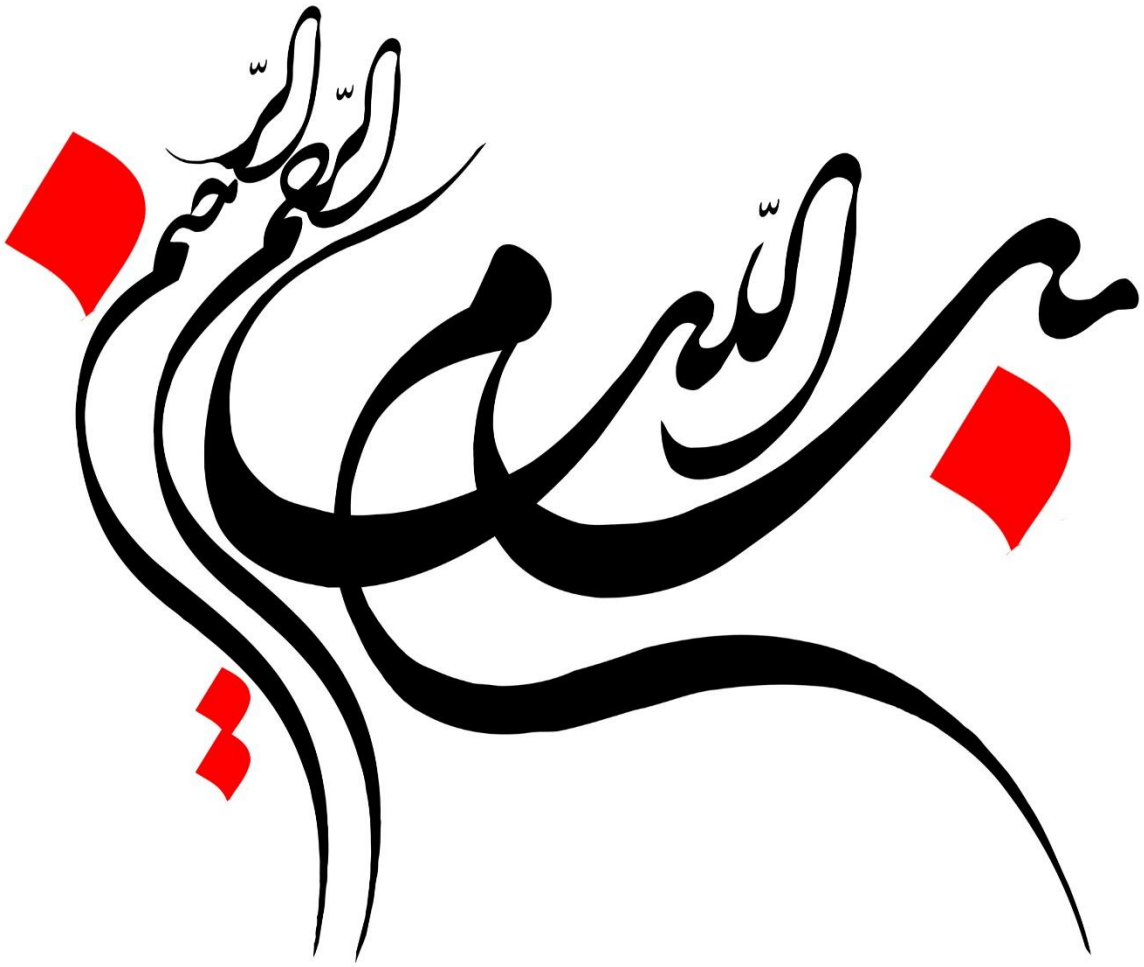

بسم الله الرحمن الرحيم

مهندسی نرم افزار

عنوان تمرین :

برآورد هزینه های پروژه نرم افزاری نمونه مطالعاتی دبیر خانه دانشگاه

استاد راهنما : خانم دکتر استاد زاده

تهیه کننده :

جابر بابکی

Jabber.babaki94@gmail.com

خرداد ۹۴

مقدمه

امروزه نرم افزار از گران ترین اجزاء یک سیستم کامپیوتری محسوب می گردد. تخمین درست هزینه تولید یک سیستم نرم افزاری، باعث می شود که مدیر پروژه در طول چرخه حیات نرم افزار، از پشتوانه قدرتمندی جهت اتخاذ تصمیمات مختلف برخوردار باشد و مدیر پروژه، تحلیل گر، طراح، برنامه نویس و سایر نیروهای تیم توسعه نرم افزار می دانند که برای تولید یک محصول مناسب به چه میزان تلاش و زمان نیاز دارند. بدون داشتن یک تخمین مناسب از هزینه مورد نیاز، مدیر پروژه نمی تواند تشخیص دهد که به چه میزان زمان و چه حجمی از نیروی انسانی و سایر منابع جهت انجام پروژه نیاز دارد و در صورت تشخیص اشتباه، پروژه در مسیر شکست حتمی حرکت خواهد کرد. بررسی های صورت گرفته در مورد بیست هزار پروژه نرم افزاری در طول هجده سال نشان داد، که اکثر پروژه های نرم افزاری به دلیل تخمین هزینه غلط و در نتیجه برنامه ریزی و زمان بندی نادرست مدیران با شکست مواجه شده اند.

