



# مدیران حوزه نرم افزار

مقدمه:

زمینه بهبود مستمر فرآیندهای نرم افزاری، مدل CMM است. در این مدل برای حرکت به سطوح بالاتر، باید دسته‌ای از اهداف و الزامات را برآورده ساخت و زیر ساختهای مناسب را تدارک دید. اما چگونه باید از نیل به 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.

بی شک امروزه در فضای رقابتی شرکتهای نرم افزاری، ارتقا کیفیت و بهره‌وری راز ماندگاری و بقا است. برای نیل به دو اهداف مذکور، تلاشهای وسیعی صورت گرفته و استانداردهایی در سطح بین المللی تدوین و ارائه شده‌اند. این استانداردها بر بهبود مستمر فرآیندها و فعالیتهای سازمانهای نرم افزاری تاکید دارند. بهبود مستمر، فرآیند تدریجی و آرام تحول پذیری است. یکی از مدل‌های معتبر در

## سطوح پنج گانه بلوغ فرآیندهای

### نرم افزار

بهبود مستمر بر گامهای کوچک و تدریجی استوار است. ساختار مرحله‌ای CMM بر مبنای اصول کیفیت محصول بنا نهاده شده است. CMM چارچوبی ایجاد می‌کند که در آن پنج سطح تعبیه شده و بهبود مستمر و تدریجی فرآیندها بدین طریق حاصل می‌شود.

این سطوح، سنجش میزان بلوغ فرآیند نرم افزار و ارزیابی توانایی نرم افزار را میسر می‌سازند. همچنین به اولویت بندی اقدامات بهبود کمک می‌کنند. در هر سطح، مجموعه‌ای از اهداف در نظر گرفته شده‌اند، که نیل به این اهداف موجب تثبیت شدن یک جز (مهم از فرآیند نرم افزار شده و این امر به افزایش قابلیت فرآیندهای سازمان منجر می‌شود. 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 اجرای یکسری از اقدامات سطوح دوم تا پنجم قابل دسترسی می‌شوند.

### سطح اول: سطح اولیه

ویژگیهای عمده این سطح توانایی کارکنان است. از جمله دیگر ویژگیهای سطح اول می‌توان به موارد زیر اشاره کرد:

● وجود محیط پایدار جهت توسعه و نگهداری نرم افزار

● تخطی از رویه‌های برنامه‌ریزی شده

● وابستگی به مدیر توانا و کارآفرین

● به اتمام رساندن پروژه‌ها 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

● بروز شگفتی‌های فنی زیاد

### سطح دوم: سطح قابل تکرار

ویژگی اصلی این سطح قابلیت تکرار برنامه‌هاست. از جمله دیگر ویژگیهای قابل ملاحظه سطح دوم می‌توان به موارد زیر اشاره کرد:

● تدوین خط مشی جهت مدیریت پروژه

● تدوین رویه‌هایی جهت به اجرا در آوردن خط استراتژی‌ها

● وجود کنترل‌های مدیریت نرم افزار

● تهیه و تدوین استانداردهایی جهت پروژه‌های نرم افزاری و پیروی از آنها

● احتمال متفاوت بودن فرآیندها در پروژه‌های مختلف

● تضمین ساده محصول و فرآیند

● مدیریت فروشنده

● برنامه‌ریزی و مدیریت پروژه‌های جدید بر اساس پروژه‌های مشابه قبلی صورت می‌گیرد

● تعهدات پروژه فعلی بر مبنای عملکرد و نتایج پروژه‌های قبلی تنظیم می‌شوند

● استراتژی، راهنمای پروژه است

### سطح سوم: سطح تعریف شده

ویژگی این سطح استانداردها هستند. از خصوصیات این سطح می‌توان به موارد زیر اشاره کرد:

● تدوین فرآیندی استاندارد جهت توسعه و نگهداری نرم افزار

● تلفیق مهندسی نرم افزار با فرآیندهای مدیریت

چالشهای پیش روی شرکتهای نرم افزاری

با توسعه فن آوری اطلاعات و افزایش بی سابقه شرکتهای نرم افزاری کوچک و متوسط رقابت بین این شرکتهای به طرز چشم گیری افزایش یافته است. از این رو، توجه به خواسته‌ها و تأمین نیازهای مشتری و حتی در مواردی فرای درک و خواست مشتری عمل کردن ضامن بقا و رشد شرکتهای نرم افزاری است اما در حال حاضر اکثر شرکتهای نرم افزاری کوچک و جدید و بعضی شرکتهای متوسط با مشکلاتی مشابه رو به رو هستند:

