

به نام خدا

تمرین یک

نیم سال ۱۴۰۲۱

توضیحات

- لطفاً پاسخ‌ها را به صورت تایپ شده در قالب فایل PDF، حداکثر تا ساعت ۲۳:۵۹ تاریخ تعیین شده در صفحه‌ی درس‌افزار درس بارگذاری نمایید.
- ذکر **نام و نام خانوادگی** به همراه **شماره دانشجویی** همه‌ی اعضای گروه، هم‌چنین **شماره‌ی تیم** در فایل PDF پاسخ‌ها ضروری است. در صورتی که نام هر یک از اعضای گروه در فایل پاسخ‌ها نباشد، به منزله عدم همکاری آن عضو در گروه و نارضایتی سایر هم‌گروهی‌ها محسوب شده و نمره تمرین برای آن فرد لحاظ نخواهد شد.
- در صورت ارسال پاسخ‌ها به صورت دست‌نویس تضمینی در تصحیح آن وجود نخواهد داشت.
- هدف درس مهندسی نرم‌افزار آشنایی شما با دنیای نرم‌افزار و افزایش مهارت تحلیل شماسست. استفاده از ربات‌های هوشمند مانند ChatGPT برای پاسخ‌دهی به سوالات، مغایر با اهداف گفته شده است؛ از این رو توصیه می‌کنیم که برای پاسخ‌دهی به تمرین‌ها از این ربات‌ها استفاده نکنید.
- تمرین از ۱۰۰ نمره است و ۳۰ نمره امتیازی دارد. نمرات اضافی هر تمرین فقط می‌تواند برای جبران نمرات از دست‌رفته‌ی سایر تمرین‌ها استفاده شود و به بخش‌های دیگر مانند آزمون‌ها منتقل نمی‌شود.
- سیاست ارسال با تاخیر برای این تمرین به صورت زیر است:
 - تا ۲۴ ساعت نمره‌ای کسر نمی‌شود.
 - پس از ۲۴ ساعت، به ازای هر ساعت تاخیر ۱ درصد نمره کسر می‌شود.

موفق باشید

تیم آموزش مهندسی نرم‌افزار

sharif.software.engineering@gmail.com

سوال ۱ (۱۰ نمره)

به نظرتان چرا دستیاران آموزشی درس مهندسی نرم افزار به این نتیجه رسیده اند که درس، پروژه یا تمرین عملی نداشته باشد؟

پاسخ:

تصمیمی که دستیاران آموزشی درس گرفته اند، لزوماً بهترین تصمیم نبوده، ولی دلایلی که برای آن داشته اند عبارتند از:

1. در درس مهندسی نرم افزار، جوانب مختلف این حوزه به صورت نه چندان عمیق مطرح می شود؛ از این رو، به دنبال این نیستیم که مبحثی را عمیق یاد بگیریم. پروژه، باعث یادگرفتن عمیق برخی از مباحث درس می شود، که این، هدف درس نیست.
2. یک پروژه ایجاد نرم افزار، دست کم چندین ماه یا حتی چندین سال طول می کشد. در طول چهار ماهی که در یک ترم داریم، امکان شبیه سازی یک پروژه واقعی نیست. توجه کنید که ما می خواهیم ساختن یک ساختمان ۳۰ یا ۳۰۰ طبقه را یاد بگیریم، نه یک ساختمان ۳ طبقه!
3. درسی به نام «آزمایشگاه مهندسی نرم افزار» وجود دارد که مکمل این درس است و در آن، می توانید با برخی از بخش های حوزه مهندسی نرم افزار به صورت عملی آشنا شوید.

سوال ۲ (۱۰ نمره)

تفاوت بین «مهندس نرم افزار»^۱ و «سازنده نرم افزار»^۲ چیست؟ این مقایسه را بر مبنای تفاوت بین ساختن یک ساختمان سه طبقه و ساختمان ۳۰ طبقه انجام دهید.

