

فهرست مطالب

۶	۱ فصل اول - محصول
۶	۱.۱ نرم افزار چیست؟
۶	۲.۱ تفاوت نرم افزار با سخت افزار
۶	۳.۱ قطعیت اطلاعاتی به چه معناست؟
۶	۴.۱ انواع گروه بندی نرم افزار
۷	۵.۱ نرم افزارهای سیستمی
۷	۱.۵.۱ مشخصه های نرم افزار های سیستمی
۷	۶.۱ نرم افزارهای زمان حقیقی
۷	۱.۶.۱ عناصر نرم افزار زمان حقیقی
۸	۷.۱ نرم افزار های تجاری
۸	۸.۱ نرم افزارهای تعبیه شده
۸	۹.۱ بحران نرم افزار
۸	۱۰.۱ اسطوره های نرم افزاری در انواع سطوح
۹	۱۱.۱ اسطوره های مدیریتی
۹	۱.۱۱.۱ اسطوره
۹	۲.۱۱.۱ واقعیت
۹	۳.۱۱.۱ اسطوره
۹	۴.۱۱.۱ واقعیت
۹	۵.۱۱.۱ اسطوره
۹	۶.۱۱.۱ واقعیت
۹	۱۲.۱ اسطوره های مشتریان
۹	۱.۱۲.۱ اسطوره
۹	۲.۱۲.۱ واقعیت
۱۰	۱۳.۱ اسطوره های سازندگان
۱۰	۱.۱۳.۱ اسطوره
۱۰	۲.۱۳.۱ واقعیت
۱۰	۳.۱۳.۱ اسطوره
۱۰	۴.۱۳.۱ واقعیت
۱۰	۵.۱۳.۱ اسطوره
۱۰	۶.۱۳.۱ واقعیت
۱۰	۲ فصل دوم - فرآیند
۱۰	۱.۲ فرآیند چیست؟
۱۰	۲.۲ دلیل اهمیت فرآیند
۱۰	۳.۲ حاصل فرآیند چیست؟
۱۱	۴.۲ معیارهای ارزیابی فرآیند
۱۱	۵.۲ آیا فرآیند مترادف با مهندسی نرم افزار است؟
۱۱	۶.۲ تعریف مهندسی نرم افزار بر اساس فریتز باور؟

۷.۲	لایه های مهندسی نرم افزار	۱۱
۸.۲	زمینه های فرآیند کلیدی یا KPA را تعریف کنید ؟	۱۱
۹.۲	روشهای مهندسی نرم افزار	۱۲
۱۰.۲	مراحل مهندسی	۱۲
۱۱.۲	فاز های مهندسی نرم افزار	۱۲
۱۲.۲	فاز تعریف	۱۲
۱۳.۲	۳ کار عمده در فاز تعریف	۱۳
۱۴.۲	فاز توسعه	۱۳
۱۵.۲	۳ وظیفه ی فنی در فاز توسعه	۱۳
۱۶.۲	فاز پشتیبانی	۱۳
۱۷.۲	۴ نوع تغییر در فاز پشتیبانی	۱۳
۱۸.۲	اعمال چتری بر کل فاز های مهندسی نرم افزار	۱۴
۱۹.۲	پنج سطح بلوغ فرآیند	۱۴
۲۰.۲	KPA چیست	۱۴
۲۱.۲	ویژگی های KPA	۱۴
۲۲.۲	تعریف مدل فرآیند نرم افزار	۱۵
۲۳.۲	مدلهای فرآیند نرم افزار	۱۵
۲۴.۲	مدل ترتیبی خطی	۱۵
۲۵.۲	فعالیت های مدل ترتیبی خطی	۱۶
۲۶.۲	مشکلاتی که به هنگام اجرای مدل ترتیبی خطی به وجود می آید ؟	۱۶
۲۷.۲	مدل ساخت نمونه اولیه	۱۶
۲۸.۲	مشکلات ساخت نمونه اولیه	۱۷
۲۹.۲	فازهای مدل RAD	۱۷
۳۰.۲	نتایج منفی مدل RAD	۱۷
۳۱.۲	انواع مدل های تکاملی فرآیند نرم افزار	۱۷
۳۲.۲	۳۲ مقایسه ی مدل ترتیبی خطی و مدل های تکاملی	۱۸
۳۳.۲	۳۳ مدل گام به گام	۱۸
۳۴.۲	۳۴ مدل مارپیچی	۱۸
۳۵.۲	۳۵ ناحیه ی کاری مدل مارپیچی	۱۸
۳۶.۲	۳۶ مدل مارپیچی winwin	۱۹
۳۷.۲	۳۷ سه مرحله فرآیند در مدل winwin را نام ببرید ؟ (نقاط لنگر گاه)	۱۹
۳۸.۲	۳۸ مدل بسط همزمان را توضیح دهید؟	۱۹
۳۹.۲	۳۹ بسط مبتنی بر مولفه ها	۱۹
۴۰.۲	۴۰ مدل روشهای رسمی	۱۹
۴۱.۲	۴۱ علم مهندسی نرم افزار را تعریف کنید ؟	۱۹
۴۲.۲	۴۲ اعمال مباحثاتی در مدل مارپیچی winwin	۱۹

۳ فصل سوم - مفاهیم مدیریت پروژه

۱.۳	در مدیریت پروژه های نرم افزاری بر چند نکته تاکید می شود ؟	۲۰
۲.۳	مدل بلوغ مدیریت افراد، چه زمینه های کلیدی را برای نرم افزار نویسان تعیین می کند؟	۲۰

۳۰	بازیگران فرآیند نرم افزار	۳.۳
۴۰	مدل MOI برای رهبری	۴.۳
۵۱	چهار نکته ی کلیدی در مدیریت پروژه	۵.۳
۶۱	ساختار بهترین تیم به چه عواملی بستگی دارد؟	۶.۳
۷۱	سازماندهی کلی برای یک تیم بر اساس نظریه ی ماننتی	۷.۳
۸۰	بر اساس نظریه ی ماننتی چه عواملی را باید هنگام طرح ریزی ساختار تیم های مهندسی نرم افزار در نظر گرفت؟	۸.۳
۹۰	بر اساس نظریه ی کنستانتین، چهار الگوی سازمانی برای تیمهای مهندسی نرم افزار را توضیح دهید؟	۹.۳
۱۰۰	برای دستیابی به تیمی با کارایی بالا	۱۰.۳
۱۱۰	تیم ژل شده چه نوع تیمی است؟	۱۱.۳
۱۲۰	پنج عامل مسمومیت تیم از دیدگاه جکمن	۱۲.۳
۱۳۰	یک عامل مسمومیت تیم دیگر	۱۳.۳
۱۴۰	مجموعه ای از تکنیک های هماهنگ سازی پروژه از دیدگاه کراول و استریتر	۱۴.۳
۱۵۰	دامنه کاربرد نرم افزار	۱۵.۳
۱۶۰	مدیر پروژه با در نظر گرفتن کدام موارد، مدل فرآیند نرم افزار را انتخاب می کند؟	۱۶.۳
۱۷۰	در یک پروژه کوچک، برای فعالیت ارتباط با مشتری نیاز به چه وظایف کاری می باشد؟	۱۷.۳
۱۸۰	در یک پروژه پیچیده و بزرگ، برای فعالیت ارتباط با مشتری نیاز به چه وظایف کاری می باشد؟	۱۸.۳
۱۹۰	علائم در خطر بودن پروژه از منظر جان ریل	۱۹.۳
۲۰۰	قاعده ۹۰-۹۰	۲۰.۳
۲۱۰	روش ۵ بخشی جان ریل مبتنی بر عقل سلیم برای پروژه های نرم افزاری	۲۱.۳
۲۲۰	اصل W5HH	۲۲.۳
۲۳۰	مجموعه پرسش ها برای اینکه سازمانی تعیین کند که آیا پروژه ای اعمال بحرانی را پیاده سازی کرده است؟	۲۳.۳
۲۴۰	چهار مورد که تاثیری اساسی بر مدیریت پروژه های نرم افزاری دارند؟	۲۴.۳

