

۷	۴	۱
جمع نمرات	۵	۲
	۶	۳



دانشگاه صنعتی شریف

شماره دانشجویی: ۹۹۱۵۲۲۰۷

نام و نام خانوادگی: ایمان محمدی

شماره صندلی: نام استاد: دکتر ریواده

شماره سوال: ۱) سه نوع مختلف هزینه‌هایی که ممکن است در راه افزایش کیفیت نرم افزار مجبوریم پرداخت کنیم: صفحه ۱ از ۸

۱) Prevention costs - هزینه‌های پیشگیری

این دسته از هزینه‌ها به فعالیت‌هایی اشاره دارد که قبل از وقوع هرگونه خطا انجام می‌شود تا از بروز آن جلوگیری کند
 برنامه‌ریزی کیفیت - quality planning

تمرین استراتژی‌ها و روش‌هایی برای اطمینان از کیفیت محصول نهایی
 بررسی‌های فنی رسمی - formal technical reviews

جلسات بررسی کد و طراحی برای یافتن و رفع ایرادات پیش از پیاده‌سازی
 تجهیزات تست - test equipment

خرید و نگهداری ابزارهایی برای انجام تست‌های نرم افزاری
 آموزش - training
 برگزاری دوره‌های آموزشی برای بهبود مهارت‌های کارکنان

مثال در E-commerce: یک شرکت مبتنی بر تجارت الکترونیک سرمایه‌گذاری می‌کند در آموزش تیم توسعه برای به کارگیری بهترین شیوه‌های برنامه‌نویسی و خرید ابزارهای تست خودکار برای تضمین کیفیت و بسایر یا اپلیکیشن خود

۲) Internal Failure Costs - هزینه‌های ناشی از شکست داخلی

این هزینه‌ها زمانی ایجاد می‌شوند که خطاهای داخل سازمان قبل از ارسال محصول به مشتری تشخیص داده شوند.

ترمیم و اصلاح - Rework and Repair

اصلاح خطاهای شناسایی شده در نرم افزار

تجزیه و تحلیل حالت شکست - Failure mode analysis

عوامل دلایل بروز خطاها و شکست‌ها

مثال در E-commerce: اگر در سیستم خریدی یک شرکت آنلاین خطایی شناسایی شود، باید فوراً

برای اصلاح آن هزینه‌هایی را متقبل شود تا از اختلال در عملیات فروش جلوگیری شود

۳) External Failure Costs - هزینه‌های ناشی از شکست خارجی

این هزینه‌ها زمانی بروز می‌کنند که محصول نقص دارد به دست مشتری رسیده و خطاها پس از تحویل محصول شناخته می‌شوند.

ادامه در صفحه بعد



حل شکایات - complaint resolution

پاسخگویی به شکایات مشتریان و رفع مشکل آن‌ها

بازگشت و جایگزینی محصول - product return and replacement

مدیریت بازگشت کالاهای نقص‌دار و ارسال جایگزین

پشتیبانی خط‌گرمی - help line support

ارائه پشتیبانی فنی به مشتریان برای حل مشکلات محصول

کارهای تحت پوشش گارانتی - warranty work

تعمیرات و خدماتی که در چارچوب گارانتی ارائه می‌شود

فصل در E-Commerce

اگر مشتریان یک فروشگاه آنلاین به دلیل مشکلات در سیستم پرداخت شکایت کنند، شرکت باید هزینه‌هایی را برای پشتیبانی و احتمالاً جایگزینی محصولات معیوب یا بازگرداندن وجه پرداختی به مشتریان و تقبل سود



الف - شناسایی نقض‌های خط‌مشی

۱) لمح انتقادی و تمرکز بر فرد - باید بر خطاها و نقض‌ها تمرکز شود نه بر فردی که خطا را انجام داده است. لمح انتقادی و اشاره مستقیم به فرد مسئول خطا، باعث ایجاد فضای منفی و دفاعی در تیم می‌شود.

۲) منحرف شدن از دستورکار و هدف جلسه - باید بر اساس دستورکار مستحق بررسی بود FTR و تمرکز اصلی بر بازبینی اسناد طراحی باشد. در این ستارویه، جلسه به سرعت به بحث‌های غیرمرتبط منحرف شده است.

