



به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
پلی تکنیک تهران

درس مهندسی نرم افزار ۲، نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

تمرین دو

توضیحات تکمیلی:

- مباحث این تمرین از فصل پنجم کتاب پرسمان و اسلاید فرایند های چابک می باشد.
- مهلت تحویل تمرین ۲۱ فروردین در نظر گرفته شده و قابل تمدید نمی باشد.
- پاسخ به تمرین ها به صورت گروهی می باشد و همه اعضای گروه موظفند در آن مشارکت داشته باشند.
- در صورت برخورد با پاسخ های مشابه بیش از حد بین گروه های مختلف، نمره کسر خواهد شد.
- ارسال فایل مربوطه در سامانه کورسز توسط سر گروه (نام ذکر شده در گوگل شیت گروه بندی های درس) کافی می باشد و نیازی به بار گذاری توسط تمام اعضای گروه نیست.
- نظم و خوانایی تمرین از اهمیت بالایی برخوردار می باشد.
- خواهش می شود تمرین خود را در قالب یک فایل PDF با نام "SE2_HW[number]_[Group Number]" مانند: "SE2_HW2_6.pdf" در مهلت یاد شده در سایت درس بار گذاری فرمایید.
- توصیه می کنیم قبل از حل سوالات تشریحی، قسمت امتیازی مربوط به جیرا را مطالعه کنید.
- پرسش های خود درباره این تمرین را می توانید از راه ایمیل se2springta@gmail.com بیان کنید.

۱. الف) در فصل های گذشته به فرایند های سنتی تولید و توسعه نرم افزار پرداخته شد. به چه دلیل فرایند ها و اصول چابک^۱ ظهور کردند؟

تا مدت ها فرایند های سنتی نظیر آبشاری استفاده می شد. فرایند ها و اصول چابک به دلیل نیاز به تغییرات در روش های مدیریت پروژه و توسعه نرم افزار به وجود آمدند. در گذشته، روشهای سنتی مدیریت پروژه و توسعه نرم افزار، معمولاً بر اساس مدل های آبشاری بودند و برای هر فاز پروژه، مراحل خاص خود (نظیر جمع آوری نیازمندی، طراحی، پیاده سازی، تست، نگهداری) را داشتند که باید به ترتیب انجام شوند. اما این روش، امکان اصلاح و تغییر در پروژه را به شدت محدود میکرد. با ظهور فرایند ها و اصول چابک، روشهای مدیریت پروژه و توسعه نرم افزار تغییر کردند. در روش چابک، تمرکز بر روی ارائه محصولات قابل استفاده و کاربردی به مشتریان است و فرایند توسعه در قالب Incremental و Iterative صورت میگیرد. همچنین در این روش، توجه بیشتری به همکاری و تعامل بین اعضای تیم و توسعه دهندگان، بهبود فرایند تحویل و همچنین کاهش هدر رفت زمان و هزینه ها میشود.

ب) چابکی^۲ را به اختصار توضیح دهید.

پاسخ موثر (سریع و سازگار) به تغییرات – ارتباط موثر با ذی نفعان و مشارکت مشتری در پروژه – تحویل سریع نرم افزار – سازماندهی تیم و برنامه ریزی منعطف

ج) در چه پروژه هایی متدلوژی چابک کاربرد دارد و در چه پروژه هایی نباید از آن استفاده کنیم؟

توسعه چابک مزایای زیادی دارد ولی برای تمام محصولات و پروژه ها مناسب نمی باشد. پروژه هایی که باید بر اساس الزامات بسیار خاص، اغلب قانونی یا نظارتی ارائه شوند، برای چابکی نیز مناسب نیستند. برای پروژه های خیلی بزرگ که میزان تلاش و زمان و ریسک نیاز به محاسبه دقیق دارد این روش چالش دارد. در پروژه های بسیار کوچک و پروژه هایی که نیازمندی ها و شرایط ثابت است، نیازی به استفاده از این متدلوژی نیست. در پروژه هایی که امکان ارتباط با مشتری وجود ندارد یا تیم مورد نظر این روش را بلد نبوده یا خودساز مانده نیستند مناسب نیست. همچنین در صورتی که مستند سازی در پروژه اهمیت زیادی داشته باشد این روش معایب زیادی دارد.

