

۱- در معنی و مفهوم علمی Function Point و ترجمه مناسب آن به فارسی بحث نمائید؟

برای عبارت function point معنی فارسی مناسبی در لغت نامه و فرهنگ اصطلاحات کامپیوتر و نرم افزار درج نشده است ولی در <https://translate.google.com> عبارت function point به مفهوم نقطه تابع، نقطه عملکرد ترجمه شده و با توجه به اینکه مفهوم هر عبارتی در جمله خود را نشان می‌دهد بیشتر به نقاط کاربردی، میزان عملکرد سیستم یا میزان عملکرد محصول نزدیک است.

function point (میزان عملکرد یک سیستم یا محصول) یک "واحد اندازه گیری" برای بیان میزان عملکرد تجاری است که یک سیستم اطلاعاتی (به عنوان یک محصول) در اختیار کاربر قرار می‌دهد. Function points برای محاسبه اندازه گیری میزان عملکردی (FSM) functional size measurement نرم افزار استفاده می‌شود. هزینه (به دلار یا ساعت) یک واحد از پروژه های گذشته محاسبه می‌شود.

FP ها اندازه نرم افزار را اندازه گیری می‌کنند. آنها به طور گسترده ای به عنوان یک استاندارد صنعتی برای اندازه عملکرد پذیرفته شده اند.

برای اندازه گیری نرم افزار بر اساس FP، از سال ۲۰۱۳ چندین استاندارد شناخته شده و/یا مشخصات عمومی به وجود آمده اند. این موارد هستند -

ISO Standards

- **COSMIC** – ISO/IEC ۱۹۷۶۱:۲۰۱۱ Software engineering. A functional size measurement method.
- **FiSMA** – ISO/IEC ۲۹۸۸۱:۲۰۰۸ Information technology - Software and systems engineering - FiSMA ۱.۱ functional size measurement method.
- **IFPUG** – ISO/IEC ۲۰۹۲۶:۲۰۰۹ Software and systems engineering - Software measurement - IFPUG functional size measurement method.
- **Mark-II** – ISO/IEC ۲۰۹۶۸:۲۰۰۲ Software engineering - Ml II Function Point Analysis - Counting Practices Manual.
- **NESMA** – ISO/IEC ۲۴۵۷۰:۲۰۰۵ Software engineering - NESMA function size measurement method version ۲.۱ - Definitions and counting guidelines for the application of Function Point Analysis.

FP به چه معناست؟ یک (FP) جزء توسعه نرم افزار است که به تقریب هزینه توسعه در مراحل اولیه کمک می‌کند. این فرآیندی است که عملکردهای مورد نیاز و پیچیدگی آنها را در یک نرم افزار تعریف می‌کند تا اندازه و محدوده نرم افزار پس از تکمیل برآورد شود.

یک FP اندازه نرم افزار را با کمک طراحی منطقی و کاربرد عملکردها مطابق با نیاز کاربر محاسبه می کند. همچنین به تعیین عملکرد تجاری یک برنامه نرم افزاری کمک می کند. میزان عملکرد یک سیستم یا محصول (FP) دارای چندین مزیت از جمله افزایش بهره وری و کاهش ریسک تورم کد ایجاد شده است. میزان عملکرد سیستم را می توان از نیازهای یک نرم افزار استخراج کرد و در مراحل اولیه توسعه نرم افزار، قبل از تعیین خطوط واقعی کد، تخمین زد. تعداد (FP) ها در یک کد به پیچیدگی تابع بستگی دارد.

هر نرم افزار توسعه یافته یا پیاده سازی شده برای انجام مجموعه ای از عملکردهای تجاری طراحی شده است. تجزیه و تحلیل نقطه تابع شامل استفاده از یک معیار استاندارد برای تعیین پیچیدگی و اندازه کلی هر برنامه در یک سیستم است. یک نقطه عملکرد واحد استاندارد اندازه گیری برای تعریف عملکرد تجاری است که به کاربر واقعی ارائه می شود.

۲- در معنی و مفهوم و رابطه «تضمین کیفیت نرم افزار» و «کیفیت نرم افزار» بحث کنید؟

کیفیت نرم افزار یک مفهوم انتزاعی است. تعریف آن دشوار است، اما نبود مفهوم کیفیت نرم افزار را می توان مشاهده کرد. بنابراین، برای بهبود کیفیت نرم افزار، ابتدا باید تعریف کیفیت نرم افزار را درک کنیم. ویکی پدیا کیفیت نرم افزار را به شرح زیر توصیف می کند:

"در زمینه مهندسی نرم افزار، کیفیت نرم افزار اندازه گیری می کند که نرم افزار چقدر خوب طراحی شده است (کیفیت طراحی)، و چقدر نرم افزار با آن طراحی مطابقت دارد (کیفیت انطباق). اغلب به عنوان "تناسب برای هدف" یک نرم افزار توصیف می شود."

بسته به دیدگاه (توسعه دهنده، کاربر نهایی، مدیریت...)، تغییرات زیادی در تعریف کیفیت نرم افزار وجود دارد، اگر تعریف «مناسب بودن برای هدف» بررسی کنید، این سوال را مطرح می شود که آیا نرم افزار به هدف خود عمل می کند یا خیر، یا اینکه اصطلاحات مردمی، "آیا کاری را که باید انجام دهد انجام می دهد؟ یا خیر؟" اینها ویژگی هایی هستند که ما به عنوان کاربران نهایی می بینیم. کیفیت طراحی و کیفیت انطباق با آن طراحی به جنبه های داخلی نرم افزار مربوط می شود که ممکن است برخی از آنها را ببینیم، مانند ناوبری رابط کاربری، قرار دادن کنترل ها و غیره. موارد دیگر را معمولاً نمی بینیم، مانند معماری کد، کیفیت کد و امنیت کد. به همین دلیل است که وقتی با مشتریان برای بهبود کیفیت نرم افزارشان کار می کنیم، با بررسی عمیق تر در معنای کیفیت نرم افزار در زمینه و اهداف آنها شروع می کنیم. این پایه و اساس همه تعاملات مشاوره است.

