

سوال (1)

(الف)

در چنین شرایطی، چندین رویکرد میتوانیم داشته باشیم که می توان به عنوان یک عضو انجام داد :

1. **ارتباط با همکار مربوطه:** اول باید در صورت ممکن با همکار خود تماس بگیرم تا درک بهتری از وضعیت او و تأثیر او بر پروژه پیدا کنم. اگر ممکن است، برنامه ریزی کنم که چگونه می توانم کارهای او را به صورت دورکاری یا با همکاری دیگر اعضای تیم به پیش ببرم.
2. **جلسه فوری با تیم اسکرام:** برگزاری یک جلسه فوری با تیم اسکرام برای بحث در مورد این چالش و یافتن راه حل های ممکن.
3. **بازتوزیع وظایف:** در نظر گرفتن امکان بازتوزیع وظایف میان اعضای دیگر تیم. البته با توجه به اینکه ممکن است برای این کار نیازمند بازبینی **priority** ها و توانایی های اعضای تیم باشیم.
4. **تنظیم دوباره اهداف اسپرینت:** بر اساس شرایط جدید، به علت محدودیت به وجود آمده ممکن است لازم داشته باشیم تا اهداف اسپرینت را تنظیم دوباره کنیم تا انتظارات واقع بینانه باشند.
5. **بررسی منابع جایگزین:** در صورت نیاز، بررسی امکان جذب منابع جدید یا موقتی برای پر کردن خلا نیروی کار به وجود آمده.
6. **ارتباط با مدیریت:** به مدیریت و صاحبان سهام پروژه در مورد تغییرات و چالش های پیش آمده اطلاع رسانی میکنم و اگر نیاز باشد درخواست حمایت میکنم.
7. **به روزرسانی Backlog:** اطمینان از اینکه تمامی تغییرات و تاخیرات در **backlog** پروژه به روزرسانی شده باشند.

(ب)

در مواجهه با چنین چالشی، چندین اقدام را میتوان انجام داد تا تأثیر منفی این تأخیر را کاهش داد و به پیشرفت اسپرینت کمک کرد :

1. **ارتباط فوری با سرویس دهنده:** برقراری ارتباط فوری با سرویس دهنده برای درک دقیق تاخیر فرایند و بررسی امکان سرعت بخشیدن به فرآیند. همچنین بررسی امکان دریافت دسترسی موقت یا محدود برای شروع کار.

2. **هماهنگی با تیم اسکرام:** مطرح کردن مسئله در جلسات دوره‌ای اسکرام و درخواست کمک یا پیشنهادات از سایر اعضای تیم.

3. **طرح ریزی مجدد وظایف:** بازنگری و طرح ریزی مجدد وظایف بر اساس این تأخیر. این برنامه ریزی شامل تنظیم مجدد اولویت‌ها و تمرکز بر روی بخش‌های دیگر پروژه تا زمانی که دسترسی به سرویس فراهم شود است.

4. **استفاده از محیط‌های شبیه‌سازی:** بررسی امکان استفاده از محیط‌های شبیه‌سازی برای تداوم توسعه و تست‌ها به جای انتظار دسترسی به سرویس واقعی.

5. **ارتباط با مدیریت:** اطلاع‌رسانی به مدیریت پروژه و صاحبان سهام در مورد تأخیرات و تأثیرات احتمالی آن بر جدول زمانی پروژه.

6. **بازنگری در Backlog:** به‌روزرسانی backlog پروژه برای نشان دادن وضعیت کنونی و اولویت‌بندی مجدد وظایف بر اساس دسترسی‌های فعلی.

7. **بررسی گزینه‌های جایگزین:** ارزیابی گزینه‌های جایگزین برای پیشبرد پروژه مثل استفاده از سرویس‌های مشابه یا توسعه‌ی موقتی قابلیت‌های domestic تا زمان دسترسی به سرویس اصلی.