● عدم شناسایی کامل نیازهای کاربر

● تخمین نادرست زمان بندی پروژه

● تخمین نادرست اندازه و هزینه‌های پروژه

● عدم تخصیص منابع کافی (اعم از انسانی، مالی، تجهیزاتی) جهت اجرای پروژه

● پایین بودن سطح بهره‌وری

● نبود توانایی لازم جهت مدیریت صحیح نیازهای در حال تغییر مشتری

● نبود اطلاعات دقیق و صحیح در هنگام مباحثات عقد قرار داد

همچنین شرکتهایی که قصد دارند فعالیتهای تولید نرم افزار خود را به دیگران بسپارند 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 می‌کند.

به منظور افزایش بهره‌وری، تعیین صحیح زمان بندی و لیست ملزومات پروژه و افزایش توانایی مدیریت و کنترل پروژه، استفاده از روشهای بهبود فرآیندهای نرم افزاری و تدوین شاخصهای سنجش اجتناب ناپذیر به نظر می‌رسند. اما این کار را چگونه باید آغاز کرد؟

### مدل CMM

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

از همین رو، در نوامبر ۱۹۸۶، موسسه مهندسی نرم افزار (SEI) با مشارکت شرکت مینتر اقدام به تدوین چارچوب بلوغ فرآیند نمود.

هدف از این کار، بهبود بخشیدن فرآیندهای نرم افزاری بود. در نتیجه در سپتامبر ۱۹۸۷، SEI توانست دو روش با نامهای ارزیابی فرآیند

نرم افزار، ارزیابی قابلیت نرم افزار و یک پرسشنامه بلوغ تدوین و ارائه کند. سالها بعد، SEI به دلیل محک زدن روشهای فوق و کسب

تجربه‌های گسترده در حوزه صنعت و سازمانهای دولتی، مدل بلوغ توانایی (CMM) را برای سازمانهای نرم افزاری عرضه کرد. CMM به

ارائه مجموعه‌ای از اقدامات در مجموع منجر به افزایش توانایی فرآیندهای نرم افزار می‌شود.

CMM برای انتخاب استراتژیهای بهبود فرآیند طراحی شده است. این انتخاب بر مبنای دو فاکتور اساسی "میزان بلوغ فرآیند جاری" و "شناسایی

موضوعات کوچک اما بسیار مهم" در زمینه فرآیند و کیفیت نرم افزار امکان پذیر می‌گردد. پس از انتخاب استراتژی بهبود، سازمان باید یکسری

اقدامات محدود را اجرا کند تا بتواند فرآیند نرم افزاری خود را بهبود بخشد.

### واژه‌های کلیدی:

پیش از تشریح سطوح مدل CMM، بهتر است واژه‌های کلیدی بحث به صورت مختصر توضیح داده شوند.

● توانایی فرآیند نرم افزاری: به نتایج مورد انتظار از یک فرآیند نرم افزاری خاص اشاره دارد.

● عملکرد فرآیند نرم افزار: نشانگر نتایج واقعی حاصل از فرآیند نرم افزار است.

● بلوغ فرآیند نرم افزار: نشانگر درجه‌ای است که یک فرآیند به روشنی تعریف و مدیریت شده، مورد سنجش قرار گرفته، کنترل شده و مؤثر واقع شده است.

بلوغ تلویحاً به پتانسیل رشد توانایی اشاره می‌کند.

## پرش از بعضی سطوح بلوغ

علیرغم این که می‌توان بعضی از سطوح را جا گذاشته و به سطح بالاتر پرید، اما به سازمانها توصیه می‌شود، به این کار دست نزنند زیرا در هر سطح پی‌ریزی ملزومات زیربنایی سطح بالاتر صورت می‌گیرد.

### کاربردهای CMM

مدل CMM، مسیر بهبود خاصی را برای سازمانهای نرم‌افزاری فراهم آورده و به آنها توصیه می‌کند در صورت تمایل به افزایش توانایی نرم‌افزاری، از مدل CMM استفاده کنند.