کیفیت در مهندسی نرم افزار

به طور گسترده تر، تعریف کیفیت نرم افزار از "مناسب بودن برای هدف" برای ارضای نیازها اشاره دارد. اما الزامات چیست؟ نیازمندی‌ها که در اصطلاح Agile امروزی داستان‌های کاربر نیز نامیده می‌شوند، می‌توانند به دو دسته کاربردی و غیر کاربردی دسته‌بندی شوند. الزامات عملکردی به عملکردهای خاصی اشاره دارد که نرم افزار باید قادر به انجام آنها باشد. به عنوان مثال، توانایی چاپ بر روی چاپگر HP Inkjet ۲۳۳۰ یک نیاز کاربردی است. با این حال، فقط به این دلیل که نرم افزار عملکرد خاصی دارد یا کاربر می‌تواند کار خود را با استفاده از نرم افزار انجام دهد، به این معنی نیست که نرم افزار از کیفیت خوبی برخوردار است. احتمالاً موارد زیادی وجود دارد که از نرم‌افزار استفاده کرده‌اید و آن کاری را که قرار بود انجام دهد، مانند پیدا کردن یک پرواز یا رزرو هتل، انجام داده‌اید، اما فکر می‌کنید کیفیت پایینی دارد. این به دلیل "چگونگی" اجرای تابع است. نارضایتی از "چگونه" نشان دهنده برآورده نشدن الزامات غیرعملکردی است.

برای این منظور، سازمان بین‌المللی استاندارد ISO (ISO ۲۵۰۱۰) را به عنوان مدلی برای تعیین الزامات غیرعملکردی توسعه داد. مدل نشان داده شده در زیر دسته بندی نیازمندی‌های غیرعملکردی را نشان می‌دهد.

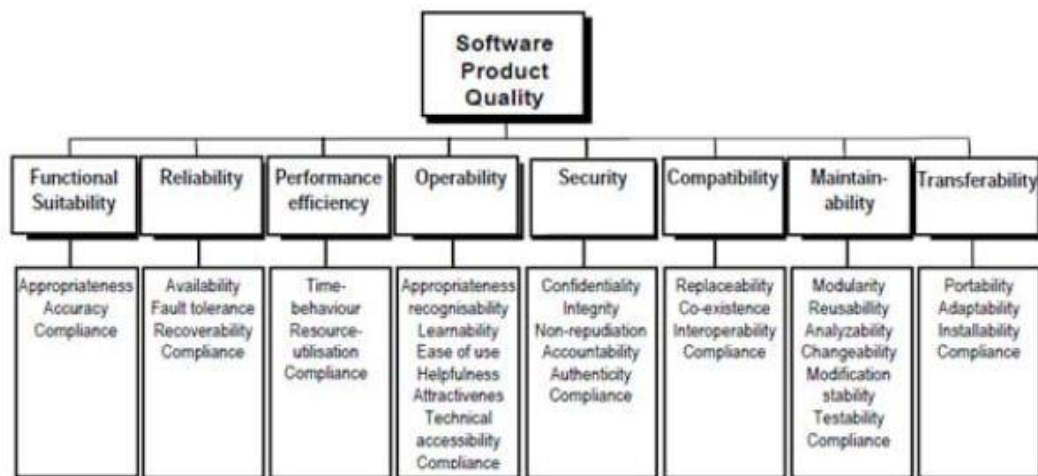


Figure 1- [1] ISO/IEC CD 25010 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality model and guide, 2011.

در نگاه اول، ممکن است فکر کنید که سمت چپ ترین مشخصه، مناسب بودن عملکردی، معادل یک نیاز کاربردی است، اما اینطور نیست. ویژگی‌های فرعی کامل بودن عملکرد، صحت عملکردی و تناسب عملکردی برای ویژگی‌های آن توابع اعمال می‌شود. برای مثال، کامل بودن عملکردی به عنوان درجه ای تعریف می‌شود که مجموعه توابع تمام وظایف مشخص شده و اهداف کاربر را پوشش می‌دهد. بنابراین، "Printer ۲۳۳۰ Print from HP Inkjet" می‌توانست از نقطه نظر نیاز عملکردی پیاده سازی شود. اما چگونه اجرا شد؟ برای همه گزینه ها کامل بود؟ چاپ دو رو هم داشت؟ اگر

نه، پس ممکن است از نظر کامل بودن عملکرد کیفیت خوبی نداشته باشد. بنابراین، در حالی که الزامات غیر کاربردی ممکن است ویژگی های اصلی و ویژگی های فرعی خود را داشته باشند، آنها می توانند افکار شما را در مورد نیازهای عملکردی نیز تحریک کنند.

چرا مولفه های با کیفیت غیر کاربردی مهم هستند؟

ارضای الزامات غیرعملکردی مانند عملکرد، سهولت استفاده و یادگیری ابتدا نیاز به مشخص کردن و تعریف دارد. تنها پس از تعریف، می توان آنها را ارضا کرد و ارضای آنها می تواند حتی دشوارتر از برآوردن نیازهای عملکردی باشد. پس این برای شما چه معنایی دارد؟ بیایید با استفاده از همین مثال چاپی، ویژگی های غیر کاربردی زیر را بررسی کنیم:

تناسب عملکردی (مناسب بودن عملکرد) - آیا این عملکرد تکمیل وظایف و اهداف کاربر را تسهیل می کند؟ اگر کاربر نمی خواهد روی آن چاپگر چاپ کند یا می خواهد یک PDF چاپ کند اما این گزینه ها به او داده نمی شود، شاید نه.

کارایی عملکرد (رفتار زمانی) - آیا عملکرد چاپگر در عرض سه ثانیه پاسخ می دهد؟

سازگاری (عملکرد متقابل) - آیا کاربر می تواند روی انواع شبکه ها و چاپگرها و روی رایانه هایی با سیستم عامل های مختلف (ویندوز و مک) چاپ کند؟

قابلیت استفاده (قابلیت یادگیری) - آیا کاربر می تواند نحوه چاپ را بفهمد یا به یک دانشمند نیاز دارد؟

سهولت استفاده (راهنمای حساس به متن) - تشخیص، اشتباه کردن و سپس بازیابی برای کاربر چقدر آسان است؟

قابلیت اطمینان (بازیابی) - وقتی پرینتر در وسط چاپ یک کار قطع می شود، به کاربر اطلاع داده می شود؟

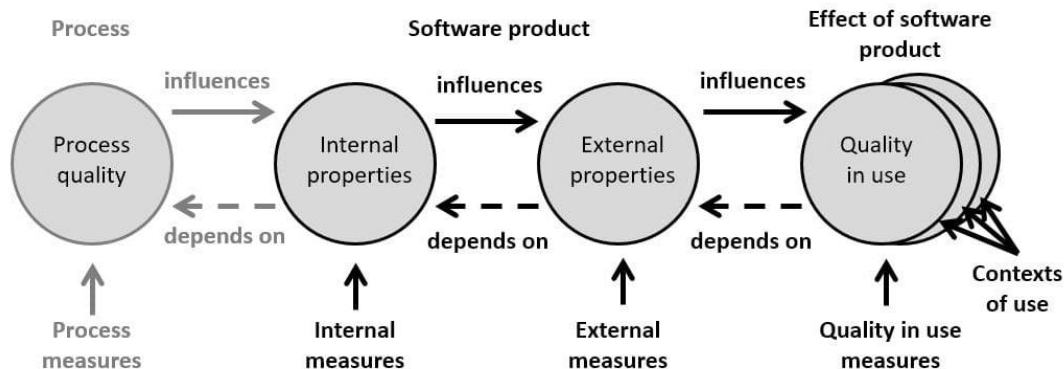
امنیت (غیر انکار) - آیا سابقه ای وجود دارد که چاپگر فایل را با موفقیت چاپ کرده باشد؟

