علی نظری - ۹۶۳۱۰۷۵
گروه ۲
پروژه مدیریت پروژه‌ی برخط

سوال ۱)

الف) منظور از DOR، استاندارهایی است که باید به یوزر استوری‌ها اضافه شود تا توسط تیم توسعه به عنوان فیچرهایی برای پیاده سازی، مورد پذیرش قرار گیرد و بحث‌های اضافی در طول اسپرینت به خاطر شفاف نبودن ویژگی‌ها را کاهش دهد.

ب) منظور از Architectural Tactics یک سری اصول است که هر کدام از آن‌ها روی کیفیت یک ویژگی تمرکز دارند و کمک می‌کنند که یک ویژگی در انتها بتواند به بهترین شکل به انجام برسد.

ج)

منظور از modularity این است که نرم‌افزار را به بخش‌های کوچک‌تری تجزیه کنیم که بتوانند از طریق واسطهایی که تعریف می‌کنیم با هم در ارتباط داشته باشند و البته هر کدام هم بتوانند به صورت مستقل کارشان را انجام دهند.

منظور از coupling نیز معیاری است که از طریق آن میزان استقلال ماژول‌ها را می‌سنجیم و باید تلاش کنیم که ترجیحا این مقدار را کاهش دهیم.

منظور از cohesion نیز معیاری است که از طریق آن بیان می‌کنیم که همبستگی درونی یک ماژول چگونه است یعنی بهتر است این مقدار زیاد باشد که به معنی همبستگی و ارتباط درست اجزای یک ماژول برای انجام یک کار با یک‌دیگر است.

د) یک سری کار است که صورت آماری میریم سراغ کارهایی که میدونیم برای تضمین کیفیت محصولمان مهم تر هستند چون فرصت کم است و نمی‌توانیم سراغ تمامی موارد برویم.

ه) منظور از technical debt بدهی‌های فنی‌ای است که در راه توسعه‌ی محصول مثلا مجبور می‌شویم به خاطر کمبود وقت یا نداشتن راه حل مناسب، راهکار نامناسبی را استفاده کنیم. این موارد را به عنوان بدهی فنی به آینده موکول می‌کنیم تا به آن‌ها رسیدگی کنیم.

سوال ۲) من سه فعالیت sprint planning, daily scrum, sprint retro رو فکر می‌کنم که مهمتر هستند چون در ابتدا کمک می‌کنند که یک برنامه‌ریزی داشته باشیم و بعد هر روز این برنامه‌ریزی را بررسی کنیم و ببینیم که تیم برای ادامه‌ی این ارزیابی مشکلی دارد یا نه و در انتها نیز این برنامه ریزی را بررسی می‌کنیم که در برنامه‌ریزی بعدی به ما تجربه‌هایی اضافه می‌کند. همچنین نقش اسکرام مستر رو در پیشبرد و حراست از این برنامه‌ریزی خیلی مهم می‌دونم چون یک نفری هست که بر جلسات نظارت می‌کنه و کمک می‌کنه که افراد در جلسات نقش خودشون رو درستی ایفا کنن که هدف استفاده از اسکرام حفظ بشه.

سوال ۳) ما می‌خواستیم یک نرم‌افزار مدیریت پروژه بزنیم.
مصاحبه: از اون جایی که ما میدیم که بعضا مدیرای تیم‌ها و محصول‌ها با نرم‌افزارهای موجود مشکلاتی داشتند پس این روش رو برای بدست آوردن نیازمندی‌های خودمون مهم دیدیم که با اون‌ها صحبت کنیم و ببینیم چه نیازمندی‌هایی از صحبت با اون‌ها بدست میاریم.
دیدن موارد مشابه: این مورد هم خیلی به ما میتونه کمک کنه چون نرم‌افزارهای مشابه در این حوزه زیاده و اگر ما بریم و اون‌ها رو بررسی کنیم قطعا خیلی می‌تونه به ما کمک کنه تا بتونیم نیازمندی‌های سیستم خودمون رو پیدا کنیم.

سوال ۴)

الف) به خاطر مستقل بودن لایه‌ها ما می‌توانیم هر کدام از این لایه‌ها را جداگانه تغییر دهیم بدون اینکه کل سیستم دچار مشکل و دستکاری شود که باعث می‌شود ما بتوانیم ساده‌تر به تغییرات نگاه کنیم و با رفتن به لایه مورد نظر تغییرمان را انجام دهیم.

ب) در شروع می‌توان گفت که معماری یکپارچه قابل فهم‌تر است و سریع‌تر می‌توان پیاده‌سازی آن را آغاز کرد البته از آن طرف ایراداتی هم دارد که در اسکیل کردن پروژه کار ما سخت‌تر است همچنین در دیپلوی کردن و نسخه دادن هم ما باید همه چیز را با هم بالا بیاوریم که کار سخت تری نسبت به مایکروسرویس‌ها است که در آن‌ها هر چیز به صورت مستقل می‌تواند نسخه‌اش تغییر کند همچنین توی استفاده از استک‌های مختلف تکنولوژی نیز در مایکروسرویس کار ساده‌تری را داریم.

سوال ۵)

در مورد builder:

در مورد Separation of Concern تاثیر مثبتی می‌گذارد زیرا ساختن مدل‌های مختلف را از هم جدا می‌کند و در لحظه باعث می‌شود فقط به خود آن مدل فکر کنیم.

در مورد Information Hiding مثبت عمل می‌کند زیرا کلاس اصلی از چیزایی که بیلدر میسازه بی اطلاعه.

در مورد Cohesion خوب عمل می‌کند و آن را زیاد می‌کند چون هر کلاس کار مستقل به مدل خودش را انجام می‌دهد.

در مورد Coupling خوب کار نمیکند و آن را زیاد می‌کند زیرا کلاس‌های جدیدی برای هر مدل می‌سازد.

در مورد abstract factory:

در مورد Separation of Concern تاثیر مثبتی می‌گذارد زیرا ساختن مدل‌های مختلف را از هم جدا می‌کند و در لحظه باعث می‌شود فقط به خود آن مدل فکر کنیم.

در مورد Information Hiding تاثیر مثبت می‌گذارد زیرا هر کدام پیاده‌سازی مستقل به خودشان را جداگانه انجام می‌دهند.

در مورد Cohesion خوب عمل می‌کند و آن را زیاد می‌کند چون هر کس کار مستقل به مدل خودش را انجام می‌دهد.

در مورد Coupling به نظر خوب عمل میکنه چون ابسترکت‌ها رو جدا میکنه

سوال ۶)

قسمت الف) کلا مقایسه باید در شرایط یکسانی انجام شود و اگر این گونه نیست باید این مسئله را در نتیجه نظر بگیریم. مثلا باید پیچیدگی پروژه‌ها یکسان باشد تا بتوان این برنامه نویسان را با هم مقایسه کرد همچنین توانایی این برنامه نویسان نیز در تعداد باگی که تولید می‌کنند تاثیر می‌گذارد.

قسمت ب) باید دنبال یک متریک درست باشیم و بر اساس آن ببینیم چگونه می‌توانیم بهبود داشته باشیم.