مدل CMM چهار کاربرد عمده دارد:

- تیم ارزیابی می‌تواند برای شناسایی نقاط قوت و ضعف سازمان از CMM استفاده کند
- تیم‌های ارزیابی می‌تواند برای شناسایی مشکلات انتخاب پیمانکاران مختلف و همچنین نظارت بر قراردادها از CMM استفاده کند.
- مدیری ارشد برای درک اقدامات ضروری جهت بهبود فرآیند نرم‌افزاری سازمان از CMM استفاده می‌کند.
- کارکنان فنی و گروه‌های بهبود فرآیند، از CMM به عنوان راهنمایی جهت تعریف و بهبود فرآیند نرم‌افزار در سازمان بهره می‌گیرند.

### زمینه‌های کلیدی فرآیند

زمینه‌های کلیدی فرآیند (KPA) به گروهی از اقدامات مرتبط اطلاق می‌شود که در صورت اجرا شدن تمامی آنها، مجموعه‌ای از اهداف در زمینه گسترش توانایی فرآیند، تحقق می‌یابند.

به این ترتیب، برای رسیدن به هر سطح بلوغ، زمینه‌های کلیدی فرآیند آن سطح باید ارضا شود. از آنجایی که سطح اول، پایین‌ترین سطح می‌باشد، برای حرکت به سطوح بالاتر، KPAها عبارتند از:

- برای رسیدن به سطح دوم:
  - مدیریت نیازمندیها
  - برنامه‌ریزی پروژه نرم‌افزار
  - پیگیری و نظارت بر پروژه نرم‌افزار
  - مدیریت قرار دادهای نرم‌افزار (یعنی انتخاب پیمانکار شایسته و مدیریت مؤثر وی)
  - تضمین کیفیت نرم‌افزار
- مدیریت پیکره بندی نرم‌افزار (ایجاد و حفظ یکپارچگی محصول در طول چرخه حیات پروژه نرم‌افزاری)
  - برای رسیدن به سطح سوم:
    - تمرکز بر فرآیند سازمان
    - تعریف فرآیند سازمان
  - ادامه برنامه‌های آموزشی به منظور توسعه مهارتها و دانش افراد در گیر پروژه
  - مدیریت یکپارچه نرم‌افزار
  - مهندسی محصول نرم‌افزاری
  - ایجاد هماهنگی بین گروه‌های کاری
  - بازنگری نظیر به نظیر
  - برای رسیدن به سطح چهارم:
    - مدیریت کمی فرآیند
    - مدیریت کیفی نرم‌افزار
    - 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 پروژه‌هائمی پردازد. فرضاً CMM در حال حاضر به موضوعاتی نظیر حمایت از فناوریهای



- آموزش مدیران و کارکنان برای آشنایی آنها با دانش و مهارتهای لازم
- وجود گروه‌های مسئول در قبال فرآیندهای نرم‌افزاری سازمان
- تعریف شاخص آمادگی، ورودیها، خروجیها، استانداردها، روشهای اجرا) کار، مکانیزمهای تصدیق
- تدوین شاخصهایی جهت کنترل پروژه، برنامه زمان‌بندی، هزینه‌ها، وظایف و وجود خلاصه‌ای از کیفیت نرم‌افزار

### سطح چهارم: سطح مدیریت شده

- ویژگی مهم این سطح، قابلیت اندازه‌گیری و پیش‌بینی است. مشخصه‌های دیگر سطح چهارم عبارتند از:
  - تدوین اهداف کمی و کیفی برای محصولات و فرآیندها
  - سنجش بهره‌وری و کیفیت در قالب یک برنامه‌اندازه‌گیری
  - ایجاد بانک اطلاعاتی جهت جمع‌آوری و تحلیل داده‌ها
  - 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 می‌توان این موارد را نام برد:
  - تمرکز سازمان بر بهبود مستمر
  - وجود وسائلی جهت شناسایی نقاط قوت و ضعف سازمان
  - هدف، جلوگیری از انجام خطا و اشتباه است.
  - اجرا تحلیل هزینه‌ها - منفعت جهت تحلیل فن‌آوریهای جدید و تغییرات پیشنهادی
  - تحلیل خطاها به منظور یافتن دلایل بروز خطا
  - ارزیابی فرآیندها به منظور جلوگیری از رخدادهای شناخته شده
  - انتقال بهترین نوآوریها در زمینه مهندسی نرم‌افزار به تمام سازمان
  - اصلاح، تغییر و بهبود فرآیندها در صورت وجود اتلاف
  - بهبود به دو طریق حاصل می‌شود:
    - ۱- ایجاد پیشرفتهای تدریجی در فرآیندهای موجود
    - ۲- ایجاد نوآوری با استفاده از فن‌آوریها و متدهای نوین

