

سوالات میان ترم

۱- در معنی و مفهوم علمی **Function point** و ترجمه مناسب آن به فارسی بحث نمایید.

Function point یک "واحد اندازه گیری" برای بیان میزان کارایی کسب و کار است که یک سیستم اطلاعاتی (به عنوان یک محصول) در اختیار کاربر قرار می دهد. از نقاط عملکرد برای محاسبه اندازه گیری میزان عملکرد نرم افزار استفاده می شود. هزینه و زمان یک واحد از پروژه های گذشته محاسبه می شود. چندین استاندارد شناخته شده و یا مشخصات عمومی برای اندازه گیری نرم افزار بر اساس **Function Point** وجود دارد.

FiSMA: ISO / IEC 29881: 2010 روش اندازه گیری اندازه عملکرد **FiSMA 1.1**

IFPUG: ISO / IEC 20926: 2009 روش اندازه گیری اندازه عملکردی **IFPUG**

Mark-II: ISO / IEC 20968: 2002 تجزیه و تحلیل عملکرد - **MI II** کتابچه راهنمای عمل شمارش

ISO / IEC 24570: 2018 روش اندازه گیری اندازه عملکردی نسما نسخه ۲,۳ - تعاریف و دستورالعمل های شمارش برای استفاده از تجزیه و تحلیل عملکرد

COSMIC: ISO / IEC 19761: 2011 یک روش اندازه گیری اندازه عملکردی.

OMG: ISO / IEC 19515 فناوری اطلاعات ۲۰۱۹ - نقاط عملکرد خودکار گروه مدیریت شی (AFP)

Function point در سال ۱۹۷۹ در اندازه گیری بهره وری توسعه برنامه توسط آلن آلبرشت در **IBM** تعریف شد. نیازهای کاربردی کاربر از این نرم افزار مشخص شده و هر یک از آن ها به پنج دسته تقسیم می شوند: خروجی ها ، سوالات ، ورودی ها ، پرونده های داخلی و رابط های خارجی. هنگامی که تابع شناسایی و به یک نوع طبقه بندی می شود ، سپس از نظر پیچیدگی ارزیابی می شود و تعدادی از نقاط تابع اختصاص می یابد. هر یک از این نیازهای کاربردی کاربر ، به یک عملکرد تجاری کاربر نهایی مانند ورودی داده برای ورودی یا سؤال کاربر برای یک پرسش ، ترسیم می شود. این تمایز از آن جهت مهم است که تمایل دارد توابع اندازه گیری شده در نقشه ارزیابی عملکرد را به راحتی به نیازهای کاربر محور تبدیل کند ، اما همچنین تمایل دارد عملکردهای داخلی (به عنوان مثال الگوریتم ها) را که برای اجرای آن ها نیز به منابع نیاز دارد.

در حال حاضر هیچ روش **FSM** شناخته شده ایزو وجود ندارد که شامل پیچیدگی الگوریتمی در نتیجه اندازه باشد. اخیراً رویکردهای مختلفی برای مقابله با این ضعف درک شده ارائه شده است که در چندین محصول نرم افزاری تجاری اجرا شده است. تغییرات روش **IFPUG** مبتنی بر آلبرشت که برای جبران این (و سایر نقاط ضعف) طراحی شده است ، شامل موارد زیر است:

ارزیابی عملکرد زودرس و آسان - پیچیدگی مسئله و داده را با دو سال تنظیم می کند که اندازه گیری پیچیدگی تا حدودی ذهنی را ارائه می دهد. با حذف نیاز به شمارش عناصر داده ، اندازه گیری را ساده می کند.

ارزیابی عملکرد مهندسی - عناصر (نام متغیرها) و عملگرها (به عنوان مثال ، حساب ، برابری / نابرابری ، بولی) محاسبه می شوند. این تغییر عملکرد محاسباتی را برجسته می کند. هدف مشابه اقدامات پیچیدگی Halstead بر اساس عملگر / عملوند است.