در ادامه این تحقیق درباره انواع روش های برآورد و نقاط قوت و ضعف آنها صحبت می شود 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.

❖ ۱ - اندازه نرم افزار

یکی از مهم ترین فاکتورها در تخمین هزینه یک سیستم نرم افزاری، اندازه و حجم آن است. برای تخمین اندازه یک سیستم نرم افزاری یکی از معیارهای زیر مورد استفاده قرار می گیرد .

❖ ۱-۲ LineofCode

تعداد خطوط کد منبع برنامه ای که به کاربر ارائه می شود، به استثناء توضیحات و خطوط خالی و به LOC معروف است و از زبان برنامه نویسی مستقل است. اندازه دقیق LOC پس از پایان پروژه مشخص می شود. یک روش معمول برای تخمین LOC، استفاده از تجربه به همراه تکنیک PERT می باشد. در این روش سه متغیر L برای کمترین اندازه ممکن، H برای بیشترین اندازه ممکن و M برای اندازه معمول کد برنامه در نظر گرفته می شوند. اندازه کد S از فرمول زیر تخمین زده می شود :

$$S = (L + H + 4M)/6$$

❖ ۱-۲ Software Science

در این روش از معیارهای طول کد (Codelength) وحجم (Volume) برای تخمین اندازه نرم افزار استفاده می شود. طول کد، برای محاسبه طول کد منبع برنامه است و با فرمول زیر محاسبه می شود :

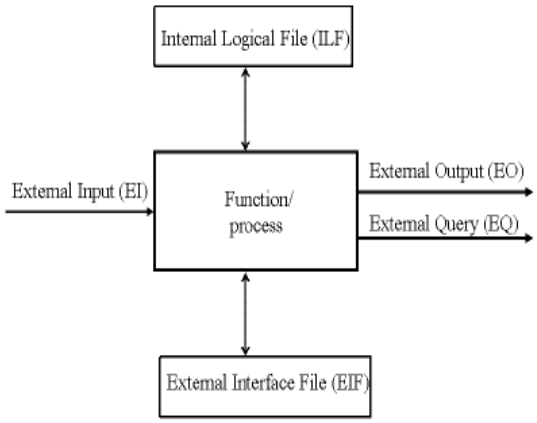
$$N = P + Q$$

که P تعداد کل عملگرها و Q تعداد کل عملوندها در برنامه است.

Function Points ۱-۳

این روش تخمین بر اساس کاربرد نرم افزار می باشد. و یکی از پر کاربرد ترین روش ها می باشد نوع داده های زیر در سیستم بررسی می شوند :

- *External Input (EI)*: user data or user control input type that enters the external boundary of the software system being measured
- *External Output (EO)*: user data or control output type that leaves the external boundary of the software system being measured
- *Internal Logical File (ILF)*: logical file types that are generated, used, or maintained by the software system
- *External Interface File (EIF)*: file types passed or shared between software systems
- *External Inquiry (EQ)*: input-output combination, where input causes and generates an immediate output.



تعریف دیگر از نوع داده های سیستم

1. External Input (EI)
   Functions that move data into the application without presenting data manipulation.
2. External Output (EO)
   Functions that move data to user and presents some data manipulation.
3. External Inquiries (EQ)
   Functions that move data to user without presenting data manipulation.
4. Internal Logical Files (ILF)
   The logic in the form of fixed data managed by the application through the use of External Input (EI)
5. External Interface Files (EIF)
   The logic in the form of fixed data used by the application but did not run in it

بعد از مشخص کردن نوع داده و تعداد آن ها به هر یک از آن ها بر اساس کارکرد هر یک پیچدگی زیر را میدهیم :

- ساده
- متوسط
- پیچیده

و سپس به هر یک بر اساس جدول زیر امتیازی داده

	Complexity		
Function Type	**Low**	**Average**	**High**
External Input	3	4	6
External Output	4	5	7
External Inquiry	3	4	6
Internal Logical File	7	10	15
External Interface File	5	7	10

و سپس تعداد را در امتیازی که داده شده در هر نوع داده ضرب کردن و در پایان این پنج حاصل ضرب را با یکدیگر جمع کرده تا نقاط عملکرد تعدیل نشده یا همان UFP بدست بیاید .

بدست آوردن مجموع درجه تاثیر (TDI) :

برای بدست آوردن TDI باید به ۱۴ فاکتور تعیین شده پاسخ داده بدین صورت که به هر سوال در جه ای از صفر تا ۵ می دهیم و سپس این در جات را با یکدیگر جمع می کنیم .