۳) تبدیل جلسه FTR به جلسه حل مسئله - هدف از FTR، شناسایی مشکلات است نه حل آن‌ها در همان جلسه. این تبدیل جلسه FTR به جلسه حل مسئله می‌تواند باعث از بین رفتن تمرکز و افزایش طول جلسه شود.

ب - اقدامات مورد نیاز برای هماهنگی با خط‌مشی‌های FTR

- ۱) باید تأکید بر این باشد که خطاها و نقض‌ها به صورت بی‌طرفانه و بدون نسبت دادن آن‌ها به فرد خاصی بررسی شوند. هدف اصلی یافتن راه حل‌ها و بهبود فرآیند است نه ایجاد حس گناه در افراد
- ۲) جلسه باید با دستورکار مشخص و محدود به موضوعات مربوط به اسناد طراحی بررسی بوده. باید از پراکندگی بحث‌ها و قطع شدن موضوعات غیرمرتبط جلوگیری شود
- ۳) باید تأکید شود که هدف از جلسه FTR، شناسایی مشکلات است و نه حل آن‌ها در همان جلسه. برای حل مشکلات باید جلسات جداگانه‌ای ترتیب داده شود.

ج - تأثیر نقض خط‌مشی‌ها بر روند جلسه و پروژه

۱) این رفتار می‌تواند باعث ایجاد فضای ناآرام و دفاعی در تیم شود، که این خود مانع برای همکاری مورد نیاز است. همچنین ممکن است تمایل به استراک گذاری ایده‌ها و خلافت را کاهش دهد.

۲) منحرف شدن از دستورکار باعث می‌شود که جلسه کارآمدی خود را از دست بدهد و ممکن است برخی موضوعات مهم نادیده گرفته شوند. این می‌تواند روی زمان بندی پروژه و کیفیت کار تأثیر بگذارد

۳) تبدیل جلسه FTR به جلسه حل مسئله می‌تواند باعث افزایش زمان و کاهش تمرکز روی هدف اصلی جلسه شود. این موضوع باعث می‌شود که تأثیر منفی روی پیشرفت کلی پروژه بیش‌تر و باعث تأخیر در رسیدن به مراحل بعدی می‌شود



الف - عوامل رهنمین کیفیت که نقصی شده اند

۱) کیفیت نیازمندی ها - در این شماره ممکن نیازمندی های مطرح شده از سمت مدیر محصول، ناکافی یا ناکامل یا ناسازگار باشد که این اتفاقات افتاده و باعث سردرگمی کاربران و خطاهای ناشی شده

۲) کیفیت طراحی - ممکن است طراحی انجام شده برای این ویژگی جدید با نیازمندی های تعریف شده همخوانی نداشته باشد یا کیفیت لازم را نداشته باشد. به ویژه در زمینه رابط کاربری و تعامل با درگاه پرداخت

۳) کیفیت کد - کد زده شده باید مطابق با استانداردهای برنامه نویسی محلی باشد و ویژگی هایی داشته باشد که قابلیت نگهداری و تعمیر آن را فراهم کند. خطاهای پس از استقرار نشان می دهند که این مورد ممکن است نادیده گرفته شده باشد به خاطر سرعت کار شایع و دلایل دیگری نیز

ب - تغییرات مورد نیاز در فرآیند

۱) بهبود کیفیت نیازمندی ها - تعریف دقیق تر و کامل تر نیازمندی ها با درگیر کردن تمام ذی نفعان از جمله کاربران نهایی

استفاده از روش هایی مانند مصاحبه با کاربران و آشنایی نیازمندی ها برای اطمینان از صحت و کامل بودن آن ها

۲) بهبود کیفیت طراحی - اطمینان از همخوانی طراحی با نیازمندی ها و استفاده از بررسی های متنی برای ارزیابی کیفیت طراحی

تمرکز بر بهبود تعامل کاربر با ویژگی های جدید و ایجاد رابط کاربری واضح و قابل فهم

۳) ارتقاء کیفیت کد - استفاده از تست های خودکار و دستی برای اطمینان از اینکه کد نوشته شده با استانداردهای کیفیت مطابقت دارد.

