشور می باشد. فعالیت های اجرایی این شرکت در قالب پروژه های تولید یا نگهداری و پشتیبانی تعریف می شوند. شرکت همکاران سیستم برای تولید محصولات نرم افزاری خود از مدل RUP استفاده می کند.

مدل RUP یکی از مدل های تولید نرم افزار است که توسط شرکت IBM طراحی شد. در این مدل چرخه عمر پروژه به چهار مرحله تقسیم شده است: مرحله درک[۲۶]، شرح جزئیات[۲۷]، ساخت[۲۸] و انتقال[۲۹]. در پایان هر مرحله اگر پروژه ناموفق باشد آن مرحله مجدد تکرار می شود . در هر مرحله چند فعالیت[۳۰] و جریان کار[۳۱] تعریف شده با توجه به میزان اهمیت آن فعالیت در هر مرحله اجرا می شود. این فعالیتها عبارتند از: مدلسازی کسب و کار[۳۲]، نیازسنجی، تحلیل و طراحی، کدنویسی، تست، استقرار[۳۳]، مدیریت تنظیملت و تغییرات، مدیریت پروژه و آماده سازی شرایط محیط برای اجرای پروژه. شکل ۲ یک درک کلی از مدل RUP را به تصویر کشیده است.

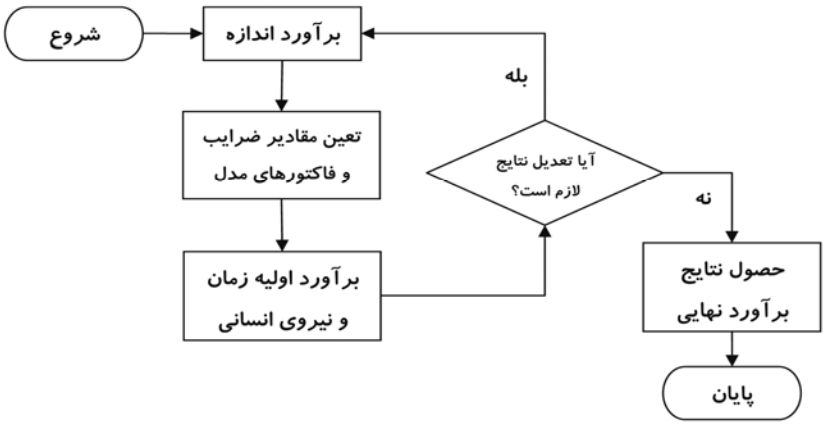
شکل ۲ : مدل RUP



برای اجرای مدل، یکی از پروژه های جاری شرکت همکاران سیستم انتخاب شد. موضوع این پروژه، نرم افزار لیزینگ بود. شرکت در حال تصمیم گیری برای خرید یک نرم افزار لیزینگ از شرکت دیگری بود و قصد داشت درباره خرید یا تولید این نرم افزار تصمیم گیری کند.

شکل ۳ نمایانگر تصویر کلی مراحل اجرای این مدل می باشد. در ابتدا لازم بود که مشخص شود کدام مدل COCOMO II مناسب پروژه می باشد. با توجه به اینکه پروژه در مرحله نیازسنجی بوده و صرفا مشخصات و امکانات کلی آن تعیین شده بود، مدل Early Design به عنوان مناسب ترین گزینه انتخاب شد.

شکل ۳ : مراحل اجرای مدل COCOMO II



اولین قدم برای اجرای مدل COCOMO II تعیین اندازه نرم افزار است. برای این منظور با توجه به اینکه هنوز کد نویسی نرم افزار آغاز نشده بود و به دلیل محدودیت های این روش اندازه گیری (که به آنها اشاره شد) از روش Function Point استفاده شد.

نحوه انجام این کار بدین صورت بود که بر اساس طرح اولیه موجود از نرم افزار در حال طراحی، ابتدا عناصر پنجگانه روش Function Point تعیین و تعداد آنها شمارش شد. پس از شمارش، تعداد Function Point تعدیل نشده (UFP) برابر با ۱۵۴ واحد محاسبه شد (جدول ۵).