۴ فصل چهارم - معیارهای پروژه و فرآیند نرم افزار

۲۸	اندازه گیری در سرتاسر پروژه چه نتایجی دارد؟	۱.۴
۲۸	چهار دلیل برای اندازه گیری فرآیندهای نرم افزاری، محصولات و منابع را نام ببرید؟	۲.۴
۲۹	موازن، معیارها و شاخصها	۳.۴
۲۹	معیار	۴.۴
۲۹	بهبود بخشیدن به فرآیند	۵.۴
۳۰	بازدهی یک فرآیند نرم افزار را توضیح دهید؟	۶.۴
۳۰	معیارهای فرآیند توسط چه ویژگی هایی اندازه گیری می شوند؟	۷.۴
۳۰	چند مورد از معیار های خصوصی را نام ببرید؟	۸.۴
۳۰	معیارهای عمومی	۹.۴
۳۰	بهبود آماری فرآیند نرم افزاری یا SSPI را توضیح دهید؟	۱۰.۴
۳۱	مراحل تحلیل شکست را توضیح دهید؟	۱۱.۴
۳۱	نمودار ستون فقرات ماهی گریدی را توضیح دهید؟	۱۲.۴
۳۱	هر پروژه باید چه مواردی را اندازه گیری کند؟	۱۳.۴
۳۱	اندازه گیری های جهان فیزیکی را به چند گروه می توان تقسیم کرد؟	۱۴.۴

۳۲	۱۵.۴ تقسیم بندی دامنه معیارهای نرم افزار
۳۲	۱۶.۴ معیار های جنبه ی خصوصی، معیارهای جنبه ی عمومی
۳۲	۱۷.۴ معیارهای اندازه گرا
۳۲	۱۸.۴ تعداد خطوط کد
۳۲	۱۹.۴ مجموعه ای از معیارهای اندازه گیری که برای هر پروژه می توان استفاده کرد؟
۳۲	۲۰.۴ معیارهای عملکردگرا
۳۳	۲۱.۴ نقطه عملکرد (Function Point) چیست ؟
۳۳	۲۲.۴ مقادیر دامنه اطلاعات به چه صورت تعیین می شوند؟
۳۳	۲۳.۴ هدف اصلی مهندسی نرم افزار
	۲۴.۴ مجموعه عوامل کیفیتی مک کال و کارائو برای توسعه ی کیفیت نرم افزار، نرم افزار را از چند دیدگاه ارزیابی می کند ؟
۳۳	۲۵.۴ شاخص های کیفیت نرم افزار را نام ببرید ؟
۳۳	۲۶.۴ درستی
۳۴	۲۷.۴ قابلیت نگهداری
۳۴	۲۸.۴ یکپارچگی
۳۴	۲۹.۴ قابلیت استفاده

۵ فصل پنجم - برنامه ریزی برای پروژه ی نرم افزاری

۳۴	۱.۵ برنامه ریزی پروژه چیست؟
۳۵	۲.۵ برنامه ریزی پروژه شامل چه مواردی می شود ؟
۳۵	۳.۵ هنگام انتخاب مدیر پروژه کدام ویژگی اهمیت بیشتری دارد ؟
۳۵	۴.۵ برآورد منابع، هزینه و زمانبندی برای کارهای مهندسی نرم افزار به چه ویژگی هایی نیاز دارد؟
۳۵	۵.۵ نکته - عدم قطعیت در برآورد
۳۵	۶.۵ تاثیر پیچیدگی بر عدم قطعیت
۳۵	۷.۵ نسبی بودن پیچیدگی
۳۵	۸.۵ تاثیر اندازه پروژه بر برآورد
۳۵	۹.۵ درجه ی عدم قطعیت ساختاری
۳۵	۱۰.۵ منظور از ساختار در کلمه ی 'درجه ی عدم قطعیت ساختاری'
۳۶	۱۱.۵ امکان سنجی یا امکان پذیری نرم افزار چند بعد دارد ؟
۳۶	۱.۱۱.۵ فناوری
۳۶	۲.۱۱.۵ مالی
۳۶	۳.۱۱.۵ زمان
۳۶	۴.۱۱.۵ منابع
۳۶	۱۲.۵ منابع
۳۶	۱۳.۵ انواع منابع توسعه
۳۷	۱۴.۵ ویژگی های مشخص کننده منابع را نام ببرید؟
۳۷	۱۵.۵ منابع نرم افزاری قابل استفاده مجدد
۳۷	۱۶.۵ چهار دسته از مولفه های نرم افزاری قابل استفاده مجدد
۳۷	۱۷.۵ مولفه های آماده
۳۷	۱۸.۵ مولفه های تجربه کامل

۳۷	۱۹.۵ مولفه های تجربه ناقص
۳۸	۲۰.۵ مولفه های جدید
۳۸	۲۱.۵ دستورالعمل هایی برای استفاده از مولفه های قابل استفاده مجدد

۱ فصل اول - محصول

۱.۱ نرم افزار چیست؟

- از دیدگاه مهندس نرم افزار محصولی است شامل برنامه، مستندات و داده ها
- از دیدگاه کاربر اطلاعاتی است که به نحوی به درد می خورد

۲.۱ تفاوت نرم افزار با سخت افزار

- نرم افزار دارای ویژگی هایی است که تفاوت چشمگیری با ویژگی های سخت افزار دارد .
- نرم افزار، مهندسی و بسط داده می شود و چیزی نیست که به معنای کلاسیک ساخته شود
- چه در نرم افزار و چه در سخت افزار کیفیت بالا از طریق طراحی خوب به دست می آید .
- نرم افزار فرسوده نمی شود، بلکه فاسد می شود .
- نرم افزار نسبت به ناملازمات محیطی که باعث فرسایش نرم افزار می شود، نفوذ پذیر نیست .
- گرچه صنعت در حال حرکت به سوی مونتاژ قطعات است، اکثر نرم افزارها همچنان به صورت سفارشی ساخته می شوند .

۳.۱ قطعیت اطلاعاتی به چه معناست؟

به معنای قابلیت پیش بینی ترتیب و زمان بندی اطلاعات است .

۴.۱ انواع گروه بندی نرم افزار

- نرم افزارهای سیستمی
- نرم افزارهای زمان حقیقی
- نرم افزارهای تجاری
- نرم افزار های مهندسی و علمی
- نرم افزار های تعبیه شده
- نرم افزار های کامپیوترهای شخصی
- نرم افزارهای مبتنی بر وب
- نرم افزارهای هوش مصنوعی

۵.۱ نرم افزارهای سیستمی

مجموعه ای از برنامه هاست که برای سرویس دهی به برنامه های دیگر نوشته شده اند .
مثل :

- کامپایلر ها
- ویراستار ها
- برنامه های مدیریت فایل

۱.۵.۱ مشخصه های نرم افزار های سیستمی

- بر هم کنش سنگین با سخت افزار کامپیوتر
- استفاده سنگین توسط چند کاربر
- لزوم زمان بندی برای انجام کارها
- مدیریت فرآیند و اشتراک منابع
- ساختمان داده های پیچیده
- واسطهای خارجی چندگانه

۶.۱ نرم افزارهای زمان حقیقی

نرم افزاری که رویداد های جهان واقعی را همانطوری که رخ می دهند، نظارت ، تحلیل و کنترل می کنند .

۱.۶.۱ عناصر نرم افزار زمان حقیقی

- قطعه جمع آوری کننده داده ها
- اطلاعات را از محیط خارجی جمع آوری و قالب بندی می کند .
- قطعه تحلیل کننده
- اطلاعات را بنا به نیاز کاربردی انتقال می دهد .
- قطعه کنترل / خروجی
- به محیط خارجی پاسخ می دهد
- قطعه نظارت
- همه ی قطعات دیگر را هماهنگ می کند .

۷.۱ نرم افزارهای تجاری

این نوع برنامه های کاربردی، داده های موجود را دوباره به شیوه ای سازماندهی می کند که عملیات تجاری و تصمیم گیری مدیریتی تسهیل شوند .

۸.۱ نرم افزارهای تعبیه شده

برای کنترل محصولات و سیستم های مربوط به بازارهای صنعتی و مصرفی به کار می رود .
مثل :

- صفحه کلید برای ماکروویو
- عملیات دیجیتال در خودرو
- کنترل سوخت
- صفحه نمایش داشبورد
- سیستم ترمز

۹.۱ بحران نرم افزار

- مشکلات مرتبط با نرم افزار را بحران می نامند .
مجموعه مشکلات موجود بر سر راه بسط نرم افزار های کامپیوتری شامل :
- مشکلات مرتبط با چگونگی بسط نرم افزار
 - چگونگی پشتیبانی حجم رو به رشدی از نرم افزار های موجود
 - چگونگی همگام ماندن با تقاضای فزاینده برای نرم افزار های جدید .

۱۰.۱ اسطوره های نرم افزاری در انواع سطوح

- اسطوره های مدیریتی
- اسطوره های مشتریان
- اسطوره های سازندگان

۱۱.۱ اسطوره های مدیریتی

۱.۱۱.۱ اسطوره

ما کتابی داریم که شامل استاندارد ها و روال های لازم برای ساخت نرم افزار هاست و آنچه را که افراد من باید بدانند در اختیارشان قرار می دهد .