ID	System Characteristic
C1	Data communications
C2	Distributed functions
C3	Performance objectives
C4	Heavily used configuration
C5	Transaction rate
C6	On-line data entry
C7	End-user efficiency
C8	On-line update
C9	Complex processing
C10	Reusability
C11	Installation ease
C12	Operational ease
C13	Multiple sites
C14	Facilitate change

- آیا سیستم به پشتیبانی و بازیابی قابل اطمینان نیاز دارد؟
- آیا ارتباطات داده ای مورد نیاز است؟
- آیا عملیات پردازشی توزیع شده وجود دارد؟
- آیا کارایی اهمیت دارد؟
- آیا سیستم در یک محیط عملیاتی موجود و پرکاربرد اجرا می شود؟
- آیا سیستم به مدخل داده های آنلاین نیاز دارد؟
- آیا مدخل داده های آنلاین نیاز به ساخت تراکنش ورودی روی عملیات یا صفحات نمایش چندگانه دارد؟
- آیا فایل های اصلی به صورت آنلاین به هنگام می شوند؟
- آیا ورودی ها، خروجی ها، فایل ها و درخواست ها پیچیده اند؟
- آیا پردازش داخلی پیچیده است؟
- آیا کد طوری طراحی شده است که دوباره قابل استفاده باشد؟
- تبدیل و نصب در طراحی لحاظ شده است؟
- آیا سیستم برای نصب چندگانه در سازمان های متفاوت طراحی شده است؟
- آیا برنامه کاربردی طوری طراحی شده است که تغییرات را تسهیل کند و به آسانی توسط کاربر استفاده شود ؟

حال بعد از بدست آوردن TDI که از مجموع امتیاز های که به سوال های بالا پاسخ داده شده است باید تکنیک پیچیدگی تعدیل شده را بدست آورد (TCA) که از فرمول زیر بدست می آید :

$$TCA = 0.65 + 0.01*TDI$$

محاسبه FP :

حال با داشتن TCA و UFP می توان FP را از فرمول زیر بدست اورد :

$$FP = TCA * UFP$$

❖ ۱-۴ ObjectPoint

این روش بر اساس تعداد و پیچیدگی فرمها، گزارشات و اجزاء نرم افزاری زبانهای نسل جدید عمل می کند. هر یک از اشیاء فوق بر اساس تعدادشان وزنی از ۱ (برای فرم های ساده) و ۱۰ (برای اجزاء زبانهای نسل جدید) را می پذیرند و نتیجه، جمع وزنی موارد فوق است. این روش هم در گامهای اولیه پروژه قابل اجراست .

❖ ۲-محاسبه تعداد خطوط کد با استفاده از FP و جدول کاپرز جونز

تعداد خطوط را می توان از فرمول زیر بدست آورد :

Count OF Line = FP * ضریب تبدیل

ضریب تبدیل از جدول زیر بدست می آید :

زبان برنامه نویسی	ضریب تبدیل
زبان اسمبلی	۳۲۰
C	۱۲۸
فرترن	۱۰۶
پاسکال	۹۰
C++	۶۴
ویژوال بیسیک	۳۲
SQL	۱۲
Java	۵۳
پرولوگ	۶۴

همانطور که دیده می شود بعضی از زبان های برنامه نویس در این لیست موجود نمی باشد که با توجه به نزدیکی زبان ها به یکدیگر می توان میانگینی را انتخاب کرد .

❖ ۳- تخمین هزینه

سازمان ها به برآورد هزینه و نیروی انسانی نیازمندند. برای این منظور یکی از روش های موجود استفاده می شود 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.

روش های تخمین هزینه نرم افزار به دو دسته کلی الگوریتمی و غیر الگوریتمی تقسیم می شوند :

- ۱- ۳ روش های غیر الگوریتمی
- ۲ – ۳ روش های الگوریتمی

❖ ۱-۱-۳ برآورد براساس قیاس (AnalogyCosting)

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 چند پروژه مشابه از میان مجموعه پروژه ها انتخاب می شوند . در مرحله نهایی، هزینه پروژه های همسایه با هم ترکیب می شوند تا برآورد هزینه پروژه مورد نظر را شکل دهند (معمولابه 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.

❖ ۲-۱-۳ داوری کارشناسان (ExpertJudgment)

در این روش تخمین هزینه بر اساس تجربه شخصی و روشهای ابدایی افراد متبحر در زمینه توسعه نرم افزار، صورت می گیرد. سپس برای رفع ناهماهنگی های احتمالی در تخمین های ارائه شده توسط افراد

مختلف، از تکنیک هایی که باعث اجماع در تخمین ها می شود، مانند Delphi و PERT استفاده خواهد شد. به طور مثال تکنیک Delphi به شکل زیر عمل می کند :

الف) مشخصات سیستم برای هر یک از افراد شرح داده می شود.

ب) افراد مستقل از یکدیگر و بدون مشورت، تخمین خود را اعلام می کنند.

ج) تخمین های اعلام شده فهرست می شوند و با اعلام آنها به افراد، مجددا درخواست می شود که تخمین دیگری را انجام دهند و منطق مورد استفاده در تخمین را هم ارائه دهند.