به عنوان اسکرام مستر (Scrum Master)، من نقشی کلیدی در تضمین اجرای صحیح کارها و حل چالش‌هایی که تیم با آن‌ها روبرو می‌شود، دارم. برای اطمینان از اینکه کارها به درستی انجام میشوند و برای تعامل مؤثر با اعضای تیم در شرایط بد، می‌توانم اقدامات زیر را انجام دهم:

1. **برگزاری جلسات دوره‌ای و روزانه (Daily meetings):** این جلسات فرصتی برای بررسی پیشرفت کار و شناسایی موانع احتمالی هستند. اطمینان از اینکه هر عضو تیم فرصت بیان پیشرفت و مشکلات خود را دارد، بسیار مهم است.
2. **استفاده از ابزارهای مدیریت پروژه:** استفاده از ابزارهای مدیریت پروژه برای نمایش وضعیت کارها و پیگیری پیشرفت‌ها.
3. **گفتگوهای منظم با اعضای تیم:** برقراری ارتباط منظم و شخصی با اعضای تیم برای درک بهتر نیازها و موانع آن‌ها. این گفتگوها همچنین فرصتی برای حمایت و راهنمایی اعضای تیم است.
4. **بررسی متریک‌ها و شاخص‌های کلیدی عملکرد (KPIs):** استفاده از داده‌ها و متریک‌ها برای ارزیابی پیشرفت پروژه و تشخیص نقاط ضعف و قوت.
5. **رسیدگی به مشکلات و موانع:** اگر شرایط بد باشد، فوراً به شناسایی و حل موانع پرداخته شود. به عنوان مثال میتوانیم تسهیل گفتگوها برای حل تعارضات یا کمک به اعضای تیم برای پیدا کردن راه‌حل‌های خلاقانه را داشته باشیم.
6. **فرهنگ تیمی مثبت و حمایتی:** حفظ یک فرهنگ تیمی مثبت و حمایتی، مخصوصاً در زمان‌های چالش‌برانگیز. این کار به اعضای تیم من کمک می‌کند تا احساس تعلق و انگیزه داشته باشند.
7. **جلسات بازخورد و Retrospectives:** برگزاری جلسات بازخورد منظم برای بررسی آنچه خوب پیش رفته و زمینه‌های قابل بهبود. این جلسات فرصتی برای یادگیری و بهبود مستمر هستند.

شاخص‌های عملکردی	اسپرینت 1	اسپرینت 2	اسپرینت 3	اسپرینت 4	اسپرینت 5
وظایف تعهد شده (SP)	50	46	80	78	75
وظایف اجرا شده (SP)	26	53	69	66	72

برای ارزیابی شاخص سرعت تیم بر اساس جدول داده‌ها، ابتدا به تعداد کل نقاط داستان تکمیل شده در هر اسپرینت نیاز داریم. در اینجا، می‌توانیم میانگین نقاط داستان تکمیل شده در پنج اسپرینت را محاسبه کنیم تا یک تصویر کلی از سرعت تیم دریافت کنیم. همچنین، می‌توانیم نوسانات در سرعت تیم را بررسی کنیم که می‌تواند نشان‌دهنده تغییرات در بهره‌وری یا موانعی باشد که تیم داده شده با آن روبرو شده است.

ابتدا میانگین نقاط داستان تکمیل شده در پنج اسپرینت را محاسبه کرده و سپس بر اساس آن، تحلیلی از عملکرد تیم ارائه می‌دهیم. کد زیر پروسه مورد نیاز را برای ما انجام می‌دهد:

```
import pandas as pd

# Data
data = {
    'Sprint 1': [50, 26],
    'Sprint 2': [46, 53],
    'Sprint 3': [80, 69],
    'Sprint 4': [78, 66],
    'Sprint 5': [75, 72]
}

# Convert the data into a DataFrame
df = pd.DataFrame(data)

# Calculate the total story points for each sprint
df['Total'] = df.sum(axis=1)

# Calculate the average velocity (mean of total story points per sprint)
average_velocity = df['Total'].mean()

# Extract individual sprint totals for analysis
sprint_totals = df['Total'].tolist()

average_velocity, sprint_totals
```

و خروجی کد ما به صورت زیر است :

(307.5, [329, 286])

توصیف کلی:

بر اساس داده‌های موجود در جدول و محاسباتی که انجام شد، میانگین شاخص سرعت (Velocity) تیم برای پنج اسپرینت گذشته 307.5 (Story Points) است. این نشان می‌دهد که به طور متوسط، تیم در هر اسپرینت تقریباً 308 (Story Point) را تکمیل کرده است.