قابلیت نگهداری (آزمایش پذیری) - آیا می توان معیارهای تست را برای عملکرد چاپ مشخص کرد؟

قابلیت حمل (انطباق پذیری) - آیا نرم افزار می تواند به طور خودکار با مدل های چاپگر جدید یا به روز رسانی نرم افزار درایور چاپگر سازگار شود؟ آیا عملکرد چاپ می تواند میانبرهایی را برای کاربران بسیار پیچیده فراهم کند؟

اکنون که درک درستی از الزامات غیرعملکردی داریم، چرخه عمر کیفیت را در نمودار زیر بررسی کنیم. با نگاهی به سه دایره زیر، کیفیت داخلی نشان دهنده کیفیتی است که شما نمی بینید و با ویژگی های داخلی مانند کیفیت کد اندازه گیری می شود. کیفیت خارجی همان چیزی است که در بالا در مدل کیفیت غیرعملکردی بحث کردیم و معمولاً با اجرای واقعی کد و بررسی رفتار نرم افزار اندازه گیری می شود.

Software Quality Lifecycle



(مثال: یک محصول نرم افزاری از نظر عملکرد صحیح را در نظر بگیرید. یعنی تمام وظایفی را که در سند SRS مشخص شده انجام می دهد. اما، دارای یک رابط کاربری تقریباً غیرقابل استفاده است. حتی اگر از نظر عملکردی درست باشد، نمی توانیم آن را یک محصول با کیفیت بدانیم.)

برای تضمین کیفیت نرم افزار نیز تعاریف زیادی شده است که می توان به موارد ذیل اشاره نمود

تضمین کیفیت نرم افزار (SQA) فرآیندی است که تضمین می کند کلیه فرآیندها، روش ها، فعالیت ها و موارد کاری مهندسی نرم افزار پایش و مطابق با استانداردهای تعریف شده هستند. این استانداردهای تعریف شده می تواند یکی یا ترکیبی از هر یک مانند ISO ۹۰۰۰، مدل CMMI، ISO ۱۵۵۰۴ و غیره باشد.

تضمین کیفیت نرم افزار عبارتست 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 که هریک انطباق با چند مورد از استانداردها (مثل ایزو ۹۰۰۰ و یک مدل مثل CMMI) را بررسی و تضمین می کنند.

SQA تمامی فرآیندهای توسعه نرم افزار را از تعریف نیازمندی ها گرفته تا کدنویسی تا زمان انتشار را در بر می گیرد. هدف اصلی آن تضمین کیفیت است.

Software Quality Assurance Plan طرح تضمین کیفیت نرم افزار که به اختصار SQAP نامیده می شود، شامل رویه ها، تکنیک ها و ابزارهایی است که برای اطمینان از همسویی یک محصول یا خدمات با الزامات تعریف شده در SRS (software requirement specification) (مشخصات نیازمندی نرم افزار) به کار می رود.

روش‌های تضمین کیفیت نرم‌افزار کل فرایند توسعه نرم‌افزار را دربرمی‌گیرند. از جمله مراحل توسعه نرم‌افزار می‌توان به تعریف نیازمندی‌ها، طراحی نرم‌افزار، کدنویسی، بازمینی کد، مدیریت پیکربندی نرم‌افزار، تست، مدیریت پخش، یکپارچه سازی محصول اشاره کرد. تضمین کیفیت نرم‌افزار به اهداف، الزامات، توانایی‌ها، فعالیت‌ها، اندازه‌گیری‌ها و ارزیابی‌ها سامان می‌بخشد.



این طرح مسئولیت های SQA یک تیم را مشخص می کند، مناطقی را که نیاز به بررسی و ممیزی دارند فهرست می کند. همچنین محصولات کاری SQA را شناسایی می کند.

تضمین کیفیت نرم‌افزار برحسب (ISO ۱۵۵۰۴/IEC) یک فرایند حمایتی است که باید به وسیله طرح‌های از پیش تعیین شده و ISO ۱۵۵۰۴ کیفیت را در تک تک محصولات، فعالیت‌ها و فرایندها به‌طور مستقل تضمین کند.

پس از مباحث مطرح در چگونگی ایجاد نرم‌افزار و بحث‌های مربوطه؛ یکی از چالش‌های موجود بررسی مفهوم چیستی کیفیت نرم‌افزار و متعاقب آن معیارهای سنجش و روش‌های تضمین کیفیت آن است. صحبت از کیفیت نرم‌افزار و بررسی و سنجش آن تنها از طریق پاسخگویی به سوالات زیر و مشخص کردن روشی برای رسیدن به این پاسخ امکانپذیر است:

۱. تعریف صریح و دقیق معنا و مفهوم «کیفیت نرم‌افزار»

۲. ایجاد مجموعه ای فعالیت‌هایی که ما را از بالا بودن کیفیت محصولات نرم‌افزاری ارائه شده مطمئن می‌کنند.

۳. اجرا کردن فعالیت‌های کنترل و ضمانت کیفیت در همه پروژه‌های نرم‌افزاری

۴. استفاده از ابزارها در راستای توسعه استراتژی‌های بهبود فرایندهای نرم‌افزار و در نتیجه باکیفیت شدن محصول نهایی

سند طرح SQA شامل بخش‌های زیر است:

بخش هدف، بخش مرجع، بخش مدیریت پیکربندی نرم افزار، بخش گزارش مشکل و اقدامات اصلاحی، بخش ابزارها، فناوری‌ها و روش‌ها، بخش کنترل کد، سوابق: بخش جمع‌آوری، نگهداری و نگهداری متدولوژی تست

فعالیت‌های SQA:

(۱) ایجاد یک برنامه مدیریت SQA:

مهمترین فعالیت شامل تنظیم یک برنامه مناسب در مورد نحوه اجرای SQA در پروژه است. همراه با رویکرد SQA که قرار است دنبال شود، چه فعالیت‌های مهندسی انجام خواهد شد، و همچنین شامل اطمینان از داشتن ترکیب استعداد مناسب در تیم خود می‌شود.