انجام کد بررسی ها (code reviews) برای اطمینان از پایداری و قابلیت نگهداری کد

۴) بهبود ارزشمندی کنترل کیفیت - تخصیص منابع به سبدهای که اطمینان حاصل کند فرآیندهای کنترل کیفیت به طور کامل و موثر انجام می شوند

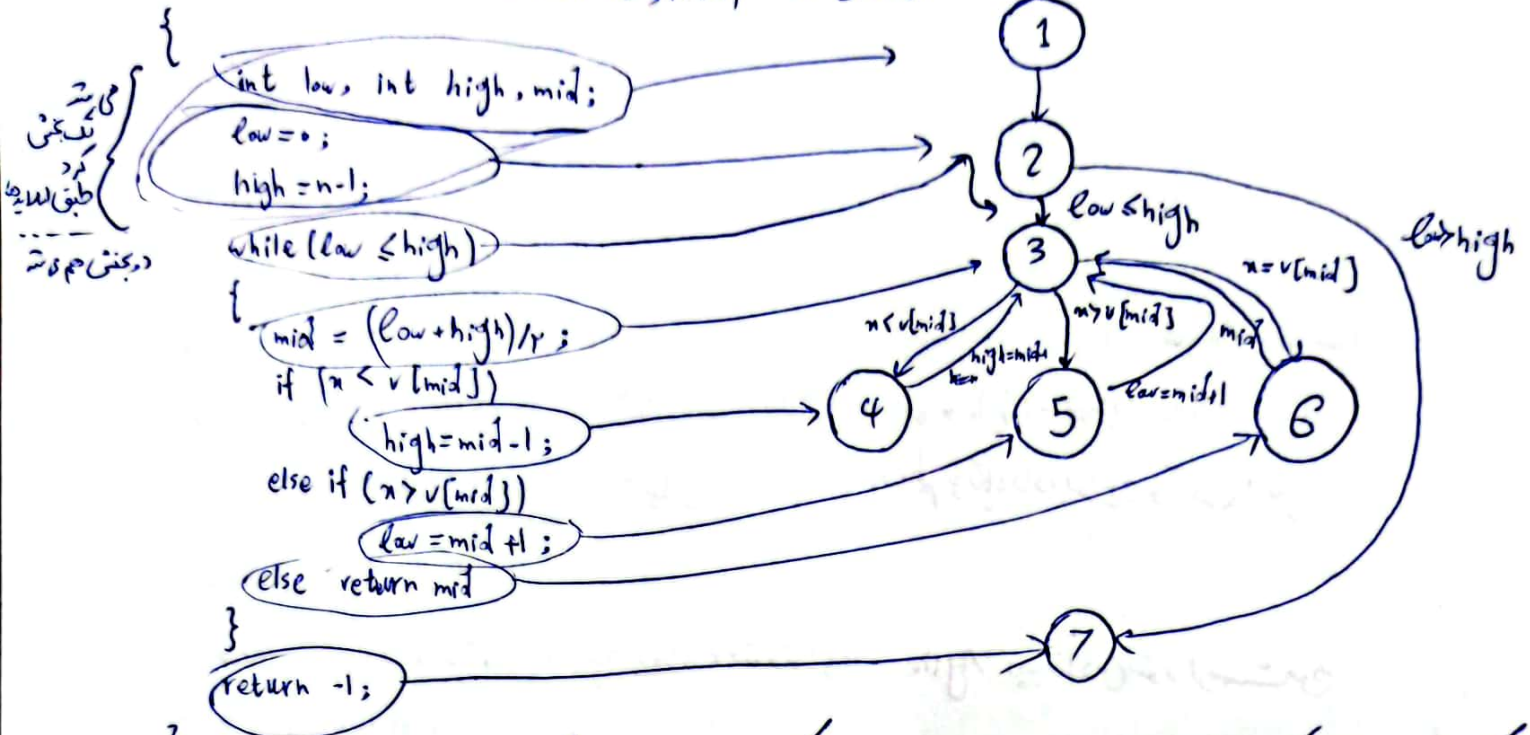
اجرای بازبینی های منظم و ارزیابی های کیفی در تمام مراحل توسعه نرم افزار