۲.۱۱.۱ واقعیت

همچین کتابی وجود ندارد، حتی اگر وجود داشته باشد، بروز و کامل نیست .

۳.۱۱.۱ اسطوره

افراد من ابزارهای نرم افزارسازی حرفه ای را دارند و جدیدترین کامپیوتر ها را برای آنها خواهیم خرید .

۴.۱۱.۱ واقعیت

برای ساخت نرم افزار با کیفیت، فقط داشتن سخت افزار خوب کافی نیست بلکه داشتن نرم افزار های کمکی نیز اهمیت بالایی دارند

۵.۱۱.۱ اسطوره

اگر از برنامه عقب بیفتیم، می توانیم بر تعداد برنامه نویسان بیفزاییم و عقب افتادگی را جبران کنیم .

۶.۱۱.۱ واقعیت

با افزودن افراد جدید به تیم ساخت نرم افزار بر میزان تاخیر ساخت آن افزوده می شود زیرا با از راه رسیدن افراد جدید، افراد قدیمی باید زمانی را صرف آموزش آنها کنند، در نتیجه زمانی که باید صرف کار روی نرم افزار شود، هدر می رود .

۱۲.۱ اسطوره های مشتریان

۱.۱۲.۱ اسطوره

نیازهای پروژه پیوسته در حال تغییر است و این تغییرات را به راحتی می توان در نرم افزار جای داد زیرا نرم افزار انعطاف پذیر است .

۲.۱۲.۱ واقعیت

تاثیر تغییر به زمان اعمال تغییر بستگی دارد،

درخواستهای اولیه به تغییر را به راحتی می توان پاسخ داد .

هنگامی که تغییرات در اثنای طراحی نرم افزار درخواست می شوند ، هزینه ها بالاتر می رود .

تغییر ، در صورتی که نرم افزار به مرحله استفاده رسید، هزینه ها به مراتب بالاتر می رود .

۱۳.۱ اسطوره های سازندگان

۱.۱۳.۱ اسطوره

هنگامی که برنامه را نوشتیم و برنامه کار کرد دیگر کار تمام است .

۲.۱۳.۱ واقعیت

تجربه نشان داده است که بین ۶۰ تا ۸۰ درصد کوششهای روی نرم افزارها ، پس از نخستین بار تحویل آنها به مشتری صورت می پذیرد .

۳.۱۳.۱ اسطوره

تا هنگامی که برنامه را اجرا نکردم، راهی برای ارزیابی کیفیت آن ندارم .

۴.۱۳.۱ واقعیت

مرور تکنیکی رسمی از زمان آغاز پروژه قابل اجراست .

۵.۱۳.۱ اسطوره

تنها چیز قابل تحویل برای یک پروژه موفق، برنامه ای است که کار کند .

۶.۱۳.۱ واقعیت

نرم افزار شامل عناصر فراوان است، برنامه ای که کار می کند فقط بخشی از پیکربندی نرم افزار است .

۲ فصل دوم - فرآیند

۱.۲ فرآیند چیست؟

یک نقشه ی راه که در ایجاد نتیجه ای با کیفیت و به موقع شما را یاری می کند .

۲.۲ دلیل اهمیت فرآیند

زیرا باعث ثبات، کنترل و سازماندهی فعالیتی می شود .

۳.۲ حاصل فرآیند چیست؟

حاصل کار فرآیند، برنامه ها ، مستندات و داده ها می باشد

۴.۲ معیارهای ارزیابی فرآیند

- کیفیت
- به موقع بودن
- کارایی دراز مدت محصول
- بهترین ملاک ها برای بازدهی فرآیند است .

۵.۲ آیا فرآیند مترادف با مهندسی نرم افزار است؟

پاسخ این است : بلی و خیر
فرآیند نرم افزار روش مهندسی را مشخص می کند .
ولی مهندسی نرم افزار شامل فناوری هایی می شود که فرآیند را تشکیل می دهند .

۶.۲ تعریف مهندسی نرم افزار بر اساس فریتز باور؟

مهندسی نرم افزار عبارت است از وضع اصول مهندسی به جا و مناسب و استفاده از آنها برای به دست آوردن یک نرم افزار مقرون به صرفه که قابل اطمینان بوده و به طرز کارآمدی عمل کند .

۷.۲ لایه های مهندسی نرم افزار

- ابزارها
- روشها
- فرآیندها
- تاکید بر کیفیت

۸.۲ زمینه های فرآیند کلیدی یا KPA را تعریف کنید ؟

زمینه های فرآیند کلیدی، پایه ای برای کنترل مدیریتی پروژه های نرم افزاری تشکیل داده و بستری برای :

- اعمال روشهای فنی
- تولید محصولات کاری
- تعیین مراحل
- حصول اطمینان از کیفیت
- مدیریت مناسب تغییرات
- ایجاد می کنند .

۹.۲ روشهای مهندسی نرم افزار

روشهای مهندسی نرم افزار، شیوه های فنی برای ساخت نرم افزار متکی بر یک مجموعه اصول بنیادی است که بر تمام زمینه های فناوری حاکم است .
روشها شامل آرایه وسیعی از وظایف است :

- تحلیل خواسته ها
- طراحی
- ساخت برنامه ها
- آزمایش
- پشتیبانی

۱۰.۲ مراحل مهندسی

- تحلیل
- طراحی
- ساخت
- بررسی
- مدیریت نهاد های فنی و اجتماعی

۱۱.۲ فاز های مهندسی نرم افزار

- فاز تعریف
- فاز توسعه
- فاز پشتیبانی

۱۲.۲ فاز تعریف

فاز تعریف بر چيستی تاکید دارد . در فاز تعریف، مهندس نرم افزار تعیین می کند :

- چه اطلاعاتی باید پردازش شود .
- کدام عمل و کارایی مطلوب است .
- چه رفتارهای سیستمی قابل انتظار است .
- چه رابطه هایی را می توان برقرار کرد .
- چه محدودیتهایی در طراحی وجود دارد
- در نهایت خواسته های کلیدی سیستم و نرم افزار تشخیص داده می شود .

۱۳.۲ ۳ کار عمده در فاز تعریف

- مهندسی اطلاعات یا سیستم
- طرح ریزی پروژه نرم افزار
- تحلیل خواسته ها

۱۴.۲ فاز توسعه

فاز توسعه بر چگونگی تاکید دارد .
در فاز توسعه مهندس نرم افزار می کوشد تعیین کند داده ها چه ساختاری داشته باشند .

۱۵.۲ ۳ وظیفه ی فنی در فاز توسعه

- طراحی نرم افزار
- تولید دستور ها
- آزمایش نرم افزار

۱۶.۲ فاز پشتیبانی

فاز پشتیبانی بر تغییرات تاکید دارد .
این تغییرات ممکن است شامل تصحیحاتی در جهت تکامل نرم افزار باشند یا تغییراتی که ناشی از تغییر خواسته های مشتریان است .

۱۷.۲ ۴ نوع تغییر در فاز پشتیبانی

- تصحیح
- نگهداری تصحیحی، نرم افزار را در جهت تصحیح معایب تغییر می دهد .
- تطابق
- نگهداری تطابقی، منجر به انجام اصلاحاتی در نرم افزار می شود که پاسخگوی تغییرات محیط خارجی شود
- بهبود
- نگهداری بهبودی، نرم افزار را به مافوق خواسته های اولیه ی خود سوق می دهد .
- جلوگیری
- نگهداری پیشگیرانه، تغییراتی در برنامه کامپیوتری اعمال می کند که بتوان برنامه را راحت تر تصحیح کرد، تطابق داد و بهبود بخشید .

۱۸.۲ اعمال چتری بر کل فاز های مهندسی نرم افزار

اعمال چتری در سرتاسر فرآیند نرم افزار اجرا می شوند .

- کنترل و دنبال کردن پروژه نرم افزاری
- مرور فنی رسمی
- تضمین کیفیت نرم افزار
- مدیریت پیکربندی نرم افزار
- تولید و تهیه مستندات
- مدیریت قابل استفاده مجدد
- سنجش
- مدیریت ریسک

۱۹.۲ پنج سطح بلوغ فرآیند

- سطح ۱ : آغاز
- سطح ۲ : تکرار پذیر
- سطح ۳ : تعریف شده
- سطح ۴ : مدیریت شده
- سطح ۵ : بهینه سازی

۲۰.۲ KPA چیست

KPA ها آن دسته از وظایف مهندسی نرم افزار را توصیف می کند که باید برای رسیدن به حد مطلوبی از عملکرد در یک سطح معین به انجام برسد .