پاسخ:

مهندس نرم افزار کسی است که گام های مختلف فرآیند ایجاد نرم افزار (تحلیل، طراحی، پیاده سازی، آزمون و نگهداری) را می داند و با به کارگیری رویه های توصیه شده در این حوزه، یک نرم افزار با کمترین هزینه، بیشترین کیفیت و در سریع ترین زمان ممکن، تولید می کند. او مانند یک سازنده یک ساختمان ۳۰ طبقه است که فرآیند و لوازم ایجاد یک ساختمان محکم، قابل اعتماد و مهندسی شده را می داند.

سازنده نرم افزار کسی است که در گام پیاده سازی از فرآیند ایجاد نرم افزار، مسئولیت کدنویسی و پیاده سازی بخش های مختلف نرم افزار را که پیش تر طراحی شده، دارد. او دید محدودتری به یک مسئله (پیاده سازی نرم افزار) دارد و مانند مهندس نرم افزار، دید جامعی به کل فرآیند ایجاد نرم افزار ندارد. سازنده نرم افزار مانند بنا یا نقشه کش یک ساختمان سه طبقه است. درست است که وظیفه ی بنا بتأی و وظیفه ی نقشه کش،

^۱ Software Engineer

^۲ Software Developer

نقشه‌کشی است، اما آن‌ها می‌توانند با کمک‌گرفتن از هم، ساختمان سه طبقه‌ای را بسازند. ساختن یک ساختمان سه طبقه، پیچیدگی و حساسیت خاصی ندارد.

سوال ۳ (۱۰ نمره)

در ابتدای فصل اول کتاب پرسمن، ایشان داستانی را از حضور خود در یک شرکت بازی‌سازی آورده‌اند. این داستان را بخوانید. به‌نظرتان چرا آقای پرسمن این داستان را در ابتدای کتاب خود آورده است؟

پاسخ:

جواب‌ها می‌تواند متفاوت باشد، ولی نکات زیر در داستان پررنگ هستند:

- در سراسر نقاط جهان (نه فقط ایران) شرکت‌ها لزوماً از مفاهیم حوزه‌ی مهندسی نرم‌افزار، به‌صورت کامل استفاده نمی‌کنند؛ تا جایی که در آماری^۳ گفته می‌شد که ۷۰ درصد پروژه‌های ایجاد نرم‌افزار، موفقیت‌آمیز نیستند.
- یک شرکت می‌تواند سالانه میلیون‌ها دلار درآمد داشته باشد، ولی از اصول و رویه‌های مهندسی نرم‌افزار به‌طور کامل استفاده نکند، و تنها شبی از مفاهیم حوزه‌ی مهندسی نرم‌افزار را در کار خود به‌کار بگیرند.
- چنانچه از مفاهیم حوزه‌ی مهندسی نرم‌افزار استفاده نکنیم، روزی می‌رسد که نگهداری و توسعه‌ی نرم‌افزارها، گلوگاه پیشرفت شرکت می‌شود و شرکت را در رقابت با دیگران عقب می‌اندازد. برای درک بهتر، روزی را به‌یاد بیاورید که Netflix نسبت به Disney ناشناخته بود، ولی اکنون به لطف استفاده از رویه‌های مهندسی نرم‌افزار، از بهترین شرکت‌های streaming و پخش فیلم جهان است.^۴

سوال ۴ (۱۰ نمره)

چرا نیازمندی‌های پروژه اینقدر تغییر می‌کنند؟ آیا مشتری نمی‌داند چه می‌خواهد؟!

