مهندسی نرم‌افزار

نیم‌سال اول ۰۴-۱۴۰۳

دانشکده‌ی مهندسی کامپیوتر

استاد: دکتر مهران ریواده

تمرین تئوری چهارم

پاسخ‌دهنده: معین آعلی - ۴۰۱۱۰۵۵۶۱

فهرست مسائل

مسئله ۱ . . . . . ۱
الف ۱
ب ۲
ج ۲
د ۲
مسئله ۲ . . . . . ۳
الف ۳
ب ۳
ج ۴
مسئله ۳ . . . . . ۵
الف ۵
ب ۵
ج ۶
مسئله ۴ . . . . . ۷
الف ۷
ب ۸
ج ۹
د ۱۰

پاسخ مسئله‌ی ۱.

الف

کیفیت نرم‌افزار به معنای میزان برآورده شدن نیازهای عملکردی و غیرعملکردی سیستم است. دستیابی به کیفیت بالا در مهندسی نرم‌افزار حیاتی است زیرا مستقیماً بر قابلیت اطمینان، نگهداشت‌پذیری، و رضایت کاربران تأثیر می‌گذارد. کیفیت پایین نرم‌افزار می‌تواند منجر به نقص‌های پرهزینه، کاهش بهره‌وری، و نارضایتی کاربران شود.

ابعاد کیفیت مطرح شده توسط گاروین عبارتند از:

- عملکرد(Performance) : میزان توانایی سیستم در انجام وظایف موردنظر. مثال: سامانه مدیریت بانکداری الکترونیکی که تراکنش‌ها را به‌سرعت پردازش می‌کند.
- ویژگی‌ها(Features) : قابلیت‌های اضافی که فراتر از نیازهای اولیه هستند. مثال: اپلیکیشن پیام‌رسانی که علاوه بر ارسال متن، امکان تماس ویدیویی نیز دارد.
- قابلیت اطمینان(Reliability) : میزان عملکرد پایدار و بدون خرابی سیستم در طول زمان. مثال: سیستم کنترلی هواپیما که باید همواره بدون نقص کار کند.
- مطابقت(Conformance) : میزان تطابق با استانداردها و مشخصات موردنظر. مثال: نرم‌افزار حسابداری که مطابق با استانداردهای مالیاتی کشور توسعه یافته است.

- دوام(Durability) : پایداری و طول عمر نرم‌افزار. مثال: سیستم عامل که به مدت طولانی قابل استفاده باشد.
- قابلیت استفاده(Usability) : میزان سهولت استفاده توسط کاربران. مثال: نرم‌افزار طراحی گرافیکی با رابط کاربری ساده و شهودی.
- قابلیت سرویس‌دهی(Serviceability) : سهولت نگهداری و اصلاح نرم‌افزار. مثال: سامانه مدیریت محتوا (CMS) که به‌روزرسانی‌های آسان دارد.

ب

متدولوژی‌های آزمون نرم‌افزار را می‌توان به دو دسته‌ی آزمون مرسوم و آزمون شی‌گرا تقسیم کرد.

- سطوح آزمون: در آزمون مرسوم، تست واحدها (Unit-Testing) به‌صورت تابعی انجام می‌شود، اما در آزمون شی‌گرا، تمرکز بر کلاس‌ها و اشیا است.
- مورد آزمون: در آزمون مرسوم، تست بر روی ماژول‌های منفرد انجام می‌شود، اما در آزمون شی‌گرا تعامل بین اشیا و پیام‌ها بررسی می‌شود.
- ابزارها: ابزارهای آزمون سنتی شامل JUnit و NUnit می‌باشند، در حالی که آزمون شی‌گرا از ابزارهایی مانند JMock و Mockito بهره می‌برد.
- مزایا: آزمون شی‌گرا مقیاس‌پذیری بیشتری دارد و قابلیت نگهداشت بالاتری را ارائه می‌دهد.

