

# مدیران حوزه نرم افزار



تهیه کننده: شبنم تدین

## بهبود مستمر فرآیندهای

## نرم افزاری را جدی بگیرید

برای حرکت به سطوح بالاتر، باید دسته‌ای از اهداف هر سطح مطمئن شد این اطمینان جز با اندازه‌گیری شاخصهای کاربردی هر سطح، ایجاد نمی‌شود. این مقاله به معرفی اجمالی مدل بهبود فرآیند نرم افزار و واحد اندازه‌گیری نرم افزار می‌پردازد.

استانداردها بر بهبود مستمر فرآیندها و فعالیتهای سازمانهای نرم افزاری تاکید دارند. بهبود مستمر فرآیند تدریجی و آرام تحول‌پذیری است. یکی از مدل‌های معتبر در زمینه بهبود مستمر فرآیندهای نرم افزاری، مدل CMM است. در این مدل

بی‌شک، امروزه در فضای رقابتی شرکتهای نرم افزاری، ارتقاء کیفیت و بهره‌وری راز ماندگاری و بقا است. برای نیل به دو مقصود فوق، تلاشهای وسیعی صورت گرفته و استانداردهایی در سطح بین‌المللی تدوین و ارائه شده‌اند. این

## چالشهای پیشاوری شرکتهای نرم افزاری

با توسعه فناوری اطلاعات، و افزایش بی سابقه شرکتهای نرم افزاری کوچک و متوسط، رقابت بین این شرکتهای به طرز چشمگیری افزایش یافته است. از اینرو، توجه به خواسته ها و ارضاء نیازهای مشتری و حتی در مواردی فرای درک و خواست مشتری عمل کردن ضامن بقاء و رشد شرکتهای نرم افزاری است. اما در حال حاضر، اکثر شرکتهای نرم افزاری کوچک و جدید و بعضی شرکتهای متوسط با مشکلاتی این چنینی روبرو هستند:

\* عدم شناسایی کامل نیازمندیهای کاربر  
\* تخمین نادرست زمانبندی پروژه  
\* تخمین نادرست اندازه و هزینه های پروژه

\* عدم تخصیص منابع کافی ( اعم از انسانی، مالی، تجهیزاتی ) جهت اجرا پروژه

\* پایین بودن سطح بهره وری  
\* نبود اطلاعات دقیق و صحیح در هنگام مباحثات عقد قرارداد

همچنین شرکتهایی که قصد دارند فعالیتهای تولید نرم افزار خود را برونسپاری کنند نیز با مشکلاتی نظیر مدیریت و کنترل نامطلوب تهیه کننده نرم افزار مواجه هستند.

بهره گیری از روشها و استانداردهای مشخص، و تدوین شاخصهای سنجش به

کاهش مشکلات فوق کمک می کنند. به منظور افزایش بهره وری، تعیین صحیح زمانبندی و صورتبندی پروژه، و افزایش توانایی مدیریت و کنترل پروژه، استفاده از روشهای بهبود فرایندهای نرم افزاری و تدوین شاخصهای سنجش اجتناب ناپذیر به نظر می رسند. اما این کار چگونه باید آغاز کرد؟

### مدل CMM

سازمانها بعد از چند دهه ناکامی در مورد استفاده از متدولوژیها و تکنولوژیهای بهبود کیفیت و بهره وری در حوزه نرم افزار، به این نتیجه رسیده اند که مشکل اساسی، عدم وجود توانایی لازم در زمینه مدیریت فرایندهای نرم افزار است.

از همین رو، در نوامبر ۱۹۸۶، موسسه مهندسی نرم افزار (SEI) با مشارکت شرکت میترا اقدام به تدوین چارچوب بلوغ فرآیند نمود. هدف از این کار، بهبود بخشیدن فرایندهای نرم افزار بود. در نتیجه در سپتامبر ۱۹۸۷، SEI توانست دو روش با نامهای ارزیابی فرآیند نرم افزار، و ارزیابی قابلیت نرم افزار و یک پرسشنامه بلوغ تدوین و ارائه کند. سالها بعد، SEI به دلیل محک زدن روشهای فوق و کسب تجربه های گسترده در حوزه صنعت و سازمانهای دولتی، مدل بلوغ توانایی (CMM) را برای سازمانهای نرم افزاری عرضه کرد. CMM به ارائه مجموعه ای از اقدامات توصیه شده می پردازد، که بکار بستن این اقدامات در مجموع منجر به افزایش توانایی فرآیندهای نرم افزار می شود.

