



دانشگاه آزاد اسلامی واحد مشهد

## برآورد هزینه نرم افزار Software Cost Estimation

استاد محترم: جناب آقای دکتر احمدی

تهیه کننده: هدی طاهری

کارشناسی ارشد مهندسی کامپیوتر – گرایش نرم افزار

**چکیده-** در روزهای اول انجام محاسبات ، هزینه نرم افزاری درصد کمی از کل هزینه سیستم کامپیوتری را شامل می شد.

ضریب خطا در تخمین های هزینه نرم افزار نسبتا کم بود. امروزه نرم افزار گران ترین عنصر تمام سیستم های کامپیوتری است. برای مثال ، در سیستم های متداول ، یک خطای عمده هزینه می تواند تفاوتی بین سود و ضرر باشد. در زمان وجود رقابت های فشرده و در حال افزایش ، توانایی تخمین دقیق تر یک فاکتور حیاتی برای بسیاری از گروه های IT است .

تخمین هزینه و فعالیت نرم افزار به عنوان یک علم شناخته می شود. متغیر های بسیاری مانند انسان ، تکنیک ، محیط ، سیاسی می توانند در هزینه تقریبی نرم افزار و کوشش های انجام شده برای توسعه آن تاثیر داشته باشند . به هر حال ، تخمین پروژه نرم افزار می تواند از یک هنر ناشناخته به یک سری مراحل سیستماتیک تبدیل شود که تخمین هایی را همراه با ریسک قابل قبول فراهم می نماید.

در این مقاله ابتدا به مفهوم تخمین و اهمیت وجود آن در پروسه مهندسی نرم افزار پرداخته می شود و در ادامه پارامترهای دخیل در طراحی نرم افزار ، انواع تخمین و **تکنیک های برآورد هزینه** مورد بررسی قرار خواهند گرفت و در انتها توجه خود را به مدل COCOMO معطوف خواهیم کرد .

**کلید واژه-** تخمین پروژه نرم افزاری - مدل برآورد - ریسک - نقاط عملکرد<sup>۱</sup> - نقاط شیء<sup>۲</sup> - زمان بندی پروژه - COCOMO

## مقدمه

می توان با رجوع به پروژه های قبلی و با مقایسه بخشهای مختلف آنها با نیازهای پروژه جدید تا حدی هزینه ها را مشخص کرد

لازمه تخمین دقیق ، تجربه خوب کاری است . لذا ، بناء پروژه بر اساس تخمینهای اولیه با ریسک همراه است . با افزایش ریسک عدم اطمینان از برنامه ریزی اولیه افزایش می یابد . با افزایش پیچیدگی و اندازه پروژه ها میزان عدم اطمینان بیشتر می شود .

عمل هزینه یابی با تعیین وظایف و مراحل انجام پروژه آغاز می گردد . برای تعیین وظایف باید مسئله را بر اساس اهداف آن افراز نمود . مراحل انجام پروژه را بر اساس فرایند و یا متدولوژی انتخابی و چهارچوب عملیاتی واحد نرم افراز مشخص می کنند . بر اساس این تفکیک و افراز است که می توان هزینه را پیش بینی کرد . در مفهوم تخمین ، فعالیت و زمان لازم برای کامل شدن هر فرایند و نیز هزینه نهایی مربوط به هر فرایند بررسی می شود.

قبل از تخمین هر گونه هزینه ای باید حوزه عملکرد پروژه مشخص شود و اندازه تخمینی آن تعیین گردد. معمولا ، حوزه عملکرد پروژه را می توان با یک پاراگراف بیان کرد . برای تخمین هزینه می توان عملیات سیستم را بر اساس وظایف ، یا بر اساس امور و یا براساس کلاسهای اشیاء مورد نیاز به واحدهای کوچکتری تجزیه نمود. سپس برای هر واحد می توان مقادیر LOC<sup>۲</sup> و FP<sup>۳</sup> را محاسبه کرد. قابلیت تولید و یا باروری را بر اساس Fp/pm یا Loc/pm تعیین نمود .