ج

راستی‌آزمایی (Verification) و صحت‌سنجی (Validation) دو مرحله‌ی مهم در آزمون نرم‌افزار هستند:

- راستی‌آزمایی: فرآیند بررسی این که آیا نرم‌افزار مطابق با مشخصات طراحی شده است یا خیر. مثال: بررسی مستندات و اجرای آزمون‌های واحد.
- صحت‌سنجی: بررسی اینکه آیا نرم‌افزار نیازهای کاربر را برآورده می‌کند یا خیر. مثال: انجام آزمون‌های پذیرش توسط کاربران.

هر دو پروسه برای تضمین کیفیت بالا ضروری هستند، زیرا راستی‌آزمایی از تطابق با طراحی و صحت‌سنجی از تطابق با نیازهای واقعی اطمینان حاصل می‌کند.

د

چندین اقدام می‌تواند تأثیر آزمون رگرسیون را بهبود ببخشد:

- استفاده از آزمون‌های خودکار برای کاهش هزینه و زمان آزمون.
- انتخاب مجموعه‌ی بهینه آزمون‌ها برای اجرای مجدد، به جای اجرای کل مجموعه.
- استفاده از آزمون‌های افزایشی به جای اجرای مجدد کل سیستم.
- ترکیب آزمون رگرسیون با تحلیل تأثیر تغییرات برای تمرکز بر قسمت‌های حساس نرم‌افزار.

پاسخ مسئله‌ی ۲.

الف

یکپارچگی در فرآیند تست نرم‌افزار به معنای بررسی نحوه تعامل و همکاری میان ماژول‌ها و اجزای مختلف یک سامانه نرم‌افزاری است. این تست اطمینان حاصل می‌کند که اجزا به درستی با یکدیگر ارتباط دارند و داده‌ها بین آن‌ها به‌صورت صحیح منتقل می‌شوند.

اهمیت تست‌های یکپارچگی برای سامانه‌های پیچیده به دلایل زیر است:

- تضمین سازگاری بین ماژول‌های مختلف.
- کشف خطاهای ناشی از تعامل اجزا که در تست‌های واحد قابل تشخیص نیستند.
- افزایش قابلیت اطمینان سامانه و کاهش مشکلات احتمالی در محیط عملیاتی.

تفاوت‌های کلیدی بین تست‌های واحد و تست‌های یکپارچگی:

- تست واحد بر روی یک ماژول یا تابع مجزا تمرکز دارد، در حالی که تست یکپارچگی نحوه تعامل چندین ماژول را بررسی می‌کند.
- تست واحد معمولاً توسط توسعه‌دهندگان انجام می‌شود، اما تست یکپارچگی بیشتر توسط تیم تست انجام می‌شود.
- در تست واحد، از شبیه‌سازی (Mock) برای بررسی عملکرد مستقل یک ماژول استفاده می‌شود، اما در تست یکپارچگی، سیستم در محیط واقعی‌تری تست می‌شود.

ب

دو استراتژی مهم تست یکپارچگی عبارتند از:

- ادغام بیگ‌بنگ: در این روش، تمام ماژول‌های سامانه به‌طور همزمان با یکدیگر ادغام و سپس تست می‌شوند. این روش برای سامانه‌های پیچیده ریسک بالایی دارد، زیرا خطاهای متعددی به‌طور همزمان پدیدار می‌شوند و یافتن منبع اصلی مشکل دشوار است.
- ادغام افزایشی: در این روش، ماژول‌ها به‌صورت تدریجی و مرحله‌به‌مرحله ادغام و تست می‌شوند. این روش شامل دو رویکرد اصلی است:
  - ادغام از بالا به پایین: ابتدا ماژول‌های سطح بالا تست می‌شوند و سپس ماژول‌های سطح پایین اضافه می‌شوند.
  - ادغام از پایین به بالا: ابتدا ماژول‌های سطح پایین ادغام و تست شده و سپس به سطح بالاتر گسترش می‌یابند.

برای تست سامانه بانک اطلاعاتی، استراتژی ادغام افزایشی مناسب‌تر است، زیرا:

- کاهش ریسک خطاهای پیچیده و امکان شناسایی زودهنگام مشکلات.
- امکان انجام تست‌های مرحله‌به‌مرحله برای اطمینان از عملکرد صحیح هر بخش.
- ساده‌تر بودن اشکال‌زدایی نسبت به روش بیگ‌بنگ.

