سوالات تستی

جواب سوالات تستی

1. B
2. D
3. C
4. D
5. C

سوال ۱

تفاوت اساسی بین نرم‌افزار و سخت‌افزار با توجه به زمان چیست؟

جواب سوال ۱

تغییرپذیری

نرم‌افزارها به راحتی قابل تغییر و به‌روزرسانی هستند، در حالی که سخت‌افزارها برای تغییر نیاز به تعویض فیزیکی یا ارتقا دارند. با گذشت زمان، این انعطاف‌پذیری نرم‌افزار امکان پاسخ‌گویی به نیازهای جدید را فراهم می‌کند.

عمر مفید

سخت‌افزارها دارای یک عمر مفید فیزیکی هستند و با گذشت زمان فرسوده می‌شوند. در مقابل، نرم‌افزار فرسودگی فیزیکی ندارد اما ممکن است به دلیل تغییرات فناوری یا نیازهای کاربری، منسوخ شود.

هزینه‌های به‌روزرسانی و نگهداری

در طول زمان، هزینه‌های نگهداری و به‌روزرسانی نرم‌افزار می‌تواند بسیار بیشتر از هزینه‌های اولیه توسعه آن باشد. در حالی که هزینه‌های سخت‌افزار بیشتر به خرید و نصب اولیه محدود می‌شود.

وابستگی به فناوری

نرم‌افزارها معمولاً به سرعت تحت تأثیر تغییرات فناوری قرار می‌گیرند. اما سخت‌افزار ممکن است برای مدت زمان طولانی‌تری قابل استفاده باقی بماند، حتی اگر فناوری پیشرفت کند.

مقیاس‌پذیری

نرم‌افزارها معمولاً به راحتی قابل مقیاس‌پذیری هستند، به این معنی که می‌توان آن‌ها را برای پاسخگویی به نیازهای در حال تغییر تنظیم کرد. سخت‌افزار اغلب نیاز به ارتقا یا تعویض دارد تا بتواند با نیازهای مقیاس بزرگ‌تر سازگار شود.

سوال ۲

توضیح دهید Prototyping در چه مواقعی می‌تواند در پیش‌برد پروژه موثر باشد؟

جواب سوال ۲

شفاف‌سازی نیازمندی‌ها

Prototyping به شناسایی و تصحیح نیازمندی‌های کاربران و مشتریان کمک می‌کند، به ویژه زمانی که این نیازمندی‌ها به‌طور کامل شناخته نشده باشند.

بازخورد سریع

پروتوتایپ‌ها امکان دریافت بازخورد سریع از کاربران و ذینفعان را فراهم می‌کنند، که این امر به بهبود و تکامل سریع‌تر محصول کمک می‌کند.

کاهش ریسک

ایجاد نمونه‌های اولیه به تیم توسعه کمک می‌کند تا ریسک‌های مربوط به فناوری یا طراحی را در مراحل اولیه پروژه شناسایی و رفع کنند.

ارتباط بهتر با ذینفعان

پروتوتایپ‌ها به تیم توسعه این امکان را می‌دهند تا ایده‌ها و پیشنهادات خود را به شکل عینی به ذینفعان نشان دهند، که این امر می‌تواند به ارتباط و درک متقابل بهتر کمک کند.

انعطاف‌پذیری در توسعه

استفاده از پروتوتایپ‌ها به تیم توسعه اجازه می‌دهد تا انعطاف‌پذیری بیشتری در تغییر جهت یا اصلاح طرح‌ها در طول فرآیند توسعه داشته باشند.