^۱ Agile^۲ Agility

۲. در بیانیه چابک^۳ چهار ارزش^۴ کلیدی و دوازده اصل^۵ وجود دارد. به نظر شما در هر کدام از اصول، کدام ارزش یا ارزش ها پر رنگ تر هستند؟

* به طور مثال برای اصل "کارآمدترین و موثرترین روش انتقال اطلاعات به تیم توسعه و تبادل آن در میان اعضای تیم، گفتگوی چهره به چهره است" ارزش "افراد و تعاملات بالاتر از فرآیندها و ابزارها" پر رنگ تر است. امکان دارد برای یک اصل چند ارزش ذکر شود. اصول در اسلاید ها و کتاب قرار دارد، برای ترجمه فارسی آن می توانید از این سایت^۶ استفاده کنید.

ارزش ها (به ترتیب سایت بالا): ۱- افراد و تعاملات بالاتر از فرآیندها و ابزارها ۲- نرم افزار کارکننده بالاتر از مستندات جامع ۳- مشارکت مشتری در انجام کار بالاتر از قرارداد کار ۴- پاسخگویی به تغییرات بالاتر از پیروی یک طرح اصول ها (به ترتیب سایت بالا نوشته شده و شماره ارزش متناظر در روبروی آن نوشته شده است):

۱- بالاترین اولویت ما جلب رضایت مشتری با تحویل زود و مداوم نرم افزاری ارزشمند می باشد: ۲ و ۳ و ۴

۲- استقبال از تغییر نیازمندی ها، حتی در اواخر فرآیند توسعه. فرآیند های چابک: ۲ و ۳ و ۴

۳- تحویل زود به زود نرم افزار قابل استفاده دو، سه هفته یک بار تا دو ، سه ماه یک بار با ترجیح بر فاصله کم: ۲ و ۳ و ۴

۴- ذی نفعان کسب و کار و توسعه دهنده ها می بایست به صورت روزانه در طول پروژه با هم کار کنند: ۱ و ۳ و ۴

۵- پروژه ها را بر دوش افراد با انگیزه بنا کنید. فضای لازم را به آنها بدهید و از نیازهای آن ها پشتیبانی کنید: ۱

۶- کارآمد ترین و موثرترین روش انتقال اطلاعات به تیم توسعه و تبادل آن در میان اعضای تیم، گفتگوی چهره به چهره: ۱

۷- نرم افزار قابل استفاده اصلی ترین معیار سنجش پیشرفت است: ۲

۸- فرآیند های چابک توسعه پایدار را ترویج می دهند حامیان مالی، توسعه دهندگان و کاربران باید بتوانند سرعت پیشرفت ثابتی را برای مدت نا محدودی حفظ کنند: ۳ و ۴