پاسخ:

موارد متعددی در تغییر نیازمندی‌ها اثرگذارند، از جمله:

- **نیازهای بازار** تغییر می‌کند و مشتری برای انطباق با بازار، نیازمندی‌های پروژه را تغییر می‌دهد.
- نیازمندی‌ها در ابتدا **نادقیق و مبهم** بودند و با گذشت زمان و جلوتررفتن پروژه، نیازهای واقعی شفاف و شناخته می‌شوند و در نتیجه، نیازمندی‌های قبلی تغییر می‌کنند.
- توجه کنید که بخشی از نادقیق بودن نیازمندی‌ها مربوط به مشتری است. این درست است که

^۳ مقدمه‌ی کتاب Learning Domain-driven Design

^۴ برای درک میزان خفن بودن شرکت Netflix، بدانید که این شرکت روزانه بیش از ۲۰.۰۰۰ استقرار در محصولات نرم‌افزاری خود دارد ([منبع](#))

- مشتری در ابتدا نیازمندی‌ها را کاملاً دقیق نمی‌داند، اما نادقیق بودن نیازمندی‌ها فقط به خاطر ندانستن مشتری نیست، بلکه به خاطر ماهیت تکاملی نیازمندی‌ها هم هست.
- با گذشت زمان، ریسک انجام بعضی از نیازمندی‌ها تغییر می‌کند و در نتیجه، شاید تصمیم گرفته شود که آن نیازمندی‌ها را انجام ندهیم.
- در روند پروژه، برخی از نیازمندی‌ها در مقابل یکدیگر قرار می‌گیرند و باید بر اساس ارزش کسب‌وکار⁵ بین آن‌ها انتخاب کرد؛ در حالی که این نیازمندی‌ها در ابتدا تراحمی با هم نداشتند.

سوال ۵ (۴۰ نمره)

به سوالات زیر ذیل حوزه‌ی مهندسی متدولوژی⁶ پاسخ دهید. به عنوان راهنمایی می‌توانید [این مطالب تکمیلی](#) در [کانال درس](#) را مطالعه کنید.

1. آیا «متدولوژی ایجاد نرم‌افزار» همان «فرایند ایجاد نرم‌افزار»⁷ است؟ از پاسخ خود با جزییات و تفصیل دفاع کنید.
2. در «مدل فرایند عمومی ایجاد نرم‌افزار»⁸ در کتاب پرسمن، پنج «فعالیت چارچوبی»⁹ معرفی می‌شود که یکی از آن‌ها مدل‌سازی است.
 - a. در این فعالیت به صورت کلی چه اقداماتی انجام می‌شود؟ امروزه در عمل¹⁰ چه زبان مدل‌سازی استاندارد شده است؟
 - b. توضیح دهید که خروجی‌های فعالیت مدل‌سازی، چگونه و با چه هدفی در فعالیت بعدی از فرایند عمومی ایجاد نرم‌افزار مورد استفاده قرار می‌گیرد.
 - c. آیا فعالیت مدل‌سازی با اصل ششم از [اصول دوازده‌گانه‌ی چابک](#) در تناقض است؟ از پاسخ خود با مثال دفاع کنید.

پاسخ:

1. خیر؛ فرآیند را می‌توان جزیی از متدولوژی دانست. فرایند نرم‌افزار به رویکرد اتخاذ شده در هنگامی که نرم‌افزار در حال مهندسی شدن می‌باشد، گفته می‌شود. فرایند نرم‌افزار تیم ایجاد را هدایت می‌کند تا از بین انبوه رویه‌های کاربردی (Practice) موجود در حوزه نرم‌افزار، تعدادی را اتخاذ کرده، با یک ترتیب خاصی انجام داده و در نهایت سیستم نرم‌افزاری با کیفیت تحویل دهد. **متدولوژی طبق** تعریف متشکل است از دو بخش اصلی: زبان مدل‌سازی و فرایند. فرایند - که توسط پرسمن هم معرفی شده و پیش‌تر در این محتوا نیز به آن اشاره شد - خود از چهار بخش متشکل است: فعالیت‌ها، مصنوعات (محصولات)، افراد و معیارهایی برای سنجش سه فاکتور پیشین فرایند [2].