اندازه گیری - Bang معیار عملکرد را بر اساس دوازده شمارش اولیه (ساده) تعریف می کند که Bang را تحت تأثیر قرار می دهند یا نشان می دهند ، تعریف شده به عنوان "اندازه گیری عملکرد واقعی که توسط کاربر درک می شود". اندازه گیری انفجار ممکن است در ارزیابی مقدار واحد نرم افزار از نظر میزان عملکرد مفید آن مفید باشد ، گرچه شواهد کمی در ادبیات چنین کاربردی وجود دارد. استفاده از معیار Bang می تواند هنگامی که مهندسی مجدد (به صورت کامل یا قطعه ای) در نظر گرفته می شود اعمال شود، همان طور که در تعمیر و نگهداری سیستم های عملیاتی - یک نمای کلی بحث شده است.

نقاط ویژگی - تغییراتی را برای بهبود کاربرد در سیستم های دارای پردازش داخلی قابل توجه (به عنوان مثال ، سیستم عامل ها ، سیستم های ارتباطی) اضافه می کند. این اجازه می دهد تا حساب کارکردهایی را که توسط کاربر به راحتی قابل درک نیستند اما برای عملکرد صحیح ضروری هستند ، حساب کنید.

Weighted Micro Function Points - یکی از مدل های جدید (۲۰۰۹) که ارزیابی عملکرد را با استفاده از وزنه های ناشی از پیچیدگی جریان برنامه ، واژگان عملوند و اپراتور ، استفاده از اشیا و الگوریتم تنظیم می کند.

با توجه به توضیحات، تعاریف و موارد استفاده Function point، ترجمه ارزیابی عملکرد مناسب است.

2- در معنی و مفهوم و رابطه «تضمین کیفیت نرم افزار» و «کیفیت نرم افزار» بحث نمائید.

تضمین کیفیت نرم افزار عبارت است از نظارت بر روند مهندسی نرم افزار و روش هایی که برای اطمینان یافتن از کیفیت آن مورد استفاده قرار می گیرند. روش هایی که بدین منظور ایجاد شده اند بسیار زیاد و متنوع هستند. روش های تضمین کیفیت نرم افزار کل فرایند توسعه نرم افزار را در برمی گیرند. از جمله مراحل توسعه نرم افزار می توان به تعریف نیازمندی ها، طراحی نرم افزار، کد نویسی، بازبینی کد، مدیریت پیکربندی نرم افزار، تست، مدیریت پخش، یکپارچه سازی محصول اشاره کرد. تضمین کیفیت نرم افزار به اهداف، الزامات، توانایی ها، فعالیت ها، اندازه گیری ها و ارزیابی ها سامان می بخشد.

بخش عمده مهندسی اندازه گیری است پس باید معیارهایی برای اندازه گیری داشته باشیم. برای اینکه یک پروژه به درستی انجام شود، انتخاب فرایند و محصولی که باید در نهایت تهیه شود، به فرایند اندازه گیری داده می شود، باید با امکانات موجود محصول تولید شود، انواع معیارها از دو نظر بررسی می شوند: اندازه سیستم چقدر باشد؟ عملکرد سیستم چگونه باشد؟

با توجه به این اندازه ها باید زمان و هزینه را مشخص کرد.

دلایل اندازه گیری:

ارزیابی پروژه در حال انجام و تبدیل به کمیت تا بتوان کیفیت پروژه را ارزیابی کرد.

کنترل ریسک هایی که ممکن است در پروژه ایجاد شود و مهار مشکلاتی مانند: تأخیر زمان، افزایش هزینه، کاهش کیفیت محصول.

اگر پروژه ها به خوبی مدیریت شود، می توان مشکلات را پیش بینی کرد و از رسیدن به نقاط بحرانی جلوگیری کرد.

کنترل جریان کار و ارزیابی و سنجش تیم پروژه

برای داشتن فرایند مؤثر باید فرایندی بکار بگیریم که در زمان مشخص و با هزینه پیش‌بینی‌شده، محصول باکیفیت تولید کند.