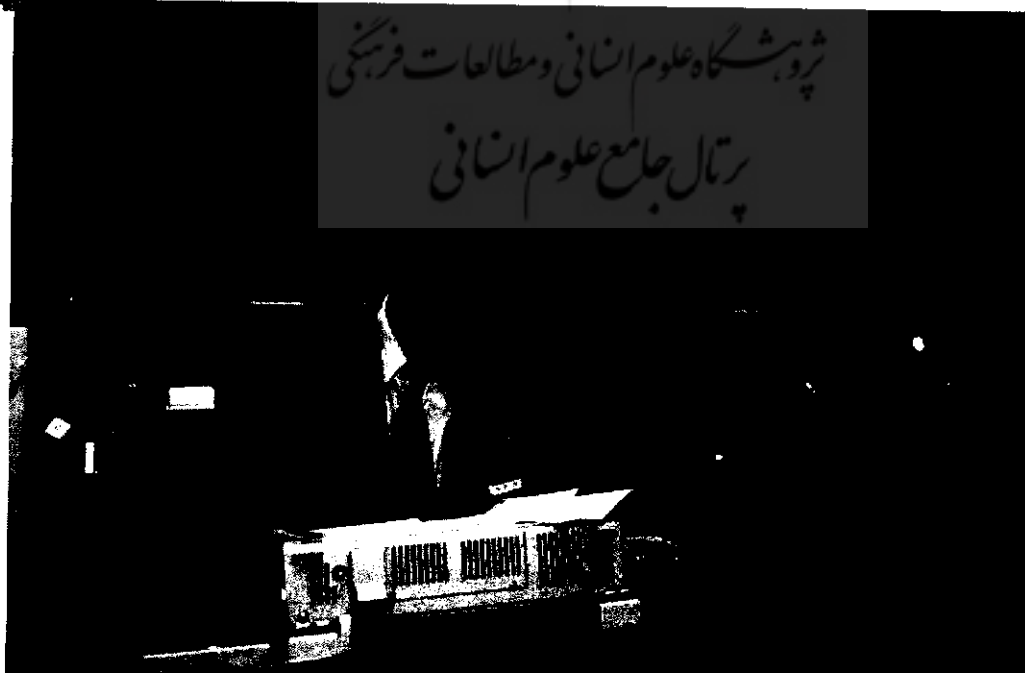
CMM برای انتخاب استراتژیهای بهبود فرآیند طراحی شده است. این انتخابات بر مبنای دو فاکتور اساسی، « میزان بلوغ فرآیند جاری » و « شناسایی موضوعات کوچک اما بسیار مهم » در زمینه بهبود فرآیند و کیفیت نرم افزار امکانپذیر می گردد. پس از انتخابات استراتژی بهبود، سازمان باید یک سری اقدامات محدود را اجراء کند تا بتواند فرآیندهای نرم افزاری خود را بهبود بخشد.

واژه های کلیدی:

پیش از تشریح سطوح مدل CMM، بهتر است واژه های کلیدی بحث به صورت مختصر توضیح داده شوند. توانایی فرآیند نرم افزار: به نتایج مورد انتظار از یک فرآیند نرم افزاری خاص اشاره دارد.

Software Engineering Institute  
Mitre corporation  
Software process assessment  
Softwarc Capability evaluation  
Capability maturity model  
Software process capability

عملکرد فرآیند نرم افزار: نشانگر نتایج واقعی حاصل از فرآیند نرم افزار است. بلوغ فرآیند نرم افزار: نشانگر درجه ای است که یک فرآیند به روشنی تعریف و مدیریت شده، مورد سنجش قرار گرفته، کنترل شده، و موثر واقع شده است. بلوغ تلویحا به پتانسیل رشد توانایی ها اشاره



\* عدم وجود محیط پایدار جهت توسعه و نگهداری نرم افزار،  
\* تخطی از رویه های برنامه ریزی شده،

\* وابستگی به مدیر توانا و کار آفرین،  
\* به اتمام رساندن پروژه ها حتی به قیمت افزایش زیاد هزینه و طولانی شدن زمان بندی پروژه  
\* وابستگی سازمان به کارکنان شایسته و قوی،

\* وجود دوباره کاری زیاد،  
\* ناامید شدن مشتریان، و  
\* بروز شگفتی های فنی زیاد اشاره کرد.

سطح دوم: سطح قابل تکرار ویژگی اصلی این سطح قابلیت تکرار برنامه ها است. از جمله دیگر ویژگیهای قابل ملاحظه سطح دوم می توان به موارد زیر اشاره کرد:

\* تدوین خط مشی جهت مدیریت پروژه،  
\* تدوین رویه هایی جهت به اجرا درآوردن خط مشی ها،

\* وجود کنترل های مدیریت نرم افزار،  
\* تهیه و تدوین استانداردهای جهت پروژه های نرم افزاری و پیروی از آنها،  
\* احتمال متفاوت بودن فرآیندها در پروژه های مختلف،

\* تضمین ساده محصول و فرآیند،  
\* مدیریت فروشنده،

\* برنامه ریزی و مدیریت پروژه های جدید بر اساس تجربیات پروژه های مشابه قبلی صورت می گیرد،

\* تعهدات پروژه فعلی بر مبنای عملکرد و نتایج پروژه های قبلی تنظیم می شوند،

\* خط مشی، راهنمای پروژه است.