د) گام های ب و ج، تا رسیدن به کسب نتیجه مناسب تکرار می شوند

۳ – ۱ – ۳ پاریکسون (Parkinson)

در این روش هزینه نرم افزار تخمین زده نمی شود، بلکه با توجه به منابع موجود (بدون توجه به اهداف پروژه) تعیین می شود. به طور مثال اگر مدت زمان اجری پروژه ۱۲ ماه و تعداد نیروهای موجود ۵ نفر باشد، میزان ۶۰ نفر- ماه تخمین زده می شود. اگر چه این روش در برخی موارد تخمین قابل قبولی را ارائه می دهد ولی تکنیک مناسبی برای تخمین هزینه پروژه نیست.

۴ – ۱ – ۳ تاکید بر فایده (PricingtoWin)

در این روش به جای توجه به نرم افزار، قابلیت ها و کاربردهای آن، هزینه بر اساس میزان بودجه کارفرما تخمین زده می شود. به طور مثال اگر تخمین واقعی پروژه ۱۰۰ نفر- ماه باشد، اما کارفرما بودجه برای ۶۰ نفر- ماه در نظر گرفته باشد، تخمین مجددا بر اساس ۶۰ نفر- ماه انجام می شود.

۵-۱-۳ پایین-بالا (Bottoms-Up)

در این روش هر یک از اجزاء سیستم (Component) به طور مجزا تخمین زده می شوند و سپس مجموع این تخمین ها به عنوان تخمین هزینه کلی پروژه در نظر گرفته می شود. برای استفاده از این روش لازم است که در ابتدا یک طراحی اولیه از سیستم انجام دهیم تا ساختار اجزاء را بدست آوریم.

۶-۱-۳ بالا-پایین (Top-Down)

این روش بر خلاف روش قبلی است و هزینه پروژه با استفاده از روشهای الگوریتمی یا غیر الگوریتمی به شکل یکجا و بر اساس معیارهای کلی تخمین زده می شود. در گام های بعدی این هزینه می تواند بین اجزاء مختلف سیستم توزیع شود.

۲-۳ روش های الگوریتمی

روش های الگوریتمی از مدل های ریاضی برای تخمین هزینه پروژه استفاده می کنند. هر مدل الگوریتمی به صورت تابعی از فاکتورهای هزینه تعریف می شود. روشهای الگوریتمی موجود در دو جنبه با یکدیگر متفاوت هستند، یکی انتخاب فاکتورهای هزینه و دیگری تعریف تابع محاسبه هزینه. ابتدا فاکتورهای هزینه را بررسی می کنیم و سپس روشها را با توجه به تعریف تابع محاسبه عنوان کرده و تحلیلی (Analytical) یا تجربی بودن (Empirical) آنها را بیان می کنیم.

- **ProductFactors** : دارا بودن قابلیت اطمینان، میزان پیچیدگی، حجم پایگاه داده، دارا بودن قابلیت استفاده مجدد، میزان مطابقت مستندات پروژه با نیازهای چرخه حیات پروژه.
- **ComputerFactors** : محدودیت زمان اجرای سیستم، محدودیت فضای ذخیره سازی، محدودیت راه اندازی مجدد کامپیوتر، تنوع پلتفورم.
- **PersonnelFactors** : مهارت تیم تحلیل، مهارت کدنویس ها، میزان تسلط بر پلتفورم، میزان تسلط بر زبان برنامه نویسی و ابزارها، میزان هماهنگی اعضاء تیم.
- **ProjectFactors** : توسعه توزیع شده سیستم(MultisiteDevelop) ، استفاده از ابزارهای نرم افزاری .

- پیچیدگی مدل ها : شرایط خاصی که در هر سازمان وجود دارد در بهره وری آنها موثر است خیلی از مدل ها 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 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 هز ینه 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.

❖ ۱-۲-۳ مدل COCOMO (ConstructiveCostModel)

مدل COCOMO یک مدل تجربی است که از مدل 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 COCOMOII را ارائه کردند که باعث تحولات زیادی در مهندسی نرم افزار شد.

COCOMO مدل 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.

❖ ۲-۲-۳مدل COCOMOII

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 فرایند های نرم افزار افزاری پویا و ریسک محور، زبان های برنامه نویسی نسل چهارم، بر نامه های رایانه ای مولد نرم افزار ، نرم افزارهای تجاری آماده برای استفاده ، روش میانبر تولید نرم افزار 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 . مدل اولیه COCOMO 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 COCOMOII شامل مدل های فرعی زیر می باشد :

- مدل ApplicationComposition که شامل روش تهیه یک نمونه اولیه از نرم افزار، که در اصطلاح به آن پروتو تا یپ می گویند، برای کار روی مسائل پر ریسک مثل رابط کاربر، عملکرد نرم افزار و فناوری مورد استفاده می باشد.
- مدل EarlyDesign 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 ضریب تعدیل برای برآورد تشکیل شده است .
- مدل Post-Architecture مفصل ترین و دقیق ترین مدل COCOMOII است که مراحل تولید 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.

بهای تمام شده در COCOMOII :

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

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

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

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

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

۱- **سابقه اجرایی (PREC) Precedentedness** : 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 (FLEX) Development Flexibility** : 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 (RESL) Architecture / Ris k Resolution** : میزان تحلیل و بررسی ریسک انجام شده را اندازه می گیرد . خیلی کم یعنی بر رسی 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 (TEAM) Team Cohesion** : 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 (PMAT) Process Maturity** : میزان بهبود انجام گرفتن در فرآیندهای تولید را نشان می دهد .

