

به نام خدا

# آزمون میان‌ترم اول + پاسخ

نیم‌سال دوم ۱۴۰۱-۱۴۰۲

۲۶ فروردین ۱۴۰۲

1. تعریف نرم‌افزار، شامل کدام یک از گزینه‌های زیر نمی‌شود؟
  1. مجموعه‌ای از برنامه‌ها که روی سیستم اجرا می‌شوند.
  2. داده‌ساختارهایی که به برنامه اجازه‌ی کار با داده را می‌دهند.
  3. مستنداتی که کارکرد برنامه را شرح می‌دهند.
  4. کنار هم قرار گرفتن لایه‌های انتزاع مختلف که در کنار یکدیگر، یک نیاز را برطرف می‌کنند.

پاسخ: ۴

اسلاید ۱، صفحه‌ی ۲

2. کامپایلرها در کدام دسته از نرم‌افزارها قرار می‌گیرند؟

1. Application software
2. Product-line software
3. Engineering software
4. System software

پاسخ: ۴

کتاب، صفحه‌ی ۶ و ۷

3. کدام گزینه به طور کلی درباره‌ی legacy software ها صحیح نیست؟

1. عدم وجود مستندات یا مستندات ضعیف و بی‌کیفیت در بسیاری از موارد.
2. به طور معمول شامل بخش‌ها اصلی و ضروری یک business نیستند.
3. معمولاً طراحی غیرقابل توسعه دارند.
4. تست‌کیس‌ها و نتایج آنها آرشیو نشده‌اند.

پاسخ: ۲

کتاب، صفحه‌ی ۸

4. کدام گزینه درباره WebApp ها صحیح نیست؟

1. ویژگی اصلی بسیاری از WebApp ها ارائه‌ی داده در اشکال مختلف به کاربر است.
2. به طور پیوسته در حال تغییر هستند.
3. به طور معمول در WebApp ها عامل time to market نقش مهمی ندارد.
4. گزینه‌های الف و ب

پاسخ: ۳

اسلاید ۱، صفحه‌ی ۱۲

5. اگر به مهندسی نرم‌افزار به چشم یک تکنولوژی لایه‌لایه نگاه کنیم، کدام گزینه جزو لایه‌های مهندسی نرم‌افزار به حساب نمی‌آید؟

1. Tools
2. Process model
3. Documentation
4. Methods

پاسخ: ۳

اسلاید ۲، صفحه‌ی ۴

6. کدام گزینه جزو فعالیت‌های چتری به حساب نمی‌آید؟

1. مدیریت ریسک
2. مدل‌سازی
3. اندازه‌گیری
4. کیفیت سنجی

پاسخ: ۲

اسلاید ۲، صفحه‌ی ۷

7. مدل ایجاد نرم‌افزار آبشاری:

1. در صورتی که نیازمندی‌ها به خوبی تعریف شده باشند، رویکرد مناسبی است.
2. در صورتی که برنامه کاری به سرعت باید انجام شود، رویکرد خوبی است.
3. بهترین رویکرد برای زمانی است که تیم ایجاد نرم‌افزار بزرگ است.
4. یک مدل قدیمی است که امروزه استفاده نمی‌شود.

پاسخ: ۱

کتاب، صفحه‌ی ۴۱ و ۴۲

8. کدام یک از فعالیتهای چتری زیر می‌تواند در ارتباط با سایر فرایندهای چتری انجام پذیرد؟

- 1. Technical reviews
- 2. Risk management
- 3. Measurement
- 4. Software configuration management

پاسخ: ۳

کتاب، صفحه‌ی ۱۸

9. کدام گزینه درباره‌ی process pattern ها صحیح نیست؟

- 1. با ترکیب process ها، یک تیم ایجاد نرم‌افزار می‌تواند مشکلات مختلف را حل کند.
- 2. در مواردی می‌توان از pattern ها برای توصیف مشکل و ارائه راه‌حل در رابطه با یک framework activity استفاده کرد.
- 3. Process pattern یعنی ارائه‌ی راه‌حل برای مشکلات به وجود آمده.
- 4. pattern ها نمی‌توانند در سطوح انتزاع مختلف تعریف شوند.