۹- توجه مداوم به برتری فنی و طراحی خوب باعث افزایش چابکی می شود: ۲ و ۴

³ Manifesto for Agile Software Development

⁴ Value

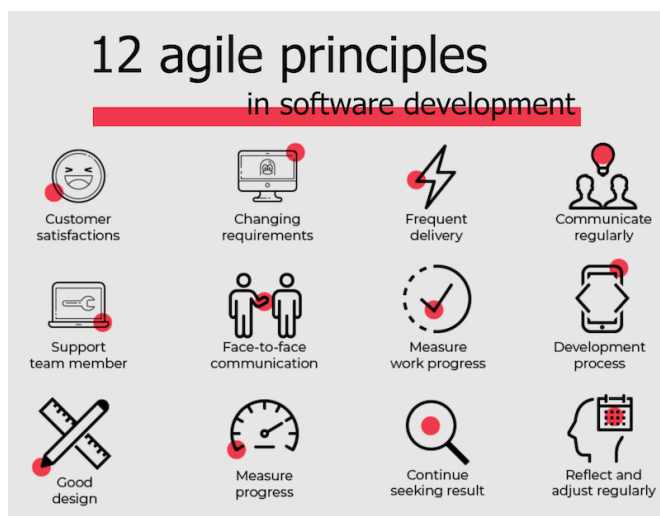
⁵ Principles

⁶ <https://agilemanifesto.org/iso/pr/manifesto.html>

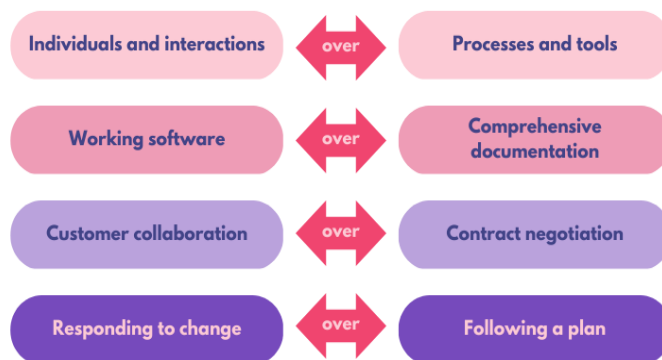
۱۰- سادگی، هنر به حداکثر رساندن مقدار کار انجام نشده، ضروری است: ۲

۱۱- بهترین معماری ها، نیازمندی ها و طراحی ها از تیم های خود سازمان ده پدید آور می شود: ۱ و ۲

۱۲- در فواصل منظم، تیم بر چگونگی موثرتر شدن تامل و تفکر می نماید و سپس تیم رفتار خود را بر اساس بازتاب این تفکر تنظیم و هم سو می نماید: ۱ و ۴



4 Agile Manifesto values



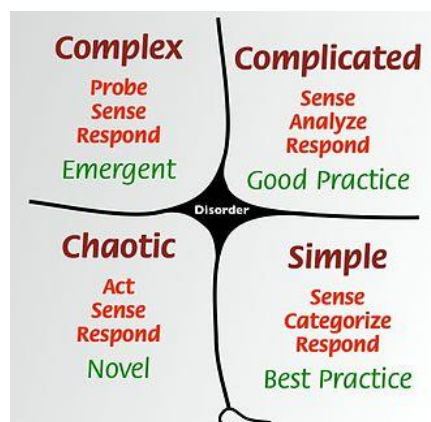
۳. الف) چارچوب کانوین^۷ چارچوبی است که به فهم موقعیت و انتخاب رویکرد مناسب برای تصمیم گیری در مورد سیستم کمک می کند. در مورد این چارچوب و دسته بندی (دامنه^۸) های آن توضیح مختصری دهید و برای هر کدام از این دسته ها نمونه ای مثال بزنید و دلیل انتخاب آن برای آن دسته را بنویسید.

* توجه کنید سیستمی مانند فروشگاه اینترنتی بالقوه می تواند در چهار دسته قرار بگیرد ولی اهمیت دارد که این سیستم چه ویژگی هایی دارد و (به طور مثال) اگر در دسته complex قرار می گیرد به علت چه ویژگی هایی در آن است، در غیر این صورت اگر سیستمی واضح، بدون هر گونه پیچیدگی و ابهام، دارای best practice باشد در دامنه Clear - Obvious قرار می گیرد. بنابراین تحلیل گروه شما در هر مورد اهمیت زیادی دارد.

چارچوب کانوین چارچوبی است که به فهم موقعیت و انتخاب رویکرد مناسب برای تصمیم گیری در مورد سیستم کمک می کند. چارچوب کانوین پنج «دامنه» یا بافت تصمیم گیری پیشنهاد می کند.

⁷ Cynefin

⁸ Domain



:Clear

در این حوزه، رابطه علت و معلولی واضح و آشکار است. به نوعی "شناخته شده های معلوم" را نشان می دهد. راه حل اغلب بهترین روش **best practice** یا یک رویکرد مبتنی بر قانون است. اسنودن استدلال می کند که باید به تصمیمات واضح "حس کنید - طبقه بندی کنید - پاسخ دهید" پاسخ داد. به زبان ساده، ابتدا باید موقعیت ارزیابی شود، نوع آن دسته بندی شود، و سپس پاسخ بر اساس بهترین عملکرد بر اساس یک فرآیند یا رویه موجود وجود دارد. به طور مثال پرداخت وام در این دسته وجود دارد. این مسئله بسیار ساده بوده و بهترین روش برای آن می توان ارائه کرد و مشکل را می توان شناسایی کرد (وام گیرنده هزینه کمتری را پرداخت کرده) و آنرا طبقه بندی کرد (اسناد وام ها بررسی شود) و در نهایت پاسخ داد (مطابق قانون های وام پیگیری شده و پرداخت انجام شود)