جدول ۵ : نتایج نهایی برآورد اندازه پروژه

	EI	EO	EQ	ILF	EIF	
تعداد	21	5	1	2	3	
Complexity	متوسط	متوسط	متوسط	متوسط	متوسط	
	4	5	4	10	7	
UFP	84	25	4	20	21	154

در مرحله بعد مقدار محاسبه شده می بایست به صورت تعداد خطوط کد سورس برنامه تبدیل شده و در مدل COCOMO II وارد شود. برای این کار از نسخه ۸,۲ جدول ارائه شده توسط کاپرز جونز[۳۴] در سال ۱۹۹۶ استفاده شد. جدول مذکور فهرستی از زبان های برنامه نویسی مختلف است که در آن تعداد متوسط خطوط برنامه نویسی به ازای هر واحد Function Point به تفکیک زبان های برنامه نویسی محاسبه شده است. در زمان ارائه جدول مذکور حدود ۵۰۰ زبان برنامه نویسی وجود داشته که رایج ترین آنها در جدول آمده است. زبان های برنامه نویسی ابتدا سطح بندی شده اند. به این معنی که هر چه سطح زبان ها بالاتر می رود تعداد خطوط برنامه نویسی به ازاء هر Function Point کم می شود.

پس از انجام محاسبات مقدار برآوردی تعداد خطوط کد برنامه (ESLOC) برابر با ۳۹۴۲ خط بدست آمد (جدول ۶).

جدول ٦ : برآورد اندازه نرم افزار لیزینگ

زبان	ضریب تبدیل	UFP	*ESLOC
Delphi / VB	29	123	3567
Database	12	31	372
جمع		154	۳۹٤۲
* ESLOC = Estimated Source Lines Of Code			

مرحله بعدی تعیین مقادیر ضریب B و A در مدل است. این ضرایب بوسیله فرمول های شماره ۲ و ۳ محاسبه می شوند:

$$B = 0.91 + 0.01\sum W_i \qquad \text{(فرمول ۲)}$$

$$A = 2.94 \times \prod EM_i \qquad \text{(فرمول ۳)}$$

مقدار ضریب B براساس فرمول ۲ برابر با ۱,۰۵۲۷ طبق جدول ۷ محاسبه شده است.

جدول ۷ : محاسبه ضریب B در مدل

فاکتورهای تعدیل	مقدار	
PREC	معمولی	3.72
FLEX	معمولی	3.04
RESL	زیاد	2.83
TEAM	خیلی زیاد	0
PMAT	معمولی	4.68
B =	**1.0527**	

مقدار ضرایب EM_i بر اساس فرمول ۳ برابر با ۰,۴۷۶۷ محاسبه شده است (جدول ۸). در نتیجه مقدار A برابر خواهد شد با ۱,۴۰۱۵ (۲,۹۴ × ۰,۴۷۶۷).

جدول ۸ : محاسبه ضریب *A* در مدل

نام ضرایب	مقدار ضرایب	
RCPX	زیاد	1.33
RUSE	فوق العاده زیاد	1.24
PDIF	معمولی	1
PERS	خیلی زیاد	0.63
PREX	خیلی زیاد	0.74
FCIL	فوق العاده زیاد	0.62
SCED	زیاد	1
∏EM =	***0.4767***	

در مرحله آخر با داشتن مقدار تمام ضرایب مقدار نیروی انسانی و زمان لازم برای تولید نرم افزار طبق فرمول های ۱ و ۴ بدست می آید.

$$PM = A \times (Size)^{B} \qquad \text{(فرمول ۱)}$$

$$TDEV = 3.\text{۶۷} \times PM^{(0.\text{۲۸} + 0.2 \times (B - 0.91))} \times \%SCED / 100 \qquad \text{(فرمول ۴)}$$

جدول ۹ : نتایج محاسبات مدل

PM	*TDEV*	*B*	*Size*	*EM*	ضریب ثابت
۵/۹۴	27/8	0527/1	942/3	4767/0	2.94

محاسبات جدول فوق بیانگر آن است که برای اجرای پروژه نرم افزار لیزینگ با توجه به شرایطی که طبق ضرایب مدل برای آن فرض شد حدود ۶ نفر – ماه کار نیروی انسانی و ۸ ماه زمان لازم است. مدل COCOMO II هر نفر – ماه را معادل ۱۵۲ ساعت در نظر می گیرد. بنابراین برای تکمیل این پروژه در مجموع 914 ساعت فعالیت مورد نیاز است.