---

## ۱- Function Point ( FP)

### ۳- Line Of Code

### ۱- ۲- Object Point

چرا برآورد هزینه نرم افزار از اهمیت برخوردار است؟

هزینه بالای ساخت سیستمهای نرمافزاری ، مصیبتبار بودن افزایش بیش از حد هزینهها برای سازنده نرم افزار ، انجام پروژه های

نرمافزاری با زمان و هزینه از پیش تعیین شده ، تخمین اولیه به منظور تشخیص امکان پذیر بودن یک پروژه ، تخمین در طول فرایند

توسعه به منظور تخصیص مجدد منابع از جمله مواردی هستند که لزوم برآورد هزینه نرم افزار را مشخص می کنند.

مهندسين نرم افزار) سه پارامتر موثر هستند که بیشترین هزینه مربوط به هزینه عملکرد می باشد که مواردی مانند هزینه توسعه ، گرما ، روشنایی محل و شبکه و وسایل ارتباطی و امکانات رفاهی و نیز هزینه اجتماعی و هزینه حسابداران ، مدیران ، مستخدم و تکنیسینها را شامل می شود. امروزه با تکنولوژی

## قسمت ۱: پارامتر های موثر در محاسبه هزینه کلی طراحی یک نرم افزار

هزینه سخت افزار و نرم افزار به همراه پشتیبانی آن ، هزینه آموزش و مسافرت ، هزینه عملکرد(هزینه حقوق پرداختی به

های نقل و انتقال اطلاعات ، هزینه مربوط به آموزش و مسافرت به میزان قابل توجهی کاهش می یابد.

## قسمت ۲: تخمین بر چه اساسی انجام می گیرد؟

- تخمین بر اساس تعداد خطوط: یکی از معیارهای ارزیابی کمی نرم افزار ، اندازه آن است . اندازه نرم افزار را معمولاً با تعیین تعداد خطوط برنامه های مربوطه مشخص می کنند . مثلاً بر همین اساس، در سیستمی بر اساس معیارهای کمی، نکات زیر مطرح است:

- |      |  |
|------|--|
| i.   | تعداد متوسط خطا در هر هزار خط برنامه   |
| ii.  | تعداد متوسط نقص در هر هزار خط برنامه   |
| iii. | هزینه برای هر خط برنامه                |
| iv.  | تعداد متوسط صفحات در هر هزار خط برنامه |
| v.   | تعداد متوسط خطا در مقایسه با نفر ساعت  |
| vi.  | تعداد متوسط سطر در مقایسه با نفر ساعت  |
| vii. | هزینه در مقایسه با تعداد صفحات مستندات |

اکنون تعداد خطوط لازم کد برنامه برای انجام هر وظیفه را باید مشخص نمود . برای این منظور حداکثر ، حداقل و متوسط تعداد خطوط لازم را مشخص نموده و تعداد تخمینی خطوط لازم یا

در اصطلاح LOC را از رابطه زیر می توان مشخص کرد:

$$= \text{تعداد خطوط لازم}$$

$$6 / (\text{حداکثر خطوط} + 4 * \text{متوسط خطوط} + \text{حداقل خطوط})$$

### • تخمین بر اساس وظیفه :

میزان وظیفه مندی برای نرم افزارهای تجاری معمولاً براساس اطلاعات و داده ها در حوزه مساله محاسبه می شود.

### • تخمین براساس فرآیند نرم افزار:

کار تخمین پروژه های نرم افزاری را به تخمین وظایف در مراحل چرخه حیات یا مراحل فرایند تولید نرم افزار می توان افزاز نمود. میزان تلاش و کار لازم برای انجام هر فعالیت از وظایف تعیین شده معمولاً بر اساس کارایی نفر در ماه مشخص می شود. ممکن است هزینه افراد بر اساس نوع وظیفه محوله متفاوت باشد. معمولاً به تحلیلگران با سابقه دستمزدی بیشتر از برنامه نویسان تعلق می گیرد .

