

۱۳۹۹/۸/۳



## تمرین دوم



### مهندسی نرم افزار ۲

گروه {۲}

اعضاء گروه:

(۱) محمدرضا اخگری زیری ۹۶۳۱۰۰۱

(۲) محمدعلی کشاورز ۹۶۳۱۰۶۱

(۳) علی نظری ۹۶۳۱۰۷۵



## ۱) دسته‌بندی<sup>۱</sup> نیازمندی‌ها

الف) چرا نیازمندی‌ها را دسته‌بندی می‌کنیم؟ کارکرد(های) اصلی تعریف دسته‌بندی‌ها چیست؟

**پاسخ:** دسته‌بندی نیازمندی‌ها کمک می‌کند که بتوانیم تصمیم بگیریم که برای هر نیازمندی، با توجه به این که در کدام دسته‌بندی(ها) قرار می‌گیرد، چه روشی را در استخراج و تضمین کیفیت نیازمندی‌ها در پیش بگیریم.

برای مثال یکی از دسته‌بندی‌های ارائه شده برای نیازمندی‌ها دسته‌بندی زیر است:

- عملکردی<sup>۲</sup>
- غیرعملکردی<sup>۳</sup>
- دامنه‌ای.

شناخت نوع و دسته‌بندی نیازمندی‌ها، سبب می‌شود که بتوانیم نیازمندی‌ها را به صورت دقیق‌تر تحلیل کنیم؛ مثلاً چنانچه دریابیم که نیازمندی از نوع دامنه‌ای است، می‌دانیم که برای روشن شدن نیازمندی و همچنین تحلیل درستی پیاده‌سازی نیازمندی نیاز داریم تا فردی مسلط به دامنه موضوع به کمک تیم توسعه نرم‌افزار بیاید؛ یا اگر نیازمندی از نوع غیرعملکردی بود درمی‌یابیم که نیاز است متریک‌هایی برای سنجش عملکرد نرم‌افزار در آن نیازمندی، تعریف کنیم.

ب) پس از مطالعه دسته‌بندی‌های موجود و با توجه به پاسخ قسمت قبل، دسته‌بندی پیشنهادیتان را معرفی کرده (با ذکر دلایل انتخاب یا تعریف دسته‌بندی جدید) و مختصراً توضیح دهید. دسته‌بندی پیشنهادی ممکن است شامل یک یا چند دسته‌بندی موجود یا یک دسته‌بندی کاملاً جدید یا برگرفته از دسته‌بندی‌های موجود باشد.

**پاسخ:** پس از مطالعه دسته‌بندی‌های موجود، دیدیم که دسته‌بندی قسمت قبل، به صورت کلی شامل دسته‌های مناسبی بودند، اما بیش از حد انتزاعی<sup>۴</sup> بودند و ما تصمیم گرفتیم که به هر دسته جزئی‌تر بپردازیم و زیربخش‌هایی به آن اضافه کنیم. در نهایت دسته‌بندی پیشنهادی ما به شرح زیر است:

<sup>1</sup> Categorization

<sup>2</sup> Functional

<sup>3</sup> Non-functional

<sup>4</sup> Abstract



- عملکردی
  - غیرعملکردی
    - کاربرد پذیری<sup>۵</sup>
    - قابلیت اطمینان<sup>۶</sup>
    - کارایی
    - قابلیت پشتیبانی
  - دامنه‌ای (ممکن است شامل دسته‌های عملکردی و غیرعملکردی باشد)
  - پیاده‌سازی (شامل محدودیت‌های ساخت، زبان برنامه‌نویسی مورد استفاده، معماری، سیستم‌های خارجی، محدودیت‌های فیزیکی و...)
- \* این دسته‌بندی الهام گرفته از دسته‌بندی **FURPS Plus** و دسته‌بندی گفته شده در قسمت الف است.

ج) برای هر کدام از دسته‌های (قابل اعمال) دسته‌بندی پیشنهادی، یک مثال از سیستم پروژه گروهتان بیان کنید.