سطح سوم: سطح تعریف شده ویژگی این سطح استانداردها هستند. از خصوصیات این سطح می توان به موارد زیر اشاره کرد:

\* تدوین فرآیندی استاندارد جهت توسعه و نگهداری نرم افزار،

\* تلفیق مهندسی نرم افزار با فرآیندهای مدیریت،

\* آموزش مدیران و کارکنان برای آشنایی آنها با دانش و مهارت های لازم،

\* وجود گروه های مسوول در قبال فرآیندهای نرم افزاری سازمان،

\* تعریف شاخص آمادگی، ورودی ها، خروجیها، استانداردها، رویه های اجراء



می کند.

### سطوح پنج گانه بلوغ فرآیندهای نرم افزار

بهبود مستمر بر گامهای کوچک و تدریجی استوار است. ساختار مرحله ۱ CMM بر مبنای اصول کیفیت محصول بنا نهاده شده است CMM. چارچوبی ایجاد کند که در آن پنج سطح تعبیه شده و بهبود مستمر و تدریجی فرآیندها بدین طریق حاصل می شود. این سطوح، سنجش میزان بلوغ فرآیند نرم افزار و ارزیابی توانایی فرآیند نرم افزار را میسر می سازند. همچنین به اولویت بندی اقدامات بهبود کمک می کنند. در هر سطح، مجموعه ای از اهداف در نظر گرفته شده اند، که نیل به این اهداف موجب تثبیت شدن یک جزء مهم از فرآیند نرم افزار شده و این امر به

افزایش قابلیت فرآیندهای سازمان منجر می شود. این سطوح پنج گانه عبارتند از:

\* سطح اولیه

\* سطح قابل تکرار

\* سطح تعریف شده

\* سطح مدیریت شده

\* سطح بهینه

### ویژگیهای رفتاری سطوح بلوغ

در مدل CMM، سطح اول مبنای مقایسه قلمداد شده و با اجراء یکسری اقدامات، سطوح دوم تا پنجم قابل حصول می شوند.

### سطح اول: سطح اولیه

ویژگیهای عمده این سطح، توانایی کارکنان است.

از جمله دیگر ویژگیهای سطح اول می توان به:

کار، مکانیزمهای تصدیق،

\* تدوین شاخصهایی جهت کنترل پروژه، برنامه زمانبندی، هزینه ها، وظایف و وجود شمایی از کیفیت نرم افزار.

### سطح چهارم: سطح مدیریت شده

ویژگی مهم این سطح، قابلیت اندازگی و پیش بینی است. مشخصه های دیگر سطح چهارم عبارتند از:

\* تدوین اهداف کمی و کیفی برای محصولات و فرآیندها،

\* سنجش بهره وری و کیفیت در قالب یک برنامه اندازه گیری،

\* ایجاد بانک اطلاعاتی جهت جمع آوری و تحلیل داده ها،

\* شناسایی و بیان مشکلات خاص و مسائل استثنایی،

\* انجام اقدام اصلاحی در صورت ضرورت در هنگام بروز خطا.

### سطح پنجم: سطح بهینه

ویژگی مهم این سطح بهبود مستمر است. از دیگر خصوصیات مهم سطح پنجم می توان این موارد را نام برد:

\* تمرکز سازمان بر بهبود مستمر،

\* وجود وسایلی جهت شناسایی نقاط قوت و ضعف سازمان،

\* هدف، جلوگیری از رخداد خطا و اشتباه است.

\* اجراء تحلیل هزینه منفعت جهت تحلیل فن آوریهای جدید و تغییرات پیشنهادی،

\* تحلیل خطاها به منظور یافتن دلایل بروز خطا،

\* ارزیابی فرآیندها، به منظور جلوگیری از رخدادهای شناخته شده،

\* انتقال بهترین نوآوری ها در زمینه مهندسی نرم افزار به تمام سازمان،

\* اصلاح، تغییر و بهبود فرآیندها در صورت وجود اتلاف،

\* بهبود به دو طریق حاصل می شود:

\* ایجاد پیشرفتهای تدریجی در فرآیندهای موجود

\* ایجاد نوآوری با استفاده از فن آوریها و متدهای نوین

### پرش از بعضی سطوح بلوغ

علیرغم این که می توان بعضی از سطوح را جا زده و به سطوح بالاتر

پرسید، اما به سازمانها توصیه می شود، به این کار دست نزنند زیرا در هر سطح

پی ریزی ملزومات زیربنایی سطح بالاتر صورت می گیرد.

### کاربردهای CMM:

مدل CMM مسیر بهبود خاصی را برای

سازمانهای نرم افزاری فراهم آورده و به آنها توصیه می کند در صورت تمایل به افزایش توانایی فرآیندهای نرم افزاری، از مدل CMM استفاده کنند.