## قسمت ۳: تکنیک های برآورد هزینه

. بالا - پایین (Top-Down)

. پایین - بالا (Bottoms-Up)

. تاکید بر فایده (Pricing to Win)

. مدلسازی الگوریتمی هزینه

. داوری کارشناسان (Expert Judgement)

. بر اساس قیاس (By Analogy)

. قانون پارکینسون

. نقاط عملکرد (Points Function)

. نقاط اشیاء (Points Object)

### برآورد بالا به پایین

برآورد هزینه ، با در نظر گرفتن عملکرد محصول ، و چگونگی فراهم آمدن آن عملکرد توسط عملیات فرعی در حال بر همکنش ، انجام می شود. برآورد های هزینه بر مبنای عمل منطقی صورت می گیرد و نه بر اساس قطعاتی که آن عمل را پیاده سازی می کنند.

### برآورد پایین به بالا

هزینه هر قطعه برآورد می شود . همه ی این هزینه ها با هم جمع می شود تا هزینه نهایی برآورد گردد.

### برآورد تاکید بر فایده

هزینه نرم افزار بر اساس قدرت خرید مشتری برآورد می گردد. هزینه برآورد شده به بودجه مشتری بستگی دارد نه به عملکرد نرم افزار.

### . مدلسازی الگوریتمی هزینه

مدلی با استفاده از اطلاعات هزینه ای گذشته توسعه می یابد و برخی از معیار های نرم افزاری (معمولاً اندازه آن) را با هزینه پروژه مرتبط می سازد. آن معیار برآورد می شود و از روی مدل مذکور ، کار لازم پیش بینی می شود.

### . داوری کارشناسان

از یک یا چند کارشناس تکنیک های توسعه نرم افزار استفاده می شود و در باره دامنه کاربرد ، بحث و مشاوره می شود. هر یک از آن ها ، هزینه پروژه را برآورد می سازند و بر سر برآورد نهایی هزینه به توافق می رسند.

### . بر آورد بر اساس قیاس

این تکنیک هنگامی عملی است که پروژه های دیگری در همان دامنه کاربرد ، به پایان رسیده باشند. هزینه پروژه جدید ، از روی شباهت با پروژه های پایان یافته برآورد می شود.

## قانون پارکینسون

این قانون بیان می کند که کار در طول زمان معین توسعه می یابد. در تعیین هزینه نرم افزار، این بدان معناست که هزینه را منابع در دسترس تعیین می کنند تا ارزیابی عینی. اگر قرار است نرم افزاری در عرض ۱۲ ماه تحویل گردد و ۵ نفر در دسترس اند، کار مورد نیاز ۶۰ نفر-ماه برآورد می گردد.

## نقاط عملکرد

محاسبه نقاط عملکرد طبق رابطه زیر می باشد:

$$FP = UFC * [ 0.65 + 0.01 * \sum (F_i) ]$$

$$UFC = \sum (\text{number of elements of given type}) \times (\text{weight})$$

مقادیر تنظیم پیچیدگی  $F_i$  مبتنی بر پاسخ به ۱۴ پرسش زیر حاصل می شود:

SLOC / FP	زبان برنامه نویسی
۳۲۰	زبان اسمبلی
۱۲۸	C
۱۰۶	فرترن
۹۰	پاسکال
۶۴	++C
۵۳	۹۵ ادا
۳۲	ویزوال بیسیک
۱۲	SQL
۶۴	پرولوگ

## رابطه FP و LOC

## نقاط اشیاء

یک میزان نرم افزاری غیرمستقیم به حساب می آید و کاربرد اصلی آن در مدل Composition Application از CASE COCOMOII است. نشاندهنده استفاده از ابزارهای CASE برای ایجاد سریع نمونه اولیه

(aided software engineering computer)

می باشد و قابلیت استفاده مجدد از مؤلفه ها را فراهم می آورد. در دو مورد زیر برآورد نقطه شیئی نسبت بره نقطه تابعی مزیت دارد:

- به جزئیات پیاده سازی وابسته نیست.
- تخمین ضریب پیچیدگی خیلی ساده تر است.

$$NOP = [(100 - \%reuse) * \text{نقاط اشیاء}] / 100$$

که در آن تخمین یک درصد استفاده مجدد (Reuse) در نظر گرفته شده است. و معادله کار:

$$\text{Effort} = NOP / PROD$$

و PROD میزان ثمر بخشی است.