#### پاسخ:

عملکردی: بتوان تخته<sup>۷</sup> جدید ایجاد کرد و هر کدام از کارت<sup>۸</sup>ها را بین استیت‌های مختلف جابجا کند.  
غیرعملکردی: با زیاد شدن تعداد درخواست‌ها بتواند هر درخواست را در حداکثر ۲۰۰ میلی‌ثانیه، پاسخ دهد. این نیازمندی مربوط به زیردسته‌بندی **کارایی** است.  
دامنه‌ای: در حال حاضر نیازمندی دامنه‌ای برای پروژه ما وجود نداشت.  
پیاده‌سازی: با توجه به ویژگی‌های گروه ما استفاده از تکنولوژی **React.js** و **NetCore**. برای پیاده‌سازی به عنوان نیازمندی پیاده‌سازی به‌شمار می‌آید.

<sup>5</sup> Usability

<sup>6</sup> Reliability

<sup>7</sup> Board

<sup>8</sup> Card



۲) یکی از روش‌های شناسایی نیازمندی<sup>۹</sup>، استفاده از سناریوها<sup>۱۰</sup>، به عنوان مبنای فعالیت‌های شناسایی، است<sup>۱۱</sup>.  
(الف) به نظر شما نقاط ضعف استفاده از سناریو در شناخت نحوه تعامل کاربران با سیستم چیست؟ راه حل پیشنهادی شما برای رفع مشکلات احتمالی چیست؟

**پاسخ:** طبق تعریف، سناریوها با زاویه دید کاربرها نوشته می‌شوند. ممکن است ما ویژگی‌های خلاقانه‌ای داشته باشیم که کاربران تا به حال آن‌ها را ندیده باشند و در نتیجه در مورد استفاده و کاربرد آن نتوانیم به مطلب سودمندی از زاویه دید کاربران برسیم.

برای حل این مشکل می‌توانیم تصمیم‌گیری برای این ویژگی‌های خلاقانه را به بعد از انتشار محصول موکول کنیم تا درباره نحوه‌ی تعامل کاربران با محصولمان اطلاعاتی در دست داشته باشیم و با نوشتن سناریوهای جدید محصول را به صورت پیوسته بهبود دهیم.

<sup>9</sup> Requirements Elicitation

<sup>10</sup> Scenarios

<sup>11</sup> Scenario-based Requirements Elicitation



ب) پس از مطالعه (اجمالی!) فصل سوم از کتاب <sup>12</sup> Engineering Software Products، تکنیک مطرح شده برای شناسایی و تعریف ویژگی‌ها<sup>۱۳</sup> را مختصراً توضیح دهید.

موارد آبی برای اعضای تیم است که برای خواندن دوباره تمرین مشکل نداشته باشند.

پاسخ:

در این فصل از کتاب، ابتدا مفاهیم پرسنا<sup>۱۴</sup>، سناریو<sup>۱۵</sup> و داستان<sup>۱۶</sup> معرفی شدند و در ادامه نیز روش تشخیص ویژگی‌ها<sup>۱۷</sup> با کمک داستان‌ها توضیح داده شد. ما ابتدا تکنیک مطرح شده را که از مفاهیم سناریو و داستان برای شناسایی و تعریف ویژگی‌ها، استفاده می‌کند را توضیح می‌دهیم و در انتها نیز برای درک بهتر، این مفاهیم را شرح می‌دهیم.

می‌خواهیم یک لیست از ویژگی‌ها برای پیاده‌سازی تهیه کنیم؛ به این صورت که باید عملیات<sup>۱۸</sup>ها و کار<sup>۱۹</sup>هایی که در سناریورها و داستان‌ها می‌بینیم را استخراج کنیم. این موارد استخراج شده برای پیاده‌سازی باید دقیق‌تر شوند، تا بتوان آن‌ها را ویژگی نامید. اگرچه این ویژگی‌ها باید مستقل از هم باشند ولی هرکدام باید به یک عملیات اصلی مرتبط باشند.

