۱.در مورد اصطلاحات و مفاهیم زیر توضیحات کافی ارائه دهید:

الف) Definition of Ready (DOR): آیا کار انجام‌شده با توضیحاتی که در User Story آمده مطابقت می‌کند یا خیر.

ب) Architectural Tactic: اجزای بنیادی معماری یک نرم‌افزار که یک معمار سیستم پیاده‌سازی می‌کند تا Quality Requirementها را برآورده کند.

ج) Modularity, Coupling and Cohesion:

Modularity: میزان شکسته‌شدن یک برنامه به اجزای کوچک‌تر

Coupling: میزان وابستگی بین اجزای تشکیل‌دهنده نرم‌افزار

Cohesion: چقدر اجزای هر بخش (از نظر کاربردی) به یکدیگر تعلق دارند

د) Statistical SQA: تمرکزمان را روی چیزهایی که واقعاً اهمیت دارند بگذاریم؛ برای این منظور از تکنیک‌ها و ابزارهایی استفاده کنیم تا به ما نشان بدهند (به عنوان مثال درصدگیری شدت اهمیت) که چه چیزهایی اهمیت دارند

ه) Technical Debt: اگر بخشی از کارهای انجام‌شده کیفیت لازمه را نداشته‌باشند، هنگام integration باید دوباره انجام‌شوند که نوعی بدهی به‌حساب می‌آیند که با باید پرداخت‌شود.

۲. به نظر شما کدام فعالیت‌ها یا رویه‌ها به برنامه‌ریزی موفق و تحقق‌پذیر در اسکرام کمک می‌کند؟ توضیح دهید.

۱- **تعریف دقیق Sprint**: اگر هر Sprint به صورت واضح تعریف‌شود (هدف و چگونگی رسیدن به آن). برای این منظور، هدف باید مشخص و قابل اندازه‌گیری باشد.

۲- **اولویت‌بندی Taskها**: Taskهای با اولویت بالاتر باید زودتر انجام شوند

۳- **برنامه‌ریزی متناسب با توانایی گروه:** میزان فعالیت‌های محول‌شده باید درخور تیم باشد؛ اگر کمتر یا بیشتر باشد، نتیجه‌ی مطلوبی در پی نخواهدداشت.

۳. چه روش ها و تکنیک هایی برای شناسایی و استخراج نیازمندی های سیستم نرم افزاری پروژه گروهتان، توصیه می کنید (با ذکر دلیل)؟ آنها را مختصرا معرفی کنید.

Scenario-driven: برای هر سناریو یک User Story تعریف کرده و در هرکدام از آن‌ها سناریو را توضیح می‌دهیم؛ به این صورت که سناریو توصیف و خروجی موفق آن نیز تشریح شود. می‌توان برای هر سناریو یک تست نیز در نظر گرفت.

این روش برای سیستم درنظر گرفته پروژه ما مناسب‌است چرا که برای ما کلیت سیستم و رفتار آن اهمیت بیشتری دارد.

Prototyping: به منظور مطابقت‌دادن پروژه با نیازهای مشتری، می‌توان Prototype تولیدکرد. چون رفتار کلی سیستم مد نظر ماست، این روش نیز می‌تواند مفید باشد.

۴.معماری نرم افزار

الف)استفاده از سبک معماری لایه ای چه تاثیری بر قابلیت تغییر دارد؟ توضیح دهید.

معماری لایه‌ای معمولاً انعطاف‌پذیری کمی دارد؛ چرا که هرگونه تغییر منجر به تغییر رابطه‌ی بین لایه‌ها می‌شود و هر چه لایه‌ها درونی(مرکزی)تر می‌شوند، اعمال تغییر روی آن‌ها سخت‌تر خواهد بود.

ب)مزایا و معایب معماری یکپارچه چیست؟

معماری Monolithic یا یکپارچه از نظر ساخت، تست و پیاده‌سازی راحت هستند و معمولاً سریع‌تر از دیگر معماری‌ها هستند؛ اما اگر یک قسمت از آن‌ها دچار اشکال شود سیستم از کار می‌افتد(Single point of Failure). پس خیلی قابل اطمینان نیستند. هم‌چنین، به‌دلیل Coupling بالا، در برابر تغییرات خیلی انعطاف‌پذیر نیستند.

۵. اجرای دو الگوی طراحی Abstract Factoryو Builder بر روی مفاهیم زیر چگونه تاثیر می گذارد؟ دلایل خود را در مورد نحوه و میزان تاثیرگذاری به صورت کامل ذکر کنید.

- **Separation of Concern**
- **Information Hiding**
- **Cohesion**
- **Coupling**

Abstract Factory:

- Separation of Concern:
- Information Hiding:
- Cohesion:
- Coupling:

۶. جدول ۱تعداد برنامه نویسان چهار پروژه و جدول ۲ تعداد خط کد و باگ کشف شده تا پایان هر هفته از یک بازه کد نویسی را نشان می‌دهد .به طور مثال تا پایان هفته سوم بازه مورد نظر، پروژه دلتا شامل ۳۵۱۵خط کد تولید و ۶۰باگ کشف شده است.

الف)در چه صورت از داده های جدول ۱و ،۲برای مقایسه معنادار کارایی برنامه نویسان و نرخ باگ می‌توانیم استفاده کنیم؟

در صورت یکسان‌بودن مشخصه‌های دیگر مانند نوع پروژه، میزان Reliability لازم و ...

ب)با فرض مشابه بودن میزان پیچیدگی پروژه ها، چگونه کارایی و نرخ باگ پروژه ها را مقایسه می‌کنید؟ توضیح دهید.

با در نظر گرفتن یک معیار مشترک می‌توان کارایی و نرخ باگ پروژه‌ها را مقایسه کرد.

معیار کارایی: $\frac{\sqrt{LOC}\times \#\text{برنامه‌نویس}}{bug} \times Week$، نرخ باگ = $\frac{LOC}{Bug}$

در معیار کارایی باید باگ‌ها در نظر گرفته‌شوند و هر چه تعداد خطوط کد کمتر باشد، پروژه سایز کمتری دارد.

محاسبات شما چه چیز را نشان می دهد؟ با توجه به نتایج به دست آمده، چه بهبودهایی را پیشنهاد می کنید؟

گروه الفا تعداد باگ زیاد و پروژه‌شان حجم زیادی دارد.

گروه دلتا حجم کم و باگ کمی نیز دارد.