ج

سناریوی اول:

- مرحله ۱: کاربر اطلاعات ورود را در سامانه ورود وارد می‌کند.
- مرحله ۲: سامانه ورود، اطلاعات کاربر را به سامانه مدیریت داده‌ها ارسال می‌کند.
- مرحله ۳: سامانه مدیریت داده‌ها بررسی می‌کند که آیا اطلاعات کاربر معتبر است یا خیر.
- مرحله ۴: در صورت تأیید، مجوز ورود داده شده و اطلاعات به کاربر نمایش داده می‌شود.

خطاهای محتمل:

- نامعتبر بودن اطلاعات ورود (ورود نادرست رمز عبور).
- اختلال در ارتباط میان سامانه ورود و سامانه مدیریت داده‌ها.
- ارسال اطلاعات نادرست از سامانه ورود به سامانه مدیریت داده‌ها.

سناریوی دوم:

- مرحله ۱: یک تراکنش مالی توسط کاربر انجام می‌شود.
- مرحله ۲: سامانه مدیریت تراکنش‌ها جزئیات تراکنش را پردازش و ذخیره می‌کند.
- مرحله ۳: سامانه مدیریت تراکنش‌ها داده‌های مرتبط را به سامانه گزارش‌دهی ارسال می‌کند.
- مرحله ۴: سامانه گزارش‌دهی داده‌ها را دریافت کرده و یک گزارش مالی تولید می‌کند.

خطاهای محتمل:

- تأخیر یا عدم ارسال اطلاعات از سامانه مدیریت تراکنش‌ها به سامانه گزارش‌دهی.
- نمایش نادرست اطلاعات تراکنش در گزارش‌ها.
- ناهماهنگی در پردازش داده‌ها و ایجاد گزارش‌های ناقص.

پاسخ مسئله‌ی ۳.

الف

عناصر اصلی تضمین کیفیت نرم‌افزار شامل موارد زیر است:

- برنامه‌ریزی کیفیت: تعیین استانداردها و روش‌های ارزیابی کیفیت در مراحل مختلف توسعه.
- تضمین کیفیت: فرآیندها و فعالیت‌هایی که اطمینان حاصل می‌کنند نرم‌افزار مطابق با استانداردها و الزامات تعیین‌شده است.
- کنترل کیفیت: شامل فعالیت‌های تست و بررسی که برای شناسایی و رفع خطاهای نرم‌افزار انجام می‌شود.
- بهبود مستمر کیفیت: تجزیه و تحلیل عملکرد گذشته و اعمال تغییرات برای افزایش کیفیت.

تضمین کیفیت به عنوان یک فعالیت چتری در نظر گرفته می‌شود زیرا تمامی مراحل چرخه عمر نرم‌افزار را در بر می‌گیرد، از جمله:

- مرحله نیازسنجی: بررسی نیازها برای جلوگیری از ابهامات و تناقضات.
- طراحی: ارزیابی معماری و طراحی برای بهینه‌سازی ساختار نرم‌افزار.
- پیاده‌سازی: بررسی کدها و انجام آزمون‌های خودکار برای شناسایی خطاها.
- آزمون و اعتبارسنجی: انجام تست‌های واحد، یکپارچگی و پذیرش برای اطمینان از صحت عملکرد.
- استقرار و نگهداری: ارزیابی عملکرد سیستم و مدیریت تغییرات.

ب

وظایف اصلی تیم تضمین کیفیت شامل موارد زیر است:

- تعریف استانداردها و فرآیندهای کیفیت.
- طراحی و اجرای تست‌های مختلف نرم‌افزار.
- تحلیل نتایج آزمون‌ها و ارائه گزارش‌های کیفی.
- بررسی و مدیریت مشکلات و پیشنهاد راه‌حل‌های بهبود.
- اجرای ممیزی‌های کیفی و بهبود فرآیندهای توسعه.