(۲) تنظیم نقاط بازرسی: تیم SQA پست‌های بازرسی مختلفی را بر اساس آن‌ها کیفیت فعالیت‌های پروژه را در هر مرحله ایست بازرسی/پروژه ارزیابی می‌کند. این امر بازرسی منظم کیفیت و کار طبق برنامه را تضمین می‌کند.

(۳) از تکنیک‌های مهندسی نرم افزار استفاده شود: به کارگیری برخی از تکنیک‌های مهندسی نرم افزار به طراح نرم افزار در دستیابی به مشخصات با کیفیت بالا کمک می‌کند. برای جمع‌آوری اطلاعات، یک طراح ممکن است از تکنیک‌هایی مانند مصاحبه و FAST (Functional Analysis System Technique) (تکنیک سیستم تحلیل عملکردی) استفاده کند. بعداً، بر اساس اطلاعات جمع‌آوری‌شده، طراح نرم‌افزار می‌تواند با استفاده از تکنیک‌هایی مانند WBS (work breakdown structure) (ساختار شکست کار)، SLOC (source line of codes) (خط منبع کدها)، و FP (functional point) (مقیاس عملکردی) تخمین پروژه را آماده کند.

(۴) اجرای بررسی‌های فنی رسمی:

یک (FTR (Formal Technical Reviews) برای ارزیابی کیفیت و طراحی نمونه اولیه انجام می‌شود.

در این فرآیند، جلسه ای با کادر فنی برگزار می شود تا در مورد الزامات کیفی واقعی نرم افزار و کیفیت طراحی نمونه اولیه بحث شود. این فعالیت به شناسایی خطاها در مرحله اولیه SDLC کمک می کند و تلاش مجدد را در مراحل بعدی کاهش می دهد.

۵) داشتن یک استراتژی چند آزمایشی: منظور ما از استراتژی چند تستی این است که نباید به هیچ رویکرد آزمایشی تکیه کرد، در عوض، باید چندین نوع آزمایش انجام شود تا محصول نرم افزاری از همه جهات به خوبی آزمایش شود تا کیفیت بهتری تضمین شود.

۶) اجرای تبعیت از فرآیند: این فعالیت بر نیاز به رعایت فرآیند در طول فرآیند توسعه نرم افزار اصرار دارد. فرآیند توسعه نیز باید به رویه های تعریف شده پایبند باشد.

سوالات نیمه ۲

۱- چگونه میتوان در محیط دانشگاه تیمهای دانشجویی برای تولید «موثر» نرم افزار تشکیل داد؟

تولید نرم افزار کاری گروهی است. جهت تولید ارزش افزوده در کسب و کار همکاری کامل بین اعضای گروه تولید لازم است. در صورت عدم همکاری به احتمال بسیار زیاد ارزشهای مورد انتظار حاصل نخواهد شد و احتمال ناکامی و شکست بسیار زیاد وجود دارد.

از آنجایی که به مجموعه‌ای از گام‌ها برای ایجاد برنامه‌های نرم افزاری چرخه حیات توسعه نرم افزار (Software Development Life Cycle | SDLC) گفته می‌شود. می‌توان به وسیله این گام‌ها فرآیند توسعه نرم افزار را به وظایفی (Task) تقسیم کرد و این وظایف را به افراد مختلف تیم برنامه نویسی و توسعه محصول محول کرد. سپس این وظایف توسط اعضای تیم برنامه نویسی و توسعه محصول تکمیل می‌شوند و قابل ارزیابی خواهند بود. (SDLC). این مراحل ساخت نرم افزار یا همان مراحل توسعه محصول نرم افزاری به شرح ذیل می باشد:

برنامه‌ریزی (Planning)، برآورد نیازمندی‌ها و ملزومات (Requirements)، طراحی محصول نرم افزاری (Design)، ساخت محصول نرم افزاری (Build)، مستندسازی (Document)، آزمایش و تست نرم‌افزار (Test)، استقرار (Deploy)، نگهداری (Maintain)

این گام‌ها ممکن است در بسیاری از سازمانها به بخشهای کوچکتری تقسیم شوند. و از آنجایی که یک تیم برنامه نویسی گروهی از افراد را شامل می‌شود که توسعه و نگهداری نرم افزارهای کامپیوتری را بر عهده دارند. اعضای تیم برنامه نویسی و توسعه نرم افزار برای ساخت یک پروژه و محصول فنی با یکدیگر همکاری می‌کنند. هر شخص در تیم توسعه برای رسیدن به مقصود اصلی نقش دارد و باید در قبال کار خود جوابگو باشد. انجام وظایف به صورت ساختارمند و بهینه در سطح فردی، اثربخشی کلی تیم توسعه را بهبود می‌بخشد. در عین حال، هنگام برقراری ارتباط و انجام کارهای گروهی نیز اعضای تیم باید همکاری و تعامل سازنده‌ای با یکدیگر داشته باشند. برای دستیابی به این هم افزایی در ساختار تیم و توانمندسازی، بسیار مهم است که تیم‌های توسعه خصوصیت‌های ویژه‌ای داشته باشند.

سه نوع ساختار رایج برای تیم‌های برنامه نویسی وجود دارد: کلی‌گرا (Generalist)، تخصصی (Specialist)،

تیم ترکیبی (Hybrid)

هر یک از این نوع تیم‌ها ممکن است در یک پروژه عملکرد خوب داشته و در پروژه دیگر دچار مشکل شوند. برای جلوگیری از شکست تیم‌ها، باید عواملی از جمله پیچیدگی‌ها، محدودیت‌های زمانی و بودجه را در نظر گرفت و سپس بر اساس این عوامل در خصوص اینکه کدام ساختار بهتر است تصمیم‌گیری نمود. به طور کلی، تیم برنامه نویسی ممکن است به روش‌های گوناگونی سازمان‌دهی شود. برای مثال، تیم برنامه نویسی بی‌تکلف (Egoless) و تیم‌های دارای برنامه‌نویس ارشد (Chief Programmer Team) دو ساختار رایج دیگری هستند که معمولاً به کار گرفته می‌شوند.