سوال ۳

چرا ممکن است یک سیستم با عمر طولانی به اسناد طراحی بیشتری نیاز داشته باشد؟

جواب سوال ۳

پشتیبانی و نگهداری

اسناد طراحی کامل و به‌روز به تیم‌های مختلف کمک می‌کنند تا درک بهتری از سیستم داشته باشند، به ویژه در زمان انتقال مسئولیت نگهداری. به خصوص زمان انتقال یک سیستم و ملحقاتش، اسناد طراحی باعث می‌شوند این انتقال مسئولیت، با درک خوبی همراه شود و جزئیات زیادی که ممکن است در نگاه‌های اول دیده نشوند، ثبت شده باشند و توسط تیم جدید، تشخیص داده شوند.

تغییرات و به‌روزرسانی‌ها

اسناد طراحی دقیق می‌توانند به ثبت تغییرات و اطمینان از انسجام کلی سیستم در طول زمان کمک کنند. در کنار کیفیت بالای داکیومنتیشن یا همون مستندات، کیفیت بالای به‌روزرسانی‌ها و تغییرات نیز می‌تواند رخ دهد با توجه به اسناد طراحی دقیق.

کاهش خطای انسانی

اسناد طراحی کامل به کاهش خطر از دست دادن دانش و تجربه مرتبط با سیستم و حفظ دانش حیاتی کمک می‌کنند. این مورد نیز مشابه مورد اول در زمان انتقال سیستم تاثیرگذار است.

سازگاری با محیط‌های جدید

اسناد طراحی به شناسایی بخش‌هایی از سیستم که نیاز به تغییر یا به‌روزرسانی دارند، کمک می‌کنند، به خصوص در زمان انطباق با فناوری‌های جدید.

سوال ۴

.A

چرا باید برای ایجاد یک نرم‌افزار براساس یک مدل پیش برویم و در طول پروژه پایبند به آن مدل باشیم؟

.B

یک تیم مهندس نرم‌افزار برای پروژه‌ای در یک شرکت نفت بزرگ دعوت شده است. این شرکت چندین دپارتمان دارد و تیم مهندسی نرم‌افزار با دپارتمان مدیریت اطلاعات (MIS) تعامل می‌کند. سیستم MIS این شرکت قدیمی (Legacy) است و هدف، انتقال داده‌ها به یک سیستم جدید (مهاجرت داده) است. فرآیندها، قراردادهای قانونی و معیارهای پذیرش این شرکت بسیار خاص و حساس هستند. به نظر شما چه مدل ایجاد نرم‌افزاری برای راه‌اندازی این سیستم انتقال داده را تیم مهندس نرم‌افزار انتخاب خواهد کرد؟ نام مدل و علت اصلی انتخاب آن کافی است.

.C

مهم‌ترین مشکلات مدل‌های سنتی (مثل مدل آبشاری) نسبت به مدل‌های چابک، چیست؟ ()اشاره به ۳ مورد و توضیح کامل آنها کفایت می‌کند.)

جواب سوال ۴

A. اهمیت استفاده از یک مدل در ایجاد نرم‌افزار

ساختار و نظم: مدل‌ها ساختار و نظمی به فرآیند توسعه نرم‌افزار می‌دهند که باعث می‌شود پروژه قابل پیش‌بینی و مدیریت‌پذیر باشد.

تعریف وظایف و مسئولیت‌ها: مدل‌ها وظایف و مسئولیت‌های اعضای تیم توسعه را مشخص می‌کنند.

کاهش ریسک: استفاده از مدل‌ها به شناسایی و مدیریت ریسک‌ها در مراحل اولیه پروژه کمک می‌کند.

B. مدل ایجاد نرم‌افزار برای شرکت نفتی

مدل انتخابی: مدل ترکیبی (Hybrid Model) که ترکیبی از مدل آبشاری و چابک است.

دلیل انتخاب:

- **پیچیدگی و حساسیت بالا:** با توجه به پیچیدگی و حساسیت‌های موجود در فرآیندها و داده‌های شرکت نفتی، مدل ترکیبی اجازه می‌دهد تا بخش‌های حساس و پیچیده با دقت بالا و با رویکرد آبشاری پیاده‌سازی شوند.
- **انعطاف‌پذیری:** در بخش‌های کمتر حساس، استفاده از رویکردهای چابک به تیم اجازه می‌دهد تا به سرعت به تغییرات پاسخ دهد.