پاسخ: ۴

کتاب، صفحه‌ی ۳۵



10. با توجه به این که علوم ایجاد نرم‌افزار به طور پیوسته به‌روز می‌شوند، چرا افراد میان‌سالی مانند Martin Fowler که ۵۸ سال سن دارد، از بهترین مهندسين نرم‌افزار جهان هستند؟

- 1. این افراد به طور پیوسته آخرین تکنولوژی‌های کدنویسی را مطالعه و تمرین می‌کنند.
- 2. تجربه‌ی آن‌ها در انجام پروژه‌های بزرگ دولتی و خصوصی باعث معروف‌شدنشان شده است.
- 3. این افراد با اصول مهندسی نرم‌افزار آشنا شده‌اند که مفاهیمی تقریباً ثابت و مشخص در طول زمان است.
- 4. همه‌ی موارد

پاسخ: ۳

اسلاید ۷ صفحه‌ی ۲ (اصول مهندسی نرم‌افزار ثابت هستند و در طول زمان به ندرت تغییر نمی‌کنند)

11. کدام گزینه از فازهای متدولوژی UP نیست؟

- 1. Inception
- 2. Elaboration
- 3. Construction
- 4. Validation

پاسخ: ۴

اسلاید ۴، صفحه ۱۱

12. در کدام فاز از متدولوژی UP، طراحی وزن بیشتری دارد؟

- 1. Inception
- 2. Elaboration
- 3. Construction
- 4. Transition

پاسخ: ۲

اسلاید ۴، صفحه ۱۱

13. فاز inception در متدولوژی UP معادل کدام فاز در مدل waterfall است؟

- 1. Modeling
- 2. Deployment
- 3. Planning
- 4. Communication

پاسخ: ۴

اسلاید ۴، صفحه ۳

14. کدام یک جزو معایب formal methods models به حساب نمی‌آید؟

- 1. زمانبر و پرهزینه بودن ایجاد مدل‌های فرمال
- 2. نیاز به آموزش گسترده افراد تیم ایجاد نرم‌افزار
- 3. استفاده از این مدل جهت ارتباط با مشتریانی که از نظر فنی دانش بالایی ندارند مناسب نیست.
- 4. هیچ کدام

پاسخ: د

کتاب، صفحه ۵۴

15. کدام مدل ایجاد نرم‌افزار از ویژگی iterative بودن پیروی نمی‌کند؟

- 1. Spiral
- 2. Unified Process
- 3. Concurrent
- 4. Waterfall

پاسخ: د

اسلاید ۴

16. کدام یک از موارد زیر نادرست است؟

- 1. در متودولوژی‌های قدیمی‌تر، هزینه‌ی تغییرات به صورت نمایی با جلو رفتن در پروژه افزایش می‌یافت.
- 2. اگر در میانه‌ی جمع‌آوری نیازمندی‌ها تغییری صورت بگیرد، بهتر از آن است که در هنگام ایجاد نرم‌افزار این اتفاق بیافتد.
- 3. در متودولوژی‌های agile، انجام دادن pair programming، داشتن unit testing و کشیدن نمودارهایی از قبیل UML برای مدل‌سازی ضروری است.
- 4. به علت وجود نداشتن (یا کم‌رنگ بودن) مراحل قبل از ایجاد، هزینه‌ی تغییر در ابتدای پروژه با متودولوژی‌های قدیمی‌تر، کمتر بوده است.

پاسخ: ۳

صفحه ۶۸ و ۶۹ کتاب پرسمن، بخش 5.2

17. کدام یک از گزاره‌های زیر، جزو اصول دوازده‌گانه‌ی agile نیست؟

- 1. سادگی (پیدا کردن ساده‌ترین روش برای انجام تسک‌ها) ضروری است.
- 2. توجه مداوم به برتری فنی و طراحی خوب باعث افزایش چابکی می‌شود.
- 3. نرم‌افزار قابل استفاده اصلی‌ترین معیار سنجش پیشرفت است.
- 4. بالاترین اولویت ما جلب رضایت مشتری با تحویل زود و مداوم نرم‌افزاری ارزشمند می‌باشد.

پاسخ: ۱

صفحات ۷۰ و ۷۱ کتاب پرسمن، بخش 5.3.1

18. کدام یک از متودولوژی‌های زیر، از خانواده‌ی agile نیست؟

- 1. UP
- 2. AUP
- 3. DSDM
- 4. IXP

پاسخ: ۱

کتاب پرسمن، بخش 5.4 و 5.5

19. کدام یک از گزینه‌های زیر، جزو Practice های مورد تاکید متدولوژی XP نیست؟

1. Tailoring Questionnaire

2. Iterative Development

3. Unit Testing

4. CRC Cards

پاسخ: ۱

گزینه‌ی ۱ از practice های مورد تاکید در متدولوژی DSDM است. بقیه موارد را می‌توانید در بخش 5.4 کتاب پرسمن پیدا کنید.

20. کدام یک از موارد زیر در مورد متدولوژی اسکرام درست است؟

1. یکی از مهم‌ترین سوالات جلسه‌ی روزانه، پرسیدن این سوال است که "برای فردا چه کاری قرار است انجام دهی؟".