:Complicated

در این دامنه رابطه بین علت و معلول نیاز به تجزیه و تحلیل یا تخصص دارد. به نوعی "نا شناخته های معلوم" را نشان می دهد. طیفی از پاسخهای صحیح وجود دارد. این چارچوب "حس کنید - تجزیه و تحلیل کنید - پاسخ دهید" را توصیه می کند واقعیت ها را "ارزیابی کنید، تجزیه و تحلیل کنید" و روش عملی مناسب را اعمال نمایید. در این مسائل **good practice** وجود دارد. به عنوان مثال تشخیص بیماری: یک پزشک از تخصص و دانش خود در علم پزشکی برای تشخیص وضعیت بیمار استفاده می کند. رابطه علت و معلولی بلافاصله مشخص نیست، اما می توان از طریق تحلیل و کارشناسی تعیین کرد.

:Complex

در این دامنه علت و معلول فقط با نگاه به گذشته قابل استنباط است و هیچ پاسخ صحیحی وجود ندارد. در واقع "ناشناخته های نامعلوم" را نشان می دهد. اسنودن و بون می نویسند: "ممکن است الگوهای آموزنده ای ظهور کنند، اگر رهبر آزمایش هایی انجام

دهد که در صورت شکست ایمن باشد"، چارچوب کانوین این فرایند را "بکاو - حس کن - پاسخ بده" می‌نامد. معمولاً این دامنه برای مسائل مبهم و نامعلوم استفاده می‌شود.

مثلاً قبل از وجود شرکت‌هایی نظیر دیوار و دیجی کالا فرض کنید این دو وجود نداشته و محصول شرکت مورد نظر تلفیقی از این دو سامانه بوده است. یعنی فروشگاه اینترنتی که اجناس دسته دوم هم دارد و مردم در خرید و فروش مشارکت می‌کنند. پس شرکت می‌خواهد یک محصول کاملاً جدید که قبلاً نمونه‌ای از آن وجود نداشته است را تولید کند. به دلیل اینکه این محصول قبلاً وجود نداشته پس ما نمیدانیم با چه شرایطی روبرو خواهیم شد و باید ابتدا در مورد مسئله جستجو و کاوش کنیم و بعد از بدست آوردن درک مناسب از شرایط، پاسخ مناسب را بدست آوریم.

:Chaotic

در این دامنه علت و معلول نامشخص است. ر نتیجه هدف اولیه‌ی شما این است که نظم و پایداری را برقرار کنید. معمولاً بحران و سر برآوردن سناریوها در این ناحیه اتفاق می‌افتند. پاتریک لمب می‌نویسد: «وقایع در این حوزه» بیش از حد گیج کننده هستند که نمی‌توان در انتظار پاسخی دانش بنیان بود. در این زمینه، مدیران "عمل می‌کنند - حس می‌کنند - پاسخ می‌دهند" یعنی برای ایجاد نظم عمل کنید؛ احساس کنید که ثبات در چیست و سپس پاسخی دهید تا آشوب را به پیچیدگی تبدیل کنید. برای هدایت موفقیت‌آمیز یک وضعیت آشفته، مدیریت ریسک را انجام دهید تا خطرات ممکن را مشخص و بتوانید آنها را اولویت بندی کنید تا مطمئن شوید که یک برنامه‌ی مدیریت بحران همه جانبه را در اختیار دارید. در این شرایط، آماده شدن برای تمامی وضعیت‌ها غیرممکن است، اما معمولاً برنامه‌ریزی برای خطرهای قابل تشخیص، میتواند کمک‌کننده باشد. به طور مثال واکنش به حمله تروریستی: بلافاصله پس از یک حمله تروریستی، وضعیت آشفته و غیرقابل پیش بینی است. پاسخ دهندگان ممکن است نیاز به اقدام سریع و قاطع برای تثبیت وضعیت داشته باشند، مانند تخلیه مردم از منطقه یا مهار خسارت.