مدیریت تغییرات نقش مهمی در تضمین کیفیت دارد زیرا هر تغییری در نرم‌افزار ممکن است بر کیفیت آن تأثیر بگذارد. این فرآیند شامل موارد زیر است:

- بررسی و ارزیابی تأثیر تغییرات بر عملکرد کلی نرم‌افزار.
- اجرای آزمون‌های رگرسیون برای اطمینان از عدم ایجاد مشکلات جدید.
- مستندسازی تغییرات برای شفافیت و قابل‌ردیابی بودن فرآیند.
- مدیریت بازخورد کاربران و اصلاح نواقص احتمالی.

تأثیر مدیریت تغییرات بر کیفیت نرم‌افزار شامل افزایش پایداری، کاهش خطاها و بهبود تجربه کاربری است.

ج

مدیریت خطاها و تجزیه و تحلیل آن‌ها نقش حیاتی در بهبود کیفیت نرم‌افزار دارد. فرآیند مدیریت خطا شامل:

- شناسایی و ثبت خطاها در طول فرآیند تست.
- تحلیل ریشه‌ای مشکلات برای درک دلایل اصلی آن‌ها.
- اولویت‌بندی و اصلاح خطاها برای بهینه‌سازی عملکرد.
- نظارت بر روند اصلاحات و ارزیابی تأثیر آن‌ها.

فعالیت‌های تضمین کیفیت به کاهش ریسک‌های ایجاد نرم‌افزار کمک می‌کنند، از جمله:

- انجام تست‌های مستمر برای جلوگیری از بروز مشکلات در محیط عملیاتی.
- استفاده از رویکردهای مهندسی کیفیت برای جلوگیری از ایجاد خطاها.
- مستندسازی فرآیندها برای افزایش قابلیت نگهداری و بهبود مداوم.
- تحلیل بازخورد کاربران برای شناسایی نقاط ضعف و ارتقای نرم‌افزار.

مثال: در یک سامانه مدیریت بانک اطلاعاتی، تیم تضمین کیفیت پس از شناسایی خطای تأخیر در پردازش گزارش‌ها، یک تحلیل ریشه‌ای انجام داده و متوجه شده که پایگاه داده بهینه‌سازی نشده است. با اجرای تغییرات لازم در شاخص‌های پایگاه داده، زمان پردازش کاهش یافته و عملکرد سامانه بهبود یافته است.

پاسخ مسئله‌ی ۴.

الف

در دنیای نرم‌افزار و فناوری، کیفیت یکی از مهم‌ترین عوامل موفقیت شرکت‌ها محسوب می‌شود. بسیاری از شرکت‌های مطرح، باوجود داشتن بازار گسترده و نوآوری، به دلیل نادیده گرفتن برخی از فاکتورهای کلیدی کیفیت، متحمل شکست‌های سنگینی شده‌اند. در این مقاله به بررسی سه شرکت مشهور که به علت ضعف در طراحی و کیفیت نرم‌افزاری شکست خوردند، می‌پردازیم.

۱. نوکیا(Nokia)

حوزه فعالیت: تولید گوشی‌های موبایل و تجهیزات مخابراتی
محصولات: تلفن‌های همراه کلاسیک و هوشمند
ایراد: عدم توجه به تجربه کاربری و به‌روزرسانی سیستم‌عامل
نوکیا به عنوان یکی از غول‌های صنعت موبایل، در سال‌های اولیه‌ی دهه ۲۰۰۰ سلطه‌ی کاملی بر بازار داشت. اما عدم تمرکز کافی بر سیستم‌عامل‌های هوشمند و تجربه‌ی کاربری مناسب باعث شد تا در رقابت با اندروید و iOS شکست بخورد.

۲. بلک‌بری(BlackBerry)

حوزه فعالیت: تولید تلفن‌های هوشمند و تجهیزات امنیتی
محصولات: گوشی‌های مجهز به صفحه‌کلید فیزیکی و نرم‌افزارهای امنیتی
ایراد: نادیده گرفتن طراحی کاربرپسند و محدودیت اکوسیستم نرم‌افزاری
بلک‌بری به دلیل تمرکز بیش‌ازحد بر امنیت و عدم توجه کافی به اکوسیستم نرم‌افزارهای کاربردی و طراحی بصری جذاب، سهم بازار خود را از دست داد.