⁵ Business

⁶ Methodology Engineering

⁷ Software Development Process

⁸ Generic Software Development Process Model

⁹ Framework Activity

¹⁰ de facto

Scrum و DSDM - که در فصل 5 کتاب درس نیز به آنها اشاره شده است - دو نمونه از ده‌ها متدولوژی موجود ایجاد نرم‌افزار می‌باشند.

2.

a. مثال Extreme Programming که پیش‌تر در [کانال_درس](#) مطرح شد به وضوح نشان می‌دهد که صراحت موجود در یک مدل بصری برای مهندسين بسیار قابل درک‌تر از هر گونه تعاملات کلامی و نوشتاری است. نرم‌افزار نیز از این موضوع مستثنی نبوده و مدل‌سازی در آن ارتباطات و انتقال پیام بین مهندسان را بسیار ساده و کارا می‌کند. در نگرش پیشین به مهندسی نرم‌افزار که هنوز هم کاربرد دارد - و از متدولوژی سنگین وزن UP نشئت می‌گیرد - هر تکرار از تولید نرم‌افزار برای حل یک مسئله، پنج جریان‌کاری¹¹ به ترتیب انجام خواهد شد: نیازمندی، تحلیل، طراحی، پیاده‌سازی و تست. پس از استخراج نیازمندی‌ها، مهندس نرم‌افزار با تمرکز بر ماهیت‌های مسئله، تلاش به مصور ساختن اجزای مسئله با مدل خواهد داشت. به چنین عملیاتی که کارکرد مسئله (برای مثال کار کردن اجزا و ماهیت‌های یک فروشگاه) توسط یک زبان مدل‌سازی تصویر می‌شود، مدل‌سازی تحلیل گفته می‌شود. سپس مدل‌های تحلیل تهیه شده در جریان کاری بعدی تدقیق¹² شده، به آنها جزئیات اضافه می‌شود. این جزئیات بالاخص شامل ماهیت‌های راه‌حل نرم‌افزاری (پایگاه‌داده، کارگزار، رابط کاربری و ...) خواهد بود.

برای مثال یک نموداری کلاسی UML در مرحله تحلیل احتمالاً شامل ماهیت‌های قابل مشاهده و درک درون یک فروشگاه است، اما در مرحله طراحی بسیاری از کلاس‌های مربوط به کار با Database، خدمت‌رسانی در UI، نما¹³های ارتباط بین ماژول‌ها، واسط¹⁴ها و کلاس‌های انتزاعی¹⁵ ایجاد شده برای برقراری اصل DIP¹⁶ و بسیاری از کلاس‌ها و روابط دیگری که فقط در راه‌حل نرم‌افزاری دارای موضوعیت‌اند، به آن مدل اضافه خواهد شد. در نهایت در جریان‌کاری بعدی مدل طراحی عیناً پیاده‌سازی می‌شود. همچنین گرایش موسوم به مهندسی مدل‌رانه¹⁷ تمرکز بر ایجاد ابزارهای تبدیل خودکار مدل به کد نیز دارد. مدل‌سازی واضحاً علاوه بر همه این‌ها، در مستندسازی و ثبت و تصویر کردن اطلاعات نیز کاربرد دارد.

زبان مدل‌سازی در عمل استاندارد شده نیز، بعد از جنگ متدولوژی‌ها در دهه 90 میلادی، UML بود. این زبان حاصل گردهم‌آوری زبان‌های مدل‌سازی متدولوژی‌های پیشین بود که توسط رامبا، بوچ و جیکوبسن¹⁸ صورت پذیرفت.

b. همانطور که در بخش a ذکر شد، مدل‌ها با یک نگرش دیرینه در مهندسی نرم‌افزار، در واقع نقشه راهی برای پیاده‌سازی می‌شوند. حتی این گذر از مدل به کد می‌تواند به صورت بی‌درز