با توجه به اینکه عوامل تاثیرگذار در انتخاب نوع ساختاری تیم برنامه نویسی در یک پروژه خاص به شرح زیرند:

سطح دشواری پروژه (پیچیدگی)، اندازه پروژه، مدت زمانی پروژه، میزان پیمانه‌ای بودن پروژه (Modularity)،

قابلیت اطمینان (Reliability) و سایر موارد

از آنجایی که بعضی از تیم‌های برنامه نویسی بی‌تکلف (ساده) هستند. و یک تیم برنامه نویسی بی‌تکلف شامل گروه‌های ده نفره یا کمتر است. کدها در میان اعضای گروه تبادل و اهدافی تعیین می‌شوند. رهبری در داخل گروه بر اساس نیازها و مهارت‌های مورد نیاز در طول یک زمان مشخص جا به جا می‌شود. کمبود ساختار در تیم برنامه نویسی بی‌تکلف می‌تواند منجر به ضعف در کارایی، اثربخشی و شناسایی خطا برای پروژه‌های با مقیاس بزرگ شود. تیم‌های برنامه نویسی بی‌تکلف بهترین عملکرد را در زمینه پروژه‌های بسیار پیچیده از خود نشان می‌دهند.

نوع دیگری از تیم برنامه نویسی تیم دارای برنامه نویس ارشد است. در یک تیم دارای برنامه نویس ارشد معمولاً از تیم‌های سه نفره تشکیل می‌شود که شامل یک برنامه نویس ارشد، برنامه نویس سطح بالا (Senior Level Programmer) و یک متصدی برنامه (Program Librarian) است. برنامه نویسان و تحلیلگران دیگری هم در صورت لزوم به تیم اضافه خواهند شد. نقطه ضعف چنین ساختاری، کمبود ارتباط و تعامل میان اعضای تیم، کمبود همکاری در انجام تسک‌ها (وظایف) و تکمیل تسک‌های پیچیده را شامل می‌شود. ساختار تیم‌های دارای برنامه نویس ارشد برای انجام وظایف یا همان تسک‌های ساده‌تر بازدهی بیشتری را از خود نشان می‌دهند. چون جریان اطلاعات در این تیم‌ها دارای محدودیت است. افراد در این تیم‌ها روحیه کاری ضعیف‌تری از خود بروز می‌دهند.

نوع دیگری از ساختار تیم‌های برنامه نویسی، تیم‌های با ایستگاه کاری مشترک هستند. این نوع ساختار خود به دو نوع تقسیم می‌شوند:

برنامه نویسی دو نفره: روش توسعه‌ای است که در آن دو برنامه نویس در یک ایستگاه کاری با هم کار می‌کنند.

برنامه نویسی گروهی: در این رویکرد توسعه نرم افزار، کل تیم روی یک موضوع به طور همزمان در یک فضای مشترک و با یک کامپیوتر کار می‌کند.

در تشکیل تیم برنامه نویسی باید به نکاتی ذیل توجه شود.

انتخاب یک تیم برنامه نویسی و توسعه نرم افزار مناسب می‌تواند به وضوح یکی از مهم‌ترین تصمیماتی باشد که در زمان راه‌اندازی یک کسب و کار گرفته می‌شود. در نهایت، این چگونگی تشکیل تیم برنامه نویسی است که موفقیت یا شکست یک کسب و کار را تعیین می‌کند. تشکیل تیم برنامه نویسی موفق و آرمانی چیزی بیش‌تر از استخدام چند کارمند است که با یکدیگر همکاری می‌کنند و پروژه مربوطه را توسعه می‌دهند. در واقع، تشکیل تیم برنامه نویسی، انتخاب شریکان تجاری است که به چشم‌اندازها و تصورات ذهنی مدیران و صاحبان پروژه زندگی می‌بخشد.

باید اطمینان حاصل شود که نقش‌ها و مسئولیت‌های لازم پیش از تشکیل تیم برنامه نویسی مشخص شده باشند. باید تعیین شود که چه کسی تیم برنامه نویسی را مدیریت خواهد کرد؟ و همین‌طور چه تعداد توسعه دهنده مورد نیاز است؟ باید به یاد داشت که فرآیند توسعه نرم افزار و همکاری و شراکت تیم برای تولید محصول می‌تواند از چند ماه تا چندین سال زمان ببرد. بنابراین، در تشکیل تیم برنامه نویسی بسیار مهم است که افراد درستی انتخاب شوند و این افراد باید تناسب لازم را هم با پروژه و هم با کسب و کار مربوطه داشته باشند.

چهار نکته مهم در هنگام تشکیل تیم برنامه نویسی اهمیت دارد .

۱. تفکر انتقادی
۲. تخصص در یک حوزه خاص
۳. پروژه‌های پیشین (پروژه‌های قبلی که اعضای بالقوه تیم برنامه نویسی اجرا کرده‌اند، کیفیت کار آن‌ها را بازتاب می‌دهند).
۴. پیشنهاد مبلغ عادلانه

جهت مدیریت تیم برنامه نویسی و توسعه نرم افزار موفق موارد ذیل را باید در نظر گرفت

۱. تحت کنترل گرفتن فرآیند استخدام در تشکیل تیم برنامه نویسی

۲. استخدام افراد مشتاق برای تشکیل تیم برنامه نویسی

۳. ارزیابی و پرورش مهارت‌های نرم

۴. فراهم کردن ابزار و محیط کاری مناسب برای اعضای تیم

۵. توجه و ارتباط فعالانه

۶. تلاش برای بهبود روابط اجتماعی اعضای تیم

۷. جلوگیری از بروز فرسودگی شغلی

۸. خودداری از اضافه کردن نیروی انسانی اضافه برای رفع سریع مشکلات

۹. تولید نرم افزاری کاربردی و کاربرپسند

۱۰. در نظر داشتن استخدام دورکاری

برای تولید یک نرم افزار کاربردی تعدادی نقش و مهارت مورد نیاز می باشد و بایستی آنها را بین اعضای گروه تقسیم بندی کرد هر چند ممکن است تمامی آنها در یک فرد جمع گردد

مهمترین نقش های موثر در فرآیند مدیریت تولید نرم افزار موارد ذیل می باشند :

مدیر کسب و کار ، تحلیلگر کسب و کار ، ذینفعان ، مدیر پروژه ، مالک محصول یا ارشد تیم اسکرام، تصمیم گیرندگان در اداره مدیریت پروژه، معمار ، مدیران پایگاه داده ، توسعه دهندگان ، تیم طراحی تجربه کاربری ، آزمونگر ها، کارکنان عملیات و نگهداری

آنچه یک برنامه نویس خوب را از یک توسعه دهنده بی نقص متمایز می سازد، زمانی اتفاق می افتد که اعضای تیم برنامه نویسی عزم خود را در دستیابی به اهداف مشترک جزم کرده باشند. در حقیقت، اکثر پروژه های توسعه نرم افزار به دلیل کمبود اعتماد به نفس جمعی و فردی و عدم اطمینان به موفقیت پروژه در همان مراحل اولیه شکست می خورند. بنابراین، آشنایی با خط مشی های نتیجه بخش و بهترین شیوه ها برای تشکیل تیم برنامه نویسی موفق از پیدا کردن افراد مناسب گرفته تا روش های مدیریت پروژه برنامه نویسی بسیار اهمیت دارد.