نرم‌افزاری خاص، یا ارائه راه حل در زمینه نحوه انتخاب، استخدام، انگیزش و حفظ نیروهای شایسته نمی‌پردازد. 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 کاملی نیست. اما ابزاری مفید برای جهت دادن به تلاشهای بهبود فرآیند نرم‌افزار است. CMM ساختاری مفهومی جهت بهبود مدیریت و توسعه محصولات نرم‌افزاری به شکل منظم و سازگار است.

اینک این سؤال مطرح می‌شود که چگونه می‌توان از دستیابی به اهداف هر سطح و بهبود فرآیندهای نرم‌افزاری اطمینان حاصل کرد؟ این کار جز از طریق سنجش فرآیندها امکانپذیر نیست.

### واحداندازه‌گیری FP

اما چگونه باید نرم‌افزارها را مورد سنجش قرار داد؟ واحداندازه‌گیری نرم‌افزار چیست؟ آیا اندازه نرم‌افزار هم مانند مسافت که با کیلومتر یا درجه حرارت که با سانتیگراد بیان می‌شود، قابل اندازه‌گیری و بیان است، پاسخ این سؤال مثبت است. از سالها پیش، سازمانهای نرم‌افزاری اقدام به معرفی واحداندازه‌گیری نرم‌افزار نمودند. اولین واحداندازه‌گیری 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 معتبر از آن
- عدم پشتیبانی از اکثر زبانهای نسل چهارم
- عدم پشتیبانی از reusable component ها
- دقیق نبودن شاخص بهره‌وری با توجه به LOC

علیرغم تمام این محدودیتهای، طرفداران LOC اعتقاد دارند این واحد می‌تواند میزان واقعی کار تهیه کنندگان نرم‌افزار را اندازه‌گیری کند.

به دلیل محدودیتهای زیاد LOC، موسسه IFPUG در 19۷۴، واحداندازه‌گیری FP را معرفی و عرضه کرد. FP نرم‌افزار را از دیدگاه کاربر یا کارکرداندازه‌گیری می‌کند.

FP کاملاً مستقل از نوع برنامه نویسی، متدولوژی و تکنولوژی یا توانایی تیم توسعه نرم‌افزاری بوده و شاخصهای توسعه و نگهداری نرم‌افزار را اندازه‌گیری می‌کند. FP می‌تواند به تدوین شاخصهای استاندارد در سطح سازمان و جامعه نرم‌افزار کمک کند.

حال با داشتن توانایی سنجش اندازه نرم‌افزار، قادر به انجام موارد زیر می‌باشیم

- سنجش بهره‌وری و کیفیت نرم‌افزار
- مدیریت دامنه پروژه
- بهبود بخشیدن به تخمین‌های پروژه
- شفاف ساختن نیازمندیهای کاربر
- مدیریت مشتری و قراردادهای برونسپاری
- مقایسه عملکرد سازمان با دیگر سازمانها و داده‌های صنعت و الگو برداری
- تدوین استراتژیهای نگهداری نرم‌افزار
- تدوین استراتژیهای نگهداری نرم‌افزار
- سنجش ارزش و هزینه‌ها
- شناسایی و نظارت بر بهبود فرآیند
- مستند سازی پروژه‌ها و سیستم‌ها

همچنین FP کاربردهای دیگری نیز دارد که از جمله می‌توان به بهره‌گیری از FP جهت:

- تعیین زمان و مکان اجرا مهندسی مجدد
- تخمین تعداد TestCase
- درک خطاهای بالقوه
- تدوین مجموعه‌ای از استانداردهای اندازه‌گیری

### دامنه کاربرد FP

FP قادر به اندازه‌گیری کارکرد سیستم‌های تجارت عادی، سیستمهای GUI، Server Client و Object-Oriented می‌باشد.