تعداد خطای کشف نشده قبل از تحویل نرم‌افزار

تعداد خطای کشف‌شده قبل از تحویل نرم‌افزار

تعداد خطای کشف‌شده توسط کاربر نهایی

چه میزان کار انجام‌شده؟

چه میزان نیرو مصرف‌شده؟

از زمان چگونه استفاده‌شده؟

باید برای هر سازمان معیارهای مناسب با آن سازمان برای اندازه‌گیری انتخاب شود.

در مبحث مهندسی نرم‌افزار، کیفیت نرم‌افزار به دو رده مرتبط اما مجزای زیر اشاره دارد:

کیفیت عملیاتی نرم‌افزار: (Software Functional Quality) شاخصی جهت نشان دادن میزان تطابق نرم‌افزار با نیازمندی‌های عملیاتی تعریف‌شده برای نرم‌افزار.

کیفیت ساختاری نرم‌افزار: (Software Structural Quality) که منعکس‌کننده میزان دستیابی به نیازمندی‌های غیرعملیاتی مانند استحکام (Robustness) و قابلیت نگهداری (Maintainability) نرم‌افزار است.

بسیاری از جنبه‌های کیفیت ساختاری نرم‌افزار تنها با تحلیل و بررسی ساختار درونی و کد آن در سطح واحد، سطح تکنولوژی و سطح سیستم بررسی می‌شود. اما برخی خصوصیات ساختاری مثل قابلیت استفاده بودن فقط به‌صورت پویا قابل ارزیابی می‌باشند. جنبه‌های دیگر مثل قابلیت اطمینان ممکن است علاوه بر نرم‌افزار، سخت‌افزار را نیز در برگیرد. پس می‌توان آن را به‌صورت ایستا و پویا ارزیابی کرد. کیفیت عملیاتی نرم‌افزار معمولاً به‌صورت پویا بررسی می‌شود اما می‌توان بررسی‌های ایستا هم برای آن در نظر گرفت.

سوالات پایان ترم

۱- چگونه می‌توان در محیط دانشگاه تیم‌های دانشجویی برای تولید «مؤثر» نرم‌افزار تشکیل داد؟

برای ساخت یک نرم‌افزار، مهم‌ترین عامل نیروی انسانی است. از طرفی معیار اصلی سنجش پروژه‌های دانشجویی تشکیل تیم‌های درست و کارآمد می‌باشد. موارد قابل توجه در تیم‌های نرم‌افزاری، مدیریت، سازمان‌دهی، همکاری و انگیزه می‌باشد. افراد باید با توجه به توانایی خود برای کار در گروه‌ها انتخاب شوند و تمام تمرکز افراد روی یک قسمت از کار مثلاً کد نویسی، نباشد.

موارد زیر باید مورد توجه قرار گیرد:

پیدا کردن راه حل برای مشکلات- طول عمر تیم- کیفیت مورد نیاز و قابل اطمینان بودن سیستم- همکاری تیم- تعیین و مدیریت هزینه‌ها- مدیریت زمان.

یک نکته قابل توجه این است که بکار گرفتن فرایندی مؤثر است که در زمان مشخص و با هزینه پیش‌بینی شده، محصول باکیفیت تولید کند. در پروژه‌های دانشجویی، مدیریت پروژه باید به‌طور صحیح صورت گیرد و محصولی که قرار است تولید شود باید با به‌کارگیری درست منابع در اختیار و زمان معین، تحویل داده شود.

موارد مورد نیاز برای برنامه‌ریزی پروژه : تعیین محدوده پروژه، تعیین امکان‌سنجی، تجزیه و تحلیل ریسک، تعیین منابع مورد نیاز پروژه، تخمین هزینه و زمان‌بندی توسعه پروژه.