۳. یاهو(Yahoo)

حوزه فعالیت: ارائه خدمات اینترنتی، موتور جستجو، ایمیل و تبلیغات آنلاین
محصولات: موتور جستجوی یاهو، یاهو میل، یاهو مسنجر
ایراد: عدم نوآوری در طراحی سرویس‌های دیجیتال و تجربه کاربری ضعیف
یاهو نتوانست خود را با تغییرات سریع در بازار تطبیق دهد و طراحی نامناسب محصولات باعث شد که کاربران به سمت رقبایی مانند گوگل و جیمیل مهاجرت کنند.

بر اساس مدل مک‌مال، این شرکت‌ها سه فاکتور مهم کیفیت را نادیده گرفتند:

- تجربه کاربری(Usability): طراحی ضعیف رابط کاربری و پیچیدگی در استفاده باعث کاهش رضایت کاربران شد.
- نگهداشت‌پذیری(Maintainability): سیستم‌های نرم‌افزاری ناکارآمد که به سختی قابلیت بروزرسانی داشتند، منجر به از دست رفتن مزیت رقابتی شد.
- انعطاف‌پذیری(Flexibility): عدم تطابق با تغییرات بازار و روندهای نوظهور فناوری، موجب شکست در رقابت شد.

این شرکت‌ها هزینه‌های مختلفی را متحمل شدند که باعث شد نتوانند شکست خود را جبران کنند:

- هزینه‌ی از دست دادن بازار: نوکیا و بلک‌بری میلیاردها دلار از سهم بازار خود را از دست دادند، زیرا کاربران به گزینه‌های بهتری روی آوردند.
- هزینه‌ی از دست دادن اعتماد مشتریان: هنگامی که کاربران متوجه شدند این برندها نمی‌توانند نیازهای آن‌ها را برآورده کنند، دیگر به آن‌ها بازنگشتند.
- هزینه‌ی توسعه‌ی ناکارآمد: تلاش برای نجات کسب‌وکار با سرمایه‌گذاری در فناوری‌های جدید، اما بدون درک عمیق از نیاز بازار، به هدر رفتن منابع منجر شد.

ب

در دنیای نرم‌افزار و فناوری، کیفیت یکی از مهم‌ترین عوامل موفقیت شرکت‌ها محسوب می‌شود. بسیاری از شرکت‌های مطرح، باوجود داشتن بازار گسترده و نوآوری، به دلیل نادیده گرفتن برخی از فاکتورهای کلیدی کیفیت، متحمل شکست‌های سنگینی شده‌اند. در این مقاله به بررسی سه شرکت مشهور که به علت ضعف در طراحی و کیفیت نرم‌افزاری شکست خوردند، می‌پردازیم. همچنین به بررسی شرکت‌هایی می‌پردازیم که با وجود نادیده گرفتن برخی فاکتورهای کیفیت، همچنان در بازار موفق هستند.

:نمونه‌هایی از شرکت‌های موفق باوجود ضعف در کیفیت

۱. مایکروسافت (Microsoft)

حوزه فعالیت: توسعه نرم‌افزار و سیستم‌عامل
محصولات: ویندوز، آفیس، آژور
ایراد: وجود مشکلات امنیتی و تجربه کاربری نه‌چندان ایده‌آل در برخی نسخه‌ها
مایکروسافت با وجود عرضه نسخه‌هایی از ویندوز که از نظر عملکرد و امنیت دچار مشکل بودند (مانند ویندوز ویستا)، همچنان توانست با عرضه‌ی نسخه‌های بهبود یافته جایگاه خود را حفظ کند.

۲. فیسبوک (Meta)

حوزه فعالیت: شبکه‌های اجتماعی و فناوری
محصولات: فیسبوک، اینستاگرام، واتساپ
ایراد: نگرانی‌های مربوط به حریم خصوصی و داده‌های کاربران
با وجود رسوایی‌های مربوط به حریم خصوصی، فیسبوک توانست با ارائه ویژگی‌های جدید و خرید پلتفرم‌های محبوبی مانند اینستاگرام، همچنان جزو شرکت‌های موفق باقی بماند.