از ۱ شروع کردم به جای ۰، فرقی نمی‌کند و قراردادی است

الف -

int bin search (int x, int v[], int n)



نکته این است که چنین شکل می‌توان CFG را کشید و من لا راسی کشیم برای نشان دادن بهتر روند اجرای آن

Edge Coverage ← هدف و یعنی تمام یال‌های CFG است

Node Coverage ← هدف، پوشش دادن تمام رأس‌های CFG است

Testcases:

۱) منفرد آرایه وجود دارد

تابع ۳ دارد آرایه باز می‌گرداند ⇒ $v[3] = [1, 2, 3, 4, 5]$, $n = 5$, $x = 3$
در واقع انتقال را این است

۲) منفرد آرایه وجود ندارد

انتظار می‌رود تابع ۱ را برگرداند ⇒ $v[3] = [1, 2, 3, 4, 5]$, $n = 5$, $x = 6$

Test Paths:

۱) [1, 2, 3, 4, 6]

۲) [1, 2, 3, 4, 3, ..., 2, 3] → انتظاری می‌بود تا شرط حلقه اشتباه شود و سپس از حلقه بیرون می‌آید

چون عدد در آرایه وجود ندارد

در این مراحل تلاش کردم باطریک‌های گانگی، همه یال‌ها و رأس‌ها را گاور کنیم که این لا است با هم این‌ها را گاور می‌کنند



توضیح سارجوی این جواب در صفحه بعد

الف - نیازمندی های نرم افزار

- ۱) قابلیت جستجوی پیشرفته - امکان جستجوی دقیق و چند وجهی برای یافتن قوانین، رأی ها و کتب های قانون
- ۲) دسترسی به پایگاه داده قوانین - قابلیت دسترسی به پایگاه داده ای جامع از قوانین کشور و بین المللی
- ۳) دسترسی به تاریخچه پرونده های قضایی - قابلیت جستجو و دسترسی به رأی های صادر شده در پرونده های قضایی
- ۴) اشتراک گذاری - طراحی رابط کاربری آسان برای استفاده، با امکانات جستجوی ساده و پیشرفته
- ۵) قابلیت هوش مصنوعی - استفاده از هوش مصنوعی برای ارائه پیشنهادات بر اساس جستجوهای قبلی و ارتباط
- ۶) امنیت داده ها - تضمین امنیت و حریم خصوصی کاربران و داده های جستجو شده
- ۷) پشتیبانی از زبان های مختلف - قابلیت جستجو و نمایش به چندین زبان
- ۸) به روز رسانی مداوم داده ها - اطمینان از به روز رسانی منظم پایگاه داده قوانین و آرای قضایی

ب - متدولوژی پیشنهادی

با توجه به پیچیدگی و نوآوری در این پروژه، متدولوژی Agile پیشنهاد می شود از سمت من
دلایل این پیشنهاد:

تغییر پذیری و انعطاف پذیری: Agile اجازه می دهد تا تیم به سرعت به تغییرات نیازمندی ها و بازار واکنش نشان دهد.
تمرکز بر همکاری تیمی و مشتری: این متدولوژی تاکید زیادی بر همکاری نزدیک بین اعضای تیم و مشتریان دارد.
رویکرد تکراری و تدریجی: Agile به تیم اجازه می دهد تا نسخه های اولیه محصول را سریع توسعه دهد و به تدریج آن را بهبود بخشد.

ج - نیازمندی های غیر عملکردی

امنیت: حفاظت از داده ها، اطلاعات کاربران
پایداری و قابلیت اطمینان: عملکرد مطمئن حتی در شرایط بارگذاری بالا
قابلیت استفاده: رابط کاربری ساده و قابل فهم برای تمامی کاربران
توسعه پذیری: قابلیت افزودن ویژگی های جدید و به روز رسانی به سیستم

د - سطوح آزمون مورد نیاز

آزمون واحد (Unit Testing): برای اطمینان از صحت کارکرد هر بخش جداگانه که
آزمون ادغام (Integration Testing): برای تست تعامل بین ماژول های مختلف سیستم
آزمون سیستم (System Testing): برای اطمینان از عملکرد کلی سیستم

آزمون پذیرش (Acceptance Testing): برای اطمینان از مطابقت سیستم با نیازمندی ها و انتظارات

این سطوح آزمون به تیم اطمینان می دهد که محصول نهایی مطابق با نیازمندی های تعریف شده و با کیفیت بالا عرضه می شود.