۲۱.۲ ویژگی های KPA

- اهداف
- وظایف
- تواناییها
- روشهای نظارت بر پیاده سازی
- روشهای بازبینی پیاده سازی

۲۲.۲ تعریف مدل فرآیند نرم افزار

برای حل مسائل در یک مجموعه صنعتی، یک مهندس نرم افزار باید یک راهبرد توسعه تعیین کند که در برگیرنده لایه های فرآیند، روشها و ابزارها و فازهای مهندسی نرم افزار (تعریف، توسعه، پشتیبانی) باشد، این راهبرد را مدل فرآیند یا الگوی مهندسی نرم افزار می نامند.

یک مدل فرآیند برای مهندسی نرم افزار بر اساس:

- ماهیت پروژه و نوع کاربرد
 - روشها و ابزارهای مورد استفاده
 - کنترلها و قطعات قابل تحویل مورد نیاز
- انتخاب می شود.

۲۳.۲ مدلهای فرآیند نرم افزار

- مدل ترتیبی خطی
- مدل ساخت نمونه اولیه
- مدل RAD

۲۴.۲ مدل ترتیبی خطی

این مدل که مدل آبشار یا چرخه ی حیات کلاسیک خوانده می شود، یک روش سیستماتیک و ترتیبی برای بسط نرم افزار پیشنهاد می کند که در سطح سیستمی آغاز می شود و به

- تحلیل
 - طراحی
 - کدنویسی
 - آزمایش
 - پشتیبانی
- پیشروی می کند.

۲۵.۲ فعالیت های مدل ترتیبی خطی

- مهندسی سیستم / اطلاعات و مدلسازی
- تحلیل و مهندسی سیستم شامل جمع آوری خواسته ها در سطح سیستم و طراحی سطح بالاست .
- مهندسی اطلاعات شامل جمع آوری خواسته ها در سطح تجارت راهبردی و در سطح زمینه ی کاری است .
- تحلیل خواسته های نرم افزار
- طراحی
- تمرکز طراحی بر روی چهار صفات از برنامه تاکید دارد :
- ساختمان داده ها
- معماری نرم افزار
- نمایش واسطها
- جزئیات رویه ای
- تولید کد
- آزمایش
- پشتیبانی

۲۶.۲ مشکلاتی که به هنگام اجرای مدل ترتیبی خطی به وجود می آید ؟

- با پیش رفتن تیم پروژه، ممکن است تغییرات باعث ایجاد سردرگمی شود .
- مدل ترتیبی خطی به بیان واضح نیاز دارد و غالباً برای مشتری دشوار است که همه ی نیازهای خود را به وضوح بیان کند .
- مشتری باید حوصله داشته باشد زیرا تا آخرین روز پروژه هیچ نسخه ای از برنامه در دسترس او قرار نخواهد گرفت .

۲۷.۲ مدل ساخت نمونه اولیه

- غالباً مشتری یک مجموعه اهداف برای نرم افزار تعیین می کند، ولی جزئیات ورودی ها و یا خواسته های خروجی را مشخص نمی کند یا سازنده از کارایی الگوریتم و یا تطابق با سیستم عامل مطمئن نباشد، در این شرایط الگوی ساخت نمونه اولیه ممکن است روش خوبی باشد .
- الگوی ساخت نمونه اولیه با جمع آوری خواسته ها آغاز می شود و پس از ملاقات مشتری و سازنده و تعیین اهداف کلی نرم افزار با یک طراحی سریع صورت میگیرد .
- در طراحی سریع، هدف اصلی ارائه ی آن دسته از ویژگی های نرم افزار است که به چشم مشتری می آیند (مثل روشهای ورود اطلاعات و فرمت های خروجی)
- طراحی سریع منجر به ساخت یک نمونه اولیه می شود و نمونه اولیه به عنوان راهکاری برای تشخیص خواسته های نرم افزار عمل می کند .

۲۸.۲ مشکلات ساخت نمونه اولیه

- معمولاً مشتری تقاضا می کند که با چند ترمیم جزئی این نمونه اولیه به محصول نهایی تبدیل شود بدون اینکه کیفیت کلی نرم افزار و قابلیت نگهداری مد نظر باشد و اکثر اوقات مدیریت هم قبول می کند
- سازندگان ممکن است برای ساختن هرچه سریعتر نمونه اولیه، در پیاده سازی آن کوتاهی کنند و دست به انتخاب های نامناسب در پیاده سازی نرم افزار بزنند .

۲۹.۲ فازهای مدل RAD

- مدلسازی تجاری
- مدلسازی داده ای
- مدلسازی فرآیند
- تولید برنامه کاربردی
- آزمایش و توان عملیاتی

۳۰.۲ نتایج منفی مدل RAD

- برای ایجاد تعداد مناسبی از تیم های RAD به منابع انسانی کافی نیاز دارد
- نیاز به سازندگان و مشتریانی دارد که برای کامل کردن سیستم به یک چارچوب زمانی فشرده تر، معتقد هستند
- اگر سیستمی را نتوان به طور مناسب تقسیم بندی کرد، ساخت مولفه های لازم مشکل آفرین خواهد شد
- وقتی احتمال بروز خطرات فنی بالا باشد، مدل مناسبی نیست .

۳۱.۲ انواع مدل های تکاملی فرآیند نرم افزار

- مدل گام به گام
- مدل مارپیچی
- مدل مارپیچی winwin
- مدل بسط همزمان
- بسط مبتنی بر مولفه ها
- مدل روشهای رسمی
- تکنیکهای نسل چهارم

۳۲.۲ مقایسه ی مدل ترتیبی خطی و مدل های تکاملی

مدل ترتیبی خطی برای توسعه ی آسان طراحی شده است و در این مدل فرض می شود که یک سیستم کامل پس از طی شدن یک ترتیب خطی کامل شده است .
مدل ساخت نمونه اولیه برای کمک به مشتری در شناخت خواسته ها طراحی می شود و این مدل اساساً برای تحویل یک سیستم آماده و کامل طراحی نشده است .
مدل های تکاملی ، تکراری هستند . این مدل ها به شیوه ای طراحی می شوند که مهندس نرم افزار را قادر سازند تا نسخه هایی از نرم افزار را توسعه دهند که هر یک از قبلی کاملتر باشند .

۳۳.۲ مدل گام به گام

مدل گام به گام عناصر مدل ترتیبی خطی را با فلسفه ی تکراری مدل ساخت نمونه اولیه تلفیق می کند . هر ترتیبی خطی یک قطعه قابل تحویلی از نرم افزار (گام) را ارائه می دهد .

۳۴.۲ مدل مارپیچی

مدل مارپیچی ، ماهیت تکراری مدل ساخت نمونه اولیه را با جنبه های کنترلی و سیستماتیک مدل ترتیبی خطی تلفیق می کند .
این مدل پتانسیل لازم برای بسط سریع نسخه های تکاملی نرم افزار را دارا می باشد .
با استفاده از مدل مارپیچی، نرم افزار به صورت یک سری نگارشهای تکاملی توسعه می یابد .
طی نخستین تکرار ها، نگارش تکاملی ممکن است یک مدل کاغذی یا یک نمونه اولیه باشد، طی تکرارهای بعدی، هر بار نسخه ی کاملتری از سیستم، تولید می شود .
مدل مارپیچی به چند فعالیت چارچوبی تقسیم می شود که نواحی کاری نامیده می شوند .
یک مدل مارپیچی حاوی ۶ ناحیه ی کاری است .

۳۵.۲ ۶ ناحیه ی کاری مدل مارپیچی

- ارتباط با مشتری
- طرح ریزی
- تحلیل ریسک
- مهندسی
- ساخت و ارائه
- ارزیابی مشتری

۳۶.۲ مدل مارپیچی winwin

در این مدل، مشتری و سازنده وارد یک فرآیند گفتگو و بحث می شوند و در این فرآیند ممکن است از مشتری خواسته شود تا میان عملکرد، کارایی و ویژگی های دیگر سیستم یا محصول در مقابل هزینه و زمان بازیابی موازنه برقرار کند تا یک نتیجه برد برد حاصل شود .

یعنی مشتری با دستیابی به محصول یا سیستمی که واجد اکثر نیازهای اوست برنده می شود و سازنده با کار کردن در چارچوب مهلت و بودجه ای واقعی و قابل حصول برنده می شود .