توصیف جداگانه:

در بررسی جداگانه عملکرد تیم در هر اسپرینت، داده‌هایی که بدست اوردم نشان می‌دهند که در اسپرینت اول تیم توانسته 329 (Story Point) را به سرانجام برساند و در اسپرینت دوم این میزان به 286 (Story Point) کاهش یافته است. این تغییرات می‌تواند ناشی از عوامل مختلفی باشد، از جمله تغییر در حجم کار، پیچیدگی وظایف، مهارت‌های تیم، و یا عوامل بیرونی که بر توانایی تیم در تکمیل وظایف تأثیر داشته است.

(ب)

از نمودار داده شده، چندین نکته قابل استنباط است:

1. **شروع اسپرینت:** نمودار با حدود 25 نقطه داستانی شروع می‌شود، که کار کلی که باید در اسپرینت انجام شود را نشان می‌دهد.
2. **روزهای کاری و غیرکاری:** نمودار شامل نشان‌گرهایی برای روزهای غیرکاری است، که مرتبط با آخر هفته‌ها یا تعطیلات عمومی. در این روزها پیشرفت کار انتظار رو انتظار نداریم.
3. **پیشرفت:** در برخی از روزها، مقدار کار باقی‌مانده کاهش پیدا کرده، که نشان می‌دهد تیم در حال انجام کارها و بستن داستان‌ها است.
4. **خط راهنما:** خط راهنمای خاکستری نشان می‌دهد که کارها باید با چه سرعتی برای رسیدن به پایان اسپرینت انجام شوند. این خط پیشرفت ایده‌آل را نشان می‌دهد.

5. **تحلیل پیشرفت:** بر اساس نمودار، تیم دوره‌هایی از پیشرفت و توقف داشته است. در بعضی روزها، میزان کار باقی‌مانده کاهش محسوسی داشته است، در حالی که در دیگر روزها کاهشی نداشته یا بسیار اندک بوده است. این می‌تواند نشان‌دهنده چالش‌ها یا موانعی باشد که تیم با آن‌ها مواجه شده.

6. **تحلیل نهایی:** بر اساس داده‌های نمودار به نظر می‌رسد که تیم ما عقب‌تر از خط راهنمای ایده‌آل می‌باشد که یعنی تیم ممکن است با چالش‌هایی مواجه شده باشد که باعث کندی در پیشرفت شده است. در مجموع تحلیل، تیم باید بررسی کند که چه عواملی باعث تاخیر در پیشرفت شده و راه‌حل‌هایی برای افزایش بهره‌وری برای باقیمانده اسپرینت پیدا کند. اگر آنها نتوانند سرعت خود را افزایش دهد، ممکن است تمام کارها تا پایان اسپرینت به اتمام نرسند.

(پ)

محاسبه ولاسیتی (Velocity) تیم در یک اسپرینت از مجموع امتیازات (Story Points) کارهای انجام شده تقسیم بر مدت زمان اسپرینت بدست می‌آید. در ابتدا نحوه محاسبه ولاسیتی را بر اساس داده‌های ارائه شده در فایل اکسل توضیح می‌دهم:

1. **مجموع امتیازات (Total Story Points):** ابتدا، مجموع امتیازات تمام کارهایی که در اسپرینت تکمیل شده‌اند را محاسبه کردم. این کار با جمع کردن امتیازات هر کار (story points) در فایل اکسل انجام می‌شود.

2. **مدت زمان اسپرینت (Sprint Length):** برای محاسبه مدت زمان اسپرینت، تاریخ شروع و پایان اسپرینت را بررسی کردم. تاریخ شروع اسپرینت، تاریخی است که کار اول در اسپرینت شروع شده و تاریخ پایان، تاریخی است که آخرین کار در اسپرینت تکمیل شده است. این تاریخ‌ها از ستون "Resolved" در فایل اکسل استخراج کردم.