ضرایب تعدیل کننده برآورد فعالیت Effort Multipliers (EM) : در مدل Post-Architecture هفده ضریب تعدیل برآورد برای نشان دادن اثر شرایط حا کم بر فرآیند تولید در برآورد نرم افزار تعریف شده است . مقدار A در فرمول ۱ به وسیله فرمول زیر محاسبه می شود :

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

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

۱- فاکتور های محصول که ویژگی لازم برای تولید محصول نرم افزاری هستند

- ✓ **قابلیت های الزامی نرم افزار Required Software Reliability (RELY) :** این اندازه گیری حدی را برای عملکرد نرم افزار که در یک دوره زمانی انجام می شود را نشان می دهد . اگر تاثیر شکست نرم افزار جزئی باشد LOW در نظر گرفته می شود ، اگر تاثیر شکست ریسک اساسی را ایجاد کند بسیار زیاد در نظر گرفته می شود .
- ✓ **حجم پایگاه داده Required Software Reliability (RELY)** : این اندازه گیری تلاش میکند تاثیرات داده های بزرگ مورد نیاز را بر روی محصول بررسی کند رتبه بندی تعیین می شود با محاسبه D/P بررسی اندازه پایگاه داده مهمه ، زیرا با در نظر گرفتن این موضوع سعی شده نیازمندی های تولید داده های مورد استفاده در برنامه آزمایش شود .

$$\frac{D}{P} = \frac{DataBaseSize(Bytes)}{\{\Pr ogramSize(SLOC)}$$

- ✓ **پیچیدگی محصول Product Complexity (CPLX)** : پیچیدگی به ۵ زمینه تقسیم می شود عملیات کنترل ، عملیات محاسباتی ، عملیات وابسته به دستگاه ، عملیات مدیریت داده ، و کاربر عملیات مدیریت رابط ، انتخاب یک زمینه یا ترکیبی از زمینه زیرسیستم مشخص می باشد ، رتبه بندی پیچیدگی به طور متوسط وزن ذهنی از این مناطق است .
- ✓ **نیازمندی استفاده مجدد Required Reusability (RUSE)** : این معیار ساخت قطعاتی که قابلیت استفاده مجدد را در حال حاضر و آینده دارند حساب می کند

- ✓ **اسناد مورد نیاز با چرخه عمر(Document Match to Life-Cycle Needs (DOCU**

بررسی اسناد پروژه جهت نیاز های چرخه عمر ، رتبه بندی بسیار پایین یعنی بسیاری از نیاز ها چرخه عمر کشف شد و فوق العاده زیاد یعنی نیاز های چرخه عمر کاملا کشف شده است

۲- فاکتور های پلت فرم که به محدودیت های سخت افزاری موجود در نرم افزار دلالت دارد

- ✓ **محدودیت زمان اجرا (Execution Time Constraint (TIME**: این معیار محدودیت زمان اجرا که بر سیستم تحمیل شده است را بررسی میکند ، محدوده رتبه اسمس کمتر از ۵۰ در صد استفاده از منابع زمان اجرا ، خیلی زیاد ۹۵ ٪ از منابع زمان اجرا مصرف می شود .
- ✓ **محدودیت ذخیره سازی (Main Storage Constraint (STOR** : این معیار نشان دهنده درجه محدودیت دخیره سازی بر روی سیستم نرم افزاری است .
- ✓ **نوسانات سخت افزاری (Platform Volatility (PVOL** : کلمه " پلت فرم " استفاده شده در این جمله معنی پیچیده از سخت افزار و نرم افزار (سیستم عامل ، DBMS ، و ...) است که محصول نرم افزار انجام می دهد کار ها را ، نرم افزار توسعه داده شده بر روی سیستم عامل و پلت فرم سخت افزار کامپیوتر است برا DBMS نیز همینطور پلت فرم شامل کامپایلر یا اسمبلر است که حمایت می کند از توسعه سیستم نرم افزاری ، این رتبه بندی در محدوده کم که در آن تغییر عمده هر ۱۲ ماه وجود دارد و بسیار زیاد که در آن یک تغییر عمده هر دو هفته وجود دارد .

۳- فاکتور های نیروی انسانی که به قابلیت ها و تجربیات تیم تولید کننده نرم افزار می پردازد

- ✓ **قابلیت تحلیلگر (Analyst Capability (ACAP** : تحلیلگران مورد نیاز ، طراحی در سطح بالا و جزیئات کار را انجام می دهند ویزگی های اصلی که در این رتبه بندی باید در نظر گرفته شود ، تجزیه و تحلیل ،توانایی و بازدهی ، نظم ودقت ، توانایی برقراری ارتباط و همکاری است ، درجه ۱۵ برای پایین و ۹۵ برای خیلی بالا در نظر گرفته می شود
- ✓ **قابلیت برنامه نویس (Programmer Capability (PCAP** : روند کنونی اهمیت زیادی به تحلیلگران توانا می دهد در نتیجه گرایش به سمت اهمیت بالاتری از توانایی برنامه نویس است ارزیابی قابلیت برنامه نویس به عنوان تیم است نه یک فرد و عواملی که باید در نظر گرفته شود می توان بازدهی ، نظم ودقت ، و توانایی برقراری ارتباط و همکاری است .