۳۷.۲ سه مرحله فرآیند در مدل winwin را نام ببرید ؟ (نقاط لنگر گاه)

- نخستین نقطه ی لنگرگاهی اهداف چرخه ی حیات نام دارد
- دومین نقطه ی لنگرگاهی معماری چرخه ی حیات نام دارد
- سومین نقطه ی لنگرگاهی قابلیت عملیاتی اولیه است .

۳۸.۲ مدل بسط همزمان را توضیح دهید؟

مدل فرآیند همزمانی را می توان به عنوان یک سری فعالیت هایی فنی اصلی، وظایف و حالت های مرتبط با آن نمایش داد .

۳۹.۲ بسط مبتنی بر مولفه ها

فناوری شی گرا چارچوبی فنی برای مدل فرآیند مبتنی بر مولفه جهت مهندسی نرم افزار فراهم می کند .
مدل بسط مبتنی بر مولفه ها بسیاری از ویژگی های مدل مارپیچی را در خود دارد .
مدل بسط مبتنی بر مولفه ها برنامه های کاربردی را با استفاده از مولفه های نرم افزاری بسته بندی شده می سازد .

۴۰.۲ مدل روشهای رسمی

مدل روشهای رسمی شامل مجموعه ای از فعالیتها است که به مشخص کردن ریاضی و رسمی نرم افزار کامپیوتری منجر می شود .

۴۱.۲ علم مهندسی نرم افزار را تعریف کنید ؟

مهندسی نرم افزار علمی است که فرآیند، روشها و ابزار ها را با هم تلفیق می سازد تا نرم افزار کامپیوتر را پدید آورد .

۴۲.۲ اعمال مباحثاتی در مدل مارپیچی winwin

- شناسایی واگذارنده کلیدی سیستم یا زیر سیستم
- تعیین شرایط برد واگذارنده
- بحث و گفتگو درباره شرایط برد واگذارنده

۳ فصل سوم - مفاهیم مدیریت پروژه

۱.۳ در مدیریت پروژه های نرم افزاری بر چند نکته تاکید می شود ؟

چهار نکته :

- افراد
- محصول
- فرآیند
- پروژه

۲.۳ مدل بلوغ مدیریت افراد، چه زمینه های کلیدی را برای نرم افزار نویسان تعیین می کند؟

- یافتن افراد جدید
- انتخاب
- مدیریت اجرا
- آموزش
- جبران
- توسعه حرفه ای
- سازماندهی و طراحی کار
- توسعه تیمی و فرهنگی

۳.۳ بازیگران فرآیند نرم افزار

- مدیران ارشد
- مدیران پروژه
- سازندگان
- مشتریان
- کاربران نهایی

۴.۳ مدل MOI برای رهبری

- انگیزش
- سازماندهی
- ایده ها و نوآوری ها

۵.۳ چهار نکته ی کلیدی در مدیریت پروژه

- حل مسئله
- هویت مدیریتی
- نیل به اهداف
- تاثیر و ساخت تیم

۶.۳ ساختار بهترین تیم به چه عواملی بستگی دارد؟

- به شیوه ی مدیریت سازمان
- به تعداد افراد تشکیل دهنده تیم
- سطوح مهارتی آنان
- میزان کلی دشواری مسئله

۷.۳ سازماندهی کلی برای یک تیم بر اساس نظریه ی مانتی

- تمرکز زدایی دموکراتیک (DD)
 - تیم مهندسی نرم افزار فاقد رهبری دائمی است . تصمیم گیری درباره ی مشکلات و روش کار از طریق اجماع گروه صورت می پذیرد .
- تمرکز زدایی کنترل شده (CD)
 - تیم مهندسی نرم افزار دارای رهبری مشخص است که وظایف معینی را هماهنگ می کند ، حل مشکلات از طریق فعالیت های گروهی است .
- تمرکز کنترل شده (CC)
 - هماهنگ سازی داخلی تیم و حل مشکلات سطح بالا بر عهده مدیر تیم است .

۸.۳ بر اساس نظریه ی مانتی چه عواملی را باید هنگام طرح ریزی ساختار تیم های مهندسی نرم افزار در نظر گرفت؟

- دشواری مسئله ای که قرار است حل شود
- اندازه برنامه های حاصل بر حسب تعداد خطوط کد یا تعداد نقاط تابع
- زمان کنار هم ماندن اعضای تیم
- میزان قابلیت پیمانه ای مسئله (تقسیم پذیری مسئله به اجزای کوچکتر)
- کیفیت و قابلیت اطمینان مورد نیاز برای سیستمی که قرار است ساخته شود
- قطعیت تاریخ تحویل
- میزان ارتباطات مورد نیاز برای پروژه

۹.۳ بر اساس نظریه ی کنستانتین، چهار الگوی سازمانی برای تیمهای مهندسی نرم افزار را توضیح دهید؟

- الگوی بسته
 - هنگام ساخت نرم افزارهایی خوب عمل می کند که نیاز به کارهای روتین دارد ولی برای نوآوری و کارهای جدید احتمال موفقیت کمتر است .
- الگوی تصادفی
 - به ظرفیت تک تک افراد وابسته است . در صورت نیاز به کشفیات فنی یا نوآوری، تیمی که از الگوی تصادفی پیروی می کند بهترین نتیجه را می دهد ولی در صورتی که کارکردی منظم مورد نیاز باشد، چنین تیمی به دردسر می افتد .
- الگوی باز
 - در الگوی باز هدف بر آن است که در عین دستیابی به نتایج الگوی بسته، به نوآوری های الگوی تصادفی نیز برسیم . در این الگو کارها از طریق همکاری و ارتباطات سنگین و تصمیم گیری های مبتنی بر اجماع انجام می پذیرد . الگوی باز برای حل مسائل پیچیده مناسب است ولی ممکن است بازدهی تیم های دیگر را نداشته باشد .
- الگوی همزمان
 - اعضای تیم دسته بندی می شوند و هرکدام روی بخشی از مسئله یا پروژه کار می کنند . ارتباطات بین دسته ها بسیار کم می باشد .

۱۰.۳ برای دستیابی به تیمی با کارایی بالا

- اعضای تیم باید به یکدیگر اطمینان داشته باشند
- توزیع مهارت ها مناسب حل مسئله باشد
- اگر قرار است یکپارچگی حفظ شود، افراد خودکامه باید از تیم بروند

۱۱.۳ تیم ژل شده چه نوع تیمی است؟

- تیم ژل شده گروهی از افراد است، که چنان به هم پیوند خورده اند که کلیت آنها بزرگتر از حاصل جمع تک تک آنهاست
- با ژل شدن یک تیم احتمال موفقیت بالا می رود، نیازی به مدیریت آنها به شیوه های سنتی نیست، و قطعاً نیازی به انگیزش ندارند، آنها تکانه ی لازم را دارند.

۱۲.۳ پنج عامل مسمومیت تیم از دیدگاه جکمن

- فضای کاری آشفته که در آن انرژی اعضای تیم هدر می رود، و توجه آنان را از اهداف کاری دور می کند
- ناراحتی زیاد ناشی از عوامل فنی، کاری و پرسنلی
- رویه های قطعه قطعه شده و با هماهنگی ضعیف، یا مدل فرآیندی که خوب مشخص نشده باشد
- تعریف ناواضح نقشها که منجر به فقدان مسئولیت پذیری می شود
- قرار گرفتن پیوسته در معرض شکست که منجر به از دست رفتن اعتماد به نفس می شود

۱۳.۳ یک عامل مسمومیت تیم دیگر

علاوه بر ۵ سمی که جکمن توصیف می کند، تیم نرم افزاری غالباً با اختلاف تمایلات انسانی در اعضا نیز مشکل دارد، ذکر این نکته اهمیت دارد که شناخت تفاوت های انسانی نخستین گام در راه ایجاد تیمهایی است که ژل می شوند

۱۴.۳ مجموعه ای از تکنیک های هماهنگ سازی پروژه از دیدگاه کراول و استریتر

- روشهای رسمی غیر شخصی
 - شامل :
 - * مستندات مهندسی نرم افزار
 - * یادداشتهای فنی
 - * نقاط عطف پروژه
 - * ابزارهای کنترل پروژه
 - * برنامه های زمانبندی
 - * گزارش خطاها
 - می شود
- روشهای رسمی بین اشخاص
 - شامل جلسات مرور وضعیت و طراحی و بازرسی کد می شود .
- روشهای غیر رسمی بین اشخاص
 - شامل جلسات گروهی برای پخش اطلاعات و حل مسئله و جمع آوری خواسته های پرسنل
- ارتباط الکترونیکی
 - شامل پست الکترونیکی، سیستم های ویدئو کنفرانس و . . .
- شبکه ی بین اشخاص
 - بحث های غیر رسمی با اعضای تیم و آنان که در خارج از پروژه هستند ولی ممکن است دارای تجربه یا دیدگاهی باشند که می تواند به اعضای تیم کمک کند .