باید توجه کنیم که منظور ما از ویژگی، بخشی از یک عملیات است که کاربر ما در هنگام استفاده از محصول ما به آن احتیاج پیدا می‌کند.

پس از تهیه لیست ویژگی‌ها، توصیه می‌شود که برای هر کدام از این ویژگی‌ها یک توضیح به همراه محدودیت‌ها و سایر نظرات در مورد آن تهیه کنیم.

در قسمت بعدی (ج) به صورت واضح‌تر و با مثال به این مفاهیم و شناسایی ویژگی‌ها می‌پردازیم.

- پرسنا: باید توجه کنیم که ما خودمان (احتمالاً) مخاطب محصولی که داریم تولید می‌کنیم، نیستیم و باید مخاطبانمان را بهتر و دقیق‌تر بشناسیم. پرسنا در حقیقت یک تصویر نسبتاً شفاف و فرضی از کسانی است که فکر می‌کنیم از محصول ما استفاده می‌کنند. حالا می‌توانیم به این فکر کنیم که یک پرسنا چگونه در هر شرایطی با محصول ما برخورد می‌کند و به ما کمک می‌کند که تنها براساس افکار خودمان برای کاربرانمان تصمیم‌گیری نکنیم.

<sup>12</sup> کتاب از طریق لینک زیر در دسترس است: [yun.ir/jt5g5f](http://yun.ir/jt5g5f)