¹¹ Workflow

¹² Refine

¹³ Facade

¹⁴ Interface

¹⁵ Abstract

¹⁶ Dependency Inversion Principle

¹⁷ Model-Driven Engineering

¹⁸ Rumbaugh, Booch, and Jacobson

(که آرزوی مهندسی نرم‌افزار در بسیاری از ابعاد آن است) و با استفاده از ابزارهای تبدیل مدل به کد صورت پذیرد.

c. علی‌رغم اینکه چابک‌های اولیه به اصطلاح مدل‌هراس (Model-Phobic) بودند، استفاده از هرگونه مدل‌سازی را نهی کرده و پیرش به پیاده‌سازی را توصیه می‌کردند، اما در چابک‌های جدیدتر (به عنوان مثال DAD - Disciplined Agile Delivery) مدل‌سازی حتی به عنوان رویه کاربردی اجباری نهادینه شده است. همچنین بسیاری از فرایندهای چابک دیگر از زبان مدل‌سازی موسوم به «مدل‌سازی چابک» استفاده می‌کنند. اصل ششم که بر «مکالمه رو در رو» تمرکز دارد در ابتدا به اشتباه برداشت می‌شد و جامعه مهندسی را به سمت فاصله گرفتن از مدل‌سازی سوق داد. دومین بند از منشور چابک (رجحان نرم‌افزار کارا بر مستندات قابل درک) نیز به این آتش دامن می‌زد. اما رفته رفته جامعه مهندسی و مدیریت انسانی متوجه کمبودهای انتقال اطلاعات تنها از روش مکالمه را دریافت و مدل‌سازی به عنوان مکمل برای ارتباطات رو در رو قرار گرفته است.

سوال ۶ (۵۰ نمره)

1. (۵ نمره) تفاوت «مدل» ایجاد نرم‌افزار مانند آبشاری¹⁹ یا حلزونی²⁰ با «متدولوژی» ایجاد نرم‌افزار مانند XP یا RUP چیست؟

پاسخ کلی: «مدل» درشت‌دانه‌تر و عمومی‌تر است، ولی «متدولوژی» با جزئیات بیشتر و اختصاصی‌تر است. متدولوژی concrete تر از مدل است.

2. (۲۵ نمره) انجمن علمی دانشکده مهندسی کامپیوتر از شما خواسته تا «مدل»ای برای برگزاری رویدادهای دانشجویی ایجاد کنید. برخی از ویژگی‌های رویدادهای دانشجویی انجمن علمی دانشکده مهندسی کامپیوتر به صورت زیر است:

- حق‌الزحمه‌ای به نیروهای برگزارکننده پرداخت نمی‌شود.
- از آنجایی که نیروهای برگزارکننده تعهد رسمی به رویداد ندارند، احتمال دارد که هر کدامشان وظایفش را انجام ندهد.
- برگزارکنندگان رویداد دانشجویان هستند. دانشجویان وقت محدودی در طول سال تحصیلی دارند.
- موضوعات رویداد حول مباحث رشته‌ی مهندسی کامپیوتر است.
- آورده‌ی رویداد برای برگزارکنندگان، در وهله‌ی اول یادگیری مباحث مختلف و در وهله‌ی بعد، لذت‌بردن از انجام کار تیمی است.
- مخاطبین رویدادها اکثراً دانشجویان یا دانش‌آموزان هستند.

¹⁹ Waterfall

²⁰ Spiral

هم‌چنین در طراحی این مدل به موارد زیر توجه کنید:

- g. جامعه مخاطبین
- h. ثبت‌نام مخاطبین
- i. جذب داوطلبین برگزاری
- j. انتخاب افراد داوطلب
- k. تخمین هزینه‌ها
- l. حامی مالی²¹
- m. تبلیغات و برندینگ
- n. خط زمانی²² رویداد
- o. هماهنگی‌های اداری

راهنمایی:

1. می‌توانید برای دریافت اطلاعات بیشتر، به اتاق انجمن علمی در لابی دانشکده مراجعه کنید. هم‌چنین می‌توانید مستندات رویدادهای قبلی برگزارشده‌ی انجمن را پیدا و از آن‌ها استفاده کنید. توجه کنید که فهمیدن دامنه‌ی مسئله، کمک بزرگی به شما برای ایجاد مدل برگزاری رویداد می‌کند.
2. برای درک بهتر مفهوم «مدل»، می‌توانید در مورد مدل‌های زیر جستجو کنید:
 - a. مدل‌های مدیریت رویداد (Event Management Models) (مثال)
 - b. مدل‌های یادگیری (Learning Models) (مثال)
 - c. مدل‌های رشد شهری (Urban Growth Models) (مثال)
 - d. مدل‌های ایجاد محصول (Product Development Models) (مثال)

خروجی:

خروجی شما در این قسمت، باید سندی شامل توضیحات و تصاویر مربوط به مدل پیشنهادی‌تان باشد که اگر در اختیار اعضای انجمن علمی دانشکده مهندسی کامپیوتر قرار بگیرد، به آن‌ها در برگزاری بهتر رویدادها کمک کند.

معیارهای نمره‌دهی:

- چند تا فاز/بخش کلی داریم
- در هر فاز/بخش چه کارهایی انجام می‌شود
- یک تصویر جهت شفاف‌سازی آورده شود
- مدل باید درجه آزادی داشته باشد و دست استفاده‌کننده را تا حد زیادی باز بگذارد
 - کلی باشد و کلیت را بگوید

²¹ Sponsor

²² Timeline

3. (۲۰ نمره) با توجه به مدلی که در قسمت قبل تهیه کرده‌اید، «متدولوژی»ای برای برگزاری یک رویداد خاص ایجاد کنید. شما باید موقعیت²³ خاصی را در نظر بگیرید، به صورت دقیق ویژگی‌های آن را بیان کنید و در متدولوژی خود، به ویژگی‌های این موقعیت پاسخ دهید.

راهنمایی:

1. در [این اسلاید](#)، به ۱۳ معیار برای تحلیل متدولوژی‌های ایجاد نرم‌افزار پرداخته شده است. می‌توانید از این معیارها برای طراحی متدولوژی برگزاری رویداد خود ایده بگیرید. توجه کنید که این معیارها برای تحلیل متدولوژی‌های ایجاد نرم‌افزار بیان شده‌اند، در حالی که شما باید متدولوژی برگزاری یک رویداد دانشجویی را بسازید!
2. می‌توانید از [منشور جابک](#) و اصول Lean نیز برای طراحی متدولوژی برگزاری رویداد خود ایده بگیرید.

خروجی:

توجه کنید که در این سوال از شما توقع نداریم متدولوژی کامل، جامع و دقیقی برای موقعیت خاص مدنظرتان طراحی کنید. این کار نیاز به زمان بیشتری از یک تمرین دارد. خروجی شما در این قسمت، باید سندی شامل توضیحات و تصاویر مربوط به متدولوژی پیشنهادی‌تان باشد که اگر در اختیار اعضای انجمن علمی دانشکده مهندسی کامپیوتر قرار بگیرد، بتوانند با کمک مستندات شما، در موقعیت خاصی که توصیف کرده‌اید، رویداد خوبی را برگزار کنند.

معیارهای نمره‌دهی:

- متدولوژی پیشنهادی باید دقیق‌ترشده‌ی مدل پیشنهادی در قسمت قبل باشد
- درجه آزادی داشته باشد، ولی کمتر از مدل بخش قبل باشد
- یک موقعیت/رویداد خاص را توصیف کند
- فرآیند، راه‌کار و practice بیان کرده باشد
- محصولاتی²⁴ را به عنوان خروجی گام‌های مختلف متدولوژی تعریف کرده باشد
- نقش‌های موجود در رویداد و وظایفشان را بیان کرده باشد
- اشاره‌ای به مدل‌سازی کرده باشد



Duplication is the primary enemy of a well-designed system.

- Robert Martin

²³ Situation

²⁴ Artifacts