به علاوه اگر مایلیم این پروژه به بهترین شکل اجرا گردد ضروری است که این همکاری فراتر از مرز های گروه تولید شکل گرفته و تمامی نقشه های درگیر در فرآیند مدیریت تولید را در بر بگیرد.

مهندسی نرم افزار پیشرفته
به نام خدا
زمان تحویل: ۴۸ ساعت
نام و نام خانوادگی: اکبر حمیدی
شماره دانشجویی: ۹۹۰۱۸۲۰۶۹
ارشد مهندسی نرم افزار
دانشگاه پیام نور
رشته و مقطع تحصیلی: ارشد مهندسی نرم افزار
۱. Dropbox نرم افزار ذخیره سازی و اشتراک گذاری فایل

۲. Mendeley Desktop نرم افزار مدیریت و به اشتراک گذاری منابع تحقیقاتی

۳. Evernote نرم افزار یادداشت برداری

۴. AudioNote نرم افزار صوتی و تایپ

۵. Sumatra نرم افزار مشاهده کتاب های الکترونیکی

۶. Amazon Kindle نرم افزار کتاب یاب

۷. LectureNotes نرم افزار نت برداری

۲- نقش « مهندسی نرم افزار پیشرفته » برای تاثیر گذاری برای تحقق شعار سال « تولید ، پشتیبانی ها و مانع زدایی » چه می تواند باشد ؟

امروزه رشد روزافزون تکنولوژی به گونه ای غیرقابل انکار بر صنایع مختلف تاثیر گذاشته است. با توجه به اهمیت عامل مدیریت کیفیت جامع و فناوری اطلاعات در صنعت و توجه روزافزون بدان، نگرش های مختلفی در خصوص راه ها و ابزار رسیدن به بهره وری توسط این منابع ارائه شده است. در نتیجه هدف از این پژوهش مطالعه و بررسی تاثیر تکنولوژی پیشرفته تولید بر بهره وری با در نظر گرفتن نقش میانجی مدیریت کیفیت جامع و استفاده از فناوری اطلاعات در صنایع کوچک و متوسط می باشد.

تحقق شعار سال ۱۴۰۰ نیازمند نقش آفرینی همه مسؤولان و مردم است تا نه فقط یک پشتیبانی و مانع زدایی محدود، بلکه مجموعه ای از «پشتیبانی ها» و «مانع زدایی ها» بتواند مؤثر واقع شود و پیشرفتی همه جانبه را در عرصه «تولید» به دنبال داشته باشد.

به گزارش خبرگزاری فارس از قم، سالیانی است که جمع بندی مسائل کشور در انتخاب و اعلام شعار سال توسط رهبر انقلاب اسلامی و در لحظه مهم تحویل سال نمود دارد.

در این بین و از سال ۱۳۹۰، یک شعار اقتصادی از جمله «جهاد اقتصادی»، «حماسه اقتصادی»، «اشتغال آفرینی»، «اصلاح الگوی مصرف»، «تولید ملی، حمایت از کار و سرمایه ایرانی»، «اقتصاد مقاومتی»، «حمایت از کالای ایرانی»، «رونق تولید» و «جهش تولید»، مبنا و محور فعالیت ها در قریب به اتفاق این سال ها بوده و همچنان هست، ضمن آن که تکرار معنادار کلیدواژه «تولید» در اعلام شعار سال های اخیر کشور توسط رهبر بصیر انقلاب اسلامی به وضوح از اهمیت این کلیدواژه سخن می گوید.

اکنون سال ۱۴۰۰ به عنوان آخرین سال قرن چهاردهم و در آستانه سال ۱۴۰۱ به عنوان آغاز قرن جدید، باز هم کلیدواژه «تولید» را بر پیشانی خود دارد و با عنوان «تولید، پشتیبانی ها، مانع زدایی ها» مزین شده تا مشخص شود اصلاح مسائل مرتبط با تولید، در یک مدت محدود خلاصه نمی شود و باید در این زمینه، نگاهی بلندمدت داشت.

باید گفت شعار «تولید، پشتیبانی ها، مانع زدایی ها» از چند منظر قابل بررسی و تحلیل است، نخست آن که کلیدواژه «تولید» را نباید صرفاً یک موضوع اقتصادی دانست، بلکه باید همه پیشرفت های فرهنگی، اجتماعی، امنیتی و حتی سیاسی را در تحقق عینی تولید به مثابه یک راهبرد تلقی کرد، به بیان دیگر باید توجه داشت که تولید به عنوان یک کلیدواژه در همه عرصه ها - و نه فقط عرصه اقتصادی - مطرح است.

و اما عبارت «تولید، پشتیبانی ها، مانع زدایی ها» کاملاً بیانگر این نکته است که «پشتیبانی» و «مانع زدایی» در کنار هم می توانند مشکلات گوناگون عرصه تولید را مرتفع سازند، یعنی چه بسا پشتیبانی و حمایت ها از عرصه تولید به دلیل وجود برخی موانع تأثیر لازم را نداشته باشد، ضمن آن که رفع موانع نیز بدون پشتیبانی های مستمر از تولید، کارا نخواهد بود، اگرچه خود رفع موانع، نوعی از پشتیبانی و حمایت است

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 در درس را با اصلاحاتی جزئی برای کارآمدی در محیط دانشگاه مناسب سازی نمائید)

با توجه به بررسی های به عمل آمده عوامل متعددی باعث بوجود آمدن مشکلات در اجزای نظام آموزشی می گردد و مطالب کلانی می توان در این خصوص و نظرات متعددی جهت برطرف نمودن آنها به نگارش در آورد .

تأثیر عوامل اقتصادی ، اجتماعی ، سیاسی، فرهنگی، قانونی، روان شناختی، تکنولوژیکی، اعتقادی و آرمانی بر سیاست گذاری نظام آموزشی:

برای این منظور ، بهتر است که نظام آموزشی به عنوان یک سیستم مورد تجزیه و تحلیل قرار گیرد. به این منظور، اول خود نظام و اجزای آن مورد بررسی قرار گرفته و سپس نظام آموزشی را بطور کامل و علمی مورد بررسی قرار می‌دهیم و در نهایت فرا سیستم‌های موثر بر آن را مشخص کنیم .

نظام عبارتست از مجموعه‌ای از اجزای بهم پیوسته که در حال تعامل باهم بوده و برای رسیدن به هدف مشترک باهم تلاش می‌کنند. حال اگر یکی از این اجزاء با بقیه همکاری نکند در کار نظام اختلال بوجود می‌آید. و برای رسیدن به هدف با مشکل مواجه می‌شود.