<sup>13</sup> Features

<sup>14</sup> Persona

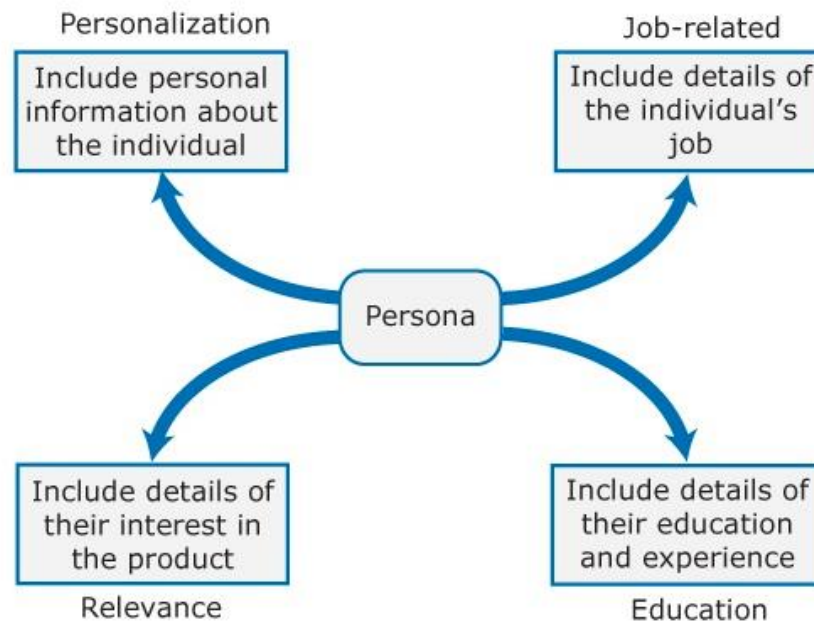
<sup>15</sup> Scenario

<sup>16</sup> Story

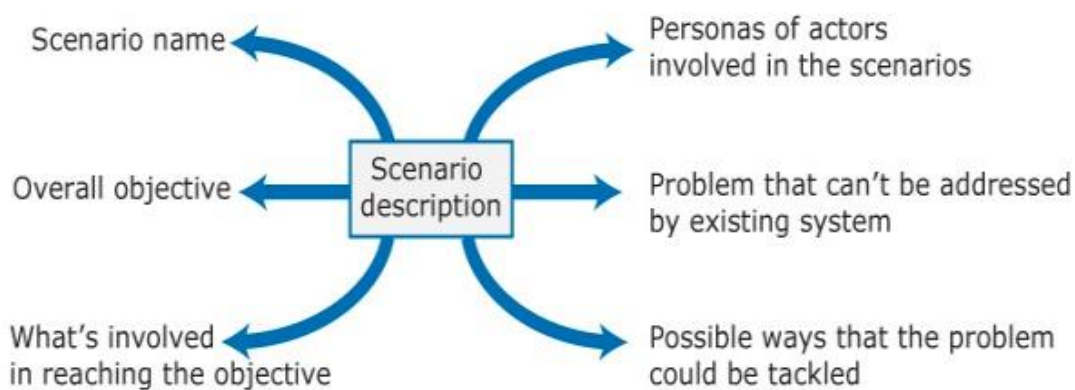
<sup>17</sup> Features

<sup>18</sup> Functionality

<sup>19</sup> Action



- **سناریو:** سناریو توضیحی سطح بالا است که از طریق آن شرایطی که یک کاربر برای انجام کار خودش از یکی از ویژگی‌های محصول ما استفاده می‌کند را بیان می‌کنیم. سناریو باید با زاویه دید خود کاربر به مسئله بپردازد و نیازی نیست که خیلی جزئی باشد زیرا به معنی مشخصات محصول ما نیست و بیشتر برای کاربران قابل فهم است. همچنین برای هر کدام از پرسنالهایی که مشخص کردیم می‌توانیم چند سناریو بنویسیم و محدودیتی نداریم.

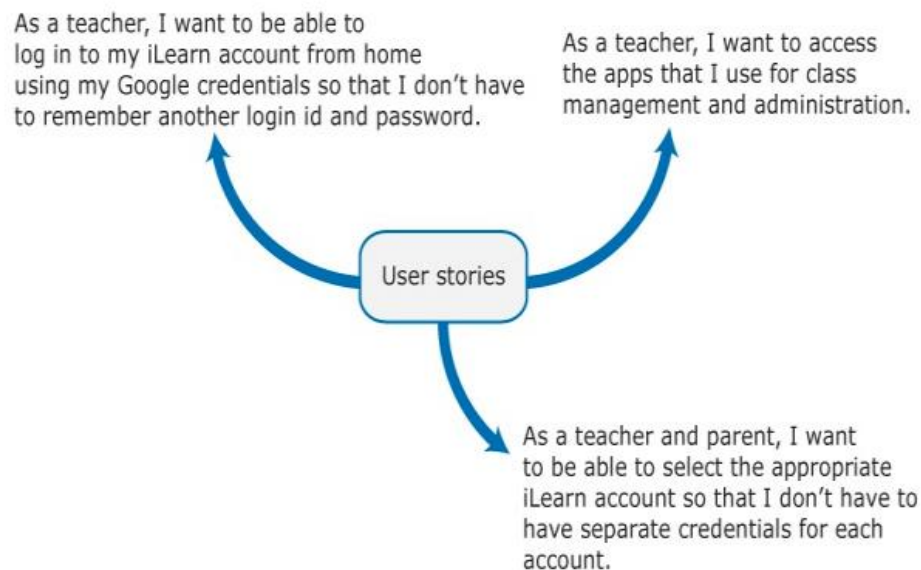


- **داستان:** داستان‌ها یک توضیح دقیق‌تر و شفاف‌تر در مورد کارهایی است که کاربران از سیستم ما می‌خواهند که به صورتی ساختاریافته<sup>۲۰</sup> نوشته می‌شود و جزئیات آن از سناریوها بیشتر است و حتی می‌توان آن‌ها را با دقیق‌تر کردن سناریوها تهیه کرد.

<sup>20</sup> structured



داستان‌ها کار ما را برای برنامه‌ریزی آسان‌تر می‌کنند زیرا به نوعی هستند که می‌توان آن‌ها را در مدت یک اسپرینت پیاده‌سازی کرد و توضیحات کافی در مورد هر ویژگی را شامل می‌شوند. ساختار داستان‌ها به این صورت است که مشخص می‌کند، هر نقشی می‌خواهد در قالب این ویژگی چه کاری را (و با چه هدفی) انجام دهد.