- ✓ **سابقه نرم افزار Application Experience (AEXP)** : این امتیاز بسیار وابسته به سطح تجربه تیم توسعه دهنده سیتم نرم افزاری است . امتیاز بسیار پایین برای تجربه استفاده می شود که کمتر از ۲ ماه است و امتیاز بسیار بالا برای تجربه ۶ سال یا بیشتر است
- ✓ **سابقه پلت فرم Platform Experience (PEXP)** : این معیار اهمیت شناخت و درک استفاده بسیاری از پلت فرم های قدرتمند ر ا می دهد که شامل رابط گرافیکی ، دیتابیس ، شبکه ها ، و قابلیت میان توزیع است .
- ✓ **زبان و تجربه ابزار Language and Tool Experience (LTEX)** : این معیار سطح زبان برنامه نویس را مشخص می کند
- ✓ **پیوستگی کارکنان Personnel Continuity (PCON)** : مقیاس رتبه بندی PCON از نظر گردش مالی سالانه پرسنل است بسیار کم ۴۸٪ و بسیار زیاد ۳٪ در سال است

۴- فاکتور های پروژه که شرایط خاص پروژه تولید نرم افزار را مورد توجه قرار می دهد

- ✓ **استفاده از ابزار نرم افزار Use of Software Tools (TOOL)** : ابزار های نرم افزار به طور قابل توجهی از پروژه های ۱۹۷۰ به بعد بهبود یافته محدوده رتبه بسیار کم ویرایش ، و محدوده بسیار زیاد مدیریت چرخه عمر یکپارچه است
- ✓ **توسعه چند سایت Multisite Development (SITE)** :
- ✓ **زمانبندی مورد نیاز برنامه Required Development Schedule (SCED)** : این معیار محدودیت زمانی بر تیم توسعه نرم افزار را مشخص می کند ۷۴٪ برای زمانبندی فشرده و ۱۶۰٪ برای زمانبندی با دقت بیشتر و بالاتر است .

ضریب تعدیل برای EarlyDesign طبق جدول زیر می باشد :

Early Design Cost Driver	Counterpart Combined Post-Architecture Cost Drivers
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PERS	ACAP, PCAP, PCON
PREX	AEXP, PEXP, LTEX
FCIL	TOOL, SITE
SCED	SCED

برای محاسبه ضرایب تعدیل می توان از جداول زیر استفاده کرد :

Cost Driver	Rating					
	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.75	0.88	1.00	1.15	1.39	
DATA		0.93	1.00	1.09	1.19	
CPLX	0.75	0.88	1.00	1.15	1.30	1.66
RUSE		0.91	1.00	1.14	1.29	1.49
DOCU	0.89	0.95	1.00	1.06	1.13	
TIME			1.00	1.11	1.31	1.67
STOR			1.00	1.06	1.21	1.57
PVOL		0.87	1.00	1.15	1.30	
ACAP	1.50	1.22	1.00	0.83	0.67	
PCAP	1.37	1.16	1.00	0.87	0.74	
PCON	1.24	1.10	1.00	0.92	0.84	
AEXP	1.22	1.10	1.00	0.89	0.81	
PEXP	1.25	1.12	1.00	0.88	0.81	
LTEX	1.22	1.10	1.00	0.91	0.84	
TOOL	1.24	1.12	1.00	0.86	0.72	
SITE	1.25	1.10	1.00	0.92	0.84	0.78
SCED	1.29	1.10	1.00	1.00	1.00	

W(i)	Very Low	Low	Nominal	High	Very High	Extra High
Precedentedness	4.05	3.24	2.43	1.62	0.81	0.00
Development Flexibility	6.07	4.86	3.64	2.43	1.21	0.00
Architecture / Risk Resolution	4.22	3.38	2.53	1.69	0.84	0.00
Team Cohesion	4.94	3.95	2.97	1.98	0.99	0.00
Process Maturity	4.54	3.64	2.73	1.82	0.91	0.00

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 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:

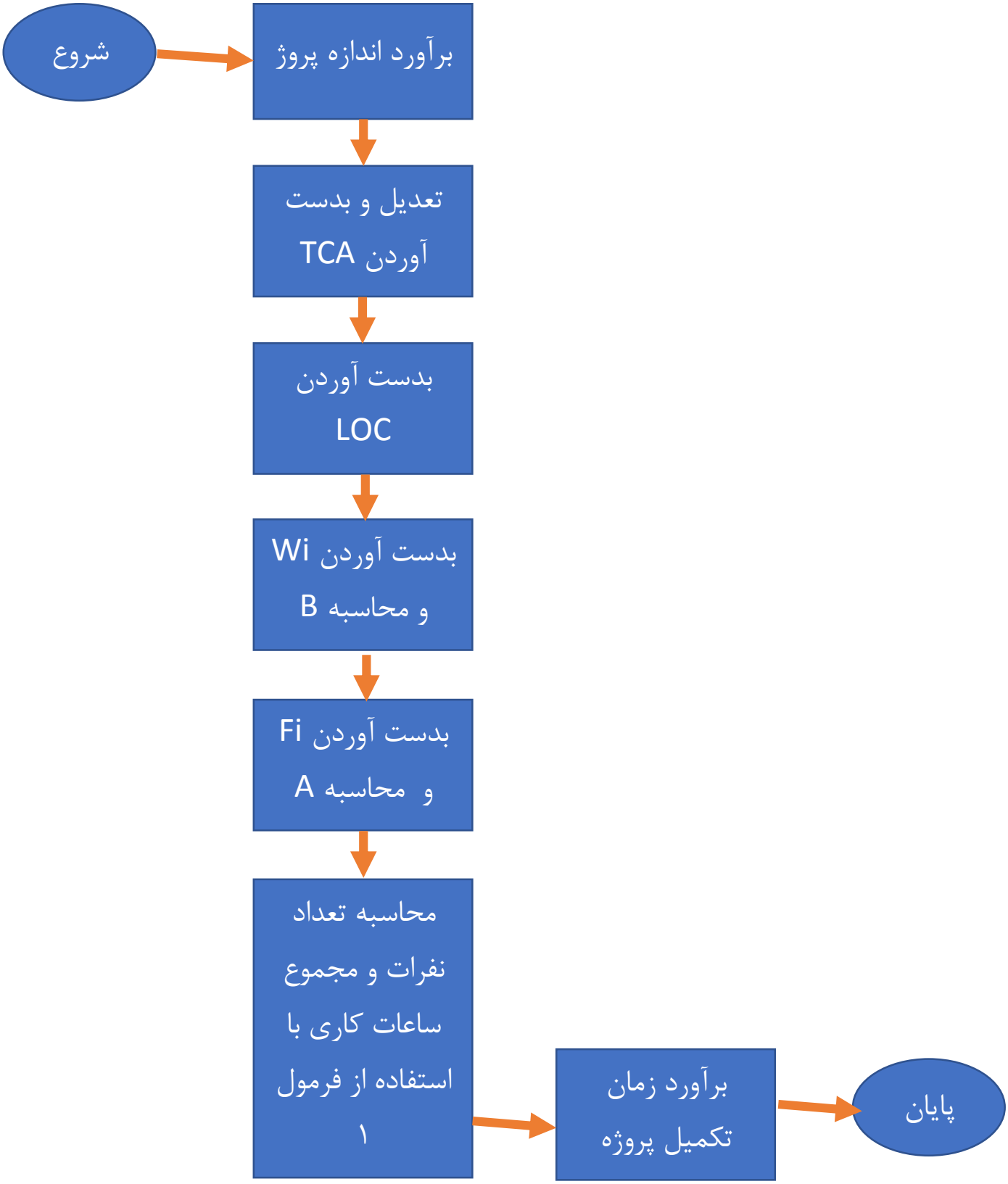
(فرمول ۴)

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

TDEV مدت زمان برآوردی بر حسب ماه از زمان شناسایی نیازمندی های سیستم تا هنگامی که تائید می شود که نرم افز ار 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.

❖ ۴- پروژه برآورد دبیرخانه دانشگاه

مراحل اجرای برآورد :



۴-۱ برآورد اندازه پروژه

در این مرحله با استفاده از روش $\mathcal{FP}$ اندازه پروژه ر اتخمین می زنیم .

۴-۲ تعیین الکانات و درجه بندی

External Input (EI)		
۱	فرم لاگین	$\mathcal{L}$
۲	صفحه ثبت پرسنل	$\mathcal{A}$
۳	انتخاب عکس	$\mathcal{A}$
۴	ثبت حکم	$\mathcal{A}$
۵	ثبت مزایای ثابت	$\mathcal{L}$
۶	ثبت اطلاعات پایه	$\mathcal{L}$
۷	ثبت یادداشت	$\mathcal{L}$

External Output (EO)		
۱	لیست احکام	$\mathcal{L}$
۲	گزارش پرسنل	$\mathcal{L}$
۳	چاپ حکم	$\mathcal{A}$

External Queries (EQ)		
۱	ویرایش / جستجو / حذف پرسنل	$\mathcal{A}$
۲	ویرایش / جستجو / حذف حکم	$\mathcal{A}$
۳	ویرایش / جستجو / حذف یادداشت	$\mathcal{L}$

Internal Logic File (ILF)		
۱	جدول پرسنل	£
۲	جدول احکام	£
۳	جدول ورود	£
۴	جدول یادداشت	£

External Interface File (EIF)		
۱	عکس و پایگاه داده که وارد می شوند	£
۲	عکس و پایگاه داده ای که خارج می شوند	£