اما این واحداندازه‌گیری نیز با محدودیتهایی مواجه است. FP نمی‌تواند برای سنجش کارکرد سیستم‌های real-time به کار رود. همچنین نمی‌تواند معیار کامل و مناسبی جهت تعیین ارزش کسب و کار یا سنجش میزان تلاش مورد نیاز جهت تهیه نرم‌افزار باشد. از طرفی برای تعیین میزان تلاش لازم جهت خطایابی یا تنظیم عملکرد نیز نمی‌توان از FP استفاده کرد. از همین رو، برای پوشش دادن نقاط ضعف FP، واحدهای اندازه‌گیری تکمیلی دیگری چون points Feature و MKII FP ابداع و ارائه شده‌اند که شرح آنها در این مقاله نمی‌گنجد.

### اهمیت FP از دیدگاه ذینفعان

اگر ذینفعان یک سازمان نرم‌افزاری را مدیران ارشد، کارکنان و مشتریان سازمان را 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 مؤثر 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 بیشتر ارزیابی کرده و پیشرفت پروژه را پیگیری نمایند و 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 دقیق تر و شفاف تر می‌شوند و کنترل بودجه نیز تسهیل می‌شود.

### FPA چیست؟

FPA متدی مطمئن جهت سنجش اندازه نرم‌افزار است. یکی از اهداف FPA ارزیابی توانایی‌های سیستم از دیدگاه کاربر است. برای رسیدن به چنین هدفی تحلیل باید مبتنی بر روشهای مختلف تعامل کاربر با کامپیوتر باشد. از دیدگاه کاربر، هر سیستم از طریق پنج 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 دو کارکرد فایلهای منطقی داخلی (ILF) و فایلهای روابط خارجی (EF) تشکیل شده است.

**فایلهای منطقی داخلی:** ILF به داده‌هایی اشاره دارد که کاربر در قبال نگهداری آنها مسئولیت دارد. برای نمونه داده‌هایی که کاربر سیستم پرسنلی در مورد مشخصات فردی پرسنل، شماره پرسنلی، تاریخ ثبت نام و ... ثبت سیستم کرده یا از آنها گزارش می‌گیرد، از این دسته هستند.

**فایلهای رابط خارجی:** EIF داده‌هایی هستند که کاربر سیستم مسئولیتی در قبال 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 و مشخصات فردی کارمندان را از سیستم پرسنلی استخراج می‌کند.

## کارکردهای مرادده‌ای؛

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

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

**خروجی خارجی:** این کارکرد به کاربر امکان می‌دهد، خروجی تولید کند و این خروجی می‌تواند با استفاده از داده‌های سیستم موجود یا سیستم‌های مستقل دیگر تهیه شود. برای نمونه می‌توان به امکانات گزارش گیری 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 بازبازی مستقیم آنها را استخراج می‌کند. برای نمونه کاربر، فهرست افرادی را که حقوق بیش از ۲۰۰۰۰۰ تومان گرفته‌اند را به صورت مستقیم از فایل بازبازی می‌کند.

همانطور که ملاحظه شد پنج کارکرد متفاوت وجود دارند که این کارکردها باید ارزیابی شوند. در راهنمای ۴.۰ IFPUG، قواعدی جهت شمارش کارکردهای پنج گانه ذکر شده‌اند. بر اساس این قواعد، پیچیدگی هر کارکرد با توجه به جدول خاصی تعیین و امتیازدهی می‌شود. 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 چهارده مشخصه عمومی قید شده که بر اساس آن فاکتور کارکرد کلی نرم‌افزار نهایی به دست می‌آید. به این فاکتور، فاکتور تعدیل شده FP (VAP) گویند. بعضی از این مشخصه‌های عمومی عبارتند از:

ارتباطات، پردازش توزیعی، قابلیت استفاده مجدد، سهولت نصب، سهولت اجرا، به روز آوری سریع و ...

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 FP را به محض اتمام مرحله نیازمندیها شمارش کرد.

در مرحله امکان سنجی، امکان شمارش FP وجود ندارد. اما می‌توان FP را با استفاده از بعضی تکنیکها مشخص ساخت. به عنوان مثال، اگر پروژه مشابهی FP ۲۰۰۰ داشته است، حدس زده می‌شود که این پروژه هم FP ۲۰۰۰ داشته باشد در خلال مرحله جمع آوری نیازمندیها، تخمین اندازه FP را می‌توان مرتباً پالایش کرد. بعضی از 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 به دقت نسبی به دست آورد.