- 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؟

معادله کار در این مدل :

$$\text{Effort} = \text{NOP} / \text{PROD}$$

$$\text{NOP} = [(100 - \% \text{reuse}) * \text{اشياء نقاط}] / 100$$

### مدل Early Design

این روش پس از تثبیت خواسته‌ها به کار می‌رود و مبتنی بر تخمین اندازه به روش FP است .  
معادله کار :

$$\text{Effort} = A * \text{Size}^B * M$$

ضریب A برابر ۲,۹۴ است.

توان B افزایش میزان فعالیت را در حالی که اندازه پروژه بزرگ می‌شود نشان می‌دهد که برای انواع سیستم‌های مختلف یکسان نمی‌باشد. می‌تواند بین ۱,۱ و ۱,۲۴ متغیر باشد که بستگی به میزان تازگی و جدید بودن پروژه ، انعطاف پذیری توسعه ، میزان ریسک موجود در پروسه ، نحوه ارتباط افراد تیم با یکدیگر و میزان بلوغ سازمان دارد.  
ضریب M وابسته به ویژگی‌هایی است که در مدل طراحی اولیه استفاده می‌شوند:

$$M = \text{PERS} * \text{RCPX} * \text{RUSE} * \text{PDIF} * \text{PREX} * \text{FCIL} * \text{SCED}$$

و به ترتیب راست به چپ ، سختی زمان بندی ، امکانات موجود ، تجربیات شخصی ، سختی سیستم مورد طراحی ، استفاده مجدد مورد نیاز ، میزان اعتماد و پیچیدگی سیستم و توانایی‌های شخصی می‌باشد.

### مدل Post Architecture

کاربرد این روش در اثنای ساخت نرم افزار بوده و معادله کار در این مدل مشابه رابطه کار در مدل Early Design است .

تفاوت این روش با روش Early Design :

- تخمین اندازه در این مدل باید دقیق‌تر

از مدل Early Design باشد.

- خصوصیات مورد استفاده در ضریب M در

این مدل بیشتر از خصوصیات مورد استفاده در مدل

Early Design می‌باشد.

### مدل Reused Code

دو نوع کد قابل استفاده مجدد در COCOMO II

به شرح زیر است:

- کد جعبه سیاه : عدم نیاز به شناخت یا اعمال تغییر

هر تکنیک ، دارای مزیتها و معایبی است. برای پروژه‌های بزرگ ، چندین تکنیک برآورد هزینه را باید به طور موازی به کار برد و نتایج آن‌ها را با هم مقایسه کرد.

اگر هزینه‌هایی که با این تکنیک‌ها پیش بینی شود دارای تفاوت اساسی باشند ، نتیجه می‌شود که اطلاعات کافی برای تعیین هزینه در دسترس نبوده است . باید اطلاعات بیشتری در نظر گرفته شود و فرایند تعیین هزینه تکرار گردد. این فرایند باید آن قدر ادامه یابد که برآوردها همگرا شوند.

### قسمت ۴: COCOMO

COCOMO مخفف واژه Constructive Cost Model

است . مدل COCOMO یک مدل تجربی است که از جمع‌آوری داده‌های پروژه‌های نرم افزاری مختلف بدست آمده است. مستندات این مدل به صورت عمومی موجود است و در بسیاری از سازمانها این مدل مورد استفاده قرار می‌گیرد.

COCOMO81 (ارائه شده توسط بوهم در سال ۱۳۸۱) اولین نسخه از این مدل بوده و فرض می‌کند که نرم افزار تحت طراحی بر طبق مدل آبشاری تولید می‌شود و با استفاده از زبان‌های ساخت یافته‌ای نظیر C یا FORTRAN پیاده‌سازی شده است.

مدل COCOMO به شکل بنیادی (ساده) ، متوسط و مفصل وجود دارد. این مدل هنگام برآورد دقیق هزینه ، ویژگی‌های پروژه ، محصول ، سخت افزار و کارمندان را بحساب می‌آورد.  
مدل COCOMO II (ارائه شده توسط بوهم در سال ۲۰۰۰) یک مدل حلزونی را پشتیبانی می‌کند و شامل زیرمدل‌های زیر است:

Application Composition –

Early Design –

Post Architecture –

Reused Code –

### مدل Application Composition

مختص سیستم‌های تشکیل شده از مولفه‌های قابل استفاده مجدد و مبتنی بر تخمین نقاط اشياء (OP) است.