هر نظامی اعم از اقتصادی، ، اجتماعی ، سیاسی، آموزشی و غیره از چهار عنصر اصلی تشکیل شده است:

۱- ورودی (Input): یعنی آنچه که به سیستم وارد می‌شود.

۲- فرآیند (Process): یعنی مجموعه فعل و انفعالاتی که روی ورودی ها انجام می‌گیرد.

۳- خروجی (Out put): همان نتیجه و محصول نظام است.

۴- باز خورد (Feed back): قسمتی از خروجی است که مجدداً جذب سیستم می‌شود.

از موارد فوق ، باز خورد (Feed back) اهمیت زیادی بر خوردار است چون باعث ثبات و ماندگاری سیستم می‌شود. بر همین مبنا، نظام آموزشی را مورد بررسی قرار می‌دهیم.

الف- ورودی شامل استاد، دانشجو، معلم، شاگرد، برنامه، بودجه، فضاهای آموزشی، قوانین و مقررات، خط و مشی‌های سیاسی، اقتصادی و ...

ب- فرآیند شامل تدریس، یادگیری، پژوهش، خدمات راهنمایی و مشاوره و ...

ج- خروجی شامل ترک تحصیل کنندگان و فارغ التحصیلان است. از نظام آموزشی انتظار میرود که فارغ التحصیلانی تحویل جامعه بدهد که دارای علم، دانایی، نگرش جدید، بینش جدید، خود رهبری و در نهایت تغییر در رفتار باشند.

د- باز خورد که قسمتی از فارغ التحصیلان هستند که مجدداً جذب سیستم می‌شوند. لذا سیستم هیچ وقت نابود نمی‌شود اما تغییر شکل می‌دهد. هر نظام آموزشی که عنصر بازخورد را نداشته باشد محکوم به شکست و نابودی است و پدیده خطرناک **brain drain** دلیلی بر این ادعاست که نمونه بارز آن نظام آموزشی کشور خودمان است.

نظام آموزشی بوسیله نظام اقتصادی، اجتماعی، فرهنگی، سیاسی و حتی بین المللی احاطه شده و از آنها تأثیر می‌پذیرد. یعنی هر تغییری در نظام اقتصادی، اجتماعی، فرهنگی و سیاسی رخ بدهد در نظام آموزشی هم تأثیر می‌گذارند. البته نظام آموزشی همیشه خرده سیستمی از سیستم‌های مذکور نیست بلکه بعنوان یک نظام همکار با سایر نظام‌ها فعالیت می‌کند و تأثیر انکار ناپذیری بر آنها دارد.

عوامل موثر بر نظام آموزش عبارتند از:

- ۱- عوامل اقتصادی
- ۲- عوامل اجتماعی
- ۳- عوامل فرهنگی
- ۴- عوامل سیاسی:
- ۵- عوامل قانونی:
- ۶- عوامل روان شناختی:
- ۷- عوامل تکنولوژیکی:
- ۸- عوامل اعتقادی و آرمانی:

عوامل فوق گاهی اوقات جزو عوامل مشکل زا در نظام آموزشی محسوب می‌شوند

ارکان نظام آموزشی شامل:

- ۱- اجزای انسانی
- ۲- اجزای فیزیکی
- ۳ اجزای ساختاری
- ۴- اجزای محتوایی

با توجه به تغییرات سریعی که در توسعه روز افزون علوم و فناوری‌ها صورت می‌گیرد و با وجود اینترنت و امکان ذخیره و انتقال حجم‌های زیاد اطلاعات و انجام بخش مهمی از فعالیت‌ها در آینده توسط ربات‌های هوشمند و توقع جامعه از دانشگاه در آینده، دانشگاه‌ها باید:

- آمادگی کافی برای هماهنگی با این تغییرات داشته باشند و گرنه دچار عقب ماندگی خواهند شد.
- شرایط روز را بشناسند و با آینده نگری فعالیت کنند.
- با تحولات دنیا و نیازهای روز کشور هماهنگی داشته باشند.
- دانشگاه‌ها اصلی ترین مؤسساتی هستند که می‌توانند بر اساس واقعیت‌های هر جامعه دردهای آن را دریابند و راهکار ارائه دهند و جامعه انتظارات خاص از آنها دارد. این مأموریت را دانشگاه ها باید بپذیرند.

اشاره ای به برخی از مشکلات دانشگاه‌های ایران

- عدم برخورداری از معیارهای خاص دانشگاهی
- عدم اشتغال دانش آموختگان و مهاجرت بخشی از آنان که توانایی بالاتری دارند.
- عدم مهارت آموزی دانشجویان
- بی انگیزه شدن دانشجویان و عدم توجه به کارآفرینی و ایجاد خلاقیت در دانشجو
- افزایش کمی دانشجو به ویژه در دوره های تحصیلات تکمیلی بدون توجه به کیفیت
- عدم ارتقای بینش دانشجویان نسبت به تفکر علمی و آینده نگری
- عدم توجه کافی به دوره های فنی و حرفه‌ای، کاردانی و کارشناسی مورد نیاز کشور
- عدم توجه به نیازهای کشور از نظر تعداد و نوع افراد متخصص لازم، و رشد بی‌رویه دانشگاه هایی که معیارهای دانشگاهی ندارند.
- وابستگی زیاد به بودجه دولتی
- عدم ارتباط کافی بین جامعه و دانشگاه
- ضعف همکاری های بین المللی
- تمرکزگرایی و یکنواختی سیستم آموزش و پژوهش در کشور
- عدم توجه کافی به ارتقای اخلاق علمی
- عدم توجه دانشگاه‌ها به مسئولیت‌های اجتماعی
- عدم توجه کافی دانشگاه‌ها به مشکلات حاد اجتماعی مثل تغییر اقلیم، بحران آب، محیط زیست و الگوهای سبز و کاهش آلودگی‌ها
- عدم توجه دانشگاه‌ها به سالم ماندن انسان‌ها به ویژه سلامت اجتماعی و معنوی
- عدم ارتباط کافی بین دانشمندان دانشگاهی و سیاستگذاران