### مدل CMM چهار کاربرد عمده دارد:

\* تیم ارزیابی می تواند برای شناسایی نقاط قوت و ضعف سازمان از CMM استفاده کند.

\* تیم های ارزیابی می تواند برای شناسایی مخاطرات انتخاب پیمانکاران

مختلف و همچنین نظارت بر قراردادها از CMM استفاده کنید.

\* مدیری ارشد برای درک اقدامات ضروری جهت بهبود فرآیند نرم افزاری

سازمان از CMM استفاده می کند.

\* کارکنان فنی و گروههای بهبود فرآیند، از CMM به عنوان راهنمای جهت

تعریف و بهبود فرآیند نرم افزار در سازمان بهره می گیرند.

زمینه های کلیدی فرآیند

زمینه های کلیدی فرآیند (KPA) به گروهی از اقدامات مرتبط اطلاق می شود

که در صورت اجراء شدن تمامی آنها، مجموعه ای از اهداف در زمینه گسترش توانایی فرآیند، تحقق می یابند.

بدین ترتیب، برای رسیدن به هر سطح بلوغ، زمینه های کلیدی فرآیند آن سطح

باید ارضاء شوند.

از آنجائی که سطح اول، پایین ترین سطح می باشد، برای حرکت به سطوح بالاتر، KPAها عبارتند از:

برای رسیدن به سطح دوم

\* مدیریت نیازمندیها،

\* برنامه ریزی پروژه نرم افزار،

\* پیگیری و نظارت بر پروژه نرم افزار،

\* مدیریت فرارادهای نرم افزار (یعنی

انتخاب پیمانکار شایسته و مدیریت موثر وی)،

\* تضمین کیفیت نرم افزار،

\* مدیریت پیکره بندی نرم افزار

(ایجاد و حفظ یکپارچگی محصول در

طول چرخه حیات پروژه نرم افزاری).

برای رسیدن به سطح سوم

\* تمرکز بر فرآیند سازمان

\* تعریف فرآیند سازمان

\* ادامه برنامه های آموزشی به منظور

توسعه مهارتها و دانش افراد درگیر پروژه

\* مدیریت یکپارچه نرم افزار

\* مهندسی محصول نرم افزاری

\* ایجاد هماهنگی بین گروههای کاری

\* بازنگری نظیر به نظیر

برای رسیدن به سطح چهارم

\* مدیریت کمی فرآیند

\* مدیریت کیفی نرم افزار

برای رسیدن به سطح پنجم

\* جلوگیری از رخداد خطا

\* مدیریت تغییر تکنولوژی

\* مدیریت تغییر فرآیند

### محدودیتهای CMM:

CMM به تمامی موضوعات مهم برای

موفقیت آمیز شدن پروژه ها نمی پردازد.

فرضیه CMM در حال حاضر به موضوعاتی نظیر حمایت از فناوریهای نرم افزاری

خاص، یا ارائه طریق در زمینه نحوه انتخاب، استخدام، انگیزختن و حفظ

نیروهای شایسته نمی پردازد. اگر چه این موضوعات برای موفقیت پروژه

حائز اهمیت هستند، اما مدل کنونی CMM پاسخی برای این موارد ندارد.

در نهایت باید گفت CMM مدل کیفی نیست. اما ابزاری مفید برای جهت دادن به

تلاشها بهبود فرآیند نرم افزار است CMM.

ساختاری مفهومی جهت بهبود مدیریت و توسعه محصولات نرم افزاری به شکل

نظاممند و سازگار است.

اینک این سوال مطرح شود که چگونه می توان از دستیابی به اهداف هر سطح

و بهبود فرآیندهای نرم افزاری اطمینان حاصل کرد؟ این کار جز از طریق سنجش فرآیندها امکانپذیر نیست.

### واحد اندازه گیری FP

اما چگونه باید نرم افزارها را مورد سنجش قرارداد؟ واحد اندازه گیری

نرم افزار چیست؟ آیا اندازه نرم افزار هم مانند مسافت که با کیلومتر یا درجه

حرارت که با سانتیگراد بیان می شوند، قابل اندازه گیری و بیان است؟ پاسخ

این سوال مثبت است. از سالها پیش، سازمانهای نرم افزاری اقدام به معرفی

واحد اندازه گیری نرم افزار نمودند. اولین واحد اندازه گیری بر حسب تعداد خطوط

برنامه (LOC) بود که تا چند سال پیش از مقبولیت زیادی برخوردار بود. این واحد

با محدودیتهایی به شرح زیر مواجه است.