❖ ۳-۴ محاسبه الکانات و بدست آوردن FP تعدیل نشده

----------	Low	Average	High	مجموع
EI	۳ * ۴	۴ * ۳	۶ * ۰	۲۴
EO	۴ * ۲	۵ * ۱	۷ * ۰	۱۳
EQ	۳ * ۱	۴ * ۲	۶ * ۰	۱۱
ILF	۷ * ۴	۱۰ * ۰	۱۵ * ۰	۲۸
EIF	۵ * ۲	۷ * ۰	۱۰ * ۰	۱۰
				۸۶

۴-۴ بدست آوردن مجموع درجه تاثیر (TDI)

ردیف	پرسش	نمره
۱	آیا سیستم به پشتیبانی و بازیابی قابل اطمینان نیاز دارد ؟	۱
۲	آیا ارتباطات داده ای مورد نیاز است ؟	۰
۳	آیا عملیات پردازشی توزیع شده وجود دارد ؟	۰
۴	آیا کاربی اهمیت دارد ؟	۲
۵	آیا سیستم در یک محیط عملیاتی موجود و پر کاربرد اجرا می شود ؟	۱
۶	آیا سیستم به مدل داده های آنلاین نیاز دارد ؟	۰
۷	آیا مدخل داده های آنلاین نیاز به ساخت تراکنش ورودی روی عملیات یا صفحات نمایش چند گانه دارد ؟	۰
۸	آیا فایل های اصلی به صورت آنلاین به هنگام می شوند ؟	۰
۹	آیا ورودی ها ، خروجی ها ، فایل ها و درخواست ها پیچیده اند ؟	۰
۱۰	آیا پردازش داخلی پیچیده است ؟	۰
۱۱	آیا کد طوری طراحی شده است که دوباره قابل استفاده باشد ؟	۲
۱۲	تبدیل و نصب در طراحی لحاظ شده است ؟	۱
۱۳	آیا سیستم برای نصب چندگانه در سازمان های متفاوت طراحی شده است ؟	۲
۱۴	آیا برنامه کاربردی طوری طراحی شده است که تغییرات را تسهیل کند و به آسانی توسط کاربر استفاده شود ؟	۳
		۱۲

۴-۵ بدست آوردن تکنیک پیچیدگی تعدیل شده (TCA)

TCA = 0.065 + 0.01 * 12

TCA=0.77

۴-۶ محاسبه FP

حال با داشتن TCA و UFP می توان FP را بدست آورد :

FP = 0.77 * 86

FP = 66.22

۴-۷ محاسبه تعداد خط کد (LOC)

زبان برنامه نویسی که این برنامه با آن نوشته خواهد شد زبان C# خواهد بود ، و از این نظر که زبان C# نزدیک به زبان جاوا می باشد ، در اینجا ضریب تبدیل را ۵۳ در نظر می گیریم .

LOC=66.22*53

LOC=3509

۵- تخمین هزینه پروژه

در این مرحله با استفاده از روش cocomo هزینه توسعه نرم افزار را محاسبه می نمایم .

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

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

(فرمول ۲) $$B = 0.91 + 0.01\sum W_i$$

(فرمول ۳) $$A = 2.94 \times \prod EM_i$$

❖ ۵-۱ تعیین مقدار B

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

فاکتور تعدیل	مقدار ضریب	
سابقه اجرایی (PREC)	L	3.24
قابلیت انعطاف (FLEX)	L	4.86
ریسک معماری نرم افزار (RESL)	VL	4.22
همکاری تیمی (TEAM)	L	3.95
بلوغ فرآیندهای تولید (PMAT)	L	3.64
		19.91

B = 0.91 + 0.01*19.91

B = 1.1091

❖ ۵-۲ تعیین مقدار A در مدل EarlyDesign

فاکتور تعدیل	مقدار ضریب	
RCPX	پایین	0.91
REUSE	پایین	0.91
PDIF	معمولی	0.95
PERS	خیلی بالا	0.63
PREX	خیلی بالا	0.74
FCIL	خیلی خیلی بالا	0.62
SCED	معمولی	1
		0/671

A= 2.94 * 0.227

A= 0/667

۵-۳ مقدار نیروی انسانی و زمان لازم برای تولید نرم افزار (pm)

PM= 0/667 * 3.509 ^ 1.109
PM = 3

۵-۴ برآورد زمان تکمیل پروژه (TDEV)

TDEV= 3.67 * (PM ^ 0.28*f)*d
f= 0.2*(B-0.91)
d=%SCED / 100
****
f = 0.2 *(1.1091-0.91)
f = 0/04
****
d = 125 / 100
d = 1.25
****
TDEV = 3.67*(PM^0.28*0.04)*1.25
TDEV = 5

۵-۵ نتیجه

بر طبق نتایج بدست آمده ، نشان می دهد که نرم افزار دبیر خانه با توجه به شرایطی که جداول تعدیل در نظر گرفته شده حدود ۳ نفر – ماه نیروی انسانی و حدود ۵ ماه زمان لازم است . مدل **cocomo** ***II*** هر نفر – ماه را ۱۵۲ ساعت در نظر گرفته است . بنابراین برای تکمیل این پروژه ۴۵۶ ساعت فعالیت لازم است .