دو نمونه دیگر

برای حصول اطمینان از صحت برآوردها، مدل برای دو پروژه دیگر شرکت اجرا شد. پروژه های مذکور از برنامه های سال ۸۵ شرکت بودند. سایر اطلاعات مربوط به آنها به قرار جدول زیر است:

جدول ۱۰ : مشخصات دو نرم افزار دیگر

نام پروژه	تعداد خطوط کد	مجموع ساعات کارکرد واقعی
سیستم اموال	۳۴۵۰۰	۱۲۶۳
گزارش های مدیریتی	۵۸۰۰۰	۳۵۱۵

هنگام برآورد این پروژه ها معلوم شد که در تولید سیستم اموال از نرم افزارهای تولید خودکار کد استفاده شده است. بنابراین برای تعیین اندازه آن باید از مدل Reuse استفاده کرد. جدول ۱۱ خلاصه محاسبات اندازه را نشان می دهد.

جدول ۱۱ : محاسبه اندازه پروژه ها در مدل Reuse

	کل	٪ AT	معادل کد	اندازه نهایی
سیستم اموال	58000	33 %	19140	38860
سیستم گزارشات مدیریت	34500	0%	0	34500

همانطور که در جدول نشان داده شده ، از مجموع ۵۸۰۰۰ خط کدنویسی انجام شده برای سیستم اموال 19140 خط آن از به طور خودکار تولید شده است. برای تولید این تعداد خط برنامه نویسی ۸ نفر- ماه فعالیت لازم می باشد. در نهایت اندازه سیستم اموال که در محاسبات برآورد استفاده می شود تعداد ۳۸۸۶۰ خط کدنویسی می باشد.

نتایج برآورد در جدول ۱۲ آمده است. طبق جدول ذیل سیستم اموال ۱۳/۵۹ نفر- ماه کار لازم دارد. اما باید ۸ نفر – ماه هم برای تولید خودکار کدها به آن اضافه کرد که در نهایت کل فعالیت لازم ۲۱/۵۹ نفر- ماه می شود. با احتساب ۱۵۲ ساعت کار برای هر نفر میزان کل ساعت کار براوردی ۳۸۶۸ می شود. این مقدار با ساعات واقعی کار در این ساخت سیستم اموال بسیار نزدیک است. در مورد سیستم گزارشات مدیریت براورد انجام شده حاکی از نیاز به ۸/۲۲ نفر-ماه فعالیت است که معادل ۱۴۷۳/۵ ساعت کار می باشد.

جدول ۱۲ : نتایج برآورد مدل COCOMO

	A	EM	Size	B	PM
سیستم اموال	۲/۹۴	۰/۱۳۷۰	۳۸/۸۶۰	۰/۹۶۱۵	۱۳/۵۹
سیستم گزارشات مدیریت	۲/۹۴	۰/۰۸۷۴	۳۴/۵۰۰	۰/۹۷۸۴	۸/۲۲

می توان برآوردها را به جای نقطه ای به صورت فاصله ای انجام داد تا میزان دقت و تطبیق آنها با واقعیت بهتر نمایان شود. برای این منظور از داده های جدول ۴ که در مورد محدوده نتایج خروجی ارائه شد، باید استفاده کنیم. با توجه به اینکه در انجام دو برآورد اخیر از مدل Post-Architecture استفاده کردیم، محدوده نتایج برآورد مدل از ۰/۸۰ برابر مقدار محاسبه شده تا ۱/۲۵ برابر آن می باشد.

با این حساب محدوده برآورد دو پروژه طبق جدول ۱۳ می باشد. مشاهده می شود که مقادیر ساعات واقعی هر دو پروژه در محدوده برآورد قرار دارد.

جدول ۱۳ : مقایسه محدوده نتایج برآورد با مقادیر واقعی

	بدبینانه	متوسط	خوش بینانه	مقدار واقعی
سیستم اموال	۴۸۳۵/۸۹	۳۸۶۸/۷۱	۳۰۹۴/۹۷	3515.5
سیستم گزارشات مدیریت	۱۸۴۱/۹۱	۱۴۷۳/۵۳	۱۱۷۸/۸۲	1263

۵- جمع بندی و نتیجه گیری

مدل های برآورد بهای تمام شده نرم افزار، ابزار مدیریتی مناسبی هستند که این امکان را برای تولیدکننده نرم افزار فراهم می کنند تا پیش از اینکه پروژه نرم افزاری شروع شود و پیش از اینکه شرکت متحمل هزینه های هنگفتی گردد درباره اجرا یا تداوم پروژه تصمیم گیری شود.

از این نکته نباید غافل شد که هیچ برآوردی بدون خطا نیست و همواره درصدی از خطا وجود دارد. البته می توان از طریق تنظیم[۳۵] مدل با استفاده از داده های حاصل از فعالیت یک شرکت نرم افزاری و تحلیل آماری آن ها، برآوردهای بهتری برای پروژه های آن

شرکت بدست آورد که این موضوع خود جای تحقیق و بررسی دارد.