- کد جعبه سفید : لزوم تغییر کد به منظور تلفیق با کد جدید کار توسعه برای کد جعبه سفید صفر در نظر گرفته می شود.

تعداد خطوط هم ارز در کد منبع جدید :

$$E\ SLOC = ASLOC * AAM$$

$$AAM = AA + AAF ( SU * UNFM)$$

ASLOC : تعداد خطوط موجود در اجزایی که باید تغییر داده شوند. ASLOC بر اساس در صد کدهایی که به طور اتوماتیک تولید می شود کاهش می یابد.

AAM ضریب تکامل نامیده می شود و در برگیرنده میزان

تلاش لازم برای انجام استفاده مجدد می باشد. به طور ساده AAM مجموع سه قلم زیر است :

- میزان تغییرات در اجزا (AAF)

- شناخت اجزا (SU)

- ضریب برآورد (AA)

میزان تغییرات در اجزا ، نشان دهنده هزینه مربوط به انجام تغییرات در کد مورد استفاده مجدد است و شامل طراحی کد و انجام تغییرات جهت استفاده می باشد.

شناخت اجزا نشان دهنده هزینه مربوط به شناخت کدهایی است که مجددا استفاده می شوند و اینکه میزان آشنایی مهندسان با آن ها چقدر است. SU در محدوده ۵۰ برای کدهای پیچیده و غیر ساختاری و ۱۰ برای کدهای ساده و شیئی گرا متغیر می باشد.

ضریب برآورد نشان دهنده هزینه مربوط به انجام تصمیمات می باشد . AA در محدوده ۰ تا ۸ متغیر است که بستگی به میزان بررسی لازم دارد.

مدل COCOMO شامل یک فرمول برای تخمین زمان

مورد نیاز برای کامل شدن پروژه است (TDEV) و این فرمول برای تمام مراحل COCOMO یکسان می باشد.

$$TDEV = 3 * Effort^{SE}$$

$$SE = 0.33 + 0.2 * (B - 1.01)$$

توان B در مدل نمونه سازی ۱ است.

برنامه ریز پروژه ، باید سه چیز را قبل از شروع پروژه تخمین بزند: چه مدت طول می کشد ، چه میزان فعالیت لازم است ، و چند نفر نیاز می باشند. علاوه بر آن ، برنامه ریز باید منابع مورد نیاز (سخت افزار و نرم افزار ) را به همراه ریسک آن ها پیش بینی نماید. با مقایسه و کنار هم قرار دادن تخمین های انجام شده با استفاده از روش های متفاوت ، برنامه ریز می تواند امکان ارتقای دقت تخمین را فراهم سازد.

واقعیت برآورد این است که ، هزینه های برآورد شده اغلب برای بدست آوردن یک قرارداد ارزیابی میی شوند و عملکرد سیستم طوری تنظیم می شود تا با مقادیر برآورد شده جور شود.

- Hussein, A. (2001). Introduction to Software Process Management. University of Calgary, Calgary, Canada. <http://sern.ucalgary.ca/courses/SENG/621/W01/intro.ppt>.
- Londeix, B. (1987). Cost Estimation for Software Development. Addison-Wesley Publishing Company, New York, NY.
- Pressman, R.S. (2001). Software Engineering: A Practitioner's Approach. McGraw-Hill Higher Education, New York, NY.
- Vigder, M. R. and Kark, A. W. (1994). Software Cost Estimation and Control. Software Engineering Institute for Information Technology. <http://www.sel.iit.nrc.ca/seldocs/cpdocs/NRC37116.pdf>.
- Wu, L. (1997). The comparison of the Software Cost Estimating Methods. University of Calgary, Calgary, Canada. [http://sern.ucalgary.ca/courses/seng/621/W97/wul/seng621\\_11.html](http://sern.ucalgary.ca/courses/seng/621/W97/wul/seng621_11.html).
- Basic COCOMO Software Cost Model. <http://www.jsc.nasa.gov/bu2/COCOMO.html>.
- COCOMO 2, Softstar Systems. <http://www.softstarsystems.com/cocomo2.htm>.

Answers to Frequently Asked Questions, Softstar Systems. <http://www.softstarsystems.com/faq.htm>