:Confusion

موقعیت‌هایی را نشان می‌دهد که هیچ شفافیتی در مورد این که مسئله در کدامیک از دامنه‌های دیگر مصداق یافته‌است، وجود ندارد. تشخیص اینکه آیا در وضعیت اختلال قرار دارید یا خیر، می‌تواند بسیار دشوار باشد. در اینجا روشن نیست که کدامیک از چهار دامنه غالب است و افراد عموماً به تکنیک‌هایی برای تصمیم‌گیری اعتماد می‌کنند که شناخته‌شده و راحت باشند. هدف اولیه‌ی شما در این وضعیت این است که اطلاعات بیشتری جمع‌آوری کنید، به این ترتیب می‌توانید به دامنه‌های شناخته‌شده وارد شوید و سپس تصمیم مناسب را اتخاذ کنید. اسنودن و بون می‌نویسند: «در اینجا، چشم اندازه‌های متعدد به برجستگی رخ می‌نماید، رهبران جناح با یکدیگر بحث می‌کنند و آشفتگی حکم فرمایی می‌کند». راه برون

رفت از این قلمرو این است که شرایط را به قسمت‌های تشکیل دهنده تقسیم کنیم و هر کدام را به یکی از چهار قلمرو دیگر نسبت دهیم. سپس رهبران می‌توانند تصمیم‌سازی کنند و به روش‌هایی متناسب با زمینه دخالت کنند.

مثال: پاسخ به یک بیماری همه گیر جهانی: در مراحل اولیه یک بیماری همه گیر جهانی، ممکن است مشخص نباشد که کدام حوزه اعمال می‌شود، زیرا وضعیت به سرعت در حال تحول و غیر قابل پیشبینی است. به نوعی گام‌های هیچ کدام از دامنه‌ها وجود ندارد و در این مثال نه می‌توان سریع عمل کرد و واکنش‌ها تولید کرد نه می‌توان خود مسئله را دقیق کاوید چرا که شرایط آشوب‌ناکی هم دارد و به طور کلی در هیچ کدام از دسته‌ها قرار نمی‌گیرد.

(ب) به نظر شما مدل چابک برای کدام دامنه (یا دامنه‌ها) عملکرد بهتری دارد؟ چرا؟

مدل Agile برای حوزه complex مناسب‌ترین است. در این حوزه، عوامل متعددی دخیل هستند و روابط علت و معلولی همیشه مشخص نیست. در چنین شرایطی، رویکردهای سنتی و خطی مدیریت پروژه ممکن است مؤثر نباشند. رویکرد تکراری و افزایشی مدل Agile می‌تواند به مدیریت چنین پیچیدگی کمک کند. Agile به تیم‌ها اجازه می‌دهد تا در تکرارهای کوتاه کار کنند، به سرعت با تغییرات سازگار شوند و ارزش افزایشی را به مشتری ارائه دهند. این رویکرد به تیم‌ها امکان می‌دهد از اشتباهات خود درس گرفته و راه‌حلهای خود را به طور مداوم اصلاح کنند. بنابراین، مدل Agile برای حوزه complex چارچوب Cynefin، جایی که تاکید بر همکاری، آزمایش و انطباق است، مناسب‌تر است. برای دامنه complicated و clear هم می‌توان از روش چابک استفاده کرد ولی معمولاً این مسائل تنوع بیشتری در متدلوژی‌های مختلف دارند ولی دسته‌ای که به طور خاص به روش چابک مربوط می‌شود complex است.

۴. الف) یکی از تکنیک‌های مورد استفاده در متدلوژی چابک و چارچوب XP، Pair Programming می‌باشد. آنرا به اختصار توضیح دهید.

یک مفهوم کلیدی در طول فعالیت کدنویسی (و یکی از جنبه‌های مورد بحث XP) برنامه نویسی زوجی است. XP توصیه می‌کند که دو نفر با هم در یک ایستگاه کاری کامپیوتری برای ایجاد کد برای یک استوری کار کنند. این مکانیسمی را برای حل مشکلات بلادرنگ (دو فکر اغلب بهتر از یک فکر هستند) و تضمین کیفیت در زمان واقعی (کد هنگام ایجاد بررسی می‌شود) فراهم می‌کند. همچنین توسعه دهندگان را بر روی مشکل موجود متمرکز می‌کند. در عمل، هر فردی نقش کمی متفاوت را بر عهده می‌گیرد. برای مثال، یک نفر ممکن است در مورد جزئیات کدگذاری بخش خاصی از طراحی فکر کند، در حالی که دیگری اطمینان حاصل کند که استانداردهای کدنویسی (بخش مورد نیاز XP) رعایت می‌شود یا اینکه کد استوری با آزمون واحدی که توسعه داده شده است را برای تأیید اعتبار کد برآورده می‌کند. همانطور که برنامه نویسان جفت کار خود را تکمیل