متداول ترین روش های برآورد عبارتند از: روش برآورد بر اساس مقایسه، روش قضاوت حرفه یی و روش مدل سازی الگوریتمی. از میان این روش ها، مدلسازی الگوریتمی جایگاه ویژه یی دارد. یکی از محبوب ترین مدل های الگوریتمی مدل COCOMOمی باشد. این مدل مجموعه یی از چند مدل فرعی است که هر کدام کاربرد خاص خود را دارد.

مدل COCOMO قابلیت لحاظ کردن اثر استفاده از تکه های نرم افزاری آماده و قابل استفاده مجدد[۳۶] را دارد. این به معنای استفاده از قطعات برنامه نویسی شده برنامه های قبلی در پروژه های جدید است. یعنی به جای اینکه یک قسمت از نرم افزار را از ابتدا بنویسند، کد برنامه های قبلی را اصلاح کرده و در کد برنامه جدید می گنجانند. این کار باعث صرفه جویی در زمان و هزینه می شود. این قابلیت مدل COCOMO شایسته بررسی های بیشتر است. همچنین روش تحلیل Function Point خود به عنوان یک مدل برآورد نرم افزار قابل استفاده است. مقایسه نتایج برآورد این مدل با نتایج مدل COCOMO می تواند تصویر بهتری از قابلیت های این دو روش ارائه دهد.

فهرست منابع

1. Boehm, Barry W.,(2000) "Software Cost Estimation with COCOMO II", prentice Hall.
2. Boehm, B.W., B. K. Clark, E. Horowitz, R. Madachy, R.W .Selby, and C. Westland, (1995) "Cost Models for Future Software Processes: COCOMO ",۲,۰ Annals of Software Engineering,
3. Boehm, B.W., B. K. Clark, (1995) "An Overview of the COCOMO 2.0 Software Cost Model"
4. Johnson, k.,(1998) " Software Cost Estimation: Metrics and Models ",
5. Jørgensen, M. and D.I.K. Sjøberg, (2004).The Impact of Customer Expectation on Software Development Effort Estimates. International Journal of Project Management,
6. Jørgensen, M. and Martin Shepperd, (2007) "A Systematic Review of Software Development Cost Estimation Studies", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 33, NO. 1,
7. Khoshgoftaar, Taghi M.,(2004) "Identification of fuzzy models of software cost estimation", Fuzzy Sets and Systems Vol. 145.
8. Longsteert, D.(2004), "Function Point Training Course ", www.softwaremetrics.com
9. Mittas, N., el. al., (2007)"Improving Analogy – Based Software Cost Estimation by a Resampling Method ", Information and Software Technology.
10. Sicilia, M.A., et al, (1999) "Software Cost Estimation with Fuzzy Inputs: Fuzzy Modeling and Aggregation of Cost Drivers", KYBERNETIKA – Vol. 35.
11. Stamelos, J., et al,(2003) "Estimating the development cost of custom software", Information & Management – Vol. 40.

یادداشت‌ها

[1] Enterprise Resource Planning
[2] WWW.STANDISHGROUP.COM
[3] Software Development Life-cycle (SDLC)

[4] Effort Cost
[5] Neighbor
[6] Analogy
[7] Lines of Source Code
[8] Data Functions
[9] Transaction Functions
[10] Unadjusted Function Point
[11] Kemerer
[12] Kemerer
[13] از جمله سازمان ها و شرکت های استفاده کننده از این مدل می توان به این موارد اشاره کرد: مایکروسافت، موتورولا، دایملر کرایسلر، بوئینگ، زیراکس، آی بی ام، سان مایکروسیستمز، و وزارت دفاع ایالات متحده
[14] Application generator
[15] Commercial of-the-shelf (COTS)
[16] Fast-track software development approaches
[17] Prototype
[18] Person – Month (PM)
[19] Precedentedness (PREC)
[20] Development Flexibility (FLEX)
[21] Architecture/risk resolution
[22] Team Cohesion (TEAM)
[23] Process Maturity (PMAT)
[24] Effort Multipliers (EM)
[25] Project Schedule
[26] Inception
[27] Elaboration
[28] Construction
[29] Transition
[30] Discipline
[31] Workflow
[32] Business Modeling
[33] Deployment
[34] Capers Jones
[35] Calibration
[36] Reusable Components