۱۵.۳ دامنه کاربرد نرم افزار

نخستین فعالیت مدیریت نرم افزار، تعیین دامنه کاربرد نرم افزار است .
دامنه کاربرد نرم افزار با پاسخ دادن به چه پرسش هایی مشخص می شود .

- محیط
- اهداف اطلاعات
- عملکرد و کارایی

۱۶.۳ مدیر پروژه با در نظر گرفتن کدام موارد، مدل فرآیند نرم افزار را انتخاب می کند ؟

- مشتریانی که محصول را درخواست کرده اند و کسانی که کار را انجام می دهند
- ویژگی های خود محصول
- محیط پروژه که تیم نرم افزاری در آن کار می کند

۱۷.۳ در یک پروژه کوچک، برای فعالیت ارتباط با مشتری نیاز به چه وظایف کاری می باشد ؟

- تهیه ی لیستی از مسائل مربوط به روشن کردن امور
- ملاقات با مشتری برای پرداختن به مسائل مربوط به روشن کردن امور
- تهیه بیان حوزه
- مرور بیان حوزه با همه دست اندرکاران
- اصلاح بیان حوزه در صورت نیاز

۱۸.۳ در یک پروژه پیچیده و بزرگ، برای فعالیت ارتباط با مشتری نیاز به چه وظایف کاری می باشد ؟

- بازبینی درخواست مشتری
- طرح ریزی و زمانبندی یک ملاقات رسم با مشتری
- پژوهش برای تعیین راهکار پیشنهادی و روشهای موجود
- آماده سازی یک سند کاری و برنامه برای ملاقات رسمی
- اجرای ملاقات
- تهیه ی مشخصات جزئی که ویژگی های رفتاری، عملیاتی و داده ای نرم افزار را منعکس کند
- بازبینی هر مشخصات جزئی برای تصحیح، سازگاری و عدم ابهام
- مونتاژ مشخصات جزئی در مستندات تعیین حوزه
- بازبینی مستندات تعیین حوزه
- اصلاح مستندات تعیین حوزه بنا به نیاز

۱۹.۳ علائم در خطر بودن پروژه از منظر جان ریل

- افراد نرم افزاری نیازهای مشتری را درک نمیکنند
- حوزه محصول خوب تعیین نشده است
- تغییرات خوب مدیریت نمی شود
- فناوری انتخاب شده تغییر می کند
- نیازهای تجاری تغییر می کند، یا خوب تعریف نمی شوند
- مهلت ها واقع بینانه نیست
- کاربران مقاومت می کنند
- حمایت مالی از بین می رود یا هرگز به طور مناسب به دست نیامده است
- تیم پروژه فاقد افرادی با مهارتهای لازم است
- مدیران و سازندگان از بهترین کوشش ها و دروس فراگرفته شده پرهیز می کنند

۲۰.۳ قاعده ۹۰-۹۰

۹۰% اول یک سیستم، ۹۰% زمان و کار را به خود اختصاص می دهد

۲۱.۳ روش ۵ بخشی جان ریل مبتنی بر عقل سلیم برای پروژه های نرم افزاری

- در آغاز کار درست گام بردارید
- تکانه را حفظ کنید
- پیشرفت کار را دنبال کنید
- هوشمندانه تصمیم گیری کنید
- تحلیلی نوگرایانه ارائه دهید

۲۲.۳ اصل W5HH

روش پیشنهادی بوهم، که اهداف پروژه، نقاط عطف و زمانبندی ها، مسئولیت ها، روش فنی و مدیریتی و منابع مورد نیاز را در بر می گیرد که پس از یک سری پرسش ها منجر به تعریف ویژگی های کلیدی پروژه و طرح پروژه می شود .

- این سیستم چرا توسعه می یابد؟
 - چه چیزی و در چه زمانی انجام خواهد شد؟
 - چه کسی مسئول یک عمل است؟
 - از لحاظ سازمانی چه جایگاهی دارند؟
 - کار از نظر مدیریتی و فنی چگونه انجام خواهد شد؟
 - از هر منبع چه میزان لازم است؟
- اصل W5HH برای همه ی پروژه های نرم افزاری، بدون توجه به اندازه و پیچیدگی آنها، قابل استفاده است .

۲۳.۳ مجموعه پرسش ها برای اینکه سازمانی تعیین کند که آیا پروژه ای اعمال بحرانی را پیاده سازی کرده است ؟

- مدیریت ریسک رسمی
- ریسک های اصلی که پروژه را تهدید می کنند، کدام است ؟
- برآورد زمانبندی و هزینه ی تجربی
- برآورد هزینه و زمان ساخت نرم افزار چقدر است ؟
- مدیریت پروژه مبتنی بر معیارها
- معیارهایی که مشکلات تکاملی را نشان می دهد چیست ؟
- دنبال کردن ارزش به دست آمده
- آیا ارزش های به دست آمده را گزارش می کنیم ؟
- دنبال کردن نقایص در برابر اهداف کیفیتی
- آیا تعداد نقایص در هر بازرسی را پیگیری و گزارش می کنید ؟
- مدیریت برنامه آگاهی از افراد
- بازدهی کارمندان در توسعه ی نرم افزار چقدر است ؟

۲۴.۳ چهار مورد که تاثیری اساسی بر مدیریت پروژه های نرم افزاری دارند؟

- افراد
- محصول
- فرآیند
- پروژه

۴ فصل چهارم - معیارهای پروژه و فرآیند نرم افزار

۱.۴ اندازه گیری در سرتاسر پروژه چه نتایجی دارد؟

- کمک به برآورد
- کنترل کیفیت
- ارزیابی بهره وری
- کنترل پروژه

۲.۴ چهار دلیل برای اندازه گیری فرآیندهای نرم افزاری، محصولات و منابع را نام ببرید؟

- مشخص کردن ویژگی ها
- ارزیابی کردن
- پیش بینی کردن
- بهبود بخشیدن

۳.۴ موازین، معیارها و شاخصها

میزان عبارت از کمیتی است که نشانگر حد، مقدار، ابعاد، ظرفیت یا اندازه صفتی از محصول یا فرآیند است . معیار چنین تعریف می شود :

میزان کمی از حدی که یک سیستم، مولفه یا فرآیند می تواند دارای یک صفت مفروض باشد . شاخص، معیار یا ترکیبی از معیارهاست که درکی از فرآیند نرم افزار، پروژه نرم افزاری یا خود محصول به دست می دهد .

۴.۴ معیار

معیار ها را باید طوری جمع آوری کرد که شاخص های فرآیند و محصول را بتوان کشف کرد، شاخصهای فرآیند به سازمان مهندسی نرم افزار این امکان را می دهند که از بازدهی فرآیند موجود دیدی درست به دست آورند ،

مدیران و سازندگان را قادر می سازند تا بتواند ارزیابی کنند چه چیزهایی عملی هستند و چه چیزهایی نیستند . معیارهای فرآیند از همه ی پروژه ها و در مدت زمان طولانی جمع آوری می شوند . هدف از استفاده از آنها فراهم آوردن شاخصهایی است که منجر به بهبود فرآیند نرم افزار در بلند مدت می شود . شاخصهای پروژه مدیر پروژه را قادر می سازد تا :

- به وضعیت پروژه در حال پیشرفت دست پیدا کند
- خطرات بالقوه را دنبال کند
- نواحی مشکل آفرین را پیش از بحرانی شدن آنها کشف کند
- جریان کار یا وظایف را تنظیم کند
- توانایی تیم پروژه در کنترل کیفیت محصولات نرم افزار را ارزیابی کند

۵.۴ بهبود بخشیدن به فرآیند

تنها راه موجه برای بهبود بخشیدن به هر فرآیندی، اندازه گیری صفات مشخصی از آن فرآیند است ، توسعه یک مجموعه معیارهای با معنی بر اساس این صفات ، و سپس استفاده از این معیارها برای فراهم آوردن شاخصهایی است، که منجر به راهبردی برای بهبود کار می شوند .