3. محاسبه ولاسیتی (Calculating Velocity): ولاسیتی با تقسیم مجموع امتیازات بر تعداد روزهای

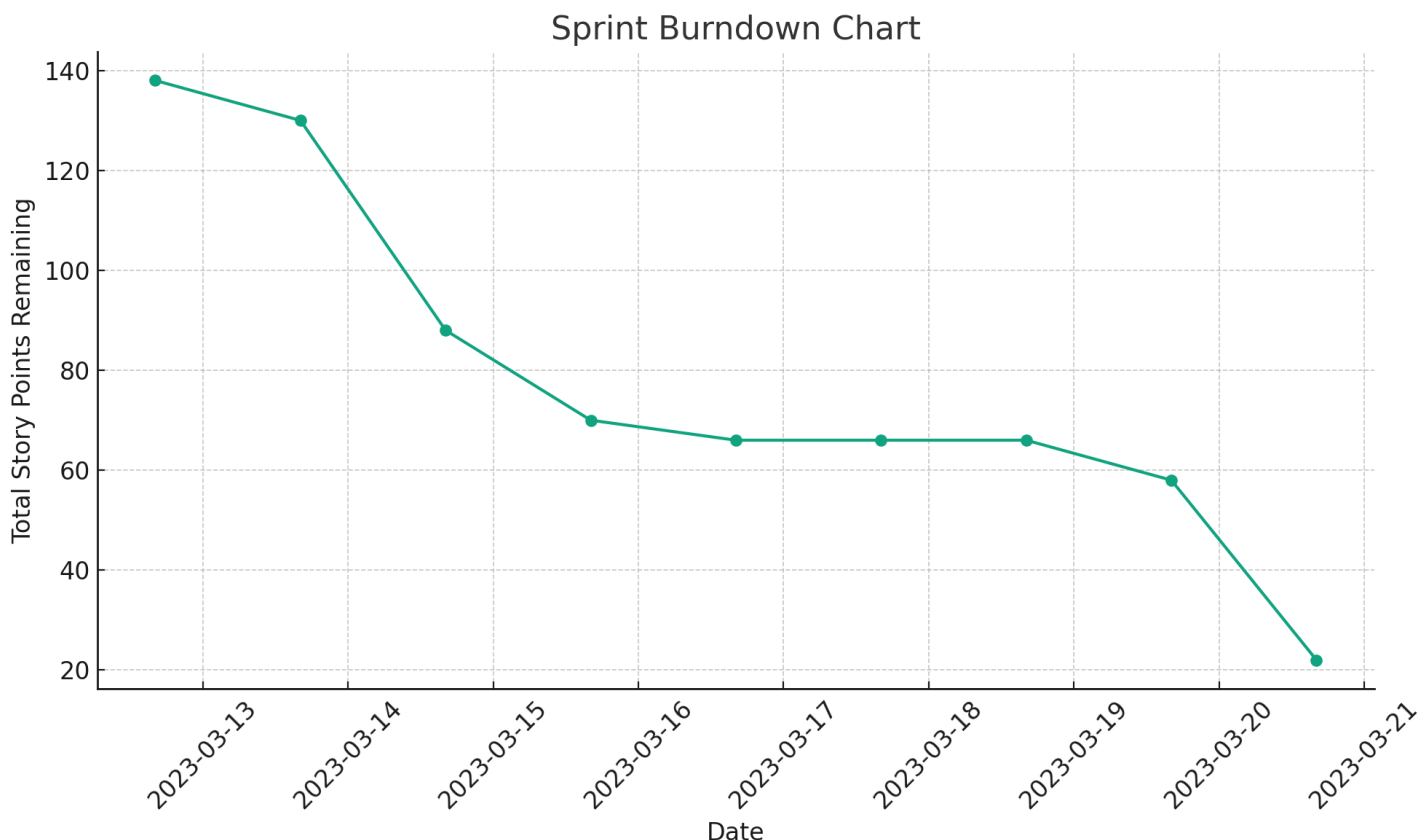
اسپرینت محاسبه میشود که به ما می گوید که به طور متوسط، تیم چه میزان امتیاز را در هر روز اسپرینت پیش میبرد.

بنابراین، فرمول محاسبه ولاسیتی به شکل زیر است:

$$\text{Velocity} = \frac{\text{Total Story Points}}{\text{Sprint Length (days)}}$$

در مورد داده های فایل اکسل، ولاسیتی داده شده تقریباً 16.11 امتیاز در روز بدست آمد. این مقدار نشان دهنده میزان پیشرفت متوسط تیم در هر روز اسپرینت است.