\* نداشتن تعریف استاندارد و مدرن

\* عدم پشتیبانی سازمانی معتبر از آن

\* عدم پشتیبانی از اکثر زبانهای نسل

چهارم

\* عدم پشتیبانی از reusable componentها

\* دقیق نبودن شاخص بهره وری با

توجه به LOC

علیرغم تمام این محدودیتهای طرفداران LOC اعتقاد دارند، این واحدها می تواند میزان واقعی کار تهیه کنندگان



FP قادر به اندازه گیری کارکرد سیستم های تجاری عادی، سیستم های GUI، سیستم های Object-Oriented Client/Server می باشد. اما این واحد اندازه گیری نیز با محدودیتهایی مواجه است. FP نمی تواند برای سنجش کارکرد سیستم های real-time به کار رود. همچنین نمی تواند معیار کامل و مناسبی جهت تعیین ارزش کسب و کار و یا سنجش میزان تلاش مورد نیاز جهت تهیه نرم افزار باشد. از طرفی برای تعیین میزان تلاش لازم جهت خطایابی یا تنظیم عملکرد نیز نمی توان از FP استفاده کرد. از همین رو، برای پوشش دادن نقاط ضعف FP، واحدهای اندازه گیری تکمیلی دیگری چون Feature points و MKII FP ابداع و ارائه شده اند که شرح آنها در این مقاله نمی گنجد.

**اهمیت FP از دیدگاه ذینفعان**  
اگر ذینفعان یک سازمان نرم افزاری را مدیران ارشد، کارکنان و مشتریان سازمان

برونسپاری  
\* مقایسه عملکرد سازمان با دیگر سازمانها و داده های صنعت و الگوبرداری  
\* تدوین استراتژیهای نگهداری نرم افزار  
\* سنجش ارزش و هزینه ها  
\* شناسایی و نظارت بر بهبود فرآیند  
\* مستندسازی پروژه ها و سیستم ها نیز می باشند.  
همچنین FP کاربردهای دیگری نیز دارد که از جمله می توان به بهره گیری از FP جهت:  
\* تعیین زمان و مکان اجراء مهندسی مجدد  
\* تخمین تعداد Test Case ها  
\* درک خطاهای بالقوه  
\* تدوین مجموعه ای از استانداردهای اندازه گیری  
دامنه کاربرد FP

نرم افزار را اندازه گیری کند. به دلیل محدودیتهای زیاد LOC، موسسه IFPUG در ۱۹۷۴، واحد اندازه گیری FP را معرفی و عرضه کرد. نرم افزار را از دیدگاه کاربر یا کارکرد اندازه گیری می کند.  
FP کاملاً مستقل از نوع زبان برنامه نویسی، متدولوژی، تکنولوژی، با توانایی تیم توسعه نرم افزار بوده و شاخصهای توسعه و نگهداری نرم افزار را اندازه گیری می کند. FP می تواند به تدوین شاخصهای استاندارد در سطح سازمان، و جامعه نرم افزار کمک کند.  
حال با داشتن توانایی سنجش اندازه نرم افزار، قادر به:  
\* سنجش بهره وری و کیفیت نرم افزار  
\* مدیریت دامنه پروژه  
\* بهبود بخشیدن به تخمین های پروژه  
\* شفاف ساختن نیازمندیهای کاربر  
\* مدیریت مشتری و قراردادهای

در نظر بگیریم، هر کدام از آنها به شکلی از وجود FP سود خواهند برد.

الف) مدیران ارشد:

مدیران ارشد با داشتن FP و شاخص‌های اندازه‌گیری، امکان مدیریت و کنترل موثر پروژه‌ها را به دست می‌آورند. همچنین می‌توانند به مقایسه شاخص‌های خود با شاخص‌های صنعت پرداخته و وضعیت دقیق خود را در حوزه رقابت مشخص سازند. مسلماً دانستن وضعیت کنونی به تنظیم استراتژی‌ها و برنامه‌های آینده کمک می‌کند. ب) مدیران پروژه و برنامه‌نویس‌ها:

مدیران پروژه می‌توانند با کمک FP طرح‌های پروژه را با دقت بیشتری کمی کرده و پیشرفت پروژه را پیگیری نمایند. همچنین وظایف برنامه‌نویس‌ها نیز کاملاً مشخص بوده و تخمین‌های اولیه در مورد پروژه به واقعیت نزدیکتر می‌شوند.

ج) مشتری:

FP به ایجاد زبان مشترک بین برنامه‌نویس‌ها و مشتری کمک می‌کند. نیازمندی‌ها به صورت شفاف مشخص می‌شوند. مذاکرات هنگام عقد قرارداد دقیقتر و شفاف‌تر می‌شوند. کنترل بودجه نیز تسهیل می‌شود.