۲- نقش «مهندسی نرم‌افزار پیشرفته» برای تأثیرگذاری در تحقق شعار سال «تولید، پشتیبانی‌ها و مانع زدایی‌ها» چه می‌تواند باشد؟

کارشناسان همواره بر این باور هستند که اساس اقتصاد، بخش تولید است، چراکه اگر تولید یک کشور پویا و قدرتمند باشد اقتصاد آن کشور نیز ثبات می‌یابد و شکوفا می‌شود. در یک کلام کلید کسب ثروت و فعالیت، تولید می‌باشد. در صورت شکل‌گیری و سامان یافتن تولید داخلی تحولات بنیادینی در اقتصاد ملی و در نتیجه در معیشت مردم و به‌خصوص در حل مسائل و مشکلات اقتصادی زندگی رخ خواهد داد. سال ۱۴۰۰، سال تولید، پشتیبانی‌ها، مانع زدایی‌ها نام‌گذاری شده است و پرواضح است که برای تحقق این شعار بایستی به کلیه عوامل مؤثر چه سخت‌افزاری و چه نرم‌افزاری در تولید توجه شود. در تبیین عوامل تولید نظریات مختلفی وجود دارد ولیکن در تمام سناریوهای مربوط به تعیین عوامل مؤثر در تولید، سه عامل کار و سرمایه و امکانات به‌عنوان اصلی‌ترین عوامل مطرح بوده و هستند. که این سه عامل را می‌توان به‌راحتی با مهندسی نرم‌افزار مورد بررسی قرارداد. به این گونه که با الگوریتم‌های موجود بهترین راهکار و عملیات را برای مدیریت این سه بخش پیدا کرد. پس با مهندسی نرم‌افزار ما می‌توانیم منابع موجود را با کمترین هزینه و صرف کمترین زمان به بهترین نتیجه که شامل شناسایی و مدیریت و پیاده‌سازی است رسید. به این گونه که با الگوریتم‌های موجود ابتدا مسئله تولید که شامل طراحی خط تولید و یا المان‌های مورد نیاز تولید و همچنین نیازهای کارفرما از آن خط تولید را ساده‌سازی کنیم در گام بعد به جمع‌آوری منابع و قطعات مورد نیاز بپردازیم که همه این‌ها به‌صورت نرم‌افزاری و بر اساس مشاهدات عینی و بررسی موضوع از نزدیک و جلسات پرسش و پاسخ با کارفرما و مسئولین زیربط انجام جمع‌آوری می‌شود. در اینجا باید به این موضوع پرداخت که موانع ذهنی، اجتماعی و فرهنگی اساسی‌ترین موانع پیش روی تولید است که موانع فنی، ساختاری و قانونی حاصل موانع ذهنی می‌باشد، به‌طوری‌که اگر حمایت و پشتیبانی از تولید، تسهیل و تسریع صدور مجوزهای کسب و کار، روانی و راحتی امور مربوط به تولید؛ باور و اعتقاد سیاست‌گذاران و مجریان امر باشد، ضوابط، مقررات، قوانین و سیاست‌های موازی، دست‌وپا گیر، محرومیت‌زا و منع‌کننده تولید هرگز وضع نخواهد شد. بر این اساس موانع ذهنی به دو بخش موانع ذهنی و موانع غیر ذهنی یا واقعی تقسیم می‌شود. در بخش موانع ذهنی در وهله اول باید مردم‌باور کنند که ما می‌توانیم کالای خوب باکیفیت داخلی بسازیم و تولیدکنندگان نیز بر این باور برسند که می‌توانند کالایی تولید کنند که نه تنها در داخل کاربرد داشته باشد، بلکه به خارج از کشور نیز صادر شود و مورد استفاده مردم دیگر کشورها هم قرار گیرد و قابل رقابت در بازارهای جهانی باشد. علیرغم تمامی پیشرفت‌های فناورانه و ظهور ماشین‌ها پس از انقلاب صنعتی و ظهور کامپیوترها و ربات‌ها پس از عصر اطلاعاتی و شبکه‌ای و شکل‌گیری فضای مجازی کسب و کار در کنار فضای واقعی کسب و کار، هنوز سه عامل سرمایه و کار و امکانات جزو عوامل مؤثر و غالب تحقق تولید مطرح می‌باشند.