نمودار burn-down به صورت زیر است :



سوال 3

برای پیاده‌سازی یک پروژه نرم‌افزاری با معماری سرویس‌گرا و خصوصیات ذکر شده، می‌توان ساختار تیم و مهارت‌های لازم برای اعضای تیم را بر اساس مدل چهار وجهی Topology Team به این شکل پیشنهاد داد:

1. تیم توسعه وب و موبایل

- **وظایف:** این تیم مسئول طراحی و توسعه رابط‌های کاربری وبسایت و اپلیکیشن‌های موبایل است. آن‌ها باید تجربه کاربری (UX) جذاب و کاربرپسندی را فراهم کنند.
- **مهارت‌ها:** تسلط بر HTML, CSS, JavaScript ، و فریمورک‌هایی مانند React یا Angular برای وب، و آشنایی با Swift برای iOS و Java یا Kotlin برای اندروید.
- **مثال:** توسعه یک وبسایت responsive با استفاده از React که با واسط‌های برنامه‌نویسی نرم‌افزار (APIs) برای دریافت و ارسال داده‌ها به سرور ارتباط برقرار می‌کند.

2. تیم پایگاه داده

- **وظایف:** مدیریت داده‌ها، اطمینان از دسترسی سریع و امن به داده‌ها، و بهینه‌سازی پردازش‌های داده‌ای.
- **مهارت‌ها:** دانش SQL و NoSQL ، تجربه با پایگاه داده‌هایی مانند MySQL, PostgreSQL, MongoDB، و توانایی بهینه‌سازی پرس‌وجوها و تراکنش‌ها.
- **مثال:** طراحی یک ساختار پایگاه داده‌ای برای ذخیره‌سازی و پردازش داده‌های کاربران با استفاده از MongoDB ، به گونه‌ای که اجازه پردازش و تحلیل داده‌ها را به طور موثر فراهم کند.

3. تیم یادگیری ماشین

- **وظایف:** تجزیه و تحلیل داده‌های کاربران و استفاده از الگوریتم‌های یادگیری ماشین برای استخراج بینش‌ها و توسعه ویژگی‌های هوشمند.
- **مهارت‌ها:** دانش ریاضی و آمار، مسلط به زبان‌های برنامه‌نویسی مانند Python ، و آشنایی با کتابخانه‌های یادگیری ماشین مانند TensorFlow یا PyTorch .

- مثال: ایجاد یک مدل predictive که رفتار کاربران را بر اساس داده‌های جمع‌آوری شده تحلیل کرده و پیشنهادات شخصی‌سازی شده در اپلیکیشن موبایلی ارائه دهد.

4. تیم زیرساخت و استقرار

- وظایف: راه‌اندازی، نگهداری و به‌روزرسانی محیط‌های توسعه، آزمایش و تولید.
- مهارت‌ها: تجربه در DevOps، مدیریت سرور، و آشنایی با ابزارهای استقرار و مدیریت نرم‌افزار مانند Docker، Kubernetes و سیستم‌های CI/CD.
- مثال: ایجاد یک فرایند CI/CD با استفاده از Jenkins که به طور خودکار کد جدید را تست کرده و در محیط Stage برای تست‌های بیشتر قرار دهد.

5. تیم ارزیابی کیفیت (QA)

- وظایف: انجام تست‌های دقیق و مستقل برای اطمینان از کیفیت بالای نرم‌افزار قبل از انتشار نهایی.
- مهارت‌ها: دانش در تست نرم‌افزار، تست اتوماسیون، و استفاده از ابزارهای تست مانند Selenium یا QTP.
- مثال: اجرای یک سری تست‌های اتوماتیک با استفاده از Selenium برای بررسی عملکرد وب‌سایت در مرورگرهای مختلف، تضمین کردن عملکرد صحیح وب‌سایت در شرایط متفاوت.

این ساختار تیم باعث می‌شود که هر بخش از پروژه توسط متخصصانی با مهارت‌های مرتبط اداره شود و تضمین می‌شود که تمام جنبه‌های پروژه با دقت و کیفیت بالا پیش برود. همچنین، با توجه به نرخ توسعه بالا و حساسیت پروژه، این ساختار تیمی اطمینان می‌دهد که تمامی جنبه‌های فنی و کیفی به طور موثر مدیریت شوند.