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

## چه کسانی می‌توانند FP را شمارش کنند؟

هر کسی می‌تواند این کار را انجام دهد. تنها گذراندن کلاس یک روزه و 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 FP 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 FP ها گردند؟

سه روش برای تخصیص افراد وجود دارد در روش اول: هر یک از اعضا تیم پروژه، درگیر شمارش FP 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 روش سوم: بعضی از سازمانها نیز کار شمارش را به مشاور می‌سپارند. این فرد باید تجربه و اطلاعات به روز داشته باشد. همچنین بهتر است از اعضا 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 خیره بودن سازمان خارجی، آزاد سازی منابع درون سازمان، به روز بودن دانش و اعتبار سازمان خارجی، اطمینان از صحت شمارش و مستقل بودن شمارش از تیم تهیه نرم‌افزار اشاره کرد.

اما پرونسیباری معایب خاص خود را نیز دارد:

- از آنجایی که سازمانها تمایل به درک مفهوم FP، و اندازه‌گیری نرم‌افزار دارند، با پرونسیباری این فعالیت از درک فلسفه وجودی شمارش عاجز خواهند شد.
- با پرونسیباری شمارش FP، تخصص شمارش آن 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، 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.

#### ● محاسبه فاکتور تطبیق ارزش (VAF)

این مرحله با استفاده از VAF سطح پیچیدگی سیستم برای متخصصان آشکارا شود.

#### ● شمارش انواع کارکردهای داده‌ای:

در این گام LF و EF شمارش می‌شوند.

#### ● شناسایی انواع کارکردهای مرادده‌ای:

این دسته شامل شمارش EO، EI، EQ می‌باشد و طولانی‌ترین مرحله شمارش را تشکیل می‌دهد.

#### ● اجرا محاسبات

#### ● تعیین اعتبار شمارش:

در این مرحله نسخه‌ای از شمارش در اختیار مدیر پروژه قرار می‌گیرد 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 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.

### نتیجه‌گیری:

همانطور که پیش از این نیز بیان شد، امروزه شرکت‌های نرم‌افزاری در جستجوی راه‌هایی جهت بهبود فرآیندهای نرم‌افزاری و در نتیجه افزایش بهره‌وری و کیفیت محصول هستند. مدل CMM به مدیران شرکت‌های نرم‌افزاری امکان می‌دهد تا با ویژگی‌های سطوح مختلف، جایگاه خود را تعیین و اقدام به بهبود فرآیندهای نرم‌افزاری نمایند. برای تشخیص وضعیت جاری فرآیندهای سازمان، پرسشنامه‌ای استاندارد نیز موجود است که می‌توان از آن بهره گرفت. سازمانها به منظور کسب اطمینان نسبت به نیل به اهداف هر سطح، می‌توانند از شاخصهای سنجش استفاده کنند. یکی از

واحدهای معتبر برای تعیین شاخصهای اندازه‌گیری، واحد FP است. FP نرم‌افزار را از دیدگاه کاربر یا کارکرد اندازه‌گیری می‌کند. و این به نوبه خود به ایجاد زبان مشترک بین کاربر (مشتری) و متخصصان فنی سیستم کمک می‌کند. از طرفی FP در زمینه تدوین شاخصهای بهره‌وری و شاخصهای کیفیت نقش حیاتی دارد. با بهره‌گیری از نتایج سنجش شاخصها، می‌توان فرآیندها و فرآیند سازمان را تدوین و در یک دوره زمانی قابل پیش‌بینی به اهداف مورد نظر دست یافت. همچنین امروزه بیشتر شرکتها فرآیند مهندسی مجدد را بدون توجه به تحلیل‌های هزینه منفعت انجام می‌دهند. درحالی‌که با اجرای تحلیل‌های دقیق می‌توان بخشهایی را که واقعا نیازمند مهندسی مجدد هستند، شناسایی کرد. FP به متخصصان کمک می‌کند تا به صورت دقیق تحلیل هزینه منفعت را انجام داده و مناطق نیازمند مهندسی مجدد را مشخص سازند.

### منابع:

- 1) The capability MarkC.pauik etc. Maturity Model For Soft ware
- 2)Function point F A Q
- 3)IFPUG User Manual 4.0
- 4)IFPUG Site
- 5)Using Function Points€ David Longstreet
- 6)Test cases & Defects€ David Longstreet
- 7)Use case and Function points€ David Longstreet
- 8)What problem is your organization trying to solve ? Davind 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.