2. Product owner مسئول هدایت جلسات روزانه‌ی اسکرام است.

3. فقط و فقط product owner می‌تواند تسک‌های موجود در اسپرینت فعلی را تغییر دهد.

4. بک‌لاگ محصول به صورت مرتب مورد بازنگری قرار می‌گیرد و اولویت‌بندی می‌شود.

پاسخ: ۴

صفحات ۷۸ و ۷۹ کتاب پرسمن بخش scrum

21. کدام یک از موارد زیر درست است؟

1. استفاده از اصل KISS بدین معناست که هر آنچه می‌توانید به محصول اضافه کنید تا بعداً با راحتی (simplicity) بتوانید آن‌ها را به پروژه اضافه کنید.

2. در متدولوژی اسکرام بعد از هر اسپرینت، می‌توانیم خواسته‌ی ذی‌نفعان را کامل تحویل دهیم.

3. در متدولوژی XP، سرعت پروژه (project velocity) معادل تعداد customer story هایی است که تیم در انتشار نسخه‌ی اول انجام داده است.

4. کارتهای CRC مکانیزمی برای فکر کردن به نرم‌افزار در زمینه‌های برنامه‌سازی functionall هستند.

پاسخ: ۳

صفحه‌ی ۷۳ و ۷۹ کتاب پرسمن

22. کدام یک از practice های زیر در متدولوژی IXP مفهوم اشتباهی درون خودش دارد؟
1. Readiness assessment: تیم مشخص می‌کند آیا تمامی اعضای پروژه (اعم از ذی‌نفعان، اعضای تیم ایجاد، مدیران) از شرایط اطلاع دارند و محیط درست را برای تیم قرار داده‌اند.
  2. Project community: تیم IXP مشخص می‌کند که آیا افراد درستی با توانایی درستی برای پروژه قرار داده شده‌اند یا خیر.
  3. Project chartering: تیم مطمئن می‌شود که این پروژه از لحاظ تکنولوژیکی قابل انجام است و تمامی ریسک‌های آن را در این پروژه قبل از شروع پیدا می‌کند.
  4. Retrospectives: در این جلسات در مورد مشکلات، پیشامدها و درس‌های یادگرفته شده صحبت می‌شود و review می‌کنیم.

پاسخ: ۳

صفحه ۷۶، تیم در این مورد دنبال توجیه بیزینسی (اقتصادی) است و دنبال توانایی انجام کار با تکنولوژی نیست.

23. تمامی موارد زیر از کارهایی است که در یک iteration متدولوژی XP انجام می‌شود، به جز...

1. Planning
2. Release
3. Design
4. Testing

پاسخ: ۲

صفحه ۷۲، شکل پایین صفحه

24. کدام یک از موارد زیر در مورد تعریف نقش‌ها در یک تیم نرم‌افزاری درست نیست؟
1. Ambassador: تیم را به حوزه‌های انتخابی بیرونی نمایش می‌دهد تا در صورت امکان اعضای تیم بتواند کاری در آن حوزه‌ها پیدا کند.
  2. Scouting: می‌تواند شامل جست‌وجو برای مارکت‌های بیرونی، یا جست‌وجو برای تکنولوژی‌های جدید باشد.
  3. Sentry: جریان داده‌ای که ذی‌نفعان و دیگر افراد به تیم ارسال می‌کند را کنترل می‌کند.
  4. Guard: از محصول تیم و محصولات اطلاعاتی دیگر محافظت می‌کند.

پاسخ: ۱

صفحه ۸۹ و ۹۰، بخش 6.2 کتاب پرسمن

25. کدام یک از حس‌های زیر، از حس‌های تیم موثر در کار ایجاد نرم‌افزار نیست؟

1. حس هدف داشتن
2. حس درگیری داخل پروژه
3. حس دوستی
4. حس پیشرفت

پاسخ: ۳

صفحه ۵ اسلاید فصل 6

26. توضیح کدام یک از paradigm‌های زیر درست نیست؟

1. closed paradigm: یک تیم را به مواضع قدرت به صورت سنتی پیش می‌برد.
2. random paradigm: تیم به صورت رندوم یکی از حالت‌های پیش رو را انتخاب می‌کند.
3. open paradigm: تلاش برای ساختن یک تیم به گونه‌ای که برخی از کنترل‌های مرتبط با پارادایم closed را به دست آورد، اما همچنین بسیاری از نوآوری‌هایی که هنگام استفاده از پارادایم random رخ می‌دهد.
4. synchronous paradigm: متکی به تقسیم طبیعی یک مشکل است و اعضای تیم را سازماندهی می‌کند تا روی تکه‌هایی از مشکل کار کنند و ارتباطات فعال کمی بین خودشان وجود دارد.