۶.۴ بازدهی یک فرآیند نرم افزار را توضیح دهید؟

بازدهی یک فرآیند نرم افزار را به طور غیر مستقیم اندازه گیری می کنیم .
یعنی یک مجموعه معیارها را بر اساس نتایج به دست می آوریم که از فرآیند قابل حصول هستند . شامل :

- موازین خطاهای کشف شده، پیش از ارائه نرم افزار
- نقایص گزارش شده توسط کاربران نهایی
- محصولات کاری تحویل شده
- انرژی انسانی صرف شده
- زمان صرف شده
- مطابقت با برنامه زمان بندی

۷.۴ معیارهای فرآیند توسط چه ویژگی هایی اندازه گیری می شوند ؟

- انرژی و زمان صرف شده در اجرای فعالیت های چتری
- فعالیت های کلی مهندسی نرم افزار

۸.۴ چند مورد از معیار های خصوصی را نام ببرید؟

- تعداد نقایص (توسط فرد)
- تعداد نقایص (توسط پیمانہ)
- خطاهای یافت شده در اثنای توسعه

۹.۴ معیارهای عمومی

معیارهای عمومی اطلاعاتی را جذب می کنند که در آغاز برای افراد جنبه خصوصی داشتند .

۱۰.۴ بهبود آماری فرآیند نرم افزاری یا SSPI را توضیح دهید؟

هرچه سازمانی در جمع آوری و استفاده از معیارهای فرآیند احساس راحتی بیشتری داشته باشد ، به دست آوردن شاخصهای ساده روشی دقیق تر را نتیجه می دهد که به آن SSPI می گویند .
SSPI از تحلیل شکست نرم افزار برای جمع آوری داده های مربوط به همه ی خطاها و نقایص مشاهده شده در اثنای توسعه و به کارگیری سیستم یا محصول استفاده می کند .

۱۱.۴ مراحل تحلیل شکست را توضیح دهید ؟

- همه ی خطاها و نقایص ، بر حسب منشا، دسته بندی می شوند ، مثل :
 - نقص در مشخصات
 - نقص در منطق
 - عدم تطابق با استاندارد ها
- هزینه تصحیح هر خطا و نقص ثبت می شود
- تعداد خطاها و نقایص در هر دسته شمارش و به ترتیب نزولی مرتب می شوند
- هزینه کلی خطا و نقایص در هر گروه محاسبه می شود
- داده های حاصل مورد تحلیل قرار می گیرند تا دسته هایی که منجر به بالاترین هزینه ها در سازمان می شوند، کشف شوند .
- تدابیری برای اصلاح فرآیند با هدف حذف دسته ای از خطاها و نقایص که بیشترین هزینه را دارند اندیشیده می شود .

۱۲.۴ نمودار ستون فقرات ماهی گریدی را توضیح دهید؟

نمودار ستون فقرات ماهی (خط مرکزی) نشانگر عامل کیفیت مورد نظر است ، هر یک از دنده ها که به ستون فقرات متصل است نشانگر علل بالقوه برای مشکل کیفیتی است (از قبیل کمبود خواسته ها، مشخصه ای مبهم ، خواسته های نادرست ، تغییر خواسته ها) سپس ستون فقرات و دنده ها به هر یک از دنده های اصلی نمودار، افزوده می شوند تا علت آن گسترش داده شود .

با تحلیل یک نمودار استخوان ماهی، می توان شاخصهایی به دست آورد که سازمان نرم افزاری را قادر می سازد که در جهت کاهش دادن فرکانس خطاها و نقایص فرآیند را اصلاح کند .

۱۳.۴ هر پروژه باید چه مواردی را اندازه گیری کند؟

- ورودی ها
- خروجی ها
- نتایج

۱۴.۴ اندازه گیری های جهان فیزیکی را به چند گروه می توان تقسیم کرد ؟

- موازین مستقیم
 - موازین غیر مستقیم
- موازین مستقیم فرآیند مهندسی نرم افزار، شامل هزینه و انرژی به کار رفته است .
- موازین مستقیم محصول، شامل تعداد خطوط کد تولید شده (LOC) ، سرعت اجرا ، اندازه حافظه و نقایص گزارش شده در یک مدت زمان تعیین شده است .
- موازین غیر مستقیم محصول عبارتند از : عملکرد ، کیفیت ، پیچیدگی ، بازدهی ، قابلیت نگهداری

۱۵.۴ تقسیم بندی دامنه معیارهای نرم افزار

- معیار های فرآیند
- معیار های پروژه
- معیار های محصول

۱۶.۴ معیار های جنبه ی خصوصی، معیارهای جنبه ی عمومی

معیارهایی از محصول که برای افراد جنبه ی خصوصی دارند غالباً ترکیب می شوند و معیارهای پروژه را توسعه می دهند که برای تیم نرم افزاری جنبه ی عمومی دارند .

۱۷.۴ معیارهای اندازه گرا

معیارهای اندازه گرا از نرمال سازی موازین کیفیتی و یا بهره وری و با در نظر گرفتن اندازه نرم افزار به دست می آیند .

۱۸.۴ تعداد خطوط کد

برای توسعه ی معیارهایی که با معیارهای مشابه حاصل از پروژه های دیگر قابل امتزاج باشد، تعداد خطوط کد را به عنوان مقدار نرمال سازی انتخاب می کنیم .

۱۹.۴ مجموعه ای از معیارهای اندازه گیری که برای هر پروژه می توان استفاده کرد؟

- تعداد خطا به ازای هر هزار خط کد
- نقایص به ازای هر هزار خط کد
- هزینه ی مصرفی برای هر خط
- تعداد صفحات مستند به ازای هر هزار خط کد
- تعداد خطاها به ازای هر نفر در ماه
- تعداد خط به ازای هر نفر در ماه
- هزینه ی مصرفی به ازای هر صفحه از مستندات

۲۰.۴ معیارهای عملکردگرا

معیارهای عملکردگرا از میزان عملکرد ارائه شده توسط برنامه کاربردی، به عنوان مقداری برای نرمال سازی استفاده می کنند .

۲۱.۴ نقطه عملکرد (Function Point) چیست ؟

نقاط عملکرد، با استفاده از یک رابطه ی تجربی مبتنی بر موازین قابل شمارش (مستقیم) از دامنه ی اطلاعات نرم افزار و ارزیابی پیچیدگی نرم افزار به دست می آیند .
 هنگامی که نقاط عملکرد محاسبه شدند، از آنها به شیوه ای مشابه LOC برای نرمال سازی موازین مربوط به بهره برداری نرم افزار، کیفیت و صفات دیگر استفاده می شود .

۲۲.۴ مقادیر دامنه اطلاعات به چه صورت تعیین می شوند؟

- تعداد ورودی های کاربر
- تعداد خروجی های کاربر
- تعداد درخواست های کاربر
- تعداد فایلها
- تعداد واسط های خارجی

۲۳.۴ هدف اصلی مهندسی نرم افزار

هدف اصلی مهندسی نرم افزار، تولید سیستم، برنامه کاربردی یا محصولی با کیفیت بالاست .

۲۴.۴ مجموعه عوامل کیفیتی مک کال و کارائو برای توسعه ی کیفیت نرم افزار، نرم افزار را از چند دیدگاه ارزیابی می کند ؟

- کارکرد محصول (استفاده از آن)
- بازیابی محصول (تغییر دادن آن)
- گذار محصول (اصلاح آن برای کار کردن در محیطی متفاوت)

۲۵.۴ شاخص های کیفیت نرم افزار را نام ببرید ؟

- درستی
- قابلیت نگهداری
- یکپارچگی
- قابلیت استفاده

۲۶.۴ درستی

یک برنامه باید درست کار کند وگرنه برای کاربران خود ارزشی ندارد ، متداولترین میزان برای درستی ، تعداد نقایص به ازای هر KLOC می باشد .