۳- تجزیه و تحلیل نمائید مشکل اجزای نظام آموزشی از ابعاد مختلف چیست که دانشجویان دوره‌های کارشناسی و کارشناسی ارشد در این حوزه قادر به تولید نرم‌افزار و ارائه خروجی مطلوب نمی‌باشند. سپس، برای حل مشکل کارآمدی دانشجویان مهندسی نرم‌افزار برای مشارکت در تولید نرم‌افزار در دوران دانشجویی و یا پس از آن پیشنهاداتی را ارائه نمائید. (راهنمایی: تلاش نمائید بعضی از فرایندهای نرم‌افزار مورد مطالعه درس را با اصطلاحاتی جزئی برای کارآمدی در محیط دانشگاه مناسب‌سازی نمائید)

نظام آموزشی در راستای اجتماعی کردن دانش‌آموختگان تأسیس شده و نتیجه آن باید تولید افرادی مفید و منطبق با انتظارات جامعه باشد، همچنین یکی از اشتباهات سنتی نظام آموزشی از دیرباز اصرار روی حفظیات است. آن‌هم اطلاعاتی که به هیچ دردی نه در زمان تحصیل در مدرسه و نه تحصیل در دانشگاه و نه در زندگی عادی نمی‌خورد و خیلی زود فراموش می‌شود. عصر اصرار بر حفظیات گذشت. عصر مهارت‌آموزی فرارسید. اشتباه دیگر نظام آموزشی این است که به دانش‌آموختگان یاد می‌دهند که در مقابل مدرسین، سکوت کنند و تصور دانشجویان این است که اساتید باید گوینده باشند و آن‌ها شنونده باشند. یکی دیگر از مشکلات این است که هر استاد پیرامون مطالبی که خود می‌داند سیلابس آموزشی درسی را می‌چیند که این موضوع باعث می‌شود دانشجو اطلاعات پراکنده‌ای درمورد مسائلی مختلف به دست بیاورد که هیچ‌بدرد آن نمی‌خورد. مشکل بعدی آپدیت نبودن سرفصل‌های درسی با تکنولوژی‌های روز دنیا به این‌گونه که هنوز در برخی از دانشگاه‌ها مطالبی تدریس می‌شود که مربوط به ده‌ها سال پیش می‌باشد و این باعث دلسردی دانشجو نسبت به آموختن می‌باشد و همچنین مدرک گرا بودن در ایران باعث شده است که دانشجو به هر نحوی فقط به دنبال گرفتن مدرک و نه آموختن علم خاصی باشد. دیدگاه دیگر پیرامون این ایرادها این است که برای انجام یک پروژه نیاز به داشتن علوم مختلف دارد که از جمله‌ی آن‌ها علم مهندسی نرم‌افزار، برنامه‌نویسی، مدیریت پروژه و ... می‌باشد و اغلب دانشجویان روی یک یا دو مورد اجماع اطلاعاتی دارند و باقی علوم را به دلایل مختلف که مهم‌ترین آن‌ها نبودن بازار کار خوب در کشور پیگیری و دنبال نمی‌کنند به همین دلیل اغلب این پروژه‌ها به شکست ختم می‌شود و یا ناکام باقی می‌مانند.

در مورد مهندسی نرم‌افزار عدم وجود یا کمبود شرکت‌های مشاوره‌ای برای انتخاب متدولوژی و معیارهای سنجش درست و انتخاب فرایندها، باعث می‌شود تا دانشجویان با دانش و تجربه کم و غیرعملیاتی این کار انجام دهند.

برای پیشرفت کشور و انجام پروژه‌های کارآمد و رسیدن به سطح خوب و قابل قبولی در زمینه تولید نرم‌افزار باید انجام پروژه‌ها خیلی بیشتر از پیش در دانشگاه‌ها انجام شود و سیاست دانشگاه موافق این موضوع باشد.

دانشجویان با تشکیل تیم‌های درست و استفاده صحیح از منابع در اختیار و استفاده از تکنیک‌های مهندسی برای هر چه بهتر شدن محصول نهایی تلاش کنند و سعی کنند برای ورود به بازار کار و تولید نرم‌افزارهای قابل استفاده، آموزش و تجربه کافی را کسب کنند.