پاسخ: ۲

صفحه ۸ اسلاید فصل 6

27. کدام یک از موارد زیر نادرست است؟

1. برای تیم‌های Global ایجاد نرم‌افزار، هرچه فاصله بیشتر باشد، collaboration (همکاری) نیز بیشتر می‌شود.
2. یکی از ارزش‌های خیلی مهم تیم‌های XP، دادن Feedback به همدیگر است.
3. تمرکز بیشتر سرویس‌های cloud روی کارایی بوده تا امن بودن سیستم.
4. کار فردی همراه با همکاری گروهی است که به عنوان یکی از فاکتورهای ضروری برای موفقیت نامبرده می‌شود.

پاسخ: ۱

صفحه‌های ۹ و ۱۰ و ۱۲ و ۱۵، اسلاید فصل 6



28. کدام یک از کارهای زیر را باید قبل از دست به کد شدن انجام دهیم؟

1. باید داده‌ساختارهایی مناسب با design پیدا کنیم.
2. محیط برنامه‌نویسی‌ای را انتخاب کنیم که با توجه به ابزارهایی که در اختیارمان می‌گذارد، نیازهای محصول را راحت‌تر و سریع‌تر مرتفع کند.
3. معماری نرم‌افزار را متوجه شویم و interface هایی مطابق با آن معماری درست کنیم.
4. یک لایه‌ی بصری برای کمک به تفهیم محصول بسازیم.

پاسخ: ۲

صفحه‌های 20، 21، 22 اسلاید 7

29. کدام یک از موارد زیر نادرست است؟

1. دانش ایجاد نرم‌افزار عمری سه ساله دارد، اما اصول مهندسی نرم‌افزار عمری جاودان دارند.
2. پروسه یک تجربه‌ی دینی نیست که درونش دُگم باشیم. وقتی لازم است، باید رویکرد خود را با محدودیت‌های تحمیل شده توسط مسئله، افراد و خود پروژه تطبیق می‌دهیم.
3. جداسازی نگرانی‌ها (separation of concerns) یک practice است که با بخش‌بندی سیستم نرم‌افزاری پیچیدگی را مدیریت می‌کند، به طوری که هر بخش مسئولیت یک نگرانی جداگانه را بر عهده دارد و تا آنجا که ممکن است هم‌پوشانی نگرانی‌ها را به حداقل می‌رساند.
4. هر جلسه‌ای باید یک مدیر داشته باشد که مطمئن شود حرف‌ها دارد به شکل سازنده‌ای جلو می‌رود، اگر مخالفتی پیش آمد مداخله کند و مطمئن شود که بقیه‌ی اصول ارتباطی دنبال می‌شود.

پاسخ: ۱

صفحه‌های 2، 3، 5 و 7 از اسلاید 7

30. «داستان فیل» از مولانا را شنیده‌اید؟

داستان از این قرار است که در هندوستان فیلی آوردند و در یک خانه‌ی تاریک قرار دادند و مردمی که نمی‌دانستند آنجا چه خبر است، در خانه‌ی تاریک بر روی فیل دست می‌کشیدند و هر کس تصویری از فیل در ذهن خود می‌ساخت و بیان می‌کرد؛ به عنوان مثال کسی که بر روی خرطومش دست می‌کشید فیل را به شکل یک لوله تصور کرد. کسی که بر گوش فیل دست می‌کشید و او را مانند یک بادبزن مجسم کرد. فردی که دست به پای فیل، او را مانند یک ستون محکم تجسم کرد. فردی دیگری که دست بر کمر فیل کشید و فیل را مانند تخت تصور کرد.

به نظر شما این داستان با کدام یک از مراحل ایجاد نرم‌افزار مرتبط‌تر است؟

1. مدل‌سازی نیازمندی‌ها

2. ایجاد مدل‌های طراحی

3. پیاده‌سازی

4. آزمون و استقرار

پاسخ: ۱

فیل را مانند یک محصولی نرم‌افزاری که قرار است به زودی ساخته شود، تصور می‌کنیم. مشتری این محصول خواسته‌های متفاوتی دارد. او فقط می‌داند که پاهای این محصول باید محکم باشد، خرطومش باید دراز باشد، گوش‌هایش باید پهن باشد و ... مشتری ایده‌ای ندارد که در نهایت، خواسته‌های او چه چیزی از آب درمی‌آید، او صرفاً ویژگی‌هایی را که به نظرش می‌رسد، می‌گوید.

وظیفه‌ی مهندس نرم‌افزار این است که این نیازمندی‌های متفاوت مشتری (که در اصل همگی از یک محصول نرم‌افزاری هستند) را شناسایی و مدیریت کند. یکی از روش‌های شناسایی و مدیریت نیازمندی‌ها، مدل‌سازی آن‌ها است.

*"It's really complex to make something simple."*

- Jack Dorsey