C. مشکلات مدل‌های سنتی (مانند مدل آبشاری) نسبت به مدل‌های چابک

- **انعطاف‌پذیری کم:** مدل‌های سنتی اغلب انعطاف‌پذیری کمتری در برابر تغییرات دارند و تغییرات اساسی در مراحل پایانی پروژه دشوار است.
- **تأخیر در بازخورد:** در مدل‌های سنتی، بازخورد کاربران و ذینفعان معمولاً در مراحل پایانی پروژه جمع‌آوری می‌شود، که می‌تواند منجر به تأخیر در شناسایی و حل مشکلات شود.
- **ریسک بالا و هزینه‌های تغییر:** به دلیل تأخیر در دریافت بازخورد و انعطاف‌پذیری کم، ریسک شکست پروژه‌ها و هزینه‌های ایجاد تغییرات افزایش می‌یابد.

سوال ۵

با در نظر گرفتن رویکرد چابک به سوالات زیر پاسخ دهید:

A. در اکثر پروژه‌های نرم‌افزاری پیش‌بینی موارد زیر سخت است:

- اینکه کدام نیازمندی‌های مشتری تغییر خواهند کرد و کدام نیازمندی‌ها ثابت خواهند بود؟
- اینکه به چه میزان طراحی پیش از پیاده‌سازی احتیاج داریم؟
- و چه مقدار زمان از نظر برنامه‌ریزی برای تحلیل و طراحی، پیاده‌سازی و تست محصول نیاز خواهد بود؟

فرآیندهای چابک چگونه در جهت رفع این شرایط‌های نیاز به پیش‌بینی پاسخ می‌دهند؟

.B

اگر برای سیستم‌های بزرگ و با عمر طولانی که توسط یک شرکت نرم‌افزاری برای مشتریان خارجی توسعه داده می‌شوند، از رویکرد چابک استفاده شود، چه مشکلاتی ممکن است بوجود آید؟. ۳ مورد از مشکلات ممکن را ذکر کنید.

.C

فکر می‌کنید مدل‌های چابک خود چه مشکلاتی داشته باشند؟ (حداقل ۴ مورد)

جواب سوال ۵

.A

در فرآیندهای چابک، به جای تلاش برای پیش‌بینی دقیق تمام نیازمندی‌ها و طراحی‌ها از ابتدا، تمرکز بر توسعه تدریجی و تکراری محصول است. این رویکرد اجازه می‌دهد تا تیم‌ها به سرعت و با انعطاف‌پذیری بالا به تغییرات نیازمندی‌ها واکنش نشان دهند. زمان‌بندی برای تحلیل، طراحی، پیاده‌سازی و تست نیز به صورت انعطاف‌پذیر در چرخه‌های کوتاه مدت مدیریت می‌شود.

.B

برای سیستم‌های بزرگ و با عمر طولانی، استفاده از رویکرد چابک ممکن است مشکلاتی از قبیل:

- دشواری در مدیریت و هماهنگی بین تیم‌های بزرگ و پراکنده.
- چالش‌های مربوط به برقراری ارتباط مؤثر با مشتریان بین‌المللی.
- مسائل مربوط به مقیاس‌پذیری و ادغام مداوم تغییرات در یک سیستم بزرگ.

.C

مدل‌های چابک خود می‌توانند مشکلاتی داشته باشند از جمله:

- نیاز به ارتباط و همکاری مداوم و نزدیک با مشتری.
- دشواری در پیش‌بینی هزینه‌ها و زمان‌بندی‌های بلندمدت.
- خطر انحراف از اهداف اصلی در صورت عدم وجود برنامه‌ریزی دقیق.
- ممکن است کیفیت کد در اثر تغییرات مکرر کاهش یابد.