**FPA چیست؟**

FPA متدی مطمئن جهت سنجش اندازه نرم‌افزار است. یکی از اهداف FPA، ارزیابی توانایی‌های سیستم از دیدگاه کاربران است. برای رسیدن به چنین هدفی، تحلیل باید مبتنی بر روش‌های مختلف تعامل کاربر با کامپیوتر باشد. از دیدگاه کاربر، هر سیستم از طریق ۵ کارکرد، فعالیت‌های مورد نظر را اجرا می‌کند. این کارکردها به دو دسته تقسیم می‌شوند:

- کارکردهای داده‌ای (نیازمندی‌های داده‌ای کاربر نهایی)

- کارکردهای مرادده‌ای (نیازمندی‌های دسترسی به داده‌ها)

در سطور ذیل این کارکردها به طور مختصر شرح داده می‌شوند. کارکردهای داده‌ای:

این دسته از دو کارکرد فایل‌های منطقی داخلی (ILF)، و فایل‌های رابط خارجی (EIF) تشکیل شده است.

فایل‌های منطقی داخلی ILF: به داده‌هایی اشاره دارد که کاربر در قبالت نگهداری آنها مسئولیت دارد. برای نمونه، داده‌هایی که کاربر سیستم پرسنلی در مورد مشخصات فردی پرسنل، شماره

پرسنلی، تاریخ استخدام، ... ثبت سیستم کرده یا از آنها گزارش می‌گیرد، از این دسته هستند.

فایل‌های رابط خارجی EIF: داده‌هایی هستند که کاربر سیستم مسئولیتی در قبالت نگهداری آنها ندارد. داده‌ها در سیستم مستقل دیگری موجودند و توسط کاربر دیگری حفظ و نگهداری می‌شوند. کاربر سیستم تنها جهت پاره‌ای از نیازها به این نوع داده‌ها EIF رجوع می‌کند. مانند سیستم حقوق و دستمزد، که جهت چاپ گزارش حقوقی کارکنان با داشتن شماره پرسنلی، مشخصات فردی کارمندان را از سیستم پرسنلی استخراج می‌کند. کارکردهای مرادده‌ای:

این دسته از سه کارکرد ورودی خارجی (EI)، خروجی خارجی (EO)، و جستجوی خارجی (EQ) تشکیل شده است.

ورودی خارجی: این کارکرد به کاربر امکان می‌دهد از طریق امکانات افزودن، تغییر، و حذف داده‌ها ILF را حفظ و نگهداری کند. به عنوان مثال، اضافه و حذف رکوردها در سیستم پرسنلی از این دست است.

خروجی خارجی: این کارکرد به کاربر امکان می‌دهد، خروجی تولید کند. این خروجی می‌تواند با استفاده از داده‌های سیستم موجود یا سیستم‌های مستقل دیگر تهیه شود. برای نمونه می‌توان به امکانات گزارش‌گیری و نمایش لیست حقوق کارکنان در سیستم حقوق و دستمزد اشاره کرد.

جستجوی خارجی: آخرین کارکرد به امکان انتخاب و نمایش داده‌های خاصی از درون فایل اشاره دارد. در این حالت کاربر با تعیین یک شرط یا چندین شرط، و بدون دستکاری داده‌ها به صورت بازایی مستقیم، آنها را استخراج می‌کند. برای نمونه کاربر، فهرست افرادی را که حقوق بیش از ۲۰۰۰۰۰ تومان گرفته‌اند را به صورت مستقیم از فایل بازایی می‌کند.

همانطور که ملاحظه شد پنج کارکرد متفاوت وجود دارند که این کارکردها باید کمی شوند.

در راهنمای ۴.۰ IFPUG، قواعدی جهت شمارش کارکردهای پنج‌گانه ذکر شده‌اند. بر اساس این قواعد، پیچیدگی هر کارکرد با توجه به جدول خاصی تعیین و امتیازدهی می‌شود. به جمع امتیازات کارکردها، عدد تعدیل نشده (UFP) FP اطلاق می‌شود.

از طرفی در این راهنما برای هر سیستم ۱۴ مشخصه عمومی قید شده که بر اساس آن فاکتور کارکرد کلی نرم‌افزار نهایی به دست می‌آید. به این فاکتور، فاکتور تعدیل شده (VAF) FP گویند. بعضی از این مشخصه‌های عمومی عبارتند از:

ارتباطات، پردازش توزیعی، قابلیت استفاده مجدد، سهولت نصب، سهولت اجرا، بروزآوری درنگ، ...

در نهایت مقادیر UFP و VFP در هم ضرب می‌شوند و مقدار FP به دست می‌آید.

لازم به ذکر است، اگر پروژه در حالت ایجاد یا توسعه باشد، فرمول فوق کمی تغییر می‌کند.

در چه مرحله‌ای از چرخه حیات سیستم، می‌توان FP را شمارش کرد؟

در رویکرد سنتی، بیشتر افراد فکر می‌کنند که شمارش FP نباید زودتر از مرحله طراحی محصول انجام شود، اما در ۴.۰ IFPUG، با توجه به راهنماها و قواعد موجود می‌توان FP‌ها را به محض اتمام مرحله نیازمندی‌ها شمارش کرد.