می کنند، کدی که توسعه می دهند با کار دیگران یکپارچه می شود. مهمترین هدف برنامه نویسی دوفره افزایش کیفیت نرم افزار است. به این صورت که با بررسی برنامه نویسی دوم از پیاده سازی راه حل های مشکل دار و غلط جلوگیری می شود. این روش همچنین سبب تبادل دانش میان دو برنامه نویسی می شود.

ب) مزایا و معایب این تکنیک را ذکر کنید. چه زمانی استفاده از این روش مناسب است؟

مزایا: باگ و خطای کمتر - بهبود تست - بهبود کیفیت کد - ارتقا روحیه تیمی - یادگیری سریعتر و انتقال دانش - انعطاف بیشتر به این معنا که اگر یک شخص غایب شود دیگری کامل کد را می شناسد

معایب: هزینه بیشتر - چالش مدیریت مهارت بین دو نفر و نیاز به هماهنگی - خستگی و کاهش تمرکز - نیاز به مهارت اجتماعی - نمی توان برای زمان زیاد این روش را اجرا کرد - عدم استقلال

استفاده از این روش وقتی مناسب است که: کیفیت کد از سرعت توسعه آن برایمان بیشتر اهمیت داشته باشد. دو نفر دارای خصوصیات، مانند مدل تایم استراحتشان، توانایی فنی، و تواناییهای ارتباطی مشابه باشند. نویسنده کد معمولاً خطا زیاد دارد (در این صورت جفت شدن وی با یک برنامه نویس دقیق میتواند ایده خوبی باشد). در پروژه هایی که اعضا نیاز به ارتباط فراوان و سر در آوردن از کارکرد قسمتهای مختلف سیستم دارند. برای سیستم هایی که احتمال بروز مشکل در آنها در آینده زیاد است. در کنار توسعه محصول، تربیت و کسب تجربه اعضا تیم نیز اولویت دارد. برای افزایش روحیه تیمی و بهبود تعاملات هم می توان استفاده کرد و در صورتی که اعضای تیم اصرار بر کار فردی داشته باشند این روش مناسب نیست.

۵. الف) نقش 'های اسکرام' را نام برده و در حد یک خط وظیفه هر کدام را بنویسید.

Product owner: نماینده مشتری که ارتباط با ذی نفعان دارد و اطمینان می یابد در راستای بالاترین کیفیت محصول هستیم. مدیریت بک لاگ و چشم انداز محصول هم وظیفه او می باشد.

Scrum master: کوچینگ که جلسات را رهبری می کند و مطمئن می شود تیم بر اساس اسکرام و اجایل پیش می رود. وظیفه برگزاری جلسات با او می باشد.

Development team: توسعه نرم افزار و طراحی و پیاده سازی و تست و نگهداری همگی با این تیم می باشد. مسئولیت پیشرفت کار ها در اسپرینت را دارند.

ب) Sprint Backlog و Product Backlog را توضیح دهید.

¹⁰ Role¹¹ Scrum

Product backlog لیستی اولویت دار شامل تمام نیازمندی ها و الزامات و استوری ها و کار هایی است که باید انجام شود. توسط product owner نگهداری می شود که شامل تمام نیازها و ویژگی های محصول است. Sprint backlog به عنوان یک زیرمجموعه بک لاگ محصول تلقی می شود که فقط شامل الزامات و ویژگی های مربوط به یک اسپرینت خاص است که تیم توسعه مسئول انتخاب و بروز رسانی آیتم های آن لیست است.