- عدم خلاقیت دانشگاه‌ها در اصلاح برنامه های آموزشی بر اساس نیازهای روز و آینده، جامع نگر نبودن و عدم توجه کافی به بین رشته ای
- عدم وجود اختیارات کافی و خلاقیت در تصمیم گیری هیأت امنای دانشگاه‌ها و رؤسای دانشگاه ها
- عدم توجه به امکان ثبت پتنت ها و استفاده معنوی و مادی از آنها
- عدم توجه کافی به علوم نوپدید
- سیاست زدگی و عدم توجه به مدیریت‌های مناسب در مقاطعی از زمان‌ها
- ارتباط ضعیف دانشگاه با بخش‌های اقتصادی، صنعتی، اجرایی و خدماتی
- ضعف در آموزش زبان فارسی
- ضعف در آموزش زبان انگلیسی
- ضعف در جامع نگری علمی و سخت گیری در حفظ درز و مرزهای ایجادشده در علوم
- گرفتار شدن در روش های سنتی بدون توجه کافی به تغییرات سریع در جهان و نیازها و امتیازات خاص کشور، معمولاً اصلاحات جزئی در بازنگری برنامه ها
- ضعف در جذب دانش آموختگان ممتاز از دانشگاه‌های خوب خارج و داخل (سیستم فراخوان استخدام فعلی و شرایط حاکم، جذب کننده نیست)
- جهت گیری پژوهش برای آموزش و مقاله و غلبه پژوهش های پایان نامه‌ای و رساله و عدم توجه به طرح های کلان و مورد نیاز برای رفع مشکلات کشور و نداشتن ساز و کارهای استفاده از علم در عمل
- نامشخص بودن جایگاه و وضع علوم انسانی و اجتماعی در کشور

از آنجا که مساعدترین و حساس‌ترین موضوع در نظام فکری هر کشور، بخش تعلیم و تربیت آن کشور است و پویایی و خلاقیت این بخش به پویایی و تحول عمیق در سطح کشور می‌انجامد، لذا باید نظام آموزش و پرورش کشور را چنان متحول ساخت که بتواند انسان‌هایی آگاه، بااخلاق و خلاق تربیت کند. انسان‌هایی که در همه ابعاد زندگی اجتماعی، سیاسی، فرهنگی و اقتصادی جامعه، منشأ برکت و تحول باشند؛ بنابراین نقش آموزش و پرورش در آینده تحولات کشور، بسیار محوری و کلیدی است و همین نکته بر ضرورت بررسی ویژگی‌های کنونی آموزش و پرورش می‌افزاید. برای برنامه‌ریزی آینده و حضور قوی در عرصه تحولات اندیشمندانه جهان، راهی جز نقد شرایط حاضر و یافتن نقاط کاستی نظام حاضر و ارائه راهکار برای بهبود آن‌ها نمی‌ماند و به همین سبب لازم است ناکارایی آموزش و پرورش مورد بحث قرار گیرد.

این ناکارایی را می‌توان در چند بخش ساختار تعلیم و تربیت کشور، دسترسی به تحصیلات و مدرسه، نابرابری فرصت‌های آموزشی، کیفیت سیستم آموزشی و مطالبات فرهنگیان مورد بررسی قرار داد. مقایسه آموزش مجازی در ایران با سایر کشورها نیز یکی دیگر از مواردی است که می‌تواند به‌خوبی نقاط ضعف و قوت ساختار آموزشی کشور را عیان کند.

معمولا دانشجویان ایرانی علاقه بسیاری به نوشتن کدهای برنامه نویسی دارند و در بحث های تجزیه و تحلیل و مدیریت پروژه بسیار ضعیف هستند.

مشکل امروز نرم افزار های و یا حتی میتوان گفت علت شکست بسیاری از نرم افزارها نبود توازن در زمان ، هزینه و کیفیت نرم افزار میباشد که همه این ها ریشه در عدم مدیریت درست نرم افزار دارد.

کنترل و موازنه منابع نرم افزار بسیار کار حساسی میباشد و اگر هر کدام به درستی مدیریت نشوند در نهایت پروژه به مشکل خواهد خورد .

در دانشگاه ها معمولا مباحث روش های مدیریت نرم افزار به صورت تئوری و نوشتن کد به زبان های رایج و پرکاربردتر آموزش داده میشود در حالی که خبری از مدیر پروژه و مدیریت پروژه نیست و کاملا با محیط عملیاتی و کاری متفاوت میباشد و اکثر دانشجویان نرم افزار معمولا کدنویس های خوبی میشوند در حالی که مدیریت پروژه در فضای ذهنی آنها بسیار کم رنگ تر است .

همین دانشجویان در آینده مدیریت شرکت های نرم افزاری را عهده دار هستند و معمولا در محیط کار هم تمایل به شروع کد نویسی دارند و بعد از تعیین نیازمندی ها پروژه شروع به ایجاد پروژه و بحث کد نویسی میکنند و خبری از تجزیه و تحلیل نیست ، در واقع مباحث تئوری آموزش داده شده را نمی توانند در فضای کار مورد استفاده قرار دهند. برای مثال در متدولوژی های مدیریت نرم افزار روش های مختلفی در دانشگاه ها و فضاهای آموزشی تدریس و توضیح داده میشود ، مثلا متدولوژی اسکرام .

دانشجو گذشته در فضای کار حتی نام ابزار های پیاده سازی این متدها را نمی داند و چون آشنایی با نحوه پیاده سازی این متدها را نمی داند فکر می کند کار بیهوده و زمان بری خواهد بود و سریعاً وارد بحث کد نویسی می شود و در نهایت بدون شک پروژه یا از نظر زمان و یا از نظر هزینه و کیفیت به شکست منجر خواهد شد .

اگر در پروژه هزینه و زمان به درستی برآورد نشود حتما به شکست منجر خواهد شد و برآورد این دو به عهده مدیر پروژه می باشد . یعنی پروژه باید تا حد امکان به قسمت های کوچکتر تجزیه و تمام این واحدها از نظر زمان و نهایتاً هزینه برآورد شوند ، اما متأسفانه در خیلی مواقع چنین اتفاقی نمی افتد و شروع به کد نویسی

میکنند و چون پروژه به درستی تجزیه و تحلیل نشده است زمان کد نویسی هم بسیار طولانی خواهد شد و به کرار کدهای نوشته شده تغییر خواهند کرد که این کار بسیار زمان گیر خواهد بود .

مدیر پروژه باید قبل از شروع کدنویسی پروژه در جلسات متعدد و به کمک ابزارهای این حوزه بخش های مختلف پروژه را تجزیه و تحلیل کرده و موارد قابل پیاده سازی را به صورت شفاف تعیین کند تا بعد از اعلام هزینه و زمان به مشتری بتواند به وعده های خود عمل کند .

در همین راستا و برای بهبود این وضعیت و ارائه خروجی های بهتر علاوه بر بحث های کد نویسی حتما باید در دانشگاه ها و موسسات مرتبط بحث های تجزیه و تحلیل و مدیریت پروژه به صورت عملی و اجرایی با دانشجویان کار شود.