ج) برای سیستم مورد نظرتان، یک پرسنا، یک سناریو (برای پرسنای نمونه) به همراه دو داستان (مرتبط با سناریو) و نهایتاً لیست ویژگی‌های استخراج شده از داستان‌ها را بنویسید.

• **پرسنا:** سعید (اسکرام مستر یک تیم نرم افزاری)

سعید ۲۹ سال دارد و ۲ سال است که به عنوان اسکرام مستر و مدتی هم به عنوان مدیر محصول در شرکت خرمالو (هلو سابق) مشغول به کار است.

سعید مدرک مهندسی نرم‌افزار دارد و در گذشته نیز به عنوان برنامه‌نویس فعالیت می‌کرده اما با گذشت زمان به جایگاهی که اکنون در آن حضور دارد علاقه پیدا کرده است و کمی حوزه فعالیتش را تغییر داده است.

او به عنوان اسکرام‌مستر در تیمش باید فعالیت‌های تیم را تحت نظر داشته باشد و نحوه انجام کارهای تیم را مدیریت کند و برای این کار از نرم افزارهای مدیریت پروژه استفاده می‌کند.

• **سناریو:** مشاهده‌ی روند انجام فعالیت‌ها

سعید که به عنوان اسکرام مستر در تیم فعالیت می‌کند، می‌خواهد روند انجام فعالیت‌های تیم را مشاهده کند. در تیمی که او حضور دارد قسمتی از فعالیت‌ها مربوط به برنامه‌نویسی سمت سرور و بخشی نیز مربوط به برنامه‌نویسی سمت کلاینت است.

او هر روز به سراغ تخته (board) فعالیت‌ها می‌رود و می‌بیند که هر کدام از وظایفی که به اعضای تیم محول شده بود در کدام لیست هستند.

• **داستان‌ها**

- به عنوان اسکرام‌مستر، من می‌خواهم ببینم که هر کدام از وظایف در چه مرحله‌ای هستند تا بتوانم آن‌ها را پیگیری کنم.
- به عنوان اسکرام مستر، من نیاز دارم که بتوانم وظایف جدید را برای اعضای تعریف کنم.

• **لیست ویژگی‌ها**

- وجود یک سری لیست از وظایف
- وظایف می‌توانند در لیست‌هایی مجزا قرار بگیرند تا از این طریق بر اساس وضعیت انجام آن‌ها دسته بندی شوند.
- وجود تگ بر روی وظایف
- با کمک تگ‌ها می‌توان مشخص کرد که هر وظیفه مربوط به کدام بخش از تیم است.





تمرین دوم

مهندسی نرم افزار ۲

دکتر طارمی راد



دانشکده مهندسی  
کامپیوتر و فناوری اطلاعات

- وجود نام افراد مسئول هر وظیفه  
با کدام این ویژگی مشخص می شود که متعلق به چه کسی است که بتوان به سهولت پیگیری نیز انجام  
شود.



۳) سه تکنیک تضمین کیفیت<sup>۲۱</sup> نیازمندی‌ها را مختصراً معرفی کنید. مزایا و کاربرد هر کدام را بررسی کنید.

### (امتیازی)

موارد آبی برای اعضای تیم است که برای خواندن دوباره تمرین مشکل نداشته باشند. تضمین کیفیت، یکی از مهم‌ترین بخش‌های توسعه نرم‌افزار است، به طور سنتی، تکنیک‌های تضمین کیفیت عمدتاً در مراحل بعدی توسعه، مانند پیاده‌سازی و تست‌های مربوط به آن، متمرکز شده‌اند؛ اما این فرآیند باید زودتر و در مرحله مهندسی نیازمندی‌ها شروع شود.

اهمیت تشخیص نقص در نیازمندی‌ها بدین جهت است که این نقص‌ها نه تنها در نیازمندی‌ها تاثیر می‌گذارند، بلکه ممکن است تاثیراتی در مراحل بعد بگذارد و اشتباهاتی برای معماری<sup>۲۲</sup>، طراحی<sup>۲۳</sup>، کد<sup>۲۴</sup> و تست ایجاد کند (پس تضمین کیفیت فقط در مراحل پیاده‌سازی، هزینه‌ی زیادی خواهد داشت).