برای تعریف نیازمندی‌ها، ابتدا ساریوی ذهنی از software مد نظر خودمون ایجاد و ترفیق می‌کنیم

ساریو:

یک دستور جستجو که برای علم به قوانین و محدوده‌های فضای و رأی‌های آن‌ها تشکیل می‌دهیم که حافظه جستجوی آن یک نتود باهدف استفاده از هوش مصنوعی برای بهتر جواب دادن به هر کاربرد باتوجه به سچ‌های پیشین و فعالیت‌های قدیمی اون شخص

حالا بی‌همه‌ی نیازمندی‌ها باتوجه به این ساریو طراحی شده است که ویژگی‌های خاصی

و جدیدی دارد که خیلی از software های مشابه از جمله برای فایل B نیز دارا است



الف - پیشنهاد معماری میکروسرویس برای پروژه اسارتاب

بله، پیشنهاد می‌دهم از معماری *Micro service* استفاده شود چون:

توسعه و نگهداری مستقل: میکروسرویس‌ها اجازه می‌دهند هر بخشی از سیستم به صورت مستقل توسعه یابد و نگهداری شود که برای تک اسارتاب در حال رشد، ایده آل است

قابلیت مقیاس‌پذیری: با توجه به پیچیدگی‌های ممکن در جستجوی قوانین و داده‌های قضایی، میکروسرویس‌ها به راحتی قابل مقیاس‌پذیری و افزایش ظرفیت پاسخگویی هستند

متمرکز روی تخصص گزینی: هر میکروسرویس می‌تواند بر یک جنبه خاص از سیستم متمرکز شود مانند جستجوی قوانین، تحلیل داده‌های قضایی و یا هوش مصنوعی

ب- الگوی *Decomposition* میکروسرویس‌ها

این الگوها باید طوری باشند که هر میکروسرویس یک وظیفه مشخص را انجام دهد و بتواند به طور مستقل مدیریت و توسعه یابد.

Decomposition → *self-contained service*

میکروسرویس جستجوی قانونی: مسئول جستجو در میان متون قانونی و ارائه نتایج مرتبط

میکروسرویس تحلیل پرونده‌های قضایی: تحلیل داده‌ها و رأی‌های قضایی و ارائه خلاصه یا اطلاعات مرتبط با جستجوی کاربر

میکروسرویس هوش مصنوعی و پیشنهاد: استفاده از الگوریتم‌های یادگیری ماشین برای ارائه پیشنهادات و بهبود نتایج جستجو بر اساس رفتار کاربر

ج- کاربرد *QRS* در این سیستم

بله می‌تواند کاربرد داشته باشد *Command Query Responsibility Segregation*

مثال: جداسازی عملیات نوشتن (*Command*) و خواندن (*Query*) در حالی که عملیات نوشتن می‌تواند شامل بروزرسانی پایگاه داده قوانین و آرای قضایی باشد و عملیات خواندن می‌تواند شامل جستجو و بازیابی اطلاعات از پایگاه داده برای مقایسه به کاربران باشد. این تفکیک بهینه‌سازی و کارایی بیشتر سیستم کمک می‌کند.

د- روش *Messaging* در سیستم

برای این سیستم، استفاده از *Kafka* یا *Rabbit MQ* به عنوان سیستم پیام‌رسانی پیشنهاد می‌شود چون:

کارایی بالا در پردازش پیام: هر دو سیستم قادر به کنترل و مدیریت حجم بالایی از پیام‌ها هستند

قابلیت اطمینان و دوام: این سیستم‌ها قابلیت تحمل خطا و اطمینان بالایی دارند

تضمین ترتیب پیام‌ها: به ویژه *Kafka*، ترتیب پیام‌ها را حفظ می‌کند که برای پردازش‌های وابسته به ترتیب

پشتیبانی از معماری میکروسرویس نیز هر دو در دسترس هستند با این معماری اهمیت دارد