در مرحله امکان‌سنجی، امکان شمارش FP‌ها وجود ندارد. اما می‌توان FP‌ها را با استفاده از بعضی تکنیک‌ها مشخص ساخت. به عنوان مثال، اگر پروژه مشابهی ۲۰۰۰ FP داشته است، حدس زده می‌شود که این پروژه هم ۲۰۰۰ FP داشته باشد.

در خلال مرحله جمع‌آوری نیازمندی‌ها، تخمین اندازه FP‌ها را می‌توان مرتباً پالایش کرد. برای بعضی پروژه‌ها، مدل منطقی داده‌ای تهیه می‌شود. اگر نمودار زمینه‌ای یوردون برای پروژه تهیه شود، تعامل کاربر و سیستم‌های خارجی مشخص می‌شود، و بر اساس آن می‌توان فاکتور تطبیق ارزش (VAF) را با دقت نسبی به دست آورد.

به محض مشخص شدن نیازمندی‌های کسب و کار، شمارش دقیق FP برای نرم‌افزار میسر می‌شود. از این زمان به بعد باید تغییرات نیازمندی‌ها و تأثیر آن بر پروژه را مورد نظر قرار داد. ساده‌ترین نوع تغییرات، افزایش دامنه کار است. (فرضاً در انتهای مرحله تعیین نیازمندی‌ها ۱۰۰۰ FP برآورده شده، در حالی که در انتهای مرحله طراحی ۱۵۰۰ FP داریم). یک سری تغییرات دیگر نیز باید مد نظر باشد. ممکن است تعداد FP‌ها در انتهای مرحله نیازمندی‌ها و طراحی ۱۰۰۰ باشند ولی بسیاری از کارکردها تغییر

کرده باشند.

چه کسانی می‌توانند FPها را شمارش کنند؟

هر کسی می‌تواند این کار را انجام دهد. تنها گذراندن یک کلاس یک روزه و یک آموزش ضمن خدمت کوتاه توسط یک متخصص شمارش برای مسلط شدن در این مقوله کفایت می‌کند. متخصصان فنی، کاربران سیستم و حتی مدیران ارشد می‌توانند FPها را شمارش کنند. البته شمارش FP توسط هر یک از این گروه‌ها مزایا و معایبی دارد که باید با توجه به وضعیت سازمان، راهکار مناسب را در این زمینه انتخاب کرد.

چند نفر باید درگیر شمارش FPها گردند؟

سه روش برای تخصیص افراد وجود دارد. در روش اول، هر یک از اعضای تیم پروژه، درگیر شمارش FPها می‌گردند. البته این کار مشکلاتی به همراه دارد. از آنجایی که قواعد FP فرار بوده و به مرور زمان در اثر عدم استفاده به فراموشی سپرده می‌شوند، شمارش FP در طی زمان از دقت یکسانی برخوردار نخواهد بود. از این رو باید به دفعات و در فواصل معین بازآموزی صورت گیرد.

در روش دوم، در سازمان‌های بزرگ گروه کوچکی مسئولیت شمارش FP را به عهده می‌گیرند. این گروه باید به صورت مستمر و مداوم با قوانین شمارش FP آشنایی کافی داشته باشند. مزیت این روش سازگاری شمارش است. همچنین می‌توان در موارد دشوار، از وجود یک فرد خبره در گروه نیز بهره برد.

در بعضی از سازمان‌ها نیز کار شمارش را به مشاور می‌سپارند. این فرد باید تجربه و اطلاعات بروز داشته باشد. همچنین بهتر است از اعضای IFPUG بوده و در کنفرانس‌های سالانه آن شرکت کند.

**برونسپاری شمارش FP:**

می‌توان شمارش FP را به متخصصین خارج سازمان سپرد. این کار مزایای زیادی دارد که از جمله می‌توان به خبره بودن سازمان خارجی، آزادسازی منابع درون سازمان، بروز بودن دانش و اعتبار سازمان خارجی، اطمینان از صحت شمارش و مستقل بودن شمارش از تیم نرم‌افزار اشاره کرد.

اما برونسپاری معایب خاص خود را نیز دارد:

- از آنجایی که سازمان‌ها تمایل به



شمارش پروژه توسعه، و شمارش نرم‌افزار کاربردی.

**(انواع شمارش: نتیجه نهایی شمارش پروژه تهیه نرم‌افزار و نرم‌افزار کاربردی لزوماً بایستی برابر باشند. یعنی طرح ریزی برای تهیه نرم‌افزاری با ۱۰۰۰ FP باید منجر به نرم‌افزاری با ۱۰۰۰ FP شود. اما پروژه توسعه با ۵۰۰ FP ممکن است منجر به نرم‌افزار کاربردی از ۵۰۰ تا ۱۵۰۰ FP شود. زیرا در پروژه توسعه علاوه بر افزودن کارکرد، تغییر و حذف کارکردها نیز صورت می‌گیرد.)**

- تشریح فرآیند شمارش:

شما باید برای متخصص شمارش شرح دهید که از طریق شمارش قصد دارید به چه اهدافی دست یابید و از این اطلاعات می‌خواهید چه استفاده‌ای کنید.