برای اینکه بتوان درباره‌ی تضمین کیفیت بحث کرد، ابتدا باید بتوان کیفیت را تعریف کرد، به همین جهت ویژگی‌های مورد نیاز برای کیفیت با توجه به استاندارد IEEE استفاده می‌کنیم (موارد و جنبه مورد بحث در کیفیت در جدول ۸-۱ مقاله منبع آمده است).

مشخص کردن اینکه آیا موارد مورد بحث در نیازمندی‌ها (آن‌هایی که در این مرحله قابل بحثند) ویژگی‌های مورد نیاز برای کیفیت را دارد یا خیر، تضمین کیفیت نیازمندی‌ها نام دارد.

تضمین کیفیت به طور کلی دو رویکرد کلی دارد: رویکردهای سازنده<sup>۲۵</sup> و رویکردهای تحلیلی<sup>۲۶</sup>. چون رویکردهای سازنده اطمینان حاصل می‌کنند که اشتباهات در هنگام ایجاد یک محصول کاری به حداقل می‌رسد (به عنوان مثال مشخصات نیازمندی‌ها). یعنی از انتشار مشکلات جلوگیری می‌کنند. برای تضمین کیفیت نیازمندی‌ها مناسب است.

<sup>21</sup> Quality Assurance

<sup>22</sup> Architecture

<sup>23</sup> Design

<sup>24</sup> Code

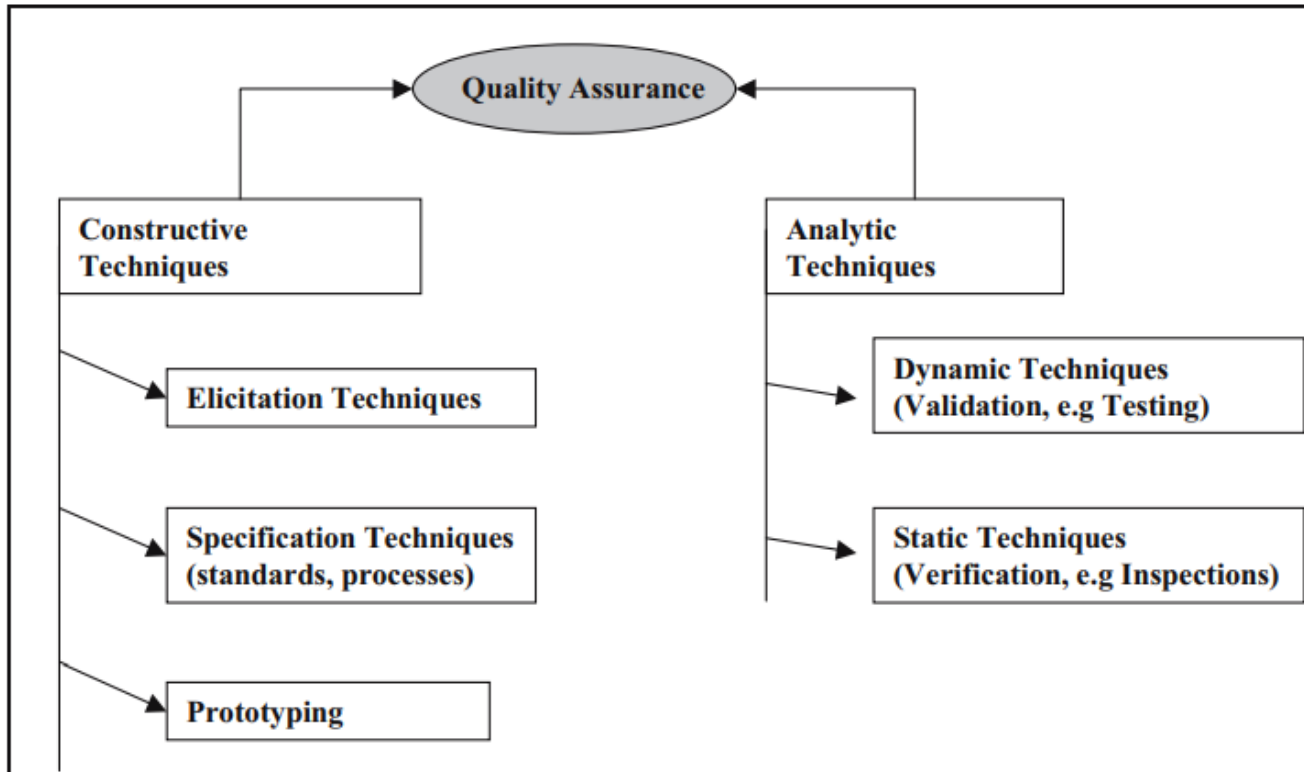
<sup>25</sup> Constructive approaches

<sup>26</sup> Analytical approaches



رویکردهای تحلیلی هم شامل دو بخش پویا<sup>۲۷</sup> (نیازمند به نسخه قابل اجرا) و بخش ایستا<sup>۲۸</sup> (بدون نیاز به اجرای کد) است. پس می توان از رویکردهای ایستا برای تضمین کیفیت نیازمندی ها استفاده کرد (زیرا معمولاً در این مرحله کدی نیست).

(بند آخر مورد قسمت ۸.۲ در منبع توضیح خوبی داده است)



شکل ۱- گزیده رویکردهای تضمین کیفیت