۲۷.۴ قابلیت نگهداری

قابلیت نگهداری عبارت است از سهولت تصحیح یک برنامه در صورت مواجهه با خطا، تطابق با تغییرات محیط یا بهبود بخشیدن به برنامه در صورت تغییر یافتن خواسته های مشتری .
راهی برای اندازه گیری مستقیم قابلیت نگهداری وجود ندارد، بنابراین باید از موازین غیر مستقیم استفاده کرد
یک معیار مبتنی بر زمان ، میانگین زمان لازم برای تغییر (MTTC) است، یعنی زمان لازم برای :

- تحلیل تقاضای تغییر
- طراحی یک اصلاح مناسب
- پیاده سازی تغییر
- آزمون آن
- و توزیع تغییر در میان همه ی کاربران

۲۸.۴ یکپارچگی

یکپارچگی، میزان توانایی سیستم برای تحمل حملاتی (چه تصادفی و چه عمدی) است که متوجه امنیت آن می شود .
این حملات روی هر سه مولفه نرم افزار صورت می پذیرد :

- برنامه ها
- داده ها
- مستندات

۲۹.۴ قابلیت استفاده

اگر قابلیت استفاده از برنامه ای آسان نباشد، سرنوشت آن شکست خواهد بود .
قابلیت استفاده کوششی برای تعیین کمی سهولت استفاده بر حسب چهار خصوصیت قابل اندازه گیری است :
• مهارت فیزیکی و هوشی لازم برای فراگیری سیستم
• زمان لازم برای بازدهی خوب در به کار گیری سیستم
• افزایش خالص در بهره وری نسبت به روشی که جایگزین می شود
• ارزیابی موضوعی از احساس کاربران نسبت به سیستم (که غالباً از طریق پرسشنامه به دست می آید)

۵ فصل پنجم - برنامه ریزی برای پروژه ی نرم افزاری

۱.۵ برنامه ریزی پروژه چیست؟

مدیریت پروژه نرم افزاری با مجموعه ای از فعالیت ها آغاز می شود که جمعاً برنامه ریزی برای پروژه نامیده می شود .

۲.۵ برنامه ریزی پروژه شامل چه مواردی می شود ؟

شامل برآوردی برای تعیین میزان پول، کار ، منابع و زمان لازم برای ساخت یک محصول یا سیستم نرم افزاری مشخص

۳.۵ هنگام انتخاب مدیر پروژه کدام ویژگی اهمیت بیشتری دارد ؟

- بداند چه چیزی خراب می شود، قبل از آنکه این اتفاق رخ دهد
- جرات برآورد آینده را داشته باشد

۴.۵ برآورد منابع، هزینه و زمانبندی برای کارهای مهندسی نرم افزار به چه ویژگی هایی نیاز دارد؟

- تجربه
- دستیابی به اطلاعات تاریخی خوب
- جرات انجام پیش بینی های کمی در هنگام وجود اطلاعات صرفاً کیفی

۵.۵ نکته - عدم قطعیت در برآورد

برآورد ذاتاً با ریسک همراه است و همین ریسک به عدم قطعیت منجر می شود

۶.۵ تاثیر پیچیدگی بر عدم قطعیت

پیچیدگی پروژه اثری شدید بر عدم قطعیت ذاتی برنامه ریزی دارد .

۷.۵ نسبی بودن پیچیدگی

پیچیدگی یک میزان نسبی است که از آشنایی با کارهای قبلی تاثیر می پذیرد .

۸.۵ تاثیر اندازه پروژه بر برآورد

اندازه پروژه عامل مهم دیگری است که می تواند ، صحت و موثر بودن برآورد ها را تحت تاثیر قرار دهد .

۹.۵ درجه ی عدم قطعیت ساختاری

درجه ی عدم قطعیت ساختاری نیز بر ریسک برآورد تاثیر می گذارید .

۱۰.۵ منظور از ساختار در کلمه ی 'درجه ی عدم قطعیت ساختاری'

- درجه ای از یکنواختی خواسته ها
- سهولت تقسیم کردن وظایف
- ماهیت سلسله مراتبی اطلاعاتی

۱۱.۵ امکان سنجی یا امکان پذیری نرم افزار چند بعد دارد ؟

- فناوری
- مالی
- زمان
- منابع

۱.۱۱.۵ فناوری

- آیا پروژه از نظر فنی امکان پذیر است ؟
- آیا می تواند پیشرفت داشته باشد ؟
- آیا نقایص را می توان تا آن حد کاهش داد، که با نیازها تطبیق پیدا کند ؟

۲.۱۱.۵ مالی

- آیا از نظر مالی امکان پذیر است ؟
- آیا می توان پروژه را با هزینه ی مناسب برای مشتری به پایان رساند ؟

۳.۱۱.۵ زمان

- آیا زمان تحویل پروژه به بازار مناسب هست ؟

۴.۱۱.۵ منابع

- آیا سازمان دارای منابع انسانی، سخت افزاری ، نرم افزاری و . . . لازم برای موفقیت هست ؟

۱۲.۵ منابع

دومین هدف برنامه ریزی برای نرم افزار، برآورد منابع لازم برای توسعه ی نرم افزار است .

۱۳.۵ انواع منابع توسعه

- محیط توسعه (ابزار های سخت افزاری و نرم افزاری)
- مولفه های نرم افزاری قابل استفاده مجدد
- افراد

۱۴.۵ ویژگی های مشخص کننده منابع را نام ببرید؟

- توصیف منبع
- بیانی از قابلیت دسترسی
- موعد زمانی نیاز به آن منبع
- مدت زمان به کارگیری آن

۱۵.۵ منابع نرم افزاری قابل استفاده مجدد

یعنی ایجاد و استفاده مجدد از مولفه های سازنده مولفه های سازنده :

- برای رجوع آسان در یک کاتالوگ لیست می شوند
- برای کاربرد آسان استاندارد می شوند
- برای یکپارچگی اعتبار سنجی می شوند

۱۶.۵ چهار دسته از مولفه های نرم افزاری قابل استفاده مجدد

- مولفه های آماده
- مولفه های تجربه کامل
- مولفه های تجربه ناقص
- مولفه های جدید

۱۷.۵ مولفه های آماده

نرم افزارهای موجود که از یک شخص ثالث قابل دسترسی هستند یا برای یک پروژه قبلی توسعه یافته اند .

۱۸.۵ مولفه های تجربه کامل

مولفه های توسعه داده شده برای پروژه های گذشته که مشابه نرم افزار فعلی هستند، اعضای تیم نرم افزار فعلی در حیطه ی کاربرد ارائه شده توسط این مولفه ها تجربه کامل دارند و اصلاحات مورد نیاز برای این مولفه ها، ریسک نسبتاً پایینی دارد .

۱۹.۵ مولفه های تجربه ناقص

مولفه هایی که برای پروژه های گذشته توسعه داده شده اند و با نرم افزار پروژه فعلی در ارتباط اند ولی نیاز به اصلاحات اساسی دارند . اعضای تیم نرم افزار فعلی در حیطه ی کاربرد این مولفه ها تجربه محدودی دارند ، از این رو اصلاحات مورد نیاز برای مولفه های تجربه ناقص، دارای درجه ی بالایی از ریسک هستند .

۲۰.۵ مولفه های جدید

مولفه های نرم افزاری که تیم نرم افزاری باید مشخصاً برای نیازهای پروژه فعلی بسازند .

۲۱.۵ دستورالعمل هایی برای استفاده از مولفه های قابل استفاده مجدد

- اگر مولفه های آماده نیازهای پروژه را برآورده می کنند آنها را به دست آورید ، هزینه ی به دست آوردن و سرهم کردن مولفه های آماده همواره کمتر از هزینه ی توسعه ی نرم افزار هم ارز است . به علاوه درجه ی ریسک هم پایین است .
- اگر مولفه های تجربه کامل در دسترس اند، ریسک ناشی از اصلاح و سرهم بندی عموماً قابل قبول است ، در برنامه ریزی پروژه باید استفاده از این مولفه ها منعکس شود .
- اگر مولفه های تجربه ناقص در دسترس باشند، استفاده از آنها برای پروژه فعلی باید مورد تحلیل قرار گیرد ، هزینه ی اصلاح مولفه های تجربه ناقص، گاه می تواند بیش از هزینه ی لازم برای توسعه ی یک نرم افزار جدید باشد .

۶ نمونه سوالات

۱. بحران نرم افزار و عوامل موثر بر آن را تشریح کنید ؟
۲. نرم افزار های بلادرنگ و توکار را تشریح کنید ؟
۳. مدل ساخت نمونه و مدل افزایشی را شرح داده و فازهای آن را با رسم شکل بیان نمایید ؟
۴. بر اساس نظریه ی مانتی چهار الگوی سازمانی برای تیم های مهندسی نرم افزار را تشریح کنید ؟
۵. اصل $W5HH$ را تشریح کنید ؟
۶. امکان سنجی برای یک پروژه نرم افزاری چیست ؟ ابعاد آن کدام است، تشریح نمایید ؟
۷. موازین کیفیت نرم افزار را نام برده و یکی را به دلخواه تشریح کنید ؟
۸. انواع منابع نرم افزاری قابل استفاده مجدد را نام برده و تشریح نمایید ؟
۹. نرم افزار های سیستمی و زمان حقیقی را تشریح نمایید ؟
۱۰. مدل ساخت ترتیبی خطی و مدل افزایشی را شرح داده و فازهای آن را با رسم شکل بیان کنید ؟
۱۱. منظور از قابلیت نگهداری و یکپارچگی نرم افزار چیست ؟