درک مفهوم FP و اندازه‌گیری نرم‌افزار دارند، با برونسپاری این فعالیت از درک فلسفه وجودی شمارش عاجز خواهند شد.

- با برونسپاری شمارش FP، تخصص شمارش FP در سازمان به تدریج کاهش می‌یابد. چون معمولاً کارکنان قواعد شمارش را به تدریج از یاد می‌برند.

- عقد قرارداد با شرکت‌های مشاوره در زمینه شمارش FP، هزینه‌هایی را بر سازمان تحمیل می‌کند که بسیار بیشتر از هزینه‌های شمارش به صورت خانگی است.

**گام‌های شمارش:**

- شناسایی طرح شمارش:

در این گام باید نوع شمارش و دامنه شمارش مشخص شود. سه نوع شمارش عبارتند از: شمارش پروژه تهیه نرم‌افزار،



تعیین و اقدام به بهبود فرآیندهای نرم افزاری نمایند. برای تشخیص وضعیت جاری فرآیندهای سازمان، پرسشنامه استاندارد نیز موجود است که می توان از آن بهره گرفت. سازمان ها به منظور کسب اطمینان نسبت به نیل به اهداف هر سطح، می توانند از شاخص های سنجش استفاده کنند. یکی از واحدهای معتبر برای تعیین شاخص های اندازه گیری، واحد FP است. FP نرم افزار را از دیدگاه کاربر یا کارکرد اندازه گیری می کند و این به نوبه خود به ایجاد زبان مشترک بین کاربر (مشتری) و متخصصان فنی سیستم کمک می کند. از طرفی، FP در زمینه تدوین شاخص های بهره وری و شاخص های کیفیت، نقش حیاتی دارد. با بهره گیری از نتایج سنجش شاخص ها، می توان فرآیندها و محصولات نرم افزاری را بهبود بخشید. بدین منظور می توان با استفاده از CMM استراتژی های بهبود فرآیند سازمان را تدوین و در یک دوره زمانی قابل پیش بینی به اهداف مورد نظر دست یافت.

همچنین، امروزه بیشتر شرکت ها، فرآیند مهندسی مجدد را بدون توجه به تحلیل های هزینه منفعت انجام می دهند. در حالی که با اجراء تحلیل های دقیق می توان بخش هایی را که واقعاً نیازمند مهندسی مجدد هستند، شناسایی کرد. FP به متخصصان کمک می کند تا به صورت دقیق تحلیل هزینه منفعت را انجام داده و مناطق نیازمند مهندسی مجدد را مشخص سازند.

منابع:

- 1) The Capability Maturity Model for Software, Mark C. Paulk, etc.
- 2) Function point FAQ
- 3) IFPUG User Manual 4.0
- 4) IFPUG Site
- 5) Using Function Points, David Longstreet.
- 6) Test cases & Defects, David Lonstreet.
- 7) Use Case and Function points, David Longstreet.
- 8) What problem is your organization trying to solve? David longstreet.
- 9) Software Metrics.com - Applied software economics, David Longstreet.
- 10) An introduction to Function Point Analysis, Roger Heller.
- 11) Estimating software from requirements, Arlence Minkiewics.

- محاسبه فاکتور تطبیق ارزش (VAF):  
در این مرحله، با استفاده از VAF، سطح پیچیدگی سیستم برای متخصصان آشکار می شود.  
- شمارش انواع کارکردهای داده ای:  
در این گام ILF ها و EIF ها شمارش می شوند.  
- شناسایی انواع کارکردهای مرادده ای:  
این دسته شامل شمارش EO، EI و EQ هاست و طولانی ترین مرحله شمارش را تشکیل می دهد.  
- اجراء محاسبات  
- تعیین اعتبار شمارش:  
در این مرحله نسخه ای از شمارش در اختیار مدیر پروژه قرار می گیرد تا از لحاظ شدن تمامی کارکردها در هنگام شمارش اطمینان حاصل کند.  
- بازنگری نتایج:  
نتایج شمارش باید پس از یک یا دو روز بعد از آن، در اختیار پرسنل پروژه و مسئولین ذیربط قرار گیرد.  
طول مدت شمارش:  
مشاوران شمارش FP اعتقاد دارند که شمارش باید در طی نصف یا یک روز به اتمام رسد.  
نتیجه گیری:  
همانطور که پیش از این نیز بیان شد، امروزه شرکت های نرم افزاری در جستجوی راه هایی جهت بهبود فرآیندهای نرم افزاری و بالتبع افزایش بهره وری و کیفیت محصول هستند. مدل CMM به مدیران شرکت های نرم افزاری امکان می دهد تا با توجه به ویژگی های سطوح مختلف، جایگاه خود را