۳. تسلا (Tesla)

حوزه فعالیت: خودروسازی الکتریکی و انرژی پاک
محصولات: خودروهای الکتریکی، پنل‌های خورشیدی
ایراد: مشکلات مربوط به کنترل کیفیت در تولید خودرو
با وجود گزارش‌های متعدد از مشکلات تولیدی و کیفیتی در خودروهای تسلا، این شرکت توانسته است با نوآوری‌های خود و محبوبیت برند، به فعالیت خود ادامه دهد.

بر اساس مدل مک‌مال، این شرکت‌ها سه فاکتور مهم کیفیت را نادیده گرفتند:

- امنیت(Security): مایکروسافت و فیسبوک در برخی محصولات خود با چالش‌های امنیتی و حفاظت از داده‌های کاربران روبرو بوده‌اند.

- قابلیت اطمینان(Reliability): تسلا به دلیل مشکلات کنترل کیفیت، در برخی مواقع با کاهش اعتماد کاربران مواجه شده است.
- پایداری(Stability): برخی نسخه‌های نرم‌افزاری، مانند ویندوز ویستا، مشکلات زیادی از نظر عملکردی و پایداری داشتند.

این شرکت‌ها هزینه‌های مختلفی را متحمل شدند اما توانستند آن‌ها را جبران کنند:

- هزینه‌ی مالی: فیسبوک و مایکروسافت جریمه‌های سنگینی بابت نقض حریم خصوصی و ضعف‌های امنیتی پرداخت کردند.
- هزینه‌ی از دست دادن اعتماد کاربران: تسلا با وجود مشکلات کیفیتی، توانسته با نوآوری و تبلیغات قوی، برند خود را حفظ کند.
- هزینه‌ی توسعه‌ی مجدد: مایکروسافت با عرضه نسخه‌های بهبودیافته، مانند ویندوز ۱۰، توانست نواقص محصولات قبلی را برطرف کند.

شکست برخی شرکت‌ها و موفقیت برخی دیگر با وجود ضعف‌های کیفیتی نشان می‌دهد که عواملی مانند نوآوری، بازاریابی و پاسخ سریع به نیازهای کاربران می‌تواند بر ضعف‌های فنی غلبه کند. شرکت‌هایی که به موقع ضعف‌های خود را اصلاح کنند، همچنان می‌توانند جایگاه خود را حفظ کنند.

ج

در دنیای نرم‌افزار و فناوری، کیفیت یکی از مهم‌ترین عوامل موفقیت شرکت‌ها محسوب می‌شود. بسیاری از شرکت‌های مطرح، باوجود داشتن بازار گسترده و نوآوری، به دلیل نادیده گرفتن برخی از فاکتورهای کلیدی کیفیت، متحمل شکست‌های سنگینی شده‌اند. در این مقاله به بررسی سه شرکت مشهور که به علت ضعف در طراحی و کیفیت نرم‌افزاری شکست خوردند، می‌پردازیم. همچنین به بررسی شرکت‌هایی می‌پردازیم که با وجود نادیده گرفتن برخی فاکتورهای کیفیت، همچنان در بازار موفق هستند.

معاوضات انجام‌شده توسط این شرکت‌ها

شرکت‌های موفقی که برخی فاکتورهای کیفیت را نادیده گرفته‌اند، این کار را در ازای مزایای دیگری انجام داده‌اند:

- مایکروسافت: کاهش هزینه‌های توسعه و تسریع در عرضه نسخه‌های جدید نرم‌افزار.
- فیسبوک: حفظ مدل تبلیغاتی سودآور به بهای نگرانی‌های مربوط به حریم خصوصی.
- تسلا: افزایش سرعت نوآوری در فناوری‌های خودرویی به بهای کنترل کیفیت نامناسب.