### ۱. تکنیک های استخراج<sup>۲۹</sup>:

مرحله استخراج برای کیفیت کلی نیازمندی ها و پذیرش سیستم نهایی مهم است. در طول مرحله استخراج، نیازها از منابع مختلف مانند مشتری، کاربران، پروژه های قبلی، مطالعات بازار و... گرفته می شود. در این فرایند، ذینفعان مختلف مانند مشتریان، کادر فنی (توسعه دهندگان) و کاربران نهایی با هم کار می کنند برای به دست آوردن یک مجموعه مناسب از الزامات. مهندسين نیازمندی می توانند تکنیک های مختلفی را برای حمایت از ذینفعان مختلف در کشف الزامات استفاده کنند، به عنوان مثال مصاحبه ها، پرسشنامه ها، کارگاه ها و گروه ها. با استفاده از تکنیک های استخراج، می توان از ویژگی های کیفیت زیر اطمینان حاصل کرد:

<sup>27</sup> dynamic

<sup>28</sup> Static

<sup>29</sup> Elicitation Techniques



- **قابل درک بودن:** با تدوین اصطلاحات مشترک و اطمینان از صحبت کردن ذینفعان مختلف به یک زبان، درک مطلب بهبود می‌یابد.
- **کامل بودن:** در صورت انجام صحیح استخراج، باید کلیه ذینفعان و سهم فردی آن‌ها مشخص شود. در اینجا، فعالیت های استخراج به کیفیت بالاتری کمک می‌کنند زیرا از مهندسان مورد نیاز در فرایندهای شناسایی پشتیبانی می‌کنند.
- **راستی آزمایی و امکان‌سنجی:** باز هم با درگیر شدن ذینفعان مربوطه می‌توان کیفیت را تضمین کرد. با درگیر کردن آزمایشگرها قابلیت راستی‌آزمایی ویژگی بهبود می‌یابد و با درگیر کردن توسعه‌دهندگان امکان‌سنجی بهبود می‌یابد.
- **صحت:** مناسب بودن، به عنوان بخشی از صحت، توسط این امر پشتیبانی می‌شود، زیرا احتمالاً نرم‌افزار توسعه‌یافته سود واقعی مالی را در زمینه استفاده به ارمغان می‌آورد.