ج) در جلسات روزانه اسکرام باید به چه سوال هایی پاسخ داده شود؟

جلسات روزانه اسکرام که معمولاً ۱۵ دقیقه ای بوده و توسط اسکرام مستر برگزار می شود که اعضای تیم توسعه باید به سه سوال پاسخ دهند: از جلسه قبل چه کارهایی انجام شده - با چه موانعی روبرو می شوید - چه برنامه ای برای اتمام کار ها تا جلسه بعدی دارید.

به زبان ساده تر: کار های دیروز، کار های امروز، چالش ها و مشکلات

۶. چارچوب XP و اسکرام را با هم مقایسه کنید.

شماره	Extreme Programming (XP)	Scrum
۱.	در این مدل چرخه ها ۱ تا ۲ هفته طول می کشند.	چرخه های این مدل Sprint نام دارند که از ۲ هفته تا ۱ ماه طول می کشند.
۲.	این مدل اجازه ی ایجاد تغییر در timeline را می دهد.	در این مدل اجازه ی تغییر در timeline ها و guideline ها را نداریم.
۳.	این مدل بر strong engineering practices تاکید دارد.	این مدل بر self-organization تاکید دارد.
۴.	در این مدل تیم باید یک ترتیب اولویت دقیق یا ترتیب اولویت از پیش تعیین شده را دنبال کند.	در این مدل تیم توالی توسعه محصول را تعیین می کند.
۵.	این مدل به صورت کامل تعریف شده است و برای اعمال آن نیاز به کار اضافی نیست.	این مدل به صورت کامل و دقیق تعریف نشده است و هر تیم می تواند با توجه به نیاز خود این بخش های ناقص را کامل کند.
۶.	این مدل بر روش ها و تکنیک هایی تاکید دارد که برنامه نویسان برای اطمینان از نتیجه بهتر، باید از آن ها استفاده کنند.	این مدل بر روی شیوه های مهندسی نرم افزار که توسعه دهندگان باید از آن ها استفاده کنند تاکید نمی کند.

۷.	این مدل بر روش‌های برنامه نویسی گروهی، طراحی ساده، بازسازی و ... برای اطمینان از پیشرفت یا کیفیت بهتر بسیار سختگیر است.	در این مدل برنامه نویسان ملزم هستند که از روش‌هایی استفاده کنند که از پیشرفت و کیفیت بهتر پروژه آگاه باشند.
۸.	در ترجیح ویژگی‌ها، تقاضا مطابق با اولویت است.	در ترجیح ویژگی‌ها، تقاضا و اولویت نباید با یکدیگر همسو باشند.
۹.	در این مدل، مشتری اولویت‌های شغلی را به عنوان product owner تعیین می‌کند و سپس انتشارات را تجزیه و تحلیل می‌کند.	در این مدل، scrum master از product owner می‌خواهد که وظایف را با توجه به نیاز خود اولویت بندی کند.
۱۰.	وظایف توسط مشتری اولویت بندی می‌شوند و اولویت‌های وظایف توسط تیم توسعه قابل تغییر نیستند.	وظایف توسط product owner اولویت بندی می‌شوند، اما با انعطاف پذیری که در صورت نیاز، تیم برنامه نویسی می‌تواند اولویت‌ها را بعداً تغییر دهد.
۱۱.	ارزش‌ها: ارتباط، سادگی، بازخورد	ارزش‌ها: منعطف نسبت به تغییرات، تمرکز، تعهد
۱۲.	مشارکت مشتری در پروژه بیشتر است.	مشارکت مشتری در پروژه کمتر است.

هر دو چارچوب فرایندهای چابک می‌باشند، پس تمام ارزش‌ها و اصول‌های اجایل (نظیر تحویل زودهنگام، ارتباط موثر و ...) در آنها می‌باشد. به طور کلی Scrum بر روی مدیریت پروژه و تولید نرم افزار قابل عرضه تمرکز دارد، XP بر روی بهبود فرآیندهای توسعه نرم افزار و کیفیت تمرکز دارد و XP به نوعی بیشتر وارد پیاده سازی و جزئیات می‌شود. باید توجه داشته باشیم اسکرام مستر در اسکرام نقشی رسمی بوده ولی چنین نقشی در XP وجود ندارد. در کل بسته به نیاز حالت ایده آل این است که تلفیقی از مزایای آنها را در پروژه مورد نظر اجرا کرد.