دلایل ارائه‌ی محصولات پر ایراد

سه دلیل اصلی که شرکت‌ها معمولاً محصولات پر ایراد ارائه می‌دهند، شامل موارد زیر است:

- فشار زمانی برای ورود به بازار: رقابت شدید باعث می‌شود شرکت‌ها محصولاتی ناپایدار را برای زودتر در دسترس قرار دادن، عرضه کنند.
- کاهش هزینه‌های توسعه: بعضی شرکت‌ها برای بهینه‌سازی هزینه‌ها، تست‌های کیفی کافی را انجام نمی‌دهند.
- تمرکز بر ویژگی‌های جدید به جای بهبود کیفیت: بسیاری از شرکت‌ها برای جذب مشتریان جدید، روی نوآوری تمرکز کرده و کیفیت را به عنوان اولویت دوم در نظر می‌گیرند.

شکست برخی شرکت‌ها و موفقیت برخی دیگر با وجود ضعف‌های کیفیتی نشان می‌دهد که عواملی مانند نوآوری، بازاریابی و پاسخ سریع به نیازهای کاربران می‌تواند بر ضعف‌های فنی غلبه کند. شرکت‌هایی که به موقع ضعف‌های خود را اصلاح کنند، همچنان می‌توانند جایگاه خود را حفظ کنند.

د

درستی (Correctness)

مثال: اطمینان از نمایش صحیح اطلاعات محصولات و خدمات.

راهکار پیاده‌سازی: استفاده از اعتبارسنجی داده‌ها در فرم‌های ورودی و بررسی صحت اطلاعات بارگذاری‌شده از طریق فیلترها و قوانین اعتبارسنجی.

قابلیت اطمینان (Reliability)

مثال: سرورهای پایدار و جلوگیری از قطعی‌های ناگهانی.

راهکار پیاده‌سازی: بهره‌گیری از معماری متمرکز بر تحمل خطا (Fault-Tolerance) و پایگاه داده‌های توزیع‌شده برای افزایش قابلیت اطمینان.

کارایی (Efficiency)

مثال: بارگذاری سریع صفحات و نمایش بهینه‌ی تصاویر محصولات.

راهکار پیاده‌سازی: بهینه‌سازی درخواست‌های پایگاه داده، استفاده از فشرده‌سازی تصاویر، و بهره‌گیری از سامانه‌ی کشینگ.

قابلیت استفاده (Usability)

مثال: طراحی رابط کاربری ساده و قابل فهم برای کاربران مبتدی.

راهکار پیاده‌سازی: استفاده از اصول طراحی رابط کاربری کاربرمحور (User-Centered-Design) و انجام تست‌های تجربه‌ی کاربری.

نگهداری‌پذیری (Maintainability)

مثال: امکان بروزرسانی آسان نرم‌افزار و افزودن قابلیت‌های جدید.

راهکار پیاده‌سازی: پیاده‌سازی معماری ماژولار، مستندسازی کامل کدها، و استفاده از اصول برنامه‌نویسی شیءگرا.

قابلیت انتقال (Portability)

مثال: اجرای نرم‌افزار بر روی سیستم‌عامل‌های مختلف مانند اندروید و iOS .

راهکار پیاده‌سازی: استفاده از فریمورک‌های چندسکویی مانند Flutter یا React-Native برای توسعه‌ی هم‌زمان بر روی پلتفرم‌های مختلف.

در صورتی که مجبور به ارائه‌ی نرم‌افزاری با حداقل امکانات باشیم، باید فاکتورهایی را انتخاب کنیم که بیشترین تأثیر را در عملکرد و رضایت کاربران داشته باشند. این فاکتورها عبارت‌اند از:

- درستی(Correctness): اگر اطلاعات نادرست نمایش داده شوند، کاربران اعتماد خود را از دست خواهند داد.
- قابلیت اطمینان(Reliability): اگر نرم‌افزار پایدار نباشد و دچار قطعی شود، کاربران آن را ترک خواهند کرد.
- قابلیت استفاده(Usability): تجربه‌ی کاربری خوب باعث جذب کاربران و افزایش تعامل آن‌ها می‌شود.
- کارایی(Efficiency): سرعت و عملکرد بهینه برای کاربران حیاتی است، زیرا نرم‌افزار کند تجربه‌ی کاربری نامطلوبی را ایجاد می‌کند.