## ۲. تکنیک‌های مشخصات<sup>۳۰</sup>:

هدف اصلی مرحله مشخصات، مستند کردن نیازها به گونه‌ای است که بتوان از آن‌ها به عنوان پایه‌ای برای توسعه استفاده کرد. معمولاً، خروجی فعالیت مشخصات یک سند نیازمندی است که جنبه مربوط به سیستم ساخته شده را ضبط می‌کند (به عنوان مثال جنبه‌های عملکردی، غیرعملکردی، محدودیت‌ها و...). با استفاده از تکنیک‌های استخراج، می‌توان از ویژگی‌های کیفیت زیر اطمینان حاصل کرد:

- **کامل بودن:** در صورتی که مهندسان نیازمندی به توصیه‌های استاندارد پردازند و الگوهای از پیش تعریف شده را اعمال کنند، می‌توان اطمینان حاصل کرد که تمام جنبه‌های مربوط به یک سند نیازمندی، یعنی کامل بودن سند در نظر گرفته شده است.
- **قابل فهم و تغییر پذیری:** ساختار ارائه شده توسط الگوها و استانداردها تضمین می‌کند که سند مورد نیاز از نظر پروژه‌های مختلف در یک شرکت مشابه است. استانداردسازی اسناد نیازمندی از ابهام جلوگیری می‌کند.

## ۳. نمونه‌سازی اولیه<sup>۳۱</sup>:

نمونه اولیه یک نسخه قابل اجرا از سیستم در دست توسعه است. هدف یک نمونه اولیه این است که ذینفعان بتوانند سیستم را امتحان کنند و پیشنهادهای بهبود را ارائه دهند. با این کار، آن‌ها درک بهتری از اینکه آیا سیستم، سیستم مورد نیاز آنها را نشان می‌دهد، پیدا می‌کنند و بنابراین به شناسایی نیازهای از دست‌رفته و

<sup>30</sup> Specification Techniques

<sup>31</sup> Prototyping



تشخیص موارد غلط کمک می‌کند. مهمترین ارزش یک نمونه اولیه این است که فاصله بین توصیف و اجرا را از بین می‌برد. بعلاوه، یک مسئله کاملاً رایج در مورد نیازمندی‌ها این است که مشتری اغلب دقیقاً نمی‌داند که چه می‌خواهد.

با استفاده از نمونه‌سازی اولیه، می‌توان از ویژگی‌های کیفیت زیر اطمینان حاصل کرد:

- **ناسازگاری و ناقص بودن:** فرآیند تولید نمونه اولیه، در خود، ناسازگاری و ناقص بودن نیازمندی‌ها را نشان می‌دهد و در نتیجه کیفیت آن‌ها را بهبود می‌بخشد.
- **صحت:** صحت با اجازه دادن به ذینفعان مختلف برای کار با یک شی مشخص و ارزیابی آن به جای نیازهای انتزاعی، بهبود می‌یابد.
- **امکان سنجی:** با آزمایش راه‌حل‌های مختلف، در مرحله نیازها، امکان‌سنجی بهبود می‌یابد. در صورت شناسایی بن‌بست‌ها در مراحل اولیه، می‌توانید در وقت و هزینه زیادی صرفه‌جویی کنید.

درباره‌ی تکنیک‌های ایستا هم می‌توان بسط داد که در منبع آمده است.

منبع

[8 Quality Assurance in Requirements Engineering, Christian Denger, Thomas Olsson](#)

- پاسخ تمرین‌ها را به زبان فارسی و به صورت تایپ شده، در قالب یک فایل Pdf، در مودل بارگزاری کنید.
- سوالات خود را می‌توانید از طریق ایمیل از دستیاران تدریس بپرسید.
- فایل پاسخ تمرین را تنها با قالب **SE2-HW2-Group[#].pdf** در مودل بارگزاری کنید.
- بارگزاری تمرین توسط یکی از اعضای گروه کافی است.
- برای پاسخ‌های هر قسمت منابع استفاده شده را درج نمایید.
- فایل زیپ ارسال نکنید.
- به ازای هر روز تاخیر در تحویل تمرین ۲۰٪ از نمره تمرین کسر خواهد شد.
- حداقل برخورد به پاسخ‌های مشابه، تخصیص نمره کامل منفی به طرفین خواهد بود.